Package 'samadb'

July 23, 2025

Title South Africa Macroeconomic Database API

Version 0.3.0

Description An R API providing access to a relational database with macroeconomic time series data for South Africa,
obtained from the South African Reserve Bank (SARB) and Statistics South Africa (STATSSA), and updated on a weekly basis
via the EconData <https://www.econdata.co.za/> platform and automated scraping of the SARB and STATSSA websites.
The database is maintained at the Department of Economics at Stellenbosch University.

BugReports https://github.com/Stellenbosch-Econometrics/SAMADB-Issues/issues

License GPL-3

Encoding UTF-8

Imports DBI, RMySQL, writexl, data.table, collapse (>= 2.0.0)

RoxygenNote 7.2.3

Depends R (>= 3.3.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Sebastian Krantz [aut, cre]

Maintainer Sebastian Krantz <sebastian.krantz@graduateinstitute.ch>

Repository CRAN

Date/Publication 2024-05-24 18:00:08 UTC

Contents

`samadb-package` *South Africa Macroeconomic Database API*

Description

An R API providing access to a relational database with public macroeconomic data for South Africa, obtained from from the South African Reserve Bank (SARB) and Statistics South Africa (STATSSA), and updated on a regular basis via the EconData (https://www.econdata.co.za/) platform and automated scraping of the SARB and STATSSA websites. The database is maintained at the Department of Economics at Stellenbosch University.

Functions

Functions and data providing information about the available data

`sm_datasources()`
`sm_datasets()`
`sm_series()`

Function to retrieve the data from the database

`sm_data()`

Functions to reshape data and add temporal identifiers

`sm_pivot_wider()`
`sm_pivot_longer()`
`sm_expand_date()`

Function to export wide format data to Excel

`sm_write_excel()`

Helper functions to convert inputs to R dates and transpose the data

`sm_as_date()`
`sm_transpose()`

Global Macros with core ID variables in the database

`.SAMADB_ID`
`.SAMADB_T`

.SAMADB_ID *Global Identifier Macros*

Description

The macro `.SAMADB_ID` contains the string `c("dsid", "series")` denoting the names of ID variables that identify the cross-sectional dimension in the database. All series codes are unique across datasets.

The macro `.SAMADB_T` contains the string `c("date", "year", "quarter", "month", "day")` denoting temporal identifiers generated by `sm_expand_date`. The `"date"` variable is sufficient to uniquely identify a point in time in the database.

Each value in the database is uniquely identified by dsid, series and date.

Usage

```
.SAMADB_ID
.SAMADB_T
```

See Also

`samadb`

Examples

```
.SAMADB_ID
.SAMADB_T
```

sm_as_date *Coerce Vectors to Dates*

Description

This function coerces date strings i.e. `"YYYY-MM-DD"` or `"YYYY-MM"`, years e.g. `2015` (numeric or character), year-quarters e.g. `"2015Q1"` or `"2015-Q1"`, year-months e.g. `"2015M01"` or `"2015-M01"` or numeric values representing dates (e.g. previously imported Excel date) to a regular R date.

Usage

```
sm_as_date(x, end = FALSE, origin = "1899-12-30")
```

Arguments

x	a character date string "YYYY-MM-DD" or "YYYY-MM", year-quarter "YYYYQN" or "YYYY-QN", year-month "YYYYMNN" or "YYYY-MNN" or calendar year YYYY (numeric or character), or a numeric value corresponding to a date that can be passed to as.Date.numeric.
end	logical. TRUE replaces missing time information with a period end-date which is the last day of the period. FALSE replaces missing month and day information with "-01", so the year date is the 1st of January, and for months / quarters the 1st day of the month / quarter.
origin	a date or date-string that can be used as reference for converting numeric values to dates. The default corresponds to dates generated in Excel for Windows. See as.Date.numeric.

Value

A Date vector.

See Also

sm_expand_date, samadb

Examples

```
sm_as_date("2011-05")
sm_as_date(2011)
sm_as_date("2011Q1")
sm_as_date("2011Q1", end = TRUE)
sm_as_date("2011M2")
sm_as_date("2011M2", end = TRUE)
```

---

sm_data *Retrieve Data from the Database*

Description

This is the main function of the package to retrieve data from the database.

