Package ‘jmv’

February 17, 2020

Type Package
Title The ‘jamovi’ Analyses
Version 1.2.5
Date 2020-02-16
Author Ravi Selker, Jonathon Love, Damian Dropmann
Maintainer Jonathon Love <jon@thon.cc>
Description A suite of common statistical methods such as descriptives, t-tests, ANOVAs, regression, correlation matrices, proportion tests, contingency tables, and factor analysis. This package is also useable from the ‘jamovi’ statistical spreadsheet (see <https://www.jamovi.org> for more information).
License GPL (>= 2)
Depends R (>= 3.2)
Imports jmvcore (>= 1.0.8), R6, car (>= 3.0.0), multcomp, ggplot2 (>= 2.2.1), PMCMR, emmeans (>= 1.1.3), vcd, vcdExtra, GGally, BayesFactor, psych (>= 1.7.5), GPArotation, afex (>= 0.20.2), mvnormtest, lavaan, ggridges, ROCR, nnet, MASS
Suggests exact2x2, testthat, semPlot, carData, knitr
Encoding UTF-8
RoxygenNote 6.1.1
VignetteBuilder knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2020-02-17 16:40:14 UTC

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ancova</td>
<td>2</td>
</tr>
<tr>
<td>ANOVA</td>
<td>4</td>
</tr>
<tr>
<td>anovaNP</td>
<td>7</td>
</tr>
</tbody>
</table>
ancova

Description

The Analysis of Covariance (ANCOVA) is used to explore the relationship between a continuous dependent variable, one or more categorical explanatory variables, and one or more continuous explanatory variables (or covariates). It is essentially the same analysis as ANOVA, but with the addition of covariates.

Usage

ancova(data, dep, factors = NULL, covs = NULL, effectSize = NULL, modelTerms = NULL, ss = "3", homo = FALSE, norm = FALSE, qq = FALSE, contrasts = NULL, postHoc = NULL, postHocCorr = list("tukey"), postHocES = list(), emMeans = list(list()), emmPlots = TRUE, emmPlotData = FALSE, emmPlotError = "ci", emmTables = FALSE, emmWeights = TRUE, ciWidthEmm = 95, formula)
Arguments

- **data**: the data as a data frame
- **dep**: the dependent variable from data, variable must be numeric (not necessary when providing a formula, see examples)
- **factors**: the explanatory factors in data (not necessary when providing a formula, see examples)
- **covs**: the explanatory covariates (not necessary when providing a formula, see examples)
- **effectSize**: one or more of 'eta', 'partEta', or 'omega'; use eta², partial eta², and omega² effect sizes, respectively
- **modelTerms**: a formula describing the terms to go into the model (not necessary when providing a formula, see examples)
- **ss**: '1', '2' or '3' (default), the sum of squares to use
- **homo**: TRUE or FALSE (default), perform homogeneity tests
- **norm**: TRUE or FALSE (default), perform Shapiro-Wilk tests of normality
- **qq**: TRUE or FALSE (default), provide a Q-Q plot of residuals
- **contrasts**: a list of lists specifying the factor and type of contrast to use, one of 'deviation', 'simple', 'difference', 'helmert', 'repeated' or 'polynomial'
- **postHoc**: a formula containing the terms to perform post-hoc tests on (see the examples)
- **postHocCorr**: one or more of 'none', 'tukey', 'scheffe', 'bonf', or 'holm'; provide no, Tukey, Scheffe, Bonferroni, and Holm Post Hoc corrections respectively
- **postHocES**: a possible value of 'd'; provide cohen's d measure of effect size for the post-hoc tests
- **emMeans**: a formula containing the terms to estimate marginal means for (see the examples)
- **emmPlots**: TRUE (default) or FALSE, provide estimated marginal means plots
- **emmPlotData**: TRUE or FALSE (default), plot the data on top of the marginal means
- **emmPlotError**: 'none', 'ci' (default), or 'se'. Use no error bars, use confidence intervals, or use standard errors on the marginal mean plots, respectively
- **emmTables**: TRUE (default) or FALSE, provide estimated marginal means tables
- **emmWeights**: TRUE (default) or FALSE, weigh each cell equally or weight them according to the cell frequency
- **ciWidthEmm**: a number between 50 and 99.9 (default: 95) specifying the confidence interval width for the estimated marginal means
- **formula**: (optional) the formula to use, see the examples

