Package 'Surrogate'

July 21, 2025

Type Package

Title Evaluation of Surrogate Endpoints in Clinical Trials

Version 3.4.1

Description In a clinical trial, it frequently occurs that the most credible outcome to evaluate the effectiveness of a new therapy (the true endpoint) is difficult to measure. In such a situation, it can be an effective strategy to replace the true endpoint by a (bio)marker that is easier to measure and that allows for a prediction of the treatment effect on the true endpoint (a surrogate endpoint). The package 'Surrogate' allows for an evaluation of the appropriateness of a candidate surrogate endpoint based on the meta-analytic, information-theoretic, and causal-inference frameworks. Part of this software has been developed using funding provided from the European Union's Seventh Framework Programme for research, technological development and demonstration (Grant Agreement no 602552), the Special Research Fund (BOF) of Hasselt University (BOF-number: BOF2OCPO3), GlaxoSmithKline Biologicals, Baekeland Mandaat (HBC.2022.0145), and Johnson & Johnson Innovative Medicine.

Imports MASS, lattice, latticeExtra, survival, nlme, lme4, logistf, rms, ks, extraDistr, pbapply, flexsurv, rvinecopulib, maxLik, purrr, MBESS, tidyr, dplyr, tibble, lifecycle

Depends R (>= 3.5.0)

License GPL (>= 2)

Repository CRAN

NeedsCompilation no

RoxygenNote 7.3.1

Encoding UTF-8

Suggests copula, FNN, mgcv, testthat (>= 3.0.0), vdiffr, parallel, fitdistrplus, kdecopula, cubature, mvtnorm, withr, stringr, ggplot2, sn, pracma

Config/testthat/edition 3

URL https://github.com/florianstijven/Surrogate-development

BugReports https://github.com/florianstijven/Surrogate-development/issues

Author Wim Van Der Elst [cre, aut],
Florian Stijven [aut],
Fenny Ong [aut],
Dries De Witte [aut],
Gokce Deliorman [aut],
Paul Meyvisch [aut],
Alvaro Poveda [aut],
Ariel Alonso [aut],
Hannah Ensor [aut],
Christoper Weir [aut],
Geert Molenberghs [aut]

Maintainer Wim Van Der Elst <wim.vanderelst@gmail.com>

Date/Publication 2025-04-29 04:40:02 UTC

Contents

`AA.MultS` *Compute the multiple-surrogate adjusted association*

Description

The function `AA.MultS` computes the multiple-surrogate adjusted correlation. This is a generalisation of the adjusted association proposed by Buyse & Molenberghs (1998) (see `Single.Trial.RE.AA`) to the setting where there are multiple endpoints. See **Details** below.

Usage

```
AA.MultS(Sigma_gamma, N, Alpha=0.05)
```

Arguments

`Sigma_gamma` The variance covariance matrix of the residuals of regression models in which the true endpoint (T) is regressed on the treatment (Z), the first surrogate ($S1$) is regressed on Z, ..., and the k-th surrogate (Sk) is regressed on Z. See **Details** below.

`N` The sample size (needed to compute a CI around the multiple adjusted association; γ_M)

`Alpha` The α-level that is used to determine the confidence interval around γ_M. Default 0.05.

Details

The multiple-surrogate adjusted association (γ_M) is obtained by regressing T, $S1$, $S2$, ..., Sk on the treatment (Z):

$$T_j = \mu_T + \beta Z_j + \varepsilon_{Tj},$$

$$S1_j = \mu_{S1} + \alpha_1 Z_j + \varepsilon_{S1j},$$

$$\ldots,$$

$$Sk_j = \mu_{Sk} + \alpha_k Z_j + \varepsilon_{Skj},$$

where the error terms have a joint zero-mean normal distribution with variance-covariance matrix:

$$\mathbf{\Sigma} = \left(\begin{array}{cc} \sigma_{TT} & \Sigma_{\boldsymbol{ST}} \\ \Sigma_{\boldsymbol{ST}}^{'} & \Sigma_{\boldsymbol{SS}} \end{array} \right).$$

The multiple adjusted association is then computed as

$$\gamma_M = \sqrt{(}\frac{\left(\Sigma_{ST}^{'}\Sigma_{SS}^{-1}\Sigma_{ST}\right)}{\sigma_{TT}})$$

Value

An object of class `AA.MultS` with components,

`Gamma.Delta` An object of class `data.frame` that contains the multiple-surrogate adjusted association (i.e., γ_M), its standard error, and its confidence interval (based on the Fisher-Z transformation procedure).

`Corr.Gamma.Delta` An object of class `data.frame` that contains the bias-corrected multiple-surrogate adjusted association (i.e., corrected γ_M), its standard error, and its confidence interval (based on the Fisher-Z transformation procedure).

`Sigma_gamma` The variance covariance matrix of the residuals of regression models in which T is regressed on Z, $S1$ is regressed on Z, ..., and Sk is regressed on Z.

`N` The sample size (used to compute a CI around the multiple adjusted association; γ_M)

`Alpha` The α-level that is used to determine the confidence interval around γ_M.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Buyse, M., & Molenberghs, G. (1998). The validation of surrogate endpoints in randomized experiments. *Biometrics, 54,* 1014-1029.

Van der Elst, W., Alonso, A. A., & Molenberghs, G. (2017). A causal inference-based approach to evaluate surrogacy using multiple surrogates.

See Also

`Single.Trial.RE.AA`

Examples

```
data(ARMD.MultS)

# Regress T on Z, S1 on Z, ..., Sk on Z
# (to compute the covariance matrix of the residuals)
Res_T <- residuals(lm(Diff52~Treat, data=ARMD.MultS))
Res_S1 <- residuals(lm(Diff4~Treat, data=ARMD.MultS))
Res_S2 <- residuals(lm(Diff12~Treat, data=ARMD.MultS))
Res_S3 <- residuals(lm(Diff24~Treat, data=ARMD.MultS))
Residuals <- cbind(Res_T, Res_S1, Res_S2, Res_S3)

# Make covariance matrix of residuals, Sigma_gamma
Sigma_gamma <- cov(Residuals)

# Conduct analysis
Result <- AA.MultS(Sigma_gamma = Sigma_gamma, N = 188, Alpha = .05)
```

```
# Explore results
summary(Result)
```

ARMD — *Data of the Age-Related Macular Degeneration Study*

Description

These are the data of a clinical trial involving patients suffering from age-related macular degeneration (ARMD), a condition that involves a progressive loss of vision. A total of 181 patients from 36 centers participated in the trial. Patients' visual acuity was assessed using standardized vision charts. There were two treatment conditions (placebo and interferon-α). The potential surrogate endpoint is the change in the visual acuity at 24 weeks (6 months) after starting treatment. The true endpoint is the change in the visual acuity at 52 weeks.

Usage

```
data(ARMD)
```

Format

A `data.frame` with 181 observations on 5 variables.

`Id` The Patient ID.

`Center` The center in which the patient was treated.

`Treat` The treatment indicator, coded as -1 = placebo and 1 = interferon-α.

`Diff24` The change in the visual acuity at 24 weeks after starting treatment. This endpoint is a potential surrogate for `Diff52`.

`Diff52` The change in the visual acuity at 52 weeks after starting treatment. This outcome serves as the true endpoint.

ARMD.MultS — *Data of the Age-Related Macular Degeneration Study with multiple candidate surrogates*

Description

These are the data of a clinical trial involving patients suffering from age-related macular degeneration (ARMD), a condition that involves a progressive loss of vision. A total of 181 patients participated in the trial. Patients' visual acuity was assessed using standardized vision charts. There were two treatment conditions (placebo and interferon-α). The potential surrogate endpoints are the changes in the visual acuity at 4, 12, and 24 weeks after starting treatment. The true endpoint is the change in the visual acuity at 52 weeks.

Usage

```
data(ARMD.MultS)
```

Format

A `data.frame` with 181 observations on 6 variables.

`Id` The Patient ID.

`Diff4` The change in the visual acuity at 4 weeks after starting treatment. This endpoint is a potential surrogate for `Diff52`.

`Diff12` The change in the visual acuity at 12 weeks after starting treatment. This endpoint is a potential surrogate for `Diff52`.

`Diff24` The change in the visual acuity at 24 weeks after starting treatment. This endpoint is a potential surrogate for `Diff52`.

`Diff52` The change in the visual acuity at 52 weeks after starting treatment. This outcome serves as the true endpoint.

`Treat` The treatment indicator, coded as -1 = placebo and 1 = interferon-α.

`association_gof_copula` *Produce Associational GoF plot*

Description

Produce Associational GoF plot

Usage

```
association_gof_copula(
  fitted_submodel,
  treat,
  endpoint_types,
  return_data = FALSE,
  grid = NULL,
  ...
)
```

Arguments

`fitted_submodel`
List returned by `fit_copula_submodel_OrdCont()`, `fit_copula_submodel_ContCont()`, or `fit_copula_submodel_OrdOrd()`.

`treat` Value for the treatment indicator.

`endpoint_types` Character vector with 2 elements indicating the type of endpoints. Each element is either `"ordinal"` or `"continuous"`.

`return_data` (boolean) Return the data used in the goodness-of-fit plot (without the plot itself). This is useful when the user wants to customize the plots, e.g., using `ggplot2`. See Details.

`grid` (numeric) vector of values for the (surrogate) endpoint at which the regression function is evaluated.

`...` Extra argument passed onto `plot()`.

Semi-Parametric Regression estimates

See the documentation of `plot.vine_copula_fit()` for the default semi-parametric estimators.

Return Plotting Data

If `return_data` is `TRUE`, this function will return a data frame that can be used to create customized plots. The following variables are present in the returned data frame:

- `observed`: The semi-parametric estimate of the regression function

$$E(T|S)$$

.

- `upper_ci`, `lower_ci`: Upper and lower limit of the pointwise 95% confidence interval for the semi-parametric estimate of the regression function.
- `value`: Value for the surrogate endpoint at which the estimates for the regression function are evaluated.
- `model_based`: Model-based estimate of the regression function.

See Also

`plot.vine_copula_fit()`

BifixedContCont *Fits a bivariate fixed-effects model to assess surrogacy in the meta-analytic multiple-trial setting (Continuous-continuous case)*

Description

The function `BifixedContCont` uses the bivariate fixed-effects approach to estimate trial- and individual-level surrogacy when the data of multiple clinical trials are available. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below. Further, the Individual Causal Association (ICA) is computed.

Usage

```
BifixedContCont(Dataset, Surr, True, Treat, Trial.ID, Pat.ID, Model=c("Full"),
Weighted=TRUE, Min.Trial.Size=2, Alpha=.05, T0T1=seq(-1, 1, by=.2),
T0S1=seq(-1, 1, by=.2), T1S0=seq(-1, 1, by=.2), S0S1=seq(-1, 1, by=.2))
```

Arguments

`Dataset` A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.

`Surr` The name of the variable in `Dataset` that contains the surrogate endpoint values.

`True` The name of the variable in `Dataset` that contains the true endpoint values.

`Treat` The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.

`Trial.ID` The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.

`Pat.ID` The name of the variable in `Dataset` that contains the patient's ID.

`Model` The type of model that should be fitted, i.e., `Model=c("Full")`, `Model=c("Reduced")`, or `Model=c("SemiReduced")`. See the **Details** section below. Default `Model=c("Full")`.

`Weighted` Logical. If `TRUE`, then a weighted regression analysis is conducted at stage 2 of the two-stage approach. If `FALSE`, then an unweighted regression analysis is conducted at stage 2 of the two-stage approach. See the **Details** section below. Default `TRUE`.

`Min.Trial.Size` The minimum number of patients that a trial should contain in order to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R^2_{trial}, R_{trial}, R^2_{indiv} and R_{indiv}. Default 0.05.

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ_{Δ} (ICA). For details, see function `ICA.ContCont`. Default `seq(-1, 1, by=.2)`.

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.2)`.

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.2)`.

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.2)`.

Details

When the full bivariate mixed-effects model is fitted to assess surrogacy in the meta-analytic framework (for details, Buyse & Molenberghs, 2000), computational issues often occur. In that situation, the use of simplified model-fitting strategies may be warranted (for details, see Burzykowski et al., 2005; Tibaldi et al., 2003).

The function `BifixedContCont` implements one such strategy, i.e., it uses a two-stage bivariate fixed-effects modelling approach to assess surrogacy. In the first stage of the analysis, a bivariate linear regression model is fitted. When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), the following bivariate model is fitted:

$$S_{ij} = \mu_{Si} + \alpha_i Z_{ij} + \varepsilon_{Sij},$$
$$T_{ij} = \mu_{Ti} + \beta_i Z_{ij} + \varepsilon_{Tij},$$

where S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_{Si} and μ_{Ti} are the fixed trial-specific intercepts for S and T, and α_i and β_i are the trial-specific treatment effects on S and T, respectively. When a reduced model is requested (by using the argument `Model=c("Reduced")` in the function call), the following bivariate model is fitted:

$$S_{ij} = \mu_S + \alpha_i Z_{ij} + \varepsilon_{Sij},$$
$$T_{ij} = \mu_T + \beta_i Z_{ij} + \varepsilon_{Tij},$$

where μ_S and μ_T are the common intercepts for S and T (i.e., it is assumed that the intercepts for the surrogate and the true endpoints are identical in all trials). The other parameters are the same as defined above.

In the above models, the error terms ε_{Sij} and ε_{Tij} are assumed to be mean-zero normally distributed with variance-covariance matrix $\mathbf{\Sigma}$:

$$\mathbf{\Sigma} = \left(\begin{array}{cc} \sigma_{SS} & \\ \sigma_{ST} & \sigma_{TT} \end{array} \right).$$

Based on $\mathbf{\Sigma}$, individual-level surrogacy is quantified as:

$$R^2_{indiv} = \frac{\sigma^2_{ST}}{\sigma_{SS}\sigma_{TT}}.$$

Next, the second stage of the analysis is conducted. When a full model is requested by the user (by using the argument `Model=c("Full")` in the function call), the following model is fitted:

$$\widehat{\beta_i} = \lambda_0 + \lambda_1 \widehat{\mu_{Si}} + \lambda_2 \widehat{\alpha_i} + \varepsilon_i,$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on the full model that was fitted in stage 1.

When a reduced or semi-reduced model is requested by the user (by using the arguments `Model=c("Reduced")` or `Model=c("SemiReduced")` in the function call), the following model is fitted:

$$\widehat{\beta_i} = \lambda_0 + \lambda_1 \widehat{\alpha_i} + \varepsilon_i.$$

where the parameter estimates for β_i and α_i are based on the semi-reduced or reduced model that was fitted in stage 1.

When the argument `Weighted=FALSE` is used in the function call, the model that is fitted in stage 2 is an unweighted linear regression model. When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), the information that is obtained in stage 1 is weighted according to the number of patients in a trial.

The classical coefficient of determination of the fitted stage 2 model provides an estimate of R^2_{trial}.

Value

An object of class `BifixedContCont` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Results.Stage.1` The results of stage 1 of the two-stage model fitting approach: a `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`Residuals.Stage.1` A `data.frame` that contains the residuals for the surrogate and true endpoints that are obtained in stage 1 of the analysis (ε_{Sij} and ε_{Tij}).

`Results.Stage.2` An object of class `lm` (linear model) that contains the parameter estimates of the regression model that is fitted in stage 2 of the analysis.

`Trial.R2` A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval.

`Indiv.R2` A `data.frame` that contains the individual-level coefficient of determination (R^2_{indiv}), its standard error and confidence interval.

`Trial.R` A `data.frame` that contains the trial-level correlation coefficient (R_{trial}), its standard error and confidence interval.

`Indiv.R` A `data.frame` that contains the individual-level correlation coefficient (R_{indiv}), its standard error and confidence interval.

`Cor.Endpoints` A `data.frame` that contains the correlations between the surrogate and the true endpoint in the control treatment group (i.e., ρ_{T0S0}) and in the experimental treatment group (i.e., ρ_{T1S1}), their standard errors and their confidence intervals.

D.Equiv The variance-covariance matrix of the trial-specific intercept and treatment effects for the surrogate and true endpoints (when a full or semi-reduced model is fitted, i.e., when Model=c("Full") or Model=c("SemiReduced") is used in the function call), or the variance-covariance matrix of the trial-specific treatment effects for the surrogate and true endpoints (when a reduced model is fitted, i.e., when Model=c("Reduced") is used in the function call). The variance-covariance matrix D.Equiv is equivalent to the $\boldsymbol{D}$ matrix that would be obtained when a (full or reduced) bivariate mixed-effect approach is used; see function BimixedContCont).

Sigma The 2 by 2 variance-covariance matrix of the residuals (ε_{Sij} and ε_{Tij}).

ICA A fitted object of class ICA.ContCont.

T0T0 The variance of the true endpoint in the control treatment condition.

T1T1 The variance of the true endpoint in the experimental treatment condition.

S0S0 The variance of the surrogate endpoint in the control treatment condition.

S1S1 The variance of the surrogate endpoint in the experimental treatment condition.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

Tibaldi, F., Abrahantes, J. C., Molenberghs, G., Renard, D., Burzykowski, T., Buyse, M., Parmar, M., et al., (2003). Simplified hierarchical linear models for the evaluation of surrogate endpoints. *Journal of Statistical Computation and Simulation, 73,* 643-658.

See Also

UnifixedContCont, UnimixedContCont, BimixedContCont, plot Meta-Analytic

Examples

```
## Not run:  # time consuming code part
# Example 1, based on the ARMD data
data(ARMD)

# Fit a full bivariate fixed-effects model with weighting according to the
# number of patients in stage 2 of the two stage approach to assess surrogacy:
Sur <- BifixedContCont(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Trial.ID=Center,
Pat.ID=Id, Model="Full", Weighted=TRUE)

# Obtain a summary of the results
summary(Sur)
```

```
# Obtain a graphical representation of the trial- and individual-level surrogacy
plot(Sur)


# Example 2
# Conduct a surrogacy analysis based on a simulated dataset with 2000 patients,
# 100 trials, and Rindiv=Rtrial=.8
# Simulate the data:
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8, R.Indiv.Target=.8,
Seed=123, Model="Reduced")

# Fit a reduced bivariate fixed-effects model with no weighting according to the
# number of patients in stage 2 of the two stage approach to assess surrogacy:
\dontrun{ #time-consuming code parts
Sur2 <- BifixedContCont(Dataset=Data.Observed.MTS, Surr=Surr, True=True, Treat=Treat,
Trial.ID=Trial.ID, Pat.ID=Pat.ID, , Model="Reduced", Weighted=FALSE)

# Show summary and plots of results:
summary(Sur2)
plot(Sur2, Weighted=FALSE)}

## End(Not run)
```

`BimixedCbCContCont` *Fits a bivariate mixed-effects model using the cluster-by-cluster (CbC) estimator to assess surrogacy in the meta-analytic multiple-trial setting (Continuous-continuous case)*

Description

The function `BimixedCbCContCont` uses the cluster-by-cluster (CbC) estimator of the bivariate mixed-effects to estimate trial- and individual-level surrogacy when the data of multiple clinical trials are available. See the **Details** section below.

Usage

```
BimixedCbCContCont(Dataset, Surr, True, Treat, Trial.ID,Min.Treat.Size=2,Alpha=0.05)
```

Arguments

`Dataset` A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, and a trial ID.

`Surr` The name of the variable in `Dataset` that contains the surrogate endpoint values.

`True` The name of the variable in `Dataset` that contains the true endpoint values.

`Treat` The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.

`Trial.ID` The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.

`Min.Treat.Size` The minimum number of patients in each group (control or experimental) that a trial should contain to be included in the analysis. If the number of patients in a group of a trial is smaller than the value specified by `Min.Treat.Size`, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R^2_{trial} and R^2_{indiv}. Default 0.05.

Details

The function `BimixedContCont` fits a bivariate mixed-effects model using the CbC estimator (for details, see Florez et al., 2019) to assess surrogacy (for details, see Buyse et al., 2000). In particular, the following mixed-effects model is fitted:

$$S_{ij} = \mu_S + m_{Si} + (\alpha + a_i)Z_{ij} + \varepsilon_{Sij},$$
$$T_{ij} = \mu_T + m_{Ti} + (\beta + b_i)Z_{ij} + \varepsilon_{Tij},$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_S and μ_T are the fixed intercepts for S and T, m_{Si} and m_{Ti} are the corresponding random intercepts, α and β are the fixed treatment effects for S and T, and a_i and b_i are the corresponding random treatment effects, respectively.

The vector of the random effects (i.e., m_{Si}, m_{Ti}, a_i and b_i) is assumed to be mean-zero normally distributed with variance-covariance matrix $\boldsymbol{D}$:

$$\boldsymbol{D} = \left(\begin{array}{cccc} d_{SS} & & & \\ d_{ST} & d_{TT} & & \\ d_{Sa} & d_{Ta} & d_{aa} & \\ d_{Sb} & d_{Tb} & d_{ab} & d_{bb} \end{array}\right).$$

The trial-level coefficient of determination (i.e., R^2_{trial}) is quantified as:

$$R^2_{trial} = \frac{\left(\begin{array}{c} d_{Sb} \\ d_{ab} \end{array}\right)^{'} \left(\begin{array}{cc} d_{SS} & d_{Sa} \\ d_{Sa} & d_{aa} \end{array}\right)^{-1} \left(\begin{array}{c} d_{Sb} \\ d_{ab} \end{array}\right)}{d_{bb}}.$$

The error terms ε_{Sij} and ε_{Tij} are assumed to be mean-zero normally distributed with variance-covariance matrix $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma} = \left(\begin{array}{cc} \sigma_{SS} & \\ \sigma_{ST} & \sigma_{TT} \end{array}\right).$$

Based on $\boldsymbol{\Sigma}$, individual-level surrogacy is quantified as:

$$R^2_{indiv} = \frac{\sigma^2_{ST}}{\sigma_{SS}\sigma_{TT}}.$$

Note The CbC estimator for the full bivariate mixed-effects model is closed-form (for details, see Florez et al., 2019). Therefore, it is fast. Furthermore, it is recommended when computational issues occur with the full maximum likelihood estimator (implemented in function `BimixedContCont`).

The CbC estimator is performed in two stages: (1) a linear model is fitted in each trial. Evidently, it is require that the design matrix (X_i) is full column rank within each trial, allowing estimation of the fixed effects. When X_i is not full rank, trial i is excluded from the analysis. (2) a global estimator of the fixed effects (β) is obtained by weighted averaging the sets of estimates of each trial, and $\boldsymbol{D}$ is estimated using a method-of-moments estimator. Optimal weights (for details, see Molenberghs et al., 2018) are used as a weighting scheme.

The estimator of $\boldsymbol{D}$ might lead to a non-positive-definite solution. Therefore, the eigenvalue method (for details, see Rousseeuw and Molenberghs, 1993) is used for non-positive-definiteness adjustment.

Value

An object of class `BimixedContCont` with components,

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (after excluding clusters). Clusters are excluded for two reasons: (i) the number of patients is smaller than the value especified by `Min.Trial.Size`, and (ii) the design matrix (X_i) is not full rank.

`Trial.removed` Number of trials excluded from the analysis

`Fixed.Effects` A `data.frame` that contains the fixed intercept and treatment effects for the true and the surrogate endpoints (i.e., μ_S, μ_T, α, and β) and their corresponding standard error.

`Trial.R2` A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval.

`Indiv.R2` A `data.frame` that contains the individual-level coefficient of determination (R^2_{indiv}), its standard error and confidence interval.

`D` The variance-covariance matrix of the random effects (the $\boldsymbol{D}$ matrix), i.e., a 4 by 4 variance-covariance matrix of the random intercept and treatment effects.

`DH.pd` `DH.pd=TRUE` if an adjustment for non-positive definiteness was not needed to estimate $\boldsymbol{D}$. `DH.pd=FALSE` if this adjustment was required.

`Sigma` The 2 by 2 variance-covariance matrix of the residuals (ε_{Sij} and ε_{Tij}).

Author(s)

Alvaro J. Florez, Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

Florez, A. J., Molenberghs G, Verbeke G, Alonso, A. (2019). A closed-form estimator for meta-analysis and surrogate markers evaluation. *Journal of Biopharmaceutical Statistics, 29(2)* 318-332.

Molenberghs, G., Hermans, L., Nassiri, V., Kenward, M., Van der Elst, W., Aerts, M. and Verbeke, G. (2018). Clusters with random size: maximum likelihood versus weighted estimation. *Statistica Sinica, 28*, 1107-1132.

Rousseeuw, P. J. and Molenberghs, G. (1993) Transformation of non positive semidefinite correlation matrices. *Communications in Statistics, Theory and Methods, 22*, 965-984.

See Also

`BimixedContCont`, `UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`

Examples

```
# Open the Schizo dataset (clinial trial in schizophrenic patients)
data(Schizo)

# Fit a full bivariate random-effects model by the cluster-by-cluster (CbC) estimator
# a minimum of 2 subjects per group are allowed in each trial
  fit <- BimixedCbCContCont(Dataset=Schizo, Surr=BPRS, True=PANSS, Treat=Treat,Trial.ID=InvestId,
                              Alpha=0.05, Min.Treat.Size = 10)
# Note that an adjustment for non-positive definiteness was requiered and 113 trials were removed.

# Obtain a summary of the results
summary(fit)
```

`BimixedContCont` *Fits a bivariate mixed-effects model to assess surrogacy in the meta-analytic multiple-trial setting (Continuous-continuous case)*

Description

The function `BimixedContCont` uses the bivariate mixed-effects approach to estimate trial- and individual-level surrogacy when the data of multiple clinical trials are available. The user can specify whether a full or reduced model should be fitted. See the **Details** section below. Further, the Individual Causal Association (ICA) is computed.

Usage

```
BimixedContCont(Dataset, Surr, True, Treat, Trial.ID, Pat.ID, Model=c("Full"),
Min.Trial.Size=2, Alpha=.05, T0T1=seq(-1, 1, by=.2), T0S1=seq(-1, 1, by=.2),
T1S0=seq(-1, 1, by=.2), S0S1=seq(-1, 1, by=.2), ...)
```

Arguments

`Dataset` A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.

`Surr` The name of the variable in `Dataset` that contains the surrogate endpoint values.

`True` The name of the variable in `Dataset` that contains the true endpoint values.

`Treat` The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.

`Trial.ID` The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.

`Pat.ID` The name of the variable in `Dataset` that contains the patient's ID.

`Model` The type of model that should be fitted, i.e., `Model=c("Full")` or `Model=c("Reduced")`. See the **Details** section below. Default `Model=c("Full")`.

`Min.Trial.Size` The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R^2_{trial}, R_{trial}, R^2_{indiv} and R_{indiv}. Default 0.05.

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ_{Δ} (ICA). For details, see function `ICA.ContCont`. Default `seq(-1, 1, by=.2)`.

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.2)`.

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.2)`.

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.2)`.

`...` Other arguments to be passed to the function `lmer` (of the R package `lme4`) that is used to fit the geralized linear mixed-effect models in the function `BimixedContCont`.

Details

The function `BimixedContCont` fits a bivariate mixed-effects model to assess surrogacy (for details, see Buyse et al., 2000). In particular, the following mixed-effects model is fitted:

$$S_{ij} = \mu_S + m_{Si} + (\alpha + a_i)Z_{ij} + \varepsilon_{Sij},$$
$$T_{ij} = \mu_T + m_{Ti} + (\beta + b_i)Z_{ij} + \varepsilon_{Tij},$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_S and μ_T are the fixed intercepts for S and T, m_{Si} and m_{Ti} are the corresponding random intercepts, α and β are the fixed treatment effects for S and T, and a_i and b_i are the corresponding random treatment effects, respectively.

The vector of the random effects (i.e., m_{Si}, m_{Ti}, a_i and b_i) is assumed to be mean-zero normally distributed with variance-covariance matrix $\boldsymbol{D}$:

$$\boldsymbol{D} = \begin{pmatrix} d_{SS} & & & \\ d_{ST} & d_{TT} & & \\ d_{Sa} & d_{Ta} & d_{aa} & \\ d_{Sb} & d_{Tb} & d_{ab} & d_{bb} \end{pmatrix}.$$

The trial-level coefficient of determination (i.e., R^2_{trial}) is quantified as:

$$R^2_{trial} = \frac{\begin{pmatrix} d_{Sb} \\ d_{ab} \end{pmatrix}' \begin{pmatrix} d_{SS} & d_{Sa} \\ d_{Sa} & d_{aa} \end{pmatrix}^{-1} \begin{pmatrix} d_{Sb} \\ d_{ab} \end{pmatrix}}{d_{bb}}.$$

The error terms ε_{Sij} and ε_{Tij} are assumed to be mean-zero normally distributed with variance-covariance matrix $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{SS} & \\ \sigma_{ST} & \sigma_{TT} \end{pmatrix}.$$

Based on $\boldsymbol{\Sigma}$, individual-level surrogacy is quantified as:

$$R^2_{indiv} = \frac{\sigma^2_{ST}}{\sigma_{SS}\sigma_{TT}}.$$

Note

When the full bivariate mixed-effects approach is used to assess surrogacy in the meta-analytic framework (for details, see Buyse & Molenberghs, 2000), computational issues often occur. Such problems mainly occur when the number of trials is low, the number of patients in the different trials is low, and/or when the trial-level heterogeneity is small (Burzykowski et al., 2000).

In that situation, the use of a simplified model-fitting strategy may be warranted (for details, see Burzykowski et al., 2000; Tibaldi et al., 2003).

For example, a reduced bivariate-mixed effect model can be fitted instead of a full model (by using the `Model=c("Reduced")` argument in the function call). In the reduced model, the random-effects structure is simplified (i) by assuming that there is no heterogeneity in the random intercepts, or (ii) by assuming that the covariance between the random intercepts and random treatment effects is zero. Note that under this assumption, the computation of the trial-level coefficient of determination (i.e., R^2_{trial}) simplifies to:

$$R^2_{trial} = \frac{d^2_{ab}}{d_{aa}d_{bb}}.$$

Alternatively, the bivariate mixed-effects model may be abandonned and the user may fit a univariate fixed-effects model, a bivariate fixed-effects model, or a univariate mixed-effects model (for details, see Tibaldi et al., 2003). These models are implemented in the functions `UnifixedContCont`, `BifixedContCont`, and `UnimixedContCont`).

Value

An object of class `BimixedContCont` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Trial.Spec.Results`

A `data.frame` that contains the trial-specific intercepts and treatment effects on the surrogate and the true endpoints when a full model is requested (i.e., $\mu_S + m_{Si}$, $\mu_T + m_{Ti}$, $\alpha + a_i$, and $\beta + b_i$), or the trial-specific treatment effects on the surrogate and the true endpoints when a reduced model is requested (i.e., $\alpha + a_i$, and $\beta + b_i$). Note that the results that are contained in `Trial.Spec.Results` are equivalent to the results in `Results.Stage.1` that are obtained when the functions `UnifixedContCont`, `UnimixedContCont`, or `BifixedContCont` are used.

`Residuals` A `data.frame` that contains the residuals for the surrogate and true endpoints (ε_{Sij} and ε_{Tij}).

`Fixed.Effect.Pars`

A `data.frame` that contains the fixed intercept and treatment effects for the surrogate and the true endpoints (i.e., μ_S, μ_T, α, and β).

`Random.Effect.Pars`

A `data.frame` that contains the random intercept and treatment effects for the surrogate and the true endpoints (i.e., m_{Si}, m_{Ti}, a_i, and b_i) when a full model is fitted (i.e., when `Model=c("Full")` is used in the function call), or that contains the random treatment effects for the surrogate and the true endpoints (i.e., a_i and b_i) when a reduced model is fitted (i.e., when `Model=c("Reduced")` is used in the function call).

`Trial.R2` A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval.

`Indiv.R2` A `data.frame` that contains the individual-level coefficient of determination (R^2_{indiv}), its standard error and confidence interval.

`Trial.R` A `data.frame` that contains the trial-level correlation coefficient (R_{trial}), its standard error and confidence interval.

`Indiv.R` A `data.frame` that contains the individual-level correlation coefficient (R_{indiv}), its standard error and confidence interval.

Cor.Endpoints
: A data.frame that contains the correlations between the surrogate and the true endpoint in the control treatment group (i.e., ρ_{T0S0}) and in the experimental treatment group (i.e., ρ_{T1S1}), their standard errors and their confidence intervals.

D
: The variance-covariance matrix of the random effects (the $\boldsymbol{D}$ matrix), i.e., a 4 by 4 variance-covariance matrix of the random intercept and treatment effects when a full model is fitted (i.e., when `Model=c("Full")` is used in the function call), or a 2 by 2 variance-covariance matrix of the random treatment effects when a reduced model is fitted (i.e., when `Model=c("Reduced")` is used in the function call).

Sigma
: The 2 by 2 variance-covariance matrix of the residuals (ε_{Sij} and ε_{Tij}).

ICA
: A fitted object of class `ICA.ContCont`.

T0T0
: The variance of the true endpoint in the control treatment condition.

T1T1
: The variance of the true endpoint in the experimental treatment condition.

S0S0
: The variance of the surrogate endpoint in the control treatment condition.

S1S1
: The variance of the surrogate endpoint in the experimental treatment condition.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

Tibaldi, F., Abrahantes, J. C., Molenberghs, G., Renard, D., Burzykowski, T., Buyse, M., Parmar, M., et al., (2003). Simplified hierarchical linear models for the evaluation of surrogate endpoints. *Journal of Statistical Computation and Simulation, 73,* 643-658.

See Also

UnifixedContCont, BifixedContCont, UnimixedContCont, plot Meta-Analytic

Examples

```
# Open the Schizo dataset (clinial trial in schizophrenic patients)
data(Schizo)

## Not run:  #Time consuming (>5 sec) code part
# When a reduced bivariate mixed-effect model is used to assess surrogacy,
# the conditioning number for the D matrix is very high:
Sur <- BimixedContCont(Dataset=Schizo, Surr=BPRS, True=PANSS, Treat=Treat, Model="Reduced",
Trial.ID=InvestId, Pat.ID=Id)

# Such problems often occur when the total number of patients, the total number
# of trials and/or the trial-level heterogeneity
# of the treatment effects is relatively small
# As an alternative approach to assess surrogacy, consider using the functions
# BifixedContCont, UnifixedContCont or UnimixedContCont in the meta-analytic framework,
# or use the information-theoretic approach

## End(Not run)
```

binary_continuous_loglik *Loglikelihood function for binary-continuous copula model*

Description

Loglikelihood function for binary-continuous copula model

Usage

```
binary_continuous_loglik(para, X, Y, copula_family, marginal_surrogate)
```

Arguments

`para` Parameter vector. The parameters are ordered as follows:

- `para[1]`: mean parameter for latent true endpoint distribution
- `para[2:p]`: Parameters for surrogate distribution, more details in `?Surrogate::cdf_fun` for the specific implementations.
- `para[p + 1]`: copula parameter

`X` First variable (continuous)

`Y` Second variable (binary, $0$ or $1$)

`copula_family` Copula family, one of the following:

- `"clayton"`
- `"frank"`
- `"gumbel"`
- `"gaussian"`

`marginal_surrogate`
Marginal distribution for the surrogate. For all available options, see `?Surrogate::cdf_fun`.

Value

(numeric) loglikelihood value evaluated in `para`.

Bootstrap.MEP.BinBin *Bootstrap 95% CI around the maximum-entropy ICA and SPF (surrogate predictive function)*

Description

Computes a 95% bootstrap-based CI around the maximum-entropy ICA and SPF (surrogate predictive function) in the binary-binary setting

Usage

```
Bootstrap.MEP.BinBin(Data, Surr, True, Treat, M=100, Seed=123)
```

Arguments

`Data`	The dataset to be used.
`Surr`	The name of the surrogate variable.
`True`	The name of the true endpoint.
`Treat`	The name of the treatment indicator.
`M`	The number of bootstrap samples taken. Default `M=1000`.
`Seed`	The seed to be used. Default `Seed=123`.

Value

`R2H`	The vector the bootstrapped MEP ICA values.
`r_1_1`	The vector of the bootstrapped bootstrapped MEP $r(1,1)$ values.
`r_min1_1`	The vector of the bootstrapped bootstrapped MEP $r(-1,1)$.
`r_0_1`	The vector of the bootstrapped bootstrapped MEP $r(0,1)$.
`r_1_0`	The vector of the bootstrapped bootstrapped MEP $r(1,0)$.
`r_min1_0`	The vector of the bootstrapped bootstrapped MEP $r(-1,0)$.
`r_0_0`	The vector of the bootstrapped bootstrapped MEP $r(0,0)$.
`r_1_min1`	The vector of the bootstrapped bootstrapped MEP $r(1,-1)$.
`r_min1_min1`	The vector of the bootstrapped bootstrapped MEP $r(-1,-1)$.
`r_0_min1`	The vector of the bootstrapped bootstrapped MEP $r(0,-1)$.
`vector_p`	The matrix that contains all bootstrapped maximum entropy distributions of the vector of the potential outcomes.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., & Van der Elst, W. (2015). A maximum-entropy approach for the evluation of surrogate endpoints based on causal inference.

See Also

`ICA.BinBin`, `ICA.BinBin.Grid.Sample`, `ICA.BinBin.Grid.Full`, `plot MaxEntSPF BinBin`

Examples

```
## Not run:  # time consuming code part
MEP_CI <- Bootstrap.MEP.BinBin(Data = Schizo_Bin, Surr = "BPRS_Bin", True = "PANSS_Bin",
                     Treat = "Treat", M = 500, Seed=123)
summary(MEP_CI)

## End(Not run)
```

CausalDiagramBinBin

Draws a causal diagram depicting the median informational coefficients of correlation (or odds ratios) between the counterfactuals for a specified range of values of the ICA in the binary-binary setting.

Description

This function provides a diagram that depicts the medians of the informational coefficients of correlation (or odds ratios) between the counterfactuals for a specified range of values of the individual causal association in the binary-binary setting (R^2_H).

Usage

```
CausalDiagramBinBin(x, Values="Corrs", Theta_T0S0, Theta_T1S1,
Min=0, Max=1, Cex.Letters=3, Cex.Corrs=2, Lines.Rel.Width=TRUE,
Col.Pos.Neg=TRUE, Monotonicity, Histograms.Correlations=FALSE,
Densities.Correlations=FALSE)
```

Arguments

`x`	An object of class `ICA.BinBin`. See `ICA.BinBin`.
`Values`	Specifies whether the median informational coefficients of correlation or median odds ratios between the counterfactuals should be depicted, i.e., `Values="Corrs"` or `Values="ORs"`.
`Theta_T0S0`	The odds ratio between T and S in the control group. This quantity is estimable based on the observed data. Only has to be provided when `Values="ORs"`.
`Theta_T1S1`	The odds ratio between T and S in the experimental treatment group. This quantity is estimable based on the observed data. Only has to be provided when `Values="ORs"`.

`Min` The minimum value of R_H^2 that should be considered. Default=-1.

`Max` The maximum value of R_H^2 that should be considered. Default=1.

`Cex.Letters` The size of the symbols for the counterfactuals (S_0, S_1), T_0, T_1). Default=3.

`Cex.Corrs` The size of the text depicting the median odds ratios between the counterfactuals. Default=2.

`Lines.Rel.Width`
Logical. When `Lines.Rel.Width=TRUE`, the widths of the lines that represent the odds ratios between the counterfactuals are relative to the size of the odds ratios (i.e., a smaller/thicker line is used for smaller/higher odds ratios. When `Lines.Rel.Width=FALSE`, the width of all lines representing the odds ratios between the counterfactuals is identical. Default=`TRUE`. Only considered when `Values="ORs"`.

`Col.Pos.Neg` Logical. When `Col.Pos.Neg=TRUE`, the color of the lines that represent the odds ratios between the counterfactuals is red for odds ratios below 1 and black for the ones above 1. When `Col.Pos.Neg=FALSE`, all lines are in black. Default=`TRUE`. Only considered when `Values="ORs"`.

`Monotonicity` Specifies the monotonicity scenario that should be considered (i.e., `Monotonicity=c("No")`, `Monotonicity=c("True.Endp")`, `Monotonicity=c("Surr.Endp")`, or `Monotonicity=c("Surr.True.`

`Histograms.Correlations`
Should histograms of the informational coefficients of association R_H^2 be provided? Default `Histograms.Correlations=FALSE`.

`Densities.Correlations`
Should densities of the informational coefficients of association R_H^2 be provided? Default `Densities.Correlations=FALSE`.

Value

The following components are stored in the fitted object if histograms of the informational correlations are requested in the function call (i.e., if `Histograms.Correlations=TRUE` and `Values="Corrs"` in the function call):

`R2_H_T0T1` The informational coefficients of association R_H^2 between T_0 and T_1.

`R2_H_S1T0` The informational coefficients of association R_H^2 between S_1 and T_0.

`R2_H_S0T1` The informational coefficients of association R_H^2 between S_0 and T_1.

`R2_H_S0S1` The informational coefficients of association R_H^2 between S_0 and S_1.

`R2_H_S0T0` The informational coefficients of association R_H^2 between S_0 and T_0.

`R2_H_S1T1` The informational coefficients of association R_H^2 between S_1 and T_1.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal inference and meta-analytic paradigms for the validation of surrogate markers.

Van der Elst, W., Alonso, A., & Molenberghs, G. (submitted). An exploration of the relationship between causal inference and meta-analytic measures of surrogacy.

See Also

ICA.BinBin

Examples

```
# Compute R2_H given the marginals specified as the pi's
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.2619048, pi1_0_=0.2857143,
pi_1_1=0.6372549, pi_1_0=0.07843137, pi0_1_=0.1349206, pi_0_1=0.127451,
Seed=1, Monotonicity=c("General"), M=1000)

# Obtain a causal diagram that provides the medians of the
# correlations between the counterfactuals for the range
# of R2_H values between 0.1 and 1
   # Assume no monotonicty
CausalDiagramBinBin(x=ICA, Min=0.1, Max=1, Monotonicity="No")

   # Assume monotonicty for S
CausalDiagramBinBin(x=ICA, Min=0.1, Max=1, Monotonicity="Surr.Endp")

# Now only consider the results that were obtained when
# monotonicity was assumed for the true endpoint
CausalDiagramBinBin(x=ICA, Values="ORs", Theta_T0S0=2.156, Theta_T1S1=10,
Min=0, Max=1,  Monotonicity="True.Endp")
```

CausalDiagramContCont *Draws a causal diagram depicting the median correlations between the counterfactuals for a specified range of values of ICA or MICA in the continuous-continuous setting*

Description

This function provides a diagram that depicts the medians of the correlations between the counterfactuals for a specified range of values of the individual causal association (ICA; ρ_{Δ}) or the meta-analytic individual causal association (MICA; ρ_M).

Usage

```
CausalDiagramContCont(x, Min=-1, Max=1, Cex.Letters=3, Cex.Corrs=2,
Lines.Rel.Width=TRUE, Col.Pos.Neg=TRUE, Histograms.Counterfactuals=FALSE)
```

Arguments

x
: An object of class ICA.ContCont or MICA.ContCont. See ICA.ContCont or MICA.ContCont.

Min
: The minimum values of (M)ICA that should be considered. Default=-1.

Max
: The maximum values of (M)ICA that should be considered. Default=1.

Cex.Letters
: The size of the symbols for the counterfactuals (S_0, S_1), T_0, T_1). Default=3.

Cex.Corrs
: The size of the text depicting the median correlations between the counterfactuals. Default=2.

Lines.Rel.Width
: Logical. When Lines.Rel.Width=TRUE, the widths of the lines that represent the correlations between the counterfactuals are relative to the size of the correlations (i.e., a smaller line is used for correlations closer to zero whereas a thicker line is used for (absolute) correlations closer to 1). When Lines.Rel.Width=FALSE, the width of all lines representing the correlations between the counterfactuals is identical. Default=TRUE.

Col.Pos.Neg
: Logical. When Col.Pos.Neg=TRUE, the color of the lines that represent the correlations between the counterfactuals is red for negative correlations and black for positive ones. When Col.Pos.Neg=FALSE, all lines are in black. Default=TRUE.

Histograms.Counterfactuals
: Should plots that shows the densities for the inidentifiable correlations be shown? Default =FALSE.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal inference and meta-analytic paradigms for the validation of surrogate markers.

Van der Elst, W., Alonso, A., & Molenberghs, G. (submitted). An exploration of the relationship between causal inference and meta-analytic measures of surrogacy.

See Also

ICA.ContCont, MICA.ContCont

Examples

```
## Not run:  #Time consuming (>5 sec) code parts
# Generate the vector of ICA values when rho_T0S0=.91, rho_T1S1=.91, and when the
# grid of values {0, .1, ..., 1} is considered for the correlations
# between the counterfactuals:
SurICA <- ICA.ContCont(T0S0=.95, T1S1=.91, T0T1=seq(0, 1, by=.1), T0S1=seq(0, 1, by=.1),
T1S0=seq(0, 1, by=.1), S0S1=seq(0, 1, by=.1))
```

```
#obtain a plot of ICA

# Obtain a causal diagram that provides the medians of the
# correlations between the counterfactuals for the range
# of ICA values between .9 and 1 (i.e., which assumed
# correlations between the counterfactuals lead to a
# high ICA?)
CausalDiagramContCont(SurICA, Min=.9, Max=1)

# Same, for low values of ICA
CausalDiagramContCont(SurICA, Min=0, Max=.5)
## End(Not run)
```

cdf_fun *Function factory for distribution functions*

Description

Function factory for distribution functions

Usage

```
cdf_fun(para, family)
```

Arguments

`para`	Parameter vector.
`family`	Distributional family, one of the following: - `"normal"`: normal distribution where `para[1]` is the mean and `para[2]` is the standard deviation. - `"logistic"`: logistic distribution as parameterized in `stats::plogis()` where `para[1]` and `para[2]` correspond to `location` and `scale`, respectively. - `"t"`: t distribution as parameterized in `stats::pt()` where `para[1]` and `para[2]` correspond to `ncp` and `df`, respectively.

Value

A distribution function that has a single argument. This is the vector of values in which the distribution function is evaluated.

clayton_loglik_copula_scale

Loglikelihood on the Copula Scale for the Clayton Copula

Description

`clayton_loglik_copula_scale()` computes the loglikelihood on the copula scale for the Clayton copula which is parameterized by `theta` as follows:

$$C(u,v) = (u^{-\theta} + v^{-\theta} - 1)^{-\frac{1}{\theta}}$$

Usage

```
clayton_loglik_copula_scale(theta, u, v, d1, d2, return_sum = TRUE)
```

Arguments

`theta` Copula parameter

`u` A numeric vector. Corresponds to first variable on the copula scale.

`v` A numeric vector. Corresponds to second variable on the copula scale.

`d1` An integer vector. Indicates whether first variable is observed or right-censored,

- `d1[i] = 1` if `u[i]` corresponds to non-censored value
- `d1[i] = 0` if `u[i]` corresponds to right-censored value
- `d1[i] = -1` if `u[i]` corresponds to left-censored value

`d2` An integer vector. Indicates whether first variable is observed or right-censored,

- `d2[i] = 1` if `v[i]` corresponds to non-censored value
- `d2[i] = 0` if `v[i]` corresponds to right-censored value
- `d2[i] = -1` if `v[i]` corresponds to left-censored value

`return_sum` Return the sum of the individual loglikelihoods? If `FALSE`, a vector with the individual loglikelihood contributions is returned.

Value

Value of the copula loglikelihood evaluated in `theta`.

`colorectal` *The Colorectal dataset with a binary surrogate.*

Description

This dataset combines the data that were collected in 26 double-blind randomized clinical trials in advanced colorectal cancer.

Usage

```
data("colorectal")
```

Format

A data frame with 3943 observations on the following 7 variables.

`TRIAL` The ID number of a trial.

`responder` Binary tumor response (the candidate surrogate), coded as 2=complete response (CR) or partial response (PR) and 1=stabled disease (SD) or progressive disease (PD).

`SURVIND` Censoring indicator for survival time.

`TREAT` The treatment indicator, coded as 0=active control and 1=experimental treatment.

`CENTER` The center in which a patient was treated. In this dataset, there was only one center per trial, hence TRIAL=CENTER.

`patientid` The ID number of a patient.

`surv` Survival time (the true endpoint).

References

Alonso, A., Bigirumurame, T., Burzykowski, T., Buyse, M., Molenberghs, G., Muchene, L., ... & Van der Elst, W. (2016). Applied surrogate endpoint evaluation methods with SAS and R. CRC Press.

Examples

```
data(colorectal)
str(colorectal)
head(colorectal)
```

colorectal4 *The Colorectal dataset with an ordinal surrogate.*

Description

This dataset combines the data that were collected in 19 double-blind randomized clinical trials in advanced colorectal cancer.

Usage

```
data("colorectal4")
```

Format

A data frame with 3192 observations on the following 7 variables.

`trialend` The ID number of a trial.

`treatn` The treatment indicator, coded as 0=active control and 1=experimental treatment.

`trueind` Censoring indicator for survival time.

`surrogend` Categorical ordered tumor response (the candidate surrogate), coded as 1=complete response (CR), 2=partial response (PR), 3=stabled disease (SD) and 4=progressive disease (PD).

`patid` The ID number of a patient.

`center` The center in which a patient was treated. In this dataset, there was only one center per trial, hence TRIAL=CENTER.

`truend` Survival time (the true endpoint).

References

Alonso, A., Bigirumurame, T., Burzykowski, T., Buyse, M., Molenberghs, G., Muchene, L., ... & Van der Elst, W. (2016). Applied surrogate endpoint evaluation methods with SAS and R. CRC Press.

Examples

```
data(colorectal4)
str(colorectal4)
head(colorectal4)
```

comb27.BinBin *Assesses the surrogate predictive value of each of the 27 prediction functions in the setting where both S and T are binary endpoints*

Description

The function `comb27.BinBin` assesses a surrogate predictive value of each of the 27 possible prediction functions in the single-trial causal-inference framework when both the surrogate and the true endpoints are binary outcomes. The distribution of frequencies at which each of the 27 possible predicton functions are selected provides additional insights regarding the association between S (Δ_S) and T (Δ_T). See **Details** below.

Usage

```
comb27.BinBin(pi1_1_, pi1_0_, pi_1_1, pi_1_0,
pi0_1_, pi_0_1, Monotonicity=c("No"),M=1000, Seed=1)
```

Arguments

`pi1_1_` A scalar that contains values for $P(T = 1, S = 1|Z = 0)$, i.e., the probability that $S = T = 1$ when under treatment $Z = 0$.

`pi1_0_` A scalar that contains values for $P(T = 1, S = 0|Z = 0)$.

`pi_1_1` A scalar that contains values for $P(T = 1, S = 1|Z = 1)$.

`pi_1_0` A scalar that contains values for $P(T = 1, S = 0|Z = 1)$.

`pi0_1_` A scalar that contains values for $P(T = 0, S = 1|Z = 0)$.

`pi_0_1` A scalar that contains values for $P(T = 0, S = 1|Z = 1)$.

`Monotonicity` Specifies which assumptions regarding monotonicity should be made, only one assumption can be made at the time: `Monotonicity=c("No")`, `Monotonicity=c("True.Endp")`, `Monotonicity=c("Surr.Endp")`, or `Monotonicity=c("Surr.True.Endp")`. Default `Monotonicity=c("No")`.

`M` The number of random samples that have to be drawn for the freely varying parameters. Default `M=100000`.

`Seed` The seed to be used to generate π_r. Default `Seed=1`.

Details

In the continuous normal setting, surroagacy can be assessed by studying the association between the individual causal effects on S and T (see `ICA.ContCont`). In that setting, the Pearson correlation is the obvious measure of association.

When S and T are binary endpoints, multiple alternatives exist. Alonso et al. (2016) proposed the individual causal association (ICA; R_H^2), which captures the association between the individual causal effects of the treatment on S (Δ_S) and T (Δ_T) using information-theoretic principles.

The function `comb27.BinBin` computes R_H^2 using a grid-based approach where all possible combinations of the specified grids for the parameters that are allowed to vary freely are considered. It computes the probability of a prediction error for each of the 27 possible prediction functions.The frequency at which each prediction function is selected provides additional insight about the minimal probability of a prediction error PPE which can be obtained with `PPE.BinBin`.

Value

An object of class `comb27.BinBin` with components,

`index` count variable

`Monotonicity` The vector of Monotonicity assumptions

`Pe` The vector of the prediction error values.

`combo` The vector containing the codes for the each of the 27 prediction functions.

`R2_H` The vector of the R_H^2 values.

`H_Delta_T` The vector of the entropies of Δ_T.

`H_Delta_S` The vector of the entropies of Δ_S.

`I_Delta_T_Delta_S` The vector of the mutual information of Δ_S and Δ_T.

Author(s)

Paul Meyvisch, Wim Van der Elst, Ariel Alonso, Geert Molenberghs

References

Alonso A, Van der Elst W, Molenberghs G, Buyse M and Burzykowski T. (2016). An information-theoretic approach for the evaluation of surrogate endpoints based on causal inference.

Alonso A, Van der Elst W and Meyvisch P (2016). Assessing a surrogate predictive value: A causal inference approach.

See Also

`PPE.BinBin`

Examples

```
# Conduct the analysis assuming no montonicity

## Not run:  # time consuming code part
comb27.BinBin(pi1_1_ = 0.3412, pi1_0_ = 0.2539, pi0_1_ = 0.119,
              pi_1_1 = 0.6863, pi_1_0 = 0.0882, pi_0_1 = 0.0784,
              Seed=1,Monotonicity=c(”No”), M=500000)

## End(Not run)
```

compute_ICA — *Compute Individual Causal Association for a given D-vine copula model in the setting of choice.*

Description

The compute_ICA() function computes the individual causal association for a fully identified D-vine copula model. See details for the default definition of the ICA in each setting.

Usage

```
compute_ICA(endpoint_types, ...)
```

Arguments

endpoint_types (character) vector with two elements indicating the endpoint types: "continuous" or "ordinal".

... Arguments to pass onto compute_ICA_ContCont(), compute_ICA_OrdCont(), or compute_ICA_OrdOrd()

Value

(numeric) A Named vector with the following elements:

- ICA
- Spearman's rho, $\rho_s(\Delta S, \Delta T)$ (if asked)
- Marginal association parameters in terms of Spearman's rho (if asked):

$$\rho_s(T_0, S_0), \rho_s(T_0, S_1), \rho_s(T_0, T_1), \rho_s(S_0, S_1), \rho_s(S_0, T_1), \rho_s(S_1, T_1)$$

compute_ICA_BinCont — *Compute Individual Causal Association for a given D-vine copula model in the Binary-Continuous Setting*

Description

The compute_ICA_BinCont() function computes the individual causal association for a fully identified D-vine copula model in the setting with a continuous surrogate endpoint and a binary true endpoint.

Usage

```
compute_ICA_BinCont(
  copula_par,
  rotation_par,
  copula_family1,
  copula_family2 = copula_family1,
  n_prec,
  q_S0,
  q_S1,
  marginal_sp_rho = TRUE,
  seed = 1
)
```

Arguments

copula_par
: Parameter vector for the sequence of bivariate copulas that define the D-vine copula. The elements of copula_par correspond to $(c_{12}, c_{23}, c_{34}, c_{13;2}, c_{24;3}, c_{14;23})$.

rotation_par
: Vector of rotation parameters for the sequence of bivariate copulas that define the D-vine copula. The elements of rotation_par correspond to $(c_{12}, c_{23}, c_{34}, c_{13;2}, c_{24;3}, c_{14;23})$.

copula_family1
: Copula family of c_{12} and c_{34}. For the possible options, see loglik_copula_scale(). The elements of copula_family correspond to (c_{12}, c_{34}).

copula_family2
: Copula family of the other bivariate copulas. For the possible options, see loglik_copula_scale(). The elements of copula_family2 correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

n_prec
: Number of Monte Carlo samples for the computation of the mutual information.

q_S0
: Quantile function for the distribution of S_0.

q_S1
: Quantile function for the distribution of S_1.

marginal_sp_rho
: (boolean) Compute the sample Spearman correlation matrix? Defaults to TRUE.

seed
: Seed for Monte Carlo sampling. This seed does not affect the global environment.

Value

(numeric) A Named vector with the following elements:

- ICA
- Spearman's rho, $\rho_s(\Delta S, \Delta T)$ (if asked)
- Kendall's tau, $\tau(\Delta S, \Delta T)$ (if asked)
- Marginal association parameters in terms of Spearman's rho:

$$(\rho_s(S_0, S_1), \rho_s(S_0, T_0), \rho_s(S_0, T_1), \rho_s(S_1, T_0), \rho_s(S_0, S_1), \rho_s(T_0, T_1)$$

compute_ICA_ContCont *Compute Individual Causal Association for a given D-vine copula model in the Continuous-Continuous Setting*

Description

The `compute_ICA_ContCont()` function computes the individual causal association (and associated quantities) for a fully identified D-vine copula model in the continuous-continuous setting.

Usage

```
compute_ICA_ContCont(
  copula_par,
  rotation_par,
  copula_family1,
  copula_family2,
  n_prec,
  q_S0,
  q_T0,
  q_S1,
  q_T1,
  marginal_sp_rho = TRUE,
  seed = 1,
  ICA_estimator = NULL,
  plot_deltas = FALSE
)
```

Arguments

`copula_par` Parameter vector for the sequence of bivariate copulas that define the D-vine copula. The elements of `copula_par` correspond to $(c_{12}, c_{23}, c_{34}, c_{13;2}, c_{24;3}, c_{14;23})$.

`rotation_par` Vector of rotation parameters for the sequence of bivariate copulas that define the D-vine copula. The elements of `rotation_par` correspond to $(c_{12}, c_{23}, c_{34}, c_{13;2}, c_{24;3}, c_{14;23})$.

`copula_family1` Copula family of c_{12} and c_{34}. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family` correspond to (c_{12}, c_{34}).

`copula_family2` Copula family of the other bivariate copulas. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family2` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`n_prec` Number of Monte Carlo samples for the computation of the mutual information.

`q_S0` Quantile function for the distribution of S_0.

`q_T0` Quantile function for the distribution of T_0.

`q_S1` Quantile function for the distribution of S_1.

`q_T1` Quantile function for the distribution of T_1.

`marginal_sp_rho`

(boolean) Compute the sample Spearman correlation matrix? Defaults to `TRUE`.

seed	Seed for Monte Carlo sampling. This seed does not affect the global environment.

ICA_estimator	Function that estimates the ICA between the first two arguments which are numeric vectors. Defaults to NULL which corresponds to estimating the mutual information with FNN::mutinfo() and transforming the estimate to the squared informational coefficient of correlation.

plot_deltas	(logical) Plot the sampled individual treatment effects?

Value

(numeric) A Named vector with the following elements:

- ICA
- Spearman's rho, $\rho_s(\Delta S, \Delta T)$ (if asked)
- Marginal association parameters in terms of Spearman's rho (if asked):

$$\rho_s(T_0, S_0), \rho_s(T_0, S_1), \rho_s(T_0, T_1), \rho_s(S_0, S_1), \rho_s(S_0, T_1), \rho_s(S_1, T_1)$$

---

compute_ICA_OrdCont *Compute Individual Causal Association for a given D-vine copula model in the Ordinal-Continuous Setting*

Description

The compute_ICA_OrdCont() function computes the individual causal association for a fully identified D-vine copula model in the setting with a continuous surrogate endpoint and an ordinal true endpoint.

Usage

```
compute_ICA_OrdCont(
  copula_par,
  rotation_par,
  copula_family1,
  copula_family2 = copula_family1,
  n_prec,
  q_S0,
  q_T0,
  q_S1,
  q_T1,
  marginal_sp_rho = TRUE,
  seed = 1,
  ICA_estimator = NULL
)
```

Arguments

copula_par	Parameter vector for the sequence of bivariate copulas that define the D-vine copula. The elements of copula_par correspond to $(c_{12}, c_{23}, c_{34}, c_{13;2}, c_{24;3}, c_{14;23})$.
rotation_par	Vector of rotation parameters for the sequence of bivariate copulas that define the D-vine copula. The elements of rotation_par correspond to $(c_{12}, c_{23}, c_{34}, c_{13;2}, c_{24;3}, c_{14;23})$.
copula_family1	Copula family of c_{12} and c_{34}. For the possible options, see loglik_copula_scale(). The elements of copula_family correspond to (c_{12}, c_{34}).
copula_family2	Copula family of the other bivariate copulas. For the possible options, see loglik_copula_scale(). The elements of copula_family2 correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.
n_prec	Number of Monte Carlo samples for the computation of the mutual information.
q_S0	Quantile function for the distribution of S_0.
q_T0	Quantile function for the distribution of T_0.
q_S1	Quantile function for the distribution of S_1.
q_T1	Quantile function for the distribution of T_1.
marginal_sp_rho	(boolean) Compute the sample Spearman correlation matrix? Defaults to TRUE.
seed	Seed for Monte Carlo sampling. This seed does not affect the global environment.
ICA_estimator	Function that estimates the ICA between the first two arguments which are numeric vectors. Defaults to NULL which corresponds to using estimate_ICA_OrdCont().

Value

(numeric) A Named vector with the following elements:

- ICA
- Spearman's rho, $\rho_s(\Delta S, \Delta T)$ (if asked)
- Marginal association parameters in terms of Spearman's rho (if asked):

$$\rho_s(T_0, S_0), \rho_s(T_0, S_1), \rho_s(T_0, T_1), \rho_s(S_0, S_1), \rho_s(S_0, T_1), \rho_s(S_1, T_1)$$

---

compute_ICA_OrdOrd *Compute Individual Causal Association for a given D-vine copula model in the Ordinal-Ordinal Setting*

Description

The compute_ICA_OrdOrd() function computes the individual causal association for a fully identified D-vine copula model in the setting with an ordinal surrogate and true endpoint.

Usage

```
compute_ICA_OrdOrd(
  copula_par,
  rotation_par,
  copula_family1,
  copula_family2 = copula_family1,
  n_prec,
  q_S0,
  q_T0,
  q_S1,
  q_T1,
  marginal_sp_rho = TRUE,
  seed = 1,
  ICA_estimator = NULL
)
```

Arguments

`copula_par`	Parameter vector for the sequence of bivariate copulas that define the D-vine copula. The elements of `copula_par` correspond to $(c_{12}, c_{23}, c_{34}, c_{13;2}, c_{24;3}, c_{14;23})$.
`rotation_par`	Vector of rotation parameters for the sequence of bivariate copulas that define the D-vine copula. The elements of `rotation_par` correspond to $(c_{12}, c_{23}, c_{34}, c_{13;2}, c_{24;3}, c_{14;23})$.
`copula_family1`	Copula family of c_{12} and c_{34}. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family` correspond to (c_{12}, c_{34}).
`copula_family2`	Copula family of the other bivariate copulas. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family2` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.
`n_prec`	Number of Monte Carlo samples for the computation of the mutual information.
`q_S0`	Quantile function for the distribution of S_0.
`q_T0`	Quantile function for the distribution of T_0.
`q_S1`	Quantile function for the distribution of S_1.
`q_T1`	Quantile function for the distribution of T_1.
`marginal_sp_rho`	(boolean) Compute the sample Spearman correlation matrix? Defaults to `TRUE`.
`seed`	Seed for Monte Carlo sampling. This seed does not affect the global environment.
`ICA_estimator`	Function that estimates the ICA between the first two arguments which are numeric vectors. Defaults to `NULL` which corresponds to using `estimate_ICA_OrdOrd()`.

Value

(numeric) A Named vector with the following elements:

- ICA
- Spearman's rho, $\rho_s(\Delta S, \Delta T)$ (if asked)

- Marginal association parameters in terms of Spearman's rho (if asked):

$$\rho_s(T_0, S_0), \rho_s(T_0, S_1), \rho_s(T_0, T_1), \rho_s(S_0, S_1), \rho_s(S_0, T_1), \rho_s(S_1, T_1)$$

compute_ICA_SurvSurv *Compute Individual Causal Association for a given D-vine copula model in the Survival-Survival Setting*

Description

The compute_ICA_SurvSurv() function computes the individual causal association (and associated quantities) for a fully identified D-vine copula model in the survival-survival setting.

Usage

```
compute_ICA_SurvSurv(
  copula_par,
  rotation_par,
  copula_family1,
  copula_family2,
  n_prec,
  q_S0,
  q_T0,
  q_S1,
  q_T1,
  composite,
  marginal_sp_rho = TRUE,
  seed = 1,
  mutinfo_estimator = NULL,
  plot_deltas = FALSE,
  restr_time = +Inf
)
```

Arguments

copula_par Parameter vector for the sequence of bivariate copulas that define the D-vine copula. The elements of copula_par correspond to $(c_{12}, c_{23}, c_{34}, c_{13;2}, c_{24;3}, c_{14;23})$.

rotation_par Vector of rotation parameters for the sequence of bivariate copulas that define the D-vine copula. The elements of rotation_par correspond to $(c_{12}, c_{23}, c_{34}, c_{13;2}, c_{24;3}, c_{14;23})$.

copula_family1 Copula family of c_{12} and c_{34}. For the possible options, see loglik_copula_scale(). The elements of copula_family correspond to (c_{12}, c_{34}).

copula_family2 Copula family of the other bivariate copulas. For the possible options, see loglik_copula_scale(). The elements of copula_family2 correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

n_prec Number of Monte Carlo samples for the computation of the mutual information.

q_S0 Quantile function for the distribution of S_0.

q_T0	Quantile function for the distribution of T_0.
q_S1	Quantile function for the distribution of S_1.
q_T1	Quantile function for the distribution of T_1.
composite	(boolean) If composite is TRUE, then the surrogate endpoint is a composite of both a "pure" surrogate endpoint and the true endpoint, e.g., progression-free survival is the minimum of time-to-progression and time-to-death.
marginal_sp_rho	(boolean) Compute the sample Spearman correlation matrix? Defaults to TRUE.
seed	Seed for Monte Carlo sampling. This seed does not affect the global environment.
mutinfo_estimator	Function that estimates the mutual information between the first two arguments which are numeric vectors. Defaults to FNN::mutinfo() with default arguments.
plot_deltas	(logical) Plot the sampled individual treatment effects?
restr_time	Restriction time for the potential outcomes. Defaults to +Inf which means no restriction. Otherwise, the sampled potential outcomes are replace by pmin(S0, restr_time) (and similarly for the other potential outcomes).

Value

(numeric) A Named vector with the following elements:

- ICA
- Spearman's rho, $\rho_s(\Delta S, \Delta T)$ (if asked)
- Marginal association parameters in terms of Spearman's rho (if asked):

$$\rho_s(T_0, S_0), \rho_s(T_0, S_1), \rho_s(T_0, T_1), \rho_s(S_0, S_1), \rho_s(S_0, T_1), \rho_s(S_1, T_1)$$

- Survival classification proportions (if asked):

$$\pi_{harmed}, \pi_{protected}, \pi_{always}, \pi_{never}$$

constructor_ICA_estimator

Function constructor to estimate the ICA given a set of sampled patient-level treatment effects

Description

The constructor_ICA_estimator() function returns a function the estimates the ICA as a user-specified function of $I(\Delta S; \Delta T)$, ΔS, and ΔT.

Usage

```
constructor_ICA_estimator(endpoint_types, ICA_def)
```

Arguments

`endpoint_types` (character) vector with two elements indicating the endpoint types: `"continuous"` or `"ordinal"`.

`ICA_def` function that takes the following arguments: $I(\Delta S; \Delta T)$, ΔS, and ΔT. It returns the ICA as a function of these information-theoretic quantities.

Value

A function that estimates the user-defined definition of the ICA. This function can be used as `ICA_estimator` in `sensitivity_analysis_copula()`.

`continuous_continuous_loglik` *Loglikelihood function for continuous-continuous copula model*

Description

`continuous_continuous_loglik()` computes the observed-data loglikelihood for a bivariate copula model with two continuous endpoints.

Usage

```
continuous_continuous_loglik(
  para,
  X,
  Y,
  copula_family,
  marginal_X,
  marginal_Y,
  return_sum = TRUE
)
```

Arguments

`para` Parameter vector. The parameters are ordered as follows:

- `para[1:p1]`: Parameters for the distribution of `X` as specified in `marginal_X`.
- `para[(p1 + 1):(p1 + p2)]`: Parameters for the distribution of `Y` as specified in `marginal_Y`.
- `para[p1 + p2 + 1]`: copula parameter

`X` First variable (Continuous)

`Y` Second variable (Continuous)

`copula_family` Copula family, one of the following:

- `"clayton"`
- `"frank"`

- "gumbel"
- "gaussian"

marginal_X, marginal_Y
List with the following three elements (in order):

- Density function with first argument x and second argument para the parameter vector for this distribution.
- Distribution function with first argument x and second argument para the parameter vector for this distribution.
- Inverse distribution function with first argument p and second argument para the parameter vector for this distribution.
- The number of elements in para.
- A vector of starting values for para.

return_sum
Return the sum of the individual loglikelihoods? If FALSE, a vector with the individual loglikelihood contributions is returned.

Value

(numeric) loglikelihood value evaluated in para.

---

delta_method_log_mutinfo

Variance of log-mutual information based on the delta method

Description

delta_method_log_mutinfo() computes the variance of the estimated log mutual information, given the unidentifiable parameters.

Usage

```
delta_method_log_mutinfo(
  fitted_model,
  copula_par_unid,
  copula_family2,
  rotation_par_unid,
  n_prec,
  mutinfo_estimator = NULL,
  composite,
  seed,
  eps = 0.001
)
```

Arguments

`fitted_model` Returned value from `fit_model_SurvSurv()`. This object contains the estimated identifiable part of the joint distribution for the potential outcomes.

`copula_par_unid` Parameter vector for the sequence of *unidentifiable* bivariate copulas that define the D-vine copula. The elements of `copula_par` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`copula_family2` Copula family of the other bivariate copulas. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family2` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`rotation_par_unid` Vector of rotation parameters for the sequence of *unidentifiable* bivariate copulas that define the D-vine copula. The elements of `rotation_par` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`n_prec` Number of Monte Carlo samples for the computation of the mutual information.

`mutinfo_estimator` Function that estimates the mutual information between the first two arguments which are numeric vectors. Defaults to `FNN::mutinfo()` with default arguments.

`composite` (boolean) If `composite` is `TRUE`, then the surrogate endpoint is a composite of both a "pure" surrogate endpoint and the true endpoint, e.g., progression-free survival is the minimum of time-to-progression and time-to-death.

`seed` Seed for Monte Carlo sampling. This seed does not affect the global environment.

`eps` (numeric) Step size for finite difference in numeric differentiation

Details

This function should not be used. The ICA is computed through numerical methods with a considerable error. This error is negligible in individual estimates of the ICA; however, this error easily breaks the numeric differentiation because finite differences are inflated by this error.

Value

(numeric) Variance for the estimated ICA based on the delta method, holding the unidentifiable parameters fixed at the user supplied values.

`Dvine_ICA_confint` *Confidence interval for the ICA given the unidentifiable parameters*

Description

`Dvine_ICA_confint()` computes the confidence interval for the ICA in the D-vine copula model. The unidentifiable parameters are fixed at the user supplied values.

Usage

```
Dvine_ICA_confint(
  fitted_model,
  alpha,
  copula_par_unid,
  copula_family2,
  rotation_par_unid,
  n_prec,
  mutinfo_estimator = NULL,
  composite,
  B,
  seed
)
```

Arguments

`fitted_model` Returned value from `fit_model_SurvSurv()`. This object contains the estimated identifiable part of the joint distribution for the potential outcomes.

`alpha` (numeric) `1 - alpha` is the level of the confidence interval

`copula_par_unid` Parameter vector for the sequence of *unidentifiable* bivariate copulas that define the D-vine copula. The elements of `copula_par` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`copula_family2` Copula family of the other bivariate copulas. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family2` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`rotation_par_unid` Vector of rotation parameters for the sequence of *unidentifiable* bivariate copulas that define the D-vine copula. The elements of `rotation_par` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`n_prec` Number of Monte Carlo samples for the computation of the mutual information.

`mutinfo_estimator` Function that estimates the mutual information between the first two arguments which are numeric vectors. Defaults to `FNN::mutinfo()` with default arguments.

`composite` (boolean) If `composite` is `TRUE`, then the surrogate endpoint is a composite of both a "pure" surrogate endpoint and the true endpoint, e.g., progression-free survival is the minimum of time-to-progression and time-to-death.

`B` Number of bootstrap replications

`seed` Seed for Monte Carlo sampling. This seed does not affect the global environment.

Value

(numeric) Vector with the limits of the two-sided `1 - alpha` confidence interval.

ECT *Apply the Entropy Concentration Theorem*

Description

The Entropy Concentration Theorem (ECT; Edwin, 1982) states that if N is large enough, then $100(1-F)\%$ of all $\boldsymbol{p}*$ and ΔH is determined by the upper tail are $1-F$ of a χ^2 distribution, with $DF = q - m - 1$ (which equals 8 in a surrogate evaluation context).

Usage

```
ECT(Perc=.95, H_Max, N)
```

Arguments

Perc	The desired interval. E.g., `Perc=.05` will generate the lower and upper bounds for $H(\boldsymbol{p})$ that contain 95% of the cases (as determined by the ECT).
H_Max	The maximum entropy value. In the binary-binary setting, this can be computed using the function `MaxEntICABinBin`.
N	The sample size.

Value

An object of class `ECT` with components,

Lower_H	The lower bound of the requested interval.
Upper_H	The upper bound of the requested interval, which equals H_Max.

Author(s)

Wim Van der Elst, Paul Meyvisch, & Ariel Alonso

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2016). Surrogate markers validation: the continuous-binary setting from a causal inference perspective.

See Also

`MaxEntICABinBin`, `ICA.BinBin`

Examples

```
ECT_fit <- ECT(Perc = .05, H_Max = 1.981811, N=454)
summary(ECT_fit)
```

estimate_ICA_BinCont *Estimate ICA in Binary-Continuous Setting*

Description

`estimate_ICA_BinCont()` estimates the individual causal association (ICA) for a sample of individual causal treatment effects with a continuous surrogate and a binary true endpoint. The ICA in this setting is defined as follows,

$$R^2_H = \frac{I(\Delta S; \Delta T)}{H(\Delta T)}$$

where $I(\Delta S; \Delta T)$ is the mutual information and $H(\Delta T)$ the entropy.

Usage

```
estimate_ICA_BinCont(delta_S, delta_T)
```

Arguments

`delta_S`	(numeric) Vector of individual causal treatment effects on the surrogate.
`delta_T`	(integer) Vector of individual causal treatment effects on the true endpoint. Should take on one of the following values: `-1L`, `0L`, or `1L`.

Value

(numeric) Estimated ICA

estimate_ICA_ContCont *Estimate ICA in Ordinal-Ordinal Setting*

Description

`estimate_ICA_ContCont()` estimates the individual causal association (ICA) for a sample of individual causal treatment effects with a continuous surrogate and true endpoint. The ICA in this setting is defined as the squared informational coefficient of correlation, which is a transformation of the mutual information. The mutual information is estimated with `fnn::mutinfo()`.

Usage

```
estimate_ICA_ContCont(delta_S, delta_T)
```

Arguments

`delta_S`	(numeric) Vector of individual causal treatment effects on the surrogate.
`delta_T`	(numeric) Vector of individual causal treatment effects on the true endpoint.

Value

(numeric) Estimated ICA

`estimate_ICA_OrdCont` *Estimate ICA in Ordinal-Continuous Setting*

Description

`estimate_ICA_OrdCont()` estimates the individual causal association (ICA) for a sample of individual causal treatment effects with a continuous surrogate and an ordinal true endpoint. The ICA in this setting is defined as follows,

$$R^2_H = \frac{I(\Delta S; \Delta T)}{H(\Delta T)}$$

where $I(\Delta S; \Delta T)$ is the mutual information and $H(\Delta T)$ the entropy.

Usage

```
estimate_ICA_OrdCont(delta_S, delta_T)
```

Arguments

`delta_S`	(numeric) Vector of individual causal treatment effects on the surrogate.
`delta_T`	(integer) Vector of individual causal treatment effects on the true endpoint.

Value

(numeric) Estimated ICA

Individual Causal Association

Many association measures can operationalize the ICA. For each setting, we consider one default definition for the ICA which follows from the mutual information.

Continuous-Continuous:
The ICA is defined as the squared informational coefficient of correlation (SICC or R^2_H), which is a transformation of the mutual information to the unit interval:

$$R^2_h = 1 - e^{-2 \cdot I(\Delta S; \Delta T)}$$

where 0 indicates independence, and 1 a functional relationship between ΔS and ΔT. If $(\Delta S, \Delta T)'$ is bivariate normal, the ICA equals the Pearson correlation between ΔS and ΔT.

Ordinal-Continuous:
The ICA is defined as the following transformation of the mutual information:

$$R^2_H = \frac{I(\Delta S; \Delta T)}{H(\Delta T)},$$

where $I(\Delta S; \Delta T)$ is the mutual information and $H(\Delta T)$ the entropy.

Ordinal-Ordinal:

The ICA is defined as the following transformation of the mutual information:

$$R^2_H = \frac{I(\Delta S; \Delta T)}{\min\{H(\Delta S), H(\Delta T)\}},$$

where $I(\Delta S; \Delta T)$ is the mutual information, and $H(\Delta S)$ and $H(\Delta T)$ the entropy of ΔS and ΔT, respectively.

`estimate_ICA_OrdOrd` *Estimate ICA in Ordinal-Ordinal Setting*

Description

`estimate_ICA_OrdOrd()` estimates the individual causal association (ICA) for a sample of individual causal treatment effects with an ordinal surrogate and true endpoint. The ICA in this setting is defined as follows:

$$R^2_H = \frac{I(\Delta S; \Delta T)}{\min\{H(\Delta S), H(\Delta T)\}}$$

where $I(\Delta S; \Delta T)$ is the mutual information, and $H(\Delta S)$ and $H(\Delta T)$ the entropy of ΔS and ΔT, respectively.

