

# Package ‘tern.gee’

July 22, 2025

**Type** Package

**Title** Tables and Graphs for Generalized Estimating Equations (GEE)  
Model Fits

**Version** 0.1.5

**Date** 2024-08-21

**Description** Generalized estimating equations (GEE) are a popular choice for analyzing longitudinal binary outcomes. This package provides an interface for fitting GEE, currently for logistic regression, within the 'tern' <<https://cran.r-project.org/package=tern>> framework (Zhu, Sabanés Bové et al., 2023) and tabulate results easily using 'rtables' <<https://cran.r-project.org/package=rtables>> (Becker, Waddell et al., 2023). It builds on 'geepack' <[doi:10.18637/jss.v015.i02](https://doi.org/10.18637/jss.v015.i02)> (Højsgaard, Halekoh and Yan, 2006) for the actual GEE model fitting.

**License** Apache License 2.0

**URL** <https://insightsengineering.github.io/tern.gee/>,  
<https://github.com/insightsengineering/tern.gee/>

**BugReports** <https://github.com/insightsengineering/tern.gee/issues>

**Depends** R (>= 4.0), tern (>= 0.9.5)

**Imports** checkmate (>= 2.1.0), emmeans (>= 1.10.4), formatters (>= 0.5.8), geeasy, geepack, nlme, rtables (>= 0.6.9), stats

**Suggests** dplyr (>= 1.0.9), knitr (>= 1.42), rmarkdown (>= 2.23), testthat (>= 3.1), vdiffR (>= 1.0.0), withr (>= 2.0.0)

**VignetteBuilder** knitr, rmarkdown

**Config/Needs/verdepcheck** insightsengineering/tern, mllg/checkmate, rvlenth/emmeans, insightsengineering/formatters, annennenne/geeasy, cran/geepack, cran/nlme, insightsengineering/rtables, tidyverse/dplyr, yihui/knitr, rstudio/rmarkdown, r-lib/testthat, r-lib/vdiffR, r-lib/withr

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** en-US  
**LazyData** true  
**RoxygenNote** 7.3.2  
**NeedsCompilation** no  
**Author** Daniel Sabanés Bové [aut],  
Joe Zhu [aut, cre],  
Emily de la Rua [aut],  
F. Hoffmann-La Roche AG [cph, fnd]  
**Maintainer** Joe Zhu <joe.zhu@roche.com>  
**Repository** CRAN  
**Date/Publication** 2024-08-23 16:30:02 UTC

Contents

|                            |   |
|----------------------------|---|
| tern.gee-package . . . . . | 2 |
| fev_data . . . . .         | 3 |
| fit_gee . . . . .          | 4 |
| gee_methods . . . . .      | 5 |
| lsmeans . . . . .          | 6 |
| tabulate_gee . . . . .     | 7 |
| vars_gee . . . . .         | 9 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>11</b> |
|--------------|-----------|

---

|                  |                         |
|------------------|-------------------------|
| tern.gee-package | tern.gee <i>Package</i> |
|------------------|-------------------------|

---

Description

Create tables and graphs for GEE model fits.

Author(s)

- Maintainer:** Joe Zhu <joe.zhu@roche.com>  
Authors:
- Daniel Sabanés Bové <daniel@rconis.com>
  - Emily de la Rua <emily.de\_la\_rua@contractors.roche.com>
- Other contributors:
- F. Hoffmann-La Roche AG [copyright holder, funder]

## See Also

Useful links:

- <https://insightsengineering.github.io/tern.gee/>
- <https://github.com/insightsengineering/tern.gee/>
- Report bugs at <https://github.com/insightsengineering/tern.gee/issues>

---

fev\_data

*Example dataset for tern.gee package.*

---

## Description

Measurements of FEV1 (forced expired volume in one second) is a measure of how quickly the lungs can be emptied. Low levels of FEV1 may indicate chronic obstructive pulmonary disease (COPD).

## Usage

fev\_data

## Format

A tibble with 800 rows and 7 variables:

- USUBJID: unique subject identifier.
- AVISIT: visit number.
- ARMCD: treatment, TRT or PBO.
- RACE: 3-category race.
- SEX: sex.
- FEV1\_BL: FEV1 at baseline (%).
- FEV1: FEV1 at study visits.

