Package ‘REdaS’

November 13, 2015

Type Package
Title Companion Package to the Book 'R: Einführung durch angewandte Statistik'
Version 0.9.3
Date 2015-11-12
Description Provides functions used in the 'R: Einführung durch angewandte Statistik' (second edition).
Depends R (>= 3.0.0), grid
Imports stats
Suggests car, cluster, foreign, GPArotation, MASS, mclust, psych, UsingR, vcd
License GPL-2
Encoding UTF-8
NeedsCompilation no
Author Marco Johannes Maier [cre, aut]
Maintainer Marco Johannes Maier <marco.maier@wu.ac.at>
Repository CRAN
Date/Publication 2015-11-13 11:47:40

R topics documented:

REdaS-package ................................................................. 2
Bartlett-Sphericity .......................................................... 2
Confidence-Intervals-for-Frequencies .................................. 5
Degrees-to-Radians-to-Degrees .......................................... 7
Density-Box-Plot ............................................................... 8
Kaiser-Meyer-Olkin-Statistics ........................................... 9
Odds-Ratios ................................................................. 11

Index 14


**Bartlett-Sphericity**

---

**REdaS-package**

**The REdaS Package**

**Description**

The **REdaS** Package provides functions used in the second edition of “R: Einführung durch angewandte Statistik”.

**Details**

<table>
<thead>
<tr>
<th>Package:</th>
<th>REdaS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>0.9.3</td>
</tr>
<tr>
<td>Date:</td>
<td>2015-11-12</td>
</tr>
<tr>
<td>License:</td>
<td>GPL-2</td>
</tr>
</tbody>
</table>

**Author(s)**

Autor and Maintainer: Marco J. Maier <marco.maier@wu.ac.at>

**References**


---

**Bartlett-Sphericity**

**Bartlett’s Test of Sphericity**

**Description**

Implements Bartlett’s Test of Sphericity which tests whether a matrix is significantly different from an identity matrix.

**Usage**

```r
bart_spher(x, use = c("everything", "all.obs", "complete.obs", "na.or.complete", "pairwise.complete.obs"))
```

```r
## S3 method for class 'bart_spher'
print(x, ...)
```
Bartlett-Sphericity

Arguments

- **x**: a data matrix or the object to be printed.
- **use**: defines the method to use if missing values are present (see Examples and cor).
- ... further arguments for the print method.

Details

The test statistic $X^2$ as defined in Eq. (3) in Bartlett (1951) is

$$\frac{1}{6}\left(-[(n-1)-(2k+5)/6]\log(|R|)\right)$$

where $n$ is the number of observations, $k$ the number of variables, and $R$ the correlation matrix of the data supplied in x. $|R|$ is the determinant of R.

Bartlett’s $X^2$ is asymptotically $\chi^2$-distributed with $df = k(k - 1)/2$ under the null hypothesis.

Note that, because the bias-corrected correlation matrix is used, $(n - 1)$ is employed instead of $n$, as in the paper.

**Treatment of Missing Values:**

If no missing values are present in the data matrix x, use will work with any setting and no adjustments are necessary. In this case, $n$ is the number of rows in x.

For listwise deletion (use = "complete.obs" or "na.or.complete"), $n$ is the number of remaining rows in x.

When use = "pairwise.complete.obs", $n$ is approximated as the sum of relative non-missing responses for all observations with 2 or more valid responses.

If listwise/pairwise methods are used to compute the correlation matrix and the test statistic, a warning will be issued when printing the object.

Value

A list object of class 'bart_spher'

- **call**: the issued function call
- **x**: the original data
- **cormat**: the correlation matrix computed from the data
- **use**: treatment of NAs
- **n**: the number of used observations
- **k**: the number of variables/items
- **X2**: the computed $X^2$ value
- **df**: degrees of freedom
- **p.value**: the $p$-value
- **warn**: logical value indicating whether a warning regarding missing values will be issued (see Details)

Author(s)

Marco J. Maier
References


See Also

cor() and KMOS()