Usage

```
estimate_ICA_OrdOrd(delta_S, delta_T)
```

Arguments

`delta_S`	(integer) Vector of individual causal treatment effects on the surrogate.
`delta_T`	(integer) Vector of individual causal treatment effects on the true endpoint.

Value

(numeric) Estimated ICA

`estimate_marginal` *Estimate marginal distribution using ML*

Description

`estimate_marginal()` estimates the marginal distribution specified by `marginal_Y` using maximum likelihood. The optimizer is Newton-Raphson.

Usage

```
estimate_marginal(Y, marginal_Y, starting_values)
```

Arguments

`Y` Observations (continuous)

`marginal_Y` List with the following five elements (in order):

- Density function with first argument `x` and second argument `para` the parameter vector for this distribution.
- Distribution function with first argument `x` and second argument `para`.
- Inverse distribution function with first argument `p` and second argument `para`.
- The number of elements in `para`.
- Starting values for `para`.

`starting_values` Starting values for `marginal_Y`

Value

Estimated parameters

`estimate_mutual_information_SurvSurv` *Estimate the Mutual Information in the Survival-Survival Setting*

Description

`estimate_mutual_information_SurvSurv()` estimates the mutual information for a sample of individual causal treatment effects with a time-to-event surrogate and a time-to-event true endpoint. The mutual information is estimated by first estimating the bivariate density and then computing the mutual information for the estimated density.

Usage

```
estimate_mutual_information_SurvSurv(delta_S, delta_T, minfo_prec)
```

Arguments

`delta_S` (numeric) Vector of individual causal treatment effects on the surrogate.

`delta_T` (numeric) Vector of individual causal treatment effects on the true endpoint.

`minfo_prec` Number of quasi Monte-Carlo samples for the numerical integration to obtain the mutual information. If this value is 0 (default), the mutual information is not computed and `NA` is returned for the mutual information and derived quantities.

Value

(numeric) estimated mutual information.

`Fano.BinBin` *Evaluate the possibility of finding a good surrogate in the setting where both S and T are binary endpoints*

Description

The function `Fano.BinBin` evaluates the existence of a good surrogate in the single-trial causal-inference framework when both the surrogate and the true endpoints are binary outcomes. See **Details** below.

Usage

```
Fano.BinBin(pi1_,  pi_1, rangepi10=c(0,min(pi1_,1-pi_1)),
fano_delta=c(0.1), M=100, Seed=1)
```

Arguments

`pi1_` A scalar or a vector of plausibel values that represents the proportion of responders under treatment.

`pi_1` A scalar or a vector of plausibel values that represents the proportion of responders under control.

`rangepi10` Represents the range from which π_{10} is sampled. By default, Monte Carlo simulation will be constrained to the interval $[0, \min(\pi_{1\cdot}, \pi_{\cdot 0})]$ but this allows the user to specify a more narrow range. `rangepi10=c(0,0)` is equivalent to the assumption of monotonicity for the true endpoint.

`fano_delta` A scalar or a vector that specifies the values for the upper bound of the prediction error δ. Default `fano_delta=c(0.2)`.

`M` The number of random samples that have to be drawn for the freely varying parameter π_{10}. Default `M=1000`. The number of random samples should be sufficiently large in relation to the length of the interval `rangepi10`. Typically `M=1000` yields a sufficiently fine grid. In case `rangepi10` is a single value: `M=1`

`Seed` The seed to be used to sample the freely varying parameter π_{10}. Default `Seed=1`.

Details

Values for π_{10} have to be uniformly sampled from the interval $[0, \min(\pi_{1\cdot}, \pi_{\cdot 0})]$. Any sampled value for π_{10} will fully determine the bivariate distribution of potential outcomes for the true endpoint. The treatment effect should be positive.

The vector $\boldsymbol{\pi_{km}}$ fully determines R^2_{HL}.

Value

An object of class `Fano.BinBin` with components,

`R2_HL`	The sampled values for R^2_{HL}.
`H_Delta_T`	The sampled values for $H\Delta T$.
`PPE_T`	The sampled values for PPE_T.
`minpi10`	The minimum value for π_{10}.
`maxpi10`	The maximum value for π_{10}.
`samplepi10`	The sampled value for π_{10}.
`delta`	The specified vector of upper bounds for the prediction errors.
`uncertainty`	Indexes the sampling of $pi1\_$.
`pi_00`	The sampled values for π_{00}.
`pi_11`	The sampled values for π_{11}.
`pi_01`	The sampled values for π_{01}.
`pi_10`	The sampled values for π_{10}.

Author(s)

Paul Meyvisch, Wim Van der Elst, Ariel Alonso

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2014). Validation of surrogate endpoints: the binary-binary setting from a causal inference perspective.

See Also

`plot.Fano.BinBin`

Examples

```
# Conduct the analysis assuming no montonicity
# for the true endpoint, using a range of
# upper bounds for prediction errors
Fano.BinBin(pi1_ = 0.5951 ,  pi_1 = 0.7745,
fano_delta=c(0.05, 0.1, 0.2), M=1000)


# Conduct the same analysis now sampling from
# a range of values to allow for uncertainty

Fano.BinBin(pi1_ = runif(n=20,min=0.504,max=0.681),
pi_1 = runif(n=20,min=0.679,max=0.849),
fano_delta=c(0.05, 0.1, 0.2), M=10, Seed=2)
```

FederatedApproachStage1 *Fits the first stage model in the two-stage federated data analysis approach.*

Description

The function 'FederatedApproachStage1()' fits the first stage model of the two-stage federated data analysis approach to assess surrogacy.

Usage

```
FederatedApproachStage1(
  Dataset,
  Surr,
  True,
  Treat,
  Trial.ID,
  Min.Treat.Size = 2,
  Alpha = 0.05
)
```

Arguments

`Dataset` A data frame with the correct columns (See Data Format).

`Surr` Surrogate endpoint.

`True` True endpoint.

`Treat` Treatment indicator.

`Trial.ID` Trial indicator.

`Min.Treat.Size` The minimum number of patients in each group (control or experimental) that a trial should contain to be included in the analysis. If the number of patients in a group of a trial is smaller than the value specified by Min.Treat.Size, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R^2_{trial} and R^2_{indiv}. Default 0.05.

Value

Returns an object of class "FederatedApproachStage1()" that can be used to evaluate surrogacy in the second stage model and contains the following elements:

- Results.Stage.1: a data frame that contains the estimated fixed effects and the elements of Σ_i.
- R.i: the variance-covariance matrix of the estimated fixed effects.

Model

The two-stage federated data analysis approach that can be used to assess surrogacy in the meta-analytic multiple-trial setting (Continuous-continuous case), but without the need of sharing data. Instead, each organization conducts separate analyses on their data set using a so-called "first stage" model. The results of these analyses are then aggregated at a central analysis hub, where the aggregated results are analyzed using a "second stage" model and the necessary metrics (R^2_{trial} and R^2_{indiv}) for the validation of the surrogate endpoint are obtained. This function fits the first stage model, where a linear model is fitted, allowing estimation of the fixed effects.

Data Format

The data frame must contain the following columns:

- a column with the true endpoint
- a column with the surrogate endpoint
- a column with the treatment indicator: 0 or 1
- a column with the trial indicator
- a column with the patient indicator

Author(s)

Dries De Witte

References

Florez, A. J., Molenberghs G, Verbeke G, Alonso, A. (2019). A closed-form estimator for meta-analysis and surrogate markers evaluation. Journal of Biopharmaceutical Statistics, 29(2) 318-332.

Examples

```
## Not run:
#As an example, the federated data analysis approach can be applied to the Schizo data set
data(Schizo)
Schizo <-  Schizo[order(Schizo$InvestId, Schizo$Id),]
#Create separate datasets for each investigator
Schizo_datasets <- list()

for (invest_id in 1:198) {
Schizo_datasets[[invest_id]] <- Schizo[Schizo$InvestId == invest_id, ]
assign(paste0("Schizo", invest_id), Schizo_datasets[[invest_id]])
}
```

```
#Fit the first stage model for each dataset separately
results_stage1 <- list()
invest_ids <- list()
i <- 1
for (invest_id in 1:198) {
  dataset <- Schizo_datasets[[invest_id]]

  skip_to_next <- FALSE
 tryCatch(FederatedApproachStage1(dataset, Surr=CGI, True=PANSS, Treat=Treat, Trial.ID = InvestId,
                                   Min.Treat.Size = 5, Alpha = 0.05),
                                   error = function(e) { skip_to_next <<- TRUE})
  #if the trial does not have the minimum required number, skip to the next
  if(skip_to_next) { next }

 results_stage1[[invest_id]] <- FederatedApproachStage1(dataset, Surr=CGI, True=PANSS, Treat=Treat,
                                                 Trial.ID = InvestId, Min.Treat.Size = 5,
                                                         Alpha = 0.05)
  assign(paste0("stage1_invest", invest_id), results_stage1[[invest_id]])
 invest_ids[[i]] <- invest_id #keep a list of ids with datasets with required number of patients
  i <- i+1
}

invest_ids <- unlist(invest_ids)
invest_ids

#Combine the results of the first stage models
for (invest_id in invest_ids) {
  dataset <- results_stage1[[invest_id]]$Results.Stage.1
  if (invest_id == invest_ids[1]) {
    all_results_stage1<- dataset
 } else {
    all_results_stage1 <- rbind(all_results_stage1,dataset)
  }
}

all_results_stage1 #that combines the results of the first stage models

R.list <- list()
i <- 1
for (invest_id in invest_ids) {
  R <- results_stage1[[invest_id]]$R.i
  R.list[[i]] <- as.matrix(R[1:4,1:4])
  i <- i+1
}

R.list #list that combines all the variance-covariance matrices of the fixed effects

fit <- FederatedApproachStage2(Dataset = all_results_stage1, Intercept.S = Intercept.S,
                               alpha = alpha, Intercept.T = Intercept.T, beta = beta,
                               sigma.SS = sigma.SS, sigma.ST = sigma.ST,
                               sigma.TT = sigma.TT, Obs.per.trial = n,
                               Trial.ID = Trial.ID, R.list = R.list)
summary(fit)
```

```
## End(Not run)
```

FederatedApproachStage2

Fits the second stage model in the two-stage federated data analysis approach.

Description

The function 'FederatedApproachStage2()' fits the second stage model of the two-stage federated data analysis approach to assess surrogacy.

Usage

```
FederatedApproachStage2(
  Dataset,
  Intercept.S,
  alpha,
  Intercept.T,
  beta,
  sigma.SS,
  sigma.ST,
  sigma.TT,
  Obs.per.trial,
  Trial.ID,
  R.list,
  Alpha = 0.05
)
```

Arguments

`Dataset`	A data frame with the correct columns (See Data Format).
`Intercept.S`	Estimated intercepts for the surrogate endpoint.
`alpha`	Estimated treatment effects for the surrogate endpoint.
`Intercept.T`	Estimated intercepts for the true endpoint.
`beta`	Estimated treatment effects for the true endpoint.
`sigma.SS`	Estimated variance of the error terms for the surrogate endpoint.
`sigma.ST`	Estimated covariance between the error terms of the surrogate and true endpoint.
`sigma.TT`	Estimated variance of the error terms for the true endpoint.
`Obs.per.trial`	Number of subjects in the trial.
`Trial.ID`	Trial indicator.
`R.list`	List of the variance-covariance matrices of the fixed effects.
`Alpha`	The α-level that is used to determine the confidence intervals around R^2_{trial} and R^2_{indiv}. Default 0.05.

Value

Returns an object of class "FederatedApproachStage2()" that can be used to evaluate surrogacy.

- Indiv.R2: a data frame that contains the R^2_{indiv} and 95% confidence interval to evaluate surrogacy at the trial level.
- Trial.R2: a data frame that contains the R^2_{trial} and 95% confidence interval to evaluate surrogacy at the trial level.
- Fixed.Effects: a data frame that contains the average of the estimated fixed effects.
- D: estimated D matrix.
- Obs.Per.Trial: number of observations in each trial.

Model

The two-stage federated data analysis approach that can be used to assess surrogacy in the meta-analytic multiple-trial setting (Continuous-continuous case), but without the need of sharing data. Instead, each organization conducts separate analyses on their data set using the so-called "first stage" model. The results of these analyses are then aggregated at a central analysis hub, where the aggregated results are analyzed using a "second stage" model and the necessary metrics (R^2_{trial} and R^2_{indiv}) for the validation of the surrogate endpoint are obtained. This function fits the second stage model, where a method-of-moments estimator is used to obtain the variance-covariance matrix D from which the R^2_{trial} can be derived. The R^2_{indiv} is obtained with a weighted average of the elements in Σ_i.

Data Format

A data frame that combines the results of the first stage models and contains:

- a column with the trial indicator
- a column with the number of subjects in the trial
- a column with the estimated intercepts for the surrogate
- a column with the estimated treatment effects for the surrogate
- a column with the estimated intercepts for the true endpoint
- a column with the estimated treatment effects for the true endpoint
- a column with the variances of the error term for the surrogate endpoint
- a column with the covariances between the error terms of the surrogate and true endpoint
- a column with the variances of the error term for the true endpoint

A list that combines all the variance-covariance matrices of the fixed effects obtained using the first stage model

Author(s)

Dries De Witte

References

Florez, A. J., Molenberghs G, Verbeke G, Alonso, A. (2019). A closed-form estimator for meta-analysis and surrogate markers evaluation. Journal of Biopharmaceutical Statistics, 29(2) 318-332.

Examples

```
## Not run:
#As an example, the federated data analysis approach can be applied to the Schizo data set
data(Schizo)
Schizo <-  Schizo[order(Schizo$InvestId, Schizo$Id),]
#Create separate datasets for each investigator
Schizo_datasets <- list()

for (invest_id in 1:198) {
Schizo_datasets[[invest_id]] <- Schizo[Schizo$InvestId == invest_id, ]
assign(paste0("Schizo", invest_id), Schizo_datasets[[invest_id]])
}
#Fit the first stage model for each dataset separately
results_stage1 <- list()
invest_ids <- list()
i <- 1
for (invest_id in 1:198) {
  dataset <- Schizo_datasets[[invest_id]]

  skip_to_next <- FALSE
 tryCatch(FederatedApproachStage1(dataset, Surr=CGI, True=PANSS, Treat=Treat, Trial.ID = InvestId,
                                  Min.Treat.Size = 5, Alpha = 0.05),
                                  error = function(e) { skip_to_next <<- TRUE})
  #if the trial does not have the minimum required number, skip to the next
  if(skip_to_next) { next }

 results_stage1[[invest_id]] <- FederatedApproachStage1(dataset, Surr=CGI, True=PANSS, Treat=Treat,
                                                Trial.ID = InvestId, Min.Treat.Size = 5,
                                                        Alpha = 0.05)
  assign(paste0("stage1_invest", invest_id), results_stage1[[invest_id]])
 invest_ids[[i]] <- invest_id #keep a list of ids with datasets with required number of patients
  i <- i+1
}

invest_ids <- unlist(invest_ids)
invest_ids

#Combine the results of the first stage models
for (invest_id in invest_ids) {
  dataset <- results_stage1[[invest_id]]$Results.Stage.1
  if (invest_id == invest_ids[1]) {
    all_results_stage1<- dataset
 } else {
    all_results_stage1 <- rbind(all_results_stage1,dataset)
  }
}
```

```
all_results_stage1 #that combines the results of the first stage models

R.list <- list()
i <- 1
for (invest_id in invest_ids) {
  R <- results_stage1[[invest_id]]$R.i
  R.list[[i]] <- as.matrix(R[1:4,1:4])
  i <- i+1
}

R.list #list that combines all the variance-covariance matrices of the fixed effects

fit <- FederatedApproachStage2(Dataset = all_results_stage1, Intercept.S = Intercept.S,
                               alpha = alpha, Intercept.T = Intercept.T, beta = beta,
                               sigma.SS = sigma.SS, sigma.ST = sigma.ST,
                               sigma.TT = sigma.TT, Obs.per.trial = n,
                               Trial.ID = Trial.ID, R.list = R.list)
summary(fit)

## End(Not run)
```

fit_copula_ContCont *Fit continuous-continuous vine copula model*

Description

fit_copula_ContCont() fits the continuous-continuous vine copula model. See Details for more information about this model.

Usage

```
fit_copula_ContCont(
  data,
  copula_family,
  marginal_S0,
  marginal_S1,
  marginal_T0,
  marginal_T1,
  start_copula,
  method = "BFGS",
  ...
)
```

Arguments

data — data frame with three columns in the following order: surrogate endpoint, true endpoint, and treatment indicator (0/1 coding). Ordinal endpoints should be integers starting from 1.

`copula_family` One of the following parametric copula families: `"clayton"`, `"frank"`, `"gaussian"`, or `"gumbel"`. The first element in `copula_family` corresponds to the control group, the second to the experimental group.

`marginal_S0`, `marginal_S1`, `marginal_T0`, `marginal_T1`
List with the following three elements (in order):

- Density function with first argument `x` and second argument `para` the parameter vector for this distribution.
- Distribution function with first argument `x` and second argument `para` the parameter vector for this distribution.
- Inverse distribution function with first argument `p` and second argument `para` the parameter vector for this distribution.
- The number of elements in `para`.
- A vector of starting values for `para`.

`start_copula` Starting value for the copula parameter.

`method` Optimization algorithm for maximizing the objective function. For all options, see ?`maxLik::maxLik`. Defaults to `"BFGS"`.

`...` Extra argument to pass onto maxLik::maxLik

Value

Returns an S3 object that can be used to perform the sensitivity analysis with `sensitivity_analysis_copula()`.

Author(s)

Florian Stijven

See Also

`sensitivity_analysis_copula()`, `print.vine_copula_fit()`, `plot.vine_copula_fit()`

---

fit_copula_model_BinCont

Fit copula model for binary true endpoint and continuous surrogate endpoint

---

Description

[Superseded]

Development on `fit_copula_model_BinCont()` is complete. For new code, we recommend switching to `fit_copula_OrdCont()`, which is a more general function (it allows for ordinal endpoints, not just binary) and is still under active development.

Usage

```
fit_copula_model_BinCont(
  data,
  copula_family,
  marginal_surrogate,
  marginal_surrogate_estimator = NULL,
  twostep = FALSE,
  fitted_model = NULL,
  maxit = 500,
  method = "BFGS"
)
```

Arguments

`data` A data frame in the correct format (See details).

`copula_family` One of the following parametric copula families: `"clayton"`, `"frank"`, `"gaussian"`, or `"gumbel"`. The first element in `copula_family` corresponds to the control group, the second to the experimental group.

`marginal_surrogate` Marginal distribution for the surrogate. For all available options, see `?Surrogate::cdf_fun`.

`marginal_surrogate_estimator` Not yet implemented

`twostep` (boolean) if `TRUE`, the two step estimator implemented in `twostep_BinCont()` is used for estimation.

`fitted_model` Fitted model from which initial values are extracted. If `NULL` (default), standard initial values are used. This option intended for when a model is repeatedly fitted, e.g., in a bootstrap.

`maxit` Maximum number of iterations for the numeric optimization, defaults to 500.

`method` Optimization algorithm for maximizing the objective function. For all options, see `?maxLik::maxLik`. Defaults to `"BFGS"`.

Details

The function `fit_copula_model_BinCont()` fits the copula model for a continuous surrogate endpoint and binary true endpoint. Because the bivariate distributions of the surrogate-true endpoint pairs are functionally independent across treatment groups, a bivariate distribution is fitted in each treatment group separately.

Examples

```
# Load Schizophrenia data set.
data("Schizo_BinCont")
# Perform listwise deletion.
na = is.na(Schizo_BinCont$CGI_Bin) | is.na(Schizo_BinCont$PANSS)
X = Schizo_BinCont$PANSS[!na]
Y = Schizo_BinCont$CGI_Bin[!na]
Treat = Schizo_BinCont$Treat[!na]
```

```
# Ensure that the treatment variable is binary.
Treat = ifelse(Treat == 1, 1, 0)
data = data.frame(X,
                  Y,
                  Treat)
# Fit copula model.
fitted_model = fit_copula_model_BinCont(data, "clayton", "normal", twostep = FALSE)
# Perform sensitivity analysis with a very low number of replications.
sens_results = sensitivity_analysis_BinCont_copula(
  fitted_model,
  10,
  lower = c(-1,-1,-1,-1),
  upper = c(1, 1, 1, 1),
  n_prec = 1e3
)
```

fit_copula_OrdCont *Fit ordinal-continuous vine copula model*

Description

fit_copula_OrdCont() fits the ordinal-continuous vine copula model. See Details for more information about this model.

Usage

```
fit_copula_OrdCont(
  data,
  copula_family,
  marginal_S0,
  marginal_S1,
  K_T,
  start_copula,
  method = "BFGS",
  ...
)
```

Arguments

data — data frame with three columns in the following order: surrogate endpoint, true endpoint, and treatment indicator (0/1 coding). Ordinal endpoints should be integers starting from 1.

copula_family — One of the following parametric copula families: "clayton", "frank", "gaussian", or "gumbel". The first element in copula_family corresponds to the control group, the second to the experimental group.

marginal_S0, marginal_S1 — List with the following three elements (in order):

- Density function with first argument `x` and second argument `para` the parameter vector for this distribution.
- Distribution function with first argument `x` and second argument `para` the parameter vector for this distribution.
- Inverse distribution function with first argument `p` and second argument `para` the parameter vector for this distribution.
- The number of elements in `para`.
- A vector of starting values for `para`.

`K_T` Number of categories in the true endpoint.

`start_copula` Starting value for the copula parameter.

`method` Optimization algorithm for maximizing the objective function. For all options, see `?maxLik::maxLik`. Defaults to `"BFGS"`.

`...` Arguments passed on to `fit_copula_submodel_OrdCont`

`names_XY` Names for `X` and `Y`, respectively.

`twostep` (boolean) If `TRUE`, the starting values are fixed for the marginal distributions and only the copula parameter is estimated.

`start_Y` Starting values for the marginal distribution paramters for `Y`.

`X` First variable (Ordinal with K categories)

`Y` Second variable (Continuous)

`K` Number of categories in `X`.

`marginal_Y` List with the following five elements (in order):

- Density function with first argument `x` and second argument `para` the parameter vector for this distribution.
- Distribution function with first argument `x` and second argument `para`.
- Inverse distribution function with first argument `p` and second argument `para`.
- The number of elements in `para`.
- Starting values for `para`.

Details

Vine Copula Model for Ordinal Endpoints:

Following the Neyman-Rubin potential outcomes framework, we assume that each patient has four potential outcomes, two for each arm, represented by $\boldsymbol{Y} = (T_0, S_0, S_1, T_1)'$. Here, $\boldsymbol{Y_z} = (S_z, T_z)'$ are the potential surrogate and true endpoints under treatment $Z = z$. We will further assume that T is ordinal and S is continuous; consequently, the function argument `X` corresponds to T and `Y` to S. (The roles of S and T can be interchanged without loss of generality.)

We introduce latent variables to model $\boldsymbol{Y}$. Latent variables will be denoted by a tilde. For instance, if T_z is ordinal with K_T categories, then T_z is a function of the latent $\tilde{T}_z \sim N(0,1)$ as follows:

$$T_z = g_{T_z}(\tilde{T}_z; \boldsymbol{c}^{T_z}) = \begin{cases} 1 & \text{if } -\infty = c_0^{T_z} < \tilde{T}_z \leq c_1^{T_z} \\ \vdots & \\ k & \text{if } c_{k-1}^{T_z} < \tilde{T}_z \leq c_k^{T_z} \\ \vdots & \\ K & \text{if } c_{K_T-1}^{T_z} < \tilde{T}_z \leq c_{K_T}^{T_z} = \infty, \end{cases}$$

where $\boldsymbol{c}^{T_z} = (c_1^{T_z}, \cdots, c_{K_T-1}^{T_z})$. The latent counterpart of $\boldsymbol{Y}$ is again denoted by a tilde; for example, $\tilde{\boldsymbol{Y}} = (\tilde{T}_0, S_0, S_1, \tilde{T}_1)'$ if T_z is ordinal and S_z is continuous.

The vector of latent potential outcome $\tilde{\boldsymbol{Y}}$ is modeled with a D-vine copula as follows:

$$f_{\tilde{\mathbf{Y}}} = f_{\tilde{T}_0}\, f_{S_0}\, f_{S_1}\, f_{\tilde{T}_1} \cdot c_{\tilde{T}_0,S_0}\, c_{S_0,S_1}\, c_{S_1,\tilde{T}_1} \cdot c_{\tilde{T}_0,S_1;S_0}\, c_{S_0,\tilde{T}_1;S_1} \cdot c_{\tilde{T}_0,\tilde{T}_1;S_0,S_1},$$

where (i) f_{T_0}, f_{S_0}, f_{S_1}, and f_{T_1} are univariate density functions, (ii) c_{T_0,S_0}, c_{S_0,S_1}, and c_{S_1,T_1} are unconditional bivariate copula densities, and (iii) $c_{T_0,S_1;S_0}$, $c_{S_0,T_1;S_1}$, and $c_{T_0,T_1;S_0,S_1}$ are conditional bivariate copula densities (e.g., $c_{T_0,S_1;S_0}$ is the copula density of $(T_0, S_1)' \mid S_0$. We also make the simplifying assumption for all copulas.

Observed-Data Likelihood:

In practice, we only observe $(S_0, T_0)'$ or $(S_1, T_1)'$. Hence, to estimate the (identifiable) parameters of the D-vine copula model, we need to derive the observed-data likelihood. The observed-data loglikelihood for $(S_z, T_z)'$ is as follows:

$$f_{\mathbf{Y_z}}(s, t; \boldsymbol{\beta}) = \int_{c_{t-1}^{T_z}}^{+\infty} f_{\tilde{\mathbf{Y}}_{\boldsymbol{z}}}(s, x; \boldsymbol{\beta})\, dx - \int_{c_t^{T_z}}^{+\infty} f_{\tilde{\mathbf{Y}}_{\boldsymbol{z}}}(s, x; \boldsymbol{\beta})\, dx.$$

The above expression is used in `ordinal_continuous_loglik()` to compute the loglikelihood for the observed values for $Z = 0$ or $Z = 1$. In this function, `X` and `Y` correspond to T_z and S_z if T_z is ordinal and S_z continuous. Otherwise, `X` and `Y` correspond to S_z and T_z.

Value

Returns an S3 object that can be used to perform the sensitivity analysis with `sensitivity_analysis_copula()`.

Author(s)

Florian Stijven

See Also

`sensitivity_analysis_copula()`, `print.vine_copula_fit()`, `plot.vine_copula_fit()`

`fit_copula_OrdOrd` *Fit ordinal-ordinal vine copula model*

Description

`fit_copula_OrdOrd()` fits the ordinal-ordinal vine copula model. See Details for more information about this model.

Usage

```
fit_copula_OrdOrd(
  data,
  copula_family,
  K_S,
  K_T,
  start_copula,
  method = "BFGS",
  ...
)
```

Arguments

`data`	data frame with three columns in the following order: surrogate endpoint, true endpoint, and treatment indicator (0/1 coding). Ordinal endpoints should be integers starting from `1`.
`copula_family`	One of the following parametric copula families: `"clayton"`, `"frank"`, `"gaussian"`, or `"gumbel"`. The first element in `copula_family` corresponds to the control group, the second to the experimental group.
`K_S`, `K_T`	Number of categories in the surrogate and true endpoints.
`start_copula`	Starting value for the copula parameter.
`method`	Optimization algorithm for maximizing the objective function. For all options, see `?maxLik::maxLik`. Defaults to `"BFGS"`.
`...`	Extra argument to pass onto maxLik::maxLik

Details

Vine Copula Model for Ordinal Endpoints:

Following the Neyman-Rubin potential outcomes framework, we assume that each patient has four potential outcomes, two for each arm, represented by $\boldsymbol{Y} = (T_0, S_0, S_1, T_1)'$. Here, $\boldsymbol{Y_z} = (S_z, T_z)'$ are the potential surrogate and true endpoints under treatment $Z = z$.

The latent variable notation and D-vine copula model for $\boldsymbol{Y}$ is a straightforward extension of the notation in `ordinal_continuous_loglik()`.

Observed-Data Likelihood:

In practice, we only observe $(S_0, T_0)'$ or $(S_1, T_1)'$. Hence, to estimate the (identifiable) parameters of the D-vine copula model, we need to derive the observed-data likelihood. The observed-data loglikelihood for $(S_z, T_z)'$ is as follows:

$$f_{\boldsymbol{Y_z}}(s, t; \boldsymbol{\beta}) = P\bigg(c_{s-1}^{S_z} < \tilde{S}_z, c_{t-1}^{T_z} < \tilde{T}_z\bigg) - P\bigg(c_{s}^{S_z} < \tilde{S}_z, c_{t-1}^{T_z} < \tilde{T}_z\bigg) - P\bigg(c_{s-1}^{S_z} < \tilde{S}_z, c_{t}^{T_z} < \tilde{T}_z\bigg) + P\bigg(c_{s}^{S_z} < \tilde{S}_z, c_{t}^{T_z} <$$

The above expression is used in `ordinal_ordinal_loglik()` to compute the loglikelihood for the observed values for $Z = 0$ or $Z = 1$.

Value

Returns an S3 object that can be used to perform the sensitivity analysis with `sensitivity_analysis_copula()`.

Author(s)

Florian Stijven

See Also

sensitivity_analysis_copula(), print.vine_copula_fit(), plot.vine_copula_fit()

---

fit_copula_submodel_BinCont *Fit binary-continuous copula submodel*

Description

The fit_copula_submodel_BinCont() function fits the copula (sub)model fir a continuous surrogate and binary true endpoint with maximum likelihood.

Usage

```
fit_copula_submodel_BinCont(
  X,
  Y,
  copula_family,
  marginal_surrogate,
  method = "BFGS"
)
```

Arguments

X (numeric) Continuous surrogate variable

Y (integer) Binary true endpoint variable ($T_k \in \{0, 1\}$)

copula_family Copula family, one of the following:

- "clayton"
- "frank"
- "gumbel"
- "gaussian"

marginal_surrogate
Marginal distribution for the surrogate. For all available options, see ?Surrogate::cdf_fun.

method Optimization algorithm for maximizing the objective function. For all options, see ?maxLik::maxLik. Defaults to "BFGRS".

Value

A list with three elements:

- ml_fit: object of class `maxLik::maxLik` that contains the estimated copula model.
- marginal_S_dist: object of class `fitdistrplus::fitdist` that represents the marginal surrogate distribution.
- copula_family: string that indicates the copula family

`fit_copula_submodel_ContCont`

Fit ordinal-continuous copula submodel

Description

The `fit_copula_submodel_ContCont()` function fits the copula (sub)model for a continuous surrogate and true endpoint with maximum likelihood.

Usage

```
fit_copula_submodel_ContCont(
  X,
  Y,
  copula_family,
  marginal_X,
  marginal_Y,
  start_X,
  start_Y,
  start_copula,
  method = "BFGS",
  names_XY = c("Surr", "True"),
  twostep = FALSE,
  copula_transform = function(x) x,
  ...
)
```

Arguments

`X`	First variable (Continuous)
`Y`	Second variable (Continuous)
`copula_family`	Copula family, one of the following: • `"clayton"` • `"frank"` • `"gumbel"` • `"gaussian"`

`marginal_X`, `marginal_Y`
: List with the following three elements (in order):

  - Density function with first argument `x` and second argument `para` the parameter vector for this distribution.
  - Distribution function with first argument `x` and second argument `para` the parameter vector for this distribution.
  - Inverse distribution function with first argument `p` and second argument `para` the parameter vector for this distribution.
  - The number of elements in `para`.
  - A vector of starting values for `para`.

`start_X`, `start_Y`
: Starting values corresponding to `marginal_X` and `marginal_Y`.

`start_copula` Starting value for the copula parameter.

`method` Optimization algorithm for maximizing the objective function. For all options, see `?maxLik::maxLik`. Defaults to `"BFGS"`.

`names_XY` Names for `X` and `Y`, respectively.

`twostep` (boolean) If `TRUE`, the starting values are fixed for the marginal distributions and only the copula parameter is estimated.

`copula_transform`
: Used for reparameterizing the copula parameter. `copula_transform()` backtransforms the transformed copula parameter to the original scale. Note that `start_copula` should be specified on the transformed scale.

`...` Extra argument to pass onto maxLik::maxLik

Value

A list with five elements:

- ml_fit: object of class `maxLik::maxLik` that contains the estimated copula model.
- marginal_X: list with the estimated cdf, pdf/pmf, and inverse cdf for X.
- marginal_Y: list with the estimated cdf, pdf/pmf, and inverse cdf for X.
- copula_family: string that indicates the copula family
- data: data frame containing `X` and `Y`
- names_XY: The names (i.e., `"Surr"` and `"True"`) for `X` and `Y`

See Also

`continuous_continuous_loglik()`

fit_copula_submodel_OrdCont

Fit ordinal-continuous copula submodel

Description

The `fit_copula_submodel_OrdCont()` function fits the copula (sub)model for a continuous surrogate and an ordinal true endpoint with maximum likelihood.

Usage

```
fit_copula_submodel_OrdCont(
  X,
  Y,
  copula_family,
  marginal_Y,
  start_Y,
  start_copula,
  method = "BFGS",
  K,
  names_XY = c("Surr", "True"),
  twostep = FALSE,
  ...
)
```

Arguments

`X`	First variable (Ordinal with K categories)
`Y`	Second variable (Continuous)
`copula_family`	Copula family, one of the following: • `"clayton"` • `"frank"` • `"gumbel"` • `"gaussian"`
`marginal_Y`	List with the following five elements (in order): • Density function with first argument `x` and second argument `para` the parameter vector for this distribution. • Distribution function with first argument `x` and second argument `para`. • Inverse distribution function with first argument `p` and second argument `para`. • The number of elements in `para`. • Starting values for `para`.
`start_Y`	Starting values for the marginal distribution paramters for `Y`.
`start_copula`	Starting value for the copula parameter.

`method` Optimization algorithm for maximizing the objective function. For all options, see `?maxLik::maxLik`. Defaults to `"BFGS"`.

`K` Number of categories in `X`.

`names_XY` Names for `X` and `Y`, respectively.

`twostep` (boolean) If `TRUE`, the starting values are fixed for the marginal distributions and only the copula parameter is estimated.

`...` Extra argument to pass onto maxLik::maxLik

Value

A list with five elements:

- ml_fit: object of class `maxLik::maxLik` that contains the estimated copula model.
- marginal_X: list with the estimated cdf, pdf/pmf, and inverse cdf for X.
- marginal_Y: list with the estimated cdf, pdf/pmf, and inverse cdf for X.
- copula_family: string that indicates the copula family
- data: data frame containing `X` and `Y`
- names_XY: The names (i.e., `"Surr"` and `"True"`) for `X` and `Y`

See Also

`ordinal_continuous_loglik()`

`fit_copula_submodel_OrdOrd` *Fit ordinal-continuous copula submodel*

Description

The `fit_copula_submodel_OrdOrd()` function fits the copula (sub)model for an ordinal surrogate and true endpoint with maximum likelihood.

Usage

```
fit_copula_submodel_OrdOrd(
  X,
  Y,
  copula_family,
  start_copula,
  method = "BFGS",
  K_X,
  K_Y,
  names_XY = c("Surr", "True"),
  twostep = FALSE,
  ...
)
```

Arguments

`X` First variable (Ordinal with K_X categories)

`Y` Second variable (Ordinal with K_Y categories)

`copula_family` Copula family, one of the following:

- `"clayton"`
- `"frank"`
- `"gumbel"`
- `"gaussian"`

`start_copula` Starting value for the copula parameter.

`method` Optimization algorithm for maximizing the objective function. For all options, see `?maxLik::maxLik`. Defaults to `"BFGS"`.

`K_X` Number of categories in `X`.

`K_Y` Number of categories in `Y`.

`names_XY` Names for `X` and `Y`, respectively.

`twostep` (boolean) If `TRUE`, the starting values are fixed for the marginal distributions and only the copula parameter is estimated.

`...` Extra argument to pass onto maxLik::maxLik

Value

A list with five elements:

- ml_fit: object of class `maxLik::maxLik` that contains the estimated copula model.
- marginal_X: list with the estimated cdf, pdf/pmf, and inverse cdf for X.
- marginal_Y: list with the estimated cdf, pdf/pmf, and inverse cdf for X.
- copula_family: string that indicates the copula family
- data: data frame containing `X` and `Y`
- names_XY: The names (i.e., `"Surr"` and `"True"`) for `X` and `Y`

`fit_model_SurvSurv` *Fit Survival-Survival model*

Description

The function `fit_model_SurvSurv()` fits the copula model for time-to-event surrogate and true endpoints (Stijven et al., 2022). Because the bivariate distributions of the surrogate-true endpoint pairs are functionally independent across treatment groups, a bivariate distribution is fitted in each treatment group separately. The marginal distributions are based on the Royston-Parmar survival model (Royston and Parmar, 2002).

Usage

```
fit_model_SurvSurv(
  data,
  copula_family,
  n_knots = 2,
  fitted_model = NULL,
  method = "BFGS",
  maxit = 500
)
```

Arguments

`data` A data frame in the correct format (See details).

`copula_family` One of the following parametric copula families: `"clayton"`, `"frank"`, `"gaussian"`, or `"gumbel"`. The first element in `copula_family` corresponds to the control group, the second to the experimental group.

`n_knots` Number of internal knots for the Royston-Parmar survival models for $\tilde{S}_0$, T_0, $\tilde{S}_1$, and T_1. If `length(n_knots) == 1`, the same number of knots are assumed for the four marginal distributions.

`fitted_model` Fitted model from which initial values are extracted. If `NULL` (default), standard initial values are used. This option intended for when a model is repeatedly fitted, e.g., in a bootstrap.

`method` Optimization algorithm for maximizing the objective function. For all options, see `?maxLik::maxLik`. Defaults to `"BFGS"`.

`maxit` Maximum number of iterations for the numeric optimization, defaults to 500.

Value

Returns an S3 object that can be used to perform the sensitivity analysis with `sensitivity_analysis_SurvSurv_copula()`.

Model

In the causal-inference approach to evaluating surrogate endpoints, the first step is to estimate the joint distribution of the relevant potential outcomes. Let $(T_0, S_0, S_1, T_1)'$. denote the vector of potential outcomes where $(S_k, T_k)'$ is the pair of potential outcomes under treatment $Z = k$. T refers to the true endpoint, e.g., overall survival. S refers to the composite surrogate endpoint, e.g., progression-free-survival. Because S is usually a composite endpoint with death as possible event, modeling difficulties arise because $Pr(S_k = T_k) > 0$.

Due to difficulties in modeling the composite surrogate and the true endpoint jointly, the time-to-surrogate event ($\tilde{S}$) is modeled instead of the time-to-composite surrogate event (S). Using this new variable, $\tilde{S}$, a D-vine copula model is proposed for $(T_0, \tilde{S}_0, \tilde{S}_1, T_1)'$ in Stijven et al. (2022). However, only the following bivariate distributions are identifiable $(T_k, \tilde{S}_k)'$ for $k = 0, 1$. The margins in these bivariate distributions are based on the Royston-Parmar survival model (Roystona and Parmar, 2002). The association is modeled through two copulas of the same parametric form, but with unique copula parameters.

Two modelling choices are made before estimating the two bivariate distributions described in the previous paragraph:

- The number of internal knots for the Royston-Parmar survival models. This is specified through the `n_knots` argument. The number of knots is assumed to be equal across the four margins.
- The parametric family of the bivariate copulas. The parametric family is assumed to be equal across treatment groups. This choice is specified through the `copula_family` argument.

Data Format

The data frame should have the semi-competing risks format. The columns must be ordered as follows:

- time to surrogate event, true event, or independent censoring; whichever comes first
- time to true event, or independent censoring; whichever comes first
- treatment indicator: 0 or 1
- surrogate event indicator: 1 if surrogate event is observed, 0 otherwise
- true event indicator: 1 if true event is observed, 0 otherwise

Note that according to the methodology in Stijven et al. (2022), the surrogate event must not be the composite event. For example, when the surrogacy of progression-free survival for overall survival is evaluated. The surrogate event is progression, but not the composite event of progression or death.

Author(s)

Florian Stijven

References

Stijven, F., Alonso, a., Molenberghs, G., Van Der Elst, W., Van Keilegom, I. (2024). An information-theoretic approach to the evaluation of time-to-event surrogates for time-to-event true endpoints based on causal inference.

Royston, P., & Parmar, M. K. (2002). Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. Statistics in medicine, 21(15), 2175-2197.

See Also

`sensitivity_analysis_SurvSurv_copula()`

Examples

```
if(require(Surrogate)) {
  data("Ovarian")
  #For simplicity, data is not recoded to semi-competing risks format, but is
  #left in the composite event format.
  data = data.frame(Ovarian$Pfs,
                    Ovarian$Surv,
                    Ovarian$Treat,
                    Ovarian$PfsInd,
                  Ovarian$SurvInd)
  Surrogate::fit_model_SurvSurv(data = data,
                                copula_family = "clayton",
                                n_knots = 1)
}
```

FixedBinBinIT — *Fits (univariate) fixed-effect models to assess surrogacy in the binary-binary case based on the Information-Theoretic framework*

Description

The function `FixedBinBinIT` uses the information-theoretic approach (Alonso & Molenberghs, 2007) to estimate trial- and individual-level surrogacy based on fixed-effect models when both S and T are binary variables. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below.

Usage

```
FixedBinBinIT(Dataset, Surr, True, Treat, Trial.ID, Pat.ID,
Model=c("Full"), Weighted=TRUE, Min.Trial.Size=2, Alpha=.05,
Number.Bootstraps=50, Seed=sample(1:1000, size=1))
```

Arguments

`Dataset` A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.

`Surr` The name of the variable in `Dataset` that contains the surrogate endpoint values.

`True` The name of the variable in `Dataset` that contains the true endpoint values.

`Treat` The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.

`Trial.ID` The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.

`Pat.ID` The name of the variable in `Dataset` that contains the patient's ID.

`Model` The type of model that should be fitted, i.e., `Model=c("Full")`, `Model=c("Reduced")`, or `Model=c("SemiReduced")`. See the **Details** section below. Default `Model=c("Full")`.

`Weighted` Logical. In practice it is often the case that different trials (or other clustering units) have different sample sizes. Univariate models are used to assess surrogacy in the information-theoretic approach, so it can be useful to adjust for heterogeneity in information content between the trial-specific contributions (particularly when trial-level surrogacy measures are of primary interest and when the heterogeneity in sample sizes is large). If `Weighted=TRUE`, weighted regression models are fitted. If `Weighted=FALSE`, unweighted regression analyses are conducted. See the **Details** section below. Default `TRUE`.

`Min.Trial.Size` The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R^2_h and R^2_{ht}. Default 0.05.

`Number.Bootstraps`

The standard errors and confidence intervals for R^2_h, $R^2_{b.ind}$ and $R^2_{h.ind}$ are determined based on a bootstrap procedure. `Number.Bootstraps` specifies the number of bootstrap samples that are used. Default 50.

`Seed` The seed to be used in the bootstrap procedure. Default $sample(1 : 1000, size = 1)$.

Details

Individual-level surrogacy

The following univariate generalised linear models are fitted:

$$g_T(E(T_{ij})) = \mu_{Ti} + \beta_i Z_{ij},$$
$$g_T(E(T_{ij}|S_{ij})) = \gamma_{0i} + \gamma_{1i} Z_{ij} + \gamma_{2i} S_{ij},$$

where i and j are the trial and subject indicators, g_T is an appropriate link function (i.e., a logit link when binary endpoints are considered), S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, and Z_{ij} is the treatment indicator for subject j in trial i. μ_{Ti} and β_i are the trial-specific intercepts and treatment-effects on the true endpoint in trial i. γ_{0i} and γ_{1i} are the trial-specific intercepts and treatment-effects on the true endpoint in trial i after accounting for the effect of the surrogate endpoint.

The -2 log likelihood values of the previous models in each of the i trials (i.e., L_{1i} and L_{2i}, respectively) are subsequently used to compute individual-level surrogacy based on the so-called Variance Reduction Factor (VFR; for details, see Alonso & Molenberghs, 2007):

$$R^2_h = 1 - \frac{1}{N} \sum_i exp\left(-\frac{L_{2i} - L_{1i}}{n_i}\right),$$

where N is the number of trials and n_i is the number of patients within trial i.

When it can be assumed (i) that the treatment-corrected association between the surrogate and the true endpoint is constant across trials, or (ii) when all data come from a single clinical trial (i.e., when $N = 1$), the previous expression simplifies to:

$$R^2_{h.ind} = 1 - exp\left(-\frac{L_2 - L_1}{N}\right).$$

The upper bound does not reach to 1 when T is binary, i.e., its maximum is 0.75. Kent (1983) claims that 0.75 is a reasonable upper bound and thus $R^2_{h.ind}$ can usually be interpreted without paying special consideration to the discreteness of T. Alternatively, to address the upper bound problem, a scaled version of the mutual information can be used when both S and T are binary (Joe, 1989):

$$R^2_{b.ind} = \frac{I(T,S)}{min[H(T),H(S)]},$$

where the entropy of T and S in the previous expression can be estimated using the log likelihood functions of the GLMs shown above.

Trial-level surrogacy

When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), trial-level surrogacy is assessed by fitting the following univariate models:

$$S_{ij} = \mu_{Si} + \alpha_i Z_{ij} + \varepsilon_{Sij}, (1)$$

$$T_{ij} = \mu_{Ti} + \beta_i Z_{ij} + \varepsilon_{Tij}, (1)$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_{Si} and μ_{Ti} are the fixed trial-specific intercepts for S and T, and α_i and β_i are the fixed trial-specific treatment effects on S and T, respectively. The error terms ε_{Sij} and ε_{Tij} are assumed to be independent.

When a reduced model is requested by the user (by using the argument `Model=c("Reduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_S + \alpha_i Z_{ij} + \varepsilon_{Sij}, (2)$$

$$T_{ij} = \mu_T + \beta_i Z_{ij} + \varepsilon_{Tij}, (2)$$

where μ_S and μ_T are the common intercepts for S and T. The other parameters are the same as defined above, and ε_{Sij} and ε_{Tij} are again assumed to be independent.

When the user requested a full model approach (by using the argument `Model=c("Full")` in the function call, i.e., when models (1) were fitted), the following model is subsequently fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\mu_{Si}} + \lambda_2 \widehat{\alpha}_i + \varepsilon_i, (3)$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on models (1) (see above). When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), model (3) is a weighted regression model (with weights based on the number of observations in trial i). The -2 log likelihood value of the (weighted or unweighted) model (3) (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based based on the Variance Reduction Factor (for details, see Alonso & Molenberghs, 2007):

$$R^2_{ht} = 1 - exp\left(-\frac{L_1 - L_0}{N}\right),$$

where N is the number of trials.

When a semi-reduced or reduced model is requested (by using the argument `Model=c("SemiReduced")` or `Model=c("Reduced")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameter estimates for β_i and α_i are based on models (1) when a semi-reduced model is fitted or on models (2) when a reduced model is fitted. The -2 log likelihood value of this (weighted or unweighted) model (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based on the reduction in the likelihood (as described above).

Value

An object of class `FixedBinBinIT` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Trial.Spec.Results`
A `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`R2ht` A `data.frame` that contains the trial-level surrogacy estimate and its confidence interval.

`R2h.ind` A `data.frame` that contains the individual-level surrogacy estimate $R^2_{h.ind}$ (single-trial based estimate) and its confidence interval.

`R2h` A `data.frame` that contains the individual-level surrogacy estimate R^2_h (cluster-based estimate) and its confidence interval (based on a bootsrtrap).

`R2b.ind` A `data.frame` that contains the individual-level surrogacy estimate $R^2_{b.ind}$ (single-trial based estimate accounting for upper bound) and its confidence interval (based on a bootstrap).

`R2h.Ind.By.Trial`
A `data.frame` that contains individual-level surrogacy estimates R^2_{hInd} (cluster-based estimates) and their confidence interval for each of the trials seperately.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

Joe, H. (1989). Relative entropy measures of multivariate dependence. *Journal of the American Statistical Association, 84,* 157-164.

Kent, T. J. (1983). Information gain as a general measure of correlation. *Biometrica, 70,* 163-173.

See Also

FixedBinContIT, FixedContBinIT, plot Information-Theoretic BinCombn

Examples

```
## Not run:  # Time consuming (>5sec) code part
# Generate data with continuous Surr and True
Sim.Data.MTS(N.Total=5000, N.Trial=50, R.Trial.Target=.9, R.Indiv.Target=.9,
             Fixed.Effects=c(0, 0, 0, 0), D.aa=10, D.bb=10, Seed=1,
             Model=c("Full"))
# Dichtomize Surr and True
Surr_Bin <- Data.Observed.MTS$Surr
Surr_Bin[Data.Observed.MTS$Surr>.5] <- 1
Surr_Bin[Data.Observed.MTS$Surr<=.5] <- 0
True_Bin <- Data.Observed.MTS$True
True_Bin[Data.Observed.MTS$True>.15] <- 1
True_Bin[Data.Observed.MTS$True<=.15] <- 0
Data.Observed.MTS$Surr <- Surr_Bin
Data.Observed.MTS$True <- True_Bin

# Assess surrogacy using info-theoretic framework
Fit <- FixedBinBinIT(Dataset = Data.Observed.MTS, Surr = Surr,
True = True, Treat = Treat, Trial.ID = Trial.ID,
Pat.ID = Pat.ID, Number.Bootstraps=100)

# Examine results
summary(Fit)
plot(Fit, Trial.Level = FALSE, Indiv.Level.By.Trial=TRUE)
plot(Fit, Trial.Level = TRUE, Indiv.Level.By.Trial=FALSE)

## End(Not run)
```

FixedBinContIT *Fits (univariate) fixed-effect models to assess surrogacy in the case where the true endpoint is binary and the surrogate endpoint is continuous (based on the Information-Theoretic framework)*

Description

The function FixedBinContIT uses the information-theoretic approach (Alonso & Molenberghs, 2007) to estimate trial- and individual-level surrogacy based on fixed-effect models when T is binary and S is continuous. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below.

Usage

```
FixedBinContIT(Dataset, Surr, True, Treat, Trial.ID, Pat.ID,
Model=c("Full"), Weighted=TRUE, Min.Trial.Size=2, Alpha=.05,
Number.Bootstraps=50,Seed=sample(1:1000, size=1))
```

Arguments

Dataset — A data.frame that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.

Surr — The name of the variable in Dataset that contains the surrogate endpoint values.

True — The name of the variable in Dataset that contains the true endpoint values.

Treat — The name of the variable in Dataset that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.

Trial.ID — The name of the variable in Dataset that contains the trial ID to which the patient belongs.

Pat.ID — The name of the variable in Dataset that contains the patient's ID.

Model — The type of model that should be fitted, i.e., Model=c("Full"), Model=c("Reduced"), or Model=c("SemiReduced"). See the **Details** section below. Default Model=c("Full").

Weighted — Logical. In practice it is often the case that different trials (or other clustering units) have different sample sizes. Univariate models are used to assess surrogacy in the information-theoretic approach, so it can be useful to adjust for heterogeneity in information content between the trial-specific contributions (particularly when trial-level surrogacy measures are of primary interest and when the heterogeneity in sample sizes is large). If Weighted=TRUE, weighted regression models are fitted. If Weighted=FALSE, unweighted regression analyses are conducted. See the **Details** section below. Default TRUE.

Min.Trial.Size — The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by Min.Trial.Size, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R^2_h and R^2_{ht}. Default 0.05.

`Number.Bootstraps`
The standard errors and confidence intervals for R^2_h and $R^2_{h.ind}$ are determined based on a bootstrap procedure. `Number.Bootstraps` specifies the number of bootstrap samples that are used. Default 50.

`Seed` The seed to be used in the bootstrap procedure. Default $sample(1:1000, size=1)$.

Details

Individual-level surrogacy

The following univariate generalised linear models are fitted:

$$g_T(E(T_{ij})) = \mu_{Ti} + \beta_i Z_{ij},$$
$$g_T(E(T_{ij}|S_{ij})) = \gamma_{0i} + \gamma_{1i} Z_{ij} + \gamma_{2i} S_{ij},$$

where i and j are the trial and subject indicators, g_T is an appropriate link function (i.e., a logit link for binary endpoints and an identity link for normally distributed continuous endpoints), S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, and Z_{ij} is the treatment indicator for subject j in trial i. μ_{Ti} and β_i are the trial-specific intercepts and treatment-effects on the true endpoint in trial i. γ_{0i} and γ_{1i} are the trial-specific intercepts and treatment-effects on the true endpoint in trial i after accounting for the effect of the surrogate endpoint.

The -2 log likelihood values of the previous models in each of the i trials (i.e., L_{1i} and L_{2i}, respectively) are subsequently used to compute individual-level surrogacy based on the so-called Variance Reduction Factor (VFR; for details, see Alonso & Molenberghs, 2007):

$$R^2_h = 1 - \frac{1}{N}\sum_i exp\left(-\frac{L_{2i} - L_{1i}}{n_i}\right),$$

where N is the number of trials and n_i is the number of patients within trial i.

When it can be assumed (i) that the treatment-corrected association between the surrogate and the true endpoint is constant across trials, or (ii) when all data come from a single clinical trial (i.e., when $N = 1$), the previous expression simplifies to:

$$R^2_{h.ind} = 1 - exp\left(-\frac{L_2 - L_1}{N}\right).$$

The upper bound does not reach to 1 when T is binary, i.e., its maximum is 0.75. Kent (1983) claims that 0.75 is a reasonable upper bound and thus $R^2_{h.ind}$ can usually be interpreted without paying special consideration to the discreteness of T. Alternatively, to address the upper bound problem, a scaled version of the mutual information can be used when both S and T are binary (Joe, 1989):

$$R^2_{b.ind} = \frac{I(T,S)}{min[H(T), H(S)]},$$

where the entropy of T and S in the previous expression can be estimated using the log likelihood functions of the GLMs shown above.

Trial-level surrogacy

When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), trial-level surrogacy is assessed by fitting the following univariate models:

$$S_{ij} = \mu_{Si} + \alpha_i Z_{ij} + \varepsilon_{Sij}, (1)$$
$$T_{ij} = \mu_{Ti} + \beta_i Z_{ij} + \varepsilon_{Tij}, (1)$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_{Si} and μ_{Ti} are the fixed trial-specific intercepts for S and T, and α_i and β_i are the fixed trial-specific treatment effects on S and T, respectively. The error terms ε_{Sij} and ε_{Tij} are assumed to be independent.

When a reduced model is requested by the user (by using the argument `Model=c("Reduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_S + \alpha_i Z_{ij} + \varepsilon_{Sij}, (2)$$
$$T_{ij} = \mu_T + \beta_i Z_{ij} + \varepsilon_{Tij}, (2)$$

where μ_S and μ_T are the common intercepts for S and T. The other parameters are the same as defined above, and ε_{Sij} and ε_{Tij} are again assumed to be independent.

When the user requested a full model approach (by using the argument `Model=c("Full")` in the function call, i.e., when models (1) were fitted), the following model is subsequently fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\mu_{Si}} + \lambda_2 \widehat{\alpha}_i + \varepsilon_i, (3)$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on models (1) (see above). When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), model (3) is a weighted regression model (with weights based on the number of observations in trial i). The -2 log likelihood value of the (weighted or unweighted) model (3) (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based based on the Variance Reduction Factor (for details, see Alonso & Molenberghs, 2007):

$$R^2_{ht} = 1 - exp\left(-\frac{L_1 - L_0}{N}\right),$$

where N is the number of trials.

When a semi-reduced or reduced model is requested (by using the argument `Model=c("SemiReduced")` or `Model=c("Reduced")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameter estimates for β_i and α_i are based on models (1) when a semi-reduced model is fitted or on models (2) when a reduced model is fitted. The -2 log likelihood value of this (weighted or unweighted) model (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based on the reduction in the likelihood (as described above).

Value

An object of class `FixedBinContIT` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Trial.Spec.Results`
A `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`R2ht` A `data.frame` that contains the trial-level surrogacy estimate and its confidence interval.

`R2h.ind` A `data.frame` that contains the individual-level surrogacy estimate $R^2_{h.ind}$ (single-trial based estimate) and its confidence interval.

`R2h` A `data.frame` that contains the individual-level surrogacy estimate R^2_h (cluster-based estimate) and its confidence interval (bootstrap-based).

`R2b.ind` A `data.frame` that contains the individual-level surrogacy estimate $R^2_{b.ind}$ (single-trial based estimate accounting for upper bound) and its confidence interval (based on a bootstrap).

`R2h.Ind.By.Trial`
A `data.frame` that contains individual-level surrogacy estimates R^2_h (cluster-based estimate) and their confidence interval for each of the trials seperately.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

Joe, H. (1989). Relative entropy measures of multivariate dependence. *Journal of the American Statistical Association, 84,* 157-164.

Kent, T. J. (1983). Information gain as a general measure of correlation. *Biometrica, 70,* 163-173.

See Also

FixedBinBinIT, FixedContBinIT, plot Information-Theoretic BinCombn

Examples

```
## Not run:  # Time consuming (>5sec) code part
# Generate data with continuous Surr and True
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8,
R.Indiv.Target=.8, Seed=123, Model="Full")

# Make T binary
Data.Observed.MTS$True_Bin <- Data.Observed.MTS$True
Data.Observed.MTS$True_Bin[Data.Observed.MTS$True>=0] <- 1
Data.Observed.MTS$True_Bin[Data.Observed.MTS$True<0] <- 0

# Analyze data
Fit <- FixedBinContIT(Dataset = Data.Observed.MTS, Surr = Surr,
True = True_Bin, Treat = Treat, Trial.ID = Trial.ID, Pat.ID = Pat.ID,
Model = "Full", Number.Bootstraps=50)

# Examine results
summary(Fit)
plot(Fit, Trial.Level = FALSE, Indiv.Level.By.Trial=TRUE)
plot(Fit, Trial.Level = TRUE, Indiv.Level.By.Trial=FALSE)

## End(Not run)
```

FixedContBinIT *Fits (univariate) fixed-effect models to assess surrogacy in the case where the true endpoint is continuous and the surrogate endpoint is binary (based on the Information-Theoretic framework)*

Description

The function FixedContBinIT uses the information-theoretic approach (Alonso & Molenberghs, 2007) to estimate trial- and individual-level surrogacy based on fixed-effect models when T is continuous normally distributed and S is binary. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below.

Usage

```
FixedContBinIT(Dataset, Surr, True, Treat, Trial.ID, Pat.ID,
Model=c("Full"), Weighted=TRUE, Min.Trial.Size=2, Alpha=.05,
Number.Bootstraps=50,Seed=sample(1:1000, size=1))
```

Arguments

`Dataset` A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.

`Surr` The name of the variable in `Dataset` that contains the surrogate endpoint values.

`True` The name of the variable in `Dataset` that contains the true endpoint values.

`Treat` The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.

`Trial.ID` The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.

`Pat.ID` The name of the variable in `Dataset` that contains the patient's ID.

`Model` The type of model that should be fitted, i.e., `Model=c("Full")`, `Model=c("Reduced")`, or `Model=c("SemiReduced")`. See the **Details** section below. Default `Model=c("Full")`.

`Weighted` Logical. In practice it is often the case that different trials (or other clustering units) have different sample sizes. Univariate models are used to assess surrogacy in the information-theoretic approach, so it can be useful to adjust for heterogeneity in information content between the trial-specific contributions (particularly when trial-level surrogacy measures are of primary interest and when the heterogeneity in sample sizes is large). If `Weighted=TRUE`, weighted regression models are fitted. If `Weighted=FALSE`, unweighted regression analyses are conducted. See the **Details** section below. Default `TRUE`.

`Min.Trial.Size` The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R_h^2 and R_{ht}^2. Default 0.05.

`Number.Bootstraps`

The standard error and confidence interval for $R_{h.ind}^2$ is determined based on a bootstrap procedure. `Number.Bootstraps` specifies the number of bootstrap samples that are used. Default 50.

`Seed` The seed to be used in the bootstrap procedure. Default $sample(1:1000, size=1)$.

Details

Individual-level surrogacy

The following univariate generalised linear models are fitted:

$$g_T(E(T_{ij})) = \mu_{Ti} + \beta_i Z_{ij},$$
$$g_T(E(T_{ij}|S_{ij})) = \gamma_{0i} + \gamma_{1i} Z_{ij} + \gamma_{2i} S_{ij},$$

where i and j are the trial and subject indicators, g_T is an appropriate link function (i.e., a logit link for binary endpoints and an identity link for normally distributed continuous endpoints), S_{ij}

and T_{ij} are the surrogate and true endpoint values of subject j in trial i, and Z_{ij} is the treatment indicator for subject j in trial i. μ_{Ti} and β_i are the trial-specific intercepts and treatment-effects on the true endpoint in trial i. γ_{0i} and γ_{1i} are the trial-specific intercepts and treatment-effects on the true endpoint in trial i after accounting for the effect of the surrogate endpoint.

The -2 log likelihood values of the previous models in each of the i trials (i.e., L_{1i} and L_{2i}, respectively) are subsequently used to compute individual-level surrogacy based on the so-called Variance Reduction Factor (VFR; for details, see Alonso & Molenberghs, 2007):

$$R_h^2 = 1 - \frac{1}{N}\sum_i exp\left(-\frac{L_{2i} - L_{1i}}{n_i}\right),$$

where N is the number of trials and n_i is the number of patients within trial i.

When it can be assumed (i) that the treatment-corrected association between the surrogate and the true endpoint is constant across trials, or (ii) when all data come from a single clinical trial (i.e., when $N = 1$), the previous expression simplifies to:

$$R_{h.ind}^2 = 1 - exp\left(-\frac{L_2 - L_1}{N}\right).$$

Trial-level surrogacy

When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), trial-level surrogacy is assessed by fitting the following univariate models:

$$S_{ij} = \mu_{Si} + \alpha_i Z_{ij} + \varepsilon_{Sij}, (1)$$
$$T_{ij} = \mu_{Ti} + \beta_i Z_{ij} + \varepsilon_{Tij}, (1)$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_{Si} and μ_{Ti} are the fixed trial-specific intercepts for S and T, and α_i and β_i are the fixed trial-specific treatment effects on S and T, respectively. The error terms ε_{Sij} and ε_{Tij} are assumed to be independent.

When a reduced model is requested by the user (by using the argument `Model=c("Reduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_S + \alpha_i Z_{ij} + \varepsilon_{Sij}, (2)$$
$$T_{ij} = \mu_T + \beta_i Z_{ij} + \varepsilon_{Tij}, (2)$$

where μ_S and μ_T are the common intercepts for S and T. The other parameters are the same as defined above, and ε_{Sij} and ε_{Tij} are again assumed to be independent.

When the user requested a full model approach (by using the argument `Model=c("Full")` in the function call, i.e., when models (1) were fitted), the following model is subsequently fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\mu_{Si}} + \lambda_2 \widehat{\alpha}_i + \varepsilon_i, (3)$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on models (1) (see above). When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), model (3) is a weighted regression model (with weights based on the number of observations in trial i). The -2 log likelihood value of the (weighted or unweighted) model (3) (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based based on the Variance Reduction Factor (for details, see Alonso & Molenberghs, 2007):

$$R^2_{ht} = 1 - exp\left(-\frac{L_1 - L_0}{N}\right),$$

where N is the number of trials.

When a semi-reduced or reduced model is requested (by using the argument `Model=c("SemiReduced")` or `Model=c("Reduced")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameter estimates for β_i and α_i are based on models (1) when a semi-reduced model is fitted or on models (2) when a reduced model is fitted. The -2 log likelihood value of this (weighted or unweighted) model (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based on the reduction in the likelihood (as described above).

Value

An object of class `FixedContBinIT` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Trial.Spec.Results`
A `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`R2ht` A `data.frame` that contains the trial-level surrogacy estimate and its confidence interval.

`R2h` A `data.frame` that contains the individual-level surrogacy estimate R^2_h (cluster-based estimate) and its confidence interval.

R2h.ind A data.frame that contains the individual-level surrogacy estimate $R^2_{h.ind}$ (single-trial based estimate) and its confidence interval based on a bootstrap. The $R^2_{h.ind}$ shown is the mean of the bootstrapped values.

R2h.Ind.By.Trial A data.frame that contains individual-level surrogacy estimates R^2_h (cluster-based estimate) and their confidence interval for each of the trials seperately.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

See Also

FixedBinBinIT, FixedBinContIT, plot Information-Theoretic BinCombn

Examples

```
## Not run:  # Time consuming (>5sec) code part
# Generate data with continuous Surr and True
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8,
R.Indiv.Target=.8, Seed=123, Model="Full")

# Make S binary
Data.Observed.MTS$Surr_Bin <- Data.Observed.MTS$Surr
Data.Observed.MTS$Surr_Bin[Data.Observed.MTS$Surr>=0] <- 1
Data.Observed.MTS$Surr_Bin[Data.Observed.MTS$Surr<0] <- 0

# Analyze data
Fit <- FixedContBinIT(Dataset = Data.Observed.MTS, Surr = Surr_Bin,
True = True, Treat = Treat, Trial.ID = Trial.ID, Pat.ID = Pat.ID,
Model = "Full", Number.Bootstraps=50)

# Examine results
summary(Fit)
plot(Fit, Trial.Level = FALSE, Indiv.Level.By.Trial=TRUE)
plot(Fit, Trial.Level = TRUE, Indiv.Level.By.Trial=FALSE)

## End(Not run)
```

FixedContContIT — *Fits (univariate) fixed-effect models to assess surrogacy in the continuous-continuous case based on the Information-Theoretic framework*

Description

The function `FixedContContIT` uses the information-theoretic approach (Alonso & Molenberghs, 2007) to estimate trial- and individual-level surrogacy based on fixed-effect models when both S and T are continuous variables. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below.

Usage

```
FixedContContIT(Dataset, Surr, True, Treat, Trial.ID, Pat.ID,
Model=c("Full"), Weighted=TRUE, Min.Trial.Size=2,
Alpha=.05, Number.Bootstraps=500, Seed=sample(1:1000, size=1))
```

Arguments

`Dataset`	A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.
`Surr`	The name of the variable in `Dataset` that contains the surrogate endpoint values.
`True`	The name of the variable in `Dataset` that contains the true endpoint values.
`Treat`	The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.
`Trial.ID`	The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.
`Pat.ID`	The name of the variable in `Dataset` that contains the patient's ID.
`Model`	The type of model that should be fitted, i.e., `Model=c("Full")`, `Model=c("Reduced")`, or `Model=c("SemiReduced")`. See the **Details** section below. Default `Model=c("Full")`.
`Weighted`	Logical. In practice it is often the case that different trials (or other clustering units) have different sample sizes. Univariate models are used to assess surrogacy in the information-theoretic approach, so it can be useful to adjust for heterogeneity in information content between the trial-specific contributions (particularly when trial-level surrogacy measures are of primary interest and when the heterogeneity in sample sizes is large). If `Weighted=TRUE`, weighted regression models are fitted. If `Weighted=FALSE`, unweighted regression analyses are conducted. See the **Details** section below. Default `TRUE`.
`Min.Trial.Size`	The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R_h^2 and R_{ht}^2. Default 0.05.

`Number.Bootstraps`
The standard error and confidence interval for R_h^2 is determined based on a bootstrap procedure. `Number.Bootstraps` specifies the number of bootstrap samples that are used. Default 500.

`Seed` The seed to be used in the bootstrap procedure. Default $sample(1 : 1000, size = 1)$.

Details

Individual-level surrogacy

The following univariate generalised linear models are fitted:

$$g_T(E(T_{ij})) = \mu_{Ti} + \beta_i Z_{ij},$$
$$g_T(E(T_{ij}|S_{ij})) = \gamma_{0i} + \gamma_{1i} Z_{ij} + \gamma_{2i} S_{ij},$$

where i and j are the trial and subject indicators, g_T is an appropriate link function (i.e., an identity link when a continuous true endpoint is considered), S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, and Z_{ij} is the treatment indicator for subject j in trial i. μ_{Ti} and β_i are the trial-specific intercepts and treatment-effects on the true endpoint in trial i. γ_{0i} and γ_{1i} are the trial-specific intercepts and treatment-effects on the true endpoint in trial i after accounting for the effect of the surrogate endpoint.

The -2 log likelihood values of the previous models in each of the i trials (i.e., L_{1i} and L_{2i}, respectively) are subsequently used to compute individual-level surrogacy based on the so-called Variance Reduction Factor (VFR; for details, see Alonso & Molenberghs, 2007):

$$R^2_{h.ind} = 1 - \frac{1}{N} \sum_i exp \left(-\frac{L_{2i} - L_{1i}}{n_i} \right),$$

where N is the number of trials and n_i is the number of patients within trial i.

When it can be assumed (i) that the treatment-corrected association between the surrogate and the true endpoint is constant across trials, or (ii) when all data come from a single clinical trial (i.e., when $N = 1$), the previous expression simplifies to:

$$R^2_{h.ind.clust} = 1 - exp \left(-\frac{L_2 - L_1}{N} \right).$$

Trial-level surrogacy

When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), trial-level surrogacy is assessed by fitting the following univariate models:

$$S_{ij} = \mu_{Si} + \alpha_i Z_{ij} + \varepsilon_{Sij}, (1)$$
$$T_{ij} = \mu_{Ti} + \beta_i Z_{ij} + \varepsilon_{Tij}, (1)$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_{Si} and μ_{Ti} are the fixed trial-specific intercepts for S and T, and α_i and β_i are the fixed trial-specific treatment effects on S and T, respectively. The error terms ε_{Sij} and ε_{Tij} are assumed to be independent.

When a reduced model is requested by the user (by using the argument `Model=c("Reduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_S + \alpha_i Z_{ij} + \varepsilon_{Sij}, (2)$$

$$T_{ij} = \mu_T + \beta_i Z_{ij} + \varepsilon_{Tij}, (2)$$

where μ_S and μ_T are the common intercepts for S and T. The other parameters are the same as defined above, and ε_{Sij} and ε_{Tij} are again assumed to be independent.

When the user requested a full model approach (by using the argument `Model=c("Full")` in the function call, i.e., when models (1) were fitted), the following model is subsequently fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\mu_{Si}} + \lambda_2 \widehat{\alpha}_i + \varepsilon_i, (3)$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on models (1) (see above). When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), model (3) is a weighted regression model (with weights based on the number of observations in trial i). The -2 log likelihood value of the (weighted or unweighted) model (3) (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based based on the Variance Reduction Factor (for details, see Alonso & Molenberghs, 2007):

$$R^2_{ht} = 1 - exp\left(-\frac{L_1 - L_0}{N}\right),$$

where N is the number of trials.

When a semi-reduced or reduced model is requested (by using the argument `Model=c("SemiReduced")` or `Model=c("Reduced")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameter estimates for β_i and α_i are based on models (1) when a semi-reduced model is fitted or on models (2) when a reduced model is fitted. The -2 log likelihood value of this (weighted or unweighted) model (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R^2_{ht} is computed based on the reduction in the likelihood (as described above).

Value

An object of class `FixedContContIT` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Trial.Spec.Results` A `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`R2ht` A `data.frame` that contains the trial-level surrogacy estimate and its confidence interval.

`R2h.ind.clust` A `data.frame` that contains the individual-level surrogacy estimate and its confidence interval.

`R2h.ind` A `data.frame` that contains the individual-level surrogacy estimate and its confidence interval under the assumption that the treatment-corrected association between the surrogate and the true endpoints is constant across trials or when all data come from a single clinical trial.

`Boot.CI` A `data.frame` that contains the bootstrapped `R2h.Single` values.

`Cor.Endpoints` A `data.frame` that contains the correlations between the surrogate and the true endpoint in the control treatment group (i.e., ρ_{T0S0}) and in the experimental treatment group (i.e., ρ_{T1S1}), their standard errors and their confidence intervals.

`Residuals` A `data.frame` that contains the residuals for the surrogate and true endpoints (ε_{Sij} and ε_{Tij}) that are obtained when models (1) or models (2) are fitted (see the **Details** section above).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

See Also

`MixedContContIT`, `FixedContBinIT`, `FixedBinContIT`, `FixedBinBinIT`, `plot Information-Theoretic`

Examples

```
# Example 1
# Based on the ARMD data
```

```
data(ARMD)
# Assess surrogacy based on a full fixed-effect model
# in the information-theoretic framework:
Sur <- FixedContContIT(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Trial.ID=Center,
Pat.ID=Id, Model="Full", Number.Bootstraps=50)
# Obtain a summary of the results:
summary(Sur)

## Not run:  #time consuming code
# Example 2
# Conduct an analysis based on a simulated dataset with 2000 patients, 100 trials,
# and Rindiv=Rtrial=.8

# Simulate the data:
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8, R.Indiv.Target=.8,
             Seed=123, Model="Full")
# Assess surrogacy based on a full fixed-effect model
# in the information-theoretic framework:
Sur2 <- FixedContContIT(Dataset=Data.Observed.MTS, Surr=Surr, True=True, Treat=Treat,
Trial.ID=Trial.ID, Pat.ID=Pat.ID, Model="Full", Number.Bootstraps=50)

# Show a summary of the results:
summary(Sur2)
## End(Not run)
```

FixedDiscrDiscrIT *Investigates surrogacy for binary or ordinal outcomes using the Information Theoretic framework*

Description

The function FixedDiscrDiscrIT uses the information theoretic approach (Alonso and Molenberghs 2007) to estimate trial and individual level surrogacy based on fixed-effects models when the surrogate is binary and the true outcome is ordinal, the converse case or when both outcomes are ordinal (the user must specify which form the data is in). The user can specify whether a weighted or unweighted analysis is required at the trial level. The penalized likelihood approach of Firth (1993) is applied to resolve issues of separation in discrete outcomes for particular trials. Requires packages OrdinalLogisticBiplot and logistf.

Usage

```
FixedDiscrDiscrIT(Dataset, Surr, True, Treat, Trial.ID,
Weighted = TRUE, Setting = c("binord"))
```

Arguments

Dataset A data.frame that should consist of one line per patient. Each line contains (at least) a surrogate value, a true outcome value, a treatment indicator and a trial ID.

`Surr` The name of the variable in `Dataset` that contains the surrogate outcome values.

`True` The name of the variable in `Dataset` that contains the true outcome values.

`Treat` The name of the in `Dataset` that contains the treatment group values, 0/1 or -1/+1 are recommended.

`Trial.ID` The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.

`Weighted` Logical. In practice it is often the case that different trials (or other clustering units) have different sample sizes. Univariate models are used to assess surrogacy in the information-theoretic approach, so it can be useful to adjust for heterogeneity in information content between the trial-specific contributions (particularly when trial-level surrogacy measures are of primary interest and when the heterogeneity in sample sizes is large). If `Weighted=TRUE`, weighted regression models are fitted. If `Weighted=FALSE`, unweighted regression analyses are conducted. See the **Details** section below. Default `TRUE`.

`Setting` Specifies whether an ordinal or binary surrogate or true outcome are present in `Dataset`. `Setting=c("binord")` for a binary surrogate and ordinal true outcome, `Setting=c("ordbin")` for an ordinal surrogate and binary true outcome and `Setting=c("ordord")` where both outcomes are ordinal.

Details

Individual level surrogacy

The following univariate logistic regression models are fitted when `Setting=c("ordbin")`:

$$logit(P(T_{ij} = 1)) = \mu_{Ti} + \beta_i Z_{ij}, (1)$$

$$logit(P(T_{ij} = 1 | S_{ij} = s)) = \gamma_{0i} + \gamma_{1i} Z_{ij} + \gamma_{2i} S_{ij}, (1)$$

where: i and j are the trial and subject indicators; S_{ij} and T_{ij} are the surrogate and true outcome values of subject j in trial i; and Z_{ij} is the treatment indicator for subject j in trial i; μ_{Ti} and β_i are the trial-specific intercepts and treatment-effects on the true endpoint in trial i; and γ_{0i} and γ_{1i} are the trial-specific intercepts and treatment-effects on the true endpoint in trial i after accounting for the effect of the surrogate endpoint. The -2 log likelihood values of the previous models in each of the i trials (i.e., L_{1i} and L_{2i}, respectively) are subsequently used to compute individual-level surrogacy based on the so-called Likelihood Reduction Factor (LRF; for details, see Alonso & Molenberghs, 2006):

$$R_h^2 = 1 - \frac{1}{N} \sum_i exp\left(- \frac{L_{2i} - L_{1i}}{n_i} \right),$$

where N is the number of trials and n_i is the number of patients within trial i.

At the individual level in the discrete case R_h^2 is bounded above by a number strictly less than one and is re-scaled (see Alonso & Molenberghs (2007)):

$$\widehat{R_h^2} = \frac{R_h^2}{1 - e^{-2L_0}},$$

where L_0 is the log-likelihood of the intercept only model of the true outcome ($logit(P(T_{ij} = 1) = \gamma_3$).

In the case of `Setting=c("binord")` or `Setting=c("ordord")` proportional odds models in (1) are used to accommodate the ordinal true response outcome, in all other respects the calculation of R_h^2 would proceed in the same manner.