Value

A results object containing:

- `results$main`: a table of ANCOVA results
- `results$model`: The underlying aov object
ANOV A

results$assump$homo a table of homogeneity tests
results$assump$norm a table of normality tests
results$assump$qq a q-q plot
results$confrasts an array of contrasts tables
results$postHoc an array of post-hoc tables
results$emm an array of the estimated marginal means plots + tables

Tables can be converted to data frames with asDF or as.data.frame. For example:

results$main$asDF
as.data.frame(results$main)

Examples

data('ToothGrowth')

ancova(formula = len ~ supp + dose, data = ToothGrowth)

# ANCOVA
# -----------------------------------------------
# Sum of Squares  df  Mean Square  F    p
# -----------------------------------------------
# supp           205  1    205.4    11.4 0.001
# dose          2224  1  2224.3  124.0 < .001
# Residuals    1023  57    17.9
# -----------------------------------------------

ancova(
    formula = len ~ supp + dose,
data = ToothGrowth,
    postHoc = ~ supp,
    emMeans = ~ supp)

Description

The Analysis of Variance (ANOVA) is used to explore the relationship between a continuous dependent variable, and one or more categorical explanatory variables.
Usage

ANOVA(data, dep, factors = NULL, effectSize = NULL, 
modelTerms = NULL, ss = "3", homo = FALSE, norm = FALSE, 
qq = FALSE, contrasts = NULL, postHoc = NULL, 
postHocCorr = list("tukey"), postHocES = list(), 
emmMeans = list(list()), emmPlots = TRUE, emmPlotData = FALSE, 
emmPlotError = "ci", emmTables = FALSE, emmWeights = TRUE, 
ciWidthEmm = 95, formula)

Arguments

data the data as a data frame

dep the dependent variable from data, variable must be numeric (not necessary when providing a formula, see examples)

factors the explanatory factors in data (not necessary when providing a formula, see examples)

effectSize one or more of 'eta', 'partEta', or 'omega'; use eta², partial eta², and omega² effect sizes, respectively

modelTerms a formula describing the terms to go into the model (not necessary when providing a formula, see examples)

ss '1', '2' or '3' (default), the sum of squares to use

homo TRUE or FALSE (default), perform homogeneity tests

norm TRUE or FALSE (default), perform Shapiro-Wilk tests of normality

qq TRUE or FALSE (default), provide a Q-Q plot of residuals

contrasts a list of lists specifying the factor and type of contrast to use, one of 'deviation', 'simple', 'difference', 'helmert', 'repeated' or 'polynomial'

postHoc a formula containing the terms to perform post-hoc tests on (see the examples)

postHocCorr one or more of 'none', 'tukey', 'scheffe', 'bonf', or 'holm'; provide no, Tukey, Scheffe, Bonferroni, and Holm Post Hoc corrections respectively

postHocES a possible value of 'd'; provide cohen’s d measure of effect size for the post-hoc tests

emmMeans a formula containing the terms to estimate marginal means for (see the examples)

emmPlots TRUE (default) or FALSE, provide estimated marginal means plots

emmPlotData TRUE or FALSE (default), plot the data on top of the marginal means

emmPlotError 'none', 'ci' (default), or 'se'. Use no error bars, use confidence intervals, or use standard errors on the marginal mean plots, respectively

emmTables TRUE or FALSE (default), provide estimated marginal means tables

emmWeights TRUE (default) or FALSE, weigh each cell equally or weigh them according to the cell frequency

ciWidthEmm a number between 50 and 99.9 (default: 95) specifying the confidence interval width for the estimated marginal means

formula (optional) the formula to use, see the examples
Details

ANOVA assumes that the residuals are normally distributed, and that the variances of all groups are equal. If one is unwilling to assume that the variances are equal, then a Welch’s test can be used instead (However, the Welch’s test does not support more than one explanatory factor). Alternatively, if one is unwilling to assume that the data is normally distributed, a non-parametric approach (such as Kruskal-Wallis) can be used.

