Package ‘glioblastomaEHRsData’

May 8, 2026

Type Package

Title Descriptive Analysis on Three Glioblastoma EHRs Datasets

Version 1.1.0

Maintainer Samuele Marelli <`samu2003.marelli@gmail.com`>

Description Provides functions to load and analyze three open Electronic Health Records (EHRs) datasets of patients diagnosed with glioblastoma, previously released under the Creative Common Attribution 4.0 International (CC BY 4.0) license. Users can generate basic descriptive statistics, frequency tables and save descriptive summary tables, as well as create and export univariate or bivariate plots. The package is designed to work with the included datasets and to facilitate quick exploratory data analysis and reporting. More information about these three datasets of EHRs of patients with glioblastoma can be found in this article: Gabriel Cerono, Ombretta Melaiu, and Davide Chicco, 'Clinical feature ranking based on ensemble machine learning reveals top survival factors for glioblastoma multiforme', Journal of Healthcare Informatics Research 8, 1-18 (March 2024). <doi:10.1007/s41666-023-00138-1>.

License GPL-3

URL `https://github.com/Samumar4321/glioblastomaEHRsData`

Depends R (>= 3.5)

Imports DataExplorer, flextable, ggplot2, rmarkdown, summarytools, table1, tinytex

Suggests ragg, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

SystemRequirements pandoc (>= 1.12.3) - http://pandoc.org, tcltk (optional)

NeedsCompilation no

Author Samuele Marelli [aut, cre],
Davide Chicco [aut]

Repository CRAN

Date/Publication 2026-01-30 14:30:08 UTC

Contents

`descriptiveTableMunich2019dataset` *Descriptive statistics table for the munich2019dataset*

Description

Function to create, display and optionally export a table containing descriptive statistics for the munich2019dataset.
The table is created using the table1 package and is grouped by survival status.

Usage

```
descriptiveTableMunich2019dataset(savePath = NULL)
```

Arguments

`savePath` Character (optional). String specifying the path and filename for exporting the table.

- If NULL (default), the table will not be exported.
- If "" (empty string), the table will be saved in the working directory using a default name.
- Expected format: ’filepath/filename.extensions’ where:

- filepath is the directory (must already exist), default is the working directory;
- filename is the name of the file, default to 'table_munich2019dataset_timestamp.png';
- extension must be one of png, pdf html and must be specified, unless using the default filename.

Value

A 'table1' object containing descriptive statistics grouped by survival status.

See Also

`table1::table1()`

Examples

```
# Create and display the table, without saving anything
descriptiveTableMunich2019dataset()


# Create, display and save the table giving a path, filename and extension
descriptiveTableMunich2019dataset(savePath = "tables/munich_web_table.html")

# Create, display and save the table giving a path, filename and extension
descriptiveTableMunich2019dataset(savePath = "tables/munich_table.pdf")

# Create, display and save the table giving only the path, default name will be used
descriptiveTableMunich2019dataset(savePath = "tables/")

# Create, display and save the table giving a path and filename, default extension will be used
descriptiveTableMunich2019dataset(savePath = "tables/munich_table")
```

`descriptiveTableTainan2020dataset` *Descriptive statistic table for the tainan2020dataset*

Description

Function to create, display and optionally export a table containing descriptive statistics for the tainan2020dataset.
The table is created using the table1 package and is grouped by survival status.

Usage

```
descriptiveTableTainan2020dataset(savePath = NULL)
```

Arguments

`savePath`	Character (optional). String specifying the path and filename for exporting the table. • If NULL (default), the table will not be exported. • If "" (empty string), the table will be saved in the working directory using a default name. • Expected format: 'filepath/filename.extensions' where: – filepath is the directory (must already exist), default is the working directory; – filename is the name of the file, default to 'table_tainan2020dataset_timestamp.png'; – extension must be one of png, pdf or html and must be specified, unless using the default filename.

Value

A 'table1' object containing descriptive statistics grouped by survival status.