Trial-level surrogacy

When `Setting=c("ordbin")` trial-level surrogacy is assessed by fitting the following univariate logistic regression and proportional odds models for the ordinal surrogate and binary true response variables regressed on treatment for each trial i:

$$logit(P(S_{ij} \leq W)) = \mu_{S_{wi}} + \alpha_i Z_{ij}, (2)$$

$$logit(P(T_{ij} = 1)) = \mu_{Ti} + \beta_i Z_{ij}, (2)$$

where: i and j are the trial and subject indicators; S_{ij} and T_{ij} are the surrogate and true outcome values of subject j in trial i; Z_{ij} is the treatment indicator for subject j in trial i; $\mu_{S_{wi}}$ are the trial-specific intercept values for each cut point w, where $w = 1, .., W - 1$, of the ordinal surrogate outcome; μ_{Ti} are the fixed trial-specific intercepts for T; and α_i and β_i are the fixed trial-specific treatment effects on S and T, respectively. The mean trial-specific intercepts for the surrogate are calculated, $\overline{\mu}_{S_{wi}}$.The following model is subsequently fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\overline{\mu}}_{S_{wi}} + \lambda_2 \widehat{\alpha}_i + \varepsilon_i, (3)$$

where the parameter estimates for β_i, $\overline{\mu}_{S_{wi}}$, and α_i are based on models (2) (see above). When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), model (2) is a weighted regression model (with weights based on the number of observations in trial i). The -2 log likelihood value of the (weighted or unweighted) model (2) (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta}_i = \lambda_3$; L_0), and R_{ht}^2 is computed based on the Likelihood Reduction Factor (for details, see Alonso & Molenberghs, 2006):

$$R_{ht}^2 = 1 - exp\left(-\frac{L_1 - L_0}{N}\right),$$

where N is the number of trials.

When separation (the presence of zero cells) occurs in the cross tabs of treatment and the true or surrogate outcome for a particular trial in models (2) extreme bias can occur in R_{ht}^2. Under separation there are no unique maximum likelihood for parameters β_i, $\overline{\mu}_{S_{wi}}$ and α_i, in (2), for the affected trial i. This typically leads to extreme bias in the estimation of these parameters and hence outlying influential points in model (3), bias in R_{ht}^2 inevitably follows.

To resolve the issue of separation the penalized likelihood approach of Firth (1993) is applied. This approach adds an asymptotically negligible component to the score function to allow unbiased estimation of β_i, $\overline{\mu}_{S_{wi}}$, and α_i and in turn R_{ht}^2. The penalized likelihood R function `logitf` from the package of the same name is applied in the case of binary separation (Heinze and Schemper, 2002). The function `pordlogistf` from the package `OrdinalLogisticBioplot` is applied in the case of ordinal separation (Hern'andez, 2013). All instances of separation are reported.

In the case of `Setting=c("binord")` or `Setting=c("ordord")` the appropriate models (either logistic regression or a proportional odds models) are fitted in (2) to accommodate the form (either binary or ordinal) of the true or surrogate response variable. The rest of the analysis would proceed in a similar manner as that described above.

Value

An object of class `FixedDiscrDiscrIT` with components,

`Trial.Spec.Results`
: A `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints. Also, the number of observations per trial; whether the trial was able to be included in the analysis for both R^2_h and R^2_{ht}; whether separation occurred and hence the penalized likelihood approach used for the surrogate or true outcome.

`R2ht`
: A `data.frame` that contains the trial-level surrogacy estimate and its confidence interval.

`R2h`
: A `data.frame` that contains the individual-level surrogacy estimate and its confidence interval.

Author(s)

Hannah M. Ensor & Christopher J. Weir

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

Alonso, A, & Molenberghs, G., Geys, H., Buyse, M. & Vangeneugden, T. (2006). A unifying approach for surrogate marker validation based on Prentice's criteria. *Statistics in medicine, 25,* 205-221.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika, 80,* 27-38.

Heinze, G. & Schemper, M. 2002. A solution to the problem of separation in logistic regression. *Statistics in medicine, 21,* 2409-2419.

Hern'andez, J. C. V.-V. O., J. L. 2013. OrdinalLogisticBiplot: Biplot representations of ordinal variables. R.

See Also

`FixedContContIT`, `plot Information-Theoretic`

Examples

```
## Not run:  # Time consuming (>5sec) code part
# Example 1
# Conduct an analysis based on a simulated dataset with 2000 patients, 100 trials,
# and Rindiv=Rtrial=.8

# Simulate the data:
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8, R.Indiv.Target=.8,
Seed=123, Model="Full")

# create a binary true and ordinal surrogate outcome
Data.Observed.MTS$True<-findInterval(Data.Observed.MTS$True,
c(quantile(Data.Observed.MTS$True,0.5)))
Data.Observed.MTS$Surr<-findInterval(Data.Observed.MTS$Surr,
c(quantile(Data.Observed.MTS$Surr,0.333),quantile(Data.Observed.MTS$Surr,0.666)))

# Assess surrogacy based on a full fixed-effect model
# in the information-theoretic framework for a binary surrogate and ordinal true outcome:
SurEval <- FixedDiscrDiscrIT(Dataset=Data.Observed.MTS, Surr=Surr, True=True, Treat=Treat,
Trial.ID=Trial.ID, Setting="ordbin")

# Show a summary of the results:
summary(SurEval)
SurEval$Trial.Spec.Results
SurEval$R2h
SurEval$R2ht

## End(Not run)
```

---

frank_loglik_copula_scale

Loglikelihood on the Copula Scale for the Frank Copula

Description

`frank_loglik_copula_scale()` computes the loglikelihood on the copula scale for the Frank copula which is parameterized by `theta` as follows:

$$C(u,v) = -\frac{1}{\theta}\log\left[1 - \frac{(1-e^{-\theta u})(1-e^{-\theta v})}{1-e^{-\theta}}\right]$$

Usage

```
frank_loglik_copula_scale(theta, u, v, d1, d2, return_sum = TRUE)
```

Arguments

`theta` Copula parameter

`u` A numeric vector. Corresponds to first variable on the copula scale.

`v` A numeric vector. Corresponds to second variable on the copula scale.

`d1` An integer vector. Indicates whether first variable is observed or right-censored,

- `d1[i] = 1` if `u[i]` corresponds to non-censored value
- `d1[i] = 0` if `u[i]` corresponds to right-censored value
- `d1[i] = -1` if `u[i]` corresponds to left-censored value

`d2` An integer vector. Indicates whether first variable is observed or right-censored,

- `d2[i] = 1` if `v[i]` corresponds to non-censored value
- `d2[i] = 0` if `v[i]` corresponds to right-censored value
- `d2[i] = -1` if `v[i]` corresponds to left-censored value

`return_sum` Return the sum of the individual loglikelihoods? If `FALSE`, a vector with the individual loglikelihood contributions is returned.

Value

Value of the copula loglikelihood evaluated in `theta`.

gaussian_loglik_copula_scale

Loglikelihood on the Copula Scale for the Gaussian Copula

Description

`gaussian_loglik_copula_scale()` computes the loglikelihood on the copula scale for the Gaussian copula which is parameterized by `theta` as follows:

$$C(u, v) = \Psi\left[\Phi^{-1}(u), \Phi^{-1}(v) | \rho\right]$$

Usage

```
gaussian_loglik_copula_scale(theta, u, v, d1, d2, return_sum = TRUE)
```

Arguments

`theta` Copula parameter

`u` A numeric vector. Corresponds to first variable on the copula scale.

`v` A numeric vector. Corresponds to second variable on the copula scale.

`d1` An integer vector. Indicates whether first variable is observed or right-censored,

- `d1[i] = 1` if `u[i]` corresponds to non-censored value
- `d1[i] = 0` if `u[i]` corresponds to right-censored value
- `d1[i] = -1` if `u[i]` corresponds to left-censored value

`d2` An integer vector. Indicates whether first variable is observed or right-censored,

- `d2[i] = 1` if `v[i]` corresponds to non-censored value
- `d2[i] = 0` if `v[i]` corresponds to right-censored value
- `d2[i] = -1` if `v[i]` corresponds to left-censored value

`return_sum` Return the sum of the individual loglikelihoods? If `FALSE`, a vector with the individual loglikelihood contributions is returned.

Value

Value of the copula loglikelihood evaluated in `theta`.

gumbel_loglik_copula_scale *Loglikelihood on the Copula Scale for the Gumbel Copula*

Description

`gumbel_loglik_copula_scale()` computes the loglikelihood on the copula scale for the Gumbel copula which is parameterized by `theta` as follows:

$$C(u, v) = \exp\left[-\left\{(-\log u)^\theta + (-\log v)^\theta\right\}^{\frac{1}{\theta}}\right]$$

Usage

```
gumbel_loglik_copula_scale(theta, u, v, d1, d2, return_sum = TRUE)
```

Arguments

`theta`	Copula parameter
`u`	A numeric vector. Corresponds to first variable on the copula scale.
`v`	A numeric vector. Corresponds to second variable on the copula scale.
`d1`	An integer vector. Indicates whether first variable is observed or right-censored, • `d1[i] = 1` if `u[i]` corresponds to non-censored value • `d1[i] = 0` if `u[i]` corresponds to right-censored value • `d1[i] = -1` if `u[i]` corresponds to left-censored value
`d2`	An integer vector. Indicates whether first variable is observed or right-censored, • `d2[i] = 1` if `v[i]` corresponds to non-censored value • `d2[i] = 0` if `v[i]` corresponds to right-censored value • `d2[i] = -1` if `v[i]` corresponds to left-censored value
`return_sum`	Return the sum of the individual loglikelihoods? If `FALSE`, a vector with the individual loglikelihood contributions is returned.

Value

Value of the copula loglikelihood evaluated in `theta`.

ICA.BinBin *Assess surrogacy in the causal-inference single-trial setting in the binary-binary case*

Description

The function ICA.BinBin quantifies surrogacy in the single-trial causal-inference framework (individual causal association and causal concordance) when both the surrogate and the true endpoints are binary outcomes. See **Details** below.

Usage

```
ICA.BinBin(pi1_1_, pi1_0_, pi_1_1, pi_1_0, pi0_1_, pi_0_1,
Monotonicity=c("General"), Sum_Pi_f = seq(from=0.01, to=0.99, by=.01),
M=10000, Volume.Perc=0, Seed=sample(1:100000, size=1))
```

Arguments

pi1_1_ A scalar or vector that contains values for $P(T = 1, S = 1|Z = 0)$, i.e., the probability that $S = T = 1$ when under treatment $Z = 0$. A vector is specified to account for uncertainty, i.e., rather than keeping $P(T = 1, S = 1|Z = 0)$ fixed at one estimated value, a distribution can be specified (see **examples** below) from which a value is drawn in each run.

pi1_0_ A scalar or vector that contains values for $P(T = 1, S = 0|Z = 0)$.

pi_1_1 A scalar or vector that contains values for $P(T = 1, S = 1|Z = 1)$.

pi_1_0 A scalar or vector that contains values for $P(T = 1, S = 0|Z = 1)$.

pi0_1_ A scalar or vector that contains values for $P(T = 0, S = 1|Z = 0)$.

pi_0_1 A scalar or vector that contains values for $P(T = 0, S = 1|Z = 1)$.

Monotonicity Specifies which assumptions regarding monotonicity should be made: Monotonicity=c("General"), Monotonicity=c("No"), Monotonicity=c("True.Endp"), Monotonicity=c("Surr.Endp"), or Monotonicity=c("Surr.True.Endp"). See **Details** below. Default Monotonicity=c("General").

Sum_Pi_f A scalar or vector that specifies the grid of values $G = g_1,\ g_2,\ ...,\ g_k$ to be considered when the sensitivity analysis is conducted. See **Details** below. Default Sum_Pi_f = seq(from=0.01, to=0.99, by=.01).

M The number of runs that are conducted for a given value of Sum_Pi_f. This argument is not used when Volume.Perc=0. Default M=10000.

Volume.Perc Note that the marginals that are observable in the data set a number of restrictions on the unidentified correlations. For example, under montonicity for S and T, it holds that $\pi_{0111} <= min(\pi_{0\cdot1\cdot}, \pi_{\cdot1\cdot1})$ and $\pi_{1100} <= min(\pi_{1\cdot0\cdot}, \pi_{\cdot1\cdot0})$. For example, when $min(\pi_{0\cdot1\cdot}, \pi_{\cdot1\cdot1}) = 0.10$ and $min(\pi_{1\cdot0\cdot}, \pi_{\cdot1\cdot0}) = 0.08$, then all valid $\pi_{0111} <= 0.10$ and all valid $\pi_{1100} <= 0.08$. The argument Volume.Perc specifies the fraction of the 'volume' of the paramater space that is explored. This volume is computed based on the grids G={0, 0.01, ..., maximum possible value for the counterfactual probability at hand}. E.g., in the previous example, the 'volume' of the parameter space would be $11 * 9 = 99$, and when e.g., the argument `Volume.Perc=1` is used a total of 99 runs will be conducted for each given value of `Sum_Pi_f`. Notice that when monotonicity is not assumed, relatively high values of `Volume.Perc` will lead to a large number of runs and consequently a long analysis time.

`Seed` The seed to be used to generate π_r. Default `Seed=sample(1:100000, size=1)`.

Details

In the continuous normal setting, surroagacy can be assessed by studying the association between the individual causal effects on S and T (see `ICA.ContCont`). In that setting, the Pearson correlation is the obvious measure of association.

When S and T are binary endpoints, multiple alternatives exist. Alonso et al. (2014) proposed the individual causal association (ICA; R_H^2), which captures the association between the individual causal effects of the treatment on S (Δ_S) and T (Δ_T) using information-theoretic principles.

The function `ICA.BinBin` computes R_H^2 based on plausible values of the potential outcomes. Denote by $\boldsymbol{Y}' = (T_0, T_1, S_0, S_1)$ the vector of potential outcomes. The vector $\boldsymbol{Y}$ can take 16 values and the set of parameters $\pi_{ijpq} = P(T_0 = i, T_1 = j, S_0 = p, S_1 = q)$ (with $i, j, p, q = 0/1$) fully characterizes its distribution.

However, the parameters in π_{ijpq} are not all functionally independent, e.g., $1 = \pi_{....}$. When no assumptions regarding monotonicity are made, the data impose a total of 7 restrictions, and thus only 9 proabilities in π_{ijpq} are allowed to vary freely (for details, see Alonso et al., 2014). Based on the data and assuming SUTVA, the marginal probabilites $\pi_{1 \cdot 1 \cdot}$, $\pi_{1 \cdot 0 \cdot}$, $\pi_{\cdot 1 \cdot 1}$, $\pi_{\cdot 1 \cdot 0}$, $\pi_{0 \cdot 1 \cdot}$, and $\pi_{\cdot 0 \cdot 1}$ can be computed (by hand or using the function `MarginalProbs`). Define the vector

$$\boldsymbol{b}' = (1, \pi_{1 \cdot 1 \cdot}, \pi_{1 \cdot 0 \cdot}, \pi_{\cdot 1 \cdot 1}, \pi_{\cdot 1 \cdot 0}, \pi_{0 \cdot 1 \cdot}, \pi_{\cdot 0 \cdot 1})$$

and $\boldsymbol{A}$ is a contrast matrix such that the identified restrictions can be written as a system of linear equation

$$\boldsymbol{A\pi} = \boldsymbol{b}.$$

The matrix $\boldsymbol{A}$ has rank 7 and can be partitioned as $\boldsymbol{A} = (\boldsymbol{A_r}|\boldsymbol{A_f})$, and similarly the vector $\boldsymbol{\pi}$ can be partitioned as $\boldsymbol{\pi}^{'} = (\boldsymbol{\pi}_{\boldsymbol{r}}^{'}|\boldsymbol{\pi}_{\boldsymbol{f}}^{'})$ (where f refers to the submatrix/vector given by the 9 last columns/components of $\boldsymbol{A}/\boldsymbol{\pi}$). Using these partitions the previous system of linear equations can be rewritten as

$$\boldsymbol{A_r \pi_r} + \boldsymbol{A_f \pi_f} = \boldsymbol{b}.$$

The following algorithm is used to generate plausible distributions for $\boldsymbol{Y}$. First, select a value of the specified grid of values (specified using `Sum_Pi_f` in the function call). For $k = 1$ to M (specified using `M` in the function call), generate a vector π_f that contains 9 components that are uniformly sampled from hyperplane subject to the restriction that the sum of the generated components equals `Sum_Pi_f` (the function `RandVec`, which uses the `randfixedsum` algorithm written by Roger Stafford, is used to obtain these components). Next, $\boldsymbol{\pi_r} = \boldsymbol{A_r^{-1}}(\boldsymbol{b} - \boldsymbol{A_f \pi_f})$ is computed and the π_r vectors where all components are in the $[0;\ 1]$ range are retained. This procedure is repeated for each of the `Sum_Pi_f` values. Based on these results, R_H^2 is estimated. The obtained values can be used to conduct a sensitivity analysis during the validation exercise.

The previous developments hold when no monotonicity is assumed. When monotonicity for S, T, or for S and T is assumed, some of the probabilities of π are zero. For example, when montonicity is assumed for T, then $P(T_0 <= T_1) = 1$, or equivantly, $\pi_{1000} = \pi_{1010} = \pi_{1001} = \pi_{1011} = 0$. When monotonicity is assumed, the procedure described above is modified accordingly (for details, see Alonso et al., 2014). When a general analysis is requested (using `Monotonicity=c("General")` in the function call), all settings are considered (no monotonicity, monotonicity for S alone, for T alone, and for both for S and T.)

To account for the uncertainty in the estimation of the marginal probabilities, a vector of values can be specified from which a random draw is made in each run (see **Examples** below).

Value

An object of class `ICA.BinBin` with components,

`Pi.Vectors` An object of class `data.frame` that contains the valid π vectors.

`R2_H` The vector of the R^2_H values.

`Theta_T` The vector of odds ratios for T.

`Theta_S` The vector of odds ratios for S.

`H_Delta_T` The vector of the entropies of Δ_T.

`Monotonicity` The assumption regarding monotonicity that was made.

`Volume.No` The 'volume' of the parameter space when monotonicity is not assumed. Is only provided when the argument $Volume.Perc$ is used (i.e., when it is not equal to 0.

`Volume.T` The 'volume' of the parameter space when monotonicity for T is assumed. Is only provided when the argument $Volume.Perc$ is used.

`Volume.S` The 'volume' of the parameter space when monotonicity for S is assumed. Is only provided when the argument $Volume.Perc$ is used.

`Volume.ST` The 'volume' of the parameter space when monotonicity for S and T is assumed. Is only provided when the argument $Volume.Perc$ is used.

Author(s)

Wim Van der Elst, Paul Meyvisch, Ariel Alonso & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2015). Validation of surrogate endpoints: the binary-binary setting from a causal inference perspective.

See Also

`ICA.ContCont`, `MICA.ContCont`

Examples

```
## Not run: # Time consuming code part
# Compute R2_H given the marginals specified as the pi's, making no
# assumptions regarding monotonicity (general case)
ICA <- ICA.BinBin(pi1_1_=0.2619048, pi1_0_=0.2857143, pi_1_1=0.6372549,
pi_1_0=0.07843137, pi0_1_=0.1349206, pi_0_1=0.127451, Seed=1,
Monotonicity=c("General"), Sum_Pi_f = seq(from=0.01, to=.99, by=.01), M=10000)

# obtain plot of the results
plot(ICA, R2_H=TRUE)

# Example 2 where the uncertainty in the estimation
# of the marginals is taken into account
ICA_BINBIN2 <- ICA.BinBin(pi1_1_=runif(10000, 0.2573, 0.4252),
pi1_0_=runif(10000, 0.1769, 0.3310),
pi_1_1=runif(10000, 0.5947, 0.7779),
pi_1_0=runif(10000, 0.0322, 0.1442),
pi0_1_=runif(10000, 0.0617, 0.1764),
pi_0_1=runif(10000, 0.0254, 0.1315),
Monotonicity=c("General"),
Sum_Pi_f = seq(from=0.01, to=0.99, by=.01),
M=50000, Seed=1)

# Plot results
plot(ICA_BINBIN2)

## End(Not run)
```

ICA.BinBin.CounterAssum

ICA (binary-binary setting) that is obtaied when the counterfactual correlations are assumed to fall within some prespecified ranges.

Description

Shows the results of ICA (binary-binary setting) in the subgroup of results where the counterfactual correlations are assumed to fall within some prespecified ranges.

Usage

```
ICA.BinBin.CounterAssum(x, r2_h_S0S1_min, r2_h_S0S1_max, r2_h_S0T1_min,
r2_h_S0T1_max, r2_h_T0T1_min, r2_h_T0T1_max, r2_h_T0S1_min, r2_h_T0S1_max,
Monotonicity="General", Type="Freq", MainPlot=" ", Cex.Legend=1,
Cex.Position="topright", ...)
```

Arguments

x	An object of class ICA.BinBin. See ICA.BinBin.
r2_h_S0S1_min	The minimum value to be considered for the counterfactual correlation $r_h^2(S_0, S_1)$.
r2_h_S0S1_max	The maximum value to be considered for the counterfactual correlation $r_h^2(S_0, S_1)$.
r2_h_S0T1_min	The minimum value to be considered for the counterfactual correlation $r_h^2(S_0, T_1)$.
r2_h_S0T1_max	The maximum value to be considered for the counterfactual correlation $r_h^2(S_0, T_1)$.

`r2_h_T0T1_min` The minimum value to be considered for the counterfactual correlation $r_h^2(T_0, T_1)$.

`r2_h_T0T1_max` The maximum value to be considered for the counterfactual correlation $r_h^2(T_0, T_1)$.

`r2_h_T0S1_min` The minimum value to be considered for the counterfactual correlation $r_h^2(T_0, S_1)$.

`r2_h_T0S1_max` The maximum value to be considered for the counterfactual correlation $r_h^2(T_0, S_1)$.

`Monotonicity` Specifies whether the all results in the fitted object `ICA.BinBin` should be shown (i.e., `Monotonicity=c("General")`), or a subset of the results arising under specific assumptions (i.e., `Monotonicity=c("No")`, `Monotonicity=c("True.Endp")`, `Monotonicity=c("Surr.Endp")`, or `Monotonicity=c("Surr.True.Endp")`). Default `Monotonicity=c("General")`.

`Type` The type of plot that is produced. When `Type="Freq"` or `Type="Density"`, the Y-axis shows frequencies or densities of R_H^2. When `Type="All.Densities"` and the fitted object of class `ICA.BinBin` was obtained using a general analysis (i.e., conducting the analyses assuming no monotonicity, monotonicity for S alone, monotonicity for T alone, and for both S and T, so using `Monotonicity=c("General")` in the function call of `ICA.BinBin`), the density plots are shown for the four scenarios where different assumptions regarding monotonicity are made. Default `"Freq"`.

`MainPlot` The title of the plot. Default `" "`.

`Cex.Legend` The size of the legend when `Type="All.Densities"` is used. Default `Cex.Legend=1`.

`Cex.Position` The position of the legend, `Cex.Position="topright"` or `Cex.Position="topleft"`. Default `Cex.Position="topright"`.

`...` Other arguments to be passed to the `plot()` function.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal inference and meta-analytic paradigms for the validation of surrogate markers.

Van der Elst, W., Alonso, A., & Molenberghs, G. (submitted). An exploration of the relationship between causal inference and meta-analytic measures of surrogacy.

See Also

ICA.BinBin

Examples

```
## Not run:  #Time consuming (>5 sec) code part
# Compute R2_H given the marginals specified as the pi's, making no
# assumptions regarding monotonicity (general case)
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.261, pi1_0_=0.285,
pi_1_1=0.637, pi_1_0=0.078, pi0_1_=0.134, pi_0_1=0.127,
Monotonicity=c("General"), M=5000, Seed=1)
```

```
# Obtain a density plot of R2_H, assuming that
# r2_h_S0S1>=.2, r2_h_S0T1>=0, r2_h_T0T1>=.2, and r2_h_T0S1>=0
ICA.BinBin.CounterAssum(ICA, r2_h_S0S1_min=.2, r2_h_S0S1_max=1,
r2_h_S0T1_min=0, r2_h_S0T1_max=1, r2_h_T0T1_min=0.2, r2_h_T0T1_max=1,
r2_h_T0S1_min=0, r2_h_T0S1_max=1, Monotonicity=”General”,
Type=”Density”)

# Now show the densities of R2_H under the different
# monotonicity assumptions
ICA.BinBin.CounterAssum(ICA, r2_h_S0S1_min=.2, r2_h_S0S1_max=1,
r2_h_S0T1_min=0, r2_h_S0T1_max=1, r2_h_T0T1_min=0.2, r2_h_T0T1_max=1,
r2_h_T0S1_min=0, r2_h_T0S1_max=1, Monotonicity=”General”,
Type=”All.Densities”, MainPlot=” ”, Cex.Legend=1,
Cex.Position=”topright”, ylim=c(0, 20))

## End(Not run)
```

ICA.BinBin.Grid.Full *Assess surrogacy in the causal-inference single-trial setting in the binary-binary case when monotonicity for S and T is assumed using the full grid-based approach*

Description

The function `ICA.BinBin.Grid.Full` quantifies surrogacy in the single-trial causal-inference framework (individual causal association and causal concordance) when both the surrogate and the true endpoints are binary outcomes. This method provides an alternative for `ICA.BinBin` and `ICA.BinBin.Grid.Sample`. It uses an alternative strategy to identify plausible values for π. See **Details** below.

Usage

```
ICA.BinBin.Grid.Full(pi1_1_, pi1_0_, pi_1_1, pi_1_0, pi0_1_, pi_0_1,
Monotonicity=c(”General”), pi_1001=seq(0, 1, by=.02),
pi_1110=seq(0, 1, by=.02), pi_1101=seq(0, 1, by=.02),
pi_1011=seq(0, 1, by=.02), pi_1111=seq(0, 1, by=.02),
pi_0110=seq(0, 1, by=.02), pi_0011=seq(0, 1, by=.02),
pi_0111=seq(0, 1, by=.02), pi_1100=seq(0, 1, by=.02),
Seed=sample(1:100000, size=1))
```

Arguments

`pi1_1_` A scalar that contains $P(T = 1, S = 1|Z = 0)$, i.e., the proability that $S = T = 1$ when under treatment $Z = 0$.

`pi1_0_` A scalar that contains $P(T = 1, S = 0|Z = 0)$.

`pi_1_1` A scalar that contains $P(T = 1, S = 1|Z = 1)$.

`pi_1_0` A scalar that contains $P(T = 1, S = 0|Z = 1)$.

`pi0_1_` A scalar that contains $P(T = 0, S = 1|Z = 0)$.

`pi_0_1` A scalar that contains $P(T = 0, S = 1|Z = 1)$.

`Monotonicity` Specifies which assumptions regarding monotonicity should be made: `Monotonicity=c("General")`, `Monotonicity=c("No")`, `Monotonicity=c("True.Endp")`, `Monotonicity=c("Surr.Endp")`, or `Monotonicity=c("Surr.True.Endp")`. When a general analysis is requested (using `Monotonicity=c("General")` in the function call), all settings are considered (no monotonicity, monotonicity for S alone, for T alone, and for both for S and T. Default `Monotonicity=c("General")`.

`pi_1001` A vector that specifies the grid of values that should be considered for π_{pi_1001}. Default `pi_1001=seq(0, 1, by=.02)`.

`pi_1110` A vector that specifies the grid of values that should be considered for π_{pi_1110}. Default `pi_1110=seq(0, 1, by=.02)`.

`pi_1101` A vector that specifies the grid of values that should be considered for π_{pi_1101}. Default `pi_1101=seq(0, 1, by=.02)`.

`pi_1011` A vector that specifies the grid of values that should be considered for π_{pi_1011}. Default `pi_1011=seq(0, 1, by=.02)`.

`pi_1111` A vector that specifies the grid of values that should be considered for π_{pi_1111}. Default `pi_1111=seq(0, 1, by=.02)`.

`pi_0110` A vector that specifies the grid of values that should be considered for π_{pi_0110}. Default `pi_0110=seq(0, 1, by=.02)`.

`pi_0011` A vector that specifies the grid of values that should be considered for π_{pi_0011}. Default `pi_0011=seq(0, 1, by=.02)`.

`pi_0111` A vector that specifies the grid of values that should be considered for π_{pi_0111}. Default `pi_0111=seq(0, 1, by=.02)`.

`pi_1100` A vector that specifies the grid of values that should be considered for π_{pi_1100}. Default `pi_1100=seq(0, 1, by=.02)`.

`Seed` The seed to be used to generate π_r. Default `Seed=sample(1:100000, size=1)`.

Details

In the continuous normal setting, surroagacy can be assessed by studying the association between the individual causal effects on S and T (see `ICA.ContCont`). In that setting, the Pearson correlation is the obvious measure of association.

When S and T are binary endpoints, multiple alternatives exist. Alonso et al. (2014) proposed the individual causal association (ICA; R^2_H), which captures the association between the individual causal effects of the treatment on S (Δ_S) and T (Δ_T) using information-theoretic principles.

The function `ICA.BinBin.Grid.Full` computes R^2_H using a grid-based approach where all possible combinations of the specified grids for the parameters that are allowed that are allowed to vary freely are considered. When it is not assumed that monotonicity holds for both S and T, the computationally less demanding algorithm `ICA.BinBin.Grid.Sample` may be preferred.

Value

An object of class `ICA.BinBin` with components, `Pi.Vectors` An object of class `data.frame` that contains the valid π vectors.

`R2_H` The vector of the R_H^2 values.

`Theta_T` The vector of odds ratios for T.

`Theta_S` The vector of odds ratios for S.

`H_Delta_T` The vector of the entropies of Δ_T.

Author(s)

Wim Van der Elst, Paul Meyvisch, Ariel Alonso & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2014). Validation of surrogate endpoints: the binary-binary setting from a causal inference perspective.

Buyse, M., Burzykowski, T., Aloso, A., & Molenberghs, G. (2014). Direct estimation of joint counterfactual probabilities, with application to surrogate marker validation.

See Also

`ICA.ContCont`, `MICA.ContCont`, `ICA.BinBin`, `ICA.BinBin.Grid.Sample`

Examples

```
## Not run:  # time consuming code part
# Compute R2_H given the marginals,
# assuming monotonicity for S and T and grids
# pi_0111=seq(0, 1, by=.001) and
# pi_1100=seq(0, 1, by=.001)
ICA <- ICA.BinBin.Grid.Full(pi1_1_=0.2619048, pi1_0_=0.2857143, pi_1_1=0.6372549,
pi_1_0=0.07843137, pi0_1_=0.1349206, pi_0_1=0.127451,
pi_0111=seq(0, 1, by=.01), pi_1100=seq(0, 1, by=.01), Seed=1)

# obtain plot of R2_H
plot(ICA, R2_H=TRUE)

## End(Not run)
```

`ICA.BinBin.Grid.Sample`

Assess surrogacy in the causal-inference single-trial setting in the binary-binary case when monotonicity for S and T is assumed using the grid-based sample approach

Description

The function `ICA.BinBin.Grid.Sample` quantifies surrogacy in the single-trial causal-inference framework (individual causal association and causal concordance) when both the surrogate and the true endpoints are binary outcomes. This method provides an alternative for `ICA.BinBin` and `ICA.BinBin.Grid.Full`. It uses an alternative strategy to identify plausible values for π. See **Details** below.

Usage

```
ICA.BinBin.Grid.Sample(pi1_1_, pi1_0_, pi_1_1, pi_1_0, pi0_1_,
pi_0_1, Monotonicity=c("General"), M=100000,
Volume.Perc=0, Seed=sample(1:100000, size=1))
```

Arguments

`pi1_1_` A scalar that contains values for $P(T = 1, S = 1|Z = 0)$, i.e., the probability that $S = T = 1$ when under treatment $Z = 0$.

`pi1_0_` A scalar that contains values for $P(T = 1, S = 0|Z = 0)$.

`pi_1_1` A scalar that contains values for $P(T = 1, S = 1|Z = 1)$.

`pi_1_0` A scalar that contains values for $P(T = 1, S = 0|Z = 1)$.

`pi0_1_` A scalar that contains values for $P(T = 0, S = 1|Z = 0)$.

`pi_0_1` A scalar that contains values for $P(T = 0, S = 1|Z = 1)$.

`Monotonicity` Specifies which assumptions regarding monotonicity should be made: `Monotonicity=c("General")`, `Monotonicity=c("No")`, `Monotonicity=c("True.Endp")`, `Monotonicity=c("Surr.Endp")`, or `Monotonicity=c("Surr.True.Endp")`. When a general analysis is requested (using `Monotonicity=c("General")` in the function call), all settings are considered (no monotonicity, monotonicity for S alone, for T alone, and for both for S and T. Default `Monotonicity=c("General")`.

`M` The number of random samples that have to be drawn for the freely varying parameters. Default `M=100000`. This argument is not used when `Volume.Perc=0`. Default `M=10000`.

`Volume.Perc` Note that the marginals that are observable in the data set a number of restrictions on the unidentified correlations. For example, under montonicity for S and T, it holds that $\pi_{0111} <= min(\pi_{0 \cdot 1 \cdot}, \pi_{\cdot 1 \cdot 1})$ and $\pi_{1100} <= min(\pi_{1 \cdot 0 \cdot}, \pi_{\cdot 1 \cdot 0})$. For example, when $min(\pi_{0 \cdot 1 \cdot}, \pi_{\cdot 1 \cdot 1}) = 0.10$ and $min(\pi_{1 \cdot 0 \cdot}, \pi_{\cdot 1 \cdot 0}) = 0.08$, then all valid $\pi_{0111} <= 0.10$ and all valid $\pi_{1100} <= 0.08$. The argument `Volume.Perc` specifies the fraction of the 'volume' of the paramater space that is explored. This volume is computed based on the grids G={0, 0.01, ..., maximum possible value for the counterfactual probability at hand}. E.g., in the previous example, the 'volume' of the parameter space would be $11 * 9 = 99$, and when e.g., the argument `Volume.Perc=1` is used a total of 99 runs will be conducted. Notice that when monotonicity is not assumed, relatively high values of `Volume.Perc` will lead to a large number of runs and consequently a long analysis time.

`Seed` The seed to be used to generate π_r. Default `M=100000`.

Details

In the continuous normal setting, surroagacy can be assessed by studying the association between the individual causal effects on S and T (see `ICA.ContCont`). In that setting, the Pearson correlation is the obvious measure of association.

When S and T are binary endpoints, multiple alternatives exist. Alonso et al. (2014) proposed the individual causal association (ICA; R_H^2), which captures the association between the individual causal effects of the treatment on S (Δ_S) and T (Δ_T) using information-theoretic principles.

The function `ICA.BinBin.Grid.Full` computes R_H^2 using a grid-based approach where all possible combinations of the specified grids for the parameters that are allowed that are allowed to vary freely are considered. When it is not assumed that monotonicity holds for both S and T, the number of possible combinations become very high. The function `ICA.BinBin.Grid.Sample` considers a random sample of all possible combinations.

Value

An object of class `ICA.BinBin` with components,

`Pi.Vectors` An object of class `data.frame` that contains the valid π vectors.

`R2_H` The vector of the R_H^2 values.

`Theta_T` The vector of odds ratios for T.

`Theta_S` The vector of odds ratios for S.

`H_Delta_T` The vector of the entropies of Δ_T.

`Volume.No` The 'volume' of the parameter space when monotonicity is not assumed.

`Volume.T` The 'volume' of the parameter space when monotonicity for T is assumed.

`Volume.S` The 'volume' of the parameter space when monotonicity for S is assumed.

`Volume.ST` The 'volume' of the parameter space when monotonicity for S and T is assumed.

Author(s)

Wim Van der Elst, Paul Meyvisch, Ariel Alonso & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2014). Validation of surrogate endpoints: the binary-binary setting from a causal inference perspective.

Buyse, M., Burzykowski, T., Aloso, A., & Molenberghs, G. (2014). Direct estimation of joint counterfactual probabilities, with application to surrogate marker validation.

See Also

`ICA.ContCont`, `MICA.ContCont`, `ICA.BinBin`, `ICA.BinBin.Grid.Sample`

Examples

```
## Not run:  #time-consuming code parts
# Compute R2_H given the marginals,
# assuming monotonicity for S and T and grids
# pi_0111=seq(0, 1, by=.001) and
# pi_1100=seq(0, 1, by=.001)
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.261, pi1_0_=0.285,
pi_1_1=0.637, pi_1_0=0.078, pi0_1_=0.134, pi_0_1=0.127,
Monotonicity=c("Surr.True.Endp"), M=2500, Seed=1)

# obtain plot of R2_H
plot(ICA, R2_H=TRUE)

## End(Not run)
```

ICA.BinBin.Grid.Sample.Uncert

Assess surrogacy in the causal-inference single-trial setting in the binary-binary case when monotonicity for S and T is assumed using the grid-based sample approach, accounting for sampling variability in the marginal π.

Description

The function `ICA.BinBin.Grid.Sample.Uncert` quantifies surrogacy in the single-trial causal-inference framework (individual causal association and causal concordance) when both the surrogate and the true endpoints are binary outcomes. This method provides an alternative for `ICA.BinBin` and `ICA.BinBin.Grid.Full`. It uses an alternative strategy to identify plausible values for π. The function allows to account for sampling variability in the marginal π. See **Details** below.

Usage

```
ICA.BinBin.Grid.Sample.Uncert(pi1_1_, pi1_0_, pi_1_1, pi_1_0, pi0_1_,
pi_0_1, Monotonicity=c("General"), M=100000,
Volume.Perc=0, Seed=sample(1:100000, size=1))
```

Arguments

`pi1_1_` A vector that contains values for $P(T=1,S=1|Z=0)$, i.e., the probability that $S=T=1$ when under treatment $Z=0$. A vector is specified to account for uncertainty, i.e., rather than keeping $P(T=1,S=1|Z=0)$ fixed at one estimated value, a distribution can be specified (see **examples** below) from which a value is drawn in each run.

`pi1_0_` A vector that contains values for $P(T=1,S=0|Z=0)$.

`pi_1_1` A vector that contains values for $P(T=1,S=1|Z=1)$.

`pi_1_0` A vector that contains values for $P(T=1,S=0|Z=1)$.

`pi0_1_` A vector that contains values for $P(T = 0, S = 1|Z = 0)$.

`pi_0_1` A vector that contains values for $P(T = 0, S = 1|Z = 1)$.

`Monotonicity` Specifies which assumptions regarding monotonicity should be made: `Monotonicity=c("General")`, `Monotonicity=c("No")`, `Monotonicity=c("True.Endp")`, `Monotonicity=c("Surr.Endp")`, or `Monotonicity=c("Surr.True.Endp")`. When a general analysis is requested (using `Monotonicity=c("General")` in the function call), all settings are considered (no monotonicity, monotonicity for S alone, for T alone, and for both for S and T. Default `Monotonicity=c("General")`.

`M` The number of random samples that have to be drawn for the freely varying parameters. Default `M=100000`. This argument is not used when `Volume.Perc=0`. Default `M=10000`.

`Volume.Perc` Note that the marginals that are observable in the data set a number of restrictions on the unidentified correlations. For example, under montonicity for S and T, it holds that $\pi_{0111} <= min(\pi_{0\cdot 1\cdot}, \pi_{\cdot 1\cdot 1})$ and $\pi_{1100} <= min(\pi_{1\cdot 0\cdot}, \pi_{\cdot 1\cdot 0})$. For example, when $min(\pi_{0\cdot 1\cdot}, \pi_{\cdot 1\cdot 1}) = 0.10$ and $min(\pi_{1\cdot 0\cdot}, \pi_{\cdot 1\cdot 0}) = 0.08$, then all valid $\pi_{0111} <= 0.10$ and all valid $\pi_{1100} <= 0.08$. The argument `Volume.Perc` specifies the fraction of the 'volume' of the paramater space that is explored. This volume is computed based on the grids G={0, 0.01, ..., maximum possible value for the counterfactual probability at hand}. E.g., in the previous example, the 'volume' of the parameter space would be $11 * 9 = 99$, and when e.g., the argument `Volume.Perc=1` is used a total of 99 runs will be conducted. Notice that when monotonicity is not assumed, relatively high values of `Volume.Perc` will lead to a large number of runs and consequently a long analysis time.

`Seed` The seed to be used to generate π_r. Default `M=100000`.

Details

In the continuous normal setting, surroagacy can be assessed by studying the association between the individual causal effects on S and T (see `ICA.ContCont`). In that setting, the Pearson correlation is the obvious measure of association.

When S and T are binary endpoints, multiple alternatives exist. Alonso et al. (2014) proposed the individual causal association (ICA; R^2_H), which captures the association between the individual causal effects of the treatment on S (Δ_S) and T (Δ_T) using information-theoretic principles.

The function `ICA.BinBin.Grid.Full` computes R^2_H using a grid-based approach where all possible combinations of the specified grids for the parameters that are allowed that are allowed to vary freely are considered. When it is not assumed that monotonicity holds for both S and T, the number of possible combinations become very high. The function `ICA.BinBin.Grid.Sample.Uncert` considers a random sample of all possible combinations.

Value

An object of class `ICA.BinBin` with components,

`Pi.Vectors` An object of class `data.frame` that contains the valid π vectors.

`R2_H` The vector of the R^2_H values.

`Theta_T` The vector of odds ratios for T.

Theta_S	The vector of odds ratios for S.
H_Delta_T	The vector of the entropies of Δ_T.
Volume.No	The 'volume' of the parameter space when monotonicity is not assumed.
Volume.T	The 'volume' of the parameter space when monotonicity for T is assumed.
Volume.S	The 'volume' of the parameter space when monotonicity for S is assumed.
Volume.ST	The 'volume' of the parameter space when monotonicity for S and T is assumed.

Author(s)

Wim Van der Elst, Paul Meyvisch, Ariel Alonso & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2014). Validation of surrogate endpoints: the binary-binary setting from a causal inference perspective.

Buyse, M., Burzykowski, T., Aloso, A., & Molenberghs, G. (2014). Direct estimation of joint counterfactual probabilities, with application to surrogate marker validation.

See Also

ICA.ContCont, MICA.ContCont, ICA.BinBin, ICA.BinBin.Grid.Sample.Uncert

Examples

```
# Compute R2_H given the marginals (sample from uniform),
# assuming no monotonicity
ICA_No2 <- ICA.BinBin.Grid.Sample.Uncert(pi1_1_=runif(10000, 0.3562, 0.4868),
pi0_1_=runif(10000, 0.0240, 0.0837), pi1_0_=runif(10000, 0.0240, 0.0837),
pi_1_1=runif(10000, 0.4434, 0.5742), pi_1_0=runif(10000, 0.0081, 0.0533),
pi_0_1=runif(10000, 0.0202, 0.0763), Seed=1, Monotonicity=c("No"), M=1000)

summary(ICA_No2)

# obtain plot of R2_H
plot(ICA_No2)
```

ICA.BinCont *Assess surrogacy in the causal-inference single-trial setting in the binary-continuous case*

Description

The function ICA.BinCont quantifies surrogacy in the single-trial setting within the causal-inference framework (individual causal association) when the surrogate endpoint is continuous (normally distributed) and the true endpoint is a binary outcome. For details, see Alonso Abad *et al.* (2023).

Usage

```
ICA.BinCont(Dataset, Surr, True, Treat,
  BS=FALSE,
  G_pi_10=c(0,1),
  G_rho_01_00=c(-1,1),
  G_rho_01_01=c(-1,1),
  G_rho_01_10=c(-1,1),
  G_rho_01_11=c(-1,1),
  Theta.S_0,
  Theta.S_1,
  M=1000, Seed=123,
  Monotonicity=FALSE,
  Independence=FALSE,
  HAA=FALSE,
  Cond_ind=FALSE,
  Plots=TRUE, Save.Plots="No", Show.Details=FALSE)
```

Arguments

`Dataset` A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, and a treatment indicator.

`Surr` The name of the variable in `Dataset` that contains the surrogate endpoint values.

`True` The name of the variable in `Dataset` that contains the true endpoint values.

`Treat` The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should be coded as 1 for the experimental group and -1 for the control group.

`BS` Logical. If `BS=TRUE`, the sampling variability is accounted for in the analysis by using a bootstrap procedure. Default `BS=FALSE`.

`G_pi_10` The lower and upper limits of the uniform distribution from which the probability parameter π_{10} is sampled. Default `c(0,1)`. When `Monotonicity=TRUE` the values of these limits are set as `c(0,0)`.

`G_rho_01_00` The lower and upper limits of the uniform distribution from which the association parameter ρ_{01}^{00} is sampled. Default `c(-1,1)`.

`G_rho_01_01` The lower and upper limits of the uniform distribution from which the association parameter ρ_{01}^{01} is sampled. Default `c(-1,1)`.

`G_rho_01_10` The lower and upper limits of the uniform distribution from which the association parameter ρ_{01}^{10} is sampled. Default `c(-1,1)`.

`G_rho_01_11` The lower and upper limits of the uniform distribution from which the association parameter ρ_{01}^{11} is sampled. Default `c(-1,1)`.

`Theta.S_0` The starting values of the means and standard deviations for the mixture distribution of the surrogate endpoint in the control group. The argument should contain eight values, where the first four values represent the starting values for the means and the last four values represent the starting values for the standard deviations. These starting values should be approximated based on the data on hand. Example: `Theta.S_0=c(-10,-5,5,10,10,10,10,10)`.

Theta.S_1 The starting values of the means and standard deviations for the mixture distribution of the surrogate endpoint in the treatment group. The argument should contain eight values, where the first four values represent the starting values for the means and the last four values represent the starting values for the standard deviations. These starting values should be approximated based on the data on hand. Example: Theta.S_1=c(-10,-5,5,10,10,10,10,10).

M The number of Monte Carlo iterations. Default M=1000.

Seed The random seed to be used in the analysis (for reproducibility). Default Seed=123.

Monotonicity Logical. If Monotonicity=TRUE, the analysis is performed assuming monotonicity, i.e. $P(T_1 < T_0) = 0$. Default Monotonicity=FALSE.

Independence Logical. If Independence=TRUE, the analysis is performed assuming independence between the treatment effect in both groups, i.e. $\pi_{ij} = \pi_{i.} \times \pi_{.j}$. Default Independence=FALSE.

HAA Logical. If HAA=TRUE, the analysis is performed assuming homogeneous association, i.e. $\rho_{01}^{ij} = \rho_{01}$. Default HAA=FALSE.

Cond_ind Logical. If Cond_ind=TRUE, the analysis is performed assuming conditional independence, i.e. $\rho_{01} = 0$. Default Cond_ind=FALSE.

Plots Logical. Should histograms of S_0 (surrogate endpoint in control group) and S_1 (surrogate endpoint in experimental treatment group) be provided together with density of fitted mixtures? Default Plots=TRUE.

Save.Plots Should the plots (see previous item) be saved? If Save.Plots="No", no plots are saved. If plots have to be saved, replace "No" by the desired location, e.g., Save.Plots="C:/". Default Save.Plots="No".

Show.Details Should some details regarding the availability of some output from the function be displayed in the console when the analysis is running? Setting Show.Details=TRUE could be useful for debugging procedure (if any). Default Show.Details=FALSE.

Value

An object of class ICA.BinCont with components,

R2_H The vector of the R_H^2 values.

pi_00 The vector of π_{00}^T values.

pi_01 The vector of π_{01}^T values.

pi_10 The vector of π_{10}^T values.

pi_11 The vector of π_{11}^T values.

G_rho_01_00 The vector of the ρ_{01}^{00} values.

G_rho_01_01 The vector of the ρ_{01}^{01} values.

G_rho_01_10 The vector of the ρ_{01}^{10} values.

G_rho_01_11 The vector of the ρ_{01}^{11} values.

pi_Delta_T_min1 The vector of the $\pi_{-1}^{\Delta T}$ values.

pi_Delta_T_0 The vector of the $\pi_{0}^{\Delta T}$ values.

`pi_Delta_T_1` The vector of the $\pi_1^{\Delta T}$ values.

`pi_0_00` The vector of π_{00} values of $f(S_0)$.

`pi_0_01` The vector of π_{01} values of $f(S_0)$.

`pi_0_10` The vector of π_{10} values of $f(S_0)$.

`pi_0_11` The vector of π_{11} values of $f(S_0)$.

`mu_0_00` The vector of mean μ_0^{00} values of $f(S_0)$.

`mu_0_01` The vector of mean μ_0^{01} values of $f(S_0)$.

`mu_0_10` The vector of mean μ_0^{10} values of $f(S_0)$.

`mu_0_11` The vector of mean μ_0^{11} values of $f(S_0)$.

`sigma2_00_00` The vector of variance σ_{00}^{00} values of $f(S_0)$.

`sigma2_00_01` The vector of variance σ_{00}^{01} values of $f(S_0)$.

`sigma2_00_10` The vector of variance σ_{00}^{10} values of $f(S_0)$.

`sigma2_00_11` The vector of variance σ_{00}^{11} values of $f(S_0)$.

`pi_1_00` The vector of π_{00} values of $f(S_1)$.

`pi_1_01` The vector of π_{01} values of $f(S_1)$.

`pi_1_10` The vector of π_{10} values of $f(S_1)$.

`pi_1_11` The vector of π_{11} values of $f(S_1)$.

`mu_1_00` The vector of mean μ_1^{00} values of $f(S_1)$.

`mu_1_01` The vector of mean μ_1^{01} values of $f(S_1)$.

`mu_1_10` The vector of mean μ_1^{10} values of $f(S_1)$.

`mu_1_11` The vector of mean μ_1^{11} values of $f(S_1)$.

`sigma2_11_00` The vector of variance σ_{11}^{00} values of $f(S_1)$.

`sigma2_11_01` The vector of variance σ_{11}^{01} values of $f(S_1)$.

`sigma2_11_10` The vector of variance σ_{11}^{10} values of $f(S_1)$.

`sigma2_11_11` The vector of variance σ_{11}^{11} values of $f(S_1)$.

`mean_Y_S0` The vector of mean μ_0 values of $f(S_0)$.

`mean_Y_S1` The vector of mean μ_1 values of $f(S_1)$.

`var_Y_S0` The vector of variance σ_{00} values of $f(S_0)$.

`var_Y_S1` The vector of variance σ_{11} values of $f(S_1)$.

`dev_S0` The vector of deviance values of the normal mixture for $f(S_0)$.

`dev_S1` The vector of deviance values of the normal mixture for $f(S_1)$.

`code_nlm_0` An integer indicating why the optimization process to estimate the mixture normal parameters of $f(S_0)$ terminated: 1) relative gradient is close to zero, current iterate is probably solution; 2) successive iterates within tolerance, current iterate is probably solution; 3) last global step failed to locate a point lower than the estimate, the estimate might be an approximate local minimum of the function.

`code_nlm_1` An integer indicating why the optimization process to estimate the mixture normal parameters of $f(S_1)$ terminated: 1) relative gradient is close to zero, current iterate is probably solution; 2) successive iterates within tolerance, current iterate is probably solution; 3) last global step failed to locate a point lower than the estimate, the estimate might be an approximate local minimum of the function.

`mean.S0` The mean of S_0.

`var.S0` The variance of S_0.

`mean.S1` The mean of S_1.

`var.S1` The variance of S_1.

Author(s)

Wim Van der Elst, Fenny Ong, Ariel Alonso, and Geert Molenberghs

References

Alonso Abad, A., Ong, F., Stijven, F., Van der Elst, W., Molenberghs, G., Van Keilegom, I., Verbeke, G., & Callegaro, A. (2023). An information-theoretic approach for the assessment of a continuous outcome as a surrogate for a binary true endpoint based on causal inference: Application to vaccine evaluation.

See Also

ICA.ContCont, MICA.ContCont, ICA.BinBin

Examples

```
## Not run: # Time consuming code part
data(Schizo)
Fit <- ICA.BinCont(Dataset = Schizo, Surr = BPRS, True = PANSS_Bin,
Theta.S_0=c(-10,-5,5,10,10,10,10,10), Theta.S_1=c(-10,-5,5,10,10,10,10,10),
Treat=Treat, M=50, Seed=1)

summary(Fit)
plot(Fit)

## End(Not run)
```

ICA.BinCont.BS *Assess surrogacy in the causal-inference single-trial setting in the binary-continuous case with an additional bootstrap procedure before the assessment*

Description

The function `ICA.BinCont.BS` quantifies surrogacy in the single-trial setting within the causal-inference framework (individual causal association) when the surrogate endpoint is continuous (normally distributed) and the true endpoint is a binary outcome. This function also allows for an additional bootstrap procedure before the assessment to take the imprecision due to finite sample size into account. For details, see Alonso Abad *et al.* (2023).

Usage

```
ICA.BinCont.BS(Dataset, Surr, True, Treat,
  BS=TRUE,
  nb=300,
  G_pi_10=c(0,1),
  G_rho_01_00=c(-1,1),
  G_rho_01_01=c(-1,1),
  G_rho_01_10=c(-1,1),
  G_rho_01_11=c(-1,1),
  Theta.S_0,
  Theta.S_1,
  M=1000, Seed=123,
  Monotonicity=FALSE,
  Independence=FALSE,
  HAA=FALSE,
  Cond_ind=FALSE,
  Plots=TRUE, Save.Plots="No", Show.Details=FALSE)
```

Arguments

`Dataset` A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, and a treatment indicator.

`Surr` The name of the variable in `Dataset` that contains the surrogate endpoint values.

`True` The name of the variable in `Dataset` that contains the true endpoint values.

`Treat` The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should be coded as 1 for the experimental group and -1 for the control group.

`BS` Logical. If `BS=TRUE`, the additional bootstrap procedure is performed before the sensitivity analysis to account for the the imprecision due to finite sample size. Default `BS=TRUE`.

`nb` The number of bootstrap. Default `nb=300`.

`G_pi_10` The lower and upper limits of the uniform distribution from which the probability parameter π_{10} is sampled. Default `c(0,1)`. Even though the default is `c(0,1)`, due to the restriction that all π_{ij} should be between `(0,1)`, the value of π_{10} will always be between $(0, min(\pi_{1.}, \pi_{.0}))$. When `Monotonicity=TRUE` the values of these limits are set as `c(0,0)`.

`G_rho_01_00` The lower and upper limits of the uniform distribution from which the association parameter ρ_{01}^{00} is sampled. Default `c(-1,1)`.

G_rho_01_01 The lower and upper limits of the uniform distribution from which the association parameter ρ_{01}^{01} is sampled. Default c(-1,1).

G_rho_01_10 The lower and upper limits of the uniform distribution from which the association parameter ρ_{01}^{10} is sampled. Default c(-1,1).

G_rho_01_11 The lower and upper limits of the uniform distribution from which the association parameter ρ_{01}^{11} is sampled. Default c(-1,1).

Theta.S_0 The starting values of the means and standard deviations for the mixture distribution of the surrogate endpoint in the control group. The argument should contain eight values, where the first four values represent the starting values for the means and the last four values represent the starting values for the standard deviations. These starting values should be approximated based on the data on hand. Example: Theta.S_0=c(-10,-5,5,10,10,10,10,10).

Theta.S_1 The starting values of the means and standard deviations for the mixture distribution of the surrogate endpoint in the treatment group. The argument should contain eight values, where the first four values represent the starting values for the means and the last four values represent the starting values for the standard deviations. These starting values should be approximated based on the data on hand. Example: Theta.S_1=c(-10,-5,5,10,10,10,10,10).

M The number of Monte Carlo iterations. Default M=1000.

Seed The random seed to be used in the analysis (for reproducibility). Default Seed=123.

Monotonicity Logical. If Monotonicity=TRUE, the analysis is performed assuming monotonicity, i.e. $P(T_1 < T_0) = 0$. Default Monotonicity=FALSE.

Independence Logical. If Independence=TRUE, the analysis is performed assuming independence between the treatment effect in both groups, i.e. $\pi_{ij} = \pi_{i.} \times \pi_{.j}$. Default Independence=FALSE.

HAA Logical. If HAA=TRUE, the analysis is performed assuming homogeneous association, i.e. $\rho_{01}^{ij} = \rho_{01}$. Default HAA=FALSE.

Cond_ind Logical. If Cond_ind=TRUE, the analysis is performed assuming conditional independence, i.e. $\rho_{01} = 0$. Default Cond_ind=FALSE.

Plots Logical. Should histograms of S_0 (surrogate endpoint in control group) and S_1 (surrogate endpoint in experimental treatment group) be provided together with density of fitted mixtures? Default Plots=TRUE.

Save.Plots Should the plots (see previous item) be saved? If Save.Plots="No", no plots are saved. If plots have to be saved, replace "No" by the desired location, e.g., Save.Plots="C:/". Default Save.Plots="No".

Show.Details Should some details regarding the availability of some output from the function be displayed in the console when the analysis is running? Setting Show.Details=TRUE could be useful for debugging procedure (if any). Default Show.Details=FALSE.

Value

An object of class ICA.BinCont with components,

nboots The identification number of bootstrap samples being analyzed in the sensitivity analysis.

`R2_H`	The vector of the R_H^2 values.
`pi_00`	The vector of π_{00}^T values.
`pi_01`	The vector of π_{01}^T values.
`pi_10`	The vector of π_{10}^T values.
`pi_11`	The vector of π_{11}^T values.
`G_rho_01_00`	The vector of the ρ_{01}^{00} values.
`G_rho_01_01`	The vector of the ρ_{01}^{01} values.
`G_rho_01_10`	The vector of the ρ_{01}^{10} values.
`G_rho_01_11`	The vector of the ρ_{01}^{11} values.
`mu_0_00`	The vector of mean μ_0^{00} values of $f(S_0)$.
`mu_0_01`	The vector of mean μ_0^{01} values of $f(S_0)$.
`mu_0_10`	The vector of mean μ_0^{10} values of $f(S_0)$.
`mu_0_11`	The vector of mean μ_0^{11} values of $f(S_0)$.
`mu_1_00`	The vector of mean μ_1^{00} values of $f(S_1)$.
`mu_1_01`	The vector of mean μ_1^{01} values of $f(S_1)$.
`mu_1_10`	The vector of mean μ_1^{10} values of $f(S_1)$.
`mu_1_11`	The vector of mean μ_1^{11} values of $f(S_1)$.
`sigma_00`	The vector of variance σ_{00} values of $f(S_0)$.
`sigma_11`	The vector of variance σ_{11} values of $f(S_1)$.

Author(s)

Wim Van der Elst, Fenny Ong, Ariel Alonso, and Geert Molenberghs

References

Alonso Abad, A., Ong, F., Stijven, F., Van der Elst, W., Molenberghs, G., Van Keilegom, I., Verbeke, G., & Callegaro, A. (2023). An information-theoretic approach for the assessment of a continuous outcome as a surrogate for a binary true endpoint based on causal inference: Application to vaccine evaluation.

See Also

`ICA.BinCont`

Examples

```
## Not run: # Time consuming code part
data(Schizo)
Fit <- ICA.BinCont.BS(Dataset = Schizo, Surr = BPRS, True = PANSS_Bin, nb = 10,
Theta.S_0=c(-10,-5,5,10,10,10,10,10), Theta.S_1=c(-10,-5,5,10,10,10,10,10),
Treat=Treat, M=50, Seed=1)

summary(Fit)
plot(Fit)

## End(Not run)
```

ICA.ContCont *Assess surrogacy in the causal-inference single-trial setting (Individual Causal Association, ICA) in the Continuous-continuous case*

Description

The function `ICA.ContCont` quantifies surrogacy in the single-trial causal-inference framework. See **Details** below.

Usage

```
ICA.ContCont(T0S0, T1S1, T0T0=1, T1T1=1, S0S0=1, S1S1=1, T0T1=seq(-1, 1, by=.1),
T0S1=seq(-1, 1, by=.1), T1S0=seq(-1, 1, by=.1), S0S1=seq(-1, 1, by=.1))
```

Arguments

`T0S0` A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the control treatment condition that should be considered in the computation of ρ_{Δ}.

`T1S1` A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_{Δ}.

`T0T0` A scalar that specifies the variance of the true endpoint in the control treatment condition that should be considered in the computation of ρ_{Δ}. Default 1.

`T1T1` A scalar that specifies the variance of the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_{Δ}. Default 1.

`S0S0` A scalar that specifies the variance of the surrogate endpoint in the control treatment condition that should be considered in the computation of ρ_{Δ}. Default 1.

`S1S1` A scalar that specifies the variance of the surrogate endpoint in the experimental treatment condition that should be considered in the computation of ρ_{Δ}. Default 1.

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.1)`, i.e., the values -1, -0.9, -0.8, $\ldots$, 1.

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.1)`.

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.1)`.

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.1)`.

Details

Based on the causal-inference framework, it is assumed that each subject *j* has four counterfactuals (or potential outcomes), i.e., T_{0j}, T_{1j}, S_{0j}, and S_{1j}. Let T_{0j} and T_{1j} denote the counterfactuals for the true endpoint (T) under the control ($Z = 0$) and the experimental ($Z = 1$) treatments of subject *j*, respectively. Similarly, S_{0j} and S_{1j} denote the corresponding counterfactuals for the surrogate endpoint (S) under the control and experimental treatments, respectively. The individual causal effects of Z on T and S for a given subject *j* are then defined as $\Delta_{T_j} = T_{1j} - T_{0j}$ and $\Delta_{S_j} = S_{1j} - S_{0j}$, respectively.

In the single-trial causal-inference framework, surrogacy can be quantified as the correlation between the individual causal effects of Z on S and T (for details, see Alonso et al., submitted):

$$\rho_{\Delta} = \rho(\Delta_{T_j}, \Delta_{S_j}) = \frac{\sqrt{\sigma_{S_0S_0}\sigma_{T_0T_0}}\rho_{S_0T_0} + \sqrt{\sigma_{S_1S_1}\sigma_{T_1T_1}}\rho_{S_1T_1} - \sqrt{\sigma_{S_0S_0}\sigma_{T_1T_1}}\rho_{S_0T_1} - \sqrt{\sigma_{S_1S_1}\sigma_{T_0T_0}}\rho_{S_1T_0}}{\sqrt{(\sigma_{T_0T_0} + \sigma_{T_1T_1} - 2\sqrt{\sigma_{T_0T_0}\sigma_{T_1T_1}}\rho_{T_0T_1})(\sigma_{S_0S_0} + \sigma_{S_1S_1} - 2\sqrt{\sigma_{S_0S_0}\sigma_{S_1S_1}}\rho_{S_0S_1})}},$$

where the correlations $\rho_{S_0T_1}$, $\rho_{S_1T_0}$, $\rho_{T_0T_1}$, and $\rho_{S_0S_1}$ are not estimable. It is thus warranted to conduct a sensitivity analysis (by considering vectors of possible values for the correlations between the counterfactuals – rather than point estimates).

When the user specifies a vector of values that should be considered for one or more of the counterfactual correlations in the above expression, the function `ICA.ContCont` constructs all possible matrices that can be formed as based on these values, identifies the matrices that are positive definite (i.e., valid correlation matrices), and computes ρ_{Δ} for each of these matrices. The obtained vector of ρ_{Δ} values can subsequently be used to examine (i) the impact of different assumptions regarding the correlations between the counterfactuals on the results (see also `plot Causal-Inference ContCont`), and (ii) the extent to which proponents of the causal-inference and meta-analytic frameworks will reach the same conclusion with respect to the appropriateness of the candidate surrogate at hand.

The function `ICA.ContCont` also generates output that is useful to examine the plausibility of finding a good surrogate endpoint (see `GoodSurr` in the **Value** section below). For details, see Alonso et al. (submitted).

Notes

A single ρ_{Δ} value is obtained when all correlations in the function call are scalars.

Value

An object of class `ICA.ContCont` with components,

`Total.Num.Matrices`
An object of class `numeric` that contains the total number of matrices that can be formed as based on the user-specified correlations in the function call.

`Pos.Def` A `data.frame` that contains the positive definite matrices that can be formed based on the user-specified correlations. These matrices are used to compute the vector of the ρ_{Δ} values.

`ICA` A scalar or vector that contains the individual causal association (ICA; ρ_{Δ}) value(s).

`GoodSurr` A `data.frame` that contains the ICA (ρ_{Δ}), σ_{Δ_T}, and δ.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal-inference and meta-analytic paradigms for the validation of surrogate markers.

See Also

MICA.ContCont, ICA.Sample.ContCont, Single.Trial.RE.AA, plot Causal-Inference ContCont

Examples

```
## Not run:  #time-consuming code parts
# Generate the vector of ICA.ContCont values when rho_T0S0=rho_T1S1=.95,
# sigma_T0T0=90, sigma_T1T1=100,sigma_ S0S0=10, sigma_S1S1=15, and
# the grid of values {0, .2, ..., 1} is considered for the correlations
# between the counterfactuals:
SurICA <- ICA.ContCont(T0S0=.95, T1S1=.95, T0T0=90, T1T1=100, S0S0=10, S1S1=15,
T0T1=seq(0, 1, by=.2), T0S1=seq(0, 1, by=.2), T1S0=seq(0, 1, by=.2),
S0S1=seq(0, 1, by=.2))

# Examine and plot the vector of generated ICA values:
summary(SurICA)
plot(SurICA)

# Obtain the positive definite matrices than can be formed as based on the
# specified (vectors) of the correlations (these matrices are used to
# compute the ICA values)
SurICA$Pos.Def

# Same, but specify vectors for rho_T0S0 and rho_T1S1: Sample from
# normal with mean .95 and SD=.05 (to account for uncertainty
# in estimation)
SurICA2 <- ICA.ContCont(T0S0=rnorm(n=10000000, mean=.95, sd=.05),
T1S1=rnorm(n=10000000, mean=.95, sd=.05),
T0T0=90, T1T1=100, S0S0=10, S1S1=15,
T0T1=seq(0, 1, by=.2), T0S1=seq(0, 1, by=.2), T1S0=seq(0, 1, by=.2),
S0S1=seq(0, 1, by=.2))

# Examine results
summary(SurICA2)
plot(SurICA2)

## End(Not run)
```

ICA.ContCont.MultS *Assess surrogacy in the causal-inference single-trial setting (Individual Causal Association, ICA) using a continuous univariate T and multiple continuous S*

Description

The function `ICA.ContCont.MultS` quantifies surrogacy in the single-trial causal-inference framework where T is continuous and there are multiple continuous S.

Usage

```
ICA.ContCont.MultS(M = 500, N, Sigma,
G = seq(from=-1, to=1, by = .00001),
Seed=c(123), Show.Progress=FALSE)
```

Arguments

`M`	The number of multivariate ICA values (R^2_H) that should be sampled. Default `M=500`.
`N`	The sample size of the dataset.
`Sigma`	A matrix that specifies the variance-covariance matrix between T_0, T_1, S_{10}, S_{11}, S_{20}, S_{21}, ..., S_{k0}, and S_{k1} (in this order, the T_0 and T_1 data should be in `Sigma[c(1,2), c(1,2)]`, the S_{10} and S_{11} data should be in `Sigma[c(3,4), c(3,4)]`, and so on). The unidentifiable covariances should be defined as `NA` (see example below).
`G`	A vector of the values that should be considered for the unidentified correlations. Default `G=seq(-1, 1, by=.00001)`, i.e., values with range -1 to 1.
`Seed`	The seed that is used. Default `Seed=123`.
`Show.Progress`	Should progress of runs be graphically shown? (i.e., 1% done..., 2% done..., etc). Mainly useful when a large number of S have to be considered (to follow progress and estimate total run time).

Details

The multivariate ICA (R^2_H) is not identifiable because the individual causal treatment effects on T, S_1, ..., S_k cannot be observed. A simulation-based sensitivity analysis is therefore conducted in which the multivariate ICA (R^2_H) is estimated across a set of plausible values for the unidentifiable correlations. To this end, consider the variance covariance matrix of the potential outcomes $\mathbf{\Sigma}$ (0 and 1 subscripts refer to the control and experimental treatments, respectively):

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_{T_0T_0} & & & & & & & & \\ \sigma_{T_0T_1} & \sigma_{T_1T_1} & & & & & & & \\ \sigma_{T_0S1_0} & \sigma_{T_1S1_0} & \sigma_{S1_0S1_0} & & & & & & \\ \sigma_{T_0S1_1} & \sigma_{T_1S1_1} & \sigma_{S1_0S1_1} & \sigma_{S1_1S1_1} & & & & & \\ \sigma_{T_0S2_0} & \sigma_{T_1S2_0} & \sigma_{S1_0S2_0} & \sigma_{S1_1S2_0} & \sigma_{S2_0S2_0} & & & & \\ \sigma_{T_0S2_1} & \sigma_{T_1S2_1} & \sigma_{S1_0S2_1} & \sigma_{S1_1S2_1} & \sigma_{S2_0S2_1} & \sigma_{S2_1S2_1} & & & \\ ... & ... & ... & ... & ... & ... & \ddots & & \\ \sigma_{T_0Sk_0} & \sigma_{T_1Sk_0} & \sigma_{S1_0Sk_0} & \sigma_{S1_1Sk_0} & \sigma_{S2_0Sk_0} & \sigma_{S2_1Sk_0} & \cdots & \sigma_{Sk_0Sk_0} & \\ \sigma_{T_0Sk_1} & \sigma_{T_1Sk_1} & \sigma_{S1_0Sk_1} & \sigma_{S1_1Sk_1} & \sigma_{S2_0Sk_1} & \sigma_{S2_1Sk_1} & \cdots & \sigma_{Sk_0Sk_1} & \sigma_{Sk_1Sk_1}. \end{pmatrix}$$

The `ICA.ContCont.MultS` function requires the user to specify a distribution G for the unidentified correlations. Next, the identifiable correlations are fixed at their estimated values and the unidentifiable correlations are independently and randomly sampled from G. In the function call, the unidentifiable correlations are marked by specifying `NA` in the `Sigma` matrix (see example section below). The algorithm generates a large number of 'completed' matrices, and only those that are positive definite are retained (the number of positive definite matrices that should be obtained is specified by the `M=` argument in the function call). Based on the identifiable variances, these positive definite correlation matrices are converted to covariance matrices $\mathbf{\Sigma}$ and the multiple-surrogate ICA are estimated.

An issue with this approach (i.e., substituting unidentified correlations by random and independent samples from G) is that the probability of obtaining a positive definite matrix is very low when the dimensionality of the matrix increases. One approach to increase the efficiency of the algorithm is to build-up the correlation matrix in a gradual way. In particular, the property that a $(k \times k)$ matrix is positive definite if and only if all principal minors are positive (i.e., Sylvester's criterion) can be used. In other words, a $(k \times k)$ matrix is positive definite when the determinants of the upper-left (2×2), (3×3), ..., $(k \times k)$ submatrices all have a positive determinant. Thus, when a positive definite $(k \times k)$ matrix has to be generated, one can start with the upper-left (2×2) submatrix and randomly sample a value from the unidentified correlation (here: $\rho_{T_0T_0}$) from G. When the determinant is positive (which will always be the case for a (2×2) matrix), the same procedure is used for the upper-left (3×3) submatrix, and so on. When a particular draw from G for a particular submatrix does not give a positive determinant, new values are sampled for the unidentified correlations until a positive determinant is obtained. In this way, it can be guaranteed that the final $(k \times k)$ submatrix will be positive definite. The latter approach is used in the current function. This procedure is used to generate many positive definite matrices. Based on these matrices, $\mathbf{\Sigma_\Delta}$ is generated and the multivariate ICA (R^2_H) is computed (for details, see Van der Elst et al., 2017).

Value

An object of class `ICA.ContCont.MultS` with components,

`R2_H` The multiple-surrogate individual causal association value(s).

`Corr.R2_H` The corrected multiple-surrogate individual causal association value(s).

`Lower.Dig.Corrs.All`
A `data.frame` that contains the matrix that contains the identifiable and unidentifiable correlations (lower diagonal elements) that were used to compute (R^2_H) in the run.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Van der Elst, W., Alonso, A. A., & Molenberghs, G. (2017). Univariate versus multivariate surrogate endpoints.

See Also

MICA.ContCont, ICA.ContCont, Single.Trial.RE.AA, plot Causal-Inference ContCont, ICA.ContCont.MultS_alt

Examples

```
## Not run:  #time-consuming code parts
# Specify matrix Sigma (var-cavar matrix T_0, T_1, S1_0, S1_1, ...)
# here for 1 true endpoint and 3 surrogates

s<-matrix(rep(NA, times=64),8)
s[1,1] <- 450; s[2,2] <- 413.5; s[3,3] <- 174.2; s[4,4] <- 157.5;
s[5,5] <- 244.0; s[6,6] <- 229.99; s[7,7] <- 294.2; s[8,8] <- 302.5
s[3,1] <- 160.8; s[5,1] <- 208.5; s[7,1] <- 268.4
s[4,2] <- 124.6; s[6,2] <- 212.3; s[8,2] <- 287.1
s[5,3] <- 160.3; s[7,3] <- 142.8
s[6,4] <- 134.3; s[8,4] <- 130.4
s[7,5] <- 209.3;
s[8,6] <- 214.7
s[upper.tri(s)] = t(s)[upper.tri(s)]

# Marix looks like (NA indicates unidentified covariances):
#            T_0    T_1  S1_0  S1_1  S2_0   S2_1  S2_0  S2_1
#            [,1]  [,2]  [,3]  [,4]  [,5]   [,6]  [,7]  [,8]
# T_0  [1,] 450.0    NA 160.8    NA 208.5     NA 268.4    NA
# T_1  [2,]    NA 413.5    NA 124.6    NA 212.30    NA 287.1
# S1_0 [3,] 160.8    NA 174.2    NA 160.3     NA 142.8    NA
# S1_1 [4,]    NA 124.6    NA 157.5    NA 134.30    NA 130.4
# S2_0 [5,] 208.5    NA 160.3    NA 244.0     NA 209.3    NA
# S2_1 [6,]    NA 212.3    NA 134.3    NA 229.99    NA 214.7
# S3_0 [7,] 268.4    NA 142.8    NA 209.3     NA 294.2    NA
# S3_1 [8,]    NA 287.1    NA 130.4    NA 214.70    NA 302.5

# Conduct analysis
ICA <- ICA.ContCont.MultS(M=100, N=200, Show.Progress = TRUE,
  Sigma=s, G = seq(from=-1, to=1, by = .00001), Seed=c(123))

# Explore results
summary(ICA)
plot(ICA)

## End(Not run)
```

ICA.ContCont.MultS.MPC

Assess surrogacy in the causal-inference single-trial setting (Individual Causal Association, ICA) using a continuous univariate T and multiple continuous S, by simulating correlation matrices using a modified algorithm based on partial correlations

Description

The function ICA.ContCont.MultS.MPC quantifies surragacy in the single-trial causal-inference framework in which the true endpoint (T) and multiple surrogates (S) are continuous. This function is a modification of the ICA.ContCont.MultS.PC algorithm based on partial correlations. it mitigates the effect of non-informative surrogates and effectively explores the PD space to capture the ICA range (Florez, et al. 2021).

Usage

```
ICA.ContCont.MultS.MPC(M=1000,N,Sigma,prob = NULL,Seed=123,
Save.Corr=F, Show.Progress=FALSE)
```

Arguments

M
: The number of multivariate ICA values (R^2_H) that should be sampled. Default M=1000.

N
: The sample size of the dataset.

Sigma
: A matrix that specifies the variance-covariance matrix between T_0, T_1, S_{10}, S_{11}, S_{20}, S_{21}, ..., S_{k0}, and S_{k1} (in this order, the T_0 and T_1 data should be in Sigma[c(1,2), c(1,2)], the S_{10} and S_{11} data should be in Sigma[c(3,4), c(3,4)], and so on). The unidentifiable covariances should be defined as NA (see example below).

prob
: vector of probabilities to choose the number of surrogates (r) with their non-identifiable correlations equal to zero. The default (prob=NULL) vector of probabilities is:

$$\pi_r = \frac{\binom{p}{r}}{\sum_{i=1}^{p}\binom{p}{i}}, \text{ for } r = 0, \ldots, p.$$

In this way, each possible combination of $r$ surrogates has the same probability of being selected.

Save.Corr
: If true, the lower diagonal elements of the correlation matrix (identifiable and unidientifiable elements) are stored. If false, these results are not saved.

Seed
: The seed that is used. Default Seed=123.

Show.Progress
: Should progress of runs be graphically shown? (i.e., 1% done..., 2% done..., etc). Mainly useful when a large number of S have to be considered (to follow progress and estimate total run time).

Details

The multivariate ICA (R^2_H) is not identifiable because the individual causal treatment effects on T, S_1, ..., S_k cannot be observed. A simulation-based sensitivity analysis is therefore conducted in which the multivariate ICA (R^2_H) is estimated across a set of plausible values for the unidentifiable correlations. To this end, consider the variance covariance matrix of the potential outcomes $\mathbf{\Sigma}$ (0 and 1 subscripts refer to the control and experimental treatments, respectively):

$$\mathbf{\Sigma} = \begin{pmatrix}
\sigma_{T_0T_0} & & & & & & & & \\
\sigma_{T_0T_1} & \sigma_{T_1T_1} & & & & & & & \\
\sigma_{T_0S1_0} & \sigma_{T_1S1_0} & \sigma_{S1_0S1_0} & & & & & & \\
\sigma_{T_0S1_1} & \sigma_{T_1S1_1} & \sigma_{S1_0S1_1} & \sigma_{S1_1S1_1} & & & & & \\
\sigma_{T_0S2_0} & \sigma_{T_1S2_0} & \sigma_{S1_0S2_0} & \sigma_{S1_1S2_0} & \sigma_{S2_0S2_0} & & & & \\
\sigma_{T_0S2_1} & \sigma_{T_1S2_1} & \sigma_{S1_0S2_1} & \sigma_{S1_1S2_1} & \sigma_{S2_0S2_1} & \sigma_{S2_1S2_1} & & & \\
... & ... & ... & ... & ... & ... & \ddots & & \\
\sigma_{T_0Sk_0} & \sigma_{T_1Sk_0} & \sigma_{S1_0Sk_0} & \sigma_{S1_1Sk_0} & \sigma_{S2_0Sk_0} & \sigma_{S2_1Sk_0} & \cdots & \sigma_{Sk_0Sk_0} & \\
\sigma_{T_0Sk_1} & \sigma_{T_1Sk_1} & \sigma_{S1_0Sk_1} & \sigma_{S1_1Sk_1} & \sigma_{S2_0Sk_1} & \sigma_{S2_1Sk_1} & \cdots & \sigma_{Sk_0Sk_1} & \sigma_{Sk_1Sk_1}.
\end{pmatrix}$$

The identifiable correlations are fixed at their estimated values and the unidentifiable correlations are independently and randomly sampled using a modification of an algorithm based on partial correlations (PC), called modified partial correlation (MPC) algorithm. In the function call, the unidentifiable correlations are marked by specifying `NA` in the `Sigma` matrix (see example section below).

