Package ‘jgsbook’

June 24, 2024

Type Package

Title Package of the German Book ``Statistik mit R und RStudio'' by Joerg grosse Schlarmann

Description All datasets and functions used in the german book ``Statistik mit R und RStudio'' by grosse Schlarmann (2010-2024) <https://www.produnis.de/R/>.

Version 1.0.7

Date 2024-06-24

Language de-de

License GPL (>= 2)

Depends R (>= 3.5.0)

Encoding UTF-8

Imports statip, jsonlite, httr, curl

LazyData true

RoxygenNote 7.3.1

NeedsCompilation no

Author Jörg große Schlarmann [aut, cre]

Maintainer Jörg große Schlarmann <schlarmann@produnis.de>

Repository CRAN

Date/Publication 2024-06-24 14:20:01 UTC

Contents

`compare.lm` *Compare Linear Models*

Description

This function fits and compares several models (linear, quadratic, cubic, exponential, logarithmic, sigmoidal, power, logistic) to a given set of dependent and independent variables. It returns either a summary of the models with their R-squared values or predicted values based on the models.

Usage

```
compare.lm(dep, ind, predict = FALSE, steps = 0.01)
```

Arguments

`dep`	A numeric vector representing the dependent variable.
`ind`	A numeric vector representing the independent variable.
`predict`	Logical. If TRUE, the function returns predicted values for each model. Defaults to FALSE.
`steps`	Numeric. The step size for generating x-values for predictions. Only used if predict is TRUE. Defaults to 0.01.

Value

A data frame. If predict is FALSE, returns a data frame with the R-squared values for each model. If predict is TRUE, returns a data frame with the original data and predicted values for each model.

Examples

```
x <- c(6, 9, 12, 14, 30, 35, 40, 47, 51, 55, 60)
y <- c(14, 28, 50, 70, 89, 94, 90, 75, 59, 44, 27)
compare.lm(y, x)
compare.lm(y, x, predict=TRUE)
```

`epa` *Datatable of the epa Example*

Description

Datatable of the epa Example

Usage

```
data(epa)
```

Format

A data frame with 620 observations in 6 variables

Details

Variables in the dataset:

- sex. a factor with levels `m` `w` `d`, giving the proband's sex
- age. a numeric vector
- cms. a numeric vector
- risk. a dichotome vector, 0 = not at risk, 1 = at risk
- expert. a dichotome vector of expert's decision, 0 = not at risk, 1 = at risk
- decu. a dichotome vector, 0 = no decubitus, 1 = decubitus

Source

`https://www.produnis.de/R/`

`Faktorenbogen` *Datatable of the Faktorenbogen Example for factor analysis*

Description

Datatable of the Faktorenbogen Example for factor analysis

Usage

```
data(Faktorenbogen)
```

Format

A data frame with 150 observations in 14 variables

Details

Variables in the dataset:

- gender. a factor with levels `female` `male` `other`, giving the proband's gender
- age. a numeric vector of proband's age in years
- A. Item A of the questionnaire, numeric
- B. Item B of the questionnaire, numeric
- C. Item C of the questionnaire, numeric
- D. Item D of the questionnaire, numeric
- E. Item E of the questionnaire, numeric
- F. Item F of the questionnaire, numeric
- G. Item G of the questionnaire, numeric
- H. Item H of the questionnaire, numeric
- I. Item I of the questionnaire, numeric
- J. Item J of the questionnaire, numeric
- K. Item K of the questionnaire, numeric
- L. Item L of the questionnaire, numeric

Source

https://www.produnis.de/R/

`freqTable` *create a frequency table*

Description

returns a frequency table with absolute and relative frequencies and cumulated frequencies