See Also

`table1::table1()`

Examples

```
# Create and display the table, without saving anything
descriptiveTableTainan2020dataset()


# Create, display and save the table giving a path, filename and extension
descriptiveTableTainan2020dataset(savePath = "tables/tainan_web_table.html")

# Create, display and save the table giving a path, filename and extension
descriptiveTableTainan2020dataset(savePath = "tables/tainan_table.pdf")

# Create, display and save the table giving only the path, default name will be used
descriptiveTableTainan2020dataset(savePath = "tables/")

# Create, display and save the table giving a path and filename,default extension will be used
descriptiveTableTainan2020dataset(savePath = "tables/tainan_table")
```

`descriptiveTableUtrecht2019dataset` *Descriptive statistic table for the utrecht2019dataset*

Description

Function to create, display and optionally export a table containing descriptive statistics for the utrecht2019dataset
The table is created using the table1 package and is grouped by survival status.

Usage

```
descriptiveTableUtrecht2019dataset(savePath = NULL)
```

Arguments

`savePath`	Character (optional). String specifying the path and filename for exporting the table. • If NULL (default), the table will not be exported. • If "" (empty string), the table will be saved in the working directory using a default name. • Expected format: 'filepath/filename.extensions' where: – filepath is the directory (must already exist), default is the working directory; – filename is the name of the file, default to 'table_utrecht2019dataset_timestamp.png'; – extension must be one of png, pdf or html and must be specified, unless using the default filename.

Value

A 'table1' object containing descriptive statistics grouped by survival status.

See Also

`table1::table1()`

Examples

```
# Create and display the table, without saving anything
descriptiveTableUtrecht2019dataset()


# Create, display and save the table giving a path, filename and extension
descriptiveTableUtrecht2019dataset(savePath = "tables/utrecht_web_table.html")

# Create, display and save the table giving a path, filename and extension
descriptiveTableUtrecht2019dataset(savePath = "tables/utrecht_table.pdf")

# Create, display and save the table giving only the path, default name will be used
descriptiveTableUtrecht2019dataset(savePath = "tables/")

# Create, display and save the table giving a path and filename,default extension will be used
descriptiveTableUtrecht2019dataset(savePath = "tables/utrecht_table")
```

`global_variables` *Declarations for glibal variables*

Description

This file is used to suppress global variables NOTEs while running 'R CMD check', these variables are used as internal datasets in the package functions. They are declared in the 'data/' folder and are automatically loaded using 'lazydata = TRUE'

Usage

```
global_variables
```

Format

An object of class `NULL` of length 0.

Details

@keywords internal @name global_variables

`munich2019dataset` *EHR data of patients affected by Glioblastoma (GBM), collected from Munich, Germany.*

Description

The dataset contains data from 60 patients, treated at the 2 hospitals Klinikum rechts der Isar (TUM) and Klinikum Bogenhausen (STKM), Munich, Germany.
It focuses on the prognostic impact of cytosolic Hsp70 (cHsp70) expression and MGMT promoter methylation status.

Usage

```
data(munich2019dataset)
```

Format

A data frame with 60 rows and 8 variables:

age_years Numeric. Patient's age in years.

cHsp70_low0_high1 Factor. "Low" (0) or "High" (1) cytosolic Hsp70 expression.

MGMTmethylation_methylated1_unmethylated0 Factor "unmethylated" (0) or "methylated" (1) MGMT promoter.

OS_months Numeric. Overall survival in months.

PFS_months Numeric. Progression-free survival in months.

sex_male0_female1 Factor. "Male" (0) or "Female" (1).

survived_yes1_no0 Factor. "Dead" (0) or "Alive" (1) at their most recent check-up.

tumor_progression_yes1_no0 Factor. "No" (0) or "Yes" (1) for tumor progression after treatment.

References

Lämmer F, Delbridge C. Würstle S et al. (2019) Cytosolic Hsp70 as a biomarker to predict clinical outcome in patients with glioblastoma. PMID: 31430337. 14(8): e0221502. https://doi.org/10.1371/journal.pone.0221502

Examples

```
data(munich2019dataset)
head(munich2019dataset)
```

`Munich2019datasetDescriptiveStatistics`

Munich2019datasetDescriptiveStatistics

Description

Provides descriptive statistics for the 'munich2019dataset'. It can display summary statistic for continuous variables, frequency tables for categorical variables or both.