The PC algorithm generate each correlation matrix progressively based on parameterization of terms of the correlations $\rho_{i,i+1}$, for $i = 1, \ldots, d-1$, and the partial correlations $\rho_{i,j|i+1,\ldots,j-1}$, for $j - i > 2$ (for details, see Joe, 2006 and Florez et al., 2018). The MPC algorithm randomly fixed some of the unidentifiable correlations to zero in order to explore the PD, which is coherent with the estimable entries of the correlation matrix, to capture the ICA range more efficiently.

Based on the identifiable variances, these correlation matrices are converted to covariance matrices $\mathbf{\Sigma}$ and the multiple-surrogate ICA are estimated (for details, see Van der Elst et al., 2017).

This approach to simulate the unidentifiable parameters of $\mathbf{\Sigma}$ is computationally more efficient than the one used in the function `ICA.ContCont.MultS`.

Value

An object of class `ICA.ContCont.MultS.PC` with components,

`R2_H` The multiple-surrogate individual causal association value(s).

`Corr.R2_H` The corrected multiple-surrogate individual causal association value(s).

`Lower.Dig.Corrs.All`
A `data.frame` that contains the matrix that contains the identifiable and unidentifiable correlations (lower diagonal elements) that were used to compute (R^2_H) in the run.

`surr.eval.r` Matrix indicating the surrogates of which their unidentifiable correlations are fixed to zero in each simulation.

Author(s)

Wim Van der Elst, Ariel Alonso, Geert Molenberghs & Alvaro Florez

References

Florez, A., Molenberghs, G., Van der Elst, W., Alonso, A. A. (2021). An efficient algorithm for causally assessing surrogacy in a multivariate setting.

Florez, A., Alonso, A. A., Molenberghs, G. & Van der Elst, W. (2020). Generating random correlation matrices with fixed values: An application to the evaluation of multivariate surrogate endpoints. *Computational Statistics & Data Analysis 142*.

Joe, H. (2006). Generating random correlation matrices based on partial correlations. *Journal of Multivariate Analysis, 97(10)*:2177-2189.

Van der Elst, W., Alonso, A. A., & Molenberghs, G. (2017). Univariate versus multivariate surrogate endpoints.

See Also

MICA.ContCont, ICA.ContCont, Single.Trial.RE.AA, plot Causal-Inference ContCont, ICA.ContCont.MultS, ICA.ContCont.MultS_alt

Examples

```
## Not run:
# Specify matrix Sigma (var-cavar matrix T_0, T_1, S1_0, S1_1, ...)
# here we have 1 true endpoint and 10 surrogates (8 of these are non-informative)

Sigma = ks::invvech(
  c(25, NA, 17.8, NA, -10.6, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA,
    4, NA, -0.32, NA, -1.32, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, 16,
    NA, -4, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 1, NA, 0.48, NA,
    0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, 16, NA, 0, NA, 0, NA, 0, NA, 0,
    NA, 0, NA, 0, NA, 0, NA, 0, NA, 1, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0,
    NA, 0, 16, NA, 8, NA, 8, NA, 8, NA, 8, NA, 8, NA, 8, NA, 8, NA, 1, NA, 0.5, NA, 0.5,
    NA, 0.5, NA, 0.5, NA, 0.5, NA, 0.5, NA, 0.5, 16, NA, 8, NA, 8, NA, 8, NA, 8, NA, 8,
    NA, 8, NA, 1, NA, 0.5, NA, 0.5, NA, 0.5, NA, 0.5, NA, 0.5, NA, 0.5, 16, NA, 8, NA,
    8, NA, 8, NA, 8, NA, 8, NA, 1,NA,0.5,NA,0.5,NA,0.5,NA,0.5,NA,0.5, 16, NA, 8, NA, 8,
    NA, 8, NA, 8, NA, 1, NA, 0.5, NA, 0.5, NA, 0.5, NA, 0.5, 16, NA, 8, NA, 8, NA, 8, NA,
    1, NA, 0.5, NA, 0.5, NA, 0.5, 16, NA, 8, NA, 8, NA, 1, NA, 0.5, NA, 0.5, 16, NA, 8, NA,
    1, NA, 0.5, 16, NA, 1))

# Conduct analysis using the PC and MPC algorithm
## first evaluating two surrogates
ICA.PC.2 = ICA.ContCont.MultS.PC(M = 30000, N=200, Sigma[1:6,1:6], Seed = 123)
ICA.MPC.2 = ICA.ContCont.MultS.MPC(M = 30000, N=200, Sigma[1:6,1:6],prob=NULL,
Seed = 123, Save.Corr=T, Show.Progress = TRUE)

## later evaluating two surrogates
ICA.PC.10 = ICA.ContCont.MultS.PC(M = 150000, N=200, Sigma, Seed = 123)
ICA.MPC.10 = ICA.ContCont.MultS.MPC(M = 150000, N=200, Sigma,prob=NULL,
Seed = 123, Save.Corr=T, Show.Progress = TRUE)

# Explore results
range(ICA.PC.2$R2_H)
range(ICA.PC.10$R2_H)
range(ICA.MPC.2$R2_H)
range(ICA.MPC.10$R2_H)
## as we observe, the MPC algorithm displays a wider interval of possible values for the ICA

## End(Not run)
```

ICA.ContCont.MultS.PC *Assess surrogacy in the causal-inference single-trial setting (Individual Causal Association, ICA) using a continuous univariate T and multiple continuous S, by simulating correlation matrices using an algorithm based on partial correlations*

Description

The function `ICA.ContCont.MultS` quantifies surrogacy in the single-trial causal-inference framework where T is continuous and there are multiple continuous S. This function provides an alternative for `ICA.ContCont.MultS`.

Usage

```
ICA.ContCont.MultS.PC(M=1000,N,Sigma,Seed=123,Show.Progress=FALSE)
```

Arguments

`M`	The number of multivariate ICA values (R^2_H) that should be sampled. Default `M=1000`.
`N`	The sample size of the dataset.
`Sigma`	A matrix that specifies the variance-covariance matrix between T_0, T_1, S_{10}, S_{11}, S_{20}, S_{21}, ..., S_{k0}, and S_{k1} (in this order, the T_0 and T_1 data should be in `Sigma[c(1,2), c(1,2)]`, the S_{10} and S_{11} data should be in `Sigma[c(3,4), c(3,4)]`, and so on). The unidentifiable covariances should be defined as `NA` (see example below).
`Seed`	The seed that is used. Default `Seed=123`.
`Show.Progress`	Should progress of runs be graphically shown? (i.e., 1% done..., 2% done..., etc). Mainly useful when a large number of S have to be considered (to follow progress and estimate total run time).

Details

The multivariate ICA (R^2_H) is not identifiable because the individual causal treatment effects on T, S_1, ..., S_k cannot be observed. A simulation-based sensitivity analysis is therefore conducted in which the multivariate ICA (R^2_H) is estimated across a set of plausible values for the unidentifiable correlations. To this end, consider the variance covariance matrix of the potential outcomes $\mathbf{\Sigma}$ (0 and 1 subscripts refer to the control and experimental treatments, respectively):

$$\mathbf{\Sigma} = \left(\begin{array}{ccccccccc}
\sigma_{T_0T_0} & & & & & & & & \\
\sigma_{T_0T_1} & \sigma_{T_1T_1} & & & & & & & \\
\sigma_{T_0S1_0} & \sigma_{T_1S1_0} & \sigma_{S1_0S1_0} & & & & & & \\
\sigma_{T_0S1_1} & \sigma_{T_1S1_1} & \sigma_{S1_0S1_1} & \sigma_{S1_1S1_1} & & & & & \\
\sigma_{T_0S2_0} & \sigma_{T_1S2_0} & \sigma_{S1_0S2_0} & \sigma_{S1_1S2_0} & \sigma_{S2_0S2_0} & & & & \\
\sigma_{T_0S2_1} & \sigma_{T_1S2_1} & \sigma_{S1_0S2_1} & \sigma_{S1_1S2_1} & \sigma_{S2_0S2_1} & \sigma_{S2_1S2_1} & & & \\
... & ... & ... & ... & ... & ... & \ddots & & \\
\sigma_{T_0Sk_0} & \sigma_{T_1Sk_0} & \sigma_{S1_0Sk_0} & \sigma_{S1_1Sk_0} & \sigma_{S2_0Sk_0} & \sigma_{S2_1Sk_0} & \cdots & \sigma_{Sk_0Sk_0} & \\
\sigma_{T_0Sk_1} & \sigma_{T_1Sk_1} & \sigma_{S1_0Sk_1} & \sigma_{S1_1Sk_1} & \sigma_{S2_0Sk_1} & \sigma_{S2_1Sk_1} & \cdots & \sigma_{Sk_0Sk_1} & \sigma_{Sk_1Sk_1}.
\end{array} \right)$$

The identifiable correlations are fixed at their estimated values and the unidentifiable correlations are independently and randomly sampled using an algorithm based on partial correlations (PC). In the function call, the unidentifiable correlations are marked by specifying `NA` in the `Sigma` matrix (see example section below). The PC algorithm generate each correlation matrix progressively based on parameterization of terms of the correlations $\rho_{i,i+1}$, for $i = 1, \ldots, d-1$, and the partial correlations $\rho_{i,j|i+1,\ldots,j-1}$, for $j - i > 2$ (for details, see Joe, 2006 and Florez et al., 2018). Based on the identifiable variances, these correlation matrices are converted to covariance matrices $\mathbf{\Sigma}$ and the multiple-surrogate ICA are estimated (for details, see Van der Elst et al., 2017).

This approach to simulate the unidentifiable parameters of $\mathbf{\Sigma}$ is computationally more efficient than the one used in the function `ICA.ContCont.MultS`.

Value

An object of class `ICA.ContCont.MultS.PC` with components,

`R2_H` The multiple-surrogate individual causal association value(s).

`Corr.R2_H` The corrected multiple-surrogate individual causal association value(s).

`Lower.Dig.Corrs.All`
A `data.frame` that contains the matrix that contains the identifiable and unidentifiable correlations (lower diagonal elements) that were used to compute (R^2_H) in the run.

Author(s)

Alvaro Florez

References

Florez, A., Alonso, A. A., Molenberghs, G. & Van der Elst, W. (2018). Simulation of random correlation matrices with fixed values: comparison of algorithms and application on multiple surrogates assessment.

Joe, H. (2006). Generating random correlation matrices based on partial correlations. *Journal of Multivariate Analysis, 97(10)*:2177-2189.

Van der Elst, W., Alonso, A. A., & Molenberghs, G. (2017). Univariate versus multivariate surrogate endpoints.

See Also

MICA.ContCont, ICA.ContCont, Single.Trial.RE.AA, plot Causal-Inference ContCont, ICA.ContCont.MultS, ICA.ContCont.MultS_alt

Examples

```
## Not run:
# Specify matrix Sigma (var-cavar matrix T_0, T_1, S1_0, S1_1, ...)
# here for 1 true endpoint and 3 surrogates

s<-matrix(rep(NA, times=64),8)
s[1,1] <- 450; s[2,2] <- 413.5; s[3,3] <- 174.2; s[4,4] <- 157.5;
s[5,5] <- 244.0; s[6,6] <- 229.99; s[7,7] <- 294.2; s[8,8] <- 302.5
s[3,1] <- 160.8; s[5,1] <- 208.5; s[7,1] <- 268.4
s[4,2] <- 124.6; s[6,2] <- 212.3; s[8,2] <- 287.1
s[5,3] <- 160.3; s[7,3] <- 142.8
s[6,4] <- 134.3; s[8,4] <- 130.4
s[7,5] <- 209.3;
s[8,6] <- 214.7
s[upper.tri(s)] = t(s)[upper.tri(s)]

# Marix looks like (NA indicates unidentified covariances):
#             T_0    T_1  S1_0  S1_1  S2_0   S2_1  S2_0  S2_1
#             [,1]  [,2]  [,3]  [,4]  [,5]   [,6]  [,7]  [,8]
# T_0  [1,] 450.0    NA 160.8    NA 208.5     NA 268.4    NA
# T_1  [2,]    NA 413.5    NA 124.6    NA 212.30    NA 287.1
# S1_0 [3,] 160.8    NA 174.2    NA 160.3     NA 142.8    NA
# S1_1 [4,]    NA 124.6    NA 157.5    NA 134.30    NA 130.4
# S2_0 [5,] 208.5    NA 160.3    NA 244.0     NA 209.3    NA
# S2_1 [6,]    NA 212.3    NA 134.3    NA 229.99    NA 214.7
# S3_0 [7,] 268.4    NA 142.8    NA 209.3     NA 294.2    NA
# S3_1 [8,]    NA 287.1    NA 130.4    NA 214.70    NA 302.5

# Conduct analysis
ICA <- ICA.ContCont.MultS.PC(M=1000, N=200, Show.Progress = TRUE,
Sigma=s, Seed=c(123))

# Explore results
summary(ICA)
plot(ICA)

## End(Not run)
```

ICA.ContCont.MultS_alt

Assess surrogacy in the causal-inference single-trial setting (Individual Causal Association, ICA) using a continuous univariate T and multiple continuous S, alternative approach

---

Description

The function `ICA.ContCont.MultS_alt` quantifies surrogacy in the single-trial causal-inference framework where T is continuous and there are multiple continuous S. This function provides an alternative for `ICA.ContCont.MultS`.

Usage

```
ICA.ContCont.MultS_alt(M = 500, N, Sigma,
G = seq(from=-1, to=1, by = .00001),
Seed=c(123), Model = "Delta_T ~ Delta_S1 + Delta_S2",
Show.Progress=FALSE)
```

Arguments

`M` The number of multivariate ICA values (R^2_H) that should be sampled. Default `M=500`.

`N` The sample size of the dataset.

`Sigma` A matrix that specifies the variance-covariance matrix between T_0, T_1, S_{10}, S_{11}, S_{20}, S_{21}, ..., S_{k0}, and S_{k1}. The unidentifiable covariances should be defined as `NA` (see example below).

`G` A vector of the values that should be considered for the unidentified correlations. Default `G=seq(-1, 1, by=.00001)`, i.e., values with range -1 to 1.

`Seed` The seed that is used. Default `Seed=123`.

`Model` The multivariate ICA (R^2_H) is essentially the coefficient of determination of a regression model in which ΔT is regressed on ΔS_1, ΔS_2, ... and so on. The `Model=` argument specifies the regression model to be used in the analysis. For example, for 2 surrogates, `Model = "Delta_T ~ Delta_S1 + Delta_S2")`.

`Show.Progress` Should progress of runs be graphically shown? (i.e., 1% done..., 2% done..., etc). Mainly useful when a large number of S have to be considered (to follow progress and estimate total run time).

Details

The multivariate ICA (R^2_H) is not identifiable because the individual causal treatment effects on T, S_1, ..., S_k cannot be observed. A simulation-based sensitivity analysis is therefore conducted in which the multivariate ICA (R^2_H) is estimated across a set of plausible values for the unidentifiable correlations. To this end, consider the variance covariance matrix of the potential outcomes $\mathbf{\Sigma}$ (0 and 1 subscripts refer to the control and experimental treatments, respectively):

$$\mathbf{\Sigma} = \left(\begin{array}{ccccccccc}
\sigma_{T_0T_0} & & & & & & & & \\
\sigma_{T_0T_1} & \sigma_{T_1T_1} & & & & & & & \\
\sigma_{T_0S1_0} & \sigma_{T_1S1_0} & \sigma_{S1_0S1_0} & & & & & & \\
\sigma_{T_0S1_1} & \sigma_{T_1S1_1} & \sigma_{S1_0S1_1} & \sigma_{S1_1S1_1} & & & & & \\
\sigma_{T_0S2_0} & \sigma_{T_1S2_0} & \sigma_{S1_0S2_0} & \sigma_{S1_1S2_0} & \sigma_{S2_0S2_0} & & & & \\
\sigma_{T_0S2_1} & \sigma_{T_1S2_1} & \sigma_{S1_0S2_1} & \sigma_{S1_1S2_1} & \sigma_{S2_0S2_1} & \sigma_{S2_1S2_1} & & & \\
... & ... & ... & ... & ... & ... & \ddots & & \\
\sigma_{T_0Sk_0} & \sigma_{T_1Sk_0} & \sigma_{S1_0Sk_0} & \sigma_{S1_1Sk_0} & \sigma_{S2_0Sk_0} & \sigma_{S2_1Sk_0} & \cdots & \sigma_{Sk_0Sk_0} & \\
\sigma_{T_0Sk_1} & \sigma_{T_1Sk_1} & \sigma_{S1_0Sk_1} & \sigma_{S1_1Sk_1} & \sigma_{S2_0Sk_1} & \sigma_{S2_1Sk_1} & \cdots & \sigma_{Sk_0Sk_1} & \sigma_{Sk_1Sk_1}.
\end{array} \right)$$

The ICA.ContCont.MultS_alt function requires the user to specify a distribution G for the unidentified correlations. Next, the identifiable correlations are fixed at their estimated values and the unidentifiable correlations are independently and randomly sampled from G. In the function call, the unidentifiable correlations are marked by specifying NA in the Sigma matrix (see example section below). The algorithm generates a large number of 'completed' matrices, and only those that are positive definite are retained (the number of positive definite matrices that should be obtained is specified by the M= argument in the function call). Based on the identifiable variances, these positive definite correlation matrices are converted to covariance matrices $\boldsymbol{\Sigma}$ and the multiple-surrogate ICA are estimated.

An issue with this approach (i.e., substituting unidentified correlations by random and independent samples from G) is that the probability of obtaining a positive definite matrix is very low when the dimensionality of the matrix increases. One approach to increase the efficiency of the algorithm is to build-up the correlation matrix in a gradual way. In particular, the property that a $(k \times k)$ matrix is positive definite if and only if all principal minors are positive (i.e., Sylvester's criterion) can be used. In other words, a $(k \times k)$ matrix is positive definite when the determinants of the upper-left (2×2), (3×3), ..., $(k \times k)$ submatrices all have a positive determinant. Thus, when a positive definite $(k \times k)$ matrix has to be generated, one can start with the upper-left (2×2) submatrix and randomly sample a value from the unidentified correlation (here: $\rho_{T_0 T_0}$) from G. When the determinant is positive (which will always be the case for a (2×2) matrix), the same procedure is used for the upper-left (3×3) submatrix, and so on. When a particular draw from G for a particular submatrix does not give a positive determinant, new values are sampled for the unidentified correlations until a positive determinant is obtained. In this way, it can be guaranteed that the final $(k \times k)$ submatrix will be positive definite. The latter approach is used in the current function. This procedure is used to generate many positive definite matrices. These positive definite matrices are used to generate M datasets which contain ΔT, ΔS_1, ΔS_2, ..., ΔS_k. Finally, the multivariate ICA (R_H^2) is estimated by regressing ΔT on ΔS_1, ΔS_2, ..., ΔS_k and computing the multiple coefficient of determination.

Value

An object of class ICA.ContCont.MultS_alt with components,

R2_H
: The multiple-surrogate individual causal association value(s).

Corr.R2_H
: The corrected multiple-surrogate individual causal association value(s).

Res_Err_Delta_T
: The residual errors (prediction errors) for intercept-only models of ΔT (i.e., models that do not include ΔS_1, ΔS_2, etc as predictors).

Res_Err_Delta_T_Given_S
: The residual errors (prediction errors) for models where ΔT is regressed on ΔS_1, ΔS_2, etc.

Lower.Dig.Corrs.All
: A data.frame that contains the matrix that contains the identifiable and unidentifiable correlations (lower diagonal elements) that were used to compute (R_H^2) in the run.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Van der Elst, W., Alonso, A. A., & Molenberghs, G. (2017). Univariate versus multivariate surrogate endpoints.

See Also

MICA.ContCont, ICA.ContCont, Single.Trial.RE.AA, plot Causal-Inference ContCont

Examples

```
## Not run:  #time-consuming code parts
# Specify matrix Sigma (var-cavar matrix T_0, T_1, S1_0, S1_1, ...)
# here for 1 true endpoint and 3 surrogates

s<-matrix(rep(NA, times=64),8)
s[1,1] <- 450; s[2,2] <- 413.5; s[3,3] <- 174.2; s[4,4] <- 157.5;
s[5,5] <- 244.0; s[6,6] <- 229.99; s[7,7] <- 294.2; s[8,8] <- 302.5
s[3,1] <- 160.8; s[5,1] <- 208.5; s[7,1] <- 268.4
s[4,2] <- 124.6; s[6,2] <- 212.3; s[8,2] <- 287.1
s[5,3] <- 160.3; s[7,3] <- 142.8
s[6,4] <- 134.3; s[8,4] <- 130.4
s[7,5] <- 209.3;
s[8,6] <- 214.7
s[upper.tri(s)] = t(s)[upper.tri(s)]

# Marix looks like (NA indicates unidentified covariances):
#            T_0    T_1   S1_0  S1_1  S2_0   S2_1  S2_0  S2_1
#            [,1]  [,2]  [,3]  [,4]  [,5]   [,6]  [,7]  [,8]
# T_0  [1,] 450.0    NA 160.8    NA 208.5     NA 268.4    NA
# T_1  [2,]    NA 413.5    NA 124.6    NA 212.30    NA 287.1
# S1_0 [3,] 160.8    NA 174.2    NA 160.3     NA 142.8    NA
# S1_1 [4,]    NA 124.6    NA 157.5    NA 134.30    NA 130.4
# S2_0 [5,] 208.5    NA 160.3    NA 244.0     NA 209.3    NA
# S2_1 [6,]    NA 212.3    NA 134.3    NA 229.99    NA 214.7
# S3_0 [7,] 268.4    NA 142.8    NA 209.3     NA 294.2    NA
# S3_1 [8,]    NA 287.1    NA 130.4    NA 214.70    NA 302.5

# Conduct analysis
ICA <- ICA.ContCont.MultS_alt(M=100, N=200, Show.Progress = TRUE,
  Sigma=s, G = seq(from=-1, to=1, by = .00001), Seed=c(123),
  Model = "Delta_T ~ Delta_S1 + Delta_S2 + Delta_S3")

# Explore results
summary(ICA)
plot(ICA)

## End(Not run)
```

`ICA.Sample.ContCont` *Assess surrogacy in the causal-inference single-trial setting (Individual Causal Association, ICA) in the Continuous-continuous case using the grid-based sample approach*

Description

The function `ICA.Sample.ContCont` quantifies surrogacy in the single-trial causal-inference framework. It provides a faster alternative for `ICA.ContCont`. See **Details** below.

Usage

```
ICA.Sample.ContCont(T0S0, T1S1, T0T0=1, T1T1=1, S0S0=1, S1S1=1, T0T1=seq(-1, 1, by=.001),
T0S1=seq(-1, 1, by=.001), T1S0=seq(-1, 1, by=.001), S0S1=seq(-1, 1, by=.001), M=50000)
```

Arguments

`T0S0`	A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the control treatment condition that should be considered in the computation of ρ_{Δ}.
`T1S1`	A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_{Δ}.
`T0T0`	A scalar that specifies the variance of the true endpoint in the control treatment condition that should be considered in the computation of ρ_{Δ}. Default 1.
`T1T1`	A scalar that specifies the variance of the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_{Δ}. Default 1.
`S0S0`	A scalar that specifies the variance of the surrogate endpoint in the control treatment condition that should be considered in the computation of ρ_{Δ}. Default 1.
`S1S1`	A scalar that specifies the variance of the surrogate endpoint in the experimental treatment condition that should be considered in the computation of ρ_{Δ}. Default 1.
`T0T1`	A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.001)`.
`T0S1`	A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.001)`.
`T1S0`	A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.001)`.
`S0S1`	A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.001)`.
`M`	The number of runs that should be conducted. Default `50000`.

Details

Based on the causal-inference framework, it is assumed that each subject *j* has four counterfactuals (or potential outcomes), i.e., T_{0j}, T_{1j}, S_{0j}, and S_{1j}. Let T_{0j} and T_{1j} denote the counterfactuals for the true endpoint (T) under the control ($Z = 0$) and the experimental ($Z = 1$) treatments of subject *j*, respectively. Similarly, S_{0j} and S_{1j} denote the corresponding counterfactuals for the surrogate endpoint (S) under the control and experimental treatments, respectively. The individual causal effects of Z on T and S for a given subject *j* are then defined as $\Delta_{T_j} = T_{1j} - T_{0j}$ and $\Delta_{S_j} = S_{1j} - S_{0j}$, respectively.

In the single-trial causal-inference framework, surrogacy can be quantified as the correlation between the individual causal effects of Z on S and T (for details, see Alonso et al., submitted):

$$\rho_{\Delta} = \rho(\Delta_{T_j}, \Delta_{S_j}) = \frac{\sqrt{\sigma_{S_0S_0}\sigma_{T_0T_0}}\rho_{S_0T_0} + \sqrt{\sigma_{S_1S_1}\sigma_{T_1T_1}}\rho_{S_1T_1} - \sqrt{\sigma_{S_0S_0}\sigma_{T_1T_1}}\rho_{S_0T_1} - \sqrt{\sigma_{S_1S_1}\sigma_{T_0T_0}}\rho_{S_1T_0}}{\sqrt{(\sigma_{T_0T_0} + \sigma_{T_1T_1} - 2\sqrt{\sigma_{T_0T_0}\sigma_{T_1T_1}}\rho_{T_0T_1})(\sigma_{S_0S_0} + \sigma_{S_1S_1} - 2\sqrt{\sigma_{S_0S_0}\sigma_{S_1S_1}}\rho_{S_0S_1})}},$$

where the correlations $\rho_{S_0T_1}$, $\rho_{S_1T_0}$, $\rho_{T_0T_1}$, and $\rho_{S_0S_1}$ are not estimable. It is thus warranted to conduct a sensitivity analysis.

The function `ICA.ContCont` constructs all possible matrices that can be formed based on the specified vectors for $\rho_{S_0T_1}$, $\rho_{S_1T_0}$, $\rho_{T_0T_1}$, and $\rho_{S_0S_1}$, and retains the positive definite ones for the computation of ρ_{Δ}.

In contrast, the function `ICA.ContCont` samples random values for $\rho_{S_0T_1}$, $\rho_{S_1T_0}$, $\rho_{T_0T_1}$, and $\rho_{S_0S_1}$ based on a uniform distribution with user-specified minimum and maximum values, and retains the positive definite ones for the computation of ρ_{Δ}.

The obtained vector of ρ_{Δ} values can subsequently be used to examine (i) the impact of different assumptions regarding the correlations between the counterfactuals on the results (see also `plot Causal-Inference ContCont`), and (ii) the extent to which proponents of the causal-inference and meta-analytic frameworks will reach the same conclusion with respect to the appropriateness of the candidate surrogate at hand.

The function `ICA.Sample.ContCont` also generates output that is useful to examine the plausibility of finding a good surrogate endpoint (see `GoodSurr` in the **Value** section below). For details, see Alonso et al. (submitted).

Notes

A single ρ_{Δ} value is obtained when all correlations in the function call are scalars.

Value

An object of class `ICA.ContCont` with components,

`Total.Num.Matrices`
An object of class `numeric` that contains the total number of matrices that can be formed as based on the user-specified correlations in the function call.

`Pos.Def` A `data.frame` that contains the positive definite matrices that can be formed based on the user-specified correlations. These matrices are used to compute the vector of the ρ_{Δ} values.

`ICA` A scalar or vector that contains the individual causal association (ICA; ρ_{Δ}) value(s).

`GoodSurr` A `data.frame` that contains the ICA (ρ_{Δ}), σ_{Δ_T}, and δ.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal-inference and meta-analytic paradigms for the validation of surrogate markers.

See Also

`MICA.ContCont`, `ICA.ContCont`, `Single.Trial.RE.AA`, `plot Causal-Inference ContCont`

Examples

```
# Generate the vector of ICA values when rho_T0S0=rho_T1S1=.95,
# sigma_T0T0=90, sigma_T1T1=100,sigma_ S0S0=10, sigma_S1S1=15, and
# min=-1 max=1 is considered for the correlations
# between the counterfactuals:
SurICA2 <- ICA.Sample.ContCont(T0S0=.95, T1S1=.95, T0T0=90, T1T1=100, S0S0=10,
S1S1=15, M=5000)

# Examine and plot the vector of generated ICA values:
summary(SurICA2)
plot(SurICA2)
```

ICA.Sample.ControlTreat

Assess surrogacy in the causal-inference single-trial setting (Individual Causal Association, ICA) in the Continuous-continuous case using the grid-based sample approach when data is only avalable for the control treatment

Description

The function `ICA.Sample.ControlTreat` quantifies surrogacy in the single-trial causal-inference framework when data is only avalable for the control treatment.

Usage

```
ICA.Sample.ControlTreat(T0S0, T1S1=seq(-1, 1, by = 0.001),
T0T0=1, T1T1=1, S0S0=1, S1S1=1, T0T1=seq(-1, 1, by=.001),
T0S1=seq(-1, 1, by=.001), T1S0=seq(-1, 1, by=.001), S0S1=seq(-1, 1, by=.001),
M=50000, M.Target=NA)
```

Arguments

`T0S0` A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the control treatment condition that should be considered in the computation of ρ_{Δ}.

`T1S1` A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_{Δ}.

`T0T0` A scalar or vector that specifies the variance of the true endpoint in the control treatment condition that should be considered in the computation of ρ_{Δ}. Default 1.

`T1T1` A scalar or vector that specifies the variance of the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_{Δ}. Default 1.

`S0S0` A scalar or vector that specifies the variance of the surrogate endpoint in the control treatment condition that should be considered in the computation of ρ_{Δ}. Default 1.

`S1S1` A scalar or vector that specifies the variance of the surrogate endpoint in the experimental treatment condition that should be considered in the computation of ρ_{Δ}. Default 1.

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.001)`.

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.001)`.

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.001)`.

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.001)`.

`M` The number of runs that should be conducted. Default `50000`.

`M.Target` The number of ICA values that should be identified. Only one argument `M=` or `M.Target=` can be used.

Value

An object of class `ICA.ContCont` with components,

`Total.Num.Matrices`
An object of class `numeric` that contains the total number of matrices that can be formed as based on the user-specified correlations in the function call.

`Pos.Def` A `data.frame` that contains the positive definite matrices that can be formed based on the user-specified correlations. These matrices are used to compute the vector of the ρ_{Δ} values.

ICA A scalar or vector that contains the individual causal association (ICA; ρ_Δ) value(s).

GoodSurr A data.frame that contains the ICA (ρ_Δ), σ_{Δ_T}, and δ.

Variances A data.frame that contains the variances for S and T in both treatment conditions.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Van der Elst, W. et al. (submitted). On the Early Identification of Promising Surrogate Endpoints Using Causal Inference.

See Also

MICA.ContCont, ICA.ContCont, Single.Trial.RE.AA, plot Causal-Inference ContCont

Examples

```
# Generate the vector of ICA values when rho_T0S0=.95,
# sigma_T0T0=90, sigma_T1T1=100,sigma_ S0S0=10, sigma_S1S1=15, and
# min=-1 max=1 is considered for the correlations
# between the counterfactuals and rho_T1S1:
SurICA2 <- ICA.Sample.ControlTreat(T0S0=.95, T0T0=90, T1T1=100, S0S0=10,
S1S1=15, M=5000)

# Examine and plot the vector of generated ICA values:
summary(SurICA2)
plot(SurICA2)
```

ICA_alpha_ContCont *Assess surrogacy using a Rényi divergence based family of metrics in the causal-inference single-trial setting in normal case*

Description

The function ICA_alpha_ContCont() is a set of metrics to evaluate surrogacy. ICA_alpha have the similar mathematical properties with ICA.ContCont().

Usage

```
ICA_alpha_ContCont(
  alpha = numeric(),
  T0S0,
  T1S1,
  T0T0 = 1,
  T1T1 = 1,
  S0S0 = 1,
  S1S1 = 1,
  T0T1 = seq(-1, 1, by = 0.1),
  T0S1 = seq(-1, 1, by = 0.1),
  T1S0 = seq(-1, 1, by = 0.1),
  S0S1 = seq(-1, 1, by = 0.1)
)
```

Arguments

`alpha` (numeric) is order `alpha in [0, infinity]`

`T0S0` A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the control treatment condition

`T1S1` A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the control treatment condition

`T0T0` A scalar that specifies the variance of the true endpoint in the control treatment condition

`T1T1` A scalar that specifies the variance of the true endpoint in the control treatment condition

`S0S0` A scalar that specifies the variance of the true endpoint in the control treatment condition

`S1S1` A scalar that specifies the variance of the true endpoint in the control treatment condition

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1

Value

- Total.Num.Matrices: An object of class numeric that contains the total number of matrices that can be formed as based on the user-specified correlations in the function call.
- Pos.Def: A data.frame that contains the positive definite matrices that can be formed based on the user-specified correlations. These matrices are used to compute the vector of the ρ_{Δ} values.
- rho: A scalar or vector that contains the individual causal association ρ_{Δ}
- ICA: A scalar or vector that contains the individual causal association $\rho_{\Delta}^2 = ICA$
- ICA_alpha: A scalar or vector that contains the individual causal association ICA_{α}
- Sigmas: A data.frame that contains the $\sigma_{\Delta T}$ and $\sigma_{\Delta S}$

ICA_given_model_constructor

Constructor for the function that returns that ICA as a function of the identifiable parameters

Description

ICA_given_model_constructor() returns a function fixes the unidentifiable parameters at user-specified values and takes the identifiable parameters as argument.

Usage

```
ICA_given_model_constructor(
  fitted_model,
  copula_par_unid,
  copula_family2,
  rotation_par_unid,
  n_prec,
  measure = "ICA",
  mutinfo_estimator = NULL,
  ICA_estimator = NULL,
  seed,
  composite = NULL,
  restr_time = +Inf
)
```

Arguments

fitted_model
: Returned value from fit_copula_OrdOrd(), fit_copula_OrdCont(), or fit_copula_ContCont(). This object contains the estimated identifiable part of the joint distribution for the potential outcomes.

copula_par_unid
: Parameter vector for the sequence of *unidentifiable* bivariate copulas that define the D-vine copula. The elements of copula_par correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

copula_family2
: Copula family of the other bivariate copulas. For the possible options, see loglik_copula_scale(). The elements of copula_family2 correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

rotation_par_unid
: Vector of rotation parameters for the sequence of *unidentifiable* bivariate copulas that define the D-vine copula. The elements of rotation_par correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

n_prec
: Number of Monte Carlo samples for the computation of the mutual information.

measure
: Compute intervals for which measure of surrogacy? Defaults to "ICA". See first column names of sens_results for other possibilities.

`mutinfo_estimator`
: Function that estimates the mutual information between the first two arguments which are numeric vectors. Defaults to `FNN::mutinfo()` with default arguments in the survival-survival setting. This argument is not used for non-survival-survival settings.

`ICA_estimator`
: Function that estimates the ICA between the first two arguments which are numeric vectors. Defaults to `NULL` which corresponds to using `estimate_ICA_ContCont()`, `estimate_ICA_OrdCont()`, or `estimate_ICA_OrdOrd()` (depending on the endpoint types). This argument is not used in the survival-survival setting.

`seed`
: Seed for Monte Carlo sampling. This seed does not affect the global environment.

`composite`
: (boolean) If `composite` is `TRUE`, then the surrogate endpoint is a composite of both a "pure" surrogate endpoint and the true endpoint, e.g., progression-free survival is the minimum of time-to-progression and time-to-death.

`restr_time`
: Restriction time for the potential outcomes. Defaults to `+Inf` which means no restriction. Otherwise, the sampled potential outcomes are replace by `pmin(S0, restr_time)` (and similarly for the other potential outcomes).

Value

A function that computes the ICA as a function of the identifiable parameters. In this computation, the unidentifiable parameters are fixed at the values supplied as arguments to `ICA_given_model_constructor_SurvSurv()` or `ICA_given_model_constructor()`.

ICA_given_model_constructor_SurvSurv

Constructor for the function that returns that ICA as a function of the identifiable parameters for survival-survival

---

Description

`ICA_given_model_constructor_SurvSurv()` returns a function fixes the unidentifiable parameters at user-specified values and takes the identifiable parameters as argument.

Usage

```
ICA_given_model_constructor_SurvSurv(
  fitted_model,
  copula_par_unid,
  copula_family2,
  rotation_par_unid,
  n_prec,
  measure = "ICA",
  mutinfo_estimator,
  composite,
  seed,
  restr_time = +Inf
)
```

Arguments

fitted_model
: Returned value from `fit_model_SurvSurv()`. This object contains the estimated identifiable part of the joint distribution for the potential outcomes.

copula_par_unid
: Parameter vector for the sequence of *unidentifiable* bivariate copulas that define the D-vine copula. The elements of `copula_par` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

copula_family2
: Copula family of the other bivariate copulas. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family2` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

rotation_par_unid
: Vector of rotation parameters for the sequence of *unidentifiable* bivariate copulas that define the D-vine copula. The elements of `rotation_par` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

n_prec
: Number of Monte Carlo samples for the computation of the mutual information.

measure
: Compute intervals for which measure of surrogacy? Defaults to `"ICA"`. See first column names of `sens_results` for other possibilities.

mutinfo_estimator
: Function that estimates the mutual information between the first two arguments which are numeric vectors. Defaults to `FNN::mutinfo()` with default arguments.

composite
: (boolean) If `composite` is `TRUE`, then the surrogate endpoint is a composite of both a "pure" surrogate endpoint and the true endpoint, e.g., progression-free survival is the minimum of time-to-progression and time-to-death.

seed
: Seed for Monte Carlo sampling. This seed does not affect the global environment.

restr_time
: Restriction time for the potential outcomes. Defaults to `+Inf` which means no restriction. Otherwise, the sampled potential outcomes are replace by `pmin(S0, restr_time)` (and similarly for the other potential outcomes).

Value

A function that computes the ICA as a function of the identifiable parameters. In this computation, the unidentifiable parameters are fixed at the values supplied as arguments to `ICA_given_model_constructor_SurvSurv()` or `ICA_given_model_constructor()`.

ICA_t

The function `ICA_t()` is to evaluate surrogacy in the single-trial causal-inference framework.

Description

The function `ICA_t()` is to evaluate surrogacy in the single-trial causal-inference framework.

Usage

```
ICA_t(
  df,
  T0S0,
  T1S1,
  T0T0 = 1,
  T1T1 = 1,
  S0S0 = 1,
  S1S1 = 1,
  T0T1 = seq(-1, 1, by = 0.1),
  T0S1 = seq(-1, 1, by = 0.1),
  T1S0 = seq(-1, 1, by = 0.1),
  S0S1 = seq(-1, 1, by = 0.1)
)
```

Arguments

`df` (numeric) is degree of freedom ν. The maximum value for *df* is 342. When *df* exceeds this threshold, the model behavior aligns with the Individual Causal Association (ICA) under the normal causal model.

`T0S0` A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the control treatment condition

`T1S1` A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the control treatment condition

`T0T0` A scalar that specifies the variance of the true endpoint in the control treatment condition

`T1T1` A scalar that specifies the variance of the true endpoint in the control treatment condition

`S0S0` A scalar that specifies the variance of the true endpoint in the control treatment condition

`S1S1` A scalar that specifies the variance of the true endpoint in the control treatment condition

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1

Value

- Total.Num.Matrices: An object of class numeric that contains the total number of matrices that can be formed as based on the user-specified correlations in the function call.

- Pos.Def: A data.frame that contains the positive definite matrices that can be formed based on the user-specified correlations. These matrices are used to compute the vector of the ρ_Δ values.
- rho: A scalar or vector that contains the individual causal association ρ_Δ
- ICA: A scalar or vector that contains the individual causal association $\rho_\Delta^2 = ICA$
- ICA_t: A scalar or vector that contains the individual causal association ICA_t
- Sigmas: A data.frame that contains the $\sigma_{\Delta T}$ and $\sigma_{\Delta S}$

ISTE.ContCont *Individual-level surrogate threshold effect for continuous normally distributed surrogate and true endpoints.*

Description

Computes the individual-level surrogate threshold effect in the causal-inference single-trial setting where both the surrogate and the true endpoint are continuous normally distributed variables. For details, see paper in the references section.

Usage

```
ISTE.ContCont(Mean_T1, Mean_T0, Mean_S1, Mean_S0, N, Delta_S=c(-10, 0, 10),
zeta.PI=0.05, PI.Bound=0, PI.Lower=TRUE, Show.Prediction.Plots=TRUE, Save.Plots="No",
T0S0, T1S1, T0T0=1, T1T1=1, S0S0=1, S1S1=1, T0T1=seq(-1, 1, by=.001),
T0S1=seq(-1, 1, by=.001), T1S0=seq(-1, 1, by=.001),
S0S1=seq(-1, 1, by=.001), M.PosDef=500, Seed=123)
```

Arguments

Mean_T1	A scalar or vector that specifies the mean of the true endpoint in the experimental treatment condition (a vector is used to account for estimation uncertainty).
Mean_T0	A scalar or vector that specifies the mean of the true endpoint in the control condition (a vector is used to account for estimation uncertainty).
Mean_S1	A scalar or vector that specifies the mean of the surrogate endpoint in the experimental treatment condition (a vector is used to account for estimation uncertainty).
Mean_S0	A scalar or vector that specifies the mean of the surrogate endpoint in the control condition (a vector is used to account for estimation uncertainty).
N	The sample size of the clinical trial.
Delta_S	The vector or scalar of ΔS values for which the expected ΔT and its prediction error has to be computed.
zeta.PI	The alpha-level to be used in the computation of the prediction interval around $E(\Delta T)$. Default `zeta.PI=0.05`, i.e., the 95% prediction interval.

`PI.Bound` The ISTE is defined as the value of ΔS for which the lower (or upper) bound of the $(1-\alpha)\%$ prediction interval around $E(\Delta T)$ is 0. If another threshold value than 0 is desired, this can be requested by using the `PI.Bound` argument. For example, the argument `PI.Bound=5` can be used in the function call to obtain the values of ΔS for which the lower (or upper) bound of the $(1-\alpha)\%$ prediction intervals (in the different runs of the algorithm)around ΔT equal 5.

`PI.Lower` Logical. Should a lower (`PI.Lower=TRUE`) or upper (`PI.Lower=FALSE`) prediction interval be used in the computation of ISTE? Default `PI.Lower=TRUE`.

`Show.Prediction.Plots`
Logical. Should plots that depict $E(\Delta T)$ against ΔS (prediction function), the prediction interval, and the ISTE for the different runs of the algorithm be shown? Default `Show.Prediction.Plots=TRUE`.

`Save.Plots` Should the prediction plots (see previous item) be saved? If `Save.Plots="No"` is used (the default argument), the plots are not saved. If the plots have to be saved, replace `"No"` by the desired location, e.g., `Save.Plots="C:/Analysis directory/"` on a windows computer or `Save.Plots="/Users/wim/Desktop/Analysis directory/"` on macOS or Linux.

`T0S0` A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the control treatment condition that should be considered in the computation of ISTE.

`T1S1` A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the experimental treatment condition that should be considered in the computation of ISTE.

`T0T0` A scalar that specifies the variance of the true endpoint in the control treatment condition that should be considered in the computation of ISTE. Default 1.

`T1T1` A scalar that specifies the variance of the true endpoint in the experimental treatment condition that should be considered in the computation of ISTE. Default 1.

`S0S0` A scalar that specifies the variance of the surrogate endpoint in the control treatment condition that should be considered in the computation of ISTE. Default 1.

`S1S1` A scalar that specifies the variance of the surrogate endpoint in the experimental treatment condition that should be considered in the computation of ISTE. Default 1.

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ISTE. Default `seq(-1, 1, by=.001)`.

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1 that should be considered in the computation of ISTE. Default `seq(-1, 1, by=.001)`.

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0 that should be considered in the computation of ISTE. Default `seq(-1, 1, by=.001)`.

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1 that should be considered in the computation of ISTE. Default `seq(-1, 1, by=.001)`.

M.PosDef
: The number of positive definite Σ matrices that should be identified. This will also determine the amount of ISTE values that are identified. Default `M.PosDef=500`.

Seed
: The seed to be used in the analysis (for reproducibility). Default `Seed=123`.

Details

See paper in the references section.

Value

An object of class `ICA.ContCont` with components,

`ISTE_Low_PI` The vector of individual surrogate threshold effect (ISTE) values, i.e., the values of ΔS for which the lower bound of the $(1-\alpha)\%$ prediction interval around ΔT is 0 (or another threshold value, which can be requested by using the `PI.Bound` argument in the function call).

`ISTE_Up_PI` Same as `ISTE_Low_PI`, but using the upper bound of the $(1-\alpha)\%$ prediction interval.

`MSE` The vector of mean squared error values that are obtained in the prediction of ΔT based on ΔS.

`gamma0` The vector of intercepts that are obtained in the prediction of ΔT based on ΔS.

`gamma1` The vector of slope that are obtained in the prediction of ΔT based on ΔS.

`Delta_S_For_Which_Delta_T_equal_0` The vector of ΔS values for which $E(\Delta T = 0)$.

`S_squared_pred` The vector of variances of the prediction errors for ΔT.

`Predicted_Delta_T` The vector/matrix of predicted values of ΔT for the ΔS values that were requested in the function call (argument `Delta_S`).

`PI_Interval_Low` The vector/matrix of lower bound values of the $(1-\alpha)\%$ prediction interval around ΔT for the ΔS values that were requested in the function call (argument `Delta_S`).

`PI_Interval_Up` The vector/matrix of upper bound values of the $(1-\alpha)\%$ prediction interval around ΔT for the ΔS values that were requested in the function call (argument `Delta_S`).

`T0T0` The vector of variances of T0 (true endpoint in the control treatment) that are used in the computation (this is a constant if the variance is fixed in the function call).

`T1T1` The vector of variances of T1 (true endpoint in the experimental treatment) that are used in the computations (this is a constant if the variance is fixed in the function call).

`S0S0` The vector of variances of S0 (surrogate endpoint in the control treatment) that are used in the computations (this is a constant if the variance is fixed in the function call).

`S1S1` The vector of variances of S1 (surrogate endpoint in the experimental treatment) that are used in the computations (this is a constant if the variance is fixed in the function call).

`Mean_DeltaT` The vector of treatment effect values on the true endpoint that are used in the computations (this is a constant if the means of T0 and T1 are fixed in the function call).

`Mean_DeltaS` The vector of treatment effect values on the surrogate endpoint that are used in the computations (this is a constant if the means of S0 and S1 are fixed in the function call).

`Total.Num.Matrices` An object of class `numeric` that contains the total number of matrices that can be formed as based on the user-specified correlations in the function call.

`Pos.Def` A `data.frame` that contains the positive definite matrices that can be formed based on the user-specified correlations. These matrices are used to compute the vector of the ISTE values.

`ICA` Apart from ISTE, ICA is also computed (the individual causal association). For details, see `ICA.ContCont`.

`zeta.PI` The `zeta.PI` value specified in the function call.

`PI.Bound` The `PI.Bound` value specified in the function call.

`PI.Lower` The `PI.Lower` value specified in the function call.

`Delta_S` The `Delta_S` value(s) specified in the function call.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Van der Elst, W., Alonso, A. A., and Molenberghs, G. (submitted). The individual-level surrogate threshold effect in a causal-inference setting.

See Also

`ICA.ContCont`

Examples

```
# Define input for analysis using the Schizo dataset,
# with S=BPRS and T = PANSS.
# For each of the identifiable quantities,
# uncertainty is accounted for by specifying a uniform
# distribution with min, max values corresponding to
# the 95% confidence interval of the quantity.
T0S0 <- runif(min = 0.9524, max = 0.9659, n = 1000)
T1S1 <- runif(min = 0.9608, max = 0.9677, n = 1000)

S0S0 <- runif(min=160.811, max=204.5009, n=1000)
S1S1 <- runif(min=168.989, max = 194.219, n=1000)
```

```
T0T0 <- runif(min=484.462, max = 616.082, n=1000)
T1T1 <- runif(min=514.279, max = 591.062, n=1000)

Mean_T0 <- runif(min=-13.455, max=-9.489, n=1000)
Mean_T1 <- runif(min=-17.17, max=-14.86, n=1000)
Mean_S0 <- runif(min=-7.789, max=-5.503, n=1000)
Mean_S1 <- runif(min=-9.600, max=-8.276, n=1000)

# Do the ISTE analysis
## Not run:
ISTE <- ISTE.ContCont(Mean_T1=Mean_T1, Mean_T0=Mean_T0,
 Mean_S1=Mean_S1, Mean_S0=Mean_S0, N=2128, Delta_S=c(-50:50),
 zeta.PI=0.05, PI.Bound=0, Show.Prediction.Plots=TRUE,
 Save.Plots="No", T0S0=T0S0, T1S1=T1S1, T0T0=T0T0, T1T1=T1T1,
 S0S0=S0S0, S1S1=S1S1)

# Examine results:
summary(ISTE)

# Plots of results.
  # Plot ISTE
plot(ISTE)
  # Other plots, see plot.ISTE.ContCont for details
plot(ISTE, Outcome="MSE")
plot(ISTE, Outcome="gamma0")
plot(ISTE, Outcome="gamma1")
plot(ISTE, Outcome="Exp.DeltaT")
plot(ISTE, Outcome="Exp.DeltaT.Low.PI")
plot(ISTE, Outcome="Exp.DeltaT.Up.PI")

## End(Not run)
```

loglik_copula_scale *Loglikelihood on the Copula Scale*

Description

`loglik_copula_scale()` computes the loglikelihood on the copula scale for possibly right-censored data.

Usage

```
loglik_copula_scale(
  theta,
  u,
  v,
  d1,
  d2,
  copula_family,
  r = 0L,
  return_sum = TRUE
)
```

Arguments

theta Copula parameter

u A numeric vector. Corresponds to first variable on the copula scale.

v A numeric vector. Corresponds to second variable on the copula scale.

d1 An integer vector. Indicates whether first variable is observed or right-censored,

- d1[i] = 1 if u[i] corresponds to non-censored value
- d1[i] = 0 if u[i] corresponds to right-censored value
- d1[i] = -1 if u[i] corresponds to left-censored value

d2 An integer vector. Indicates whether first variable is observed or right-censored,

- d2[i] = 1 if v[i] corresponds to non-censored value
- d2[i] = 0 if v[i] corresponds to right-censored value
- d2[i] = -1 if v[i] corresponds to left-censored value

copula_family Copula family, one of the following:

- "clayton"
- "frank"
- "gumbel"
- "gaussian"

r rotation parameter. Should be 0L, 90L, 180L, or 270L.

The parameterization of the respective copula families can be found in the help files of the dedicated functions named copula_loglik_copula_scale().

return_sum Return the sum of the individual loglikelihoods? If FALSE, a vector with the individual loglikelihood contributions is returned.

Value

Value of the copula loglikelihood evaluated in theta.

---

log_likelihood_copula_model

Computes loglikelihood for a given copula model

Description

log_likelihood_copula_model() computes the loglikelihood for a given bivariate copula model and data set while allowin for right-censoring of both outcome variables.

Usage

```
log_likelihood_copula_model(
  theta,
  X,
  Y,
  d1,
  d2,
  copula_family,
  cdf_X,
  cdf_Y,
  pdf_X,
  pdf_Y,
  return_sum = TRUE
)
```

Arguments

theta
: Copula parameter

X
: Numeric vector corresponding to first outcome variable.

Y
: Numeric vector corresponding to second outcome variable.

d1
: An integer vector. Indicates whether first variable is observed or right-censored,

- `d1[i] = 1` if `u[i]` corresponds to non-censored value
- `d1[i] = 0` if `u[i]` corresponds to right-censored value
- `d1[i] = -1` if `u[i]` corresponds to left-censored value

d2
: An integer vector. Indicates whether first variable is observed or right-censored,

- `d2[i] = 1` if `v[i]` corresponds to non-censored value
- `d2[i] = 0` if `v[i]` corresponds to right-censored value
- `d2[i] = -1` if `v[i]` corresponds to left-censored value

copula_family
: Copula family, one of the following:

- `"clayton"`
- `"frank"`
- `"gumbel"`
- `"gaussian"`

cdf_X
: Distribution function for the first outcome variable.

cdf_Y
: Distribution function for the second outcome variable.

pdf_X
: Density function for the first outcome variable.

pdf_Y
: Density function for the second outcome variable.

return_sum
: Return the sum of the individual loglikelihoods? If `FALSE`, a vector with the individual loglikelihood contributions is returned.

Value

Loglikelihood of the bivariate copula model evaluated in the observed data.

LongToWide *Reshapes a dataset from the 'long' format (i.e., multiple lines per patient) into the 'wide' format (i.e., one line per patient)*

Description

Reshapes a dataset that is in the 'long' format into the 'wide' format. The dataset should contain a single surrogate endpoint and a single true endpoint value per subject.

Usage

```
LongToWide(Dataset, OutcomeIndicator, IdIndicator, TreatIndicator, OutcomeValue)
```

Arguments

Dataset	A data.frame in the 'long' format that contains (at least) five columns, i.e., one that contains the subject ID, one that contains the trial ID, one that contains the endpoint indicator, one that contains the treatment indicator, and one that contains the endpoint values.
OutcomeIndicator	The name of the variable in Dataset that contains the indicator that distinguishes between the surrogate and true endpoints.
IdIndicator	The name of the variable in Dataset that contains the subject ID.
TreatIndicator	The name of the variable in Dataset that contains the treatment indicator. For the subsequent surrogacy analyses, the treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group. The $-1/1$ coding is recommended.
OutcomeValue	The name of the variable in Dataset that contains the endpoint values.

Value

A data.frame in the 'wide' format, i.e., a data.frame that contains one line per subject. Each line contains a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.

Author(s)

Wim Van der Elst, Ariel Alonso, and Geert Molenberghs

Examples

```
# Generate a dataset in the 'long' format that contains
# S and T values for 100 patients
Outcome <- rep(x=c(0, 1), times=100)
ID <- rep(seq(1:100), each=2)
Treat <- rep(seq(c(0,1)), each=100)
Outcomes <- as.numeric(matrix(rnorm(1*200, mean=100, sd=10),
                                      ncol=200))
Data <- data.frame(cbind(Outcome, ID, Treat, Outcomes))

# Reshapes the Data object
LongToWide(Dataset=Data, OutcomeIndicator=Outcome, IdIndicator=ID,
           TreatIndicator=Treat, OutcomeValue=Outcomes)
```

MarginalProbs *Computes marginal probabilities for a dataset where the surrogate and true endpoints are binary*

Description

This function computes the marginal probabilities associated with the distribution of the potential outcomes for the true and surrogate endpoint.

Usage

```
MarginalProbs(Dataset=Dataset, Surr=Surr, True=True, Treat=Treat)
```

Arguments

`Dataset` A `data.frame` that should consist of one line per patient. Each line contains (at least) a binary surrogate value, a binary true endpoint value, and a treatment indicator.

`Surr` The name of the variable in `Dataset` that contains the binary surrogate endpoint values. Should be coded as 0 and 1.

`True` The name of the variable in `Dataset` that contains the binary true endpoint values. Should be coded as 0 and 1.

`Treat` The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should be coded as 1 for the experimental group and -1 for the control group.

Value

`Theta_T0S0` The odds ratio for S and T in the control group.

`Theta_T1S1` The odds ratio for S and T in the experimental group.

`Freq.Cont` The frequencies for S and T in the control group.

`Freq.Exp` The frequencies for S and T in the experimental group.

`pi1_1_` The estimated $\pi_{1\cdot 1\cdot}$.

`pi0_1_` The estimated $\pi_{0\cdot 1\cdot}$.

`pi1_0_` The estimated $\pi_{1\cdot 0\cdot}$.

`pi0_0_` The estimated $\pi_{0\cdot 0\cdot}$.

`pi_1_1` The estimated $\pi_{\cdot 1\cdot 1}$

`pi_1_0` The estimated $\pi_{\cdot 1\cdot 0}$

`pi_0_1` The estimated $\pi_{\cdot 0\cdot 1}$

`pi_0_0` The estimated $\pi_{\cdot 0\cdot 0}$

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

See Also

ICA.BinBin

Examples

```
# Open the ARMD dataset and recode Diff24 and Diff52 as 1
# when the original value is above 0, and 0 otherwise
data(ARMD)
ARMD$Diff24_Dich <- ifelse(ARMD$Diff24>0, 1, 0)
ARMD$Diff52_Dich <- ifelse(ARMD$Diff52>0, 1, 0)

# Obtain marginal probabilities and ORs
MarginalProbs(Dataset=ARMD, Surr=Diff24_Dich, True=Diff52_Dich,
Treat=Treat)
```

---

marginal_distribution *Fit marginal distribution*

Description

The `marginal_distribution()` function is a wrapper for `fitdistrplus::fitdist()` that fits a univariate distribution to a data vector.

Usage

```
marginal_distribution(x, distribution, fix.arg = NULL)
```

Arguments

`x` (numeric) data vector

`distribution` Distributional family. One of the follwing:

- `"normal"`: normal distribution
- `"logistic`: logistic distribution as parameterized in `dlogis()`
- `"t"`: student t distribution is parameterized in `dt()`
- `"lognormal"`: lognormal distribution as parameterized in `dlnorm()`
- `"gamma"`: gamma distribution as parameterized in `dgamma()`
- `"weibull"`: weibull distribution as parameterized in `dweibull()`

`fix.arg` An optional named list giving the values of fixed parameters of the named distribution or a function of data computing (fixed) parameter values and returning a named list. Parameters with fixed value are thus NOT estimated by this maximum likelihood procedure.

Value

Object of class `fitdistrplus::fitdist` that represents the marginal surrogate distribution.

`marginal_gof_copula` *Produce marginal GoF plot*

Description

Produce marginal GoF plot

Usage

```
marginal_gof_copula(
  marginal,
  observed,
  name,
  type,
  treat,
  return_data = FALSE,
  grid = NULL,
  ...
)
```

Arguments

`marginal` Estimated marginal distribution represented by a list with three elements in the following order: the estimated cdf, pdf, and inverse cdf.

`observed` Observed values. These are used for the histogram.

`name` Name of the endpoint (used in the plot title).

`type` Type of endpoint: `"ordinal"` or `"continuous"`

`treat` Value for the treatment indicator.

`return_data` (boolean) Return the data used in the goodness-of-fit plot (without the plot itself). This is useful when the user wants to customize the plots, e.g., using `ggplot2`. See Details.

`grid` (numeric) vector of values for the endpoint at which the model-based density is computed.

`...` Extra arguments passed onto `plot()` or `hist()` for an ordinal and continuous endpoint, respectively.

Return Plotting Data

If return_data is TRUE, this function will return a data frame that can be used to create customized plots. The following variables are present in the returned data frame:

- observed: The empirical proportions (type = "ordinal"). NA for type = "continuous".
- upper_ci, lower_ci: Upper limit of the 95% confidence interval for the empirical proportions. Defaults to NA if type = "continuous".
- value: Value for the continuous or ordinal variable.
- model_based: Estimated model-based density (type = "continuous") or proportions (type = "ordinal")

See Also

plot.vine_copula_fit()

marginal_gof_plots_scr

Marginal survival function goodness of fit

Description

The marginal_gof_plots_scr() function plots the estimated marginal survival functions for the fitted model. This results in four plots of survival functions, one for each of S_0, S_1, T_0, T_1.

Usage

```
marginal_gof_plots_scr(fitted_model, grid)
```

Arguments

fitted_model	Returned value from fit_model_SurvSurv(). This object essentially contains the estimated identifiable part of the joint distribution for the potential outcomes.
grid	grid of time-points for which to compute the estimated survival functions.

Examples

```
data("Ovarian")
#For simplicity, data is not recoded to semi-competing risks format, but is
#left in the composite event format.
data = data.frame(
  Ovarian$Pfs,
  Ovarian$Surv,
  Ovarian$Treat,
  Ovarian$PfsInd,
  Ovarian$SurvInd
)
ovarian_fitted =
  fit_model_SurvSurv(data = data,
                     copula_family = "clayton",
                     n_knots = 1)
grid = seq(from = 0, to = 2, length.out = 50)
Surrogate:::marginal_gof_plots_scr(ovarian_fitted, grid)
```

---

marginal_gof_scr_S_plot

Goodness-of-fit plot for the marginal survival functions

Description

The `marginal_gof_scr_S_plot()` and `marginal_gof_scr_T_plot()` functions plot the estimated marginal survival functions for the surrogate and true endpoints. In these plots, it is assumed that the copula model has been fitted for $(T_0, \tilde{S}_0, \tilde{S}_1, T_1)'$ where

$$S_k = \min(\tilde{S}_k, T_k)$$

is the (composite) surrogate of interest. In these plots, the model-based survival functions for $(T_0, S_0, S_1, T_1)'$ are plotted together with the corresponding Kaplan-Meier etimates.

Usage

```
marginal_gof_scr_S_plot(fitted_model, grid, treated, ...)

marginal_gof_scr_T_plot(fitted_model, grid, treated, ...)
```

Arguments

`fitted_model`	Returned value from `fit_model_SurvSurv()`. This object essentially contains the estimated identifiable part of the joint distribution for the potential outcomes.
`grid`	Grid of time-points at which the model-based estimated regression functions, survival functions, or probabilities are evaluated.
`treated`	(numeric) Treatment group. Should be `0` or `1`.
`...`	Additional arguments to pass to `plot()`.

Value

`NULL`

True Endpoint

The marginal goodness-of-fit plots for the true endpoint, build by `marginal_gof_scr_T_plot()`, is simply a comparison of the model-based estimate of $P(T_k > t)$ with the Kaplan-Meier (KM) estimate obtained with `survival::survfit()`. A pointwise 95% confidence interval for the KM estimate is also plotted.

Surrogate Endpoint

The model-based estimate of $P(S_k > s)$ follows indirectly from the fitted copula model because the copula model has been fitted for $\tilde{S}_k$ instead of S_k. However, the model-based estimate still follows easily from the copula model as follows,

$$P(S_k > s) = P(\min(\tilde{S}_k, T_k)) = P(\tilde{S}_k > s, T_k > s).$$

The `marginal_gof_scr_T_plot()` function plots the model-based estimate for $P(\tilde{S}_k > s, T_k > s)$ together with the KM estimate (see above).

Examples

```
# Load Ovarian data
data("Ovarian")
# Recode the Ovarian data in the semi-competing risks format.
data_scr = data.frame(
  ttp = Ovarian$Pfs,
  os = Ovarian$Surv,
  treat = Ovarian$Treat,
  ttp_ind = ifelse(
    Ovarian$Pfs == Ovarian$Surv &
      Ovarian$SurvInd == 1,
    0,
    Ovarian$PfsInd
  ),
  os_ind = Ovarian$SurvInd
)
# Fit copula model.
fitted_model = fit_model_SurvSurv(data = data_scr,
                                  copula_family = "clayton",
                                  n_knots = 1)
# Define grid for GoF plots.
grid = seq(from = 1e-3,
           to = 2.5,
           length.out = 30)
# Assess marginal goodness-of-fit in the control group.
marginal_gof_scr_S_plot(fitted_model, grid = grid, treated = 0)
marginal_gof_scr_T_plot(fitted_model, grid = grid, treated = 0)
# Assess goodness-of-fit of the association structure, i.e., the copula.
prob_dying_without_progression_plot(fitted_model, grid = grid, treated = 0)
mean_S_before_T_plot_scr(fitted_model, grid = grid, treated = 0)
```

MaxEntContCont *Use the maximum-entropy approach to compute ICA in the continuous-continuous sinlge-trial setting*

Description

In a surrogate evaluation setting where both S and T are continuous endpoints, a sensitivity-based approach where multiple 'plausible values' for ICA are retained can be used (see functions ICA.ContCont). The function MaxEntContCont identifies the estimate which has the maximuum entropy.

Usage

```
MaxEntContCont(x, T0T0, T1T1, S0S0, S1S1)
```

Arguments

x A fitted object of class ICA.ContCont.

T0T0 A scalar that specifies the variance of the true endpoint in the control treatment condition.

T1T1 A scalar that specifies the variance of the true endpoint in the experimental treatment condition.

S0S0 A scalar that specifies the variance of the surrogate endpoint in the control treatment condition.

S1S1 A scalar that specifies the variance of the surrogate endpoint in the experimental treatment condition.

Value

ICA.Max.Ent The ICA value with maximum entropy.

Max.Ent The maximum entropy.

Entropy The vector of entropies corresponding to the vector of 'plausible values' for ICA.

Table.ICA.Entropy
A data.frame that contains the vector of ICA, their entropies, and the correlations between the counterfactuals.

ICA.Fit The fitted ICA.ContCont object.

Author(s)

Wim Van der Elst, Ariel Alonso, Paul Meyvisch, & Geert Molenberghs

References

Add

See Also

ICA.ContCont, MaxEntICABinBin

Examples

```
## Not run:  #time-consuming code parts
# Compute ICA for ARMD dataset, using the grid
# G={-1, -.80, ..., 1} for the undidentifiable correlations

ICA <- ICA.ContCont(T0S0 = 0.769, T1S1 = 0.712, S0S0 = 188.926,
S1S1 = 132.638, T0T0 = 264.797, T1T1 = 231.771,
T0T1 = seq(-1, 1, by = 0.2), T0S1 = seq(-1, 1, by = 0.2),
T1S0 = seq(-1, 1, by = 0.2), S0S1 = seq(-1, 1, by = 0.2))

# Identify the maximum entropy ICA
MaxEnt_ARMD <- MaxEntContCont(x = ICA, S0S0 = 188.926,
S1S1 = 132.638, T0T0 = 264.797, T1T1 = 231.771)

  # Explore results using summary() and plot() functions
summary(MaxEnt_ARMD)
plot(MaxEnt_ARMD)
plot(MaxEnt_ARMD, Entropy.By.ICA = TRUE)

## End(Not run)
```

MaxEntICABinBin *Use the maximum-entropy approach to compute ICA in the binary-binary setting*

Description

In a surrogate evaluation setting where both S and T are binary endpoints, a sensitivity-based approach where multiple 'plausible values' for ICA are retained can be used (see functions ICA.BinBin, ICA.BinBin.Grid.Full, or ICA.BinBin.Grid.Sample). Alternatively, the maximum entropy distribution of the vector of potential outcomes can be considered, based upon which ICA is subsequently computed. The use of the distribution that maximizes the entropy can be justified based on the fact that any other distribution would necessarily (i) assume information that we do not have, or (ii) contradict information that we do have. The function MaxEntICABinBin implements the latter approach.

Usage

```
MaxEntICABinBin(pi1_1_, pi1_0_, pi_1_1,
pi_1_0, pi0_1_, pi_0_1, Method="BFGS",
Fitted.ICA=NULL)
```

Arguments

pi1_1_ A scalar that contains the estimated value for $P(T = 1, S = 1|Z = 0)$, i.e., the probability that $S = T = 1$ when under treatment $Z = 0$.

pi1_0_ A scalar that contains the estimated value for $P(T = 1, S = 0|Z = 0)$.

pi_1_1 A scalar that contains the estimated value for $P(T = 1, S = 1|Z = 1)$.

`pi_1_0` A scalar that contains the estimated value for $P(T = 1, S = 0|Z = 1)$.

`pi0_1_` A scalar that contains the estimated value for $P(T = 0, S = 1|Z = 0)$.

`pi_0_1` A scalar that contains the estimated value for $P(T = 0, S = 1|Z = 1)$.

`Method` The maximum entropy frequency vector p^* is calculated based on the optimal solution to an unconstrained dual convex programming problem (for details, see Alonso et al., 2015). Two different optimization methods can be specified, i.e., `Method="BFGS"` and `Method="CG"`, which implement the quasi-Newton BFGS (Broyden, Fletcher, Goldfarb, and Shanno) and the conjugent gradient (CG) methods (for details on these methods, see the help files of the `optim()` function and the references theirin). Alternatively, the π vector (obtained when the functions `ICA.BinBin`, `ICA.BinBin.Grid.Full`, or `ICA.BinBin.Grid.Sample` are executed) that is 'closest' to the vector π can be retained. Here, the 'closest' vector is defined as the vector where the sum of the squared differences between the components in the vectors π and π is smallest. The latter 'Minimum Difference' method can re requested by specifying the argument `Method="MD"` in the function call. Default `Method="BFGS"`.

`Fitted.ICA` A fitted object of class `ICA.BinBin`, `ICA.BinBin.Grid.Full`, or `ICA.BinBin.Grid.Sample`. Only required when `Method="MD"` is used.

Value

`R2_H` The R2_H value.

`Vector_p` The maximum entropy frequency vector p^*

`H_max` The entropy of p^*

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., & Van der Elst, W. (2015). A maximum-entropy approach for the evluation of surrogate endpoints based on causal inference.

See Also

`ICA.BinBin`, `ICA.BinBin.Grid.Sample`, `ICA.BinBin.Grid.Full`, `plot MaxEntICA BinBin`

Examples

```
# Sensitivity-based ICA results using ICA.BinBin.Grid.Sample
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078, Seed=1,
Monotonicity=c("No"), M=5000)

# Maximum-entropy based ICA
MaxEnt <- MaxEntICABinBin(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078)
```

```
# Explore maximum-entropy results
summary(MaxEnt)

# Plot results
plot(x=MaxEnt, ICA.Fit=ICA)
```

MaxEntSPFBinBin *Use the maximum-entropy approach to compute SPF (surrogate predictive function) in the binary-binary setting*

Description

In a surrogate evaluation setting where both S and T are binary endpoints, a sensitivity-based approach where multiple 'plausible values' for vector π (i.e., vectors π that are compatible with the observable data at hand) can be used (for details, see SPF.BinBin). Alternatively, the maximum entropy distribution for vector π can be considered (Alonso et al., 2015). The use of the distribution that maximizes the entropy can be justified based on the fact that any other distribution would necessarily (i) assume information that we do not have, or (ii) contradict information that we do have. The function MaxEntSPFBinBin implements the latter approach.

Based on vector π, the surrogate predictive function (SPF) is computed, i.e., $r(i,j) = P(\Delta T = i|\Delta S = j)$. For example, $r(-1,1)$ quantifies the probability that the treatment has a negative effect on the true endpoint ($\Delta T = -1$) given that it has a positive effect on the surrogate ($\Delta S = 1$).

Usage

```
MaxEntSPFBinBin(pi1_1_, pi1_0_, pi_1_1,
pi_1_0, pi0_1_, pi_0_1, Method="BFGS",
Fitted.ICA=NULL)
```

Arguments

pi1_1_ A scalar that contains the estimated value for $P(T = 1, S = 1|Z = 0)$, i.e., the probability that $S = T = 1$ when under treatment $Z = 0$.

pi1_0_ A scalar that contains the estimated value for $P(T = 1, S = 0|Z = 0)$.

pi_1_1 A scalar that contains the estimated value for $P(T = 1, S = 1|Z = 1)$.

pi_1_0 A scalar that contains the estimated value for $P(T = 1, S = 0|Z = 1)$.

pi0_1_ A scalar that contains the estimated value for $P(T = 0, S = 1|Z = 0)$.

pi_0_1 A scalar that contains the estimated value for $P(T = 0, S = 1|Z = 1)$.

Method The maximum entropy frequency vector p^* is calculated based on the optimal solution to an unconstrained dual convex programming problem (for details, see Alonso et al., 2015). Two different optimization methods can be specified, i.e., Method="BFGS" and Method="CG", which implement the quasi-Newton BFGS (Broyden, Fletcher, Goldfarb, and Shanno) and the conjugent gradient (CG) methods (for details on these methods, see the help files of the optim() function and the references theirin). Alternatively, the π vector (obtained when the functions `ICA.BinBin`, `ICA.BinBin.Grid.Full`, or `ICA.BinBin.Grid.Sample` are executed) that is 'closest' to the vector π can be retained. Here, the 'closest' vector is defined as the vector where the sum of the squared differences between the components in the vectors π and π is smallest. The latter 'Minimum Difference' method can re requested by specifying the argument `Method="MD"` in the function call. Default `Method="BFGS"`.

`Fitted.ICA` A fitted object of class `ICA.BinBin`, `ICA.BinBin.Grid.Full`, or `ICA.BinBin.Grid.Sample`. Only required when `Method="MD"` is used.

Value

`Vector_p`	The maximum entropy frequency vector p^*
`r_1_1`	The vector of values for $r(1,1)$, i.e., $P(\Delta T = 1 \vert \Delta S = 1)$.
`r_min1_1`	The vector of values for $r(-1,1)$.
`r_0_1`	The vector of values for $r(0,1)$.
`r_1_0`	The vector of values for $r(1,0)$.
`r_min1_0`	The vector of values for $r(-1,0)$.
`r_0_0`	The vector of values for $r(0,0)$.
`r_1_min1`	The vector of values for $r(1,-1)$.
`r_min1_min1`	The vector of values for $r(-1,-1)$.
`r_0_min1`	The vector of values for $r(0,-1)$.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., & Van der Elst, W. (2015). A maximum-entropy approach for the evluation of surrogate endpoints based on causal inference.

See Also

`ICA.BinBin`, `ICA.BinBin.Grid.Sample`, `ICA.BinBin.Grid.Full`, `plot MaxEntSPF BinBin`

Examples

```
# Sensitivity-based ICA results using ICA.BinBin.Grid.Sample
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078, Seed=1,
Monotonicity=c("No"), M=5000)

# Sensitivity-based SPF
SPFSens <- SPF.BinBin(ICA)

# Maximum-entropy based SPF
```

```
SPFMaxEnt <- MaxEntSPFBinBin(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078)

# Explore maximum-entropy results
summary(SPFMaxEnt)

# Plot results
plot(x=SPFMaxEnt, SPF.Fit=SPFSens)
```

mean_S_before_T_plot_scr

Goodness of fit plot for the fitted copula

Description

The mean_S_before_T_plot_scr() and prob_dying_without_progression_plot() functions build plots to assess the goodness-of-fit of the copula model fitted by fit_model_SurvSurv(). Specifically, these two functions focus on the appropriateness of the copula. Note that to assess the appropriateness of the marginal functions, two other functions are available: marginal_gof_scr_S_plot() and marginal_gof_scr_T_plot().

Usage

```
mean_S_before_T_plot_scr(fitted_model, plot_method = NULL, grid, treated, ...)

prob_dying_without_progression_plot(
  fitted_model,
  plot_method = NULL,
  grid,
  treated,
  ...
)
```

Arguments

fitted_model	Returned value from fit_model_SurvSurv(). This object essentially contains the estimated identifiable part of the joint distribution for the potential outcomes.
plot_method	Defaults to NULL. Should not be modified.
grid	Grid of time-points at which the model-based estimated regression functions, survival functions, or probabilities are evaluated.
treated	(numeric) Treatment group. Should be 0 or 1.
...	Additional arguments to pass to plot().

Value

NULL

Progression Before Death

If a patient progresses before death, this means that $S_k < T_k$. For these patients, we can look at the expected progression time given that the patient has died at $T_k = t$:

$$E(S_k | T_k = t, S_k < T_k).$$

The `mean_S_before_T_plot_scr()` function plots the model-based estimate of this regression function together with a non-parametric estimate.

This regression function can also be estimated non-parametrically by regressing S_k onto T_k in the subset of uncensored patients. This non-parametric estimate is obtained via `mgcv::gam(y~s(x))` with additionally `family = stats::quasi(link = "log", variance = "mu")` because this tends to describe survival data better. The 95% confidence intervals are added for this non-parametric estimate; although, they should be interpreted with caution because the Poisson mean-variance relation may be wrong.

Death Before Progression

If a patient dies before progressing, this means that $S_k = T_k$. This probability can be modeled as a function of time, i.e.,

$$\pi_k(t) = P(S_k = t \,|\, T_k = t).$$

The `prob_dying_without_progression_plot()` function plots the model-based estimate of this regression function together with a non-parametric estimate.

This regression function can also be estimated non-parametrically by regressing the censoring indicator for S_k, δ_{S_k}, onto T_k in the subset of patients with uncensored T_k.

Examples

```
# Load Ovarian data
data("Ovarian")
# Recode the Ovarian data in the semi-competing risks format.
data_scr = data.frame(
  ttp = Ovarian$Pfs,
  os = Ovarian$Surv,
  treat = Ovarian$Treat,
  ttp_ind = ifelse(
    Ovarian$Pfs == Ovarian$Surv &
      Ovarian$SurvInd == 1,
    0,
    Ovarian$PfsInd
  ),
  os_ind = Ovarian$SurvInd
)
# Fit copula model.
fitted_model = fit_model_SurvSurv(data = data_scr,
                                  copula_family = "clayton",
                                  n_knots = 1)
# Define grid for GoF plots.
grid = seq(from = 1e-3,
           to = 2.5,
           length.out = 30)
```

```
# Assess marginal goodness-of-fit in the control group.
marginal_gof_scr_S_plot(fitted_model, grid = grid, treated = 0)
marginal_gof_scr_T_plot(fitted_model, grid = grid, treated = 0)
# Assess goodness-of-fit of the association structure, i.e., the copula.
prob_dying_without_progression_plot(fitted_model, grid = grid, treated = 0)
mean_S_before_T_plot_scr(fitted_model, grid = grid, treated = 0)
```

MetaAnalyticSurvBin *Compute surrogacy measures for a binary surrogate and a time-to-event true endpoint in the meta-analytic multiple-trial setting.*

Description

The function 'MetaAnalyticSurvBin()' fits the model for a binary surrogate and time-to-event true endpoint developed by Burzykowski et al. (2004) in the meta-analytic multiple-trial setting.