Examples

```r
# generate a data frame with 3 variables and 100 observations
set.seed(5L)
datamatrix <- data.frame("A" = rnorm(100), "B" = rnorm(100), "C" = rnorm(100))
head(datamatrix)

# correlation matrix
cor(datamatrix)

# bartlett's test
bart_spher(datamatrix)

# effects of missing observations on correlations: to illustrate this, the first
# observation on variable A is set to NA
datamatrix[1, 1] <- NA
head(datamatrix)

# "everything" (the default) causes all correlations involving a variable with
# missing values to be NA (in this case, all pairwise correlations with the
# variable "A")
cor(datamatrix)

# "all.obs" generates an error if missing values are present.
## Not run:
cor(datamatrix, use = "all.obs")
## End(Not run)

# "complete.obs" and "na.or.complete" delete complete observations if there are
# NA (in this case, the first case would be deleted). If there are no complete
# cases left after the listwise deletion, "complete.obs" results in an error
# while "na.or.complete" returns a matrix with all elements being NA.
cor(datamatrix, use = "complete.obs")
cor(datamatrix, use = "na.or.complete")

# "pairwise.complete.obs" uses all non-missing pairwise values. If there are no
# non-missing value pairs in two variables, the results will be NA.
# It is possible that correlation matrices are not positive semi-definite.
cor(datamatrix, use = "pairwise.complete.obs")

# with the missing value in the first cell, the test does not work anymore:
```
Confidence-Intervals-for-Frequencies

Confidence Intervals for Relative Frequencies

Description

This function computes (one or more) confidence intervals (CIs) for a vector of observations or a table object and returns an object of class 'freqCI' to draw a bar plot of the results.

Usage

freqCI(x, level = 0.95)

## S3 method for class 'freqCI'
print(x, percent = TRUE, digits, ...)

## S3 method for class 'freqCI'
barplot(height, percent = TRUE, ...)

Arguments

x must either be a numeric or factor object of individual observations (character vectors are also accepted, but a warning is issued) or an object of class 'table' of frequencies (produced using table or as.table)

level a numeric vector of confidence levels in (0, 1).

percent if TRUE, all values are printed as percentages, else relative frequencies are printed.

digits the number of digits to print (default to 2 if values are represented as percents or 4 if relative frequencies are used.

height to plot the proportions and confidence intervals, an object of class 'freqCI' must be used with the generic barplot function.

... further arguments.

Details

ref to the book
Value

freqCI() returns an object of class ‘freqCI’ as a list:

call the function call issued
x the original object
level the confidence levels
freq a numeric vector of frequencies
n the number of observations
rel_freq relative frequencies
cat_names category names
CIs_low lower confidence interval boundary/boundaries
CIs_high upper confidence interval boundary/boundaries

print.freqCI() invisibly returns a matrix with the confidence intervals and estimates.

barplot.freqCI() invisibly returns a vector with the $x$-coordinates of the plotted bars.

Author(s)

Marco J. Maier

See Also

table, as.table, barplot

Examples

# generate some simple data using rep() and inspect them using table()
mydata <- rep(letters[1:3], c(100, 200, 300))
table(mydata)
100 * prop.table(table(mydata))

# compute 95% and 99% confidence intervals and print them with standard settings
res <- freqCI(mydata, level = c(.95, .99))
res

# print the result as relative frequencies rounded to 3 digits, save the result
# and print the invisibly returned matrix
resmat <- print(res, percent = FALSE, digits = 3)
resmat

# plot the results and save the $x$-coordinates
x_coo <- barplot(res)
x_coo

# use the $x$-coordinates to plot the frequencies per category
text(x_coo, 0, labels = paste0("n = ", res$freq), pos = 3)
Degrees-to-Radians-to-Degrees

Conversion between Radians and Degrees

Description

Converts radians to degrees and vice versa.

Usage

deg2rad(d)

rad2deg(r)

Arguments

d  degrees
r  radians

Details

Since $\pi \text{ rad} = 180^\circ$, degrees ($d$) can be converted to radians ($r$) using $r = d \cdot \pi/180$ and the conversion of radians to degrees is $d = r \cdot 180/\pi$.

Author(s)

Marco J. Maier

See Also

see Trigonometric Functions, Hyperbolic Functions, Constants in R

Examples

# pi is available as a constant
pi

# 180^\circ are pi radians
deg2rad(180)

# 2 * pi radians are 360^\circ
rad2deg(2 * pi)
Density-Box-Plot

Description

This function draws a (grouped) boxplot-like plot with kernel density estimators.

Usage

densbox(formula, data, rug = FALSE, from, to, gsep = .5, kernel, bw, main, ylab, var_names, box_out = TRUE, horizontal = FALSE, ...)

