Package: clinical (via r-universe)

May 7, 2026

Version 0.1

Date 2025-06-04

Maintainer Stefano Cacciatore <`tkcaccia@gmail.com`>

Title Clinical Metadata Exploration and Feature Matching

Description A collection of tools to easily analyze clinical data, including functions for correlation analysis, and statistical testing. The package facilitates the integration of clinical metadata with other omics layers, enabling exploration of quantitative variables. It also includes the utility for frequency matching samples across a dataset based on patient variables.

Depends R (>= 3.5.0), stats, minerva, Matrix, clinfun, methods

Suggests knitr, rmarkdown

VignetteBuilder knitr

SuggestsNote No suggestions

License GPL (>= 2)

NeedsCompilation yes

Author Stefano Cacciatore [aut, trl, cre] (<`https://orcid.org/0000-0001-7052-7156`>)

Repository https://tkcaccia.r-universe.dev

Date/Publication 2025-06-04 15:23:58 UTC

RemoteUrl https://github.com/tkcaccia/clinical

RemoteRef HEAD

RemoteSha 48ddc7dbaf285cb58df08f708bdce94c90e45ffa

Contents

`add_analysis` *Add analysis results to a clinical object*

Description

This function appends the results of an analysis to a clinical object.

Usage

```
add_analysis(ma, name, x)
```

Arguments

`ma`	A clinical object to which results will be attached.
`name`	The name (string) of the analysis.
`x`	The result object (e.g., data frame, matrix) to attach.

Value

Returns the updated clinical object with the new analysis appended.

Author(s)

Stefano Cacciatore

Examples

```
data(prostate)


ma=initialization(prostate[,"Hospital"])
ma=add_analysis(ma,"Gender",prostate[,"Gender"])
ma=add_analysis(ma,"Gleason score",prostate[,"Gleason score"])
ma=add_analysis(ma,"BMI",prostate[,"BMI"])
ma=add_analysis(ma,"Age",prostate[,"Age"])
ma
```

`as.data.matrix` *Convert Character Matrix to Numeric Matrix*

Description

Converts a matrix of character or factor entries to a numeric matrix, preserving row and column names.

Usage

```
as.data.matrix(x)
```

Arguments

`x`	A matrix containing character or factor values that represent numeric entries.

Details

This function is useful when a matrix has been read as character or factor and needs to be transformed into numeric format for analysis. It preserves both the row and column names of the input matrix.

Value

A numeric matrix of the same dimensions as `x`, with the same row and column names.

Author(s)

Stefano Cacciatore

Examples

```
# Create a character matrix
ma <- matrix(c("1", "2", "3", "4"), ncol = 2)

# Convert to numeric matrix
ma_numeric <- as.data.matrix(ma)

# Print result
ma_numeric
```

`categorical.test` *Categorical Information*

Description

Summarization of the categorical information.

Usage

```
categorical.test (name,x,y,total.column=FALSE,...)
```

Arguments

`name`	the name of the feature.
`x`	the information to summarize.
`y`	the classification of the cohort.
`total.column`	option to visualize the total (by default = "`FALSE`").
`...`	further arguments to be passed to the function.

Value

The function returns a table with the summarized information and The p-value computed using the Fisher's test.

Author(s)

Stefano Cacciatore

See Also

`correlation.test`,`continuous.test`, `txtsummary`

Examples

```
data(prostate)

hosp=prostate[,"Hospital"]
gender=prostate[,"Gender"]
GS=prostate[,"Gleason score"]
BMI=prostate[,"BMI"]
age=prostate[,"Age"]

A=categorical.test("Gender",gender,hosp)
B=categorical.test("Gleason score",GS,hosp)

C=continuous.test("BMI",BMI,hosp,digits=2)
D=continuous.test("Age",age,hosp,digits=1)
```

```
rbind(A,B,C,D)
```

`continuous.test` *Continuous Information*

Description

Summarization of the continuous information.