Usage

```
MetaAnalyticSurvBin(
  data,
  true,
  trueind,
  surrog,
  trt,
  center,
  trial,
  patientid,
  adjustment
)
```

Arguments

data	A data frame with the correct columns (See Data Format).
true	Observed time-to-event (true endpoint).
trueind	Time-to-event indicator.
surrog	Binary surrogate endpoint, coded as 1 or 2.
trt	Treatment indicator, coded as 0 or 1.
center	Center indicator (equal to trial if there are no different centers). This is the unit for which specific treatment effects are estimated.
trial	Trial indicator. This is the unit for which common baselines are to be used.
patientid	Patient indicator.
adjustment	The adjustment that should be made for the trial-level surrogacy, either "unadjusted", "weighted" or "adjusted"

Value

Returns an object of class "MetaAnalyticSurvBin" that can be used to evaluate surrogacy and contains the following elements:

- Indiv.Surrogacy: a data frame that contains the global odds ratio and 95% confidence interval to evaluate surrogacy at the individual level.
- Trial.R2: a data frame that contains the R^2_{trial} and 95% confidence interval to evaluate surrogacy at the trial level.
- EstTreatEffects: a data frame that contains the estimated treatment effects and sample size for each trial.
- nlm.output: output of the maximization procedure (nlm) to maximize the likelihood function.

Model

In the model developed by Burzykowski et al. (2004), a copula-based model is used for the true endpoint and a latent continuous variable, underlying the surrogate endpoint. More specifically, the Plackett copula is used. The marginal model for the surrogate endpoint is a logistic regression model. For the true endpoint, the proportional hazard model is used. The quality of the surrogate at the individual level can be evaluated by using the copula parameter Θ, which takes the form of a global odds ratio. The quality of the surrogate at the trial level can be evaluated by considering the R^2_{trial} between the estimated treatment effects.

Data Format

The data frame must contains the following columns:

- a column with the observed time-to-event (true endpoint)
- a column with the time-to-event indicator: 1 if the event is observed, 0 otherwise
- a column with the binary surrogate endpoint: 1 or 2
- a column with the treatment indicator: 0 or 1
- a column with the trial indicator
- a column with the center indicator. If there are no different centers within each trial, the center indicator can be equal to the trial indicator
- a column with the patient indicator

Author(s)

Dries De Witte

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2004). The validation of surrogate end points by using data from randomized clinical trials: a case-study in advanced colorectal cancer. Journal of the Royal Statistical Society Series A: Statistics in Society, 167(1), 103-124.

Examples

```
## Not run:
data("colorectal")
fit_bin <- MetaAnalyticSurvBin(data = colorectal, true = surv, trueind = SURVIND,
                               surrog = responder, trt = TREAT, center = CENTER,
                               trial = TRIAL, patientid = patientid,
                               adjustment="unadjusted")
print(fit_bin)
summary(fit_bin)
plot(fit_bin)

## End(Not run)
```

MetaAnalyticSurvCat   *Compute surrogacy measures for a categorical (ordinal) surrogate and a time-to-event true endpoint in the meta-analytic multiple-trial setting.*

Description

The function 'MetaAnalyticSurvCat()' fits the model for a categorical (ordinal) surrogate and time-to-event true endpoint developed by Burzykowski et al. (2004) in the meta-analytic multiple-trial setting.

Usage

```
MetaAnalyticSurvCat(
  data,
  true,
  trueind,
  surrog,
  trt,
  center,
  trial,
  patientid,
  adjustment
)
```

Arguments

data	A data frame with the correct columns (See Data Format).
true	Observed time-to-event (true endpoint).
trueind	Time-to-event indicator.
surrog	Ordinal surrogate endpoint, coded as 1 2 3 ... K.
trt	Treatment indicator, coded as 0 or 1.

`center` Center indicator (equal to trial if there are no different centers). This is the unit for which specific treatment effects are estimated.

`trial` Trial indicator. This is the unit for which common baselines are to be used.

`patientid` Patient indicator.

`adjustment` The adjustment that should be made for the trial-level surrogacy, either "unadjusted", "weighted" or "adjusted"

Value

Returns an object of class "MetaAnalyticSurvCat" that can be used to evaluate surrogacy and contains the following elements:

- Indiv.Surrogacy: a data frame that contains the Global Odds and 95% confidence interval to evaluate surrogacy at the individual level.
- Trial.R2: a data frame that contains the R^2_{trial} and 95% confidence interval to evaluate surrogacy at the trial level.
- EstTreatEffects: a data frame that contains the estimated treatment effects and sample size for each trial.
- nlm.output: output of the maximization procedure (nlm) to maximize the likelihood function.

Model

In the model developed by Burzykowski et al. (2004), a copula-based model is used for the true endpoint and a latent continuous variable, underlying the surrogate endpoint. More specifically, the Plackett copula is used. The marginal model for the surrogate endpoint is a proportional odds model. For the true endpoint, the proportional hazards model is used. The quality of the surrogate at the individual level can be evaluated by using the copula parameter Θ, which takes the form of a global odds ratio. The quality of the surrogate at the trial level can be evaluated by considering the R^2_{trial} between the estimated treatment effects.

Data Format

The data frame must contains the following columns:

- a column with the observed time-to-event (true endpoint)
- a column with the time-to-event indicator: 1 if the event is observed, 0 otherwise
- a column with the ordinal surrogate endpoint: 1 2 3 ... K
- a column with the treatment indicator: 0 or 1
- a column with the trial indicator
- a column with the center indicator. If there are no different centers within each trial, the center indicator is equal to the trial indicator
- a column with the patient indicator

Author(s)

Dries De Witte

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2004). The validation of surrogate end points by using data from randomized clinical trials: a case-study in advanced colorectal cancer. Journal of the Royal Statistical Society Series A: Statistics in Society, 167(1), 103-124.

Examples

```
## Not run:
data("colorectal4")
fit <- MetaAnalyticSurvCat(data = colorectal4, true = truend, trueind = trueind, surrog = surrogend,
                          trt = treatn, center = center, trial = trialend, patientid = patid,
                              adjustment="unadjusted")
print(fit)
summary(fit)
plot(fit)

## End(Not run)
```

MetaAnalyticSurvCont *Compute surrogacy measures for a continuous (normally-distributed) surrogate and a time-to-event true endpoint in the meta-analytic multiple-trial setting.*

Description

The function 'MetaAnalyticSurvCont()' fits the model for a continuous surrogate and time-to-event true endpoint described by Alonso et al. (2016) in the meta-analytic multiple-trial setting.

Usage

```
MetaAnalyticSurvCont(
  data,
  true,
  trueind,
  surrog,
  trt,
  center,
  trial,
  patientid,
  copula,
  adjustment
)
```

Arguments

`data`	A data frame with the correct columns (See Data Format).
`true`	Observed time-to-event for true endpoint.

`trueind` Time-to-event indicator for the true endpoint.

`surrog` Continuous surrogate endpoint.

`trt` Treatment indicator.

`center` Center indicator (equal to trial if there are no different centers). This is the unit for which specific treatment effects are estimated.

`trial` Trial indicator. This is the unit for which common baselines are to be used.

`patientid` Patient indicator.

`copula` The copula that is used, either "Clayton", "Hougaard" or "Plackett"

`adjustment` The adjustment that should be made for the trial-level surrogacy, either "unadjusted", "weighted" or "adjusted"

Value

Returns an object of class "MetaAnalyticSurvCont" that can be used to evaluate surrogacy and contains the following elements:

- Indiv.Surrogacy: a data frame that contains the measure for the individual level surrogacy and 95% confidence interval.
- Trial.R2: a data frame that contains the R^2_{trial} and 95% confidence interval to evaluate surrogacy at the trial level.
- EstTreatEffects: a data frame that contains the estimated treatment effects and sample size for each trial.
- nlm.output: output of the maximization procedure (nlm) to maximize the likelihood.

Model

In the model, a copula-based model is used for the true time-to-event endpoint and the surrogate continuous, normally distributed endpoint. More specifically, three copulas can be used: the Clayton copula, Hougaard copula and Plackett copula. The marginal model for the true endpoint is the proportional hazard model. The marginal model for the surrogate endpoint is the classical linear regression model. The quality of the surrogate at the individual level can be evaluated by either Kendall's τ or Spearman's ρ, depending on which copula function is used. The quality of the surrogate at the trial level can be evaluated by considering the R^2_{trial} between the estimated treatment effects.

Data Format

The data frame must contains the following columns:

- a column with the observed time-to-event for the true endpoint
- a column with the time-to-event indicator for the true endpoint: 1 if the event is observed, 0 otherwise
- a column with the continuous surrogate endpoint
- a column with the treatment indicator: 0 or 1
- a column with the trial indicator
- a column with the center indicator. If there are no different centers within each trial, the center indicator is equal to the trial indicator
- a column with the patient indicator

Author(s)

Dries De Witte

References

Alonso A, Bigirumurame T, Burzykowski T, Buyse M, Molenberghs G, Muchene L, Perualila NJ, Shkedy Z, Van der Elst W, et al. (2016). Applied surrogate endpoint evaluation methods with SAS and R. CRC Press New York

Examples

```
## Not run:
data("prostate")
fit <- MetaAnalyticSurvCont(data = prostate, true = SURVTIME, trueind = SURVIND, surrog = PSA,
trt = TREAT, center = TRIAL, trial = TRIAL, patientid = PATID,
copula = "Hougaard", adjustment = "weighted")
summary(fit)
print(fit)
plot(fit)

## End(Not run)
```

MetaAnalyticSurvSurv *Compute surrogacy measures for a time-to-event surrogate and a time-to-event true endpoint in the meta-analytic multiple-trial setting.*

Description

The function ’MetaAnalyticSurvSurv()’ fits the model for a time-to-event surrogate and time-to-event true endpoint developed by Burzykowski et al. (2001) in the meta-analytic multiple-trial setting.

Usage

```
MetaAnalyticSurvSurv(
  data,
  true,
  trueind,
  surrog,
  surrogind,
  trt,
  center,
  trial,
  patientid,
  copula,
  adjustment
)
```

Arguments

`data`	A data frame with the correct columns (See Data Format).
`true`	Observed time-to-event for true endpoint.
`trueind`	Time-to-event indicator for the true endpoint.
`surrog`	Observed time-to-event for surrogate endpoint.
`surrogind`	Time-to-event indicator for the surrogate endpoint.
`trt`	Treatment indicator.
`center`	Center indicator (equal to trial if there are no different centers). This is the unit for which specific treatment effects are estimated.
`trial`	Trial indicator. This is the unit for which common baselines are to be used.
`patientid`	Patient indicator.
`copula`	The copula that is used, either "Clayton", "Hougaard" or "Plackett"
`adjustment`	The adjustment that should be made for the trial-level surrogacy, either "unadjusted", "weighted" or "adjusted"

Value

Returns an object of class "MetaAnalyticSurvSurv" that can be used to evaluate surrogacy and contains the following elements:

- Indiv.Surrogacy: a data frame that contains the measure for the individual level surrogacy and 95% confidence interval.
- Trial.R2: a data frame that contains the R^2_{trial} and 95% confidence interval to evaluate surrogacy at the trial level.
- EstTreatEffects: a data frame that contains the estimated treatment effects and sample size for each trial.
- nlm.output: output of the maximization procedure (nlm) to maximize the likelihood.

Model

In the model developed by Burzykowski et al. (2001), a copula-based model is used for the true time-to-event endpoint and the surrogate time-to-event endpoint. More specifically, three copulas can be used: the Clayton copula, Hougaard copula and Plackett copula. The marginal model for the true and surrogate endpoint is the proportional hazard model. The quality of the surrogate at the individual level can be evaluated by by either Kendall's τ or Spearman's ρ, depending on which copula function is used. The quality of the surrogate at the trial level can be evaluated by considering the R^2_{trial} between the estimated treatment effects.

Data Format

The data frame must contains the following columns:

- a column with the observed time-to-event for the true endpoint
- a column with the time-to-event indicator for the true endpoint: 1 if the event is observed, 0 otherwise

- a column with the observed time-to-event for the surrogate endpoint
- a column with the time-to-event indicator for the surrogate endpoint: 1 if the event is observed, 0 otherwise
- a column with the treatment indicator: 0 or 1
- a column with the trial indicator
- a column with the center indicator. If there are no different centers within each trial, the center indicator is equal to the trial indicator
- a column with the patient indicator

Author(s)

Dries De Witte

References

Burzykowski T, Molenberghs G, Buyse M, Geys H, Renard D (2001). "Validation of surrogate end points in multiple randomized clinical trials with failure time end points." Journal of the Royal Statistical Society Series C: Applied Statistics, 50(4), 405–422

Examples

```
## Not run:
data("Ovarian")
fit <- MetaAnalyticSurvSurv(data=Ovarian,true=Surv,trueind=SurvInd,surrog=Pfs,surrogind=PfsInd,
                            trt=Treat,center=Center,trial=Center,patientid=Patient,
                            copula="Plackett",adjustment="unadjusted")
print(fit)
summary(fit)
plot(fit)

## End(Not run)
```

MICA.ContCont *Assess surrogacy in the causal-inference multiple-trial setting (Meta-analytic Individual Causal Association; MICA) in the continuous-continuous case*

Description

The function `MICA.ContCont` quantifies surrogacy in the multiple-trial causal-inference framework. See **Details** below.

Usage

```
MICA.ContCont(Trial.R, D.aa, D.bb, T0S0, T1S1, T0T0=1, T1T1=1, S0S0=1, S1S1=1,
T0T1=seq(-1, 1, by=.1), T0S1=seq(-1, 1, by=.1), T1S0=seq(-1, 1, by=.1),
S0S1=seq(-1, 1, by=.1))
```

Arguments

`Trial.R` A scalar that specifies the trial-level correlation coefficient (i.e., R_{trial}) that should be used in the computation of ρ_M.

`D.aa` A scalar that specifies the between-trial variance of the treatment effects on the surrogate endpoint (i.e., d_{aa}) that should be used in the computation of ρ_M.

`D.bb` A scalar that specifies the between-trial variance of the treatment effects on the true endpoint (i.e., d_{bb}) that should be used in the computation of ρ_M.

`T0S0` A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the control treatment condition that should be considered in the computation of ρ_M.

`T1S1` A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_M.

`T0T0` A scalar that specifies the variance of the true endpoint in the control treatment condition that should be considered in the computation of ρ_M. Default 1.

`T1T1` A scalar that specifies the variance of the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_M. Default 1.

`S0S0` A scalar that specifies the variance of the surrogate endpoint in the control treatment condition that should be considered in the computation of ρ_M. Default 1.

`S1S1` A scalar that specifies the variance of the surrogate endpoint in the experimental treatment condition that should be considered in the computation of ρ_M. Default 1.

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.1)`, i.e., the values -1, -0.9, -0.8, $\ldots$, 1.

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.1)`.

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.1)`.

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.1)`.

Details

Based on the causal-inference framework, it is assumed that each subject *j* in trial *i* has four counterfactuals (or potential outcomes), i.e., T_{0ij}, T_{1ij}, S_{0ij}, and S_{1ij}. Let T_{0ij} and T_{1ij} denote the counterfactuals for the true endpoint (T) under the control ($Z = 0$) and the experimental ($Z = 1$) treatments of subject *j* in trial *i*, respectively. Similarly, S_{0ij} and S_{1ij} denote the corresponding counterfactuals for the surrogate endpoint (S) under the control and experimental treatments of subject *j* in trial *i*, respectively. The individual causal effects of Z on T and S for a given subject *j* in trial *i* are then defined as $\Delta_{T_{ij}} = T_{1ij} - T_{0ij}$ and $\Delta_{S_{ij}} = S_{1ij} - S_{0ij}$, respectively.

In the multiple-trial causal-inference framework, surrogacy can be quantified as the correlation between the individual causal effects of Z on S and T (for details, see Alonso et al., submitted):

$$\rho_M = \rho(\Delta_{Tij},\, \Delta_{Sij}) = \frac{\sqrt{d_{bb} d_{aa}} R_{trial} + \sqrt{V(\varepsilon_{\Delta Tij}) V(\varepsilon_{\Delta Sij})} \rho_\Delta}{\sqrt{V(\Delta_{Tij}) V(\Delta_{Sij})}},$$

where

$$V(\varepsilon_{\Delta Tij}) = \sigma_{T_0 T_0} + \sigma_{T_1 T_1} - 2\sqrt{\sigma_{T_0 T_0} \sigma_{T_1 T_1}} \rho_{T_0 T_1},$$

$$V(\varepsilon_{\Delta Sij}) = \sigma_{S_0 S_0} + \sigma_{S_1 S_1} - 2\sqrt{\sigma_{S_0 S_0} \sigma_{S_1 S_1}} \rho_{S_0 S_1},$$

$$V(\Delta_{Tij}) = d_{bb} + \sigma_{T_0 T_0} + \sigma_{T_1 T_1} - 2\sqrt{\sigma_{T_0 T_0} \sigma_{T_1 T_1}} \rho_{T_0 T_1},$$

$$V(\Delta_{Sij}) = d_{aa} + \sigma_{S_0 S_0} + \sigma_{S_1 S_1} - 2\sqrt{\sigma_{S_0 S_0} \sigma_{S_1 S_1}} \rho_{S_0 S_1}.$$

The correlations between the counterfactuals (i.e., $\rho_{S_0 T_1}$, $\rho_{S_1 T_0}$, $\rho_{T_0 T_1}$, and $\rho_{S_0 S_1}$) are not identifiable from the data. It is thus warranted to conduct a sensitivity analysis (by considering vectors of possible values for the correlations between the counterfactuals – rather than point estimates).

When the user specifies a vector of values that should be considered for one or more of the correlations that are involved in the computation of ρ_M, the function `MICA.ContCont` constructs all possible matrices that can be formed as based on the specified values, identifies the matrices that are positive definite (i.e., valid correlation matrices), and computes ρ_M for each of these matrices. An examination of the vector of the obtained ρ_M values allows for a straightforward examination of the impact of different assumptions regarding the correlations between the counterfactuals on the results (see also `plot Causal-Inference ContCont`), and the extent to which proponents of the causal-inference and meta-analytic frameworks will reach the same conclusion with respect to the appropriateness of the candidate surrogate at hand.

Notes A single ρ_M value is obtained when all correlations in the function call are scalars.

Value

An object of class `MICA.ContCont` with components,

`Total.Num.Matrices`
An object of class `numeric` which contains the total number of matrices that can be formed as based on the user-specified correlations.

`Pos.Def` A `data.frame` that contains the positive definite matrices that can be formed based on the user-specified correlations. These matrices are used to compute the vector of the ρ_M values.

`ICA` A scalar or vector of the ρ_Δ values.

`MICA` A scalar or vector of the ρ_M values.

Warning

The theory that relates the causal-inference and the meta-analytic frameworks in the multiple-trial setting (as developped in Alonso et al., submitted) assumes that a reduced or semi-reduced modelling approach is used in the meta-analytic framework. Thus R_{trial}, d_{aa} and d_{bb} should be estimated based on a reduced model (i.e., using the `Model=c("Reduced")` argument in the functions `UnifixedContCont`, `UnimixedContCont`, `BifixedContCont`, or `BimixedContCont`) or based on a semi-reduced model (i.e., using the `Model=c("SemiReduced")` argument in the functions `UnifixedContCont`, `UnimixedContCont`, or `BifixedContCont`).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal-inference and meta-analytic paradigms for the validation of surrogate markers.

See Also

`ICA.ContCont`, `MICA.Sample.ContCont`, `plot Causal-Inference ContCont`, `UnifixedContCont`, `UnimixedContCont`, `BifixedContCont`, `BimixedContCont`

Examples

```
## Not run:  #time-consuming code parts
# Generate the vector of MICA values when R_trial=.8, rho_T0S0=rho_T1S1=.8,
# sigma_T0T0=90, sigma_T1T1=100,sigma_ S0S0=10, sigma_S1S1=15, D.aa=5, D.bb=10,
# and when the grid of values {0, .2, ..., 1} is considered for the
# correlations between the counterfactuals:
SurMICA <- MICA.ContCont(Trial.R=.80, D.aa=5, D.bb=10, T0S0=.8, T1S1=.8,
T0T0=90, T1T1=100, S0S0=10, S1S1=15, T0T1=seq(0, 1, by=.2),
T0S1=seq(0, 1, by=.2), T1S0=seq(0, 1, by=.2), S0S1=seq(0, 1, by=.2))

# Examine and plot the vector of the generated MICA values:
summary(SurMICA)
plot(SurMICA)


# Same analysis, but now assume that D.aa=.5 and D.bb=.1:
SurMICA <- MICA.ContCont(Trial.R=.80, D.aa=.5, D.bb=.1, T0S0=.8, T1S1=.8,
T0T0=90, T1T1=100, S0S0=10, S1S1=15, T0T1=seq(0, 1, by=.2),
T0S1=seq(0, 1, by=.2), T1S0=seq(0, 1, by=.2), S0S1=seq(0, 1, by=.2))

# Examine and plot the vector of the generated MICA values:
summary(SurMICA)
plot(SurMICA)


# Same as first analysis, but specify vectors for rho_T0S0 and rho_T1S1:
```

```
# Sample from normal with mean .8 and SD=.1 (to account for uncertainty
# in estimation)
SurMICA <- MICA.ContCont(Trial.R=.80, D.aa=5, D.bb=10,
T0S0=rnorm(n=10000000, mean=.8, sd=.1),
T1S1=rnorm(n=10000000, mean=.8, sd=.1),
T0T0=90, T1T1=100, S0S0=10, S1S1=15, T0T1=seq(0, 1, by=.2),
T0S1=seq(0, 1, by=.2), T1S0=seq(0, 1, by=.2), S0S1=seq(0, 1, by=.2))

## End(Not run)
```

MICA.Sample.ContCont *Assess surrogacy in the causal-inference multiple-trial setting (Meta-analytic Individual Causal Association; MICA) in the continuous-continuous case using the grid-based sample approach*

Description

The function MICA.Sample.ContCont quantifies surrogacy in the multiple-trial causal-inference framework. It provides a faster alternative for MICA.ContCont. See **Details** below.

Usage

```
MICA.Sample.ContCont(Trial.R, D.aa, D.bb, T0S0, T1S1, T0T0=1, T1T1=1, S0S0=1, S1S1=1,
T0T1=seq(-1, 1, by=.001), T0S1=seq(-1, 1, by=.001), T1S0=seq(-1, 1, by=.001),
S0S1=seq(-1, 1, by=.001), M=50000)
```

Arguments

Trial.R	A scalar that specifies the trial-level correlation coefficient (i.e., R_{trial}) that should be used in the computation of ρ_M.
D.aa	A scalar that specifies the between-trial variance of the treatment effects on the surrogate endpoint (i.e., d_{aa}) that should be used in the computation of ρ_M.
D.bb	A scalar that specifies the between-trial variance of the treatment effects on the true endpoint (i.e., d_{bb}) that should be used in the computation of ρ_M.
T0S0	A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the control treatment condition that should be considered in the computation of ρ_M.
T1S1	A scalar or vector that specifies the correlation(s) between the surrogate and the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_M.
T0T0	A scalar that specifies the variance of the true endpoint in the control treatment condition that should be considered in the computation of ρ_M. Default 1.
T1T1	A scalar that specifies the variance of the true endpoint in the experimental treatment condition that should be considered in the computation of ρ_M. Default 1.

`S0S0` A scalar that specifies the variance of the surrogate endpoint in the control treatment condition that should be considered in the computation of ρ_M. Default 1.

`S1S1` A scalar that specifies the variance of the surrogate endpoint in the experimental treatment condition that should be considered in the computation of ρ_M. Default 1.

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.001)`.

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.001)`.

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.001)`.

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1 that should be considered in the computation of ρ_M. Default `seq(-1, 1, by=.001)`.

`M` The number of runs that should be conducted. Default `50000`.

Details

Based on the causal-inference framework, it is assumed that each subject *j* in trial *i* has four counterfactuals (or potential outcomes), i.e., T_{0ij}, T_{1ij}, S_{0ij}, and S_{1ij}. Let T_{0ij} and T_{1ij} denote the counterfactuals for the true endpoint (T) under the control ($Z = 0$) and the experimental ($Z = 1$) treatments of subject *j* in trial *i*, respectively. Similarly, S_{0ij} and S_{1ij} denote the corresponding counterfactuals for the surrogate endpoint (S) under the control and experimental treatments of subject *j* in trial *i*, respectively. The individual causal effects of Z on T and S for a given subject *j* in trial *i* are then defined as $\Delta_{T_{ij}} = T_{1ij} - T_{0ij}$ and $\Delta_{S_{ij}} = S_{1ij} - S_{0ij}$, respectively.

In the multiple-trial causal-inference framework, surrogacy can be quantified as the correlation between the individual causal effects of Z on S and T (for details, see Alonso et al., submitted):

$$\rho_M = \rho(\Delta_{Tij}, \, \Delta_{Sij}) = \frac{\sqrt{d_{bb} d_{aa}} R_{trial} + \sqrt{V(\varepsilon_{\Delta Tij}) V(\varepsilon_{\Delta Sij})} \rho_\Delta}{\sqrt{V(\Delta_{Tij}) V(\Delta_{Sij})}},$$

where

$$\begin{aligned} V(\varepsilon_{\Delta Tij}) &= \sigma_{T_0T_0} + \sigma_{T_1T_1} - 2\sqrt{\sigma_{T_0T_0}\sigma_{T_1T_1}}\rho_{T_0T_1}, \\ V(\varepsilon_{\Delta Sij}) &= \sigma_{S_0S_0} + \sigma_{S_1S_1} - 2\sqrt{\sigma_{S_0S_0}\sigma_{S_1S_1}}\rho_{S_0S_1}, \\ V(\Delta_{Tij}) &= d_{bb} + \sigma_{T_0T_0} + \sigma_{T_1T_1} - 2\sqrt{\sigma_{T_0T_0}\sigma_{T_1T_1}}\rho_{T_0T_1}, \\ V(\Delta_{Sij}) &= d_{aa} + \sigma_{S_0S_0} + \sigma_{S_1S_1} - 2\sqrt{\sigma_{S_0S_0}\sigma_{S_1S_1}}\rho_{S_0S_1}. \end{aligned}$$

The correlations between the counterfactuals (i.e., $\rho_{S_0T_1}$, $\rho_{S_1T_0}$, $\rho_{T_0T_1}$, and $\rho_{S_0S_1}$) are not identifiable from the data. It is thus warranted to conduct a sensitivity analysis (by considering vectors of possible values for the correlations between the counterfactuals – rather than point estimates).

When the user specifies a vector of values that should be considered for one or more of the correlations that are involved in the computation of ρ_M, the function MICA.ContCont constructs all possible matrices that can be formed as based on the specified values, and retains the positive definite ones for the computation of ρ_M.

In contrast, the function MICA.Sample.ContCont samples random values for $\rho_{S_0T_1}$, $\rho_{S_1T_0}$, $\rho_{T_0T_1}$, and $\rho_{S_0S_1}$ based on a uniform distribution with user-specified minimum and maximum values, and retains the positive definite ones for the computation of ρ_M.

An examination of the vector of the obtained ρ_M values allows for a straightforward examination of the impact of different assumptions regarding the correlations between the counterfactuals on the results (see also plot Causal-Inference ContCont), and the extent to which proponents of the causal-inference and meta-analytic frameworks will reach the same conclusion with respect to the appropriateness of the candidate surrogate at hand.

Notes A single ρ_M value is obtained when all correlations in the function call are scalars.

Value

An object of class MICA.ContCont with components,

Total.Num.Matrices
An object of class numeric which contains the total number of matrices that can be formed as based on the user-specified correlations.

Pos.Def
A data.frame that contains the positive definite matrices that can be formed based on the user-specified correlations. These matrices are used to compute the vector of the ρ_M values.

ICA
A scalar or vector of the ρ_Δ values.

MICA
A scalar or vector of the ρ_M values.

Warning

The theory that relates the causal-inference and the meta-analytic frameworks in the multiple-trial setting (as developped in Alonso et al., submitted) assumes that a reduced or semi-reduced modelling approach is used in the meta-analytic framework. Thus R_{trial}, d_{aa} and d_{bb} should be estimated based on a reduced model (i.e., using the Model=c("Reduced") argument in the functions UnifixedContCont, UnimixedContCont, BifixedContCont, or BimixedContCont) or based on a semi-reduced model (i.e., using the Model=c("SemiReduced") argument in the functions UnifixedContCont, UnimixedContCont, or BifixedContCont).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal-inference and meta-analytic paradigms for the validation of surrogate markers.

See Also

ICA.ContCont, MICA.ContCont, plot Causal-Inference ContCont, UnifixedContCont, UnimixedContCont, BifixedContCont, BimixedContCont

Examples

```
## Not run:  #Time consuming (>5 sec) code part
# Generate the vector of MICA values when R_trial=.8, rho_T0S0=rho_T1S1=.8,
# sigma_T0T0=90, sigma_T1T1=100,sigma_ S0S0=10, sigma_S1S1=15, D.aa=5, D.bb=10,
# and when the grid of values {-1, -0.999, ..., 1} is considered for the
# correlations between the counterfactuals:
SurMICA <- MICA.Sample.ContCont(Trial.R=.80, D.aa=5, D.bb=10, T0S0=.8, T1S1=.8,
T0T0=90, T1T1=100, S0S0=10, S1S1=15, T0T1=seq(-1, 1, by=.001),
T0S1=seq(-1, 1, by=.001), T1S0=seq(-1, 1, by=.001),
S0S1=seq(-1, 1, by=.001), M=10000)

# Examine and plot the vector of the generated MICA values:
summary(SurMICA)
plot(SurMICA, ICA=FALSE, MICA=TRUE)


# Same analysis, but now assume that D.aa=.5 and D.bb=.1:
SurMICA <- MICA.Sample.ContCont(Trial.R=.80, D.aa=.5, D.bb=.1, T0S0=.8, T1S1=.8,
T0T0=90, T1T1=100, S0S0=10, S1S1=15, T0T1=seq(-1, 1, by=.001),
T0S1=seq(-1, 1, by=.001), T1S0=seq(-1, 1, by=.001),
S0S1=seq(-1, 1, by=.001), M=10000)

# Examine and plot the vector of the generated MICA values:
summary(SurMICA)
plot(SurMICA)

## End(Not run)
```

MinSurrContCont *Examine the plausibility of finding a good surrogate endpoint in the Continuous-continuous case*

Description

The function MinSurrContCont examines the plausibility of finding a good surrogate endpoint in the continuous-continuous setting. For details, see Alonso et al. (submitted).

Usage

```
MinSurrContCont(T0T0, T1T1, Delta, T0T1=seq(from=0, to=1, by=.01))
```

Arguments

`T0T0` A scalar that specifies the variance of the true endpoint in the control treatment condition.

`T1T1` A scalar that specifies the variance of the true endpoint in the experimental treatment condition.

`Delta` A scalar that specifies an upper bound for the prediction mean squared error when predicting the individual causal effect of the treatment on the true endpoint based on the individual causal effect of the treatment on the surrogate.

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ^2_{min}. Default `seq(0, 1, by=.1)`, i.e., the values 0, 0.10, 0.20, $\ldots$, 1.

Value

An object of class `MinSurrContCont` with components,

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that were considered (i.e., $\rho_{T_0 T_1}$).

`Sigma.Delta.T` A scalar or vector that contains the standard deviations of the individual causal treatment effects on the true endpoint as a function of $\rho_{T_0 T_1}$.

`Rho2.Min` A scalar or vector that contains the ρ^2_{min} values as a function of $\rho_{T_0 T_1}$.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal-inference and meta-analytic paradigms for the validation of surrogate markers.

See Also

`ICA.ContCont`, `plot Causal-Inference ContCont`, `plot MinSurrContCont`

Examples

```
# Assess the plausibility of finding a good surrogate when
# sigma_T0T0 = sigma_T1T1 = 8 and Delta = 1
## Not run:
MinSurr <- MinSurrContCont(T0T0 = 8, T1T1 = 8, Delta = 1)
summary(MinSurr)
plot(MinSurr)
## End(Not run)
```

MixedContContIT *Fits (univariate) mixed-effect models to assess surrogacy in the continuous-continuous case based on the Information-Theoretic framework*

Description

The function MixedContContIT uses the information-theoretic approach (Alonso & Molenberghs, 2007) to estimate trial- and individual-level surrogacy based on mixed-effect models when both S and T are continuous endpoints. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below.

Usage

```
MixedContContIT(Dataset, Surr, True, Treat, Trial.ID, Pat.ID,
Model=c("Full"), Weighted=TRUE, Min.Trial.Size=2, Alpha=.05, ...)
```

Arguments

Dataset A data.frame that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.

Surr The name of the variable in Dataset that contains the surrogate endpoint values.

True The name of the variable in Dataset that contains the true endpoint values.

Treat The name of the variable in Dataset that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.

Trial.ID The name of the variable in Dataset that contains the trial ID to which the patient belongs.

Pat.ID The name of the variable in Dataset that contains the patient's ID.

Model The type of model that should be fitted, i.e., Model=c("Full"), Model=c("Reduced"), or Model=c("SemiReduced"). See the **Details** section below. Default Model=c("Full").

Weighted Logical. In practice it is often the case that different trials (or other clustering units) have different sample sizes. Univariate models are used to assess surrogacy in the information-theoretic approach, so it can be useful to adjust for heterogeneity in information content between the trial-specific contributions (particularly when trial-level surrogacy measures are of primary interest and when the heterogeneity in sample sizes is large). If Weighted=TRUE, weighted regression models are fitted. If Weighted=FALSE, unweighted regression analyses are conducted. See the **Details** section below. Default TRUE.

Min.Trial.Size The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by Min.Trial.Size, the data of the trial are excluded from the analysis. Default 2.

Alpha The α-level that is used to determine the confidence intervals around R_h^2 and R_{ht}^2. Default 0.05.

... Other arguments to be passed to the function `lmer` (of the R package `lme4`) that is used to fit the geralized linear mixed-effect models in the function `BimixedContCont`.

Details

Individual-level surrogacy

The following generalised linear mixed-effect models are fitted:

$$g_T(E(T_{ij})) = \mu_T + m_{Ti} + \beta Z_{ij} + b_i Z_{ij},$$
$$g_T(E(T_{ij}|S_{ij})) = \theta_0 + c_{Ti} + \theta_1 Z_{ij} + a_i Z_{ij} + \theta_{2i} S_{ij},$$

where i and j are the trial and subject indicators, g_T is an appropriate link function (i.e., an identity link when a continuous true endpoint is considered), S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, and Z_{ij} is the treatment indicator for subject j in trial i. μ_T and β are a fixed intercept and a fixed treatment-effect on the true endpoint, while m_{Ti} and b_i are the corresponding random effects. θ_0 and θ_1 are the fixed intercept and the fixed treatment effect on the true endpoint after accounting for the effect of the surrogate endpoint, and c_{Ti} and a_i are the corresponding random effects.

The -2 log likelihood values of the previous models (i.e., L_1 and L_2, respectively) are subsequently used to compute individual-level surrogacy (based on the so-called Variance Reduction Factor, VFR; for details, see Alonso & Molenberghs, 2007):

$$R^2_{hind} = 1 - exp\left(-\frac{L_2 - L_1}{N}\right),$$

where N is the number of trials.

Trial-level surrogacy

When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), trial-level surrogacy is assessed by fitting the following mixed models:

$$S_{ij} = \mu_S + m_{Si} + (\alpha + a_i) Z_{ij} + \varepsilon_{Sij}, (1)$$
$$T_{ij} = \mu_T + m_{Ti} + (\beta + b_i) Z_{ij} + \varepsilon_{Tij}, (1)$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_S and μ_T are the fixed intercepts for S and T, m_{Si} and m_{Ti} are the corresponding random intercepts, α and β are the fixed treatment effects on S and T, and a_i and b_i are the corresponding random effects. The error terms ε_{Sij} and ε_{Tij} are assumed to be independent.

When a reduced model is requested by the user (by using the argument `Model=c("Reduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_S + (\alpha + a_i) Z_{ij} + \varepsilon_{Sij}, (2)$$

$$T_{ij} = \mu_T + (\beta + b_i)Z_{ij} + \varepsilon_{Tij}, (2)$$

where μ_S and μ_T are the common intercepts for S and T. The other parameters are the same as defined above, and ε_{Sij} and ε_{Tij} are again assumed to be independent.

When the user requested that a full model approach is used (by using the argument `Model=c("Full")` in the function call, i.e., when models (1) were fitted), the following model is subsequently fitted:

$$\widehat{\beta_i} = \lambda_0 + \lambda_1 \widehat{\mu_{Si}} + \lambda_2 \widehat{\alpha_i} + \varepsilon_i, (3)$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on models (1) (see above). When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), model (3) is a weighted regression model (with weights based on the number of observations in trial i). The -2 log likelihood value of the (weighted or unweighted) models (3) (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta_i} = \lambda_3$; L_0), and R^2_{ht} is computed based on the Variance Reduction Factor (VFR; for details, see Alonso & Molenberghs, 2007):

$$R^2_{ht} = 1 - exp\left(-\frac{L_1 - L_0}{N}\right),$$

where N is the number of trials.

When a semi-reduced or reduced model is requested (by using the argument `Model=c("SemiReduced")` or `Model=c("Reduced")` in the function call), the following model is fitted:

$$\widehat{\beta_i} = \lambda_0 + \lambda_1 \widehat{\alpha_i} + \varepsilon_i,$$

where the parameter estimates for β_i and α_i are based on models (2). The -2 log likelihood value of this (weighted or unweighted) model (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\widehat{\beta_i} = \lambda_3$; L_0), and R^2_{ht} is computed based on the reduction in the likelihood (as described above).

Value

An object of class `MixedContContIT` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Trial.Spec.Results`
: A `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`R2ht`
: A `data.frame` that contains the trial-level surrogacy estimate and its confidence interval.

`R2h.ind`
: A `data.frame` that contains the individual-level surrogacy estimate and its confidence interval.

`Cor.Endpoints`
: A `data.frame` that contains the correlations between the surrogate and the true endpoint in the control treatment group (i.e., ρ_{T0S0}) and in the experimental treatment group (i.e., ρ_{T1S1}), their standard errors and their confidence intervals.

`Residuals`
: A `data.frame` that contains the residuals for the surrogate and true endpoints (ε_{Sij} and ε_{Tij}) that are obtained when models (1) or models (2) are fitted (see the **Details** section above).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

See Also

FixedContContIT, plot Information-Theoretic

Examples

```
## Not run:  # Time consuming (>5sec) code part
# Example 1
# Based on the ARMD data:
data(ARMD)
# Assess surrogacy based on a full mixed-effect model
# in the information-theoretic framework:
Sur <- MixedContContIT(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Trial.ID=Center,
Pat.ID=Id, Model="Full")
# Obtain a summary of the results:
summary(Sur)

# Example 2
# Conduct an analysis based on a simulated dataset with 2000 patients, 200 trials,
# and Rindiv=Rtrial=.8
# Simulate the data:
Sim.Data.MTS(N.Total=2000, N.Trial=200, R.Trial.Target=.8, R.Indiv.Target=.8,
Seed=123, Model="Full")
# Assess surrogacy based on a full mixed-effect model
# in the information-theoretic framework:
```

```
Sur2 <- MixedContContIT(Dataset=Data.Observed.MTS, Surr=Surr, True=True, Treat=Treat,
Trial.ID=Trial.ID, Pat.ID=Pat.ID, Model="Full")

# Show a summary of the results:
summary(Sur2)
## End(Not run)
```

model_fit_measures *Goodness of fit information for survival-survival model*

Description

This function returns several goodness-of-fit measures for a model fitted by `fit_model_SurvSurv()`. These are primarily intended for model selection.

Usage

```
model_fit_measures(fitted_model)
```

Arguments

`fitted_model`	returned value from `fit_model_SurvSurv()`.

Details

The following goodness-of-fit measures are returned in a named vector:

- `tau_0` and `tau_1`: (latent) value for Kendall's tau in the estimated model.
- `log_lik`: the maximized log-likelihood value.
- `AIC`: the Aikaike information criterion of the fitted model.

Value

a named vector containing the goodness-of-fit measures

Examples

```
library(Surrogate)
data("Ovarian")
#For simplicity, data is not recoded to semi-competing risks format, but is
#left in the composite event format.
data = data.frame(
  Ovarian$Pfs,
  Ovarian$Surv,
  Ovarian$Treat,
  Ovarian$PfsInd,
  Ovarian$SurvInd
)
ovarian_fitted =
    fit_model_SurvSurv(data = data,
                       copula_family = "clayton",
                       n_knots = 1)
model_fit_measures(ovarian_fitted)
```

MufixedContCont.MultS *Fits a multivariate fixed-effects model to assess surrogacy in the meta-analytic multiple-trial setting (Continuous-continuous case with multiple surrogates)*

Description

The function MufixedContCont.MultS uses the multivariate fixed-effects approach to estimate trial- and individual-level surrogacy when the data of multiple clinical trials are available and multiple surrogates are considered for a single true endpoint. The user can specify whether a (weighted or unweighted) full or reduced model should be fitted. See the **Details** section below.

Usage

```
MufixedContCont.MultS(Dataset, Endpoints=True~Surr.1+Surr.2,
 Treat="Treat", Trial.ID="Trial.ID", Pat.ID="Pat.ID",
 Model=c("Full"), Weighted=TRUE, Min.Trial.Size=2, Alpha=.05,
 Number.Bootstraps=0, Seed=123)
```

Arguments

Dataset
: A data.frame that should consist of one line per patient. Each line contains one or more surrogate value(s), a true endpoint value, a treatment indicator, a patient ID, and a trial ID.

Endpoints
: An equation in the form True~Surr.1+Surr.2 that specifies the true endpoint followed by the surrogate endpoint(s).

Treat
: The name of the variable in Dataset that contains the treatment indicators. The treatment indicator should be coded as 1 for the experimental group and -1 for the control group.

Trial.ID
: The name of the variable in Dataset that contains the trial ID to which the patient belongs.

Pat.ID
: The name of the variable in Dataset that contains the patient's ID.

Model
: The type of model that should be fitted, i.e., Model=c("Full") or Model=c("Reduced"). For details, see below or Van der Elst *et al.* (2023). Default Model=c("Full").

Weighted
: Logical. If TRUE, then a weighted regression analysis is conducted at stage 2 of the two-stage approach. If FALSE, then an unweighted regression analysis is conducted at stage 2 of the two-stage approach. See the **Details** section below. Default TRUE.

`Min.Trial.Size` The minimum number of patients that a trial should contain in order to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R^2_{trial} and R^2_{indiv}. Default 0.05.

`Number.Bootstraps`

Lee's (Lee, 1971) approach is done by default to obtain confidence intervals around R^2_{trial} and R^2_{indiv}. Alternatively, a non-parametric bootstrap can be done. By default, `Number.Bootstraps=0` and thus no bootstrap is conducted. If a bootstrap is desired, specify the number of bootstrap samples used this argument. For example, `Number.Bootstraps=100` conducts a bootstrap with 100 bootstrap samples.

`Seed` The seed that is used in the bootstrap. Default `Seed=123`.

Details

When the full multivariate mixed-effects model is fitted to assess surrogacy in the meta-analytic framework (for details, see Van der Elst *et al.*, 2023), computational issues often occur. In that situation, the use of simplified model-fitting strategies may be warranted (for details, see Burzykowski et al., 2005; Tibaldi et al., 2003).

The function `MufixedContCont.MultS` implements one such strategy, i.e., it uses a two-stage multivariate fixed-effects modelling approach to assess surrogacy. In the first stage of the analysis, a multivariate linear regression model is fitted. When a full model is requested (by using the argument `Model=c("Full")` in the function call), the following model is fitted:

$$S1_{ij} = \mu_{S1i} + \alpha_{S1i} Z_{ij} + \varepsilon_{S1ij},$$
$$S2_{ij} = \mu_{S2i} + \alpha_{S2i} Z_{ij} + \varepsilon_{S2ij},$$
$$SK_{ij} = \mu_{SKi} + \alpha_{SKi} Z_{ij} + \varepsilon_{SKij},$$
$$T_{ij} = \mu_{Ti} + \beta_{Ti} Z_{ij} + \varepsilon_{Tij},$$

where Z_{ij} is the treatment indicator for subject j in trial i, μ_{S1i}, μ_{S2i}, ..., μ_{SKi} and μ_{Ti} are the fixed trial-specific intercepts for $S1$, $S2$, ... SK and T, and α_{S1i}, α_{S2i}, ..., α_{SKi} and β_{Ti} are the trial-specific treatment effects on the surrogates and the true endpoint, respectively. When a reduced model is requested (by using the argument `Model=c("Reduced")` in the function call), the following model is fitted:

$$S1_{ij} = \mu_{S1} + \alpha_{S1i} Z_{ij} + \varepsilon_{S1ij},$$
$$S2_{ij} = \mu_{S2} + \alpha_{S2i} Z_{ij} + \varepsilon_{S2ij},$$
$$SK_{ij} = \mu_{SK} + \alpha_{SKi} Z_{ij} + \varepsilon_{SKij},$$
$$T_{ij} = \mu_{Ti} + \beta_{Ti} Z_{ij} + \varepsilon_{Tij},$$

where μ_{S1}, μ_{S2}, ..., μ_{SK} and μ_T are the common intercepts for the surrogates and the true endpoint (i.e., it is assumed that the intercepts for the surrogates and the true endpoints are identical in all trials). The other parameters are the same as defined above.

In the above models, the error terms ε_{S1ij}, ε_{S2ij}, ..., ε_{SKij} and ε_{Tij} are assumed to be mean-zero normally distributed with variance-covariance matrix $\mathbf{\Sigma}$.

Next, the second stage of the analysis is conducted. When a full model is requested by the user (by using the argument `Model=c("Full")` in the function call), the following model is fitted:

$$\widehat{\beta}_{Ti} = \lambda_0 + \lambda_1 \widehat{\mu}_{S1i} + \lambda_2 \widehat{\alpha}_{S1i} + \lambda_3 \widehat{\mu}_{S2i} + \lambda_4 \widehat{\alpha}_{S2i} + ... + \lambda_{2K-1} \widehat{\mu}_{SKi} + \lambda_{2K} \widehat{\alpha}_{SKi} + \varepsilon_i,$$

where the parameter estimates are based on the full model that was fitted in stage 1.

When a reduced model is requested by the user (by using the argument `Model=c("Reduced")`), the $\lambda_1 \widehat{\mu}_{S1i}$, $\lambda_3 \widehat{\mu}_{S2i}$, ... and $\lambda_{2K} \widehat{\mu}_{SKi}$ components are dropped from the above expression.

When the argument `Weighted=FALSE` is used in the function call, the model that is fitted in stage 2 is an unweighted linear regression model. When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), the information that is obtained in stage 1 is weighted according to the number of patients in a trial.

The classical coefficient of determination of the fitted stage 2 model provides an estimate of R^2_{trial}.

Value

An object of class `MufixedContCont.MultS` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Results.Stage.1` The results of stage 1 of the two-stage model fitting approach: a `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate(s) and the true endpoints (when a full model is requested), or the trial-specific treatment effects for the surrogates and the true endpoints (when a reduced model is requested).

`Residuals.Stage.1` A `data.frame` that contains the residuals for the surrogate and true endpoints that are obtained in stage 1 of the analysis (ε_{Sij} and ε_{Tij}).

`Results.Stage.2` An object of class `lm` (linear model) that contains the parameter estimates of the regression model that is fitted in stage 2 of the analysis.

`Trial.R2.Lee` A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval based on the approach of Lee (1971).

`Trial.R2.Boot` A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval based on the non-parametric bootstrap.

`Trial.R2.Adj.Lee`
A `data.frame` that contains the adjusted trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval based on the approach of Lee (1971).

`Trial.R2.Adj.Boot`
A `data.frame` that contains the adjusted trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval based on the non-parametric bootstrap.

`Indiv.R2.Lee` A `data.frame` that contains the individual-level coefficient of determination (R^2_{indiv}), its standard error and confidence interval based on the approach of Lee (1971).

`Indiv.R2.Boot` A `data.frame` that contains the individual-level coefficient of determination (R^2_{indiv}), its standard error and confidence interval based on the non-parametric bootstrap.

`Fitted.Model.Stage.1`
The fitted Stage 1 model.

`Model.R2.Indiv` A linear model that regresses the residuals of T on the residuals of the different surrogates.

`D.Equiv` The variance-covariance matrix of the trial-specific intercept and treatment effects for the surrogates and true endpoints (when a full model is fitted, i.e., when `Model=c("Full")` is used in the function call), or the variance-covariance matrix of the trial-specific treatment effects for the surrogates and true endpoints (when a reduced model is fitted, i.e., when `Model=c("Reduced")` is used in the function call). The variance-covariance matrix `D.Equiv` is equivalent to the $\boldsymbol{D}$ matrix that would be obtained when a (full or reduced) mixed-effect approach is used; see function `MumixedContCont.MultS`).

Author(s)

Wim Van der Elst

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

Lee, Y. S. (1971). Tables of the upper percentage points of the multiple correlation. *Biometrika, 59,* 175-189.

Tibaldi, F., Abrahantes, J. C., Molenberghs, G., Renard, D., Burzykowski, T., Buyse, M., Parmar, M., et al., (2003). Simplified hierarchical linear models for the evaluation of surrogate endpoints. *Journal of Statistical Computation and Simulation, 73,* 643-658.

Van der Elst *et al.* (2024). Multivariate surrogate endpoints for normally distributed continuous endpoints in the meta-analytic setting.

See Also

MumixedContCont.MultS

Examples

```
## Not run:  # time consuming code part
data(PANSS)

# Do a surrogacy analysis with T=Total PANSS score, S1=Negative symptoms
# and S2=Positive symptoms
# Fit a full multivariate fixed-effects model with weighting according to the
# number of patients in stage 2 of the two stage approach to assess surrogacy:
Fit.Neg.Pos <- MufixedContCont.MultS(Dataset = PANSS,
  Endpoints = Total ~ Neg+Pos, Model = "Full",
  Treat = "Treat", Trial.ID = "Invest", Pat.ID = "Pat.ID")

# Obtain a summary of the results
summary(Fit.Neg.Pos)

## End(Not run)
```

MumixedContCont.MultS *Fits a multivariate mixed-effects model to assess surrogacy in the meta-analytic multiple-trial setting (Continuous-continuous case with multiple surrogates)*

Description

The function MumixedContCont.MultS uses the multivariate mixed-effects approach to estimate trial- and individual-level surrogacy when the data of multiple clinical trials are available and multiple surrogates are considered for a single true endpoint. See the **Details** section below.

Usage

```
MumixedContCont.MultS(Dataset, Endpoints=True~Surr.1+Surr.2,
Treat="Treat", Trial.ID="Trial.ID", Pat.ID="Pat.ID",
Model=c("Full"), Min.Trial.Size=2, Alpha=.05, Opt="nlminb")
```

Arguments

Dataset A data.frame that should consist of one line per patient. Each line contains one or more surrogate value(s), a true endpoint value, a treatment indicator, a patient ID, and a trial ID.

Endpoints An equation in the form True~Surr.1+Surr.2 that specifies the true endpoint followed by the surrogate endpoint(s).

Treat The name of the variable in Dataset that contains the treatment indicators. The treatment indicator should be coded as 1 for the experimental group and -1 for the control group.

`Trial.ID` The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.

`Pat.ID` The name of the variable in `Dataset` that contains the patient's ID.

`Model` The type of model that should be fitted, i.e., `Model=c("Full")` or `Model=c("Reduced")`. For details, see below or Van der Elst *et al.* (2023). Default `Model=c("Full")`.

`Min.Trial.Size` The minimum number of patients that a trial should contain in order to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R^2_{trial} and R^2_{indiv} (based on the approach of Lee, 1971). Default 0.05.

`Opt` The optimizer to be used by the `lme` function (the fits the mixed-effects model), with options `nlminb` or `optim`. For details, see `?lmeControl`. Default `Opt="nlminb"`.

Details

When a full model is requested (by using the argument `Model=c("Full")` in the function call), the following mixed-effects model is fitted:

$$\begin{aligned}
S1_{ij} &= \mu_{S1} + m_{S1i}(\alpha_{S1} + a_{S1i})Z_{ij} + \varepsilon_{S1ij},\\
S2_{ij} &= \mu_{S2} + m_{S2i}(\alpha_{S2} + a_{S2i})Z_{ij} + \varepsilon_{S2ij},\\
SK_{ij} &= \mu_{SK} + m_{SKi}(\alpha_{SK} + a_{SKi})Z_{ij} + \varepsilon_{SKij},\\
T_{ij} &= \mu_{T} + m_{Ti}(\beta_{T} + b_{Ti})Z_{ij} + \varepsilon_{Tij},
\end{aligned}$$

where Z_{ij} is the treatment indicator for subject j in trial i, μ_{S1}, μ_{S2}, ... μ_{SK} and μ_T are the fixed intercepts for $S1$, $S2$, ... SK and T, m_{S1i}, m_{S2i}, ... m_{SKi}, and m_{Ti} are the corresponding random intercepts, α_{S1}, α_{S2}, ..., α_{SK} and β_T are the fixed treatment effects for $S1$, $S2$, ... SK and T, and a_{S1i}, a_{S2i}, ... a_{SKi} and b_{Ti} are the corresponding random treatment effects. The vector of the random effects $(m_{S1i},\ m_{S2i},\ ...,\ m_{SKi},\ m_{Ti},\ a_{S1i},\ a_{S2i},\ ...,\ a_{SKi},\ b_{Ti})$ is assumed to be mean-zero normally distributed with unstructured variance-covariance matrix $\mathbf{D}$. Similarly, the residuals ε_{S1ij}, ε_{S2ij}, ... ε_{SKij}, ε_{Tij} are assumed to be mean-zero normally distributed with unstructured variance-covariance matrix $\mathbf{\Sigma}$.

When a reduced model is requested (by using the argument `Model=c("Reduced")` in the function call), the trial-specific intercepts for the surrogate endpoints and the true endpoint in the above model are replaced by common intercepts.

For the full model, R^2_{trial} and R^2_{indiv} are estimated based on $\mathbf{D}$ and $\mathbf{\Sigma}$, respectively:

$$R^2_{trial} = R^2_{b_{Ti}|m_{S1i},\, m_{S2i},\, ...,\, m_{SKi},\, a_{S1i},\, a_{S2i},\, ...\, a_{SKi}} = \frac{\boldsymbol{D}_{ST}^{T}\,\boldsymbol{D}_{SS}^{-1}\,\boldsymbol{D}_{ST}}{\boldsymbol{D}_{TT}},$$

$$R^2_{indiv} = R^2_{\varepsilon_{Tij}|\varepsilon_{S1ij},\, \varepsilon_{S2ij},\, ...,\, \varepsilon_{SKij}} = \frac{\mathbf{\Sigma}_{ST}^{T}\,\mathbf{\Sigma}_{SS}^{-1}\,\mathbf{\Sigma}_{ST}}{\mathbf{\Sigma}_{TT}}.$$

For the reduced model, the reduced $\mathbf{D}$ and $\mathbf{\Sigma}$ are used.

Value

An object of class `MumixedContCont.MultS` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Fixed.Effects` A `data.frame` that contains the fixed intercepts and treatment effects for the true and the surrogate endpoints.

`Random.Effects` A `data.frame` that contains the random intercepts and treatment effects for the true and the surrogate endpoints.

`Trial.R2.Lee` A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval based on the approach of Lee (1971).

`Indiv.R2.Lee` A `data.frame` that contains the individual-level coefficient of determination (R^2_{indiv}), its standard error and confidence interval based on the approach of Lee (1971).

`D` The variance-covariance matrix of the trial-specific intercepts and treatment effects for the surrogates and true endpoints (when a full model is fitted, i.e., when `Model=c("Full")` is used in the function call), or the variance-covariance matrix of the trial-specific treatment effects for the surrogates and true endpoints (when a reduced model is fitted, i.e., when `Model=c("Reduced")` is used in the function call).

`Cond.Number.D.Matrix`
The condition number of the $\mathbf{D}$ matrix.

`Cond.Number.Sigma.Matrix`
The condition number of the $\mathbf{\Sigma}$ matrix.

`Fitted.Model` The fitted mixed-effects model.

Author(s)

Wim Van der Elst

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

Lee, Y. S. (1971). Tables of the upper percentage points of the multiple correlation. *Biometrika, 59,* 175-189.

Van der Elst *et al.* (2024). Multivariate surrogate endpoints for normally distributed continuous endpoints in the meta-analytic setting.

See Also

MufixedContCont.MultS

Examples

```
## Not run:  # time consuming code part
data(PANSS)

# Do a surrogacy analysis with T=Total PANSS score,
# S1=Negative symptoms and S2=Positive symptoms
# Fit a full mixed-effects model:
Fit.Neg.Pos <- MumixedContCont.MultS(Dataset = PANSS,
  Endpoints = Total ~ Neg+Pos, Model = "Full",
  Treat = "Treat", Trial.ID = "Invest", Pat.ID = "Pat.ID")

# Model does not converge, as often happens with the
# mixed-effects approach. Instead, fit a full multivariate
# fixed-effects model with weighting according to the
# number of patients in stage 2 of the two stage approach to assess surrogacy:
Fit.Neg.Pos <- MufixedContCont.MultS(Dataset = PANSS,
  Endpoints = Total ~ Neg+Pos, Model = "Full",
  Treat = "Treat", Trial.ID = "Invest", Pat.ID = "Pat.ID")

# Obtain a summary of the results
summary(Fit.Neg.Pos)
#

## End(Not run)
```

new_vine_copula_fit *Constructor for vine copula model*

Description

Constructor for vine copula model

Usage

```
new_vine_copula_fit(fit_0, fit_1, endpoint_types)
```

Arguments

`fit_0` list returned by `fit_copula_submodel_OrdCont()`, `fit_copula_submodel_ContCont()`, or `fit_copula_submodel_OrdOrd()`.

`fit_1` list returned by `fit_copula_submodel_OrdCont()`, `fit_copula_submodel_ContCont()`, or `fit_copula_submodel_OrdOrd()`.

`endpoint_types` Character vector with 2 elements indicating the type of endpoints. Each element is either `"ordinal"` or `"continuous"`.

Value

S3 object of the class `vine_copula_fit`.

See Also

`print.vine_copula_fit()`, `plot.vine_copula_fit()` #should not be used be the user

`new_vine_copula_ss_fit` *Constructor for vine copula model*

Description

Constructor for vine copula model

Usage

```
new_vine_copula_ss_fit(
  fit_0,
  fit_1,
  copula_family,
  knots0,
  knots1,
  knott0,
  knott1,
  copula_rotations,
  data
)
```

Arguments

`fit_0` Estimated parameters in the control group.

`fit_1` Estimated parameters in the experimental group

`copula_family` Parametric copula family

`knots0` placement of knots for Royston-Parmar model

`knots1` placement of knots for Royston-Parmar model

`knott0`	placement of knots for Royston-Parmar model
`knott1`	placement of knots for Royston-Parmar model
`copula_rotations`	vector of copula rotation parameters
`data`	Original data

Value

S3 object

Examples

```
#should not be used be the user
```

`ordinal_continuous_loglik`

Loglikelihood function for ordinal-continuous copula model

Description

`ordinal_continuous_loglik()` computes the observed-data loglikelihood for a bivariate copula model with a continuous and an ordinal endpoint. The model is based on a latent variable representation of the ordinal endpoint.

Usage

```
ordinal_continuous_loglik(
  para,
  X,
  Y,
  copula_family,
  marginal_Y,
  K,
  return_sum = TRUE
)
```

Arguments

`para` Parameter vector. The parameters are ordered as follows:

- `para[1:p1]`: Cutpoints for the latent distribution of X corresponding to $c_1, \dots, c_{K-1}$ (see Details).
- `para[(p1 + 1):(p1 + p2)]`: Parameters for surrogate distribution, more details in `?Surrogate::cdf_fun` for the specific implementations.
- `para[p1 + p2 + 1]`: copula parameter

`X` First variable (Ordinal with K categories)

`Y` Second variable (Continuous)

`copula_family` Copula family, one of the following:

- `"clayton"`
- `"frank"`
- `"gumbel"`
- `"gaussian"`

`marginal_Y` List with the following five elements (in order):

- Density function with first argument `x` and second argument `para` the parameter vector for this distribution.
- Distribution function with first argument `x` and second argument `para`.
- Inverse distribution function with first argument `p` and second argument `para`.
- The number of elements in `para`.
- Starting values for `para`.

`K` Number of categories in `X`.

`return_sum` Return the sum of the individual loglikelihoods? If `FALSE`, a vector with the individual loglikelihood contributions is returned.

Details

Vine Copula Model for Ordinal Endpoints:

Following the Neyman-Rubin potential outcomes framework, we assume that each patient has four potential outcomes, two for each arm, represented by $\boldsymbol{Y} = (T_0, S_0, S_1, T_1)'$. Here, $\boldsymbol{Y_z} = (S_z, T_z)'$ are the potential surrogate and true endpoints under treatment $Z = z$. We will further assume that T is ordinal and S is continuous; consequently, the function argument `X` corresponds to T and `Y` to S. (The roles of S and T can be interchanged without loss of generality.)

We introduce latent variables to model $\boldsymbol{Y}$. Latent variables will be denoted by a tilde. For instance, if T_z is ordinal with K_T categories, then T_z is a function of the latent $\tilde{T}_z \sim N(0, 1)$ as follows:

$$T_z = g_{T_z}(\tilde{T}_z; \boldsymbol{c}^{T_z}) = \begin{cases} 1 & \text{if } -\infty = c_0^{T_z} < \tilde{T}_z \leq c_1^{T_z} \\ \vdots & \\ k & \text{if } c_{k-1}^{T_z} < \tilde{T}_z \leq c_k^{T_z} \\ \vdots & \\ K & \text{if } c_{K_T-1}^{T_z} < \tilde{T}_z \leq c_{K_T}^{T_z} = \infty, \end{cases}$$

where $\boldsymbol{c}^{T_z} = (c_1^{T_z}, \cdots, c_{K_T-1}^{T_z})$. The latent counterpart of $\boldsymbol{Y}$ is again denoted by a tilde; for example, $\tilde{\boldsymbol{Y}} = (\tilde{T}_0, S_0, S_1, \tilde{T}_1)'$ if T_z is ordinal and S_z is continuous.

The vector of latent potential outcome $\tilde{\boldsymbol{Y}}$ is modeled with a D-vine copula as follows:

$$f_{\tilde{\boldsymbol{Y}}} = f_{\tilde{T}_0}\, f_{S_0}\, f_{S_1}\, f_{\tilde{T}_1} \cdot c_{\tilde{T}_0,S_0}\, c_{S_0,S_1}\, c_{S_1,\tilde{T}_1} \cdot c_{\tilde{T}_0,S_1;S_0}\, c_{S_0,\tilde{T}_1;S_1} \cdot c_{\tilde{T}_0,\tilde{T}_1;S_0,S_1},$$

where (i) f_{T_0}, f_{S_0}, f_{S_1}, and f_{T_1} are univariate density functions, (ii) c_{T_0,S_0}, c_{S_0,S_1}, and c_{S_1,T_1} are unconditional bivariate copula densities, and (iii) $c_{T_0,S_1;S_0}$, $c_{S_0,T_1;S_1}$, and $c_{T_0,T_1;S_0,S_1}$ are conditional bivariate copula densities (e.g., $c_{T_0,S_1;S_0}$ is the copula density of $(T_0, S_1)' \mid S_0$. We also make the simplifying assumption for all copulas.

Observed-Data Likelihood:

In practice, we only observe $(S_0, T_0)'$ or $(S_1, T_1)'$. Hence, to estimate the (identifiable) parameters of the D-vine copula model, we need to derive the observed-data likelihood. The observed-data loglikelihood for $(S_z, T_z)'$ is as follows:

$$f_{\boldsymbol{Y_z}}(s, t; \boldsymbol{\beta}) = \int_{c_{t-1}^{T_z}}^{+\infty} f_{\tilde{\boldsymbol{Y}}_{\boldsymbol{z}}}(s, x; \boldsymbol{\beta}) \, dx - \int_{c_t^{T_z}}^{+\infty} f_{\tilde{\boldsymbol{Y}}_{\boldsymbol{z}}}(s, x; \boldsymbol{\beta}) \, dx.$$

The above expression is used in `ordinal_continuous_loglik()` to compute the loglikelihood for the observed values for $Z = 0$ or $Z = 1$. In this function, `X` and `Y` correspond to T_z and S_z if T_z is ordinal and S_z continuous. Otherwise, `X` and `Y` correspond to S_z and T_z.

Value

(numeric) loglikelihood value evaluated in `para`.

`ordinal_ordinal_loglik` *Loglikelihood function for ordinal-ordinal copula model*

Description

`ordinal_ordinal_loglik()` computes the observed-data loglikelihood for a bivariate copula model with two ordinal endpoints. The model is based on a latent variable representation of the ordinal endpoints.

Usage

```
ordinal_ordinal_loglik(para, X, Y, copula_family, K_X, K_Y, return_sum = TRUE)
```

Arguments

`para` Parameter vector. The parameters are ordered as follows:

- `para[1:p1]`: Cutpoints for the latent distribution of X corresponding to $c_1^X, \dots, c_{K_X - 1}^X$ (see Details).
- `para[(p1 + 1):(p1 + p2)]`: Cutpoints for the latent distribution of Y corresponding to $c_1^Y, \dots, c_{K_Y - 1}^Y$ (see Details).
- `para[p1 + p2 + 1]`: copula parameter

`X` First variable (Ordinal with K_X categories)

`Y` Second variable (Ordinal with K_Y categories)

`copula_family` Copula family, one of the following:

- `"clayton"`
- `"frank"`
- `"gumbel"`
- `"gaussian"`

`K_X`	Number of categories in `X`.
`K_Y`	Number of categories in `Y`.
`return_sum`	Return the sum of the individual loglikelihoods? If `FALSE`, a vector with the individual loglikelihood contributions is returned.

Details

Vine Copula Model for Ordinal Endpoints:

Following the Neyman-Rubin potential outcomes framework, we assume that each patient has four potential outcomes, two for each arm, represented by $\boldsymbol{Y} = (T_0, S_0, S_1, T_1)'$. Here, $\boldsymbol{Y_z} = (S_z, T_z)'$ are the potential surrogate and true endpoints under treatment $Z = z$.

The latent variable notation and D-vine copula model for $\boldsymbol{Y}$ is a straightforward extension of the notation in `ordinal_continuous_loglik()`.

Observed-Data Likelihood:

In practice, we only observe $(S_0, T_0)'$ or $(S_1, T_1)'$. Hence, to estimate the (identifiable) parameters of the D-vine copula model, we need to derive the observed-data likelihood. The observed-data loglikelihood for $(S_z, T_z)'$ is as follows:

$$f_{\boldsymbol{Y_z}}(s, t; \boldsymbol{\beta}) = P\left(c^{S_z}_{s-1} < \tilde{S}_z, c^{T_z}_{t-1} < \tilde{T}_z\right) - P\left(c^{S_z}_{s} < \tilde{S}_z, c^{T_z}_{t-1} < \tilde{T}_z\right) - P\left(c^{S_z}_{s-1} < \tilde{S}_z, c^{T_z}_{t} < \tilde{T}_z\right) + P\Big(c^{S_z}_{s} < \tilde{S}_z, c^{T_z}_{t} <$$

The above expression is used in `ordinal_ordinal_loglik()` to compute the loglikelihood for the observed values for $Z = 0$ or $Z = 1$.

Value

(numeric) loglikelihood value evaluated in `para`.

`ordinal_to_cutpoints` *Convert Ordinal Observations to Latent Cutpoints*

Description

`ordinal_to_cutpoints()` converts the ordinal endpoints to the corresponding cutpoints of the underlying latent continuous variable. Let $P(x \leq k) = G(c_k)$ where G is the distribution function of the latent variable. `ordinal_to_cutpoints()` converts x to c_k (or to c_{k-1}) if `strict = TRUE`.

Usage

```
ordinal_to_cutpoints(x, cutpoints, strict)
```

Arguments

`x`	Integer vector with values in `1:(length(cutpoints) + 1)`.
`cutpoints`	The cutpoints on the latent scale corresponding to $\boldsymbol{c} = c(c_1, \cdots, c_{K-1})$.
`strict`	(boolean) See function description.

Value

Numeric vector with cutpoints corresponding to the values in `x`.

`Ovarian` *The Ovarian dataset*

Description

This dataset combines the data that were collected in four double-blind randomized clinical trials in advanced ovarian cancer (Ovarian Cancer Meta-Analysis Project, 1991). In these trials, the objective was to examine the efficacy of cyclophosphamide plus cisplatin (CP) versus cyclophosphamide plus adriamycin plus cisplatin (CAP) to treat advanced ovarian cancer.

Usage

```
data("Ovarian")
```

Format

A data frame with 1192 observations on the following 7 variables.

`Patient` The ID number of a patient.

`Center` The center in which a patient was treated.

`Treat` The treatment indicator, coded as 0=CP (active control) and 1=CAP (experimental treatment).

`Pfs` Progression-free survival (the candidate surrogate).

`PfsInd` Censoring indicator for progression-free survival.

`Surv` Survival time (the true endpoint).

`SurvInd` Censoring indicator for survival time.

References

Ovarian Cancer Meta-Analysis Project (1991). Cclophosphamide plus cisplatin plus adriamycin versus cyclophosphamide, doxorubicin, and cisplatin chemotherapy of ovarian carcinoma: a meta-analysis. *Classic papers and current comments, 3,* 237-234.

Examples

```
data(Ovarian)
str(Ovarian)
head(Ovarian)
```

PANSS *PANSS subscales and total score based on the data of five clinical trials in schizophrenia*

Description

These are the PANSS subscale and total scale scores of five clinical trial in schizophrenia. A total of 1941 patients were treated by 126 investiagators (psychiatrists). There were two treatment conditions (risperidone and control). Patients' schizophrenic symptoms were measured using the PANSS (Kay et al., 1988).

Usage

```
data(PANSS)
```

Format

A `data.frame` with 1941 observations on 9 variables.

`Pat.Id` The patient ID.

`Treat` The treatment indicator, coded as -1 = active control and 1 = Risperidone.

`Invest` The ID of the investigator (psychiatrist) who treated the patient.

`Neg` The Negative symptoms scale score.

`Exc` The Excitement scale score.

`Cog` The Cognition scale score.

`Pos` The Positive symptoms scale score.

`Dep` The Depression scale score.

`Total` The Total PANSS score.

References

Kay, S.R., Opler, L.A., & Lindenmayer, J.P. (1988). Reliability and validity of the Positive and Negative Syndrome Scale for schizophrenics. Psychiatric Research, 23, 99-110.

pdf_fun *Function factory for density functions*

Description

Function factory for density functions

Usage

```
pdf_fun(para, family)
```

Arguments

para	Parameter vector.
family	Distributional family, one of the following: - "normal": normal distribution where para[1] is the mean and para[2] is the standard deviation. - "logistic": logistic distribution as parameterized in stats::plogis() where para[1] and para[2] correspond to location and scale, respectively. - "t": t distribution as parameterized in stats::pt() where para[1] and para[2] correspond to ncp and df, respectively.

Value

A density function that has a single argument. This is the vector of values in which the density function is evaluated.

plot Causal-Inference BinBin *Plots the (Meta-Analytic) Individual Causal Association and related metrics when S and T are binary outcomes*

Description

This function provides a plot that displays the frequencies, percentages, cumulative percentages or densities of the individual causal association (ICA; R_H^2 or R_H), and/or the odds ratios for S and T (θ_S and θ_T).

Usage

```
## S3 method for class 'ICA.BinBin'
plot(x, R2_H=TRUE, R_H=FALSE, Theta_T=FALSE,
Theta_S=FALSE, Type="Density", Labels=FALSE, Xlab.R2_H,
Main.R2_H, Xlab.R_H, Main.R_H, Xlab.Theta_S, Main.Theta_S, Xlab.Theta_T,
Main.Theta_T, Cex.Legend=1, Cex.Position="topright",
col, Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ylim, ...)
```

Arguments

`x` An object of class `ICA.BinBin`. See `ICA.BinBin`.

`R2_H` Logical. When `R2_H=TRUE`, a plot of the R_H^2 is provided. Default `TRUE`.

`R_H` Logical. When `R_H=TRUE`, a plot of the R_H is provided. Default `FALSE`.

`Theta_T` Logical. When `Theta_T=TRUE`, a plot of the θ_T is provided. Default `FALSE`.

`Theta_S` Logical. When `Theta_S=TRUE`, a plot of the θ_S is provided. Default `FALSE`.

`Type` The type of plot that is produced. When `Type="Freq"` or `Type="Percent"`, the Y-axis shows frequencies or percentages of R_H^2, R_H, θ_T, or θ_S. When `Type="CumPerc"`, the Y-axis shows cumulative percentages. When `Type="Density"`, the density is shown. When the fitted object of class `ICA.BinBin` was obtained using a general analysis (i.e., using the `Monotonicity=c("General")` argument in the function call), sperate plots are provided for the different monotonicity scenarios. Default `"Density"`.

`Labels` Logical. When `Labels=TRUE`, the percentage of R_H^2, R_H, θ_T, or θ_S values that are equal to or larger than the midpoint value of each of the bins are displayed (on top of each bin). Default `FALSE`.

`Xlab.R2_H` The legend of the X-axis of the R_H^2 plot.

`Main.R2_H` The title of the R_H^2 plot.

`Xlab.R_H` The legend of the X-axis of the R_H plot.

`Main.R_H` The title of the R_H plot.

`Xlab.Theta_S` The legend of the X-axis of the θ_S plot.

`Main.Theta_S` The title of the θ_S plot.

`Xlab.Theta_T` The legend of the X-axis of the θ_T plot.

`Main.Theta_T` The title of the θ_T plot.

`Cex.Legend` The size of the legend when `Type="All.Densities"` is used. Default `Cex.Legend=1`.

`Cex.Position` The position of the legend, `Cex.Position="topright"` or `Cex.Position="topleft"`. Default `Cex.Position="topright"`.

`col` The color of the bins. Default `col <- c(8)`.

`Par` Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.

`ylim` The (min, max) values for the Y-axis

.

`...` Extra graphical parameters to be passed to `hist()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). A causal-inference approach for the validation of surrogate endpoints based on information theory and sensitivity analysis.

See Also

ICA.BinBin

Examples

```
# Compute R2_H given the marginals,
# assuming monotonicity for S and T and grids
# pi_0111=seq(0, 1, by=.001) and
# pi_1100=seq(0, 1, by=.001)
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.261, pi1_0_=0.285,
pi_1_1=0.637, pi_1_0=0.078, pi0_1_=0.134, pi_0_1=0.127,
Monotonicity=c("General"), M=2500, Seed=1)

# Plot the results (density of R2_H):
plot(ICA, Type="Density", R2_H=TRUE, R_H=FALSE,
Theta_T=FALSE, Theta_S=FALSE)
```

plot Causal-Inference ContCont

Plots the (Meta-Analytic) Individual Causal Association when S and T are continuous outcomes

Description

This function provides a plot that displays the frequencies, percentages, or cumulative percentages of the individual causal association (ICA; ρ_Δ) and/or the meta-analytic individual causal association (MICA; ρ_M) values. These figures are useful to examine the sensitivity of the obtained results with respect to the assumptions regarding the correlations between the counterfactuals (for details, see Alonso et al., submitted; Van der Elst et al., submitted). Optionally, it is also possible to obtain plots that are useful in the examination of the plausibility of finding a good surrogate endpoint when an object of class `ICA.ContCont` is considered.

Usage

```
## S3 method for class 'ICA.ContCont'
plot(x, Xlab.ICA, Main.ICA, Type="Percent",
Labels=FALSE, ICA=TRUE, Good.Surr=FALSE, Main.Good.Surr,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), col, ...)

## S3 method for class 'MICA.ContCont'
plot(x, ICA=TRUE, MICA=TRUE, Type="Percent",
Labels=FALSE, Xlab.ICA, Main.ICA, Xlab.MICA, Main.MICA,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), col, ...)
```

Arguments

`x` An object of class `ICA.ContCont` or `MICA.ContCont`. See `ICA.ContCont` or `MICA.ContCont`.

`ICA` Logical. When `ICA=TRUE`, a plot of the ICA is provided. Default `TRUE`.

`MICA` Logical. This argument only has effect when the `plot()` function is applied to an object of class `MICA.ContCont`. When `MICA=TRUE`, a plot of the MICA is provided. Default `TRUE`.

`Type` The type of plot that is produced. When `Type=Freq` or `Type=Percent`, the Y-axis shows frequencies or percentages of ρ_Δ, ρ_M, and/or δ. When `Type=CumPerc`, the Y-axis shows cumulative percentages of ρ_Δ, ρ_M, and/or δ. Default "Percent".

`Labels` Logical. When `Labels=TRUE`, the percentage of ρ_Δ, ρ_M, and/or δ values that are equal to or larger than the midpoint value of each of the bins are displayed (on top of each bin). Default `FALSE`.

`Xlab.ICA` The legend of the X-axis of the ICA plot. Default "ρ_Δ".

`Main.ICA` The title of the ICA plot. Default "ICA".

`Xlab.MICA` The legend of the X-axis of the MICA plot. Default "ρ_M".

`Main.MICA` The title of the MICA plot. Default "MICA".

