

# Package ‘REdaS’

May 7, 2026

**Type** Package

**Title** Companion Package to the Book 'R: Einführung durch angewandte Statistik'

**Version** 0.9.4

**Date** 2022-06-11

**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@posteo.de>

**Repository** CRAN

**Date/Publication** 2022-06-13 08:36:45 UTC

## Contents

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 . . . . .	12

<b>Index</b>	<b>14</b>
--------------	-----------

---

REdaS-package

*The REdaS Package*

---

### Description

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

### Details

Package: REdaS  
Type: Package  
Version: 0.9.4  
Date: 2022-06-11  
License: GPL-2

### Author(s)

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

### References

Hatzinger, R., Hornik, K., Nagel, H., & Maier, M. J. (2014). *R: Einführung durch angewandte Statistik*. München: Pearson Studium.

---

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

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

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

**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 <code>cor</code> ).
...	further arguments for the print method.

**Details**

The test statistic  $X^2$  as defined in Eq. (3) in Bartlett (1951) is  $X^2 = -[(n-1)-(2k+5)/6] \cdot \log(|\mathbf{R}|)$  where  $n$  is the number of observations,  $k$  the number of variables, and  $\mathbf{R}$  the correlation matrix of the data supplied in `x`.  $|\mathbf{R}|$  is the determinant of  $\mathbf{R}$ .

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'`

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

**Author(s)**

Marco J. Maier

## References

Bartlett, M. S. (1951). The Effect of Standardization on a  $\chi^2$  Approximation in Factor Analysis. *Biometrika* 38(3/4), 337–344.

## See Also

[cor\(\)](#) and [KMOS\(\)](#)

## Examples

```
# 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:
```

```
## Not run:
bart_spher(datamatrix)
## End(Not run)

# deleting the whole first observation (listwise) gives
bart_spher(datamatrix, use = "na.or.complete")

# using pairwise-correlation, the result is
bart_spher(datamatrix, use = "pairwise.complete.obs")
```

---

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 <i>individual observations</i> (character vectors are also accepted, but a warning is issued) or an object of class 'table' of frequencies (produced using <a href="#">table</a> or <a href="#">as.table</a> )
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:

<code>call</code>	the function call issued
<code>x</code>	the original object
<code>level</code>	the confidence levels
<code>freq</code>	a numeric vector of frequencies
<code>n</code>	the number of observations
<code>rel_freq</code>	relative frequencies
<code>cat_names</code>	category names
<code>CIs_low</code>	lower confidence interval boundary/boundaries
<code>CIs_high</code>	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° are pi radians  
deg2rad(180)
```

```
# 2 * pi radians are 360°  
rad2deg(2 * pi)
```

---

 Density-Box-Plot      *Density-Box-Plots*


---

## Description

This function draws a (grouped) boxplot-like plot with 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 <a href="#">density</a> )
to	an optional upper boundary for the kernel density estimation (see <a href="#">density</a> )
gsep	a numeric value $\geq 0$ 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 <a href="#">density</a> )
bw	the bandwidth for kernel density estimation (see <a href="#">density</a> )
main	a character object for the title
ylab	a character object for the <i>y</i> -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 ( <i>interquartile range</i> ; see <a href="#">boxplot.stats</a> and <a href="#">fivenum</a> for details on the computation of boxplot 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**

```

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

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

```

**Arguments**

<code>x</code>	The data $\mathbf{X}$ for <code>KMOS()</code> , an object of class 'MSA_KMO' for the print method.
<code>use</code>	defines the method to use if missing values are present (for a detailed explanation see <a href="#">bart_spher</a> ; see also <a href="#">cor</a> ).
<code>stats</code>	determines if "MSA", "KMO" or "both" (default) are printed.
<code>vars</code>	can be "all" or a vector of index numbers of variables to print the MSAs for.
<code>sort</code>	sorts the MSAs in increasing order.
<code>show</code>	shows the specified number of variables (from 1 to the number of <i>potentially sorted</i> variables).
<code>digits</code>	the number of decimal places to print.
<code>...</code>	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*  $\mathbf{A}$  (for details see Kaiser & Rice, 1974) that contains all bivariate partial correlations given all other items in the  $a_{ij} = r_{ij|\mathbf{X}\setminus\{i,j\}}$  which is:

$$\mathbf{A} = [\text{diag}(\mathbf{R}^{-1})]^{-1/2} \mathbf{R}^{-1} [\text{diag}(\mathbf{R}^{-1})]^{-1/2}$$

where  $\mathbf{R}$  is the correlation matrix, based on the data  $\mathbf{X}$ .

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

## 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

- Kaiser, H. F. (1970). A Second Generation Little Jiffy. *Psychometrika*, 35(4), 401–415.
- Kaiser, H. F. (1974). An Index of Factorial Simplicity. *Psychometrika*, 39(1), 31–36.
- Kaiser, H. F., & Rice, J. (1974). Little Jiffy, Mark IV. *Educational and Psychological Measurement*, 34, 111–117.

## 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)
KMOS(daten, use = "pairwise.complete.obs")
```

**Description**

This function computes the (log-)odds ratio (OR) for a  $2 \times 2$  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**

<code>x</code>	either a $2 \times 2$ table object or a data frame where each variable has two categories.
<code>object</code>	an object of class 'REdaS_ORs'.
<code>...</code>	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':

<code>call</code>	the issued function call.
<code>x</code>	the original data.
<code>tables</code>	a list of one or more tables.
<code>comps</code>	a list of the compared variables' names.
<code>ORs</code>	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"))

tab

odds_ratios(tab)

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

\* **package**  
REdaS-package, 2

as.table, 5, 6, 12

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, 12  
odds\_ratios (Odds-Ratios), 12

print.bart\_spher (Bartlett-Sphericity), 2  
print.freqCI  
(Confidence-Intervals-for-Frequencies), 5  
print.MSA\_KMO  
(Kaiser-Meyer-Olkin-Statistics), 9  
print.REdaS\_ORs (Odds-Ratios), 12

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

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

summary.REdaS\_ORs (Odds-Ratios), 12

table, 5, 6, 12  
Trigonometric Functions, 7