Usage

```
sm_data(
  dsid = NULL,
  series = NULL,
  from = NULL,
  to = NULL,
  freq = NULL,
  labels = TRUE,
```

```
  wide = TRUE,
  expand.date = FALSE,
  ordered = TRUE,
  return.query = FALSE,
  ...
)
```

Arguments

dsid	character. (Optional) id's of datasets matching the 'dsid' column of the 'DATASET' table (retrieved using sm_datasets()). If used, all series from the dataset are returned, in addition to any other series selected with series.
series	character. (Optional) codes of series matching the 'series' column of the 'SERIES' table (retrieved using sm_series()). If 'dsid' is also specificed, the two are combined using SQL 'OR' i.e. these series are retrieved in addition to all series matched through 'dsid'.
from	set the start time of the data retrieved by either supplying a start date, a date-string of the form "YYYY-MM-DD" or "YYYY-MM", year-quarters of the form "YYYYQN" or "YYYY-QN", or a numeric year YYYY (numeric or character). These expressions are converted to a regular date by sm_as_date.
to	same as from: to set the time period until which data is retrieved. For expressions that are not full "YYYY-MM-DD" dates, the last day of the period is chosen.
freq	character. Return only series at a certain frequency. Allowed are values "D" (Daily), "W" (Weekly), "M" (Monthly), "Q" (Quarterly), "A" (Annual), "AF" (Fiscal Years), matching the 'freq' column in the 'SERIES' table (retrieved using sm_series()).
labels	logical. TRUE will also return labels (series descriptions) along with the series codes.
wide	logical. TRUE calls sm_pivot_wider on the result. FALSE returns the data in a long format without missing values (suitable for ggplot2).
expand.date	logical. TRUE will call sm_expand_date on the result.
ordered	logical. TRUE orders the result by 'date' and, if !is.null(dsid), labels = TRUE or !is.null(freq), by series, maintaining a fixed order of series. FALSE does not explicitly order the result, to the benefit of faster query execution.
return.query	logical. TRUE will not query the database but instead return the constructed SQL query as a character string (for debugging purposes).
...	further arguments passed to sm_pivot_wider (if wide = TRUE) or sm_expand_date (if expand.date = TRUE), no conflicts between these two.

Details

Series from datasets at different frequencies can be queried, but, if wide = TRUE, this will result in missing values in the lower frequency series.

Value

A data.table with the result of the query.

See Also

sm_pivot_wider, sm_expand_date, samadb

Examples

```
# Return all electricity indicators from 2000
sm_data("ELECTRICITY", from = 2000)
```

---

sm_datasets *Retrieve Datasets Table*

Description

This function pulls and return a table called 'DATASET' from the database.

Usage

```
sm_datasets(ordered = TRUE)
```

Arguments

ordered	logical. TRUE orders the result in the order data was entered into the database, while FALSE returns the result in a random order, to the benefit of faster query execution.

Details

The 'DATASET' table gives information about the different datasets fetched from different providers at regular intervals. It provides a unique id for each dataset, the frequency of data, the number of records (datapoints) in each dataset, the minimum and maximum time coverage, when the dataset was last updated, and information about the data source, provider, and method of data access.

Value

A data.table with information about the available datasets in the database.

See Also

sm_datasources, sm_series, samadb

Examples

```
sm_datasets()
```

sm_datasources *Retrieve Data Sources Table*

Description

This function pulls and returns a table called 'DATASOURCE' from the database.

Usage

```
sm_datasources(ordered = TRUE)
```

Arguments

ordered	logical. TRUE orders the result in the order data was entered into the database, while FALSE returns the result in a random order, to the benefit of faster query execution.

Details

The 'DATASOURCE' table gives information about the sources of data in this database, including the source website, and the number of datasets available from the source.

Value

A data.table with information about the sources of data in the database.

See Also

sm_datasets, sm_series, samadb

Examples

```
sm_datasources()
```

sm_expand_date *Generate Temporal Identifiers from a Date Column*

Description

This function expands a date column and generates additional temporal identifiers from it (year, month, quarter, day).