`Good.Surr` Logical. When `Good.Surr=TRUE`, a plot of δ is provided. This plot is useful in the context of examinating the plausibility of finding a good surrogate endpoint. Only applies when an object of class `ICA.ContCont` is considered. For details, see Alonso et al. (submitted). Default `FALSE`.

`Main.Good.Surr` The title of the plot of δ. Only applies when an object of class `ICA.ContCont` is considered. For details, see Alonso et al. (submitted).

`Par` Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.

`col` The color of the bins. Default `col <- c(8)`.

`...` Extra graphical parameters to be passed to `hist()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal inference and meta-analytic paradigms for the validation of surrogate markers.

Van der Elst, W., Alonso, A., & Molenberghs, G. (submitted). An exploration of the relationship between causal inference and meta-analytic measures of surrogacy.

See Also

ICA.ContCont, MICA.ContCont, plot MinSurrContCont

Examples

```
# Plot of ICA

# Generate the vector of ICA values when rho_T0S0=rho_T1S1=.95, and when the
# grid of values {0, .2, ..., 1} is considered for the correlations
# between the counterfactuals:
SurICA <- ICA.ContCont(T0S0=.95, T1S1=.95, T0T1=seq(0, 1, by=.2), T0S1=seq(0, 1, by=.2),
T1S0=seq(0, 1, by=.2), S0S1=seq(0, 1, by=.2))

# Plot the results:
plot(SurICA)

# Same plot but add the percentages of ICA values that are equal to or larger
# than the midpoint values of the bins
plot(SurICA, Labels=TRUE)

# Plot of both ICA and MICA

# Generate the vector of ICA and MICA values when R_trial=.8, rho_T0S0=rho_T1S1=.8,
# D.aa=5, D.bb=10, and when the grid of values {0, .2, ..., 1} is considered
# for the correlations between the counterfactuals:
SurMICA <- MICA.ContCont(Trial.R=.80, D.aa=5, D.bb=10, T0S0=.8, T1S1=.8,
T0T1=seq(0, 1, by=.2), T0S1=seq(0, 1, by=.2), T1S0=seq(0, 1, by=.2),
S0S1=seq(0, 1, by=.2))

# Plot the vector of generated ICA and MICA values
plot(SurMICA, ICA=TRUE, MICA=TRUE)
```

---

plot FixedDiscrDiscrIT

Provides plots of trial-level surrogacy in the Information-Theoretic framework

---

Description

Produces plots that provide a graphical representation of trial level surrogacy R^2_{ht} based on the Information-Theoretic approach of Alonso & Molenberghs (2007).

Usage

```
 ## S3 method for class 'FixedDiscrDiscrIT'
plot(x, Weighted=TRUE, Xlab.Trial, Ylab.Trial, Main.Trial,
 Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

x	An object of class `FixedDiscrDiscrIT`.

`Weighted` Logical. This argument only has effect when the user requests a trial-level surrogacy plot (i.e., when `Trial.Level=TRUE` in the function call). If `Weighted=TRUE`, the circles that depict the trial-specific treatment effects on the true endpoint against the surrogate endpoint are proportional to the number of patients in the trial. If `Weighted=FALSE`, all circles have the same size. Default `TRUE`.

`Xlab.Trial` The legend of the X-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the surrogate endpoint (α_i)".

`Ylab.Trial` The legend of the Y-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the true endpoint (β_i)".

`Main.Trial` The title of the plot that depicts trial-level surrogacy. Default "Trial-level surrogacy".

`Par` Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0),mar=c(5.1, 4.1, 4.1, 2.1))`.

`...` Extra graphical parameters to be passed to `plot()`.

Author(s)

Hannah M. Ensor & Christopher J. Weir

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

See Also

FixedDiscrDiscrIT

Examples

```
## Not run:  # Time consuming (>5sec) code part
# Simulate the data:
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8, R.Indiv.Target=.8,
             Seed=123, Model="Full")

# create a binary true and ordinal surrogate outcome
Data.Observed.MTS$True<-findInterval(Data.Observed.MTS$True,
        c(quantile(Data.Observed.MTS$True,0.5)))
Data.Observed.MTS$Surr<-findInterval(Data.Observed.MTS$Surr,
       c(quantile(Data.Observed.MTS$Surr,0.333),quantile(Data.Observed.MTS$Surr,0.666)))

# Assess surrogacy based on a full fixed-effect model
# in the information-theoretic framework for a binary surrogate and ordinal true outcome:
SurEval <- FixedDiscrDiscrIT(Dataset=Data.Observed.MTS, Surr=Surr, True=True, Treat=Treat,
Trial.ID=Trial.ID, Setting="ordbin")

## Request trial-level surrogacy plot. In the trial-level plot,
## make the size of the circles proportional to the number of patients in a trial:
plot(SurEval, Weighted=FALSE)
```

```
## End(Not run)
```

---

plot ICA.ContCont.MultS

Plots the Individual Causal Association in the setting where there are multiple continuous S and a continuous T

Description

This function provides a plot that displays the frequencies, percentages, or cumulative percentages of the multivariate individual causal association (R_H^2). These figures are useful to examine the sensitivity of the obtained results with respect to the assumptions regarding the correlations between the counterfactuals.

Usage

```
  ## S3 method for class 'ICA.ContCont.MultS'
plot(x, R2_H=FALSE, Corr.R2_H=TRUE,
   Type="Percent", Labels=FALSE,
   Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), col,
   Prediction.Error.Reduction=FALSE, ...)
```

Arguments

`x`	An object of class `ICA.ContCont.MultS`. See `ICA.ContCont.MultS` or `ICA.ContCont.MultS_alt`.
`R2_H`	Should a plot of the R_H^2 be provided? Default `FALSE`.
`Corr.R2_H`	Should a plot of the corrected R_H^2 be provided? Default `TRUE`.
`Type`	The type of plot that is produced. When `Type=Freq` or `Type=Percent`, the Y-axis shows frequencies or percentages of R_H^2. When `Type=CumPerc`, the Y-axis shows cumulative percentages of R_H^2. Default "Percent".
`Labels`	Logical. When `Labels=TRUE`, the percentage of R_H^2 values that are equal to or larger than the midpoint value of each of the bins are displayed (on top of each bin). Default `FALSE`.
`Par`	Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.
`col`	The color of the bins. Default `col <- c(8)`.
`Prediction.Error.Reduction`	Should a plot be shown that shows the prediction error (reisdual error) in predicting $Delta$T using an intercept only model, and that shows the prediction error (reisdual error) in predicting $Delta$T using $DeltaS_1$, $DeltaS_2$, ...? Default `Prediction.Error.Reduction=FALSE`.
`...`	Extra graphical parameters to be passed to `hist()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Van der Elst, W., Alonso, A. A., & Molenberghs, G. (2017). Univariate versus multivariate surrogate endpoints.

See Also

ICA.ContCont, ICA.ContCont.MultS, ICA.ContCont.MultS_alt, MICA.ContCont, plot MinSurrContCont

Examples

```
## Not run:  #time-consuming code parts
# Specify matrix Sigma (var-cavar matrix T_0, T_1, S1_0, S1_1, ...)
# here for 1 true endpoint and 3 surrogates

s<-matrix(rep(NA, times=64),8)
s[1,1] <- 450; s[2,2] <- 413.5; s[3,3] <- 174.2; s[4,4] <- 157.5;
s[5,5] <- 244.0; s[6,6] <- 229.99; s[7,7] <- 294.2; s[8,8] <- 302.5
s[3,1] <- 160.8; s[5,1] <- 208.5; s[7,1] <- 268.4
s[4,2] <- 124.6; s[6,2] <- 212.3; s[8,2] <- 287.1
s[5,3] <- 160.3; s[7,3] <- 142.8
s[6,4] <- 134.3; s[8,4] <- 130.4
s[7,5] <- 209.3;
s[8,6] <- 214.7
s[upper.tri(s)] = t(s)[upper.tri(s)]

# Marix looks like:
#            T_0    T_1  S1_0  S1_1  S2_0   S2_1  S2_0  S2_1
#            [,1]  [,2]  [,3]  [,4]  [,5]   [,6]  [,7]  [,8]
# T_0  [1,] 450.0    NA 160.8    NA 208.5     NA 268.4    NA
# T_1  [2,]    NA 413.5    NA 124.6    NA 212.30    NA 287.1
# S1_0 [3,] 160.8    NA 174.2    NA 160.3     NA 142.8    NA
# S1_1 [4,]    NA 124.6    NA 157.5    NA 134.30    NA 130.4
# S2_0 [5,] 208.5    NA 160.3    NA 244.0     NA 209.3    NA
# S2_1 [6,]    NA 212.3    NA 134.3    NA 229.99    NA 214.7
# S3_0 [7,] 268.4    NA 142.8    NA 209.3     NA 294.2    NA
# S3_1 [8,]    NA 287.1    NA 130.4    NA 214.70    NA 302.5

# Conduct analysis
ICA <- ICA.ContCont.MultS(M=100, N=200, Show.Progress = TRUE,
  Sigma=s, G = seq(from=-1, to=1, by = .00001), Seed=c(123),
  Model = "Delta_T ~ Delta_S1 + Delta_S2 + Delta_S3")

# Explore results
summary(ICA)
plot(ICA)

## End(Not run)
```

plot Information-Theoretic

Provides plots of trial- and individual-level surrogacy in the Information-Theoretic framework

Description

Produces plots that provide a graphical representation of trial- and/or individual-level surrogacy (R2_ht and R2_h) based on the Information-Theoretic approach of Alonso & Molenberghs (2007).

Usage

```
## S3 method for class 'FixedContContIT'
plot(x, Trial.Level=TRUE, Weighted=TRUE, Indiv.Level=TRUE,
Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial, Main.Trial, Main.Indiv,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)

## S3 method for class 'MixedContContIT'
plot(x, Trial.Level=TRUE, Weighted=TRUE, Indiv.Level=TRUE,
Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial, Main.Trial, Main.Indiv,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

`x`	An object of class `MixedContContIT` or `FixedContContIT`.
`Trial.Level`	Logical. If `Trial.Level=TRUE`, a plot of the trial-specific treatment effects on the true endpoint against the trial-specific treatment effect on the surrogate endpoints is provided (as a graphical representation of R_{ht}). Default `TRUE`.
`Weighted`	Logical. This argument only has effect when the user requests a trial-level surrogacy plot (i.e., when `Trial.Level=TRUE` in the function call). If `Weighted=TRUE`, the circles that depict the trial-specific treatment effects on the true endpoint against the surrogate endpoint are proportional to the number of patients in the trial. If `Weighted=FALSE`, all circles have the same size. Default `TRUE`.
`Indiv.Level`	Logical. If `Indiv.Level=TRUE`, a plot of the trial- and treatment-corrected residuals of the true and surrogate endpoints is provided. This plot provides a graphical representation of R_h. Default `TRUE`.
`Xlab.Indiv`	The legend of the X-axis of the plot that depicts individual-level surrogacy. Default "Residuals for the surrogate endpoint (ε_{Sij})".
`Ylab.Indiv`	The legend of the Y-axis of the plot that depicts individual-level surrogacy. Default "Residuals for the true endpoint (ε_{Tij})".
`Xlab.Trial`	The legend of the X-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the surrogate endpoint (α_i)".
`Ylab.Trial`	The legend of the Y-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the true endpoint (β_i)".

`Main.Indiv` The title of the plot that depicts individual-level surrogacy. Default "Individual-level surrogacy".

`Main.Trial` The title of the plot that depicts trial-level surrogacy. Default "Trial-level surrogacy".

`Par` Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.

`...` Extra graphical parameters to be passed to `plot()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

See Also

MixedContContIT, FixedContContIT

Examples

```
## Not run:
## Load ARMD dataset
data(ARMD)

## Conduct a surrogacy analysis, using a weighted reduced univariate fixed effect model:
Sur <- MixedContContIT(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Trial.ID=Center,
Pat.ID=Id, Model=c("Full"))

## Request both trial- and individual-level surrogacy plots. In the trial-level plot,
## make the size of the circles proportional to the number of patients in a trial:
plot(Sur, Trial.Level=TRUE, Weighted=TRUE, Indiv.Level=TRUE)

## Make a trial-level surrogacy plot using filled blue circles that
## are transparent (to make sure that the results of overlapping trials remain
## visible), and modify the title and the axes labels of the plot:
plot(Sur, pch=16, col=rgb(.3, .2, 1, 0.3), Indiv.Level=FALSE, Trial.Level=TRUE,
Weighted=TRUE, Main.Trial=c("Trial-level surrogacy (ARMD dataset)"),
Xlab.Trial=c("Difference in vision after 6 months (Surrogate)"),
Ylab.Trial=c("Difference in vision after 12 months (True enpoint)"))

## Add the estimated R2_ht value in the previous plot at position (X=-2.2, Y=0)
## (the previous plot should not have been closed):
R2ht <- format(round(as.numeric(Sur$R2ht[1]), 3))
text(x=-2.2, y=0, cex=1.4, labels=(bquote(paste("R"[ht]^{2}, "="~.(R2ht)))))

## Make an Individual-level surrogacy plot with red squares to depict individuals
## (rather than black circles):
plot(Sur, pch=15, col="red", Indiv.Level=TRUE, Trial.Level=FALSE)
```

```
## End(Not run)
```

`plot Information-Theoretic BinCombn`

Provides plots of trial- and individual-level surrogacy in the Information-Theoretic framework when both S and T are binary, or when S is binary and T is continuous (or vice versa)

Description

Produces plots that provide a graphical representation of trial- and/or individual-level surrogacy (R2_ht and R2_hInd per cluster) based on the Information-Theoretic approach of Alonso & Molenberghs (2007).

Usage

```
## S3 method for class 'FixedBinBinIT'
plot(x, Trial.Level=TRUE, Weighted=TRUE, Indiv.Level.By.Trial=TRUE,
Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial, Main.Trial, Main.Indiv,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)

## S3 method for class 'FixedBinContIT'
plot(x, Trial.Level=TRUE, Weighted=TRUE, Indiv.Level.By.Trial=TRUE,
Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial, Main.Trial, Main.Indiv,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)

## S3 method for class 'FixedContBinIT'
plot(x, Trial.Level=TRUE, Weighted=TRUE, Indiv.Level.By.Trial=TRUE,
Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial, Main.Trial, Main.Indiv,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

`x` An object of class `FixedBinBinIT`, `FixedBinContIT`, or `FixedContBinIT`.

`Trial.Level` Logical. If `Trial.Level=TRUE`, a plot of the trial-specific treatment effects on the true endpoint against the trial-specific treatment effect on the surrogate endpoints is provided (as a graphical representation of R_{ht}). Default `TRUE`.

`Weighted` Logical. This argument only has effect when the user requests a trial-level surrogacy plot (i.e., when `Trial.Level=TRUE` in the function call). If `Weighted=TRUE`, the circles that depict the trial-specific treatment effects on the true endpoint against the surrogate endpoint are proportional to the number of patients in the trial. If `Weighted=FALSE`, all circles have the same size. Default `TRUE`.

`Indiv.Level.By.Trial`
Logical. If `Indiv.Level.By.Trial=TRUE`, a plot that shows the estimated $R^2_{h.ind}$ for each trial (and confidence intervals) is provided. Default `TRUE`.

`Xlab.Indiv` The legend of the X-axis of the plot that depicts the estimated $R^2_{h.ind}$ per trial. Default "$R[h.ind]^2$.

`Ylab.Indiv` The legend of the Y-axis of the plot that shows the estimated $R^2_{h.ind}$ per trial. Default "Trial".

`Xlab.Trial` The legend of the X-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the surrogate endpoint (α_i)".

`Ylab.Trial` The legend of the Y-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the true endpoint (β_i)".

`Main.Indiv` The title of the plot that depicts individual-level surrogacy. Default "Individual-level surrogacy".

`Main.Trial` The title of the plot that depicts trial-level surrogacy. Default "Trial-level surrogacy".

`Par` Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.

`...` Extra graphical parameters to be passed to `plot()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

See Also

FixedBinBinIT, FixedBinContIT, FixedContBinIT

Examples

```
## Not run:  # Time consuming (>5sec) code part
# Generate data with continuous Surr and True
Sim.Data.MTS(N.Total=5000, N.Trial=50, R.Trial.Target=.9, R.Indiv.Target=.9,
             Fixed.Effects=c(0, 0, 0, 0), D.aa=10, D.bb=10, Seed=1,
             Model=c("Full"))
# Dichtomize Surr and True
Surr_Bin <- Data.Observed.MTS$Surr
Surr_Bin[Data.Observed.MTS$Surr>.5] <- 1
Surr_Bin[Data.Observed.MTS$Surr<=.5] <- 0
True_Bin <- Data.Observed.MTS$True
True_Bin[Data.Observed.MTS$True>.15] <- 1
True_Bin[Data.Observed.MTS$True<=.15] <- 0
Data.Observed.MTS$Surr <- Surr_Bin
Data.Observed.MTS$True <- True_Bin

# Assess surrogacy using info-theoretic framework
Fit <- FixedBinBinIT(Dataset = Data.Observed.MTS, Surr = Surr,
True = True, Treat = Treat, Trial.ID = Trial.ID,
Pat.ID = Pat.ID, Number.Bootstraps=100)

# Examine results
summary(Fit)
plot(Fit, Trial.Level = FALSE, Indiv.Level.By.Trial=TRUE)
plot(Fit, Trial.Level = TRUE, Indiv.Level.By.Trial=FALSE)

## End(Not run)
```

plot ISTE.ContCont *Plots the individual-level surrogate threshold effect (STE) values and related metrics*

Description

This function plots the individual-level surrogate threshold effect (STE) values and related metrics, e.g., the expected ΔT values for a vector of ΔS values.

Usage

```
## S3 method for class 'ISTE.ContCont'
plot(x, Outcome="ISTE", breaks=50, ...)
```

Arguments

`x` An object of class `ISTE.ContCont`. See `ISTE.ContCont`.

`Outcome` The outcome for which a histogram has to be produced. When `Outcome="ISTE"`, a histogram of the ISTE is produced. When `Outcome="MSE"`, a histogram of the MSE values (of regression models in which ΔT is regressed on ΔS) is given. When `Outcome="gamma0"`, a histogram of $\gamma[0]$ values (of regression models in which ΔT is regressed on ΔS) is given. When `Outcome="gamma1"`, a histogram of $\gamma[1]$ values (of regression models in which ΔT is regressed on ΔS) is given. When `Outcome="Exp.DeltaT"`, a histogram of the expected ΔT values for a vector of ΔS values (specified in the call of the `ISTE.ContCont` function) values is given. When `Outcome="Exp.DeltaT.Low.PI"`, a histogram of the lower prediction intervals of the expected ΔT values for a vector of ΔS values (specified in the call of the `ISTE.ContCont` function) values is given. When `Outcome="Exp.DeltaT.Up.PI"`, a histogram of the upper prediction intervals of the expected ΔT values for a vector of ΔS values (specified in the call of the `ISTE.ContCont` function) values is given. Dafault `Outcome="ISTE"`. When `Outcome="Delta_S_For_Which_Delta_T_equal_0"`, a histogram of $omega$ is shown with $E(DeltaT|DeltaS > omega) > 0$.

`breaks` The number of breaks used in the histogram(s). Default `breaks=50`.

`...` Extra graphical parameters to be passed to `hist()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Van der Elst, W., Alonso, A. A., and Molenberghs, G. (submitted). The individual-level surrogate threshold effect in a causal-inference setting.

See Also

ISTE.ContCont

Examples

```
# Define input for analysis using the Schizo dataset,
# with S=BPRS and T = PANSS.
# For each of the identifiable quantities,
# uncertainty is accounted for by specifying a uniform
# distribution with min, max values corresponding to
# the 95% confidence interval of the quantity.
T0S0 <- runif(min = 0.9524, max = 0.9659, n = 1000)
T1S1 <- runif(min = 0.9608, max = 0.9677, n = 1000)

S0S0 <- runif(min=160.811, max=204.5009, n=1000)
S1S1 <- runif(min=168.989, max = 194.219, n=1000)
T0T0 <- runif(min=484.462, max = 616.082, n=1000)
T1T1 <- runif(min=514.279, max = 591.062, n=1000)

Mean_T0 <- runif(min=-13.455, max=-9.489, n=1000)
Mean_T1 <- runif(min=-17.17, max=-14.86, n=1000)
Mean_S0 <- runif(min=-7.789, max=-5.503, n=1000)
Mean_S1 <- runif(min=-9.600, max=-8.276, n=1000)

# Do the ISTE analysis
## Not run:
ISTE <- ISTE.ContCont(Mean_T1=Mean_T1, Mean_T0=Mean_T0,
 Mean_S1=Mean_S1, Mean_S0=Mean_S0, N=2128, Delta_S=c(-50:50),
 alpha.PI=0.05, PI.Bound=0, Show.Prediction.Plots=TRUE,
 Save.Plots="No", T0S0=T0S0, T1S1=T1S1, T0T0=T0T0, T1T1=T1T1,
 S0S0=S0S0, S1S1=S1S1)

# Examine results:
summary(ISTE)

# Plots of results.
  # Plot main ISTE results
plot(ISTE)
  # Other plots
plot(ISTE, Outcome="MSE")
plot(ISTE, Outcome="gamma0")
plot(ISTE, Outcome="gamma1")
plot(ISTE, Outcome="Exp.DeltaT")
plot(ISTE, Outcome="Exp.DeltaT.Low.PI")
plot(ISTE, Outcome="Exp.DeltaT.Up.PI")

## End(Not run)
```

plot MaxEnt ContCont

Plots the sensitivity-based and maximum entropy based Individual Causal Association when S and T are continuous outcomes in the single-trial setting

Description

This function provides a plot that displays the frequencies or densities of the individual causal association (ICA; $rho[Delta]$) as identified based on the sensitivity- (using the functions `ICA.ContCont`) and maximum entropy-based (using the function `MaxEntContCont`) approaches.

Usage

```
## S3 method for class 'MaxEntContCont'
plot(x, Type="Freq", Xlab, col,
Main, Entropy.By.ICA=FALSE, ...)
```

Arguments

`x` An object of class `MaxEntContCont`. See `MaxEntContCont`.

`Type` The type of plot that is produced. When `Type="Freq"`, the Y-axis shows frequencies of ICA. When `Type="Density"`, the density is shown. Default `Type="Freq"`.

`Xlab` The legend of the X-axis of the plot.

`col` The color of the bins (frequeny plot) or line (density plot). Default `col <- c(8)`.

`Main` The title of the plot.

`Entropy.By.ICA` Plot with ICA on Y-axis and entropy on X-axis.

`...` Other arguments to be passed to `plot()`

Author(s)

Wim Van der Elst, Ariel Alonso, Paul Meyvisch, & Geert Molenberghs

References

Add

See Also

ICA.ContCont, MaxEntContCont

Examples

```
## Not run:  #time-consuming code parts
# Compute ICA for ARMD dataset, using the grid
# G={-1, -.80, ..., 1} for the undidentifiable correlations

ICA <- ICA.ContCont(T0S0 = 0.769, T1S1 = 0.712, S0S0 = 188.926,
S1S1 = 132.638, T0T0 = 264.797, T1T1 = 231.771,
T0T1 = seq(-1, 1, by = 0.2), T0S1 = seq(-1, 1, by = 0.2),
T1S0 = seq(-1, 1, by = 0.2), S0S1 = seq(-1, 1, by = 0.2))

# Identify the maximum entropy ICA
MaxEnt_ARMD <- MaxEntContCont(x = ICA, S0S0 = 188.926,
S1S1 = 132.638, T0T0 = 264.797, T1T1 = 231.771)

  # Explore results using summary() and plot() functions
summary(MaxEnt_ARMD)
plot(MaxEnt_ARMD)
plot(MaxEnt_ARMD, Entropy.By.ICA = TRUE)

## End(Not run)
```

plot MaxEntICA BinBin *Plots the sensitivity-based and maximum entropy based Individual Causal Association when S and T are binary outcomes*

Description

This function provides a plot that displays the frequencies or densities of the individual causal association (ICA; R_H^2) as identified based on the sensitivity- (using the functions ICA.BinBin, ICA.BinBin.Grid.Sample, or ICA.BinBin.Grid.Full) and maximum entropy-based (using the function MaxEntICABinBin) approaches.

Usage

```
## S3 method for class 'MaxEntICA.BinBin'
plot(x, ICA.Fit,
Type="Density", Xlab, col, Main, ...)
```

Arguments

x	An object of class MaxEntICABinBin. See MaxEntICABinBin.
ICA.Fit	An object of class ICA.BinBin. See ICA.BinBin.
Type	The type of plot that is produced. When Type="Freq", the Y-axis shows frequencies of R_H^2. When Type="Density", the density is shown.
Xlab	The legend of the X-axis of the plot.
col	The color of the bins (frequeny plot) or line (density plot). Default col <- c(8).
Main	The title of the plot.
...	Other arguments to be passed to plot()

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., & Van der Elst, W. (2015). A maximum-entropy approach for the evluation of surrogate endpoints based on causal inference.

See Also

ICA.BinBin, MaxEntICABinBin

Examples

```
# Sensitivity-based ICA results using ICA.BinBin.Grid.Sample
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078, Seed=1,
Monotonicity=c("No"), M=5000)

# Maximum-entropy based ICA
MaxEnt <- MaxEntICABinBin(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078)

# Plot results
plot(x=MaxEnt, ICA.Fit=ICA)
```

plot MaxEntSPF BinBin *Plots the sensitivity-based and maximum entropy based surrogate predictive function (SPF) when S and T are binary outcomes.*

Description

Plots the sensitivity-based (Alonso et al., 2015a) and maximum entropy based (Alonso et al., 2015b) surrogate predictive function (SPF), i.e., $r(i, j) = P(\Delta T = i|\Delta S = j)$, in the setting where both S and T are binary endpoints. For example, $r(-1, 1)$ quantifies the probability that the treatment has a negative effect on the true endpoint ($\Delta T = -1$) given that it has a positive effect on the surrogate ($\Delta S = 1$).

Usage

```
## S3 method for class 'MaxEntSPF.BinBin'
plot(x, SPF.Fit, Type="All.Histograms", Col="grey", ...)
```

Arguments

- `x` — A fitted object of class `MaxEntSPF.BinBin`. See `MaxEntSPFBinBin`.
- `SPF.Fit` — A fitted object of class `SPF.BinBin`. See `SPF.BinBin`.
- `Type` — The type of plot that is requested. Possible choices are: `Type="All.Histograms"`, the histograms of all 9 $r(i, j) = P(\Delta T = i|\Delta S = j)$ vectors arranged in a 3 by 3 grid; `Type="All.Densities"`, plots of densities of all $r(i, j) = P(\Delta T = i|\Delta S = j)$ vectors. Default `Type="All.Densities"`.
- `Col` — The color of the bins or lines when histograms or density plots are requested. Default `"grey"`.
- `...` — Other arguments to be passed to the `plot()` function.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2015a). Assessing a surrogate effect predictive value in a causal inference framework.

Alonso, A., & Van der Elst, W. (2015b). A maximum-entropy approach for the evluation of surrogate endpoints based on causal inference.

See Also

SPF.BinBin

Examples

```
# Sensitivity-based ICA results using ICA.BinBin.Grid.Sample
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078, Seed=1,
Monotonicity=c("No"), M=5000)

# Sensitivity-based SPF
SPFSens <- SPF.BinBin(ICA)

# Maximum-entropy based SPF
SPFMaxEnt <- MaxEntSPFBinBin(pi1_1_=0.341, pi0_1_=0.119, pi1_0_=0.254,
pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078)

# Plot results
plot(x=SPFMaxEnt, SPF.Fit=SPFSens)
```

plot Meta-Analytic — *Provides plots of trial- and individual-level surrogacy in the meta-analytic framework*

Description

Produces plots that provide a graphical representation of trial- and/or individual-level surrogacy based on the meta-analytic approach of Buyse & Molenberghs (2000) in the single- and multiple-trial settings.

Usage

```
## S3 method for class 'BifixedContCont'
plot(x, Trial.Level=TRUE, Weighted=TRUE,
Indiv.Level=TRUE, ICA=TRUE, Entropy.By.ICA=FALSE, Xlab.Indiv, Ylab.Indiv,
Xlab.Trial, Ylab.Trial, Main.Trial, Main.Indiv, Par=par(oma=c(0, 0, 0, 0),
mar=c(5.1, 4.1, 4.1, 2.1)), ...)

## S3 method for class 'BimixedContCont'
plot(x, Trial.Level=TRUE, Weighted=TRUE,
Indiv.Level=TRUE, ICA=TRUE, Entropy.By.ICA=FALSE, Xlab.Indiv, Ylab.Indiv,
Xlab.Trial, Ylab.Trial, Main.Trial, Main.Indiv, Par=par(oma=c(0, 0, 0, 0),
mar=c(5.1, 4.1, 4.1, 2.1)), ...)

## S3 method for class 'UnifixedContCont'
plot(x, Trial.Level=TRUE, Weighted=TRUE,
Indiv.Level=TRUE, ICA=TRUE, Entropy.By.ICA=FALSE,
Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial,
Main.Trial, Main.Indiv, Par=par(oma=c(0, 0, 0, 0),
mar=c(5.1, 4.1, 4.1, 2.1)), ...)

## S3 method for class 'UnimixedContCont'
plot(x, Trial.Level=TRUE, Weighted=TRUE,
Indiv.Level=TRUE, ICA=TRUE, Entropy.By.ICA=FALSE,
Xlab.Indiv, Ylab.Indiv, Xlab.Trial, Ylab.Trial,
Main.Trial, Main.Indiv, Par=par(oma=c(0, 0, 0, 0),
mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

`x` An object of class `UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`, `BimixedContCont`, or `Single.Trial.RE.AA`.

`Trial.Level` Logical. If `Trial.Level=TRUE` and an object of class `UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`, or `BimixedContCont` is considered, a plot of the trial-specific treatment effects on the true endpoint against the trial-specific treatment effect on the surrogate endpoints is provided (as a graphical representation of

R_{trial}). If `Trial.Level=TRUE` and an object of class `Single.Trial.RE.AA` is considered, a plot of the treatment effect on the true endpoint against the treatment effect on the surrogate endpoint is provided, and a regression line that goes through the origin with slope RE is added to the plot (to depict the constant RE assumption, see `Single.Trial.RE.AA` for details). If `Trial.Level=FALSE`, this plot is not provided. Default `TRUE`.

`Weighted` Logical. This argument only has effect when the user requests a trial-level surrogacy plot (i.e., when `Trial.Level=TRUE` in the function call) and when an object of class `UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`, or `BimixedContCont` is considered (not when an object of class `Single.Trial.RE.AA` is considered). If `Weighted=TRUE`, the circles that depict the trial-specific treatment effects on the true endpoint against the surrogate endpoint are proportional to the number of patients in the trial. If `Weighted=FALSE`, all circles have the same size. Default `TRUE`.

`Indiv.Level` Logical. If `Indiv.Level=TRUE`, a plot of the trial- and treatment-corrected residuals of the true and surrogate endpoints is provided (when an object of class `UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`, or `BimixedContCont` is considered), or a plot of the treatment-corrected residuals (when an object of class `Single.Trial.RE.AA` is considered). This plot provides a graphical representation of R_{indiv}. If `Indiv.Level=FALSE`, this plot is not provided. Default `TRUE`.

`ICA` Logical. Should a plot of the individual level causal association be shown? Default `ICA=TRUE`.

`Entropy.By.ICA` Logical. Should a plot that shows ICA against the entropy be shown? Default `Entropy.By.ICA=FALSE`.

`Xlab.Indiv` The legend of the X-axis of the plot that depicts individual-level surrogacy. Default "Residuals for the surrogate endpoint (ε_{Sij})" (without the i subscript when an object of class `Single.Trial.RE.AA` is considered).

`Ylab.Indiv` The legend of the Y-axis of the plot that depicts individual-level surrogacy. Default "Residuals for the true endpoint (ε_{Tij})" (without the i subscript when an object of class `Single.Trial.RE.AA` is considered).

`Xlab.Trial` The legend of the X-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the surrogate endpoint (α_i)" (without the i subscript when an object of class `Single.Trial.RE.AA` is considered).

`Ylab.Trial` The legend of the Y-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the true endpoint (β_i)" (without the i subscript when an object of class `Single.Trial.RE.AA` is considered).

`Main.Indiv` The title of the plot that depicts individual-level surrogacy. Default "Individual-level surrogacy" when an object of class `UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`, or `BimixedContCont` is considered, and "Adjusted Association (rho_Z) when an object of class `Single.Trial.RE.AA` is considered.

`Main.Trial` The title of the plot that depicts trial-level surrogacy. Default "Trial-level surrogacy" (when an object of class `UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`, or `BimixedContCont` is considered) or "Relative Effect (RE)" (when an object of class `Single.Trial.RE.AA` is considered).

Par	Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.
...	Extra graphical parameters to be passed to `plot()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

See Also

UnifixedContCont, BifixedContCont, UnifixedContCont, BimixedContCont, Single.Trial.RE.AA

Examples

```
## Not run:  # time consuming code part
##### Multiple-trial setting

## Load ARMD dataset
data(ARMD)

## Conduct a surrogacy analysis, using a weighted reduced univariate fixed effect model:
Sur <- UnifixedContCont(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Trial.ID=Center,
Pat.ID=Id, Number.Bootstraps=100, Model=c("Reduced"), Weighted=TRUE)

## Request both trial- and individual-level surrogacy plots. In the trial-level plot,
## make the size of the circles proportional to the number of patients in a trial:
plot(Sur, Trial.Level=TRUE, Weighted=TRUE, Indiv.Level=TRUE)

## Make a trial-level surrogacy plot using filled blue circles that
## are transparent (to make sure that the results of overlapping trials remain
## visible), and modify the title and the axes labels of the plot:
plot(Sur, pch=16, col=rgb(.3, .2, 1, 0.3), Indiv.Level=FALSE, Trial.Level=TRUE,
Weighted=TRUE, Main.Trial=c("Trial-level surrogacy (ARMD dataset)"),
Xlab.Trial=c("Difference in vision after 6 months (Surrogate)"),
Ylab.Trial=c("Difference in vision after 12 months (True enpoint)"))

## Add the estimated R2_trial value in the previous plot at position (X=-7, Y=11)
## (the previous plot should not have been closed):
R2trial <- format(round(as.numeric(Sur$Trial.R2[1]), 3))
text(x=-7, y=11, cex=1.4, labels=(bquote(paste("R"[trial]^{2}, "="~.(R2trial)))))

## Make an Individual-level surrogacy plot with red squares to depict individuals
## (rather than black circles):
plot(Sur, pch=15, col="red", Indiv.Level=TRUE, Trial.Level=FALSE)

## Same plot as before, but now with smaller squares, a y-axis with range [-40; 40],
## and the estimated R2_indiv value in the title of the plot:
```

```
R2ind <- format(round(as.numeric(Sur$Indiv.R2[1]), 3))
plot(Sur, pch=15, col=”red”, Indiv.Level=TRUE, Trial.Level=FALSE, cex=.5,
ylim=c(-40, 40), Main.Indiv=bquote(paste(”R”[indiv]^{2}, ”=”~.(R2ind))))


##### Single-trial setting

## Conduct a surrogacy analysis in the single-trial meta-analytic setting:
SurSTS <- Single.Trial.RE.AA(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Pat.ID=Id)

# Request a plot of individual-level surrogacy and a plot that depicts the Relative effect
# and the constant RE assumption:
plot(SurSTS, Trial.Level=TRUE, Indiv.Level=TRUE)

## End(Not run)
```

plot MinSurrContCont *Graphically illustrates the theoretical plausibility of finding a good surrogate endpoint in the continuous-continuous case*

Description

This function provides a plot that displays the frequencies, percentages, or cumulative percentages of ρ^2_{min} for a fixed value of δ (given the observed variances of the true endpoint in the control and experimental treatment conditions and a specified grid of values for the unidentified parameter $\rho_{T_0T_1}$; see MinSurrContCont). For details, see the online appendix of Alonso et al., submitted.

Usage

```
## S3 method for class 'MinSurrContCont'
plot(x, main, col, Type="Percent", Labels=FALSE,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

`x` An object of class `MinSurrContCont`. See MinSurrContCont.

`main` The title of the plot.

`col` The color of the bins.

`Type` The type of plot that is produced. When `Type=Freq` or `Type=Percent`, the Y-axis shows frequencies or percentages of ρ^2_{min}. When `Type=CumPerc`, the Y-axis shows cumulative percentages of ρ^2_{min}. Default "Percent".

`Labels` Logical. When `Labels=TRUE`, the percentage of ρ^2_{min} values that are equal to or larger than the midpoint value of each of the bins are displayed (on top of each bin). Only applies when `Type=Freq` or `Type=Percent`. Default `FALSE`.

`Par` Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.

`...` Extra graphical parameters to be passed to `hist()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal inference and meta-analytic paradigms for the validation of surrogate markers.

See Also

MinSurrContCont

Examples

```
# compute rho^2_min in the setting where the variances of T in the control
# and experimental treatments equal 100 and 120, delta is fixed at 50,
# and the grid G={0, .01, ..., 1} is considered for the counterfactual
# correlation rho_T0T1:
MinSurr <- MinSurrContCont(T0T0 = 100, T1T1 = 120, Delta = 50,
T0T1 = seq(0, 1, by = 0.01))

# Plot the results (use percentages on Y-axis)
plot(MinSurr, Type="Percent")

# Same plot, but add the percentages of ICA values that are equal to or
# larger than the midpoint values of the bins
plot(MinSurr, Labels=TRUE)
```

---

plot PredTrialTContCont

Plots the expected treatment effect on the true endpoint in a new trial (when both S and T are normally distributed continuous endpoints)

Description

The key motivation to evaluate a surrogate endpoint is to be able to predict the treatment effect on the true endpoint T based on the treatment effect on S in a new trial $i = 0$. The function `Pred.TrialT.ContCont` allows for making such predictions. The present plot function shows the results graphically.

Usage

```
## S3 method for class 'PredTrialTContCont'
plot(x, Size.New.Trial=5, CI.Segment=1, ...)
```

Arguments

`x`	A fitted object of class `Pred.TrialT.ContCont`, for details see `Pred.TrialT.ContCont`.
`Size.New.Trial`	The expected treatment effect on T is drawn as a black circle with size specified by `Size.New.Trial`. Default `Size.New.Trial=5`.
`CI.Segment`	The confidence interval around the expected treatment effect on T is depicted by a dashed horizontal line. By default, the width of the horizontal line of the horizontal section of the confidence interval indicator is 2 times the values specified by `CI.Segment`. Default $CI.Segment = 1$.
`...`	Extra graphical parameters to be passed to `plot()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

See Also

Pred.TrialT.ContCont

Examples

```
## Not run:  # time consuming code part
# Generate dataset
Sim.Data.MTS(N.Total=2000, N.Trial=15, R.Trial.Target=.95,
R.Indiv.Target=.8, D.aa=10, D.bb=50,
Fixed.Effects=c(1, 2, 30, 90), Seed=1)

# Evaluate surrogacy using a reduced bivariate mixed-effects model
BimixedFit <- BimixedContCont(Dataset = Data.Observed.MTS,
Surr = Surr, True = True, Treat = Treat, Trial.ID = Trial.ID,
Pat.ID = Pat.ID, Model="Reduced")

# Suppose that in a new trial, it was estimated alpha_0 = 30
# predict beta_0 in this trial
Pred_Beta <- Pred.TrialT.ContCont(Object = BimixedFit,
alpha_0 = 30)

# Examine the results
summary(Pred_Beta)

# Plot the results
plot(Pred_Beta)

## End(Not run)
```

plot SPF BinBin — *Plots the surrogate predictive function (SPF) in the binary-binary settinf.*

Description

Plots the surrogate predictive function (SPF), i.e., $r(i,j) = P(\Delta T = i|\Delta S = j)$, in the setting where both S and T are binary endpoints. For example, $r(-1,1)$ quantifies the probability that the treatment has a negative effect on the true endpoint ($\Delta T = -1$) given that it has a positive effect on the surrogate ($\Delta S = 1$).

Usage

```
## S3 method for class 'SPF.BinBin'
plot(x, Type="All.Histograms", Specific.Pi="r_0_0", Col="grey",
Box.Plot.Outliers=FALSE, Legend.Pos="topleft", Legend.Cex=1, ...)
```

Arguments

`x` A fitted object of class `SPF.BinBin`. See `ICA.BinBin`.

`Type` The type of plot that is requested. Possible choices are: `Type="All.Histograms"`, the histograms of all 9 $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors arranged in a 3 by 3 grid; `Type="All.Densities"`, plots of densities of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="Histogram"`, the histogram of a particular $r(i,j) = P(\Delta T = i|\Delta S = j)$ vector (the `Specific.Pi=` argument has to be used to specify the desired $r(i,j)$); `Type="Density"`, the density of a particular $r(i,j) = P(\Delta T = i|\Delta S = j)$ vector (the `Specific.Pi=` argument has to be used to specify the desired $r(i,j)$); `Type="Box.Plot"`, a box plot of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="Lines.Mean"`, a line plot the depicts the means of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="Lines.Median"`, a line plot the depicts the medians of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="Lines.Mode"`, a line plot the depicts the modes of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="3D.Mean"`, a 3D bar plot the depicts the means of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="3D.Median"`, a 3D bar plot the depicts the medians of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors; `Type="3D.Mode"`, a 3D bar plot the depicts the modes of all $r(i,j) = P(\Delta T = i|\Delta S = j)$ vectors.

`Specific.Pi` When `Type="Histogram"` or `Type="Density"` , the histogram/density of a particular $r(i,j) = P(\Delta T = i|\Delta S = j)$ vector is shown. The `Specific.Pi=` argument is used to specify the desired $r(i,j)$). Default `r_0_0`.

`Col` The color of the bins or lines when histograms or density plots are requested. Default `"grey"`.

`Box.Plot.Outliers`
Logical. Should outliers be depicted in the box plots?. Default `FALSE`.

`Legend.Pos` Position of the legend when a `type="Box.Plot"`, `type="Lines.Mean"`, `type="Lines.Median"`, or `type="Lines.Mode"` is requested. Default `"topleft"`.

`Legend.Cex` Size of the legend when a `type="Box.Plot"`, `type="Lines.Mean"`, `type="Lines.Median"`, or `type="Lines.Mode"` is requested. Default `1`.

`...` Arguments to be passed to the plot, histogram, ... functions.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2015). Assessing a surrogate effect predictive value in a causal inference framework.

See Also

SPF.BinBin

Examples

```
## Not run:
# Generate plausible values for Pi
ICA <- ICA.BinBin.Grid.Sample(pi1_1_=0.341, pi0_1_=0.119,
pi1_0_=0.254, pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078, Seed=1,
Monotonicity=c("General"), M=2500)

# Compute the surrogate predictive function (SPF)
SPF <- SPF.BinBin(ICA)

# Explore the results
summary(SPF)

# Examples of plots
plot(SPF, Type="All.Histograms")
plot(SPF, Type="All.Densities")
plot(SPF, Type="Histogram", Specific.Pi="r_0_0")
plot(SPF, Type="Box.Plot", Legend.Pos="topleft", Legend.Cex=.7)
plot(SPF, Type="Lines.Mean")
plot(SPF, Type="Lines.Median")
plot(SPF, Type="3D.Mean")
plot(SPF, Type="3D.Median")
plot(SPF, Type="3D.Spinning.Mean")
plot(SPF, Type="3D.Spinning.Median")

## End(Not run)
```

plot TrialLevelIT — *Provides a plots of trial-level surrogacy in the information-theoretic framework based on the output of the* `TrialLevelIT()` *function*

Description

Produces a plot that provides a graphical representation of trial-level surrogacy based on the output of the `TrialLevelIT()` function (information-theoretic framework).

Usage

```
## S3 method for class 'TrialLevelIT'
plot(x, Xlab.Trial,
Ylab.Trial, Main.Trial, Par=par(oma=c(0, 0, 0, 0),
mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

`x`	An object of class `TrialLevelIT`.
`Xlab.Trial`	The legend of the X-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the surrogate endpoint (α_i)".
`Ylab.Trial`	The legend of the Y-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the true endpoint (β_i)".
`Main.Trial`	The title of the plot that depicts trial-level surrogacy. Default "Trial-level surrogacy".
`Par`	Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.
`...`	Extra graphical parameters to be passed to `plot()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

See Also

UnifixedContCont, BifixedContCont, UnifixedContCont, BimixedContCont, TrialLevelIT

Examples

```
# Generate vector treatment effects on S
set.seed(seed = 1)
Alpha.Vector <- seq(from = 5, to = 10, by=.1) + runif(min = -.5, max = .5, n = 51)

# Generate vector treatment effects on T
set.seed(seed=2)
Beta.Vector <- (Alpha.Vector * 3) + runif(min = -5, max = 5, n = 51)

# Apply the function to estimate R^2_{h.t}
Fit <- TrialLevelIT(Alpha.Vector=Alpha.Vector,
Beta.Vector=Beta.Vector, N.Trial=50, Model="Reduced")

# Plot the results
plot(Fit)
```

---

plot TrialLevelMA

Provides a plots of trial-level surrogacy in the meta-analytic framework based on the output of the `TrialLevelMA()` *function*

Description

Produces a plot that provides a graphical representation of trial-level surrogacy based on the output of the `TrialLevel()` function (meta-analytic framework).

Usage

```
## S3 method for class 'TrialLevelMA'
plot(x, Weighted=TRUE, Xlab.Trial,
Ylab.Trial, Main.Trial, Par=par(oma=c(0, 0, 0, 0),
mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

`x`	An object of class `TrialLevelMA`.
`Weighted`	Logical. If `Weighted=TRUE`, the circles that depict the trial-specific treatment effects on the true endpoint against the surrogate endpoint are proportional to the number of patients in the trial. If `Weighted=FALSE`, all circles have the same size. Default `TRUE`.
`Xlab.Trial`	The legend of the X-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the surrogate endpoint (α_i)".
`Ylab.Trial`	The legend of the Y-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the true endpoint (β_i)".
`Main.Trial`	The title of the plot that depicts trial-level surrogacy. Default "Trial-level surrogacy".
`Par`	Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.
`...`	Extra graphical parameters to be passed to `plot()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

See Also

UnifixedContCont, BifixedContCont, UnifixedContCont, BimixedContCont, TrialLevelMA

Examples

```
# Generate vector treatment effects on S
set.seed(seed = 1)
Alpha.Vector <- seq(from = 5, to = 10, by=.1) + runif(min = -.5, max = .5, n = 51)
# Generate vector treatment effects on T
set.seed(seed=2)
Beta.Vector <- (Alpha.Vector * 3) + runif(min = -5, max = 5, n = 51)
# Vector of sample sizes of the trials (here, all n_i=10)
N.Vector <- rep(10, times=51)

# Apply the function to estimate R^2_{trial}
Fit <- TrialLevelMA(Alpha.Vector=Alpha.Vector,
Beta.Vector=Beta.Vector, N.Vector=N.Vector)

# Plot the results and obtain summary
plot(Fit)
summary(Fit)
```

---

plot TwoStageSurvSurv *Plots trial-level surrogacy in the meta-analytic framework when two survival endpoints are considered.*

Description

Produces a plot that graphically depicts trial-level surrogacy when the surrogate and true endpoints are survival endpoints.

Usage

```
## S3 method for class 'TwoStageSurvSurv'
plot(x, Weighted=TRUE, xlab, ylab, main,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

`x` An object of class `TwoStageContCont`.

`Weighted` Logical. If `Weighted=TRUE`, the circles that depict the trial-specific treatment effects on the true endpoint against the surrogate endpoint are proportional to the number of patients in the trial. If `Weighted=FALSE`, all circles have the same size. Default `TRUE`.

`xlab` The legend of the X-axis, default "Treatment effect on the surrogate endpoint (α_i)".

`ylab` The legend of the Y-axis, default "Treatment effect on the true endpoint (β_i)".

`main` The title of the plot, default "Trial-level surrogacy".

`Par` Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.

`...` Extra graphical parameters to be passed to `plot()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

See Also

TwoStageSurvSurv

Examples

```
# Open Ovarian dataset
data(Ovarian)
# Conduct analysis
Results <- TwoStageSurvSurv(Dataset = Ovarian, Surr = Pfs, SurrCens = PfsInd,
True = Surv, TrueCens = SurvInd, Treat = Treat, Trial.ID = Center)
# Examine results of analysis
summary(Results)
plot(Results)
```

`plot.comb27.BinBin` *Plots the distribution of prediction error functions in decreasing order of appearance.*

Description

The function `plot.comb27.BinBin` plots each of the selected prediction functions in decreasing order in the single-trial causal-inference framework when both the surrogate and the true endpoints are binary outcomes. The distribution of frequencies at which each of the 27 possible predicton functions are selected provides additional insights regarding the association between S (Δ_S) and T (Δ_T).. See **Details** below.

Usage

```
## S3 method for class 'comb27.BinBin'
plot(x,lab,...)
```

Arguments

`x`	An object of class `comb27.BinBin`. See `comb27.BinBin`.
`lab`	a supplementary label to the graph.
`...`	Other arguments to be passed

Details

Each of the 27 prediction functions is coded as x/y/z with x, y and z taking values in $-1, 0, 1$. As an example, the combination 0/0/0 represents the prediction function that projects every value of Δ_S to 0. Similarly, the combination -1/0/1 is the identity function projecting every value of Δ_S to the same value for Δ_T.

Value

An object of class `comb27.BinBin` with components,

`index`	count variable
`Monotonicity`	The vector of Monotonicity assumptions
`Pe`	The vector of the prediction error values.
`combo`	The vector containing the codes for the each of the 27 prediction functions.
`R2_H`	The vector of the R^2_H values.
`H_Delta_T`	The vector of the entropies of Δ_T.
`H_Delta_S`	The vector of the entropies of Δ_S.
`I_Delta_T_Delta_S`	The vector of the mutual information of Δ_S and Δ_T.

Author(s)

Paul Meyvisch, Wim Van der Elst, Ariel Alonso

References

Alonso A, Van der Elst W, Molenberghs G, Buyse M and Burzykowski T. (2016). An information-theoretic approach for the evaluation of surrogate endpoints based on causal inference.

Alonso A, Van der Elst W and Meyvisch P (2016). Assessing a surrogate predictive value: A causal inference approach.

See Also

`comb27.BinBin`

Examples

```
## Not run:  # time consuming code part
CIGTS_27 <- comb27.BinBin(pi1_1_ = 0.3412, pi1_0_ = 0.2539, pi0_1_ = 0.119,
                        pi_1_1 = 0.6863, pi_1_0 = 0.0882, pi_0_1 = 0.0784,
                        Seed=1,Monotonicity=c("No"), M=500000)
plot.comb27.BinBin(CIGTS_27,lab="CIGTS")

## End(Not run)
```

---

plot.Fano.BinBin — *Plots the distribution of $R\hat{}2\_HL$ either as a density or as function of π_10 in the setting where both S and T are binary endpoints*

Description

The function `plot.Fano.BinBin` plots the distribution of R^2_{HL} which is fully identifiable for given values of π_{10}. See **Details** below.

Usage

```
## S3 method for class 'Fano.BinBin'
plot(x,Type="Density",Xlab.R2_HL,main.R2_HL,
ylab="density",Par=par(mfrow=c(1,1),oma=c(0,0,0,0),mar=c(5.1,4.1,4.1,2.1)),
Cex.Legend=1,Cex.Position="top", lwd=3,linety=c(5,6,7),color=c(8,9,3),...)
```

Arguments

`x` An object of class `Fano.BinBin`. See `Fano.BinBin`.

`Type` The type of plot that is produced. When `Type="Freq"`, a histogram of R^2_{HL} is produced. When `Type="Density"`, the density of R^2_{HL} is produced. When `Type="Scatter"`, a scatter plot of R^2_{HL} is produced as a function of π_{10}. Default `Type="Scatter"`.

`Xlab.R2_HL` The label of the X-axis when density plots or histograms are produced.

`main.R2_HL` Title of the density plot or histogram.

`ylab` The label of the Y-axis when density plots or histograms are produced. Default `ylab="density"`.

`Par` Graphical parameters for the plot. Default `par(mfrow=c(1,1),oma=c(0,0,0,0),mar=c(5.1,4.1,4.1,`

`Cex.Legend` The size of the legend. Default `Cex.Legend=1`.

`Cex.Position` The position of the legend. Default `Cex.Position="top"`.

`lwd` The line width for the density plot . Default `lwd=3`.

`linety` The line types corresponding to each level of `fano_delta` . Default `linety=c(5,6,7)`.

`color` The color corresponding to each level of `fano_delta` . Default `color=c(8,9,3)`.

`...` Other arguments to be passed.

Details

Values for π_{10} have to be uniformly sampled from the interval $[0, \min(\pi_{1\cdot}, \pi_{\cdot 0})]$. Any sampled value for π_{10} will fully determine the bivariate distribution of potential outcomes for the true endpoint.

The vector $\boldsymbol{\pi_{km}}$ fully determines R^2_{HL}.

Value

An object of class `Fano.BinBin` with components,

`R2_HL`	The sampled values for R^2_{HL}.
`H_Delta_T`	The sampled values for $H\Delta T$.
`minpi10`	The minimum value for π_{10}.
`maxpi10`	The maximum value for π_{10}.
`samplepi10`	The sampled value for π_{10}.
`delta`	The specified vector of upper bounds for the prediction errors.
`uncertainty`	Indexes the sampling of $pi1\_$.
`pi_00`	The sampled values for π_{00}.
`pi_11`	The sampled values for π_{11}.
`pi_01`	The sampled values for π_{01}.
`pi_10`	The sampled values for π_{10}.

Author(s)

Paul Meyvisch, Wim Van der Elst, Ariel Alonso

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2014). Validation of surrogate endpoints: the binary-binary setting from a causal inference perspective.

See Also

`Fano.BinBin`

Examples

```
# Conduct the analysis assuming no montonicity
# for the true endpoint, using a range of
# upper bounds for prediction errors
FANO<-Fano.BinBin(pi1_ = 0.5951 ,  pi_1 = 0.7745,
fano_delta=c(0.05, 0.1, 0.2), M=1000)

plot(FANO, Type="Scatter",color=c(3,4,5),Cex.Position="bottom")
```

plot.ICA.BinCont *Plot the individual causal association (ICA) in the causal-inference single-trial setting in the binary-continuous case.*

Description

This function is used to a plot that displays the frequencies, percentages, cumulative percentages or densities of the individual causal association (ICA) in the single-trial setting within the causal-inference framework when the surrogate endpoint is continuous (normally distributed) and the true endpoint is a binary outcome. In addition, several plots to evaluate the goodness-of-fit of the mixture model used to fit the conditional distribution of potential outcomes on the surrogate endpoint can also be provided. For details, see Alonso Abad *et al.* (2023).

Usage

```
## S3 method for class 'ICA.BinCont'
plot(x, Histogram.ICA=TRUE, Mixmean=TRUE, Mixvar=TRUE, Deviance=TRUE,
                              Type="Percent", Labels=FALSE, ...)
```

Arguments

x	A fitted object of class ICA.BinCont. See ICA.BinCont or ICA.BinCont.BS.
Histogram.ICA	Logical. Should a histogram of ICA be provided? Default Histogram.ICA=TRUE.
Mixmean	Logical. Should a plot of the calculated means of the fitted mixtures for S_0 and S_1 across different iterations be provided? Default Mixmean=TRUE.
Mixvar	Logical. Should a plot of the calculated variances of the fitted mixtures for S_0 and S_1 across different iterations be provided? Default Mixvar=TRUE.
Deviance	Logical. Should a boxplot of the deviances for the fitted mixtures of S_0 and S_1 be provided? Default Deviance=TRUE.
Type	The type of plot that is produced for the histogram of ICA. When Type="Freq" or Type="Percent", the Y-axis shows frequencies or percentages of R^2_H. When Type="CumPerc", the Y-axis shows cumulative percentages. When Type="Density", the density is shown.
Labels	Logical. When Labels=TRUE, the percentage of R^2_H values that are equal to or larger than the midpoint value of each of the bins are added in the histogram of ICA (on top of each bin). Default Labels=FALSE.
...	Extra graphical parameters to be passed to plot() or hist().

Author(s)

Wim Van der Elst, Fenny Ong, Ariel Alonso, and Geert Molenberghs

References

Alonso Abad, A., Ong, F., Stijven, F., Van der Elst, W., Molenberghs, G., Van Keilegom, I., Verbeke, G., & Callegaro, A. (2023). An information-theoretic approach for the assessment of a continuous outcome as a surrogate for a binary true endpoint based on causal inference: Application to vaccine evaluation.

See Also

`ICA.BinCont`, `ICA.BinCont.BS`

Examples

```
## Not run: # Time consuming code part
data(Schizo)
Fit <- ICA.BinCont.BS(Dataset = Schizo, Surr = BPRS, True = PANSS_Bin, nb = 10,
Theta.S_0=c(-10,-5,5,10,10,10,10,10), Theta.S_1=c(-10,-5,5,10,10,10,10,10),
Treat=Treat, M=50, Seed=1)

summary(Fit)
plot(Fit)

## End(Not run)
```

---

plot.MetaAnalyticSurvBin

Generates a plot of the estimated treatment effects for the surrogate endpoint versus the estimated treatment effects for the true endpoint for an object fitted with the 'MetaAnalyticSurvBin()' function.

Description

Generates a plot of the estimated treatment effects for the surrogate endpoint versus the estimated treatment effects for the true endpoint for an object fitted with the 'MetaAnalyticSurvBin()' function.

Usage

```
## S3 method for class 'MetaAnalyticSurvBin'
plot(x, ...)
```

Arguments

`x`	An object of class 'MetaAnalyticSurvBin' fitted with the 'MetaAnalyticSurvBin()' function.
`...`	...

Value

A plot of the type ggplot

Examples

```
## Not run:
data("colorectal")
fit_bin <- MetaAnalyticSurvBin(data = colorectal, true = surv, trueind = SURVIND,
                               surrog = responder, trt = TREAT, center = CENTER,
                               trial = TRIAL, patientid = patientid,
                               adjustment="unadjusted")
plot(fit_bin)

## End(Not run)
```

---

plot.MetaAnalyticSurvCat

Generates a plot of the estimated treatment effects for the surrogate endpoint versus the estimated treatment effects for the true endpoint for an object fitted with the 'MetaAnalyticSurvCat()' function.

Description

Generates a plot of the estimated treatment effects for the surrogate endpoint versus the estimated treatment effects for the true endpoint for an object fitted with the 'MetaAnalyticSurvCat()' function.

Usage

```
## S3 method for class 'MetaAnalyticSurvCat'
plot(x, ...)
```

Arguments

x	An object of class 'MetaAnalyticSurvCat' fitted with the 'MetaAnalyticSurvCat()' function.
...	...

Value

A plot of the type ggplot

Examples

```
## Not run:
data("colorectal4")
fit <- MetaAnalyticSurvCat(data = colorectal4, true = truend, trueind = trueind, surrog = surrogend,
                          trt = treatn, center = center, trial = trialend, patientid = patid,
                               adjustment="unadjusted")
plot(fit)

## End(Not run)
```

---

plot.MetaAnalyticSurvCont

Generates a plot of the estimated treatment effects for the surrogate endpoint versus the estimated treatment effects for the true endpoint for an object fitted with the 'MetaAnalyticSurvCont()' function.

Description

Generates a plot of the estimated treatment effects for the surrogate endpoint versus the estimated treatment effects for the true endpoint for an object fitted with the 'MetaAnalyticSurvCont()' function.

Usage

```
## S3 method for class 'MetaAnalyticSurvCont'
plot(x, ...)
```

Arguments

x	An object of class 'MetaAnalyticSurvCont' fitted with the 'MetaAnalyticSurvCont()' function.
...	...

Value

A plot of the type ggplot

Examples

```
## Not run:
data("colorectal4")
data("prostate")
fit <- MetaAnalyticSurvCont(data = prostate, true = SURVTIME, trueind = SURVIND, surrog = PSA,
trt = TREAT, center = TRIAL, trial = TRIAL, patientid = PATID,
copula = "Hougaard", adjustment = "weighted")
plot(fit)
```

```
## End(Not run)
```

plot.MetaAnalyticSurvSurv

Generates a plot of the estimated treatment effects for the surrogate endpoint versus the estimated treatment effects for the true endpoint for an object fitted with the 'MetaAnalyticSurvSurv()' function.

Description

Generates a plot of the estimated treatment effects for the surrogate endpoint versus the estimated treatment effects for the true endpoint for an object fitted with the 'MetaAnalyticSurvSurv()' function.

Usage

```
## S3 method for class 'MetaAnalyticSurvSurv'
plot(x, ...)
```

Arguments

`x`	An object of class 'MetaAnalyticSurvSurv' fitted with the 'MetaAnalyticSurvSurv()' function.
`...`	...

Value

A plot of the type ggplot

Examples

```
## Not run:
data("colorectal4")
fit <- MetaAnalyticSurvSurv(data=Ovarian,true=Surv,trueind=SurvInd,surrog=Pfs,surrogind=PfsInd,
                            trt=Treat,center=Center,trial=Center,patientid=Patient,
                            copula="Plackett",adjustment="unadjusted")
plot(fit)

## End(Not run)
```

plot.PPE.BinBin *Plots the distribution of either PPE, RPE or $R\hat{}2\_H$ either as a density or as a histogram in the setting where both S and T are binary endpoints*

Description

The function `plot.PPE.BinBin` plots the distribution of PPE, RPE or R^2_H in the setting where both surrogate and true endpoints are binary in the single-trial causal-inference framework. See **Details** below.

Usage

```
## S3 method for class 'PPE.BinBin'
plot(x,Type="Density",Param="PPE",Xlab.PE,main.PE,
ylab="density",Cex.Legend=1,Cex.Position="bottomright", lwd=3,linety=1,color=1,
Breaks=0.05, xlimits=c(0,1), ...)
```

Arguments

`x` An object of class `PPE.BinBin`. See `PPE.BinBin`.

`Type` The type of plot that is produced. When `Type="Freq"`, a histogram is produced. When `Type="Density"`, a density is produced. Default `Type="Density"`.

`Param` Parameter to be plotted: is either "PPE", "RPE" or "ICA"

`Xlab.PE` The label of the X-axis when density plots or histograms are produced.

`main.PE` Title of the density plot or histogram.

`ylab` The label of the Y-axis for the density plots. Default `ylab="density"`.

`Cex.Legend` The size of the legend. Default `Cex.Legend=1`.

`Cex.Position` The position of the legend. Default `Cex.Position="bottomright"`.

`lwd` The line width for the density plot. Default `lwd=3`.

`linety` The line types for the density. Default `linety=1`.

`color` The color of the density or histogram. Default `color=1`.

`Breaks` The breaks for the histogram. Default `Breaks=0.05`.

`xlimits` The limits for the X-axis. Default `xlimits=c(0,1)`.

`...` Other arguments to be passed.

Details

In the continuous normal setting, surroagacy can be assessed by studying the association between the individual causal effects on S and T (see `ICA.ContCont`). In that setting, the Pearson correlation is the obvious measure of association.

When S and T are binary endpoints, multiple alternatives exist. Alonso et al. (2016) proposed the individual causal association (ICA; R_H^2), which captures the association between the individual causal effects of the treatment on S (Δ_S) and T (Δ_T) using information-theoretic principles.

The function `PPE.BinBin` computes R_H^2 using a grid-based approach where all possible combinations of the specified grids for the parameters that are allowed that are allowed to vary freely are considered. It additionally computes the minimal probability of a prediction error (PPE) and the reduction on the PPE using information that S conveys on T. Both measures provide complementary information over the R_H^2 and facilitate more straightforward clinical interpretation.

Value

An object of class `PPE.BinBin` with components,

`index` count variable

`PPE` The vector of the PPE values.

`RPE` The vector of the RPE values.

`PPE_T` The vector of the PPE_T values indicating the probability on a prediction error without using information on S.

`R2_H` The vector of the R_H^2 values.

`H_Delta_T` The vector of the entropies of Δ_T.

`H_Delta_S` The vector of the entropies of Δ_S.

`I_Delta_T_Delta_S`
The vector of the mutual information of Δ_S and Δ_T.

`Pi.Vectors` An object of class `data.frame` that contains the valid π vectors.

Author(s)

Paul Meyvisch, Wim Van der Elst, Ariel Alonso, Geert Molenberghs

References

Alonso A, Van der Elst W, Molenberghs G, Buyse M and Burzykowski T. (2016). An information-theoretic approach for the evaluation of surrogate endpoints based on causal inference.

Meyvisch P., Alonso A.,Van der Elst W, Molenberghs G. (2018). Assessing the predictive value of a binary surrogate for a binary true endpoint, based on the minimum probability of a prediction error.

See Also

PPE.BinBin

Examples

```
## Not run: # Time consuming part
PANSS <- PPE.BinBin(pi1_1_=0.4215, pi0_1_=0.0538, pi1_0_=0.0538,
                    pi_1_1=0.5088, pi_1_0=0.0307,pi_0_1=0.0482,
                    Seed=1, M=2500)
```

```
plot(PANSS,Type="Freq",Param="RPE",color="grey",Breaks=0.05,xlimits=c(0,1),main="PANSS")

## End(Not run)
```

plot.SPF.BinCont *Plot the surrogate predictive function (SPF) in the causal-inference single-trial setting in the binary-continuous case.*

Description

This function is used to create several plots related to the surrogate predictive function (SPF) in the single-trial setting within the causal-inference framework when the surrogate endpoint is continuous (normally distributed) and the true endpoint is a binary outcome. For details, see Alonso *et al.* (2024).

Usage

```
## S3 method for class 'SPF.BinCont'
plot(x, Histogram.SPF=TRUE, Causal.necessity=TRUE, Best.pred=TRUE, Max.psi=TRUE, ...)
```

Arguments

x
: A fitted object of class SPF.BinCont. See SPF.BinCont.

Histogram.SPF
: Logical. Should histograms of SPF be provided? When it is requested, a matrix of histograms illustrating various combination of the SPF, i.e., the $P[\Delta T|\Delta S \in I_{ab}]$, will be produced. Default Histogram.SPF=TRUE.

Causal.necessity
: Logical. Should a histogram showing the $P[\Delta T = 0|\Delta S = 0]$ be provided? Default Causal.necessity=TRUE.

Best.pred
: Logical. Should a bar plot showing the frequency of $\tilde{\psi}_{ab} = i$ for each interval (x, y) be provided? Default Best.pred=TRUE.

Max.psi
: Logical. Should a histogram showing the $P[\Delta T = \tilde{\psi}_{ab}(\Delta S)]$ be provided? Default Max.psi=TRUE.

...
: Extra graphical parameters to be passed to hist() or barplot().

Author(s)

Fenny Ong, Wim Van der Elst, Ariel Alonso, and Geert Molenberghs

References

Alonso, A., Ong, F., Van der Elst, W., Molenberghs, G., & Callegaro, A. (2024). Assessing a continuous surrogate predictive value for a binary true endpoint based on causal inference and information theory in vaccine trial.

See Also

SPF.BinCont

Examples

```
## Not run: # Time consuming code part
data(Schizo)
fit.ica <- ICA.BinCont.BS(Dataset = Schizo, Surr = BPRS, True = PANSS_Bin, nb = 10,
Theta.S_0=c(-10,-5,5,10,10,10,10,10), Theta.S_1=c(-10,-5,5,10,10,10,10,10),
Treat=Treat, M=50, Seed=1)

fit.spf <- SPF.BinCont(fit.ica, a=-5, b=5)

summary(fit.spf)
plot(fit.spf)

## End(Not run)
```

---

plot.SurvSurv

Provides plots of trial- and individual-level surrogacy in the Information-Theoretic framework when both S and T are time-to-event endpoints

---

Description

Produces plots that provide a graphical representation of trial- and/or individual-level surrogacy (R2_ht and R2_hInd per cluster) based on the Information-Theoretic approach of Alonso & Molenberghs (2007).

Usage

```
## S3 method for class 'SurvSurv'
plot(x, Trial.Level=TRUE, Weighted=TRUE,
Indiv.Level.By.Trial=TRUE, Xlab.Indiv, Ylab.Indiv, Xlab.Trial,
Ylab.Trial, Main.Trial, Main.Indiv,
Par=par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1)), ...)
```

Arguments

`x` An object of class `FixedBinBinIT`.

`Trial.Level` Logical. If `Trial.Level=TRUE`, a plot of the trial-specific treatment effects on the true endpoint against the trial-specific treatment effect on the surrogate endpoints is provided (as a graphical representation of R_{ht}). Default `TRUE`.

`Weighted` Logical. This argument only has effect when the user requests a trial-level surrogacy plot (i.e., when `Trial.Level=TRUE` in the function call). If `Weighted=TRUE`, the circles that depict the trial-specific treatment effects on the true endpoint against the surrogate endpoint are proportional to the number of patients in the trial. If `Weighted=FALSE`, all circles have the same size. Default `TRUE`.

`Indiv.Level.By.Trial`
: Logical. If `Indiv.Level.By.Trial=TRUE`, a plot that shows the estimated $R^2_{h.ind}$ for each trial (and confidence intervals) is provided. Default `TRUE`.

`Xlab.Indiv`
: The legend of the X-axis of the plot that depicts the estimated $R^2_{h.ind}$ per trial. Default "$R[h.ind]^2$.

`Ylab.Indiv`
: The legend of the Y-axis of the plot that shows the estimated $R^2_{h.ind}$ per trial. Default "Trial".

`Xlab.Trial`
: The legend of the X-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the surrogate endpoint (α_i)".

`Ylab.Trial`
: The legend of the Y-axis of the plot that depicts trial-level surrogacy. Default "Treatment effect on the true endpoint (β_i)".

`Main.Indiv`
: The title of the plot that depicts individual-level surrogacy. Default "Individual-level surrogacy".

`Main.Trial`
: The title of the plot that depicts trial-level surrogacy. Default "Trial-level surrogacy".

`Par`
: Graphical parameters for the plot. Default `par(oma=c(0, 0, 0, 0), mar=c(5.1, 4.1, 4.1, 2.1))`.

`...`
: Extra graphical parameters to be passed to `plot()`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A, & Molenberghs, G. (2007). Surrogate marker evaluation from an information theory perspective. *Biometrics, 63,* 180-186.

See Also

SurvSurv

Examples

```
# Open Ovarian dataset
data(Ovarian)

# Conduct analysis
Fit <- SurvSurv(Dataset = Ovarian, Surr = Pfs, SurrCens = PfsInd,
True = Surv, TrueCens = SurvInd, Treat = Treat,
Trial.ID = Center, Alpha=.05)

# Examine results
summary(Fit)
plot(Fit, Trial.Level = FALSE, Indiv.Level.By.Trial=TRUE)
plot(Fit, Trial.Level = TRUE, Indiv.Level.By.Trial=FALSE)
```

plot.vine_copula_fit *Goodness-of-fit plots for the fitted copula models*

Description

plot.vine_copula_fit() plots simple goodness-of-fit plots for the vine copula model fitted with fit_copula_ContCont(), fit_copula_OrdCont(), and fit_copula_OrdOrd().

Usage

```
## S3 method for class 'vine_copula_fit'
plot(x, ...)
```

Arguments

x	S3 object returned by fit_copula_ContCont(), fit_copula_OrdCont(), or fit_copula_OrdOrd().
...	Additional parameters. Currently not implemented.

Marginal Goodness-of-Fit

Continuous Endpoints:

The estimated model-based marginal density for each continuous endpoint is plotted alongside a histogram based on the observed data.

Ordinal Endpoints:

The estimated model-based marginal probabilities for each ordinal endpoint is plotted alongside the empirical proportions (red). Red whiskers represent the 95% confidence intervals for the empirical proportions. These are based on the delta method with the logit transformation for the proportion.

Goodness-of-Fit of Association Structure

Ordinal-Ordinal:

For each possible value for the surrogate, a plot is produced with (i) the model-based estimated conditional probabilities, $P(T = t|S)$, and (ii) the corresponding empirical conditional probabilities (red). Red whiskers represent the 95% confidence intervals for these empirical proportions. These are based on the delta method with the logit transformation for the proportion.

Ordinal-Continuous:

The model-based estimated regression function $E(T|S = s)$ is plotted alongside a semiparametric estimate using `mgcv::gam(y~s(x), family = stats::quasi())` (red). Dashed lines represent pointwise 95% confidence intervals based on the semiparametric estimate. These confidence intervals are not trustworthy as they are based on a constant variance assumption.

Continuous-Continuous:
The model-based estimated regression function $E(T|S = s)$ is plotted alongside a semiparametric estimate using `mgcv::gam(y~s(x), family = stats::quasi())` (red). Dashed lines represent pointwise 95% confidence intervals based on the semiparametric estimate.

`Pos.Def.Matrices` *Generate 4 by 4 correlation matrices and flag the positive definite ones*

Description

Based on vectors (or scalars) for the six off-diagonal correlations of a 4 by 4 matrix, the function `Pos.Def.Matrices` constructs all possible matrices that can be formed by combining the specified values, computes the minimum eigenvalues for each of these matrices, and flags the positive definite ones (i.e., valid correlation matrices).

Usage

```
Pos.Def.Matrices(T0T1=seq(0, 1, by=.2), T0S0=seq(0, 1, by=.2), T0S1=seq(0, 1,
by=.2), T1S0=seq(0, 1, by=.2), T1S1=seq(0, 1, by=.2), S0S1=seq(0, 1, by=.2))
```

Arguments

`T0T1`	A vector or scalar that specifies the correlation(s) between T0 and T1 that should be considered to construct all possible 4 by 4 matrices. Default `seq(0, 1, by=.2)`, i.e., the values 0, 0.20, $\ldots$, 1.
`T0S0`	A vector or scalar that specifies the correlation(s) between T0 and S0 that should be considered to construct all possible 4 by 4 matrices. Default `seq(0, 1, by=.2)`.
`T0S1`	A vector or scalar that specifies the correlation(s) between T0 and S1 that should be considered to construct all possible 4 by 4 matrices. Default `seq(0, 1, by=.2)`.
`T1S0`	A vector or scalar that specifies the correlation(s) between T1 and S0 that should be considered to construct all possible 4 by 4 matrices. Default `seq(0, 1, by=.2)`.
`T1S1`	A vector or scalar that specifies the correlation(s) between T1 and S1 that should be considered to construct all possible 4 by 4 matrices. Default `seq(0, 1, by=.2)`.
`S0S1`	A vector or scalar that specifies the correlation(s) between S0 and S1 that should be considered to construct all possible 4 by 4 matrices. Default `seq(0, 1, by=.2)`.

Details

The generated object `Generated.Matrices` (of class `data.frame`) is placed in the workspace (for easy access).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

See Also

Sim.Data.Counterfactuals

Examples

```
## Generate all 4x4 matrices that can be formed using rho(T0,S0)=rho(T1,S1)=.5
## and the grid of values 0, .2, ..., 1 for the other off-diagonal correlations:
Pos.Def.Matrices(T0T1=seq(0, 1, by=.2), T0S0=.5, T0S1=seq(0, 1, by=.2),
T1S0=seq(0, 1, by=.2), T1S1=.5, S0S1=seq(0, 1, by=.2))

## Examine the first 10 rows of the the object Generated.Matrices:
Generated.Matrices[1:10,]

## Check how many of the generated matrices are positive definite
## (counts and percentages):
table(Generated.Matrices$Pos.Def.Status)
table(Generated.Matrices$Pos.Def.Status)/nrow(Generated.Matrices)

## Make an object PosDef which contains the positive definite matrices:
PosDef <- Generated.Matrices[Generated.Matrices$Pos.Def.Status==1,]

## Shows the 10 first matrices that are positive definite:
PosDef[1:10,]
```

PPE.BinBin *Evaluate a surrogate predictive value based on the minimum probability of a prediction error in the setting where both S and T are binary endpoints*

Description

The function PPE.BinBin assesses a surrogate predictive value using the probability of a prediction error in the single-trial causal-inference framework when both the surrogate and the true endpoints are binary outcomes. It additionally assesses the indivdiual causal association (ICA). See **Details** below.

Usage

```
PPE.BinBin(pi1_1_, pi1_0_, pi_1_1, pi_1_0,
pi0_1_, pi_0_1, M=10000, Seed=1)
```

Arguments

`pi1_1_` A scalar that contains values for $P(T = 1, S = 1|Z = 0)$, i.e., the probability that $S = T = 1$ when under treatment $Z = 0$.

`pi1_0_` A scalar that contains values for $P(T = 1, S = 0|Z = 0)$.

`pi_1_1` A scalar that contains values for $P(T = 1, S = 1|Z = 1)$.

`pi_1_0` A scalar that contains values for $P(T = 1, S = 0|Z = 1)$.

`pi0_1_` A scalar that contains values for $P(T = 0, S = 1|Z = 0)$.

`pi_0_1` A scalar that contains values for $P(T = 0, S = 1|Z = 1)$.

`M` The number of valid vectors that have to be obtained. Default `M=10000`.

`Seed` The seed to be used to generate π_r. Default `Seed=1`.

Details

In the continuous normal setting, surroagacy can be assessed by studying the association between the individual causal effects on S and T (see `ICA.ContCont`). In that setting, the Pearson correlation is the obvious measure of association.

When S and T are binary endpoints, multiple alternatives exist. Alonso et al. (2016) proposed the individual causal association (ICA; R_H^2), which captures the association between the individual causal effects of the treatment on S (Δ_S) and T (Δ_T) using information-theoretic principles.

The function `PPE.BinBin` computes R_H^2 using a grid-based approach where all possible combinations of the specified grids for the parameters that are allowed to vary freely are considered. It additionally computes the minimal probability of a prediction error (PPE) and the reduction on the PPE using information that S conveys on T. Both measures provide complementary information over the R_H^2 and facilitate more straightforward clinical interpretation. No assumption about monotonicity can be made.