Usage

```
Munich2019datasetDescriptiveStatistics(show = "all")
```

Arguments

`show`	character string for the type of statistic to show. There are three options: • 'continuous', summary statistic for continuous variables will be displayed. • 'categorical', frequency tables for categorical variables will be displayed. • 'all', both summary will be displayed. • none of the above a warning will be launched.

Value

A list or an object depending on the value of show: - If `show = "continuous"` A data frame with descriptive statistics. - If `show = "categorical"` A list of frequency tables. - If `show = "all"` A list of 2 items, the first is a data frame for continuous variable, the second is a list for categorical variables. - If show is not one of the above return a warning.

See Also

`summarytools::descr()`, `summarytools::freq()`

Examples

```
# Show both continuous and categorical stats
all <- Munich2019datasetDescriptiveStatistics(show = "all")

# Mean value for the OS_months columns
all$continuous["Mean", "OS_months"]

# Min, Median and Max value for ALL the numeric columns
all$continuous[c("Min","Median","Max"),]

# Get the frequency table of the third categorical column
all$categorical[[3]]

# Only continuous variables
cont <- Munich2019datasetDescriptiveStatistics("continuous")

# Statistics for age_years column
cont[, "age_years"]

# Only categorical variables
cat <- Munich2019datasetDescriptiveStatistics("categorical")

# Frequency values for all factors of the first categorical column
cat[[1]][,"Freq"]

# Statistics for the Low factor of the first categorical column
cat[[1]]["Low",]

# Frequency of the Low factor, a value of the first categorical column
cat[[1]]["Low", "Freq"]
```

`plotMunich2019dataset` *Plot variables from the munich2019dataset*

Description

This function creates a plot of one or two variables from the 'munich2019dataset' dataframe. It automatically chooses the appropriate plot type based on the variable types. It can export and save the plot by specifying the directory, filename and extension. Provided variable names need to be in the dataset, otherwise an error will be launched. The function makes it easy to plot univariate and bivariate plots while offering a quick way to save and export them.

Usage

```
plotMunich2019dataset(name1, name2 = NA, savePath = NA)
```

Arguments

`name1`	Character. The name of the first variable to plot.
`name2`	Character (optional). The name of the second variable for bivariate plots. Default is NA.
`savePath`	Character (optional). File path where the plot should be saved. Default is NULL which means no plot will be saved. To save a plot using all the default options put an empty string. The format must be: 'filepath/filename.extension' where: • filepath is the directory (must already exist), default is the working directory; • filename is the name of the file, default to 'plot_munich2019dataset_var1_var2_timestamp.png'; • extension must be one of the supported 'ggsave' extensions and must be specified, unless using the default file name.

Details

The function supports the following plotting logic:

- If only 'name1' is provided:
    - Numeric/integer variable → Histogram.
    - Character/factor variable → Bar plot.
- If both 'name1' and 'name2' are provided:
    - One numeric/integer and one categorical → Boxplot.
    - Both categorical → Grouped bar plot.

Value

A 'ggplot2' object representing the generated plot. If the specified variables are not found in the dataset, returns a warning.

See Also

`DataExplorer::plot_bar()`, `DataExplorer::plot_histogram()`, DataExplorer::plot_boxplot, `savePlot()`

Examples

```
# Univariate plot without saving
plotMunich2019dataset("age_years")

# Bivariate plot without saving
plotMunich2019dataset("cHsp70_low0_high1", "sex_male0_female1")


# Bivariate plot saved in the path directory with the chosen name and extension
plotMunich2019dataset("age_years", "sex_male0_female1", savePath = NA)

# Univariate plot saved in the working directory with the chosen name and extension
plotMunich2019dataset("sex_male0_female1", savePath = NA)

# Univariate plot saved in the path directory with default name and extension
plotMunich2019dataset("MGMTmethylation_methylated1_unmethylated0", savePath = NA)
```

`plotTainan2020dataset` *Plot variables from the tainan2020dataset*

Description

This function creates a plot of one or two variables from the 'tainan2020dataset' dataframe. It automatically chooses the appropriate plot type based on the variable types and it can export and save the plot by specifying the directory, filename and extension. Provided variable names need to be in the dataset, otherwise an error will be launched. The function makes it easy to plot univariate and bivariate plots while offering a quick way to save and export them.