Usage

```
freqTable(werte)
```

Arguments

`werte`	factor with obeserved data

Value

dataframe table

Examples

```
x <- ceiling(stats::rnorm(20))
freqTable(x)
```

`kenngroessen` *create a tibble with kenngroessen*

Description

returns a tibble with all kenngroessen

Usage

```
kenngroessen(werte)
```

Arguments

`werte`	numeric vector

Value

tibble with all kenngroessen

Examples

```
x <- ceiling(stats::rnorm(20))
kenngroessen(x)
```

KIbinomial_a *compute confidence intervall for binomial proportions*

Description

returns borders and length of confidence intervall for binomial proportions

Usage

```
KIbinomial_a(p, n, alpha)
```

Arguments

p	proportion obeserved
n	number of observations
alpha	error niveau

Value

confidence intervall

Examples

```
KIbinomial_a(0.35, 150, 0.05)
```

KIbinomial_u *compute confidence intervall for difference of binomial proportions*

Description

returns borders and length of confidence intervall for difference of binomial proportions

Usage

```
KIbinomial_u(p1, n1, p2, n2, alpha)
```

Arguments

p1	proportion obeserved in group 1
n1	number of observations in group 1
p2	proportion obeserved in group 2
n2	number of observations in group 2
alpha	error niveau

Value

confidence intervall

Examples

```
KIbinomial_u(0.25, 100, 0.4, 150, 0.05)
```

`KInormal_a` *compute confidence intervall for mean of normal distributed data*

Description

returns borders and length of confidence intervall for mean of normal distributed data

Usage

```
KInormal_a(xquer, s, n, alpha)
```

Arguments

`xquer`	mean of obeserved data
`s`	standard deviation of observed data
`n`	number of observations
`alpha`	error niveau

Value

confidence intervall

Examples

```
KInormal_a(400, 20, 100, 0.05)
```

KInormal_u *compute confidence intervall for mean of normal distributed data*

Description

returns a data.frame with borders and length of confidence intervall for mean of normal distributed data

Usage

```
KInormal_u(x1, s1, n1, x2, s2, n2, alpha)
```

Arguments

x1	mean of obeserved data in group 1
s1	standard deviation of observed data in group 1
n1	number of observations in group 1
x2	mean of obeserved data in group 2
s2	standard deviation of observed data in group 2
n2	number of observations in group 2
alpha	error niveau

Value

data.frame of confidence intervall

Examples

```
KInormal_u(2.22, 0.255, 13, 2.7, 0.306, 10 , 0.05)
```

lon.lat.osm *get longitude and altitude from an address using OpenStreetMap's API at http://nominatim.openstreetmap.org*

Description

get longitude and altitude from an address using OpenStreetMap's API at http://nominatim.openstreetmap.org

Usage

```
lon.lat.osm(address = NULL)
```

Arguments

`address`	a character of an address

Value

a data.frame containig "address", "lon", "lat"

Examples

```
lon.lat.osm("Eiffeltower")
```

MarioANOVA *Datatable of the SuperMario Example for Friedman-ANOVA*

Description

Datatable of the SuperMario Example for Friedman-ANOVA

Usage

```
data(MarioANOVA)
```

Format

A data frame with 47 observations in 8 variables

Details

Variables in the dataset:

- Name. The characters' name
- Alter. The characters' age in years
- Kingdom. The characters' home
- Geschlecht. The characters' gender (männlich = male, weiblich = female)
- BadGuy. Whether the character is a bad guy, logical
- t1. Measure at time 1
- t2. Measure at time 2
- t3. Measure at time 3

Source

https://www.produnis.de/R/

`Messwiederholung` *Datatable of the Messwiederholung Example for ANOVA*

Description

Datatable of the Messwiederholung Example for ANOVA

Usage

```
data(Messwiederholung)
```

Format

A data frame with 200 observations in 4 variables

Details

Variables in the dataset:

- Name. The first name of the probands.
- t1. Measure at time 1
- t2. Measure at time 2
- t3. Measure at time 3

Source

https://www.produnis.de/R/

`mma` *Dataset of a work sampling study*

Description

Dataset of a work sampling study

Usage

```
data(mma)
```

Format

A data frame with 9768 observations in 6 variables.