Value

An object of class `PPE.BinBin` with components,

`index` count variable

`PPE` The vector of the PPE values.

`RPE` The vector of the RPE values.

`PPE_T` The vector of the PPE_T values indicating the probability on a prediction error without using information on S.

`R2_H` The vector of the R_H^2 values.

`H_Delta_T` The vector of the entropies of Δ_T.

`H_Delta_S` The vector of the entropies of Δ_S.

`I_Delta_T_Delta_S`
The vector of the mutual information of Δ_S and Δ_T.

Author(s)

Paul Meyvisch, Wim Van der Elst, Ariel Alonso, Geert Molenberghs

References

Alonso A, Van der Elst W, Molenberghs G, Buyse M and Burzykowski T. (2016). An information-theoretic approach for the evaluation of surrogate endpoints based on causal inference.

Meyvisch P., Alonso A.,Van der Elst W, Molenberghs G. (2018). Assessing the predictive value of a binary surrogate for a binary true endpoint, based on the minimum probability of a prediction error.

See Also

`ICA.BinBin.Grid.Sample`

Examples

```
# Conduct the analysis

## Not run:  # time consuming code part
PPE.BinBin(pi1_1_=0.4215, pi0_1_=0.0538, pi1_0_=0.0538,
           pi_1_1=0.5088, pi_1_0=0.0307,pi_0_1=0.0482,
           Seed=1, M=10000)

## End(Not run)
```

`Pred.TrialT.ContCont` *Compute the expected treatment effect on the true endpoint in a new trial (when both S and T are normally distributed continuous endpoints)*

Description

The key motivation to evaluate a surrogate endpoint is to be able to predict the treatment effect on the true endpoint T based on the treatment effect on S in a new trial $i = 0$. The function `Pred.TrialT.ContCont` allows for making such predictions based on fitted models of class `BimixedContCont`, `BifixedContCont`, `UnimixedContCont` and `UnifixedContCont`.

Usage

```
Pred.TrialT.ContCont(Object, mu_S0, alpha_0, alpha.CI=0.05)
```

Arguments

`Object` A fitted object of class `BimixedContCont`, `BifixedContCont`, `UnimixedContCont` and `UnifixedContCont`. Some of the components in these fitted objects are needed to estimate $E(\beta + b_0)$ and its variance.

`mu_S0` The intercept of a regression model in the new trial $i = 0$ where the surrogate endpoint is regressed on the true endpoint, i.e., $S_{0j} = \mu_{S0} + \alpha_0 Z_{0j} + \varepsilon_{S0j}$, where S is the surrogate endpoint, j is the patient indicator, and Z is the treatment. This argument only needs to be specified when a full model was used to examine surroacy.

`alpha_0` The regression weight of the treatment in the regression model specified under argument `mu_S0`.

`alpha.CI` The α-level to be used to determine the confidence interval around $E(\beta + b_0)$. Default `alpha.CI=0.05`.

Details

The key motivation to evaluate a surrogate endpoint is to be able to predict the treatment effect on the true endpoint T based on the treatment effect on S in a new trial $i = 0$.

When a so-called full (fixed or mixed) bi- or univariate model was fitted in the surrogate evaluation phase (for details, see `BimixedContCont`, `BifixedContCont`, `UnimixedContCont` and `UnifixedContCont`), this prediction is made as:

$$E(\beta + b_0|m_{S0}, a_0) = \beta + \begin{pmatrix} d_{Sb} \\ d_{ab} \end{pmatrix}^T \begin{pmatrix} d_{SS} & D_{Sa} \\ d_{Sa} & d_{aa} \end{pmatrix}^{-1} \begin{pmatrix} \mu_{S0} - \mu_S \\ \alpha_0 - \alpha \end{pmatrix}$$

$$Var(\beta + b_0|m_{S0}, a_0) = d_{bb} + \begin{pmatrix} d_{Sb} \\ d_{ab} \end{pmatrix}^T \begin{pmatrix} d_{SS} & D_{Sa} \\ d_{Sa} & d_{aa} \end{pmatrix}^{-1} \begin{pmatrix} d_{Sb} \\ d_{ab} \end{pmatrix},$$

where all components are defined as in `BimixedContCont`. When the univariate mixed-effects models are used or the (univariate or bivariate) fixed effects models, the fitted components contained in `D.Equiv` are used instead of those in `D`.

When a reduced-model approach was used in the surrogate evaluation phase, the prediction is made as:

$$E(\beta + b_0|a_0) = \beta + \frac{d_{ab}}{d_{aa}} + (\alpha_0 - \alpha),$$

$$Var(\beta + b_0|a_0) = d_{bb} - \frac{d_{ab}^2}{d_{aa}},$$

where all components are defined as in `BimixedContCont`. When the univariate mixed-effects models are used or the (univariate or bivariate) fixed effects models, the fitted components contained in `D.Equiv` are used instead of those in `D`.

A $(1-\gamma)100\%$ prediction interval for $E(\beta + b_0|m_{S0}, a_0)$ can be obtained as $E(\beta + b_0|m_{S0}, a_0) \pm z_{1-\gamma/2}\sqrt{Var(\beta + b_0|m_{S0}, a_0)}$ (and similarly for $E(\beta + b_0|a_0)$).

Value

`Beta_0` The predicted β_0.

`Variance` The variance of the prediction.

`Lower`	The lower bound of the confidence interval around the expected β_0, see Details above.
`Upper`	The upper bound of the confidence interval around the expected β_0.
`alpha.CI`	The α-level used to establish the confidence interval.
`Surr.Model`	The model that was used to compute β_0.
`alpha_0`	The slope of the regression model specified in the `Arguments` section.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

See Also

`UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`

Examples

```
## Not run:  #time-consuming code parts
# Generate dataset
Sim.Data.MTS(N.Total=2000, N.Trial=15, R.Trial.Target=.8,
R.Indiv.Target=.8, D.aa=10, D.bb=50, Fixed.Effects=c(1, 2, 30, 90),
Seed=1)

# Evaluate surrogacy using a reduced bivariate mixed-effects model
BimixedFit <- BimixedContCont(Dataset = Data.Observed.MTS, Surr = Surr,
True = True, Treat = Treat, Trial.ID = Trial.ID, Pat.ID = Pat.ID,
Model="Reduced")

# Suppose that in a new trial, it was estimated alpha_0 = 30
# predict beta_0 in this trial
Pred_Beta <- Pred.TrialT.ContCont(Object = BimixedFit,
alpha_0 = 30)

# Examine the results
summary(Pred_Beta)

# Plot the results
plot(Pred_Beta)

## End(Not run)
```

Prentice *Evaluates surrogacy based on the Prentice criteria for continuous endpoints (single-trial setting)*

Description

The function Prentice evaluates the validity of a potential surrogate based on the Prentice criteria (Prentice, 1989) in the setting where the candidate surrogate and the true endpoint are normally distributed endpoints.

Warning The Prentice approach is included in the *Surrogate* package for illustrative purposes (as it was the first formal approach to assess surrogacy), but this method has some severe problems that renders its use problematic (see **Details** below). It is recommended to replace the Prentice approach by a more statistically-sound approach to evaluate a surrogate (e.g., the meta-analytic methods; see the functions UnifixedContCont, BifixedContCont, UnimixedContCont, BimixedContCont).

Usage

```
Prentice(Dataset, Surr, True, Treat, Pat.ID, Alpha=.05)
```

Arguments

Dataset	A data.frame that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.
Surr	The name of the variable in Dataset that contains the surrogate values.
True	The name of the variable in Dataset that contains the true endpoint values.
Treat	The name of the variable in Dataset that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.
Pat.ID	The name of the variable in Dataset that contains the patient's ID.
Alpha	The α-level that is used to examine whether the Prentice criteria are fulfilled. Default 0.05.

Details

The Prentice criteria are examined by fitting the following regression models (when the surrogate and true endpoints are continuous variables):

$$S_j = \mu_S + \alpha Z_j + \varepsilon_{Sj}, (1)$$

$$T_j = \mu_T + \beta Z_j + \varepsilon_{Tj}, (2)$$

$$T_j = \mu + \gamma Z_j + \varepsilon_j, (3)$$

$$T_j = \tilde{\mu}_T + \beta_S Z_j + \gamma_Z S_j + \tilde{\varepsilon}_{Tj}, (4)$$

where the error terms of (1) and (2) have a joint zero-mean normal distribution with variance-covariance matrix

$$\boldsymbol{\Sigma} = \left(\begin{array}{cc} \sigma_{SS} & \\ \sigma_{ST} & \sigma_{TT} \end{array} \right)$$

,

and where j is the subject indicator, S_j and T_j are the surrogate and true endpoint values of subject j, and Z_j is the treatment indicator for subject j.

To be in line with the Prentice criteria, Z should have a significant effect on S in model 1 (Prentice criterion 1), Z should have a significant effect on T in model 2 (Prentice criterion 2), S should have a significant effect on T in model 3 (Prentice criterion criterion 3), and the effect of Z on T should be fully captured by S in model 4 (Prentice criterion 4).

The Prentice approach to assess surrogavy has some fundamental limitations. For example, the fourth Prentice criterion requires that the statistical test for the β_S in model 4 is non-significant. This criterion is useful to reject a poor surrogate, but it is not suitable to validate a good surrogate (i.e., a non-significant result may always be attributable to a lack of statistical power). Even when lack of power would not be an issue, the result of the statistical test to evaluate the fourth Prentice criterion cannot prove that the effect of the treatment on the true endpoint is fully captured by the surrogate.

The use of the Prentice approach to evaluate a surrogate is not recommended. Instead, consider using the single-trial meta-anlytic method (if no multiple clinical trials are available or if there is no other clustering unit in the data; see function `Single.Trial.RE.AA`) or the multiple-trial meta-analytic methods (see `UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`, and `BimixedContCont`).

Value

`Prentice.Model.1`
An object of class `lm` that contains the fitted model 1 (using the Prentice approach).

`Prentice.Model.2`
An object of class `lm` that contains the fitted model 2 (using the Prentice approach).

`Prentice.Model.3`
An object of class `lm` that contains the fitted model 3 (using the Prentice approach).

`Prentice.Model.4`
An object of class `lm` that contains the fitted model 4 (using the Prentice approach).

`Prentice.Passed`
`Logical`. If all four Prentice criteria are fulfilled, `Prentice.Passed=TRUE`. If at least one criterion is not fulfilled, `Prentice.Passed=FALSE`.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Prentice, R. L. (1989). Surrogate endpoints in clinical trials: definitions and operational criteria. *Statistics in Medicine, 8,* 431-440.

Examples

```
## Load the ARMD dataset
data(ARMD)

## Evaluate the Prentice criteria in the ARMD dataset
Prent <- Prentice(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Pat.ID=Id)

# Summary of results
summary(Prent)
```

---

print.MetaAnalyticSurvBin

Prints all the elements of an object fitted with the 'MetaAnalyticSurvBin()' function.

Description

Prints all the elements of an object fitted with the 'MetaAnalyticSurvBin()' function.

Usage

```
## S3 method for class 'MetaAnalyticSurvBin'
print(x, ...)
```

Arguments

x	An object of class 'MetaAnalyticSurvBin' fitted with the 'MetaAnalyticSurvBin()' function.
...	...

Value

The surrogacy measures with their 95% confidence intervals and the estimated treament effect on the surrogate and true endpoint.

Examples

```
## Not run:
data("colorectal")
fit_bin <- MetaAnalyticSurvBin(data = colorectal, true = surv, trueind = SURVIND,
                               surrog = responder, trt = TREAT, center = CENTER,
                               trial = TRIAL, patientid = patientid,
                               adjustment="unadjusted")
print(fit_bin)

## End(Not run)
```

`print.MetaAnalyticSurvCat`
Prints all the elements of an object fitted with the 'MetaAnalyticSurvCat()' function.

Description

Prints all the elements of an object fitted with the 'MetaAnalyticSurvCat()' function.

Usage

```
## S3 method for class 'MetaAnalyticSurvCat'
print(x, ...)
```

Arguments

x	An object of class 'MetaAnalyticSurvCat' fitted with the 'MetaAnalyticSurvCat()' function.
...	...

Value

The surrogacy measures with their 95% confidence intervals and the estimated treatment effect on the surrogate and true endpoint.

Examples

```
## Not run:
data("colorectal4")
fit <- MetaAnalyticSurvCat(data = colorectal4, true = truend, trueind = trueind, surrog = surrogend,
                       trt = treatn, center = center, trial = trialend, patientid = patid,
                           adjustment="unadjusted")
print(fit)

## End(Not run)
```

print.MetaAnalyticSurvCont

Prints all the elements of an object fitted with the 'MetaAnalyticSurvCont()' function.

Description

Prints all the elements of an object fitted with the 'MetaAnalyticSurvCont()' function.

Usage

```
## S3 method for class 'MetaAnalyticSurvCont'
print(x, ...)
```

Arguments

x	An object of class 'MetaAnalyticSurvCont' fitted with the 'MetaAnalyticSurvCont()' function.
...	...

Value

The surrogacy measures with their 95% confidence intervals and the estimated treatment effect on the surrogate and true endpoint.

Examples

```
## Not run:
data("colorectal4")
data("prostate")
fit <- MetaAnalyticSurvCont(data = prostate, true = SURVTIME, trueind = SURVIND, surrog = PSA,
trt = TREAT, center = TRIAL, trial = TRIAL, patientid = PATID,
copula = "Hougaard", adjustment = "weighted")
print(fit)

## End(Not run)
```

print.MetaAnalyticSurvSurv

Prints all the elements of an object fitted with the 'MetaAnalyticSurvSurv()' function.

Description

Prints all the elements of an object fitted with the 'MetaAnalyticSurvSurv()' function.

Usage

```
## S3 method for class 'MetaAnalyticSurvSurv'
print(x, ...)
```

Arguments

`x` An object of class 'MetaAnalyticSurvSurv' fitted with the 'MetaAnalyticSurvSurv()' function.

`...` ...

Value

The surrogacy measures with their 95% confidence intervals and the estimated treatment effect on the surrogate and true endpoint.

Examples

```
## Not run:
data("colorectal4")
fit <- MetaAnalyticSurvSurv(data=Ovarian,true=Surv,trueind=SurvInd,surrog=Pfs,surrogind=PfsInd,
                            trt=Treat,center=Center,trial=Center,patientid=Patient,
                            copula="Plackett",adjustment="unadjusted")
print(fit)

## End(Not run)
```

`print.vine_copula_fit` *Print summary of fitted copula model*

Description

Print summary of fitted copula model

Usage

```
## S3 method for class 'vine_copula_fit'
print(x, ...)
```

Arguments

`x` Fitted-model object returned by `fit_copula_ContCont()`, `fit_copula_OrdCont()`, or `fit_copula_OrdOrd()`.

`...` not used

PROC.BinBin *Evaluate the individual causal association (ICA) and reduction in probability of a prediction error (RPE) in the setting where both S and T are binary endpoints*

Description

The function PROC.BinBin assesses the ICA and RPE in the single-trial causal-inference framework when both the surrogate and the true endpoints are binary outcomes. It additionally allows to account for sampling variability by means of bootstrap. See **Details** below.

Usage

```
PROC.BinBin(Dataset=Dataset, Surr=Surr, True=True, Treat=Treat,
BS=FALSE, seqs=250, MC_samples=1000, Seed=1)
```

Arguments

Dataset	A data.frame that should consist of one line per patient. Each line contains (at least) a binary surrogate value, a binary true endpoint value, and a treatment indicator.
Surr	The name of the variable in Dataset that contains the binary surrogate endpoint values. Should be coded as 0 and 1.
True	The name of the variable in Dataset that contains the binary true endpoint values. Should be coded as 0 and 1.
Treat	The name of the variable in Dataset that contains the treatment indicators. The treatment indicator should be coded as 1 for the experimental group and -1 for the control group.
BS	Logical. If TRUE, then Dataset will be bootstrapped to account for sampling variability. If FALSE, then no bootstrap is performed. See the **Details** section below. Default FALSE.
seqs	The number of copies of the dataset that are produced or alternatively the number of bootstrap datasets that are produced. Default seqs=250.
MC_samples	The number of Monte Carlo samples that need to be obtained per copy of the data set. Default MC_samples=1000.
Seed	The seed to be used. Default Seed=1.

Details

In the continuous normal setting, surroagacy can be assessed by studying the association between the individual causal effects on S and T (see ICA.ContCont). In that setting, the Pearson correlation is the obvious measure of association.

When S and T are binary endpoints, multiple alternatives exist. Alonso et al. (2016) proposed the individual causal association (ICA; R_H^2), which captures the association between the individual causal effects of the treatment on S (Δ_S) and T (Δ_T) using information-theoretic principles.

The function PPE.BinBin computes R_H^2 using a grid-based approach where all possible combinations of the specified grids for the parameters that are allowed to vary freely are considered. It additionally computes the minimal probability of a prediction error (PPE) and the reduction on the PPE using information that S conveys on T (RPE). Both measures provide complementary information over the R_H^2 and facilitate more straightforward clinical interpretation. No assumption about monotonicity can be made. The function PROC.BinBin makes direct use of the function PPE.BinBin. However, it is computationally much faster thanks to equally dividing the number of Monte Carlo samples over copies of the input data. In addition, it allows to account for sampling variability using a bootstrap procedure. Finally, the function PROC.BinBin computes the marginal probabilities directly from the input data set.

Value

An object of class PPE.BinBin with components,

PPE	The vector of the PPE values.
RPE	The vector of the RPE values.
PPE_T	The vector of the PPE_T values indicating the probability on a prediction error without using information on S.
R2_H	The vector of the R_H^2 values.

Author(s)

Paul Meyvisch, Wim Van der Elst, Ariel Alonso, Geert Molenberghs

References

Alonso A, Van der Elst W, Molenberghs G, Buyse M and Burzykowski T. (2016). An information-theoretic approach for the evaluation of surrogate endpoints based on causal inference.

Meyvisch P., Alonso A.,Van der Elst W, Molenberghs G.. Assessing the predictive value of a binary surrogate for a binary true endpoint, based on the minimum probability of a prediction error.

See Also

PPE.BinBin

Examples

```
# Conduct the analysis

## Not run:  # time consuming code part
library(Surrogate)
# load the CIGTS data
data(CIGTS)
CIGTS_25000<-PROC.BinBin(Dataset=CIGTS, Surr=IOP_12, True=IOP_96,
Treat=Treat, BS=FALSE,seqs=250, MC_samples=100, Seed=1)

## End(Not run)
```

`prostate` *The prostate dataset with a continuous surrogate.*

Description

This dataset combines the data that were collected in 17 double-blind randomized clinical trials in advanced prostate cancer.

Usage

```
data("prostate")
```

Format

A data frame with 412 observations on the following 6 variables.

- `TRIAL` The ID number of a trial.
- `TREAT` The treatment indicator, coded as 0=active control and 1=experimental treatment.
- `PSA` Prostate specific antigen (surrogate endpoint)
- `SURVTIME` Survival time (the true endpoint).
- `SURVIND` Censoring indicator for survival time.
- `PATID` The ID number of a patient.

References

Alonso A, Bigirumurame T, Burzykowski T, Buyse M, Molenberghs G, Muchene L, Perualila NJ, Shkedy Z, Van der Elst W, et al. (2016). Applied surrogate endpoint evaluation methods with SAS and R. CRC Press New York

Examples

```
data(prostate)
str(prostate)
head(prostate)
```

RandVec *Generate random vectors with a fixed sum*

Description

This function generates an n by m array x, each of whose m columns contains n random values lying in the interval [a,b], subject to the condition that their sum be equal to s. The distribution of values is uniform in the sense that it has the conditional probability distribution of a uniform distribution over the whole n-cube, given that the sum of the x's is s. The function uses the randfixedsum algorithm, written by Roger Stafford and implemented in MatLab. For details, see http://www.mathworks.com/matlabcentral/fileexchange/9700-random-vectors-with-fixed-sum/content/randfixedsum.m

Usage

```
RandVec(a=0, b=1, s=1, n=9, m=1, Seed=sample(1:1000, size = 1))
```

Arguments

a	The function RandVec generates an n by m matrix x. Each of the m columns contain n random values lying in the interval [a,b]. The argument a specifies the lower limit of the interval. Default 0.
b	The argument b specifies the upper limit of the interval. Default 1.
s	The argument s specifies the value to which each of the m generated columns should sum to. Default 1.
n	The number of requested elements per column. Default 9.
m	The number of requested columns. Default 1.
Seed	The seed that is used. Default sample(1:1000, size = 1).

Value

An object of class RandVec with components,

RandVecOutput	The randomly generated vectors.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

The function is an R adaptation of a matlab program written by Roger Stafford. For details on the original Matlab algorithm, see: http://www.mathworks.com/matlabcentral/fileexchange/9700-random-vectors-with-fixed-sum/content/randfixedsum.m

Examples

```
# generate two vectors with 10 values ranging between 0 and 1
# where each vector sums to 1
# (uniform distribution over the whole n-cube)
Vectors <- RandVec(a=0, b=1, s=1, n=10, m=2)
sum(Vectors$RandVecOutput[,1])
sum(Vectors$RandVecOutput[,2])
```

Restrictions.BinBin *Examine restrictions in π_f under different montonicity assumptions for binary S and T*

Description

The function Restrictions.BinBin gives an overview of the restrictions in π_f under different assumptions regarding montonicity when both S and T are binary.

Usage

```
Restrictions.BinBin(pi1_1_, pi1_0_, pi_1_1, pi_1_0, pi0_1_, pi_0_1)
```

Arguments

pi1_1_	A scalar that contains $P(T = 1, S = 1 \| Z = 0)$, i.e., the proability that $S = T = 1$ when under treatment $Z = 0$.
pi1_0_	A scalar that contains $P(T = 1, S = 0 \| Z = 0)$.
pi_1_1	A scalar that contains $P(T = 1, S = 1 \| Z = 1)$.
pi_1_0	A scalar that contains $P(T = 1, S = 0 \| Z = 1)$.
pi0_1_	A scalar that contains $P(T = 0, S = 1 \| Z = 0)$.
pi_0_1	A scalar that contains $P(T = 0, S = 1 \| Z = 1)$.

Value

An overview of the restrictions for the freely varying parameters imposed by the data is provided

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2014). Validation of surrogate endpoints: the binary-binary setting from a causal inference perspective.

See Also

MarginalProbs

Examples

```
Restrictions.BinBin(pi1_1_=0.262, pi0_1_=0.135, pi1_0_=0.286,
pi_1_1=0.637, pi_1_0=0.078, pi_0_1=0.127)
```

`sample_copula_parameters` *Sample Unidentifiable Copula Parameters*

Description

The `sample_copula_parameters()` function samples the unidentifiable copula parameters for the partly identifiable D-vine copula model, see for example `fit_copula_model_BinCont()` and `fit_model_SurvSurv()` for more information regarding the D-vine copula model.

Usage

```
sample_copula_parameters(
  copula_family2,
  n_sim,
  eq_cond_association = FALSE,
  lower = c(-1, -1, -1, -1),
  upper = c(1, 1, 1, 1)
)
```

Arguments

`copula_family2` Copula family of the other bivariate copulas. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family2` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`n_sim` Number of copula parameter vectors to be sampled.

`eq_cond_association` (boolean) Indicates whether $\rho_{13;2}$ and $\rho_{24;3}$ are set equal.

`lower` (numeric) Vector of length 4 that provides the lower limit, $\boldsymbol{a} = (a_{23}, a_{13;2}, a_{24;3}, a_{14;23})'$. Defaults to `c(-1, -1, -1, -1)`. If the provided lower limit is smaller than what is allowed for a particular copula family, then the copula family's lowest possible value is used instead.

`upper` (numeric) Vector of length 4 that provides the upper limit, $\boldsymbol{b} = (b_{23}, b_{13;2}, b_{24;3}, b_{14;23})'$. Defaults to `c(1, 1, 1, 1)`.

Value

A `n_sim` by 4 numeric matrix where each row corresponds to a sample for $\boldsymbol{\theta}_{unid}$.

Sampling

In the D-vine copula model in the Information-Theoretic Causal Inference (ITCI) framework, the following copulas are not identifiable: c_{23}, $c_{13;2}$, $c_{24;3}$, $c_{14;23}$. Let the corresponding copula parameters be

$$\boldsymbol{\theta}_{unid} = (\theta_{23}, \theta_{13;2}, \theta_{24;3}, \theta_{14;23})'.$$

The allowable range for this parameter vector depends on the corresponding copula families. For parsimony and comparability across different copula families, the sampling procedure consists of two steps:

1. Sample Spearman's rho parameters from a uniform distribution,

$$\boldsymbol{\rho}_{unid} = (\rho_{23}, \rho_{13;2}, \rho_{24;3}, \rho_{14;23})' \sim U(\boldsymbol{a}, \boldsymbol{b}).$$

2. Transform the sampled Spearman's rho parameters to the copula parameter scale, $\boldsymbol{\theta}_{unid}$.

These two steps are repeated `n_sim` times.

Conditional Independence

In addition to range restrictions through the `lower` and `upper` arguments, we allow for so-called conditional independence assumptions. These assumptions entail that $\rho_{13;2} = 0$ and $\rho_{24;3} = 0$. Or in other words, $U_1 \perp U_3 \,|\, U_2$ and $U_2 \perp U_4 \,|\, U_3$. In the context of a surrogate evaluation trial (where $(U_1, U_2, U_3, U_4)'$ corresponds to the probability integral transformation of $(T_0, S_0, S_1, T_1)'$) this assumption could be justified by subject-matter knowledge.

`sample_deltas_BinCont` *Sample individual casual treatment effects from given D-vine copula model in binary continuous setting*

Description

Sample individual casual treatment effects from given D-vine copula model in binary continuous setting

Usage

```
sample_deltas_BinCont(
  copula_par,
  rotation_par,
  copula_family1,
  copula_family2 = copula_family1,
  n,
  q_S0 = NULL,
  q_S1 = NULL,
  q_T0 = NULL,
  q_T1 = NULL,
  marginal_sp_rho = TRUE,
  setting = "BinCont",
  composite = FALSE,
  plot_deltas = FALSE,
  restr_time = +Inf
)
```

Arguments

`copula_par` Parameter vector for the sequence of bivariate copulas that define the D-vine copula. The elements of `copula_par` correspond to $(c_{12}, c_{23}, c_{34}, c_{13;2}, c_{24;3}, c_{14;23})$.

`rotation_par` Vector of rotation parameters for the sequence of bivariate copulas that define the D-vine copula. The elements of `rotation_par` correspond to $(c_{12}, c_{23}, c_{34}, c_{13;2}, c_{24;3}, c_{14;23})$.

`copula_family1` Copula family of c_{12} and c_{34}. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family` correspond to (c_{12}, c_{34}).

`copula_family2` Copula family of the other bivariate copulas. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family2` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`n` Number of samples to be taken from the D-vine copula.

`q_S0` Quantile function for the distribution of S_0.

`q_S1` Quantile function for the distribution of S_1.

`q_T0` Quantile function for the distribution of T_0. This should be `NULL` if T_0 is binary.

`q_T1` Quantile function for the distribution of T_1. This should be `NULL` if T_1 is binary.

`marginal_sp_rho`
(boolean) Compute the sample Spearman correlation matrix? Defaults to `TRUE`.

`setting` Should be one of the following two:

- `"BinCont"`: for when S is continuous and T is binary.
- `"SurvSurv"`: for when both S and T are time-to-event variables.

`composite` (boolean) If `composite` is `TRUE`, then the surrogate endpoint is a composite of both a "pure" surrogate endpoint and the true endpoint, e.g., progression-free survival is the minimum of time-to-progression and time-to-death.

`plot_deltas` Plot the sampled individual causal effects? Defaults to `FALSE`.

`restr_time` Restriction time for the potential outcomes. Defaults to `+Inf` which means no restriction. Otherwise, the sampled potential outcomes are replace by `pmin(S0, restr_time)` (and similarly for the other potential outcomes).

Value

A list with two elements:

- `Delta_dataframe`: a dataframe containing the sampled individual causal treatment effects
- `marginal_sp_rho_matrix`: a matrix containing the marginal pairwise Spearman's rho parameters estimated from the sample. If `marginal_sp_rho = FALSE`, this matrix is not computed and `NULL` is returned for this element of the list.

`sample_dvine` *Sample copula data from a given four-dimensional D-vine copula*

Description

`sample_dvine()` is a helper function that samples copula data from a given D-vine copula. See details for more information on the parameterization of the D-vine copula.

Usage

```
sample_dvine(
  copula_par,
  rotation_par,
  copula_family1,
  copula_family2 = copula_family1,
  n
)
```

Arguments

`copula_par` Parameter vector for the sequence of bivariate copulas that define the D-vine copula. The elements of `copula_par` correspond to $(c_{12}, c_{23}, c_{34}, c_{13;2}, c_{24;3}, c_{14;23})$.

`rotation_par` Vector of rotation parameters for the sequence of bivariate copulas that define the D-vine copula. The elements of `rotation_par` correspond to $(c_{12}, c_{23}, c_{34}, c_{13;2}, c_{24;3}, c_{14;23})$.

`copula_family1` Copula family of c_{12} and c_{34}. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family` correspond to (c_{12}, c_{34}).

`copula_family2` Copula family of the other bivariate copulas. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family2` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`n` Number of samples to be taken from the D-vine copula.

Value

A $n \times 4$ matrix where each row corresponds to one sampled vector and the columns correspond to U_1, U_2, U_3, and U_4.

D-vine Copula

Let $\boldsymbol{U} = (U_1, U_2, U_3, U_4)'$ be a random vector with uniform margins. The corresponding distribution function is then a 4-dimensional copula. A D-vine copula as a family of k-dimensional copulas. Indeed, a D-vine copula is a k-dimensional copula that is constructed from a particular product of bivariate copula densities. In this function, only 4-dimensional copula densities are considered. Under the simplifying assumption, the 4-dimensional D-vine copula density is the product of the following bivariate copula densities:

- c_{12}, c_{23}, and c_{34}

- $c_{13;2}$ and $c_{24;3}$
- $c_{14;23}$

Schizo *Data of five clinical trials in schizophrenia*

Description

These are the data of five clinical trials in schizophrenia. A total of 2128 patients were treated by 198 investiagators (psychiatrists). Patients' schizophrenic symptoms were measured using the PANSS, BPRS, and CGI. There were two treatment conditions (risperidone and control).

Usage

```
data(Schizo)
```

Format

A `data.frame` with 2128 observations on 9 variables.

`Id` The patient ID.

`InvestID` The ID of the investigator (psychiatrist) who treated the patient.

`Treat` The treatment indicator, coded as -1 = control and 1 = Risperidone.

`CGI` The change in the CGI score (= score at the start of the treatment - score at the end of the treatment).

`PANSS` The change in the PANSS score.

`BPRS` The change in the BPRS score.

`PANSS_Bin` The dichotomized PANSS change score, coded as 1 = a reduction of 20% or more in the PANSS score (score at the end of the treatment relative to score at the beginning of the treatment), 0 = otherwise.

`BPRS_Bin` The dichotomized BPRS change score, coded as 1 = a reduction of 20% or more in the BPRS score (score at the end of the treatment relative to score at the beginning of the treatment), 0 = otherwise.

`CGI_Bin` The sichtomized change in the CGI score, coded as 1 = a change of more than 3 points on the original scale (score at the end of the treatment relative to score at the beginning of the treatment), 0 = otherwise.

Schizo_Bin *Data of a clinical trial in Schizophrenia (with binary outcomes).*

Description

These are the data of a clinical trial in Schizophrenia (a subset of the dataset Schizo_Bin, study 1 where the patients were administered 10 mg. of haloperidol or 8 mg. of risperidone). A total of 454 patients were treated by 117 investigators (psychiatrists). Patients' schizophrenia symptoms at baseline and at the end of the study (after 8 weeks) were measured using the PANSS and BPRS. The variables BPRS_Bin and PANSS_Bin are binary outcomes that indicate whether clinically meaningful change had occurred (1 = a reduction of 20% or higher in the PANSS/BPRS scores at the last measurement compared to baseline; 0 = no such reduction; Leucht et al., 2005; Kay et al., 1988).

Usage

```
data(Schizo_Bin)
```

Format

A data.frame with 454 observations on 5 variables.

Id The patient ID.

InvestI The ID of the investigator (psychiatrist) who treated the patient.

Treat The treatment indicator, coded as -1 = control treatment (10 mg. haloperidol) and 1 = experimental treatment (8 mg. risperidone).

PANSS_Bin The dichotomized change in the PANSS score (1 = a reduction of 20% or more in the PANSS score, 0=otherwise)

BPRS_Bin The dichotomized change in the BPRS score (1 = a reduction of 20% or more in the BPRS score, 0=otherwise)

CGI_Bin The sichtomized change in the CGI score, coded as 1 = a change of more than 3 points on the original scale (score at the end of the treatment relative to score at the beginning of the treatment), 0 = otherwise.

References

Kay, S.R., Opler, L.A., & Lindenmayer, J.P. (1988). Reliability and validity of the Positive and Negative Syndrome Scale for schizophrenics. Psychiatric Research, 23, 99-110.

Leucht, S., et al. (2005). Clinical implications of Brief Psychiatric Rating Scale scores. The British Journal of Psychiarty, 187, 366-371.

Schizo_BinCont — *Data of a clinical trial in schizophrenia, with binary and continuous endpoints*

Description

These are the data of a clinical trial in schizophrenia. Patients' schizophrenic symptoms were measured using the PANSS, BPRS, and CGI. There were two treatment conditions (risperidone and control).

Usage

```
data(Schizo)
```

Format

A `data.frame` with 446 observations on 9 variables.

- `Id` The patient ID.
- `InvestID` The ID of the investigator (psychiatrist) who treated the patient.
- `Treat` The treatment indicator, coded as -1 = control and 1 = Risperidone.
- `CGI` The change in the CGI score (= score at the start of the treatment - score at the end of the treatment).
- `PANSS` The change in the PANSS score.
- `BPRS` The change in the PANSS score.
- `PANSS_Bin` The dichotomized PANSS change score, coded as 1 = a reduction of 20% or more in the PANSS score (score at the end of the treatment relative to score at the beginning of the treatment), 0 = otherwise.
- `BPRS_Bin` The dichotomized BPRS change score, coded as 1 = a reduction of 20% or more in the BPRS score (score at the end of the treatment relative to score at the beginning of the treatment), 0 = otherwise.
- `CGI_Bin` The sichtomized change in the CGI score, coded as 1 = a change of more than 3 points on the original scale (score at the end of the treatment relative to score at the beginning of the treatment), 0 = otherwise.

`Schizo_PANSS` *Longitudinal PANSS data of five clinical trials in schizophrenia*

Description

These are the longitudinal PANSS data of five clinical trial in schizophrenia. A total of 2151 patients were treated by 198 investiagators (psychiatrists). There were two treatment conditions (risperidone and control). Patients' schizophrenic symptoms were measured using the PANSS at different time moments following start of the treatment. The variables Week1-Week8 express the change scores over time using the raw (semi-continuous) PANSS scores. The variables Week1_bin - Week8_bin are binary indicators of a 20% or higher reduction in PANSS score versus baseline. The latter corresponds to a commonly accepted criterion for defining a clinically meaningful response (Kay et al., 1988).

Usage

```
data(Schizo_PANSS)
```

Format

A `data.frame` with 2151 observations on 6 variables.

`Id` The patient ID.

`InvestID` The ID of the investigator (psychiatrist) who treated the patient.

`Treat` The treatment indicator, coded as -1 = placebo and 1 = Risperidone.

`Week1` The change in the PANSS score 1 week after starting the treatment (= score at the end of the treatment - score at 1 week after starting the treatment).

`Week2` The change in the PANSS score 2 weeks after starting the treatment.

`Week4` The change in the PANSS score 4 weeks after starting the treatment.

`Week6` The change in the PANSS score 6 weeks after starting the treatment.

`Week8` The change in the PANSS score 8 weeks after starting the treatment.

`Week1_bin` The dichotomized change in the PANSS score 1 week after starting the treatment (1=a 20% or higher reduction in PANSS score versus baseline, 0=otherwise).

`Week2_bin` The dichotomized change in the PANSS score 2 weeks after starting the treatment.

`Week4_bin` The dichotomized change in the PANSS score 4 weeks after starting the treatment.

`Week6_bin` The dichotomized change in the PANSS score 6 weeks after starting the treatment.

`Week8_bin` The dichotomized change in the PANSS score 8 weeks after starting the treatment.

References

Kay, S.R., Opler, L.A., & Lindenmayer, J.P. (1988). Reliability and validity of the Positive and Negative Syndrome Scale for schizophrenics. Psychiatric Research, 23, 99-110.

sensitivity_analysis_BinCont_copula

Perform Sensitivity Analysis for the Individual Causal Association with a Continuous Surrogate and Binary True Endpoint

Description

Perform Sensitivity Analysis for the Individual Causal Association with a Continuous Surrogate and Binary True Endpoint

Usage

```
sensitivity_analysis_BinCont_copula(
  fitted_model,
  n_sim,
  eq_cond_association = TRUE,
  lower = c(-1, -1, -1, -1),
  upper = c(1, 1, 1, 1),
  marg_association = TRUE,
  n_prec = 10000,
  ncores = 1
)
```

Arguments

`fitted_model` Returned value from `fit_copula_model_BinCont()`. This object contains the estimated identifiable part of the joint distribution for the potential outcomes.

`n_sim` Number of replications in the *sensitivity analysis*. This value should be large enough to sufficiently explore all possible values of the ICA. The minimally sufficient number depends to a large extent on which inequality assumptions are subsequently imposed (see Additional Assumptions).

`eq_cond_association`

Boolean.

- `TRUE` (default): Assume that the association in $(\tilde{S}_1, T_0)'|\tilde{S}_0$ and $(\tilde{S}_0, T_1)'|\tilde{S}_1$ are the same.
- `FALSE`: There is not specific a priori relationship between the above two associations.

`lower` (numeric) Vector of length 4 that provides the lower limit, $\boldsymbol{a} = (a_{23}, a_{13;2}, a_{24;3}, a_{14;23})'$. Defaults to `c(-1, -1, -1, -1)`. If the provided lower limit is smaller than what is allowed for a particular copula family, then the copula family's lowest possible value is used instead.

`upper` (numeric) Vector of length 4 that provides the upper limit, $\boldsymbol{b} = (b_{23}, b_{13;2}, b_{24;3}, b_{14;23})'$. Defaults to `c(1, 1, 1, 1)`.

`marg_association`

Boolean.

- TRUE: Return marginal association measures in each replication in terms of Spearman's rho. The proportion of harmed, protected, never diseased, and always diseased is also returned. See also Value.
- FALSE (default): No additional measures are returned.

n_prec Number of Monte-Carlo samples for the *numerical approximation* of the ICA in each replication of the sensitivity analysis.

ncores Number of cores used in the sensitivity analysis. The computations are computationally heavy, and this option can speed things up considerably.

Value

A data frame is returned. Each row represents one replication in the sensitivity analysis. The returned data frame always contains the following columns:

- R2H, sp_rho, minfo: ICA as quantified by R^2_H, Spearman's rho, and Kendall's tau, respectively.
- c12, c34: estimated copula parameters.
- c23, c13_2, c24_3, c14_23: sampled copula parameters of the unidentifiable copulas in the D-vine copula. The parameters correspond to the parameterization of the copula_family2 copula as in the copula R-package.
- r12, r34: Fixed rotation parameters for the two identifiable copulas.
- r23, r13_2, r24_3, r14_23: Sampled rotation parameters of the unidentifiable copulas in the D-vine copula. These values are constant for the Gaussian copula family since that copula is invariant to rotations.

The returned data frame also contains the following columns when marg_association is TRUE:

- sp_s0s1, sp_s0t0, sp_s0t1, sp_s1t0, sp_s1t1, sp_t0t1: Spearman's rho between the corresponding potential outcomes. Note that these associations refer to the observable potential outcomes. In contrary, the estimated association parameters from fit_copula_model_BinCont() refer to associations on a latent scale.

Information-Theoretic Causal Inference Framework

The information-theoretic causal inference (ITCI) is a general framework to evaluate surrogate endpoints in the single-trial setting (Alonso et al., 2015). In this framework, we focus on the individual causal effects, $\Delta S = S_1 - S_0$ and $\Delta T = T_1 - T_0$ where S_z and T_z are the potential surrogate end true endpoint under treatment $Z = z$.

In the ITCI framework, we say that S is a good surrogate for T if ΔS *conveys a substantial amount of information on* ΔT (Alonso, 2018). This amount of shared information can generally be quantified by the mutual information between ΔS and ΔT, denoted by $I(\Delta S; \Delta T)$. However, the mutual information lies in $[0, +\infty]$ which complicates the interpretation. In addition, the mutual information may not be defined in specific scenarios where absolute continuity of certain probability measures fails. Therefore, the mutual information is transformed, and possibly modified, to enable a simple interpretation in light of the definition of surrogacy. The resulting measure is termed the individual causal association (ICA). This is explained in the next sections.

While the definition of surrogacy in the ITCI framework rests on information theory, shared information is closely related to statistical association. Hence, we can also define the ICA in terms of statistical association measures, like Spearman's rho and Kendall's tau. The advantage of the latter are that they are well-known, simple and rank-based measures of association.

Quantifying Surrogacy

Alonso et al. (na) proposed to the following measure for the ICA:

$$R_H^2 = \frac{I(\Delta S; \Delta T)}{H(\Delta T)}$$

where $H(\Delta T)$ is the entropy of ΔT. By token of that transformation of the mutual information, R_H^2 is restricted to the unit interval where 0 indicates independence, and 1 a functional relationship between ΔS and ΔT.

The association between ΔS and ΔT can also be quantified by Spearman's ρ (or Kendall's τ). This quantity requires appreciably less computing time than the mutual information. This quantity is therefore always returned for every replication of the sensitivity analysis.

Sensitivity Analysis

Monte Carlo Approach:

Because S_0 and S_1 are never simultaneously observed in the same patient, ΔS is not observable, and analogously for ΔT. Consequently, the ICA is unidentifiable. This is solved by considering a (partly identifiable) model for the full vector of potential outcomes, $(T_0, S_0, S_1, T_1)'$. The identifiable parameters are estimated. The unidentifiable parameters are sampled from their parameters space in each replication of a sensitivity analysis. If the number of replications (`n_sim`) is sufficiently large, the entire parameter space for the unidentifiable parameters will be explored/sampled. In each replication, all model parameters are "known" (either estimated or sampled). Consequently, the ICA can be computed in each replication of the sensitivity analysis.

The sensitivity analysis thus results in a set of values for the ICA. This set can be interpreted as *all values for the ICA that are compatible with the observed data*. However, the range of this set is often quite broad; this means there remains too much uncertainty to make judgements regarding the worth of the surrogate. To address this unwieldy uncertainty, additional assumptions can be used that restrict the parameter space of the unidentifiable parameters. This in turn reduces the uncertainty regarding the ICA.

Intervals of Ignorance and Uncertainty:

The results of the sensitivity analysis can be formalized (and summarized) in intervals of ignorance and uncertainty using `sensitivity_intervals_Dvine()`.

Additional Assumptions

There are two possible types of assumptions that restrict the parameter space of the unidentifiable parameters: (i) *equality* type of assumptions, and (ii) *inequality* type of assumptions. These are discussed in turn in the next two paragraphs.

The equality assumptions have to be incorporated into the sensitivity analysis itself. Only one type of equality assumption has been implemented; this is the conditional independence assumption:

$$\tilde{S}_0 \perp T_1 | \tilde{S}_1 \text{ and } \tilde{S}_1 \perp T_0 | \tilde{S}_0.$$

This can informally be interpreted as “what the control treatment does to the surrogate does not provide information on the true endpoint under experimental treatment if we already know what the experimental treatment does to the surrogate", and analogously when control and experimental treatment are interchanged. Note that $\tilde{S}_z$ refers to either the actual potential surrogate outcome, or a latent version. This depends on the content of `fitted_model`.

The inequality type of assumptions have to be imposed on the data frame that is returned by the current function; those assumptions are thus imposed *after* running the sensitivity analysis. If `marginal_association` is set to `TRUE`, the returned data frame contains additional unverifiable quantities that differ across replications of the sensitivity analysis: (i) the unconditional Spearman’s ρ for all pairs of (observable/non-latent) potential outcomes, and (ii) the proportions of the population strata as defined by Nevo and Gorfine (2022) if semi-competing risks are present. More details on the interpretation and use of these assumptions can be found in Stijven et al. (2024).

Examples

```
# Load Schizophrenia data set.
data("Schizo_BinCont")
# Perform listwise deletion.
na = is.na(Schizo_BinCont$CGI_Bin) | is.na(Schizo_BinCont$PANSS)
X = Schizo_BinCont$PANSS[!na]
Y = Schizo_BinCont$CGI_Bin[!na]
Treat = Schizo_BinCont$Treat[!na]
# Ensure that the treatment variable is binary.
Treat = ifelse(Treat == 1, 1, 0)
data = data.frame(X,
                  Y,
                  Treat)
# Fit copula model.
fitted_model = fit_copula_model_BinCont(data, "clayton", "normal", twostep = FALSE)
# Perform sensitivity analysis with a very low number of replications.
sens_results = sensitivity_analysis_BinCont_copula(
  fitted_model,
  10,
  lower = c(-1,-1,-1,-1),
  upper = c(1, 1, 1, 1),
  n_prec = 1e3
)
```

`sensitivity_analysis_copula` *Perform Sensitivity Analysis for the Individual Causal Association based on a D-vine copula model*

Description

Perform Sensitivity Analysis for the Individual Causal Association based on a D-vine copula model

Usage

```
sensitivity_analysis_copula(
  fitted_model,
  n_sim,
  eq_cond_association = TRUE,
  lower = c(-1, -1, -1, -1),
  upper = c(1, 1, 1, 1),
  degrees = c(0, 90, 180, 270),
  marg_association = TRUE,
  copula_family2 = fitted_model$copula_family[1],
  n_prec = 10000,
  ncores = 1,
  ICA_estimator = NULL
)
```

Arguments

`fitted_model` Returned value from `fit_copula_OrdOrd()`, `fit_copula_OrdCont()`, or `fit_copula_ContCont()`. This object contains the estimated identifiable part of the joint distribution for the potential outcomes.

`n_sim` Number of replications in the *sensitivity analysis*. This value should be large enough to sufficiently explore all possible values of the ICA. The minimally sufficient number depends to a large extent on which inequality assumptions are subsequently imposed (see Additional Assumptions).

`eq_cond_association`

Boolean.

- `TRUE` (default): Assume that the association in $(\tilde{S}_1, T_0)'|\tilde{S}_0$ and $(\tilde{S}_0, T_1)'|\tilde{S}_1$ are the same.
- `FALSE`: There is not specific a priori relationship between the above two associations.

`lower` (numeric) Vector of length 4 that provides the lower limit, $\boldsymbol{a} = (a_{23}, a_{13;2}, a_{24;3}, a_{14;23})'$. Defaults to `c(-1, -1, -1, -1)`. If the provided lower limit is smaller than what is allowed for a particular copula family, then the copula family's lowest possible value is used instead.

`upper` (numeric) Vector of length 4 that provides the upper limit, $\boldsymbol{b} = (b_{23}, b_{13;2}, b_{24;3}, b_{14;23})'$. Defaults to `c(1, 1, 1, 1)`.

`degrees` (numeric) vector with copula rotation degrees. Defaults to `c(0, 90, 180, 270)`. This argument is not used for the Gaussian and Frank copulas since they already allow for positive and negative associations.

`marg_association`

Boolean.

- `TRUE`: Return marginal association measures in each replication in terms of Spearman's rho. The proportion of harmed, protected, never diseased, and always diseased is also returned. See also Value.
- `FALSE` (default): No additional measures are returned.

`copula_family2` Copula family of the other bivariate copulas. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family2` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`n_prec` Number of Monte-Carlo samples for the *numerical approximation* of the ICA in each replication of the sensitivity analysis.

`ncores` Number of cores used in the sensitivity analysis. The computations are computationally heavy, and this option can speed things up considerably.

`ICA_estimator` Function that estimates the ICA between the first two arguments which are numeric vectors. See also `compute_ICA_OrdOrd()`, `compute_ICA_OrdCont()`, and `compute_ICA_ContCont()`.

Value

A data frame is returned. Each row represents one replication in the sensitivity analysis. The returned data frame always contains the following columns:

- `R2H`, `sp_rho`: ICA as quantified by R^2_H and Spearman's rho, respectively.
- `c12`, `c34`: estimated copula parameters.
- `c23`, `c13_2`, `c24_3`, `c14_23`: sampled copula parameters of the unidentifiable copulas in the D-vine copula. The parameters correspond to the parameterization of the `copula_family2` copula as in the `copula` R-package.
- `r12`, `r34`: Fixed rotation parameters for the two identifiable copulas.
- `r23`, `r13_2`, `r24_3`, `r14_23`: Sampled rotation parameters of the unidentifiable copulas in the D-vine copula. These values are constant for the Gaussian copula family since that copula is invariant to rotations.

The returned data frame also contains the following columns when `marg_association` is `TRUE`:

- `sp_s0s1`, `sp_s0t0`, `sp_s0t1`, `sp_s1t0`, `sp_s1t1`, `sp_t0t1`: Spearman's rho between the corresponding potential outcomes. Note that these associations refer to the observable potential outcomes. In contrast, the estimated association parameters from `fit_copula_OrdOrd()` and fit_copula_OrdCont refer to associations on a latent scale.

Information-Theoretic Causal Inference Framework

The information-theoretic causal inference (ITCI) is a general framework to evaluate surrogate endpoints in the single-trial setting (Alonso et al., 2015). In this framework, we focus on the individual causal effects, $\Delta S = S_1 - S_0$ and $\Delta T = T_1 - T_0$ where S_z and T_z are the potential surrogate end true endpoint under treatment $Z = z$.

In the ITCI framework, we say that S is a good surrogate for T if ΔS *conveys a substantial amount of information on* ΔT (Alonso, 2018). This amount of shared information can generally be quantified by the mutual information between ΔS and ΔT, denoted by $I(\Delta S; \Delta T)$. However, the mutual information lies in $[0, +\infty]$ which complicates the interpretation. In addition, the mutual information may not be defined in specific scenarios where absolute continuity of certain probability measures fails. Therefore, the mutual information is transformed, and possibly modified, to enable a simple interpretation in light of the definition of surrogacy. The resulting measure is termed the individual causal association (ICA). This is explained in the next sections.

While the definition of surrogacy in the ITCI framework rests on information theory, shared information is closely related to statistical association. Hence, we can also define the ICA in terms of statistical association measures, like Spearman's rho and Kendall's tau. The advantage of the latter are that they are well-known, simple and rank-based measures of association.

Individual Causal Association

Many association measures can operationalize the ICA. For each setting, we consider one default definition for the ICA which follows from the mutual information.

Continuous-Continuous:
The ICA is defined as the squared informational coefficient of correlation (SICC or R^2_H), which is a transformation of the mutual information to the unit interval:

$$R^2_h = 1 - e^{-2 \cdot I(\Delta S; \Delta T)}$$

where 0 indicates independence, and 1 a functional relationship between ΔS and ΔT. If $(\Delta S, \Delta T)'$ is bivariate normal, the ICA equals the Pearson correlation between ΔS and ΔT.

Ordinal-Continuous:
The ICA is defined as the following transformation of the mutual information:

$$R^2_H = \frac{I(\Delta S; \Delta T)}{H(\Delta T)},$$

where $I(\Delta S; \Delta T)$ is the mutual information and $H(\Delta T)$ the entropy.

Ordinal-Ordinal:
The ICA is defined as the following transformation of the mutual information:

$$R^2_H = \frac{I(\Delta S; \Delta T)}{\min\{H(\Delta S), H(\Delta T)\}},$$

where $I(\Delta S; \Delta T)$ is the mutual information, and $H(\Delta S)$ and $H(\Delta T)$ the entropy of ΔS and ΔT, respectively.

Sensitivity Analysis

Monte Carlo Approach:
Because S_0 and S_1 are never simultaneously observed in the same patient, ΔS is not observable, and analogously for ΔT. Consequently, the ICA is unidentifiable. This is solved by considering a (partly identifiable) model for the full vector of potential outcomes, $(T_0, S_0, S_1, T_1)'$. The identifiable parameters are estimated. The unidentifiable parameters are sampled from their parameters space in each replication of a sensitivity analysis. If the number of replications (`n_sim`) is sufficiently large, the entire parameter space for the unidentifiable parameters will be explored/sampled. In each replication, all model parameters are "known" (either estimated or sampled). Consequently, the ICA can be computed in each replication of the sensitivity analysis.

The sensitivity analysis thus results in a set of values for the ICA. This set can be interpreted as *all values for the ICA that are compatible with the observed data*. However, the range of this set is often quite broad; this means there remains too much uncertainty to make judgements regarding the worth of the surrogate. To address this unwieldy uncertainty, additional assumptions can be used that restrict the parameter space of the unidentifiable parameters. This in turn reduces the uncertainty regarding the ICA.

Intervals of Ignorance and Uncertainty:

The results of the sensitivity analysis can be formalized (and summarized) in intervals of ignorance and uncertainty using `sensitivity_intervals_Dvine()`.

Additional Assumptions

There are two possible types of assumptions that restrict the parameter space of the unidentifiable parameters: (i) *equality* type of assumptions, and (ii) *inequality* type of assumptions. These are discussed in turn in the next two paragraphs.

The equality assumptions have to be incorporated into the sensitivity analysis itself. Only one type of equality assumption has been implemented; this is the conditional independence assumption:

$$\tilde{S}_0 \perp T_1 | \tilde{S}_1 \text{ and } \tilde{S}_1 \perp T_0 | \tilde{S}_0.$$

This can informally be interpreted as “what the control treatment does to the surrogate does not provide information on the true endpoint under experimental treatment if we already know what the experimental treatment does to the surrogate", and analogously when control and experimental treatment are interchanged. Note that $\tilde{S}_z$ refers to either the actual potential surrogate outcome, or a latent version. This depends on the content of `fitted_model`.

The inequality type of assumptions have to be imposed on the data frame that is returned by the current function; those assumptions are thus imposed *after* running the sensitivity analysis. If `marginal_association` is set to `TRUE`, the returned data frame contains additional unverifiable quantities that differ across replications of the sensitivity analysis: (i) the unconditional Spearman's ρ for all pairs of (observable/non-latent) potential outcomes, and (ii) the proportions of the population strata as defined by Nevo and Gorfine (2022) if semi-competing risks are present. More details on the interpretation and use of these assumptions can be found in Stijven et al. (2024).

References

Alonso, A. (2018). An information-theoretic approach for the evaluation of surrogate endpoints. In Wiley StatsRef: Statistics Reference Online. John Wiley & Sons, Ltd.

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., and Burzykowski, T. (2015). On the relationship between the causal-inference and meta-analytic paradigms for the validation of surrogate endpoints. Biometrics 71, 15–24.

`sensitivity_analysis_SurvSurv_copula`

Sensitivity analysis for individual causal association

Description

The `sensitivity_analysis_SurvSurv_copula()` function performs the sensitivity analysis for the individual causal association (ICA) as described by Stijven et al. (2024).

Usage

```
sensitivity_analysis_SurvSurv_copula(
  fitted_model,
  composite = TRUE,
  n_sim,
  eq_cond_association = TRUE,
  lower = c(-1, -1, -1, -1),
  upper = c(1, 1, 1, 1),
  degrees = c(0, 90, 180, 270),
  marg_association = TRUE,
  copula_family2 = fitted_model$copula_family[1],
  n_prec = 5000,
  ncores = 1,
  sample_plots = NULL,
  mutinfo_estimator = NULL,
  restr_time = +Inf
)
```

Arguments

`fitted_model` Returned value from `fit_model_SurvSurv()`. This object contains the estimated identifiable part of the joint distribution for the potential outcomes.

`composite` (boolean) If `composite` is `TRUE`, then the surrogate endpoint is a composite of both a "pure" surrogate endpoint and the true endpoint, e.g., progression-free survival is the minimum of time-to-progression and time-to-death.

`n_sim` Number of replications in the *sensitivity analysis*. This value should be large enough to sufficiently explore all possible values of the ICA. The minimally sufficient number depends to a large extent on which inequality assumptions are subsequently imposed (see Additional Assumptions).

`eq_cond_association`

Boolean.

- `TRUE` (default): Assume that the association in $(\tilde{S}_1, T_0)' | \tilde{S}_0$ and $(\tilde{S}_0, T_1)' | \tilde{S}_1$ are the same.
- `FALSE`: There is not specific a priori relationship between the above two associations.

`lower` (numeric) Vector of length 4 that provides the lower limit, $\boldsymbol{a} = (a_{23}, a_{13;2}, a_{24;3}, a_{14;23})'$. Defaults to `c(-1, -1, -1, -1)`. If the provided lower limit is smaller than what is allowed for a particular copula family, then the copula family's lowest possible value is used instead.

`upper` (numeric) Vector of length 4 that provides the upper limit, $\boldsymbol{b} = (b_{23}, b_{13;2}, b_{24;3}, b_{14;23})'$. Defaults to `c(1, 1, 1, 1)`.

`degrees` (numeric) vector with copula rotation degrees. Defaults to `c(0, 90, 180, 270)`. This argument is not used for the Gaussian and Frank copulas since they already allow for positive and negative associations.

`marg_association`

Boolean.

- `TRUE`: Return marginal association measures in each replication in terms of Spearman's rho. The proportion of harmed, protected, never diseased, and always diseased is also returned. See also Value.
- `FALSE` (default): No additional measures are returned.

`copula_family2` Copula family of the other bivariate copulas. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family2` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`n_prec` Number of Monte-Carlo samples for the *numerical approximation* of the ICA in each replication of the sensitivity analysis.

`ncores` Number of cores used in the sensitivity analysis. The computations are computationally heavy, and this option can speed things up considerably.

`sample_plots` Indices for replicates in the sensitivity analysis for which the sampled individual treatment effects are plotted. Defaults to `NULL`: no plots are displayed.

`mutinfo_estimator`

Function that estimates the mutual information between the first two arguments which are numeric vectors. Defaults to `FNN::mutinfo()` with default arguments.

`restr_time` Restriction time for the potential outcomes. Defaults to `+Inf` which means no restriction. Otherwise, the sampled potential outcomes are replace by `pmin(S0, restr_time)` (and similarly for the other potential outcomes).

Value

A data frame is returned. Each row represents one replication in the sensitivity analysis. The returned data frame always contains the following columns:

- `ICA`, `sp_rho`: ICA as quantified by $R^2_h(\Delta S^*, \Delta T^*)$ and $\rho_s(\Delta S, \Delta T)$.
- `c23`, `c13_2`, `c24_3`, `c14_23`: sampled copula parameters of the unidentifiable copulas in the D-vine copula. The parameters correspond to the parameterization of the `copula_family2` copula as in the `copula` R-package.
- `r23`, `r13_2`, `r24_3`, `r14_23`: sampled rotation parameters of the unidentifiable copulas in the D-vine copula. These values are constant for the Gaussian copula family since that copula is invariant to rotations.

  The returned data frame also contains the following columns when `get_marg_tau` is `TRUE`:
- `sp_s0s1`, `sp_s0t0`, `sp_s0t1`, `sp_s1t0`, `sp_s1t1`, `sp_t0t1`: Spearman's ρ between the corresponding potential outcomes. Note that these associations refer to the potential time-to-composite events and/or time-to-true endpoint event. In contrary, the estimated association parameters from `fit_model_SurvSurv()` refer to associations between the time-to-surrogate event and time-to true endpoint event. Also note that `sp_s1t1` is constant whereas `sp_s0t0` is not. This is a particularity of the MC procedure to calculate both measures and thus not a bug.
- `prop_harmed`, `prop_protected`, `prop_always`, `prop_never`: proportions of the corresponding population strata in each replication. These are defined in Nevo and Gorfine (2022).

Information-Theoretic Causal Inference Framework

The information-theoretic causal inference (ITCI) is a general framework to evaluate surrogate endpoints in the single-trial setting (Alonso et al., 2015). In this framework, we focus on the individual causal effects, $\Delta S = S_1 - S_0$ and $\Delta T = T_1 - T_0$ where S_z and T_z are the potential surrogate end true endpoint under treatment $Z = z$.

In the ITCI framework, we say that S is a good surrogate for T if ΔS *conveys a substantial amount of information on* ΔT (Alonso, 2018). This amount of shared information can generally be quantified by the mutual information between ΔS and ΔT, denoted by $I(\Delta S; \Delta T)$. However, the mutual information lies in $[0, +\infty]$ which complicates the interpretation. In addition, the mutual information may not be defined in specific scenarios where absolute continuity of certain probability measures fails. Therefore, the mutual information is transformed, and possibly modified, to enable a simple interpretation in light of the definition of surrogacy. The resulting measure is termed the individual causal association (ICA). This is explained in the next sections.

While the definition of surrogacy in the ITCI framework rests on information theory, shared information is closely related to statistical association. Hence, we can also define the ICA in terms of statistical association measures, like Spearman's rho and Kendall's tau. The advantage of the latter are that they are well-known, simple and rank-based measures of association.

Surrogacy in The Survival-Survival Setting

General Introduction:

Stijven et al. (2024) proposed to quantify the ICA through the squared informational coefficient of correlation (SICC or R^2_H), which is a transformation of the mutual information to the unit interval:

$$R^2_H = 1 - e^{-2 \cdot I(\Delta S; \Delta T)}$$

where 0 indicates independence, and 1 a functional relationship between ΔS and ΔT. The ICA (or a modified version, see next) is returned by `sensitivity_analysis_SurvSurv_copula()`. Concurrently, the Spearman's correlation between ΔS and ΔT is also returned.

Issues with Composite Endpoints:

In the survival-survival setting where the surrogate is a composite endpoint, care should be taken when defining the mutual information. Indeed, when S_z is progression-free survival and T_z is overall survival, there is a probability atom in the joint distribution of $(S_z, T_z)'$ because $P(S_z = T_z) > 0$. In other words, there are patient that die before progressing. While this probability atom is correctly taken into account in the models fitted by `fit_model_SurvSurv()`, this probability atom reappears when considering the distribution of $(\Delta S, \Delta T)'$ because $P(\Delta S = \Delta T) > 0$ if we are considering PFS and OS.

Because of the atom in the distribution of $(\Delta S, \Delta T)'$, the corresponding mutual information is not defined. To solve this, the mutual information is computed excluding the patients for which $\Delta S = \Delta T$ when `composite = TRUE`. The proportion of excluded patients is, among other things, returned when `marginal_association = TRUE`. This is the proportion of "never" patients following the classification of Nevo and Gorfine (2022). See also Additional Assumptions.

This modified version of the ICA quantifies the surrogacy of S when "adjusted for the composite nature of S". Indeed, we exclude patients where ΔS perfectly predicts ΔT *just because S is a composite of T (and other variables).

Other (rank-based) statistical measures of association, however, remain well-defined and are thus computed without excluding any patients.

Sensitivity Analysis

Monte Carlo Approach:

Because S_0 and S_1 are never simultaneously observed in the same patient, ΔS is not observable, and analogously for ΔT. Consequently, the ICA is unidentifiable. This is solved by considering a (partly identifiable) model for the full vector of potential outcomes, $(T_0, S_0, S_1, T_1)'$. The identifiable parameters are estimated. The unidentifiable parameters are sampled from their parameters space in each replication of a sensitivity analysis. If the number of replications (`n_sim`) is sufficiently large, the entire parameter space for the unidentifiable parameters will be explored/sampled. In each replication, all model parameters are "known" (either estimated or sampled). Consequently, the ICA can be computed in each replication of the sensitivity analysis.

The sensitivity analysis thus results in a set of values for the ICA. This set can be interpreted as *all values for the ICA that are compatible with the observed data*. However, the range of this set is often quite broad; this means there remains too much uncertainty to make judgements regarding the worth of the surrogate. To address this unwieldy uncertainty, additional assumptions can be used that restrict the parameter space of the unidentifiable parameters. This in turn reduces the uncertainty regarding the ICA.

Intervals of Ignorance and Uncertainty:

The results of the sensitivity analysis can be formalized (and summarized) in intervals of ignorance and uncertainty using `sensitivity_intervals_Dvine()`.

Additional Assumptions

There are two possible types of assumptions that restrict the parameter space of the unidentifiable parameters: (i) *equality* type of assumptions, and (ii) *inequality* type of assumptions. These are discussed in turn in the next two paragraphs.

The equality assumptions have to be incorporated into the sensitivity analysis itself. Only one type of equality assumption has been implemented; this is the conditional independence assumption:

$$\tilde{S}_0 \perp T_1 | \tilde{S}_1 \text{ and } \tilde{S}_1 \perp T_0 | \tilde{S}_0.$$

This can informally be interpreted as “what the control treatment does to the surrogate does not provide information on the true endpoint under experimental treatment if we already know what the experimental treatment does to the surrogate", and analogously when control and experimental treatment are interchanged. Note that $\tilde{S}_z$ refers to either the actual potential surrogate outcome, or a latent version. This depends on the content of `fitted_model`.

The inequality type of assumptions have to be imposed on the data frame that is returned by the current function; those assumptions are thus imposed *after* running the sensitivity analysis. If `marginal_association` is set to `TRUE`, the returned data frame contains additional unverifiable quantities that differ across replications of the sensitivity analysis: (i) the unconditional Spearman's ρ for all pairs of (observable/non-latent) potential outcomes, and (ii) the proportions of the population strata as defined by Nevo and Gorfine (2022) if semi-competing risks are present. More details on the interpretation and use of these assumptions can be found in Stijven et al. (2024).

References

Alonso, A. (2018). An information-theoretic approach for the evaluation of surrogate endpoints. In Wiley StatsRef: Statistics Reference Online. John Wiley & Sons, Ltd.

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., and Burzykowski, T. (2015). On the relationship between the causal-inference and meta-analytic paradigms for the validation of surrogate endpoints. Biometrics 71, 15–24.

Stijven, F., Alonso, a., Molenberghs, G., Van Der Elst, W., Van Keilegom, I. (2024). An information-theoretic approach to the evaluation of time-to-event surrogates for time-to-event true endpoints based on causal inference.

Nevo, D., & Gorfine, M. (2022). Causal inference for semi-competing risks data. Biostatistics, 23 (4), 1115-1132

Examples

```
# Load Ovarian data
data("Ovarian")
# Recode the Ovarian data in the semi-competing risks format.
data_scr = data.frame(
  ttp = Ovarian$Pfs,
  os = Ovarian$Surv,
  treat = Ovarian$Treat,
  ttp_ind = ifelse(
    Ovarian$Pfs == Ovarian$Surv &
      Ovarian$SurvInd == 1,
    0,
    Ovarian$PfsInd
  ),
  os_ind = Ovarian$SurvInd
)
# Fit copula model.
fitted_model = fit_model_SurvSurv(data = data_scr,
                                  copula_family = "clayton",
                                  n_knots = 1)
# Illustration with small number of replications and low precision
sens_results = sensitivity_analysis_SurvSurv_copula(fitted_model,
                  n_sim = 5,
                  n_prec = 2000,
                  copula_family2 = "clayton",
                  eq_cond_association = TRUE)
# Compute intervals of ignorance and uncertainty. Again, the number of
# bootstrap replications should be larger in practice.
sensitivity_intervals_Dvine(fitted_model, sens_results, B = 10)
```

sensitivity_intervals_Dvine *Compute Sensitivity Intervals*

Description

sensitivity_intervals_Dvine() computes the estimated intervals of ignorance and uncertainty within the information-theoretic causal inference framework when the data are modeled with a D-vine copula model.

Usage

```
sensitivity_intervals_Dvine(
  fitted_model,
  sens_results,
  measure = "ICA",
  B = 200,
  alpha = 0.05,
  n_prec = 5000,
  mutinfo_estimator = NULL,
  ICA_estimator = NULL,
  restr_time = +Inf,
  ncores = 1
)
```

Arguments

`fitted_model`	Returned value from `fit_model_SurvSurv()`. This object contains the estimated identifiable part of the joint distribution for the potential outcomes.
`sens_results`	Dataframe returned by `sensitivity_analysis_SurvSurv_copula()`. If additional assumptions need to be incorporated, this dataframe can first be filtered.
`measure`	Compute intervals for which measure of surrogacy? Defaults to `"ICA"`. See first column names of `sens_results` for other possibilities.
`B`	Number of bootstrap replications
`alpha`	(numeric) `1 - alpha` is the level of the confidence interval
`n_prec`	Number of Monte-Carlo samples for the *numerical approximation* of the ICA in each replication of the sensitivity analysis.
`mutinfo_estimator`	Function that estimates the mutual information between the first two arguments which are numeric vectors. Defaults to `FNN::mutinfo()` with default arguments.
`ICA_estimator`	Function that estimates the ICA between the first two arguments which are numeric vectors. Defaults to `NULL` which corresponds to using `estimate_ICA_ContCont()`, `estimate_ICA_OrdCont()`, or `estimate_ICA_OrdOrd()` (depending on the endpoint types). This argument is not used in the survival-survival setting.
`restr_time`	Restriction time for the potential outcomes. Defaults to `+Inf` which means no restriction. Otherwise, the sampled potential outcomes are replace by `pmin(S0, restr_time)` (and similarly for the other potential outcomes).
`ncores`	Number of cores used in the sensitivity analysis. The computations are computationally heavy, and this option can speed things up considerably.

Value

An S3 object of the class `sensitivity_intervals_Dvine` which can be printed.

Intervals of Ignorance and Uncertainty

Vansteelandt et al. (2006) formalized sensitivity analysis for partly identifiable parameters in the context of missing data and MNAR. These concepts can be applied to the estimation of the ICA. Indeed, the ICA is also partly identifiable because 50% if the potential outcomes are missing.

Vansteelandt et al. (2006) replace a point estimate with a interval estimate: the estimated interval of ignorance. In addition, they proposed several extension of the classic confidence interval together with appropriate definitions of coverage; these are termed intervals of uncertainty.

sensitivity_intervals_Dvine() implements the estimated interval of ignorance and the pointwise and strong intervals of uncertainty. Let $\boldsymbol{\nu}_l$ and $\boldsymbol{\nu}_u$ be the values for the sensitivity parameter that lead to the lowest and largest ICA, respectively, while fixing the identifiable parameter at its estimated value $\hat{\boldsymbol{\beta}}$. See also summary_level_bootstrap_ICA(). The following intervals are implemented:

1. *Estimated interval of ignorance*. This interval is defined as $[ICA(\hat{\boldsymbol{\beta}}, \boldsymbol{\nu}_l), ICA(\hat{\boldsymbol{\beta}}, \boldsymbol{\nu}_u)]$.
2. *Pointiwse interval of uncertainty*. Let C_l (and C_u) be the lower (and upper) limit of a one-sided $1 - \alpha$ CI for $ICA(\boldsymbol{\beta_0}, \boldsymbol{\nu}_l)$ (and $ICA(\boldsymbol{\beta_0}, \boldsymbol{\nu}_l)$). This interval is then defined as $[C_l, C_u]$ when the ignorance is much larger than the statistical imprecision.
3. *Strong interval of uncertainty*. Let C_l (and C_u) be the lower (and upper) limit of a two-sided $1 - \alpha$ CI for $ICA(\boldsymbol{\beta_0}, \boldsymbol{\nu}_l)$ (and $ICA(\boldsymbol{\beta_0}, \boldsymbol{\nu}_l)$). This interval is then defined as $[C_l, C_u]$.

The CIs, which are need for the intervals of uncertainty, are based on percentile bootstrap confidence intervals, as documented in summary_level_bootstrap_ICA(). In addition, $\boldsymbol{\nu}_l$ is not known. Therefore, it is estimated as

$$\arg\min_{\boldsymbol{\nu} \in \Gamma} ICA(\hat{\boldsymbol{\beta}}, \boldsymbol{\nu}),$$

and similarly for $\boldsymbol{\nu}_u$.

References

Vansteelandt, Stijn, et al. "Ignorance and uncertainty regions as inferential tools in a sensitivity analysis." Statistica Sinica (2006): 953-979.

Examples

```
# Load Ovarian data
data("Ovarian")
# Recode the Ovarian data in the semi-competing risks format.
data_scr = data.frame(
  ttp = Ovarian$Pfs,
  os = Ovarian$Surv,
  treat = Ovarian$Treat,
  ttp_ind = ifelse(
    Ovarian$Pfs == Ovarian$Surv &
      Ovarian$SurvInd == 1,
    0,
    Ovarian$PfsInd
  ),
  os_ind = Ovarian$SurvInd
)
```

```
# Fit copula model.
fitted_model = fit_model_SurvSurv(data = data_scr,
                                  copula_family = "clayton",
                                  n_knots = 1)
# Illustration with small number of replications and low precision
sens_results = sensitivity_analysis_SurvSurv_copula(fitted_model,
                  n_sim = 5,
                  n_prec = 2000,
                  copula_family2 = "clayton",
                  eq_cond_association = TRUE)
# Compute intervals of ignorance and uncertainty. Again, the number of
# bootstrap replications should be larger in practice.
sensitivity_intervals_Dvine(fitted_model, sens_results, B = 10)
```

Sim.Data.Counterfactuals

Simulate a dataset that contains counterfactuals

Description

The function Sim.Data.Counterfactuals simulates a dataset that contains four (continuous) counterfactuals (i.e., potential outcomes) and a (binary) treatment indicator. The counterfactuals T_0 and T_1 denote the true endpoints of a patient under the control and the experimental treatments, respectively, and the counterfactuals S_0 and S_1 denote the surrogate endpoints of the patient under the control and the experimental treatments, respectively. The user can specify the number of patients, the desired mean values for the counterfactuals (i.e., $\boldsymbol{\mu}_c$), and the desired correlations between the counterfactuals (i.e., the off-diagonal values in the standardized $\boldsymbol{\Sigma}_c$ matrix). For details, see the papers of Alonso et al. (submitted) and Van der Elst et al. (submitted).

Usage

```
Sim.Data.Counterfactuals(N.Total=2000,
mu_c=c(0, 0, 0, 0), T0S0=0, T1S1=0, T0T1=0, T0S1=0,
T1S0=0, S0S1=0, Seed=sample(1:1000, size=1))
```

Arguments

N.Total — The total number of patients in the simulated dataset. Default 2000.

mu_c — A vector that specifies the desired means for the counterfactuals S_0, S_1, T_0, and T_1, respectively. Default c(0, 0, 0, 0).

T0S0 — A scalar that specifies the desired correlation between the counterfactuals T0 and S0 that should be used in the generation of the data. Default 0.

T1S1 — A scalar that specifies the desired correlation between the counterfactuals T1 and S1 that should be used in the generation of the data. Default 0.

T0T1 — A scalar that specifies the desired correlation between the counterfactuals T0 and T1 that should be used in the generation of the data. Default 0.

T0S1	A scalar that specifies the desired correlation between the counterfactuals T0 and S1 that should be used in the generation of the data. Default 0.
T1S0	A scalar that specifies the desired correlation between the counterfactuals T1 and S0 that should be used in the generation of the data. Default 0.
S0S1	A scalar that specifies the desired correlation between the counterfactuals T0 and T1 that should be used in the generation of the data. Default 0.
Seed	A seed that is used to generate the dataset. Default sample(x=1:1000, size=1), i.e., a random number between 1 and 1000.

Details

The generated object Data.Counterfactuals (of class data.frame) is placed in the workspace.

The specified values for T0S0, T1S1, T0T1, T0S1, T1S0, and S0S1 in the function call should form a matrix that is positive definite (i.e., they should form a valid correlation matrix). When the user specifies values that form a matrix that is not positive definite, an error message is given and the object Data.Counterfactuals is not generated. The function Pos.Def.Matrices can be used to examine beforehand whether a 4 by 4 matrix is positive definite.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., Molenberghs, G., Buyse, M., & Burzykowski, T. (submitted). On the relationship between the causal inference and meta-analytic paradigms for the validation of surrogate markers.

Van der Elst, W., Alonso, A., & Molenberghs, G. (submitted). An exploration of the relationship between causal inference and meta-analytic measures of surrogacy.

See Also

Sim.Data.MTS, Sim.Data.STS

Examples

```
## Generate a dataset with 2000 patients, cor(S0,T0)=cor(S1,T1)=.5,
## cor(T0,T1)=cor(T0,S1)=cor(T1,S0)=cor(S0,S1)=0, with means
## 5, 9, 12, and 15 for S0, S1, T0, and T1, respectively:
Sim.Data.Counterfactuals(N=2000, T0S0=.5, T1S1=.5, T0T1=0, T0S1=0, T1S0=0, S0S1=0,
mu_c=c(5, 9, 12, 15), Seed=1)
```

Sim.Data.CounterfactualsBinBin

Simulate a dataset that contains counterfactuals for binary endpoints

Description

The function `Sim.Data.CounterfactualsBinBin` simulates a dataset that contains four (binary) counterfactuals (i.e., potential outcomes) and a (binary) treatment indicator. The counterfactuals T_0 and T_1 denote the true endpoints of a patient under the control and the experimental treatments, respectively, and the counterfactuals S_0 and S_1 denote the surrogate endpoints of the patient under the control and the experimental treatments, respectively. The user can specify the number of patients and the desired probabilities of the vector of potential outcomes (i.e., $\boldsymbol{Y'_c}$=(T_0, T_1, S_0, S_1)).

Usage

```
Sim.Data.CounterfactualsBinBin(Pi_s=rep(1/16, 16),
N.Total=2000, Seed=sample(1:1000, size=1))
```

Arguments

`Pi_s` The vector of probabilities of the potential outcomes, i.e., pi_{0000}, pi_{0100}, pi_{0010}, pi_{0001}, pi_{0101}, pi_{1000}, pi_{1010}, pi_{1001}, pi_{1110}, pi_{1101}, pi_{1011}, pi_{1111}, pi_{0110}, pi_{0011}, pi_{0111}, pi_{1100}. Default `rep(1/16, 16)`.