Usage

```
continuous.test (name,
                 x,
                 y,
                 center = c("median", "mean"),
                 digits = 3,
                 scientific = FALSE,
                 range = c("IQR","95%CI","range","sd"),
                 logchange = FALSE,
                 pos=2,
                 method=c("non-parametric","parametric"),
                 total.column=FALSE, ...)
```

Arguments

`name`	the name of the feature.
`x`	the information to summarize.
`y`	the classification of the cohort.
`center`	A character string specifying the measure of central tendency to report: either `"median"` or `"mean"`.
`digits`	how many significant digits are to be used.
`scientific`	either a logical specifying whether result should be encoded in scientific format.
`range`	the range to be visualized.
`logchange`	either a logical specifying whether log2 of fold change should be visualized.
`pos`	a value indicating the position of range to be visualized. 1 for column, 2 for row.
`method`	a character string indicating which test method is to be computed. "non-parametric" (default), or "parametric".
`total.column`	option to visualize the total (by default = "`FALSE`")
`...`	further arguments to be passed to or from methods.

Value

The function returns a table with the summarized information and the relative p-value. For non-parametric method, if the number of group is equal to two, the p-value is computed using the Wilcoxon rank-sum test, Kruskal-Wallis test otherwise. For parametric method, if the number of group is equal to two, the p-value is computed using the Student's t-Test, ANOVA one-way otherwise.

Author(s)

Stefano Cacciatore

See Also

`correlation.test`, `categorical.test`, `txtsummary`

Examples

```
data(prostate)

hosp=prostate[,"Hospital"]
gender=prostate[,"Gender"]
GS=prostate[,"Gleason score"]
BMI=prostate[,"BMI"]
age=prostate[,"Age"]

A=categorical.test("Gender",gender,hosp)
B=categorical.test("Gleason score",GS,hosp)

C=continuous.test("BMI",BMI,hosp,digits=2)
D=continuous.test("Age",age,hosp,digits=1)

rbind(A,B,C,D)
```

`correlation.test` *Continuous Information*

Description

Summarization of the continuous information.

Usage

```
correlation.test (x,
                    y,
                    method = c("pearson", "spearman","MINE"),
                    name=NA,
                    perm=100, ...)
```

Arguments

`x`	a numeric vector.
`y`	a numeric vector.
`method`	a character string indicating which correlation method is to be computed. "pearson" (default), "spearman", or "MINE".
`name`	the name of the feature.
`perm`	number of permutation needed to estimate the p-value with MINE correlation.
`...`	further arguments to be passed to or from methods.

Value

The function returns a table with the summarized information.

Author(s)

Stefano Cacciatore

See Also

`categorical.test`,`continuous.test`, `txtsummary`

Examples

```
data(prostate)

correlation.test(prostate[,"Age"],prostate[,"BMI"],name="correlation between Age and BMI")
```

`frequency_matching` *Frequency Matching*

Description

A method to select unbalanced groupd in a cohort.

Usage

```
frequency_matching (data,label,times=5,seed=1234)
```

Arguments

`data`	a data.frame of data.
`label`	a classification of the groups.
`times`	The ratio between the two groups.
`seed`	a single number for random number generation.

Value

The function returns a list with 2 items or 4 items (if a test data set is present):

`data`	the data after the frequency matching.
`label`	the label after the frequency matching.
`selection`	the rows selected for the frequency matching.

Author(s)

Stefano Cacciatore

Examples

```
data(prostate)



hosp=prostate[,"Hospital"]
gender=prostate[,"Gender"]
GS=prostate[,"Gleason score"]
BMI=prostate[,"BMI"]
age=prostate[,"Age"]

A=categorical.test("Gender",gender,hosp)
B=categorical.test("Gleason score",GS,hosp)

C=continuous.test("BMI",BMI,hosp,digits=2)
D=continuous.test("Age",age,hosp,digits=1)

# Analysis without matching
rbind(A,B,C,D)



# The order is important. Right is more important than left in the vector
# So, Ethnicity will be more important than Age
var=c("Age","BMI","Gleason score")
data.categorized=prostate[,var]

# Extract the Age vector
x <- data.categorized[["Age"]]

# Compute quantiles (0%, 25%, 50%, 75%, 100%) with NA handling
breaks <- quantile(x, probs = c(0, 0.25, 0.5, 0.75, 1), na.rm = TRUE)

# Apply the cut and update the Age column with labeled bins
data.categorized[["Age"]] <- cut(x, breaks = breaks, include.lowest = TRUE)

# Extract the Age vector
x <- data.categorized[["BMI"]]
```

```
# Compute quantiles (0%, 25%, 50%, 75%, 100%) with NA handling
breaks <- quantile(x, probs = c(0, 0.25, 0.5, 0.75, 1), na.rm = TRUE)

# Apply the cut and update the Age column with labeled bins
data.categorized[["BMI"]] <- cut(x, breaks = breaks, include.lowest = TRUE)

times=c(1,1)
names(times)=c("Hospital A","Hospital B")
t=frequency_matching(data.categorized,prostate[,"Hospital"],times=times)



newdata=prostate[t$selection,]

hosp.new=newdata[,"Hospital"]
gender.new=newdata[,"Gender"]
GS.new=newdata[,"Gleason score"]
BMI.new=newdata[,"BMI"]
age.new=newdata[,"Age"]

A=categorical.test("Gender",gender.new,hosp.new)
B=categorical.test("Gleason score",GS.new,hosp.new)

C=continuous.test("BMI",BMI.new,hosp.new,digits=2)
D=continuous.test("Age",age.new,hosp.new,digits=1)

# Analysis with matching
rbind(A,B,C,D)
```