Details

Variables in the dataset:

- day. a vector, giving the number of the observation day
- time. a factor giving the time of observation
- ward. a factor giving the ward under observation
- qual. a factor giving the qualification of the nurse
- category. a factor of qualification categories
- action. a factor giving the observed action

Source

https://www.produnis.de/R/

`Nachtwachen` *Dataset of the German Nachtwachen study*

Description

Dataset of the German Nachtwachen study

Usage

```
data(Nachtwachen)
```

Format

A data frame with 276 observations in 37 variables.

Source

https://www.produnis.de/R/

nw *Dataset of the German Nachtwachen study with labelled variables*

Description

Dataset of the German Nachtwachen study, labelled version

Usage

```
data(nw)
```

Format

A data frame with 276 observations in 37 variables.

Source

https://www.produnis.de/R/

OrdinalSample *Datatable of an Ordinal Sample*

Description

Datatable of an Ordinal Sample

Usage

```
data(OrdinalSample)
```

Format

A data frame with 415 observations in 4 variables.

Details

Variables in the dataset:

- Konflikt. a numeric vector giving the potential of conflicts.
- Zufriedenh. a numeric vector giving the satisfaction of workers
- Geschlecht. a factor of proband's sex, 1 = male, 2=female
- Stimmung. an ordinal factor of proband's mood

Source

https://www.produnis.de/R/

`pairwise.chisq.test` *Pairwise Chi-Square Tests*

Description

This function performs pairwise Chi-Square tests for two factors.

Usage

```
pairwise.chisq.test(A, B, p.adjust.method = "bonferroni")
```

Arguments

`A`	A factor with two or moew levels. The first variable.
`B`	A factor with two or more levels. The second variable.
`p.adjust.method`	A string specifying the method for adjusting p-values. Default is "bonferroni".

Details

This function creates all possible pairs of levels of factor B and performs a Chi-Square test for each pair of B on variable A. The p-values are adjusted according to the specified method. #' This function is created for educational purposes only. For exact p-values, consider using `reporttools::pairwise.fisher.test()`.

Value

A data frame with the results of the pairwise Chi-Square tests. Includes the groups, Chi-Square statistic, degrees of freedom, p-values, adjusted p-values, and significance stars.

Examples

```
set.seed(123)
A <- factor(sample(c("Male", "Female"), 100, replace = TRUE))
B <- factor(sample(c("Location1", "Location2", "Location3"), 100, replace = TRUE))
pairwise.chisq.test(A, B, "holm")
```

`pf8` *Dataset of the PF8 example.*

Description

This is the dataset of the PF8 example.

Usage

```
data(pf8)
```

Format

A data frame with 731 observations in 16 variables.

Source

https://www.produnis.de/R/

Pflegeberufe *Matrix of Pflegeberufe by Isfort et al. 2018*

Description

Matrix of Pflegeberufe by Isfort et al. 2018

Usage

```
data(Pflegeberufe)
```

Format

A matrix with 9 cols (years) and 5 rows (nursing profession).

Author(s)

Isfort et al. 2018 (Pflegethermometer)

Source

https://www.produnis.de/R/

sens.spec *compute sensitivity and specifity*

Description

returns sensitivity specifity, negativ-predictive-value, postitiv-predictive-value

Usage

```
sens.spec(rp, rn, fp, fn)
```

Arguments

rp	number of true-positive (richtig-positiv)
rn	number of true-negative (richtig-negativ)
fp	number of false-positive (falsch-positiv)
fn	number of false-negative (falsch-negativ)

Value

a data.frame with sens, spec, ppw, npw

Examples

```
sens.spec(40, 17, 85, 4)
```

`ztrans` *z-Transformation by given numbers, with z = (x - mu) / sd*

Description

z-Transformation by given numbers, with z = (x - mu) / sd

Usage

```
ztrans(x, mu = 0, sd = 1)
```

Arguments

`x`	a value to transform
`mu`	the given mu
`sd`	the given standard deviation

Value

the z-transformed value

Examples

```
ztrans(120,mu=118,sd=20)
```

Index