Arguments

- formula: a formula object that references elements in data, see Details
- data: a data frame containing the variables specified in formula
- rug: a logical value to add a rug to the individual density-boxes
- from: an optional lower boundary for the kernel density estimation (see `density`)
- to: an optional upper boundary for the kernel density estimation (see `density`)
- gsep: a numeric value ≥ 0 that specifies the length of group separation if two or more grouping variables are used
- kernel: a string specifying the type of the kernel (default: "gaussian", see `density`)
- bw: the bandwidth for kernel density estimation (see `density`)
- main: a character object for the title
- ylab: a character object for the y-axis label
- var_names: a character object to print grouping variables' names in the lower left margin – grouping variables are treated in the order they are given in the formula
- box_out: if TRUE, outliers treated as in standard boxplots (plotted as stars outside the boxplot’s whiskers; default), if FALSE, outliers are not treated differently, i.e., minimum and maximum will be over the full range, no matter how far individual observations may be from the median with respect to the IQR (interquartile range; see `boxplot.stats` and `fivenum` for details on the computation of box-plot statistics).
- horizontal: not implemented yet...
- ...: further arguments, see Details

Details

This function plots a combination of boxplots and kernel density plots to get a more informative graphic of a metric dependent variable with respect to grouped data. The central element is the formula argument that defines the dependent variable (DV) and grouping variables (independent variables, IV). For a meaningful plot, the IVs should be categorical variables (they are treated as factors).
In the simplest case, there is no grouping, so formula is $DV \sim 1$. As grouping variables are added, the plot will be split up accordingly. Note that the ordering of IVs in the formula defines how the plot is split up – the first variable is the most general grouping, the second will form subgroups in the first variable’s groups and so on ...

If there are cases where a level of a factor is completely missing ab initio, the level will be dropped. Subgroups with less than 5 observations will be dropped and “< 5” will be plotted instead.

Author(s)
Marco J. Maier

See Also
density, boxplot, grid (Package)

Examples

# plot a density-box-plot of one (log-normal) variable
set.seed(5L)
data1 <- rlnorm(100, 1, .5)
densbox(data1 ~ 1, from = 0, rug = TRUE)

# plots a continuous variable in (0, 1) with 2 grouping variables
data2 <- data.frame(y = rnorm(400, rep(c(0, 1, -1, 0), each = 100), 1),
                     x1 = rep(c("A", "B"), each = 200),
                     x2 = rep(c("X", "Y", "X", "Y"), each = 100))
with(data2, tapply(y, list(x1, x2), mean))

# a density-box-plot of the data with the kernel density
# estimator constrained to the interval 0 to 1
densbox(y ~ x2 + x1, data2, main = "Plot with some\nSpecials",
        var_names = c("Second\nVariable", "First Variable"))

# the same plot with a rug and ignoring outliers in the boxplot
densbox(y ~ x2 + x1, data2, rug = TRUE, box_out = FALSE)

# density-box-plot with the same data, but no additional space between groups
# by setting gsep = 0.
# the kernel density plots have a rectangular kernel with a bandwidth of 0.25
# which results in a "jagged" appearance.
densbox(y ~ x2 + x1, data2, gsep = 0, kernel = "rectangular", bw = 0.25)

Kaiser-Meyer-Olkin-Statistics

Kaiser-Meyer-Olkin Statistics

Description

description
Usage

```r
KMOS(x, use = c("everything", "all.obs", "complete.obs", "na.or.complete", "pairwise.complete.obs"))
```

```r
## S3 method for class 'MSA_KMO'
print(x, stats = c("both", "MSA", "KMO"), vars = "all",
       sort = FALSE, show = "all", digits = getOption("digits"), ...)
```

Arguments

- `x`: The data `X` for `KMOS()`, an object of class 'MSA_KMO' for the `print` method.
- `use`: defines the method to use if missing values are present (for a detailed explanation see `bart_spher`; see also `cor`).
- `stats`: determines if "MSA", "KMO" or "both" (default) are printed.
- `vars`: can be "all" or a vector of index numbers of variables to print the MSAs for.
- `sort`: sorts the MSAs in increasing order.
- `show`: shows the specified number of variables (from 1 to the number of potentially sorted variables).
- `digits`: the number of decimal places to print.
- `...`: further arguments.

Details

The Measure of Sampling Adequacy (MSA) for individual items and the Kaiser-Meyer-Olkin (KMO) Criterion rely on the Anti-Image-Correlation Matrix `A` (for details see Kaiser & Rice, 1974) that contains all bivariate partial correlations given all other items in the $a_{ij} = r_{ij|X\setminus\{i,j\}}$ which is:

\[
A = \frac{1}{2} R^{-1} \left( \text{diag}(R^{-1}) \right)^{-1/2}
\]

where `R` is the correlation matrix, based on the data `X`.