---

`fit_gee`*Fit a GEE Model*

---

## Description

Fit a GEE Model

## Usage

```
fit_gee(  
  vars = vars_gee(),  
  data,  
  regression = c("logistic"),  
  cor_struct = c("unstructured", "toeplitz", "compound symmetry", "auto-regressive")  
)
```

## Arguments

|                         |                                            |
|-------------------------|--------------------------------------------|
| <code>vars</code>       | (list)<br>see <a href="#">vars_gee()</a> . |
| <code>data</code>       | (data.frame)<br>input data.                |
| <code>regression</code> | (string)<br>choice of regression model.    |
| <code>cor_struct</code> | (string)<br>assumed correlation structure. |

## Details

The correlation structure can be:

- `unstructured`: No constraints are placed on the correlations.
- `toeplitz`: Assumes a banded correlation structure, i.e. the correlation between two time points depends on the distance between the time indices.
- `compound symmetry`: Constant correlation between all time points.
- `auto-regressive`: Auto-regressive order 1 correlation matrix.

## Value

Object of class `tern_gee` as well as specific to the kind of regression which was used.

## Examples

```
df <- fev_data
df$AVAL <- as.integer(fev_data$FEV1 > 30)

fit_gee(vars = vars_gee(arm = "ARMCD"), data = df)

fit_gee(vars = vars_gee(arm = "ARMCD"), data = df, cor_struct = "compound symmetry")
```

---

gee\_methods

*Methods for GEE Models*


---

## Description

Additional methods which can simplify working with the GEE result object.

## Usage

```
## S3 method for class 'tern_gee'
VarCorr(x, sigma = 1, ...)

## S3 method for class 'tern_gee'
QIC(object, ...)
```

## Arguments

|        |                                                                                                                                                            |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x      | (tern_gee)<br>result of <code>fit_gee()</code> .                                                                                                           |
| sigma  | an optional numeric value used as a multiplier for the standard deviations. The default is <code>x\$sigma</code> or 1 depending on <code>class(x)</code> . |
| ...    | further optional arguments passed to other methods (none for the methods documented here).                                                                 |
| object | (tern_gee)<br>result of <code>fit_gee()</code> .                                                                                                           |

## Value

`VarCorr()` returns the estimated covariance matrix, and `QIC()` returns the QIC value.

lsmeans

*Extract Least Square Means from a GEE Model***Description**

Extract Least Square Means from a GEE Model

**Usage**

```
lsmeans(
  object,
  conf_level = 0.95,
  weights = "proportional",
  specs = object$vars$arm,
  ...
)

## S3 method for class 'tern_gee_logistic'
lsmeans(
  object,
  conf_level = 0.95,
  weights = "proportional",
  specs = object$vars$arm,
  ...
)
```

**Arguments**

|            |                                                                                                                        |
|------------|------------------------------------------------------------------------------------------------------------------------|
| object     | (tern_gee)<br>result of <a href="#">fit_gee()</a> .                                                                    |
| conf_level | (proportion)<br>confidence level                                                                                       |
| weights    | (string)<br>type of weights to be used for the least square means, see <a href="#">emmeans::emmeans()</a> for details. |
| specs      | (string or formula) specifications passed to <a href="#">emmeans::emmeans()</a>                                        |
| ...        | additional arguments for methods                                                                                       |

**Value**

A data.frame with least-square means and contrasts. Additional classes allow to dispatch downstream methods correctly, too.

**Examples**

```
df <- fev_data
df$AVAL <- rbinom(n = nrow(df), size = 1, prob = 0.5)
fit <- fit_gee(vars = vars_gee(arm = "ARMCD"), data = df)

lsmeans(fit)

lsmeans(fit, conf_level = 0.90, weights = "equal")
```

---

|              |                                  |
|--------------|----------------------------------|
| tabulate_gee | <i>Tabulation of a GEE Model</i> |
|--------------|----------------------------------|

---

**Description**

Functions to produce tables from a fitted GEE produced with `fit_gee()`.

**Usage**

```
## S3 method for class 'tern_gee'
as.rtable(x, type = c("coef", "cov"), ...)

s_lsmeans_logistic(df, .in_ref_col)

a_lsmeans_logistic(df, .in_ref_col)

summarize_gee_logistic(
  lyt,
  ...,
  table_names = "lsmeans_logistic_summary",
  .stats = NULL,
  .formats = NULL,
  .indent_mods = NULL,
  .labels = NULL
)
```