Usage

```
plotTainan2020dataset(name1, name2 = NA, savePath = NA)
```

Arguments

`name1`	Character. The name of the first variable to plot.
`name2`	Character (optional). The name of the second variable for bivariate plots. Default is NA.
`savePath`	Character (optional). File path where the plot should be saved. Default is NULL which means no plot will be saved. To save a plot using all the default options put an empty string. The format must be: 'filepath/filename.extension' where: • filepath is the directory (must already exist), default is the working directory; • filename is the name of the file, default to 'plot_tainan2020dataset_var1_var2_timestamp.png'; • extension must be one of the supported 'ggsave' extensions and must be specified, unless using the default file name.

Details

The function supports the following plotting logic:

- If only 'name1' is provided:
    - Numeric/integer variable → Histogram.
    - Character/factor variable → Bar plot.
- If both 'name1' and 'name2' are provided:
    - One numeric/integer and one categorical → Boxplot.
    - Both categorical → Grouped bar plot.

Value

A 'ggplot2' object representing the generated plot. If the specified variables are not found in the dataset, returns a warning.

See Also

`DataExplorer::plot_bar()`, `DataExplorer::plot_histogram()`, DataExplorer::plot_boxplot, `savePlot()`

Examples

```
# Univariate plot without saving
plotTainan2020dataset("TMZ_based_chemo_yes1_no0")

# Bivariate plot without saving
plotTainan2020dataset("PFS_months", "OS_months")


# Bivariate plot saved in the specified directory with the chosen name and extension
plotTainan2020dataset("age_years", "chemo_yes1_no0", savePath = NA)

# Bivariate plot saved in the working directory with the chosen name and extension
plotTainan2020dataset("PFS_months", "OS_months", savePath = NA)

# Bivariate plot saved in a path directory with default name and extension
plotTainan2020dataset("PFS_months", "radiation_dose_Gy", savePath = NA)
```

`plotUtrecht2019dataset`

Plot variables from the utrecht2019dataset

Description

This function creates a plot of one or two variables from the 'utrecht2019dataset' dataframe. It automatically chooses the appropriate plot type based on the variable types and it can export and save the plot by specifying the directory, filename and extension. Provided variable names need to be in the dataset, otherwise an error will be launched. The function makes it easy to plot univariate and bivariate plots while offering a quick way to save and export them.

Usage

```
plotUtrecht2019dataset(name1, name2 = NA, savePath = NA)
```

Arguments

`name1`	Character. The name of the first variable to plot.
`name2`	Character (optional). The name of the second variable for bivariate plots. Default is NA.
`savePath`	Character (optional). File path where the plot should be saved. Default is NULL which means no plot will be saved. To save a plot using all the default options put an empty string. The format must be: 'filepath/filename.extension' where: • filepath is the directory (must already exist), default is the working directory; • filename is the name of the file, default to 'plot_utrecht2019dataset_var1_var2_timestamp.png'; • extension must be one of the supported 'ggsave' extensions and must be specified, unless using the default file name.

Details

The function supports the following plotting logic:

- If only 'name1' is provided:
  - Numeric/integer variable → Histogram.
  - Character/factor variable → Bar plot.
- If both 'name1' and 'name2' are provided:
  - One numeric/integer and one categorical → Boxplot.
  - Both categorical → Grouped bar plot.

Value

A 'ggplot2' object representing the generated plot. If the specified variables are not found in the dataset, returns a warning.