`N.Total` The desired number of patients in the simulated dataset. Default 2000.

`Seed` A seed that is used to generate the dataset. Default `sample(x=1:1000, size=1)`, i.e., a random number between 1 and 1000.

Details

The generated object `Data.STSBinBin.Counter` (which contains the counterfactuals) and `Data.STSBinBin.Obs` (the "observable data") (of class `data.frame`) is placed in the workspace.

Value

An object of class `Sim.Data.CounterfactualsBinBin` with components,

`Data.STSBinBin.Obs`
The generated dataset that contains the "observed" surrogate endrpoint, true endpoint, and assigned treatment.

`Data.STSBinBin.Counter`
The generated dataset that contains the counterfactuals.

`Vector_Pi` The vector of probabilities of the potential outcomes, i.e., pi_{0000}, pi_{0100}, pi_{0010}, pi_{0001}, pi_{0101}, pi_{1000}, pi_{1010}, pi_{1001}, pi_{1110}, pi_{1101}, pi_{1011}, pi_{1111}, pi_{0110}, pi_{0011}, pi_{0111}, pi_{1100}.

`Pi_Marginals` The vector of marginal probabilities $\pi_{1 \cdot 1 \cdot}$, $\pi_{0 \cdot 1 \cdot}$, $\pi_{1 \cdot 0 \cdot}$, $\pi_{0 \cdot 0 \cdot}$, $\pi_{\cdot 1 \cdot 1}$, $\pi_{\cdot 1 \cdot 0}$, $\pi_{\cdot 0 \cdot 1}$, $\pi_{\cdot 0 \cdot 0}$.

True.R2_H	The true R_H^2 value.
True.Theta_T	The true odds ratio for T.
True.Theta_S	The true odds ratio for S.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

Examples

```
## Generate a dataset with 2000 patients, and values 1/16
## for all proabilities between the counterfactuals:
Sim.Data.CounterfactualsBinBin(N.Total=2000)
```

Sim.Data.MTS *Simulates a dataset that can be used to assess surrogacy in the multiple-trial setting*

Description

The function Sim.Data.MTS simulates a dataset that contains the variables Treat, Trial.ID, Surr, True, and Pat.ID. The user can specify the number of patients and the number of trials that should be included in the simulated dataset, the desired R_{trial} and R_{indiv} values, the desired variability of the trial-specific treatment effects for the surrogate and the true endpoints (i.e., d_{aa} and d_{bb}, respectively), and the desired fixed-effect parameters of the intercepts and treatment effects for the surrogate and the true endpoints.

Usage

```
Sim.Data.MTS(N.Total=2000, N.Trial=50, R.Trial.Target=.8, R.Indiv.Target=.8,
Fixed.Effects=c(0, 0, 0, 0), D.aa=10, D.bb=10, Seed=sample(1:1000, size=1),
Model=c("Full"))
```

Arguments

N.Total	The total number of patients in the simulated dataset. Default 2000.
N.Trial	The number of trials. Default 50.
R.Trial.Target	The desired R_{trial} value in the sumilated dataset. Default 0.80
R.Indiv.Target	The desired R_{indiv} value in the simulated dataset. Default 0.80.
Fixed.Effects	A vector that specifies the desired fixed-effect intercept for the surrogate, fixed-effect intercept for the true endpoint, fixed treatment effect for the surrogate, and fixed treatment effect for the true endpoint, respectively. Default c(0, 0, 0, 0).
D.aa	The desired variability of the trial-specific treatment effects on the surrogate endpoint. Default 10.
D.bb	The desired variability of the trial-specific treatment effects on the true endpoint. Default 10.

`Model` The type of model that will be fitted on the data when surrogacy is assessed, i.e., a full, semireduced, or reduced model (for details, see `UnifixedContCont`, `UnimixedContCont`, `BifixedContCont`, `BimixedContCont`).

`Seed` The seed that is used to generate the dataset. Default `sample(x=1:1000, size=1)`, i.e., a random number between 1 and 1000.

Details

The generated object `Data.Observed.MTS` (of class `data.frame`) is placed in the workspace (for easy access).

The number of patients per trial in the simulated dataset is identical in each trial, and equals the requested total number of patients divided by the requested number of trials (=`N.Total/N.Trial`). If this is not a whole number, a warning is given and the number of patients per trial is automatically rounded up to the nearest whole number. See **Examples** below.

Treatment allocation is balanced when the number of patients per trial is an odd number. If this is not the case, treatment allocation is balanced up to one patient (the remaining patient is randomly allocated to the exprimental or the control treatment groups in each of the trials).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

See Also

`UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`, `BimixedContCont`, `Sim.Data.STS`

Examples

```
# Simulate a dataset with 2000 patients, 50 trials, Rindiv=Rtrial=.8, D.aa=10,
# D.bb=50, and fixed effect values 1, 2, 30, and 90:
Sim.Data.MTS(N.Total=2000, N.Trial=50, R.Trial.Target=.8, R.Indiv.Target=.8, D.aa=10,
D.bb=50, Fixed.Effects=c(1, 2, 30, 90), Seed=1)

# Sample output, the first 10 rows of Data.Observed.MTS:
Data.Observed.MTS[1:10,]

# Note: When the following code is used to generate a dataset:
Sim.Data.MTS(N.Total=2000, N.Trial=99, R.Trial.Target=.5, R.Indiv.Target=.8,
D.aa=10, D.bb=50, Fixed.Effects=c(1, 2, 30, 90), Seed=1)

# R gives the following warning:

# > NOTE: The number of patients per trial requested in the function call
# > equals 20.20202 (=N.Total/N.Trial), which is not a whole number.
# > To obtain a dataset where the number of patients per trial is balanced for
# > all trials, the number of patients per trial was rounded to 21 to generate
# > the dataset. Data.Observed.MTS thus contains a total of 2079 patients rather
# > than the requested 2000 in the function call.
```

Sim.Data.STS *Simulates a dataset that can be used to assess surrogacy in the single-trial setting*

Description

The function Sim.Data.STS simulates a dataset that contains the variables Treat, Surr, True, and Pat.ID. The user can specify the total number of patients, the desired R_{indiv} value (also referred to as the adjusted association (γ) in the single-trial meta-analytic setting), and the desired means of the surrogate and the true endpoints in the experimental and control treatment groups.

Usage

```
Sim.Data.STS(N.Total=2000, R.Indiv.Target=.8, Means=c(0, 0, 0, 0), Seed=
sample(1:1000, size=1))
```

Arguments

N.Total	The total number of patients in the simulated dataset. Default 2000.
R.Indiv.Target	The desired R_{indiv} (or γ) value in the simulated dataset. Default 0.80.
Means	A vector that specifies the desired mean for the surrogate in the control treatment group, mean for the surrogate in the experimental treatment group, mean for the true endpoint in the control treatment group, and mean for the true endpoint in the experimental treatment group, respectively. Default c(0, 0, 0, 0).
Seed	The seed that is used to generate the dataset. Default sample(x=1:1000, size=1), i.e., a random number between 1 and 1000.

Details

The generated object Data.Observed.STS (of class data.frame) is placed in the workspace (for easy access).

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

See Also

Sim.Data.MTS, Single.Trial.RE.AA

Examples

```
# Simulate a dataset:
Sim.Data.STS(N.Total=2000, R.Indiv.Target=.8, Means=c(1, 5, 20, 37), Seed=1)
```

Sim.Data.STSBinBin *Simulates a dataset that can be used to assess surrogacy in the single trial setting when S and T are binary endpoints*

Description

The function `Sim.Data.STSBinBin` simulates a dataset that contains four (binary) counterfactuals (i.e., potential outcomes) and a (binary) treatment indicator. The counterfactuals T_0 and T_1 denote the true endpoints of a patient under the control and the experimental treatments, respectively, and the counterfactuals S_0 and S_1 denote the surrogate endpoints of the patient under the control and the experimental treatments, respectively. In addition, the function provides the "observable" data based on the dataset of the counterfactuals, i.e., the S and T endpoints given the treatment that was allocated to a patient. The user can specify the assumption regarding monotonicity that should be made to generate the data (no monotonicity, monotonicity for S alone, monotonicity for T alone, or monotonicity for both S and T).

Usage

```
Sim.Data.STSBinBin(Monotonicity=c("No"), N.Total=2000, Seed)
```

Arguments

`Monotonicity` The assumption regarding monotonicity that should be made when the data are generated, i.e., `Monotonicity="No"` (no monotonicity assumed), `Monotonicity="True.Endp"` (monotonicity assumed for the true endpoint alone), `Monotonicity="Surr.Endp"` (monotonicity assumed for the surrogate endpoint alone), and `Monotonicity="Surr.True.Endp"` (monotonicity assumed for both endpoints). Default `Monotonicity="No"`.

`N.Total` The desired number of patients in the simulated dataset. Default 2000.

`Seed` A seed that is used to generate the dataset. Default `sample(x=1:1000, size=1)`, i.e., a random number between 1 and 1000.

Details

The generated objects `Data.STSBinBin_Counterfactuals` (which contains the counterfactuals) and `Data.STSBinBin_Obs` (which contains the observable data) of class `data.frame` are placed in the workspace. Other relevant output can be accessed based on the fitted object (see $Value$ below)

Value

An object of class `Sim.Data.STSBinBin` with components,

`Data.STSBinBin.Obs`
The generated dataset that contains the "observed" surrogate endrpoint, true endpoint, and assigned treatment.

`Data.STSBinBin.Counter`
The generated dataset that contains the counterfactuals.

`Vector_Pi` The vector of probabilities of the potential outcomes, i.e., $pi_{0000}, pi_{0100}, pi_{0010}, pi_{0001}, pi_{0101}, pi_{1000}, pi_{1010}, pi_{1001}, pi_{1110}, pi_{1101}, pi_{1011}, pi_{1111}, pi_{0110}, pi_{0011}, pi_{0111}, pi_{1100}$.

`Pi_Marginals` The vector of marginal probabilities $\pi_{1\cdot1\cdot}$, $\pi_{0\cdot1\cdot}$, $\pi_{1\cdot0\cdot}$, $\pi_{0\cdot0\cdot}$, $\pi_{\cdot1\cdot1}$, $\pi_{\cdot1\cdot0}$, $\pi_{\cdot0\cdot1}$, $\pi_{\cdot0\cdot0}$.

`True.R2_H` The true R^2_H value.

`True.Theta_T` The true odds ratio for T.

`True.Theta_S` The true odds ratio for S.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

Examples

```
## Generate a dataset with 2000 patients,
## assuming no monotonicity:
Sim.Data.STSBinBin(Monotonicity=c("No"), N.Total=200)
```

`Single.Trial.RE.AA` *Conducts a surrogacy analysis based on the single-trial meta-analytic framework*

Description

The function `Single.Trial.RE.AA` conducts a surrogacy analysis based on the single-trial meta-analytic framework of Buyse & Molenberghs (1998). See **Details** below.

Usage

```
Single.Trial.RE.AA(Dataset, Surr, True, Treat, Pat.ID, Alpha=.05,
Number.Bootstraps=500, Seed=sample(1:1000, size=1))
```

Arguments

`Dataset` A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, and a patient ID.

`Surr` The name of the variable in `Dataset` that contains the surrogate values.

`True` The name of the variable in `Dataset` that contains the true endpoint values.

`Treat` The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group. The $-1/1$ coding is recommended.

`Pat.ID` The name of the variable in `Dataset` that contains the patient's ID.

`Alpha` The α-level that is used to determine the confidence intervals around Alpha (which is a parameter estimate of a model where the surrogate is regressed on the treatment indicator, see **Details** below), Beta, RE, and γ. Default 0.05.

`Number.Bootstraps`
The number of bootstrap samples that are used to obtain the bootstrapp-based confidence intervals for RE and the adjusted association (γ). Default 500.

`Seed` The seed that is used to generate the bootstrap samples. Default `sample(x=1:1000, size=1)`, i.e., a random number between 1 and 1000.

Details

The Relative Effect (RE) and the adjusted association (γ) are based on the following bivariate regression model (when the surrogate and the true endpoints are continuous variables):

$$S_j = \mu_S + \alpha Z_j + \varepsilon_{Sj},$$

$$T_j = \mu_T + \beta Z_j + \varepsilon_{Tj},$$

where the error terms have a joint zero-mean normal distribution with variance-covariance matrix:

$$\mathbf{\Sigma} = \left(\begin{array}{cc} \sigma_{SS} & \\ \sigma_{ST} & \sigma_{TT} \end{array} \right),$$

and where j is the subject indicator, S_j and T_j are the surrogate and true endpoint values of patient j, and Z_j is the treatment indicator for patient j.

The parameter estimates of the fitted regression model and the variance-covariance matrix of the residuals are used to compute RE and the adjusted association (γ), respectively:

$$RE = \frac{\beta}{\alpha},$$

$$\gamma = \frac{\sigma_{ST}}{\sqrt{\sigma_{SS}\sigma_{TT}}}.$$

Note

The single-trial meta-analytic framework is hampered by a number of issues (Burzykowski et al., 2005). For example, a key motivation to validate a surrogate endpoint is to be able to predict the effect of Z on T as based on the effect of Z on S in a new clinical trial where T is not (yet) observed. The RE allows for such a prediction, but this requires the assumption that the relation between α and β can be described by a linear regression model that goes through the origin. In other words, it has to be assumed that the RE remains constant across clinical trials. The constant RE assumption is unverifiable in a single-trial setting, but a way out of this problem is to combine the information of multiple clinical trials and generalize the RE concept to a multiple-trial setting (as is done in the multiple-trial meta-analytic approach, see `UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`, and `BimixedContCont`).

Value

An object of class `Single.Trial.RE.AA` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Alpha` An object of class `data.frame` that contains the parameter estimate for α, its standard error, and its confidence interval. Note that `Alpha` is not to be confused with the `Alpha` argument in the function call, which specifies the α-level of the confidence intervals of the parameters.

`Beta` An object of class `data.frame` that contains the parameter estimate for β, its standard error, and its confidence interval.

`RE.Delta` An object of class `data.frame` that contains the estimated RE, its standard error, and its confidence interval (based on the Delta method).

`RE.Fieller` An object of class `data.frame` that contains the estimated RE, its standard error, and its confidence interval (based on Fieller's theorem).

`RE.Boot` An object of class `data.frame` that contains the estimated RE, its standard error, and its confidence interval (based on bootstrapping). Note that the occurence of outliers in the sample of bootstrapped RE values may lead to standard errors and/or confidence intervals that are not trustworthy. Such problems mainly occur when the parameter estimate for α is close to 0 (taking its standard error into account). To detect possible outliers, studentized deleted residuals are computed (by fitting an intercept-only model with the bootstrapped RE values as the outcome variable). Bootstrapped RE values with an absolute studentized residual larger than $t(1-\alpha/2n; n-2)$ are marked as outliers (where n = the number of bootstrapped RE values; Kutner et al., 2005). A warning is given when outliers are found, and the position of the outlier(s) in the bootstrap sample is identified. Inspection of the vector of bootstrapped RE values (see `RE.Boot.Samples` below) is recommended in this situation, and/or the use of the confidence intervals that are based on the Delta method or Fieller's theorem (rather than the bootstrap-based confidence interval).

`AA` An object of class `data.frame` that contains the adjusted association (i.e., γ), its standard error, and its confidence interval (based on the Fisher-Z transformation procedure).

`AA.Boot` An object of class `data.frame` that contains the adjusted association (i.e., γ), its standard error, and its confidence interval (based on a bootstrap procedure).

`RE.Boot.Samples`
A vector that contains the RE values that were generated during the bootstrap procedure.

`AA.Boot.Samples`
A vector that contains the adjusted association (i.e., γ) values that were generated during the bootstrap procedure.

`Cor.Endpoints` A `data.frame` that contains the correlations between the surrogate and the true endpoint in the control treatment group (i.e., ρ_{T0T1}) and in the experimental treatment group (i.e., ρ_{T1S1}), their standard errors and their confidence intervals.

`Residuals` A `data.frame` that contains the residuals for the surrogate and true endpoints that are obtained when the surrogate and the true endpoint are regressed on the treatment indicator.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., & Molenberghs, G. (1998). The validation of surrogate endpoints in randomized experiments. *Biometrics, 54,* 1014-1029.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2005). *Applied linear statistical models (5th ed.)*. New York: McGraw Hill.

See Also

`UnifixedContCont`, `BifixedContCont`, `UnimixedContCont`, `BimixedContCont`, `ICA.ContCont`

Examples

```
## Not run:  # time consuming code part
# Example 1, based on the ARMD data:
data(ARMD)

# Assess surrogacy based on the single-trial meta-analytic approach:
Sur <- Single.Trial.RE.AA(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Pat.ID=Id)

# Obtain a summary and plot of the results
summary(Sur)
plot(Sur)


# Example 2
# Conduct an analysis based on a simulated dataset with 2000 patients
# and Rindiv=.8
# Simulate the data:
Sim.Data.STS(N.Total=2000, R.Indiv.Target=.8, Seed=123)

# Assess surrogacy:
Sur2 <- Single.Trial.RE.AA(Dataset=Data.Observed.STS, Surr=Surr, True=True, Treat=Treat,
Pat.ID=Pat.ID)

# Show a summary and plots of results
summary(Sur2)
plot(Sur2)
```

```
## End(Not run)
```

SPF.BinBin *Evaluate the surrogate predictive function (SPF) in the binary-binary setting (sensitivity-analysis based approach)*

Description

Computes the surrogate predictive function (SPF) based on sensitivity-analyis, i.e., $r(i,j) = P(\Delta T = i|\Delta S = j)$, in the setting where both S and T are binary endpoints. For example, $r(-1,1)$ quantifies the probability that the treatment has a negative effect on the true endpoint ($\Delta T = -1$) given that it has a positive effect on the surrogate ($\Delta S = 1$). All quantities of interest are derived from the vectors of 'plausible values' for π (i.e., vectors π that are compatible with the observable data at hand). See **Details** below.

Usage

```
SPF.BinBin(x)
```

Arguments

`x`	A fitted object of class `ICA.BinBin`, `ICA.BinBin.Grid.Full`, or `ICA.BinBin.Grid.Sample`.

Details

All $r(i,j) = P(\Delta T = i|\Delta S = j)$ are derived from π (vector of potential outcomes). Denote by $\boldsymbol{Y}' = (T_0, T_1, S_0, S_1)$ the vector of potential outcomes. The vector $\boldsymbol{Y}$ can take 16 values and the set of parameters $\pi_{ijpq} = P(T_0 = i, T_1 = j, S_0 = p, S_1 = q)$ (with $i, j, p, q = 0/1$) fully characterizes its distribution.

Based on the data and assuming SUTVA, the marginal probabilites $\pi_{1 \cdot 1 \cdot}$, $\pi_{1 \cdot 0 \cdot}$, $\pi_{\cdot 1 \cdot 1}$, $\pi_{\cdot 1 \cdot 0}$, $\pi_{0 \cdot 1 \cdot}$, and $\pi_{\cdot 0 \cdot 1}$ can be computed (by hand or using the function `MarginalProbs`). Define the vector

$$\boldsymbol{b}' = (1, \pi_{1 \cdot 1 \cdot}, \pi_{1 \cdot 0 \cdot}, \pi_{\cdot 1 \cdot 1}, \pi_{\cdot 1 \cdot 0}, \pi_{0 \cdot 1 \cdot}, \pi_{\cdot 0 \cdot 1})$$

and $\boldsymbol{A}$ is a contrast matrix such that the identified restrictions can be written as a system of linear equation

$$\boldsymbol{A\pi} = \boldsymbol{b}.$$

The matrix $\boldsymbol{A}$ has rank 7 and can be partitioned as $\boldsymbol{A} = (\boldsymbol{A_r}|\boldsymbol{A_f})$, and similarly the vector $\boldsymbol{\pi}$ can be partitioned as $\boldsymbol{\pi}' = (\boldsymbol{\pi_r}'|\boldsymbol{\pi_f}')$ (where f refers to the submatrix/vector given by the 9 last columns/components of $\boldsymbol{A}/\boldsymbol{\pi}$). Using these partitions the previous system of linear equations can be rewritten as

$$\boldsymbol{A_r\pi_r} + \boldsymbol{A_f\pi_f} = \boldsymbol{b}.$$

The functions `ICA.BinBin`, `ICA.BinBin.Grid.Sample`, and `ICA.BinBin.Grid.Full` contain algorithms that generate plausible distributions for $\boldsymbol{Y}$ (for details, see the documentation of these functions). Based on the output of these functions, `SPF.BinBin` computes the surrogate predictive function.

Value

`r_1_1` The vector of values for $r(1, 1)$, i.e., $P(\Delta T = 1|\Delta S = 1)$.

`r_min1_1` The vector of values for $r(-1, 1)$.

`r_0_1` The vector of values for $r(0, 1)$.

`r_1_0` The vector of values for $r(1, 0)$.

`r_min1_0` The vector of values for $r(-1, 0)$.

`r_0_0` The vector of values for $r(0, 0)$.

`r_1_min1` The vector of values for $r(1, -1)$.

`r_min1_min1` The vector of values for $r(-1, -1)$.

`r_0_min1` The vector of values for $r(0, -1)$.

`Monotonicity` The assumption regarding monotonicity under which the result was obtained.

Author(s)

Wim Van der Elst, Paul Meyvisch, Ariel Alonso, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2015). Assessing a surrogate effect predictive value in a causal inference framework.

See Also

`ICA.BinBin`, `ICA.BinBin.Grid.Sample`, `ICA.BinBin.Grid.Full`, `plot.SPF.BinBin`

Examples

```
# Use ICA.BinBin.Grid.Sample to obtain plausible values for pi
ICA_BINBIN_Grid_Sample <- ICA.BinBin.Grid.Sample(pi1_1_=0.341, pi0_1_=0.119,
pi1_0_=0.254, pi_1_1=0.686, pi_1_0=0.088, pi_0_1=0.078, Seed=1,
Monotonicity=c("General"), M=2500)

# Obtain SPF
SPF <- SPF.BinBin(ICA_BINBIN_Grid_Sample)

# examine results
summary(SPF)
plot(SPF)
```

SPF.BinCont *Evaluate the surrogate predictive function (SPF) in the causal-inference single-trial setting in the binary-continuous case*

Description

The function SPF.BinCont computes the surrogate predictive function (SPF), i.e., the $P[\Delta T|\Delta S \in I_{ab}]$ in the single-trial setting within the causal-inference framework when the surrogate endpoint is continuous (normally distributed) and the true endpoint is a binary outcome. For details, see Alonso *et al.* (2024).

Usage

```
SPF.BinCont(x, a, b)
```

Arguments

x A fitted object of class ICA.BinCont.

a The lower interval a in $P[\Delta T|\Delta S \in I_{ab}]$.

b The upper interval b in $P[\Delta T|\Delta S \in I_{ab}]$.

Value

An object of class SPF.BinCont with important or relevant components:

a The lower interval a in $P[\Delta T|\Delta S \in I_{ab}]$.

b The upper interval b in $P[\Delta T|\Delta S \in I_{ab}]$.

r_min1_min1 The vector of $P[\Delta T = -1|\Delta S \in I_{(-\infty,a)}]$.

r_0_min1 The vector of $P[\Delta T = 0|\Delta S \in I_{(-\infty,a)}]$.

r_1_min1 The vector of $P[\Delta T = 1|\Delta S \in I_{(-\infty,a)}]$.

r_min1_0 The vector of $P[\Delta T = -1|\Delta S \in I_{(a,b)}]$.

r_0_0 The vector of $P[\Delta T = 0|\Delta S \in I_{(a,b)}]$.

r_1_0 The vector of $P[\Delta T = 1|\Delta S \in I_{(a,b)}]$.

r_min1_1 The vector of $P[\Delta T = -1|\Delta S \in I_{(b,\infty)}]$.

r_0_1 The vector of $P[\Delta T = 0|\Delta S \in I_{(b,\infty)}]$.

r_1_1 The vector of $P[\Delta T = 1|\Delta S \in I_{(b,\infty)}]$.

P_DT_0_DS_0 The vector of $P[\Delta T = 0|\Delta S = 0]$.

P_DT_psi_DS_max The vector of $P[\Delta T = \tilde{\psi}_{ab}(\Delta S)]$, where $\tilde{\psi}_{ab}(\Delta S) = argmax_i P[\Delta T = i|\Delta S \in (x,y)]$.

best.pred.min1 The vector of $\tilde{\psi}_{ab}(\Delta S) = argmax_i P[\Delta T = i|\Delta S \in (x,y)]$, where $(x,y) = (-\infty,a)$.

best.pred.0 The vector of $\tilde{\psi}_{ab}(\Delta S) = argmax_i P[\Delta T = i|\Delta S \in (x,y)]$, where $(x,y) = (a,b)$.

best.pred.1 The vector of $\tilde{\psi}_{ab}(\Delta S) = argmax_i P[\Delta T = i|\Delta S \in (x,y)]$, where $(x,y) = (b,\infty)$.

Author(s)

Fenny Ong, Wim Van der Elst, Ariel Alonso, and Geert Molenberghs

References

Alonso, A., Ong, F., Van der Elst, W., Molenberghs, G., & Callegaro, A. (2024). Assessing a continuous surrogate predictive value for a binary true endpoint based on causal inference and information theory in vaccine trial.

See Also

ICA.BinCont, ICA.BinCont.BS, plot.SPF.BinCont

Examples

```
## Not run: # Time consuming code part
data(Schizo)
fit.ica <- ICA.BinCont.BS(Dataset = Schizo, Surr = BPRS, True = PANSS_Bin, nb = 10,
Theta.S_0=c(-10,-5,5,10,10,10,10,10), Theta.S_1=c(-10,-5,5,10,10,10,10,10),
Treat=Treat, M=50, Seed=1)

fit.spf <- SPF.BinCont(fit.ica, a=-5, b=5)

summary(fit.spf)
plot(fit.spf)

## End(Not run)
```

---

summary.FederatedApproachStage2

Provides a summary of the surrogacy measures for an object fitted with the 'FederatedApproachStage2()' function.

Description

Provides a summary of the surrogacy measures for an object fitted with the 'FederatedApproachStage2()' function.

Usage

```
## S3 method for class 'FederatedApproachStage2'
summary(object, ...)
```

Arguments

`object`	An object of class 'FederatedApproachStage2' fitted with the 'FederatedApproachStage2()' function.
`...`	...

Value

The surrogacy measures with their 95% confidence intervals.

Examples

```
## Not run:
#As an example, the federated data analysis approach can be applied to the Schizo data set
data(Schizo)
Schizo <-  Schizo[order(Schizo$InvestId, Schizo$Id),]
#Create separate datasets for each investigator
Schizo_datasets <- list()

for (invest_id in 1:198) {
Schizo_datasets[[invest_id]] <- Schizo[Schizo$InvestId == invest_id, ]
assign(paste0("Schizo", invest_id), Schizo_datasets[[invest_id]])
}
#Fit the first stage model for each dataset separately
results_stage1 <- list()
invest_ids <- list()
i <- 1
for (invest_id in 1:198) {
  dataset <- Schizo_datasets[[invest_id]]

  skip_to_next <- FALSE
 tryCatch(FederatedApproachStage1(dataset, Surr=CGI, True=PANSS, Treat=Treat, Trial.ID = InvestId,
                                   Min.Treat.Size = 5, Alpha = 0.05),
                                   error = function(e) { skip_to_next <<- TRUE})
  #if the trial does not have the minimum required number, skip to the next
  if(skip_to_next) { next }

 results_stage1[[invest_id]] <- FederatedApproachStage1(dataset, Surr=CGI, True=PANSS, Treat=Treat,
                                                 Trial.ID = InvestId, Min.Treat.Size = 5,
                                                         Alpha = 0.05)
  assign(paste0("stage1_invest", invest_id), results_stage1[[invest_id]])
 invest_ids[[i]] <- invest_id #keep a list of ids with datasets with required number of patients
  i <- i+1
}

invest_ids <- unlist(invest_ids)
invest_ids

#Combine the results of the first stage models
for (invest_id in invest_ids) {
  dataset <- results_stage1[[invest_id]]$Results.Stage.1
  if (invest_id == invest_ids[1]) {
    all_results_stage1<- dataset
 } else {
    all_results_stage1 <- rbind(all_results_stage1,dataset)
  }
}

all_results_stage1 #that combines the results of the first stage models
```

```
R.list <- list()
i <- 1
for (invest_id in invest_ids) {
  R <- results_stage1[[invest_id]]$R.i
  R.list[[i]] <- as.matrix(R[1:4,1:4])
  i <- i+1
}

R.list #list that combines all the variance-covariance matrices of the fixed effects

fit <- FederatedApproachStage2(Dataset = all_results_stage1, Intercept.S = Intercept.S,
                               alpha = alpha, Intercept.T = Intercept.T, beta = beta,
                               sigma.SS = sigma.SS, sigma.ST = sigma.ST,
                               sigma.TT = sigma.TT, Obs.per.trial = n,
                               Trial.ID = Trial.ID, R.list = R.list)
summary(fit)

## End(Not run)
```

summary.MetaAnalyticSurvBin

Provides a summary of the surrogacy measures for an object fitted with the 'MetaAnalyticSurvBin()' function.

Description

Provides a summary of the surrogacy measures for an object fitted with the 'MetaAnalyticSurvBin()' function.

Usage

```
## S3 method for class 'MetaAnalyticSurvBin'
summary(object, ...)
```

Arguments

`object`	An object of class 'MetaAnalyticSurvBin' fitted with the 'MetaAnalyticSurvBin()' function.
`...`	...

Value

The surrogacy measures with their 95% confidence intervals.

Examples

```
## Not run:
data("colorectal")
fit_bin <- MetaAnalyticSurvBin(data = colorectal, true = surv, trueind = SURVIND,
                               surrog = responder, trt = TREAT, center = CENTER,
                               trial = TRIAL, patientid = patientid,
                               adjustment="unadjusted")
summary(fit)

## End(Not run)
```

---

summary.MetaAnalyticSurvCat

Provides a summary of the surrogacy measures for an object fitted with the 'MetaAnalyticSurvCat()' function.

Description

Provides a summary of the surrogacy measures for an object fitted with the 'MetaAnalyticSurvCat()' function.

Usage

```
## S3 method for class 'MetaAnalyticSurvCat'
summary(object, ...)
```

Arguments

object	An object of class 'MetaAnalyticSurvCat' fitted with the 'MetaAnalyticSurvCat()' function.
...	...

Value

The surrogacy measures with their 95% confidence intervals.

Examples

```
## Not run:
data("colorectal4")
fit <- MetaAnalyticSurvCat(data = colorectal4, true = truend, trueind = trueind, surrog = surrogend,
                       trt = treatn, center = center, trial = trialend, patientid = patid,
                           adjustment="unadjusted")
summary(fit)

## End(Not run)
```

summary.MetaAnalyticSurvCont

Provides a summary of the surrogacy measures for an object fitted with the 'MetaAnalyticSurvCont()' function.

Description

Provides a summary of the surrogacy measures for an object fitted with the 'MetaAnalyticSurvCont()' function.

Usage

```
## S3 method for class 'MetaAnalyticSurvCont'
summary(object, ...)
```

Arguments

`object`	An object of class 'MetaAnalyticSurvCont' fitted with the 'MetaAnalyticSurvCont()' function.
`...`	...

Value

The surrogacy measures with their 95% confidence intervals.

Examples

```
## Not run:
data("colorectal")
data("prostate")
fit <- MetaAnalyticSurvCont(data = prostate, true = SURVTIME, trueind = SURVIND, surrog = PSA,
trt = TREAT, center = TRIAL, trial = TRIAL, patientid = PATID,
copula = "Hougaard", adjustment = "weighted")
summary(fit)

## End(Not run)
```

summary.MetaAnalyticSurvSurv

Provides a summary of the surrogacy measures for an object fitted with the 'MetaAnalyticSurvSurv()' function.

Description

Provides a summary of the surrogacy measures for an object fitted with the 'MetaAnalyticSurvSurv()' function.

Usage

```
## S3 method for class 'MetaAnalyticSurvSurv'
summary(object, ...)
```

Arguments

`object`	An object of class 'MetaAnalyticSurvSurv' fitted with the 'MetaAnalyticSurvSurv()' function.
`...`	...

Value

The surrogacy measures with their 95% confidence intervals.

Examples

```
## Not run:
data("colorectal")
fit <- MetaAnalyticSurvSurv(data=Ovarian,true=Surv,trueind=SurvInd,surrog=Pfs,surrogind=PfsInd,
                             trt=Treat,center=Center,trial=Center,patientid=Patient,
                             copula="Plackett",adjustment="unadjusted")
summary(fit)

## End(Not run)
```

summary_level_bootstrap_ICA

Bootstrap based on the multivariate normal sampling distribution

Description

`summary_level_bootstrap_ICA()` performs a parametric type of bootstrap based on the estimated multivariate normal sampling distribution of the maximum likelihood estimator for the (observable) D-vine copula model parameters.

Usage

```
summary_level_bootstrap_ICA(
  fitted_model,
  copula_par_unid,
  copula_family2,
  rotation_par_unid,
  n_prec,
  B,
  measure = "ICA",
  mutinfo_estimator = NULL,
  ICA_estimator = NULL,
  composite = FALSE,
  seed,
  restr_time = +Inf,
  ncores = 1
)
```

Arguments

`fitted_model` Returned value from `fit_copula_OrdOrd()`, `fit_copula_OrdCont()`, or `fit_copula_ContCont()`. This object contains the estimated identifiable part of the joint distribution for the potential outcomes.

`copula_par_unid` Parameter vector for the sequence of *unidentifiable* bivariate copulas that define the D-vine copula. The elements of `copula_par` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`copula_family2` Copula family of the other bivariate copulas. For the possible options, see `loglik_copula_scale()`. The elements of `copula_family2` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`rotation_par_unid` Vector of rotation parameters for the sequence of *unidentifiable* bivariate copulas that define the D-vine copula. The elements of `rotation_par` correspond to $(c_{23}, c_{13;2}, c_{24;3}, c_{14;23})$.

`n_prec` Number of Monte Carlo samples for the computation of the mutual information.

`B` Number of bootstrap replications

`measure` Compute intervals for which measure of surrogacy? Defaults to `"ICA"`. See first column names of `sens_results` for other possibilities.

`mutinfo_estimator` Function that estimates the mutual information between the first two arguments which are numeric vectors. Defaults to `FNN::mutinfo()` with default arguments in the survival-survival setting. This argument is not used for non-survival-survival settings.

`ICA_estimator` Function that estimates the ICA between the first two arguments which are numeric vectors. Defaults to `NULL` which corresponds to using `estimate_ICA_ContCont()`, `estimate_ICA_OrdCont()`, or `estimate_ICA_OrdOrd()` (depending on the endpoint types). This argument is not used in the survival-survival setting.

`composite` (boolean) If `composite` is `TRUE`, then the surrogate endpoint is a composite of both a "pure" surrogate endpoint and the true endpoint, e.g., progression-free survival is the minimum of time-to-progression and time-to-death.

`seed` Seed for Monte Carlo sampling. This seed does not affect the global environment.

`restr_time` Restriction time for the potential outcomes. Defaults to `+Inf` which means no restriction. Otherwise, the sampled potential outcomes are replace by `pmin(S0, restr_time)` (and similarly for the other potential outcomes).

`ncores` Number of cores used in the sensitivity analysis. The computations are computationally heavy, and this option can speed things up considerably.

Details

Let $\hat{\boldsymbol{\beta}}$ be the estimated identifiable parameter vector, $\hat{\Sigma}$ the corresponding estimated covariance matrix, and $\boldsymbol{\nu}$ a fixed value for the sensitivity parameter. The bootstrap is then performed in the following steps

1. Resample the identifiable parameters from the estimated sampling distribution,
$$\hat{\boldsymbol{\beta}}^{(b)} \sim N(\hat{\boldsymbol{\beta}}, \hat{\Sigma}).$$
2. For each resampled parameter vector and the fixed sensitivty parameter, compute the ICA as $ICA(\hat{\boldsymbol{\beta}}^{(b)}, \boldsymbol{\nu})$.

Value

(numeric) Vector of bootstrap replications for the estimated ICA.

SurvSurv *Assess surrogacy for two survival endpoints based on information theory and a two-stage approach*

Description

The function `SurvSurv` implements the information-theoretic approach to estimate individual-level surrogacy (i.e., $R^2_{h.ind}$) and the two-stage approach to estimate trial-level surrogacy (R^2_{trial}, R^2_{ht}) when both endpoints are time-to-event variables (Alonso & Molenberghs, 2008). See the **Details** section below.

Usage

```
SurvSurv(Dataset, Surr, SurrCens, True, TrueCens, Treat,
Trial.ID, Weighted=TRUE, Alpha=.05)
```

Arguments

`Dataset`	A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value and censoring indicator, a true endpoint value and censoring indicator, a treatment indicator, and a trial ID.
`Surr`	The name of the variable in `Dataset` that contains the surrogate endpoint values.
`SurrCens`	The name of the variable in `Dataset` that contains the censoring indicator for the surrogate endpoint values (1 = event, 0 = censored).
`True`	The name of the variable in `Dataset` that contains the true endpoint values.
`TrueCens`	The name of the variable in `Dataset` that contains the censoring indicator for the true endpoint values (1 = event, 0 = censored).
`Treat`	The name of the variable in `Dataset` that contains the treatment indicators.

`Trial.ID`	The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.
`Weighted`	Logical. If `TRUE`, then a weighted regression analysis is conducted at stage 2 of the two-stage approach. If `FALSE`, then an unweighted regression analysis is conducted at stage 2 of the two-stage approach. See the **Details** section below. Default `TRUE`.
`Alpha`	The α-level that is used to determine the confidence intervals around R^2_{trial} and R_{trial}. Default 0.05.

Details

Individual-level surrogacy

Alonso & Molenbergs (2008) proposed to redefine the surrogate endpoint S as a time-dependent covariate $S(t)$, taking value 0 until the surrogate endpoint occurs and 1 thereafter. Furthermore, these author considered the models

$$\lambda[t \mid x_{ij}, \beta] = K_{ij}(t)\lambda_{0i}(t)exp(\beta x_{ij}),$$
$$\lambda[t \mid x_{ij}, s_{ij}, \beta, \phi] = K_{ij}(t)\lambda_{0i}(t)exp(\beta x_{ij} + \phi S_{ij}),$$

where $K_{ij}(t)$ is the risk function for patient j in trial i, x_{ij} is a p-dimensional vector of (possibly) time-dependent covariates, β is a p-dimensional vector of unknown coefficients, $\lambda_{0i}(t)$ is a trial-specific baseline hazard function, S_{ij} is a time-dependent covariate version of the surrogate endpoint, and ϕ its associated effect.

The mutual information between S and T is estimated as $I(T,S) = \frac{1}{n}G^2$, where n is the number of patients and G^2 is the log likelihood test comparing the previous two models. Individual-level surrogacy can then be estimated as

$$R^2_{h.ind} = 1 - exp\left(-\frac{1}{n}G^2\right).$$

O'Quigley and Flandre (2006) pointed out that the previous estimator depends upon the censoring mechanism, even when the censoring mechanism is non-informative. For low levels of censoring this may not be an issue of much concern but for high levels it could lead to biased results. To properly cope with the censoring mechanism in time-to-event outcomes, these authors proposed to estimate the mutual information as $I(T,S) = \frac{1}{k}G^2$, where k is the total number of events experienced. Individual-level surrogacy is then estimated as

$$R^2_{h.ind} = 1 - exp\left(-\frac{1}{k}G^2\right).$$

Trial-level surrogacy

A two-stage approach is used to estimate trial-level surrogacy, following a procedure proposed by Buyse et al. (2011). In stage 1, the following trial-specific Cox proportional hazard models are fitted:

$$S_{ij}(t) = S_{i0}(t)exp(\alpha_i Z_{ij}),$$

$$T_{ij}(t) = T_{i0}(t)exp(\beta_i Z_{ij}),$$

where $S_{i0}(t)$ and $T_{i0}(t)$ are the trial-specific baseline hazard functions, Z_{ij} is the treatment indicator for subject j in trial i, and α_i, β_i are the trial-specific treatment effects on S and T, respectively.

Next, the second stage of the analysis is conducted:

$$\widehat{\beta_i} = \lambda_0 + \lambda_1 \widehat{\alpha_i} + \varepsilon_i,$$

where the parameter estimates for β_i and α_i are based on the full model that was fitted in stage 1.

When the argument `Weighted=FALSE` is used in the function call, the model that is fitted in stage 2 is an unweighted linear regression model. When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), the information that is obtained in stage 1 is weighted according to the number of patients in a trial.

The classical coefficient of determination of the fitted stage 2 model provides an estimate of R^2_{trial}.

Value

An object of class `SurvSurv` with components,

`Results.Stage.1`
: The results of stage 1 of the two-stage model fitting approach: a `data.frame` that contains the trial-specific log hazard ratio estimates of the treatment effects for the surrogate and the true endpoints.

`Results.Stage.2`
: An object of class `lm` (linear model) that contains the parameter estimates of the regression model that is fitted in stage 2 of the analysis.

`R2.ht`
: A `data.frame` that contains the trial-level coefficient of determination (R^2_{ht}), its standard error and confidence interval.

`R2.hind`
: A `data.frame` that contains the individual-level coefficient of determination (R^2_{hind}), its standard error and confidence interval.

`R2h.ind.QF`
: A `data.frame` that contains the individual-level coefficient of determination using the correction proposed by O'Quigley and Flandre (2006), its standard error and confidence interval.

`R2.hInd.By.Trial.QF`
: A `data.frame` that contains individual-level surrogacy estimates using the correction proposed by O'Quigley and Flandre (2006), (cluster-based estimates) and their confidence interval for each of the trials seperately.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Alonso, A. A., & Molenberghs, G. (2008). Evaluating time-to-cancer recurrence as a surrogate marker for survival from an information theory perspective. *Statistical Methods in Medical Research, 17,* 497-504.

Buyse, M., Michiels, S., Squifflet, P., Lucchesi, K. J., Hellstrand, K., Brune, M. L., Castaigne, S., Rowe, J. M. (2011). Leukemia-free survival as a surrogate end point for overall survival in the evaluation of maintenance therapy for patients with acute myeloid leukemia in complete remission. *Haematologica, 96,* 1106-1112.

O'Quigly, J., & Flandre, P. (2006). Quantification of the Prentice criteria for surrogate endpoints. *Biometrics, 62,* 297-300.

See Also

plot.SurvSurv

Examples

```
# Open Ovarian dataset
data(Ovarian)

# Conduct analysis
Fit <- SurvSurv(Dataset = Ovarian, Surr = Pfs, SurrCens = PfsInd,
True = Surv, TrueCens = SurvInd, Treat = Treat,
Trial.ID = Center)

# Examine results
plot(Fit)
summary(Fit)
```

Test.Mono *Test whether the data are compatible with monotonicity for S and/or T (binary endpoints)*

Description

For some situations, the observable marginal probabilities contain sufficient information to exclude a particular monotonicity scenario. For example, under monotonicity for S and T, one of the restrictions that the data impose is $\pi_{0111} < min(\pi_{0 \cdot 1 \cdot}, \pi_{\cdot 1 \cdot 1})$. If the latter condition does not hold in the dataset at hand, monotonicity for S and T can be excluded.

Usage

```
Test.Mono(pi1_1_, pi0_1_, pi1_0_, pi_1_1, pi_1_0, pi_0_1)
```

Arguments

`pi1_1_`	A scalar that contains $P(T = 1, S = 1 \vert Z = 0)$.
`pi0_1_`	A scalar that contains $P(T = 0, S = 1 \vert Z = 0)$.
`pi1_0_`	A scalar that contains $P(T = 1, S = 0 \vert Z = 0)$.
`pi_1_1`	A scalar that contains $P(T = 1, S = 1 \vert Z = 1)$.
`pi_1_0`	A scalar that contains $P(T = 1, S = 0 \vert Z = 1)$.
`pi_0_1`	A scalar that contains $P(T = 0, S = 1 \vert Z = 1)$.

Author(s)

Wim Van der Elst, Ariel Alonso, Marc Buyse, & Geert Molenberghs

References

Alonso, A., Van der Elst, W., & Molenberghs, G. (2015). Validation of surrogate endpoints: the binary-binary setting from a causal inference perspective.

Examples

```
Test.Mono(pi1_1_=0.2619048, pi1_0_=0.2857143, pi_1_1=0.6372549,
pi_1_0=0.07843137, pi0_1_=0.1349206, pi_0_1=0.127451)
```

TrialLevelIT *Estimates trial-level surrogacy in the information-theoretic framework*

Description

The function TrialLevelIT estimates trial-level surrogacy based on the vectors of treatment effects on S (i.e., α_i), intercepts on S (i.e., μ_i) and T (i.e., β_i) in the different trials. See the **Details** section below.

Usage

```
TrialLevelIT(Alpha.Vector, Mu_S.Vector=NULL,
Beta.Vector, N.Trial, Model="Reduced", Alpha=.05)
```

Arguments

Alpha.Vector	The vector of treatment effects on S in the different trials, i.e., α_i.
Mu_S.Vector	The vector of intercepts for S in the different trials, i.e., μ_{Si}. Only required when a full model is requested.
Beta.Vector	The vector of treatment effects on T in the different trials, i.e., β_i.
N.Trial	The total number of available trials.
Model	The type of model that should be fitted, i.e., Model=c("Full") or Model=c("Reduced"). See the **Details** section below. Default Model=c("Reduced").
Alpha	The α-level that is used to determine the confidence intervals around R^2_{trial} and R_{trial}. Default 0.05.

Details

When a full model is requested (by using the argument `Model=c("Full")` in the function call), trial-level surrogacy is assessed by fitting the following univariate model:

$$\beta_i = \lambda_0 + \lambda_1 \mu_{Si} + \lambda_2 \alpha_i + \varepsilon_i, (1)$$

where β_i = the trial-specific treatment effects on T, μ_{Si} = the trial-specific intercepts for S, and α_i = the trial-specific treatment effects on S. The -2 log likelihood value of model (1) (L_1) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\beta_i = \lambda_3$; L_0), and R^2_{ht} is computed based based on the Variance Reduction Factor (for details, see Alonso & Molenberghs, 2007):

$$R^2_{ht} = 1 - exp\left(-\frac{L_1 - L_0}{N}\right),$$

where N is the number of trials.

When a reduced model is requested (by using the argument `Model=c("Reduced")` in the function call), the following model is fitted:

$$\beta_i = \lambda_0 + \lambda_1 \alpha_i + \varepsilon_i.$$

The -2 log likelihood value of this model (L_1 for the reduced model) is subsequently compared to the -2 log likelihood value of an intercept-only model ($\beta_i = \lambda_3$; L_0), and R^2_{ht} is computed based on the reduction in the likelihood (as described above).

Value

An object of class `TrialLevelIT` with components,

`Alpha.Vector` The vector of treatment effects on S in the different trials.

`Beta.Vector` The vector of treatment effects on T in the different trials.

`N.Trial` The total number of trials.

`R2.ht` A `data.frame` that contains the trial-level coefficient of determination (R^2_{ht}), its standard error and confidence interval.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

See Also

UnimixedContCont, UnifixedContCont, BifixedContCont, BimixedContCont, plot.TrialLevelIT

Examples

```
# Generate vector treatment effects on S
set.seed(seed = 1)
Alpha.Vector <- seq(from = 5, to = 10, by=.1) + runif(min = -.5, max = .5, n = 51)

# Generate vector treatment effects on T
set.seed(seed=2)
Beta.Vector <- (Alpha.Vector * 3) + runif(min = -5, max = 5, n = 51)

# Apply the function to estimate R^2_{h.t}
Fit <- TrialLevelIT(Alpha.Vector=Alpha.Vector,
Beta.Vector=Beta.Vector, N.Trial=50, Model="Reduced")

summary(Fit)
plot(Fit)
```

TrialLevelMA *Estimates trial-level surrogacy in the meta-analytic framework*

Description

The function `TrialLevelMA` estimates trial-level surrogacy based on the vectors of treatment effects on S (i.e., α_i) and T (i.e., β_i) in the different trials. In particular, β_i is regressed on α_i and the classical coefficient of determination of the fitted model provides an estimate of R^2_{trial}. In addition, the standard error and CI are provided.

Usage

```
TrialLevelMA(Alpha.Vector, Beta.Vector,
N.Vector, Weighted=TRUE, Alpha=.05)
```

Arguments

`Alpha.Vector`	The vector of treatment effects on S in the different trials, i.e., α_i.
`Beta.Vector`	The vector of treatment effects on T in the different trials, i.e., β_i.
`N.Vector`	The vector of trial sizes N_i.
`Weighted`	Logical. If `TRUE`, then a weighted regression analysis is conducted. If `FALSE`, then an unweighted regression analysis is conducted. Default `TRUE`.
`Alpha`	The α-level that is used to determine the confidence intervals around R^2_{trial} and R_{trial}. Default 0.05.

Value

An object of class TrialLevelMA with components,

Alpha.Vector	The vector of treatment effects on S in the different trials.
Beta.Vector	The vector of treatment effects on T in the different trials.
N.Vector	The vector of trial sizes N_i.
Trial.R2	A data.frame that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval.
Trial.R	A data.frame that contains the trial-level correlation coefficient (R_{trial}), its standard error and confidence interval.
Model.2.Fit	The fitted stage 2 model.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

See Also

UnimixedContCont, UnifixedContCont, BifixedContCont, BimixedContCont, plot Meta-Analytic

Examples

```
# Generate vector treatment effects on S
set.seed(seed = 1)
Alpha.Vector <- seq(from = 5, to = 10, by=.1) + runif(min = -.5, max = .5, n = 51)
# Generate vector treatment effects on T
set.seed(seed=2)
Beta.Vector <- (Alpha.Vector * 3) + runif(min = -5, max = 5, n = 51)
# Vector of sample sizes of the trials (here, all n_i=10)
N.Vector <- rep(10, times=51)

# Apply the function to estimate R^2_{trial}
Fit <- TrialLevelMA(Alpha.Vector=Alpha.Vector,
Beta.Vector=Beta.Vector, N.Vector=N.Vector)

# Plot the results and obtain summary
plot(Fit)
summary(Fit)
```

`TwoStageSurvSurv` *Assess trial-level surrogacy for two survival endpoints using a two-stage approach*

Description

The function `TwoStageSurvSurv` uses a two-stage approach to estimate R^2_{trial}. In stage 1, trial-specific Cox proportional hazard models are fitted and in stage 2 the trial-specific estimated treatment effects on T are regressed on the trial-specific estimated treatment effects on S (measured on the log hazard ratio scale). The user can specify whether a weighted or unweighted model should be fitted at stage 2. See the **Details** section below.

Usage

```
TwoStageSurvSurv(Dataset, Surr, SurrCens, True, TrueCens, Treat,
Trial.ID, Weighted=TRUE, Alpha=.05)
```

Arguments

`Dataset` A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value and censoring indicator, a true endpoint value and censoring indicator, a treatment indicator, and a trial ID.

`Surr` The name of the variable in `Dataset` that contains the surrogate endpoint values.

`SurrCens` The name of the variable in `Dataset` that contains the censoring indicator for the surrogate endpoint values (1 = event, 0 = censored).

`True` The name of the variable in `Dataset` that contains the true endpoint values.

`TrueCens` The name of the variable in `Dataset` that contains the censoring indicator for the true endpoint values (1 = event, 0 = censored).

`Treat` The name of the variable in `Dataset` that contains the treatment indicators.

`Trial.ID` The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.

`Weighted` Logical. If `TRUE`, then a weighted regression analysis is conducted at stage 2 of the two-stage approach. If `FALSE`, then an unweighted regression analysis is conducted at stage 2 of the two-stage approach. See the **Details** section below. Default `TRUE`.

`Alpha` The α-level that is used to determine the confidence intervals around R^2_{trial} and R_{trial}. Default 0.05.

Details

A two-stage approach is used to estimate trial-level surrogacy, following a procedure proposed by Buyse et al. (2011). In stage 1, the following trial-specific Cox proportional hazard models are fitted:

$$S_{ij}(t) = S_{i0}(t)exp(\alpha_i Z_{ij}),$$

$$T_{ij}(t) = T_{i0}(t)exp(\beta_i Z_{ij}),$$

where $S_{i0}(t)$ and $T_{i0}(t)$ are the trial-specific baseline hazard functions, Z_{ij} is the treatment indicator for subject j in trial i, μ_{Si}, and α_i and β_i are the trial-specific treatment effects on S and T, respectively.

Next, the second stage of the analysis is conducted:

$$\widehat{\beta_i} = \lambda_0 + \lambda_1 \widehat{\alpha_i} + \varepsilon_i,$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on the full model that was fitted in stage 1.

When the argument `Weighted=FALSE` is used in the function call, the model that is fitted in stage 2 is an unweighted linear regression model. When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), the information that is obtained in stage 1 is weighted according to the number of patients in a trial.

The classical coefficient of determination of the fitted stage 2 model provides an estimate of R^2_{trial}.

Value

An object of class `TwoStageSurvSurv` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of trials that do not have at least three patients per treatment arm are excluded due to estimation constraints (Burzykowski et al., 2001). `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Results.Stage.1` The results of stage 1 of the two-stage model fitting approach: a `data.frame` that contains the trial-specific log hazard ratio estimates of the treatment effects for the surrogate and the true endpoints.

`Results.Stage.2` An object of class `lm` (linear model) that contains the parameter estimates of the regression model that is fitted in stage 2 of the analysis.

`Trial.R2` A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval.

`Trial.R` A `data.frame` that contains the trial-level correlation coefficient (R_{trial}), its standard error and confidence interval.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., Buyse, M., Geys, H., & Renard, D. (2001). Validation of surrogate endpoints in multiple randomized clinical trials with failure-time endpoints. *Applied Statistics, 50,* 405-422.

Buyse, M., Michiels, S., Squifflet, P., Lucchesi, K. J., Hellstrand, K., Brune, M. L., Castaigne, S., Rowe, J. M. (2011). Leukemia-free survival as a surrogate end point for overall survival in the

evaluation of maintenance therapy for patients with acute myeloid leukemia in complete remission. *Haematologica, 96,* 1106-1112.

See Also

plot.TwoStageSurvSurv

Examples

```
# Open Ovarian dataset
data(Ovarian)

# Conduct analysis
Results <- TwoStageSurvSurv(Dataset = Ovarian, Surr = Pfs, SurrCens = PfsInd,
True = Surv, TrueCens = SurvInd, Treat = Treat, Trial.ID = Center)

# Examine results of analysis
summary(Results)
plot(Results)
```

---

twostep_BinCont — *Fit binary-continuous copula submodel with two-step estimator*

Description

The `twostep_BinCont()` function fits the copula (sub)model fir a continuous surrogate and binary true endpoint with a two-step estimator. In the first step, the marginal distribution parameters are estimated through maximum likelihood. In the second step, the copula parameter is estimate while holding the marginal distribution parameters fixed.

Usage

```
twostep_BinCont(
  X,
  Y,
  copula_family,
  marginal_surrogate,
  marginal_surrogate_estimator = NULL,
  method = "BFGS"
)
```

Arguments

`X` (numeric) Continuous surrogate variable

`Y` (integer) Binary true endpoint variable ($T_k \in \{0, 1\}$)

`copula_family` Copula family, one of the following:

- `"clayton"`

- "frank"
- "gumbel"
- "gaussian"

marginal_surrogate
: Marginal distribution for the surrogate. For all available options, see ?Surrogate::cdf_fun.

marginal_surrogate_estimator
: Not yet implemented

method
: Optimization algorithm for maximizing the objective function. For all options, see ?maxLik::maxLik. Defaults to "BFGRS".

Value

A list with three elements:

- ml_fit: object of class maxLik::maxLik that contains the estimated copula model.
- marginal_S_dist: object of class fitdistrplus::fitdist that represents the marginal surrogate distribution.
- copula_family: string that indicates the copula family

---

twostep_SurvSurv *Fit survival-survival copula submodel with two-step estimator*

Description

The twostep_SurvSurv() function fits the copula (sub)model for a time-to-event surrogate and true endpoint with a two-step estimator. In the first step, the marginal distribution parameters are estimated through maximum likelihood. In the second step, the copula parameter is estimate while holding the marginal distribution parameters fixed.

Usage

```
twostep_SurvSurv(
  X,
  delta_X,
  Y,
  delta_Y,
  copula_family,
  n_knots,
  method = "BFGS"
)
```

Arguments

`X` (numeric) Possibly right-censored time-to-surrogate event

`delta_X` (integer) Surrogate event indicator:

- `1L` if surrogate event ocurred.
- `0L` if censored.

`Y` (numeric) Possibly right-censored time-to-true endpoint event

`delta_Y` (integer) True endpoint event indicator:

- `1L` if true endpoint event ocurred.
- `0L` if censored.

`copula_family` Copula family, one of the following:

- `"clayton"`
- `"frank"`
- `"gumbel"`
- `"gaussian"`

`n_knots` Number of internal knots for the Royston-Parmar survival models for $\tilde{S}_0$, T_0, $\tilde{S}_1$, and T_1. If `length(n_knots) == 1`, the same number of knots are assumed for the four marginal distributions.

`method` Optimization algorithm for maximizing the objective function. For all options, see `?maxLik::maxLik`. Defaults to `"BFGS"`.

Value

A list with three elements:

- ml_fit: object of class `maxLik::maxLik` that contains the estimated copula model.
- marginal_S_dist: object of class `fitdistrplus::fitdist` that represents the marginal surrogate distribution.
- copula_family: string that indicates the copula family

`UnifixedContCont` *Fits univariate fixed-effect models to assess surrogacy in the meta-analytic multiple-trial setting (continuous-continuous case)*

Description

The function `UnifixedContCont` uses the univariate fixed-effects approach to estimate trial- and individual-level surrogacy when the data of multiple clinical trials are available. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below. Further, the Individual Causal Association (ICA) is computed.

Usage

```
UnifixedContCont(Dataset, Surr, True, Treat, Trial.ID, Pat.ID, Model=c("Full"),
Weighted=TRUE, Min.Trial.Size=2, Alpha=.05, Number.Bootstraps=500,
Seed=sample(1:1000, size=1), T0T1=seq(-1, 1, by=.2), T0S1=seq(-1, 1, by=.2),
T1S0=seq(-1, 1, by=.2), S0S1=seq(-1, 1, by=.2))
```

Arguments

`Dataset` A `data.frame` that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.

`Surr` The name of the variable in `Dataset` that contains the surrogate endpoint values.

`True` The name of the variable in `Dataset` that contains the true endpoint values.

`Treat` The name of the variable in `Dataset` that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.

`Trial.ID` The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.

`Pat.ID` The name of the variable in `Dataset` that contains the patient's ID.

`Model` The type of model that should be fitted, i.e., `Model=c("Full")`, `Model=c("Reduced")`, or `Model=c("SemiReduced")`. See the **Details** section below. Default `Model=c("Full")`.

`Weighted` Logical. If `TRUE`, then a weighted regression analysis is conducted at stage 2 of the two-stage approach. If `FALSE`, then an unweighted regression analysis is conducted at stage 2 of the two-stage approach. See the **Details** section below. Default `TRUE`.

`Min.Trial.Size` The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R^2_{trial}, R_{trial}, R^2_{indiv}, and R_{indiv}. Default 0.05.

`Number.Bootstraps`
The standard errors and confidence intervals for R^2_{indiv} and R_{indiv} are determined as based on a bootstrap procedure. `Number.Bootstraps` specifies the number of bootstrap samples that are used. Default 500.

`Seed` The seed to be used in the bootstrap procedure. Default $sample(1 : 1000, size = 1)$.

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ_{Δ} (ICA). For details, see function `ICA.ContCont`. Default `seq(-1, 1, by=.2)`.

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.2)`.

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.2)`.

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.2)`.

Details

When the full bivariate mixed-effects model is fitted to assess surrogacy in the meta-analytic framework (for details, Buyse & Molenberghs, 2000), computational issues often occur. In that situation, the use of simplified model-fitting strategies may be warranted (for details, see see Burzykowski et al., 2005; Tibaldi et al., 2003).

The function `UnifixedContCont` implements one such strategy, i.e., it uses a two-stage univariate fixed-effects modelling approach to assess surrogacy. In the first stage of the analysis, two univariate linear regression models are fitted to the data of each of the i trials. When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_{Si} + \alpha_i Z_{ij} + \varepsilon_{Sij},$$
$$T_{ij} = \mu_{Ti} + \beta_i Z_{ij} + \varepsilon_{Tij},$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_{Si} and μ_{Ti} are the fixed trial-specific intercepts for S and T, and α_i and β_i are the fixed trial-specific treatment effects on S and T, respectively. The error terms ε_{Sij} and ε_{Tij} are assumed to be independent.

When a reduced model is requested by the user (by using the argument `Model=c("Reduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_S + \alpha_i Z_{ij} + \varepsilon_{Sij},$$
$$T_{ij} = \mu_T + \beta_i Z_{ij} + \varepsilon_{Tij},$$

where μ_S and μ_T are the common intercepts for S and T (i.e., it is assumed that the intercepts for the surrogate and the true endpoints are identical in each of the trials). The other parameters are the same as defined above, and ε_{Sij} and ε_{Tij} are again assumed to be independent.

An estimate of R^2_{indiv} is provided by $r(\varepsilon_{Sij}, \varepsilon_{Tij})^2$.

Next, the second stage of the analysis is conducted. When a full model is requested (by using the argument `Model=c("Full")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\mu_{Si}} + \lambda_2 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on the full models that were fitted in stage 1.

When a semi-reduced or reduced model is requested (by using the argument `Model=c("SemiReduced")` or `Model=c("Reduced")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\alpha}_i + \varepsilon_i.$$

where the parameter estimates for β_i and α_i are based on the semi-reduced or reduced models that were fitted in stage 1.

When the argument `Weighted=FALSE` is used in the function call, the model that is fitted in stage 2 is an unweighted linear regression model. When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), the information that is obtained in stage 1 is weighted according to the number of patients in a trial.

The classical coefficient of determination of the fitted stage 2 model provides an estimate of R^2_{trial}.

Value

An object of class `UnifixedContCont` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Results.Stage.1`
The results of stage 1 of the two-stage model fitting approach: a `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`Residuals.Stage.1`
A `data.frame` that contains the residuals for the surrogate and true endpoints that are obtained in stage 1 of the analysis (ε_{Sij} and ε_{Tij}).

`Results.Stage.2`
An object of class `lm` (linear model) that contains the parameter estimates of the regression model that is fitted in stage 2 of the analysis.

`Trial.R2` A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval.

`Indiv.R2` A `data.frame` that contains the individual-level coefficient of determination (R^2_{indiv}), its standard error and confidence interval.

`Trial.R` A `data.frame` that contains the trial-level correlation coefficient (R_{trial}), its standard error and confidence interval.

`Indiv.R` A `data.frame` that contains the individual-level correlation coefficient (R_{indiv}), its standard error and confidence interval.

`Cor.Endpoints` A `data.frame` that contains the correlations between the surrogate and the true endpoint in the control treatment group (i.e., ρ_{T0S0}) and in the experimental treatment group (i.e., ρ_{T1S1}), their standard errors and their confidence intervals.

`D.Equiv` The variance-covariance matrix of the trial-specific intercept and treatment effects for the surrogate and true endpoints (when a full or semi-reduced model is fitted, i.e., when `Model=c("Full")` or `Model=c("SemiReduced")` is used in the function call), or the variance-covariance matrix of the trial-specific treatment effects for the surrogate and true endpoints (when a reduced model is fitted, i.e., when `Model=c("Reduced")` is used in the function call). The variance-covariance matrix `D.Equiv` is equivalent to the $\boldsymbol{D}$ matrix that would be obtained when a (full or reduced) bivariate mixed-effect approach is used; see function `BimixedContCont`).

`ICA` A fitted object of class `ICA.ContCont`.

`T0T0` The variance of the true endpoint in the control treatment condition.

`T1T1` The variance of the true endpoint in the experimental treatment condition.

`S0S0` The variance of the surrogate endpoint in the control treatment condition.

`S1S1` The variance of the surrogate endpoint in the experimental treatment condition.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

Tibaldi, F., Abrahantes, J. C., Molenberghs, G., Renard, D., Burzykowski, T., Buyse, M., Parmar, M., et al., (2003). Simplified hierarchical linear models for the evaluation of surrogate endpoints. *Journal of Statistical Computation and Simulation, 73,* 643-658.

See Also

`UnimixedContCont`, `BifixedContCont`, `BimixedContCont`, `plot Meta-Analytic`

Examples

```
## Not run:  #Time consuming (>5 sec) code parts
# Example 1, based on the ARMD data
data(ARMD)

# Fit a full univariate fixed-effects model with weighting according to the
# number of patients in stage 2 of the two stage approach to assess surrogacy:
Sur <- UnifixedContCont(Dataset=ARMD, Surr=Diff24, True=Diff52, Treat=Treat, Trial.ID=Center,
Pat.ID=Id, Model="Full", Weighted=TRUE)

# Obtain a summary and plot of the results
```

```
summary(Sur)
plot(Sur)

# Example 2
# Conduct an analysis based on a simulated dataset with 2000 patients, 100 trials,
# and Rindiv=Rtrial=.8
# Simulate the data:
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8, R.Indiv.Target=.8,
Seed=123, Model="Reduced")

# Fit a reduced univariate fixed-effects model without weighting to assess
# surrogacy:
Sur2 <- UnifixedContCont(Dataset=Data.Observed.MTS, Surr=Surr, True=True, Treat=Treat,
Trial.ID=Trial.ID, Pat.ID=Pat.ID, Model="Reduced", Weighted=FALSE)

# Show a summary and plots of results:
summary(Sur2)
plot(Sur2, Weighted=FALSE)
## End(Not run)
```

UnimixedContCont *Fits univariate mixed-effect models to assess surrogacy in the meta-analytic multiple-trial setting (continuous-continuous case)*

Description

The function UnimixedContCont uses the univariate mixed-effects approach to estimate trial- and individual-level surrogacy when the data of multiple clinical trials are available. The user can specify whether a (weighted or unweighted) full, semi-reduced, or reduced model should be fitted. See the **Details** section below. Further, the Individual Causal Association (ICA) is computed.

Usage

```
UnimixedContCont(Dataset, Surr, True, Treat, Trial.ID, Pat.ID, Model=c("Full"),
Weighted=TRUE, Min.Trial.Size=2, Alpha=.05, Number.Bootstraps=500,
Seed=sample(1:1000, size=1), T0T1=seq(-1, 1, by=.2), T0S1=seq(-1, 1, by=.2),
T1S0=seq(-1, 1, by=.2), S0S1=seq(-1, 1, by=.2), ...)
```

Arguments

Dataset	A data.frame that should consist of one line per patient. Each line contains (at least) a surrogate value, a true endpoint value, a treatment indicator, a patient ID, and a trial ID.
Surr	The name of the variable in Dataset that contains the surrogate endpoint values.
True	The name of the variable in Dataset that contains the true endpoint values.
Treat	The name of the variable in Dataset that contains the treatment indicators. The treatment indicator should either be coded as 1 for the experimental group and -1 for the control group, or as 1 for the experimental group and 0 for the control group.

`Trial.ID` The name of the variable in `Dataset` that contains the trial ID to which the patient belongs.

`Pat.ID` The name of the variable in `Dataset` that contains the patient's ID.

`Model` The type of model that should be fitted, i.e., `Model=c("Full")`, `Model=c("Reduced")`, or `Model=c("SemiReduced")`. See the **Details** section below. Default `Model=c("Full")`.

`Weighted` Logical. If `TRUE`, then a weighted regression analysis is conducted at stage 2 of the two-stage approach. If `FALSE`, then an unweighted regression analysis is conducted at stage 2 of the two-stage approach. See the **Details** section below. Default `TRUE`.

`Min.Trial.Size` The minimum number of patients that a trial should contain to be included in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded from the analysis. Default 2.

`Alpha` The α-level that is used to determine the confidence intervals around R^2_{trial}, R_{trial}, R^2_{indiv}, and R_{indiv}. Default 0.05.

`Number.Bootstraps`
The confidence intervals for R^2_{indiv} and R_{indiv} are determined as based on a bootstrap procedure. `Number.Bootstraps` specifies the number of bootstrap samples that are to be used. Default 500.

`Seed` The seed to be used in the bootstrap procedure. Default $sample(1:1000, size = 1)$.

`T0T1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and T1 that should be considered in the computation of ρ_{Δ} (ICA). For details, see function `ICA.ContCont`. Default `seq(-1, 1, by=.2)`.

`T0S1` A scalar or vector that contains the correlation(s) between the counterfactuals T0 and S1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.2)`.

`T1S0` A scalar or vector that contains the correlation(s) between the counterfactuals T1 and S0 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.2)`.

`S0S1` A scalar or vector that contains the correlation(s) between the counterfactuals S0 and S1 that should be considered in the computation of ρ_{Δ}. Default `seq(-1, 1, by=.2)`.

`...` Other arguments to be passed to the function `lmer` (of the R package `lme4`) that is used to fit the geralized linear mixed-effect models in the function `BimixedContCont`.

Details

When the full bivariate mixed-effects model is fitted to assess surrogacy in the meta-analytic framework (for details, Buyse & Molenberghs, 2000), computational issues often occur. In that situation, the use of simplified model-fitting strategies may be warranted (for details, see Burzykowski et al., 2005; Tibaldi et al., 2003).

The function `UnimixedContCont` implements one such strategy, i.e., it uses a two-stage univariate mixed-effects modelling approach to assess surrogacy. In the first stage of the analysis, two univariate mixed-effects models are fitted to the data. When a full or semi-reduced model is requested (by using the argument `Model=c("Full")` or `Model=c("SemiReduced")` in the function call), the following univariate models are fitted:

$$S_{ij} = \mu_S + m_{Si} + (\alpha + a_i)Z_{ij} + \varepsilon_{Sij},$$

$$T_{ij} = \mu_T + m_{Ti} + (\beta + b_i)Z_{ij} + \varepsilon_{Tij},$$

where i and j are the trial and subject indicators, S_{ij} and T_{ij} are the surrogate and true endpoint values of subject j in trial i, Z_{ij} is the treatment indicator for subject j in trial i, μ_S and μ_T are the fixed intercepts for S and T, m_{Si} and m_{Ti} are the corresponding random intercepts, α and β are the fixed treatment effects for S and T, and a_i and b_i are the corresponding random treatment effects, respectively. The error terms ε_{Sij} and ε_{Tij} are assumed to be independent.

When a reduced model is requested (by using the argument `Model=c("Reduced")` in the function call), the following two univariate models are fitted:

$$S_{ij} = \mu_S + (\alpha + a_i)Z_{ij} + \varepsilon_{Sij},$$

$$T_{ij} = \mu_T + (\beta + b_i)Z_{ij} + \varepsilon_{Tij},$$

where μ_S and μ_T are the common intercepts for S and T (i.e., it is assumed that the intercepts for the surrogate and the true endpoints are identical in each of the trials). The other parameters are the same as defined above, and ε_{Sij} and ε_{Tij} are again assumed to be independent.

An estimate of R^2_{indiv} is computed as $r(\varepsilon_{Sij}, \varepsilon_{Tij})^2$.

Next, the second stage of the analysis is conducted. When a full model is requested by the user (by using the argument `Model=c("Full")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\mu_{Si}} + \lambda_2 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameter estimates for β_i, μ_{Si}, and α_i are based on the models that were fitted in stage 1, i.e., $\beta_i = \beta + b_i$, $\mu_{Si} = \mu_S + m_{Si}$, and $\alpha_i = \alpha + a_i$.

When a reduced or semi-reduced model is requested by the user (by using the arguments `Model=c("SemiReduced")` or `Model=c("Reduced")` in the function call), the following model is fitted:

$$\widehat{\beta}_i = \lambda_0 + \lambda_1 \widehat{\alpha}_i + \varepsilon_i,$$

where the parameters are the same as defined above.

When the argument `Weighted=FALSE` is used in the function call, the model that is fitted in stage 2 is an unweighted linear regression model. When a weighted model is requested (using the argument `Weighted=TRUE` in the function call), the information that is obtained in stage 1 is weighted according to the number of patients in a trial.

The classical coefficient of determination of the fitted stage 2 model provides an estimate of R^2_{trial}.

Value

An object of class `UnimixedContCont` with components,

`Data.Analyze` Prior to conducting the surrogacy analysis, data of patients who have a missing value for the surrogate and/or the true endpoint are excluded. In addition, the data of trials (i) in which only one type of the treatment was administered, and (ii) in which either the surrogate or the true endpoint was a constant (i.e., all patients within a trial had the same surrogate and/or true endpoint value) are excluded. In addition, the user can specify the minimum number of patients that a trial should contain in order to include the trial in the analysis. If the number of patients in a trial is smaller than the value specified by `Min.Trial.Size`, the data of the trial are excluded. `Data.Analyze` is the dataset on which the surrogacy analysis was conducted.

`Obs.Per.Trial` A `data.frame` that contains the total number of patients per trial and the number of patients who were administered the control treatment and the experimental treatment in each of the trials (in `Data.Analyze`).

`Results.Stage.1` The results of stage 1 of the two-stage model fitting approach: a `data.frame` that contains the trial-specific intercepts and treatment effects for the surrogate and the true endpoints (when a full or semi-reduced model is requested), or the trial-specific treatment effects for the surrogate and the true endpoints (when a reduced model is requested).

`Residuals.Stage.1` A `data.frame` that contains the residuals for the surrogate and true endpoints that are obtained in stage 1 of the analysis (ε_{Sij} and ε_{Tij}).

`Fixed.Effect.Pars` A `data.frame` that contains the fixed intercept and treatment effects for S and T (i.e., μ_S, μ_T, α, and β) when a full, semi-reduced, or reduced model is fitted in stage 1.

`Random.Effect.Pars` A `data.frame` that contains the random intercept and treatment effects for S and T (i.e., m_{Si}, m_{Ti}, a_i and b_i) when a full or semi-reduced model is fitted in stage 1, or that contains the random treatment effects for S and T (i.e., a_i, and b_i) when a reduced model is fitted in stage 1.

`Results.Stage.2` An object of class `lm` (linear model) that contains the parameter estimates of the regression model that is fitted in stage 2 of the analysis.

`Trial.R2` A `data.frame` that contains the trial-level coefficient of determination (R^2_{trial}), its standard error and confidence interval.

`Indiv.R2` A `data.frame` that contains the individual-level coefficient of determination (R^2_{indiv}), its standard error and confidence interval.

`Trial.R` A `data.frame` that contains the trial-level correlation coefficient (R_{trial}), its standard error and confidence interval.

`Indiv.R` A `data.frame` that contains the individual-level correlation coefficient (R_{indiv}), its standard error and confidence interval.

`Cor.Endpoints` A `data.frame` that contains the correlations between the surrogate and the true endpoint in the control treatment group (i.e., ρ_{T0S0}) and in the experimental treatment group (i.e., ρ_{T1S1}), their standard errors and their confidence intervals.

`D.Equiv` The variance-covariance matrix of the trial-specific intercept and treatment effects for the surrogate and true endpoints (when a full or semi-reduced model is fitted, i.e., when `Model=c("Full")` or `Model=c("SemiReduced")` is used in the function call), or the variance-covariance matrix of the trial-specific treatment effects for the surrogate and true endpoints (when a reduced model is fitted, i.e., when `Model=c("Reduced")` is used in the function call). The variance-covariance matrix `D.Equiv` is equivalent to the $\boldsymbol{D}$ matrix that would be obtained when a (full or reduced) bivariate mixed-effects approach is used; see function `BimixedContCont`).

`ICA` A fitted object of class `ICA.ContCont`.

`T0T0` The variance of the true endpoint in the control treatment condition.

`T1T1` The variance of the true endpoint in the experimental treatment condition.

`S0S0` The variance of the surrogate endpoint in the control treatment condition.

`S1S1` The variance of the surrogate endpoint in the experimental treatment condition.

Author(s)

Wim Van der Elst, Ariel Alonso, & Geert Molenberghs

References

Burzykowski, T., Molenberghs, G., & Buyse, M. (2005). *The evaluation of surrogate endpoints*. New York: Springer-Verlag.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., & Geys, H. (2000). The validation of surrogate endpoints in meta-analysis of randomized experiments. *Biostatistics, 1,* 49-67.

Tibaldi, F., Abrahantes, J. C., Molenberghs, G., Renard, D., Burzykowski, T., Buyse, M., Parmar, M., et al., (2003). Simplified hierarchical linear models for the evaluation of surrogate endpoints. *Journal of Statistical Computation and Simulation, 73,* 643-658.

See Also

`UnifixedContCont`, `BifixedContCont`, `BimixedContCont`, `plot Meta-Analytic`

Examples

```
## Not run:  #Time consuming code part
# Conduct an analysis based on a simulated dataset with 2000 patients, 100 trials,
# and Rindiv=Rtrial=.8
# Simulate the data:
Sim.Data.MTS(N.Total=2000, N.Trial=100, R.Trial.Target=.8, R.Indiv.Target=.8,
Seed=123, Model="Reduced")

# Fit a reduced univariate mixed-effects model without weighting to assess surrogacy:
Sur <- UnimixedContCont(Dataset=Data.Observed.MTS, Surr=Surr, True=True, Treat=Treat,
Trial.ID=Trial.ID, Pat.ID=Pat.ID, Model="Reduced", Weighted=FALSE)
```

```
# Show a summary and plots of the results:
summary(Sur)
plot(Sur, Weighted=FALSE)
## End(Not run)
```

Index