`initialization` *Initialize a Clinical Table Object*

Description

Initializes a new object of class `"clinical.table"`, setting up the grouping variable and a flag for whether to include a total column.

Usage

```
initialization(y, total.column = FALSE)
```

Arguments

`y`	A numeric or factor vector representing the outcome or grouping variable.
`total.column`	Logical; whether to include a total column in the results (default is `FALSE`).

Value

Returns an S4 object of class `"clinical.table"`.

Examples

```
y <- factor(c("A", "B", "A", "B"))
ma <- initialization(y)
```

`intersect` *Intersection of Multiple Vectors*

Description

Returns the intersection of all vectors provided as arguments.

Usage

```
intersect(x, y, ...)
```

Arguments

`x`	A vector.
`y`	A vector.
`...`	Additional vectors to include in the intersection.

Details

This function computes the intersection of two or more vectors by identifying elements that are present in all of them. Unlike the base R `intersect`, which only compares two vectors, this version can take multiple vectors as input.

Value

A character vector (or `NULL` if no intersection exists) containing the elements that are common to all input vectors.

Author(s)

Stefano Cacciatore

See Also

`intersect`, `union`, `setdiff`

Examples

```
x <- c("a", "b", "c")
y <- c("b", "c", "d")
z <- c("c", "b", "e")

intersect(x, y, z)  # Returns "b" and "c"
intersect(x, y, c("f", "g"))  # Returns NULL
```

`multi_analysis` *Continuous Information*

Description

Summarization of the continuous information.

Usage

```
multi_analysis  (data,
                 y,
                 FUN=c("continuous.test","correlation.test"), ...)
```

Arguments

`data`	the matrix containing the continuous values. Each row corresponds to a different sample. Each column corresponds to a different variable.
`y`	the classification of the cohort.
`FUN`	function to be considered. Choices are "`continuous.test`" and "`correlation.test`"
`...`	further arguments to be passed to or from methods.

Value

The function returns a table with the summarized information. If the number of group is equal to two, the p-value is computed using the Wilcoxon rank-sum test, Kruskal-Wallis test otherwise.

Author(s)

Stefano Cacciatore

See Also

`categorical.test`,`continuous.test`,`correlation.test`, `txtsummary`

Examples

```
data(prostate)


multi_analysis(prostate[,c("BMI","Age")],prostate[,"Hospital"],FUN="continuous.test")
```

`prostate` *Clinical Data of a Cohort of Prostate Cancer Patiens*

Description

The data belong to a cohort of 35 patients with prostate cancer from two different hospitals.

Usage

```
data(prostate)
```

Value

The data.frame "`prostate`" with the following elements: "`Hospital`", "`Gender`", "`Gleason score`", "`BMI`", and "`Age`".

Examples

```
data(prostate)

head(prostate)
```

`txtsummary` *Median and Coefficient Interval*

Description

Summarization of a numeric vector.

Usage

```
txtsummary(x,
           f = c("median", "mean"),
           digits = 0,
           scientific = FALSE,
           range = c("IQR", "95%CI", "range", "sd"))
```

Arguments

`x`	a numeric vector.
`f`	xxx.
`digits`	how many significant digits are to be used.
`scientific`	either a logical specifying whether result should be encoded in scientific format.
`range`	the range to be visualized.

Value

The function returns the median and the range (interquartile or 95% coefficient interval) of numeric vetor.

Author(s)

Stefano Cacciatore

See Also

`categorical.test`,`continuous.test`,`correlation.test`, `txtsummary`

Examples

```
data(prostate)

txtsummary(prostate[,"BMI"])
```

Index