See Also

`DataExplorer::plot_bar()`, `DataExplorer::plot_histogram()`, DataExplorer::plot_boxplot, `savePlot()`

Examples

```
# Univariate plot without saving
plotUtrecht2019dataset("SVZ_status_nocontact0_contact1")

# Bivariate plot without saving
plotUtrecht2019dataset("post_surgery_therapy_none0_monotherapy1_RTandTMZ2", "OS_months")


# Bivariate plot saved in the specified directory with the chosen name and extension
plotUtrecht2019dataset("age_years", "KPS_less70.0_more70.1", savePath = NA)

# Univariate plot save in the working directory with the chosen name and extension
plotUtrecht2019dataset("survived_yes1_no0", savePath = NA)

# Bivariate plot save in the path directory using the default name
plotUtrecht2019dataset("survived_yes1_no0", savePath = NA)
```

`tainan2020dataset` *EHR data of patients affected by Glioblastoma (GBM) from Tainan and Taiwan medical centers and branches.*

Description

Data form patients affected by GBM in the Tainan and Taiwan medical centers and affiliated branches between 2005 and 2016.
It contains data from 84 patients, ranging from demographical information (age and sex) and treatment characteristics (chemotherapy, radiation volume and dose) to surgical details and outcomes (Overall and Progress Free Survival).
The dataset focuses on the impact of radiation volume and dose; it finds that patients with an elevated radiation dose (>60.0 Gy) had better median PFS and OS compared to patients who received a standard radiation dose.
Some data may be missing due to the nature of clinical records.

Usage

```
data(tainan2020dataset)
```

Format

A data frame with 84 rows and 12 variables:

age_years Numeric. Patient's age.

chemo_yes1_no0 Factor. "No" (0) or "Yes" (1) for chemotherapy.

OS_months Numeric. Overall Survival expressed in months.

PFS_months Numeric. Progress Free Survival expressed in months.

radiation_dose_Gy Numeric. Radiation dose used in chemotherapy expressed in Gy.

radiation_volume_mL Numeric. Radiation volume used in chemotherapy expressed in mL.

sex_male0_female1 Factor. "Male" (0) or "Female" (1).

surgery_resection1_biopsy0 Factor. "Biopsy" (0) or "Resection" (1) for surgery

survived_yes1_no0 Factor. "Dead" (0) or "Alive" (1) at their most recent check-up.

TMZ_based_chemo_yes1_no0 Factor. "No" (0) or "Yes" (1) for TMZ based chemotherapy.

tumorProgression_no0_yes1 Factor. "No" (0) or "Yes" (1) for tumor progression after the initial treatment.

year_of_diagnosis_05to10_0_10to16_1 Factor. "2005-2010" (0) or "2010-2016" (1) for the year of diagnosis.

References

Li-Tsun Shieh, How-Ran Guo, Chung-Han Ho et al. (2020). Survival of glioblastoma treated with a moderately escalated radiation dose—Results of a retrospective analysis. PMID: 32413077. 15(5): e0233188. https://doi.org/10.1371/journal.pone.0233188

Examples

```
data(tainan2020dataset)
head(tainan2020dataset)
```

`Tainan2020datasetDescriptiveStatistics`

Tainan2020datasetDescriptiveStatistics

Description

Provides descriptive statistics for the ’tainan2020dataset’. It can display summary statistic for continuous variables, frequency tables for categorical variables or both.

Usage

```
Tainan2020datasetDescriptiveStatistics(show = "all")
```

Arguments

`show`	character string for the type of statistic to show. There are three options: • ’continuous’, summary statistic for continuous variables will be displayed. • ’categorical’, frequency tables for categorical variables will be displayed. • ’all’, both summary will be displayed. • none of the above a warning will be launched.

Value

A list or an object depending on the value of show: - If show = `"continuous"` A data frame with descriptive statistics. - If show = `"categorical"` A list of frequency tables. - If show = `"all"` A list of 2 items, the first is a data frame for continuous variable, the second is a list for categorical variables. - If show is not one of the above return a warning.