Usage

```
sm_expand_date(
  x,
  gen = c("year", "quarter", "month"),
  origin = "1899-12-30",
  keep.date = TRUE,
  remove.missing.date = TRUE,
  sort = TRUE,
  as.factor = TRUE,
  name = "date",
  ...
)
```

Arguments

x	either a vector of class 'Date', or coercible to date using as.Date, or a data frame / list containing with a date-column called name.
gen	character. A vector of identifiers to generate from x. The possible identifiers are found in .SAMADB_T.
origin	character / Date. Passed to as.Date: for converting numeric x to date. The default reflects converting date-numbers from Excel for Windows.
keep.date	logical. TRUE will keep the date variable in the resulting dataset, FALSE will remove the date variable in favor of the generated identifiers.
remove.missing.date	logical. TRUE will remove missing values in x. If x is a dataset, rows missing the date variable will be removed.
sort	logical. TRUE will sort the data by the date column.
as.factor	TRUE will generate quarters and months as factor variables. It is also possible to use as.factor = "ordered" to generate ordered factors. FALSE will generate quarters and months as integer variables.
name	character. The name of the date variable to expand.
...	not used.

Value

A data.table containing the computed identifiers as columns. See Examples.

See Also

sm_as_date, samadb

Examples

```
# First a basic example
x <- seq.Date(as.Date("1999-01-01"), as.Date("2000-01-01"), by = "month")
sm_expand_date(x)
sm_expand_date(x, gen = .SAMADB_T[-1L], keep.date = FALSE)
```

```
# Now using the API
sm_expand_date(sm_data("BUSINESS_CYCLES"))

# Same thing
sm_data("BUSINESS_CYCLES", expand.date = TRUE)
```

`sm_pivot_longer` *Reshape Column-Based Data to Long Format*

Description

This function automatically reshapes wide (column-based) data into a long format akin to the format of the raw data coming from the database (`sm_data(..., wide = FALSE)`). Internally it uses `melt` from *data.table*.

Usage

```
sm_pivot_longer(
  data,
  id_cols = intersect(.SAMADB_T, names(data)),
  to_value = setdiff(names(data), id_cols),
  variable_name = "series",
  value_name = "value",
  label_name = "label",
  na.rm = TRUE,
  variable.factor = TRUE,
  label.factor = TRUE,
  ...
)
```

Arguments

`data`	a wide format data frame where all series have their own column.
`id_cols`	character. Temporal identifiers of the data. By default all variables in `.SAMADB_T` are selected.
`to_value`	character. The names of all series to be stacked into the long format data frame.
`variable_name`	character. The name of the variable to store the names of the series.
`value_name`	character. The name of the variable to store the data values.
`label_name`	character. The name of the variable to store the series labels.
`na.rm`	logical. `TRUE` will remove all missing values from the long data frame.
`variable.factor, label.factor`	logical. `TRUE` will code the "series" and "label" columns as factors, which is more memory efficient.
`...`	further arguments passed to `melt`.

Value

A data.table with the reshaped data.

See Also

sm_pivot_wider, samadb

Examples

```
# Return all electricity indicators from the year 2000 onwards
data <- sm_data("ELECTRICITY", from = 2000)
sm_pivot_longer(data)
```

---

sm_pivot_wider *Reshape Long API Data to Column-Based Format*

Description

This function automatically reshapes long (stacked) raw data from the API (sm_data(..., wide = FALSE)) to a wide format where each variable has its own column. Internally it uses pivot from *collapse*.

Usage

```
sm_pivot_wider(
  data,
  id_cols = intersect(.SAMADB_T, names(data)),
  names_from = "series",
  values_from = "value",
  labels_from = if (any(names(data) == "label")) "label" else NULL,
  expand.date = FALSE,
  ...
)
```

Arguments

data	raw data from the API: A long format data frame where all values are stacked in a value column.
id_cols	character. Temporal identifiers of the data. By default all variables in .SAMADB_T are selected.
names_from	character. The column containing the series codes. These will become the names of the new columns in the wider data format.
values_from	character. The column containing the data values.
labels_from	character. The column containing the labels describing the series.
expand.date	logical. TRUE will call sm_expand_date on the data after reshaping.
...	further arguments passed to pivot or sm_expand_date.

Value

A data.table with the reshaped data.