The KMO and MSAs for individual items are (adapted from Equations (3) and (4) in Kaiser & Rice, 1974; note that $a$ is $q$ in the article):

\[
KMO = \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} r_{ij}^2}{\sum_{i=1}^{k} \sum_{j=1}^{k} r_{ij}^2 + a_{ij}^2}, \quad i \neq j
\]

\[
MSA_i = \frac{\sum_{j=1}^{k} r_{ij}^2}{\sum_{j=1}^{k} r_{ij}^2 + a_{ij}^2}, \quad j \neq i
\]

Historically, as suggested in Kaiser (1974) and Kaiser & Rice (1974), a rule of thumb for those values is:

- $\geq .9$ marvelous
- [.8, .9) meritorious
- [.7, .8) middling
- [.6, .7) mediocre
- [.5, .6) miserable
- $< .5$ unacceptable
Odds-Ratios

Value
A list of class 'MSA_KMO'
- call: the issued function call
- cormat: correlation matrix
- pcormat: normalized negative inverse of the correlation matrix (pairwise correlations given all other variables)
- n: the number of observations
- k: the number of variables/items
- MSA: measure of sampling adequacy
- KMO: Kaiser-Meyer-Olkin criterion

Author(s)
Marco J. Maier

References

See Also
cor, bart_spher

Examples
set.seed(5L)
daten <- data.frame("A"=rnorm(100), "B"=rnorm(100), "C"=rnorm(100), "D"=rnorm(100), "E"=rnorm(100))
cor(daten)
KMO(daten, use = "pairwise.complete.obs")

Odds-Ratios

Compute (Log) Odds Ratios

Description
This function computes the (log-)odds ratio (OR) for a 2 × 2 table (x must be an object of class 'table' either by using table or as.table). For a data frame of k variables with 2 categories each, all k(k - 1)/2 pairwise (log-)odds-ratios are computed.
Usage

odds_ratios(x)

## S3 method for class 'REdaS_ORs'
print(x, ...)

## S3 method for class 'REdaS_ORs'
summary(object, ...)

Arguments

x

either a $2 \times 2$ table object or a data frame where each variable has two categories.

object

an object of class 'REdaS_ORs'.

... further arguments.

Details

Note that tables where one or more cells are 0 are not processed and a warning is issued in such cases.

Value

odds_ratios() returns a list of class 'REdaS_ORs':

call
the issued function call.

x
the original data.

tables
a list of one or more tables.

comps
a list of the compared variables' names.

ORs
a list with (log-)odds-ratios, standard errors, $z$- and $p$-values.

print.REdaS_ORs() invisibly returns a matrix containing all statistics shown by the print-method.

Author(s)

Marco J. Maier

Examples

# create a table from a 2 x 2 matrix of frequencies using as.table()

tab <- as.table(matrix(c(49, 1, 5, 45), 2))
dimnames(tab) <- list("LED on?" = c("no", "yes"),
                   "PC running?" = c("no", "yes"))

print(tab)

odds_ratios(tab)

# generate a matrix with 3 variables and 100 observations
# note that each variable must have exactly two categories

```r
set.seed(5)
x <- data.frame("A" = as.factor(sample(1:2, 100, TRUE)),
"B" = as.factor(sample(3:4, 100, TRUE)),
"C" = as.factor(sample(5:6, 100, TRUE)))
head(x)

res <- odds_ratios(x)

# print the results and save the summarized information in a matrix
resmat <- print(res)
resmat

# the summary method gives a rather lengthy output with all tables etc.
summary(res)
```
Index

*Topic **package**
REdaS-package, 2

as.table, 5, 6, 11

barplot, 6
barplot.freqCI
   (Confidence-Intervals-for-Frequencies), 5
bart_spher, 10, 11
bart_spher (Bartlett-Sphericity), 2
Bartlett-Sphericity, 2
boxplot, 9
boxplot.stats, 8

Confidence-Intervals-for-Frequencies, 5

Constants in R, 7
cor, 3, 10, 11
cor(), 4

deg2rad
   (Degrees-to-Radians-to-Degrees), 7

Degrees-to-Radians-to-Degrees, 7
densbox (Density-Box-Plot), 8
density, 8, 9
Density-Box-Plot, 8

fivenum, 8
freqCI
   (Confidence-Intervals-for-Frequencies), 5

grid, 9

Hyperbolic Functions, 7

Kaiser-Meyer-Olkin-Statistics, 9
KMOS (Kaiser-Meyer-Olkin-Statistics), 9
KMOS(), 4

Odds-Ratios, 11
odds_ratios (Odds-Ratios), 11

print.bart_spher (Bartlett-Sphericity),
   2
print.freqCI
   (Confidence-Intervals-for-Frequencies), 5
print.MSAMPKMO
   (Kaiser-Meyer-Olkin-Statistics), 9
print.REdaS_ORS (Odds-Ratios), 11

rad2deg
   (Degrees-to-Radians-to-Degrees), 7

REdaS (REdaS-package), 2
REdaS-package, 2

summary.REdaS_ORS (Odds-Ratios), 11

table, 5, 6, 11

Trigonometric Functions, 7