See Also

`summarytools::descr()`, `summarytools::freq()`

Examples

```
# Show both continuous and categorical stats
all <- Tainan2020datasetDescriptiveStatistics(show = "all")

# Mean value for the OS_months columns
all$continuous["Mean", "OS_months"]

# Min, Median and Max value for ALL the numeric columns
all$continuous[c("Min","Median","Max"),]

# Get the frequency table of the third categorical column
all$categorical[[3]]

# Only continuous variables
cont <- Tainan2020datasetDescriptiveStatistics("continuous")

# Statistics for age_years column
cont[, "age_years"]

# Only categorical variables
cat <- Tainan2020datasetDescriptiveStatistics("categorical")

# Frequency values for all factors of the first categorical column
cat[[1]][,"Freq"]

# Statistics for the Yes factor of the first categorical column
cat[[1]]["Yes",]

# Frequency of the No factor, a value of the first categorical column
cat[[1]]["No", "Freq"]
```

`utrecht2019dataset` *EHR data of patients affected by Glioblastoma (GBM) treated at the University Medical Centre of Utrecht.*

Description

Electronic Health Records (EHR) of patients affected by GBM at the University Medical Centre of Utrecht (UMCU) between 2005-2013.
This dataset contains clinical information from 647 patients collected to study prognostic factor influencing overall survival in GBM patients, this study focuses on the involvement of the subventricular zone (SVZ), a brain region that may play a role in tumor progression.
The dataset found out that SVZ contact meant worse prognosis for patients, in fact SVZ-contacting tumors had worse median OS compared to non SVZ-contacting patients.
Some data may be missing due to the nature of clinical records.

Usage

```
data(utrecht2019dataset)
```

Format

A data frame with 647 rows and 7 variables:

age_years Numeric. Patient's age.

KPS_less70.0_more70.1 Factor. "<70" (0) or ">70" (1) KPS level.

OS_months Numeric. Overall Survival expressed in months.

post_surgery_therapy_none0_monotherapy1_RTandTMZ2 Factor. "None" (0), "Monotherapy" (1) or "RT+TMZ" (2) for therapy regime.

surgery_biopsy0_resection1 Factor. "Biopsy" (0) or "Resection" (1) for surgery type.

survived_yes1_no0 Factor. "Dead" (0) or "Alive" (1) at their most recent check-up.

SVZ_status_nocontact0_contact1 Factor. "No" (0) or "Yes" (1) for SVZ contact.

References

Berendsen S, van Bodegraven E, Seute T et al. (2019). Adverse prognosis of glioblastoma contacting the subventricular zone: Biological correlates. PMID: 31603915. 14(10): e0222717. https://doi.org/10.1371/journal.pone.0222717

Examples

```
data(utrecht2019dataset)
head(utrecht2019dataset)
```

`Utrecht2019datasetDescriptiveStatistics` *Utrecht2019datasetDescriptiveStatistics*

Description

Provides descriptive statistics for the 'utrecht2019dataset'. It can display summary statistic for continuous variables, frequency tables for categorical variables or both.

Usage

```
Utrecht2019datasetDescriptiveStatistics(show = "all")
```

Arguments

`show`	character string for the type of statistic to show. There are three options: • 'continuous', summary statistic for continuous variables will be displayed. • 'categorical', frequency tables for categorical variables will be displayed. • 'all', both summary will be displayed. • none of the above a warning will be launched.

Value

A list or an object depending on the value of `show`: - If `show` = `"continuous"` A data frame with descriptive statistics. - If `show` = `"categorical"` A list of frequency tables. - If `show` = `"all"` A list of 2 items, the first is a data frame for continuous variable, the second is a list for categorical variables. - If show is not one of the above return a warning.

See Also

`summarytools::descr()`, `summarytools::freq()`

Examples

```
# Show both continuous and categorical stats
all <- Utrecht2019datasetDescriptiveStatistics(show = "all")

# Mean value for the OS_months columns
all$continuous["Mean", "OS_months"]

# Min, Median and Max value for ALL the numeric columns
all$continuous[c("Min","Median","Max"),]

# Get the frequency table of the third categorical column
all$categorical[[3]]

# Only continuous variables
cont <- Utrecht2019datasetDescriptiveStatistics("continuous")
```

```
# Statistics for age_years column
cont[, "age_years"]

# Only categorical variables
cat <- Utrecht2019datasetDescriptiveStatistics("categorical")

# Frequency values for all factors of the second categorical column
cat[[2]][,"Freq"]

# Statistics for the Yes factor of the second categorical column
cat[[2]]["Monotherapy",]

# Frequency of the No factor, a value of the second categorical column
cat[[2]]["RT+TMZ", "Freq"]
```

Index