See Also

sm_pivot_longer, samadb

Examples

```
# Return all electricity indicators from the year 2000 onwards
sm_pivot_wider(sm_data("ELECTRICITY", from = 2000, wide = FALSE))
```

---

sm_series *Retrieve Series Table*

Description

This function pulls the 'SERIES' table from the database, providing information about the time series in the database. Each series is given a code which unique across datasets.

Usage

```
sm_series(
  dsid = NULL,
  series = NULL,
  dataset.info = FALSE,
  ordered = TRUE,
  return.query = FALSE
)
```

Arguments

dsid	character. (Optional) id's of datasources matching the 'dsid' column of the 'DATASET' table (retrieved using sm_datasets()) for which series information is to be returned.
series	character. (Optional) codes of series for which information in to be returned. If 'dsid' is also specificed, the two are combined using SQL 'OR' i.e. these series are retrieved in addition to all series matched through 'dsid'.
dataset.info	logical. TRUE returns additional information from the 'DATASET' table (the dataset name, when data was last updated, the source id and the data provider and access mode).
ordered	logical. TRUE returns the series in a fixed order, while FALSE returns the result in a random order, to the benefit of faster query execution.
return.query	logical. TRUE will not query the database but instead return the constructed SQL query as a character string (for debugging purposes).

Details

Each series is given a code which is unique across datasets. Each series also has a label describing the series. Further information recorded are the series frequency, unit, whether it was seasonally adjusted, number of observations, minimum and maximum date, and (optionally) topic, alternative code provided by the data source (data retrieved from EconData uses EconData codes as series codes, so the 'src_code' field gives the codes used by the SARB or STATSSA), or further comments on the series.

Value

A `data.table` with information about the available series in the database.

See Also

`sm_datasources`, `sm_datasets`, `sm_data`, `samadb`

Examples

```
# By default returns all series
sm_series()

# Adding information about the dataset and provider
sm_series(dataset.info = TRUE)

# Only series in the QB
sm_series("QB")
```

`sm_transpose` *Transpose a Wide Dataset to a Row-Based Format*

Description

This function is called by `sm_write_excel` with option `transpose = TRUE` to generate a row-based tabular data format from a wide data frame in R that is suitable for exporting to Excel.

Usage

```
sm_transpose(data, date.format = "%d/%m/%Y")
```

Arguments

`data`	a wide format data frame where each column is a variable and the first variable uniquely identifies the data.
`date.format`	a format for date columns which is passed to `format.Date`. When transposing wide, dates are converted to character. The default R YYYY-MM-DD format for dates is often not recognized by Excel. By default dates are transformed to DD/MM/YYYY format which Excel (UK English) recognizes. Putting `FALSE` here does not transform dates into another format.

Value

A transposed data frame or data.table (the class of the input is preserved).

See Also

transpose, sm_pivot_wider, sm_write_excel, samadb

Examples

```
sm_transpose(sm_data("ELECTRICITY"))
```

---

sm_write_excel *Export Wide Data to Excel*

Description

This function exports a wide format dataset to a column- (default) or row-oriented Excel format.

Usage

```
sm_write_excel(
  data,
  ...,
  transpose = FALSE,
  transpose.date.format = "%d/%m/%Y"
)
```

Arguments

data	a wide dataset from sm_data or reshaped to a wide format with sm_pivot_wider.
...	further arguments to write_xlsx. As a minimum a path needs to be supplied that ends with the name of the Excel file. See Examples.
transpose	logical. If TRUE, the result is returned in a row-oriented Excel format. The default is column oriented (same as the dataset in R).
transpose.date.format	argument passed to sm_transpose, setting the format of date columns when data is transposed.

Value

Writes an Excel file to the specified path (no return value).

See Also

sm_transpose, write_xlsx, samadb

Examples

```
## Not run:
# Getting electricity indicators from 2000
data <- sm_data("ELECTRICITY", from = 2000)

# Saving to different Excel formats
sm_write_excel(data, "ELECTRICITY.xlsx")
sm_write_excel(data, "ELECTRICITY.xlsx", transpose = TRUE)

# Saving to alternative path
sm_write_excel(data, "C:/Users/.../ELECTRICITY.xlsx")

## End(Not run)
```

Index