**Arguments**

|                          |                                                                           |
|--------------------------|---------------------------------------------------------------------------|
| <code>x</code>           | (data.frame)<br>the object which should be converted to an rtable.        |
| <code>type</code>        | (character)<br>type of table to extract from tern_gee object.             |
| <code>...</code>         | additional arguments for methods.                                         |
| <code>df</code>          | (data.frame)<br>data set resulting from <code>lsmeans()</code> .          |
| <code>.in_ref_col</code> | (logical)<br>TRUE when working with the reference level, FALSE otherwise. |

|              |                                                                                                                               |
|--------------|-------------------------------------------------------------------------------------------------------------------------------|
| lyt          | (layout)<br>input layout where analyses will be added to.                                                                     |
| table_names  | (character)<br>this can be customized in case that the same vars are analyzed multiple times, to avoid warnings from rtables. |
| .stats       | (character)<br>statistics to select for the table.                                                                            |
| .formats     | (named character or list)<br>formats for the statistics.                                                                      |
| .indent_mods | (named integer)<br>indent modifiers for the labels.                                                                           |
| .labels      | (named character)<br>labels for the statistics (without indent).                                                              |

## Value

The functions have different purposes:

- `as.rtable()` returns either the coefficient table or the covariance matrix as an `rtables` object.
- `s_lsmeans_logistic()` returns several least square mean statistics from the GEE.
- `a_lsmeans_logistic()` is the formatted analysis function and returns the formatted statistics.
- `summarize_gee_logistic()` is the analyze function and returns the modified `rtables` layout.

## Functions

- `as.rtable(tern_gee)`: Extracts the coefficient table or covariance matrix estimate from a `tern_gee` object.
- `s_lsmeans_logistic()`: Statistics function which extracts estimates from a `lsmeans()` data frame based on a logistic GEE model.
- `a_lsmeans_logistic()`: Formatted Analysis function which can be further customized by calling `rtables::make_afun()` on it. It is used as `afun` in `rtables::analyze()`.
- `summarize_gee_logistic()`: Analyze function for tabulating least-squares means estimates from logistic GEE least square mean results.

## Examples

```
library(dplyr)

df <- fev_data %>%
  mutate(AVAL = as.integer(fev_data$FEV1 > 30))
df_counts <- df %>%
  select(USUBJID, ARMCD) %>%
  unique()

lsmeans_df <- lsmeans(fit_gee(vars = vars_gee(arm = "ARMCD"), data = df))
```

```

s_lsmeans_logistic(lsmeans_df[1, ], .in_ref_col = TRUE)

s_lsmeans_logistic(lsmeans_df[2, ], .in_ref_col = FALSE)
basic_table() %>%
  split_cols_by("ARMCD") %>%
  add_colcounts() %>%
  summarize_gee_logistic(
    .in_ref_col = FALSE
  ) %>%
  build_table(lsmeans_df, alt_counts_df = df_counts)

```

vars\_gee

*Set Variables to Use in GEE Model***Description**

Set Variables to Use in GEE Model

**Usage**

```

vars_gee(
  response = "AVAL",
  covariates = c(),
  id = "USUBJID",
  arm = "ARM",
  visit = "AVISIT"
)

```

**Arguments**

|            |                                                                   |
|------------|-------------------------------------------------------------------|
| response   | (character)<br>name of response variable.                         |
| covariates | (character)<br>vector of names of variables to use as covariates. |
| id         | (character)<br>name of variable to use to identify unique IDs.    |
| arm        | (character)<br>name of arm variable.                              |
| visit      | (character)<br>name of visit variable.                            |

**Value**

A list of variables that can be used as the vars argument in `fit_gee()`.

**Examples**

```
vars_gee()

vars_gee(
  response = "CHG",
  covariates = c("SEX", "RACE"),
  id = "SUBJID",
  arm = "ARMCD",
  visit = "AVISITN"
)
```

# Index

## \* datasets

fev\_data, 3

a\_lsmeans\_logistic (tabulate\_gee), 7

as.rtable.tern\_gee (tabulate\_gee), 7

class, 5

emmeans::emmeans(), 6

fev\_data, 3

fit\_gee, 4

fit\_gee(), 5–7, 9

gee\_methods, 5

lsmeans, 6

lsmeans(), 7, 8

QIC.tern\_gee (gee\_methods), 5

rtables::analyze(), 8

rtables::make\_afun(), 8

s\_lsmeans\_logistic (tabulate\_gee), 7

summarize\_gee\_logistic (tabulate\_gee), 7

tabulate\_gee, 7

tern.gee (tern.gee-package), 2

tern.gee-package, 2

VarCorr.tern\_gee (gee\_methods), 5

vars\_gee, 9

vars\_gee(), 4