Value

A results object containing:

- `results$main`: a table of ANOVA results
- `results$model`: The underlying `aov` object
- `results$assump$homo`: a table of homogeneity tests
- `results$assump$norm`: a table of normality tests
- `results$assump$qq`: a q-q plot
- `results$contrasts`: an array of contrasts tables
- `results$postHoc`: an array of post-hoc tables
- `results$emm`: an array of the estimated marginal means plots + tables

Tables can be converted to data frames with `asDF` or `as.data.frame`. For example:

```r
results$main$asDF
as.data.frame(results$main)
```

Examples

data('ToothGrowth')

ANOVA(formula = len ~ dose * supp, data = ToothGrowth)

# ANOVA
# ANOVA
# ANOVA
# # Sum of Squares df Mean Square F p
# # -----------------------------------------------------
# # dose 2426 2 1213.2 92.00 < .001
# # supp 205 1 205.4 15.57 < .001
# # dose:supp 108 2 54.2 4.11 0.022
# # Residuals 712 54 13.2
# # -----------------------------------------------------
# # ANOVA(formula = len ~ dose * supp, data = ToothGrowth,
# # emMeans = ~ supp + dose:supp, # est. marginal means for supp and dose:supp
anovaNP

emmmPlots = TRUE,  # produce plots of those marginal means
emmmTables = TRUE)  # produce tables of those marginal means

Description

The Kruskal-Wallis test is used to explore the relationship between a continuous dependent variable, and a categorical explanatory variable. It is analogous to ANOVA, but with the advantage of being non-parametric and having fewer assumptions. However, it has the limitation that it can only test a single explanatory variable at a time.

Usage

anovaNP(data, deps, group, es = FALSE, pairs = FALSE, formula)

Arguments

data the data as a data frame
deps a string naming the dependent variable in data
group a string naming the grouping or independent variable in data
es TRUE or FALSE (default), provide effect-sizes
pairs TRUE or FALSE (default), perform pairwise comparisons
formula (optional) the formula to use, see the examples

Value

A results object containing:

results$table a table of the test results
results$comparisons an array of pairwise comparison tables

Tables can be converted to data frames with asDF or as.data.frame. For example:

results$table$asDF
as.data.frame(results$table)

Examples

data('ToothGrowth')

anovaNP(formula = len ~ dose, data = ToothGrowth)
# ONE-WAY ANOVA (NON-PARAMETRIC)

# Kruskal-Wallis

---

<table>
<thead>
<tr>
<th>X²</th>
<th>df</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

anovaOneW

---

**One-Way ANOVA**

Description

The Analysis of Variance (ANOVA) is used to explore the relationship between a continuous dependent variable, and one or more categorical explanatory variables. This 'One-Way ANOVA' is a simplified version of the 'normal' ANOVA, allowing only a single explanatory factor, however also providing a Welch's ANOVA. The Welch's ANOVA has the advantage that it need not assume that the variances of all groups are equal.

Usage

```r
anovaOneW(data, deps, group, welchs = TRUE, fishers = FALSE, miss = "perAnalysis", desc = FALSE, descPlot = FALSE, norm = FALSE, qq = FALSE, eqv = FALSE, phMethod = "none", phMeanDif = TRUE, phSig = TRUE, phTest = FALSE, phFlag = FALSE, formula)
```

Arguments

- **data** the data as a data frame
- **deps** a string naming the dependent variables in data
- **group** a string naming the grouping or independent variable in data
- **welchs** TRUE (default) or FALSE, perform Welch's one-way ANOVA which does not assume equal variances
- **fishers** TRUE or FALSE (default), perform Fisher's one-way ANOVA which assumes equal variances
- **miss** 'perAnalysis' or 'listwise', how to handle missing values; 'perAnalysis' excludes missing values for individual dependent variables, 'listwise' excludes a row from all analyses if one of its entries is missing.
- **desc** TRUE or FALSE (default), provide descriptive statistics
- **descPlot** TRUE or FALSE (default), provide descriptive plots
**anovaOneW**

- **norm** TRUE or FALSE (default), perform Shapiro-Wilk test of normality
- **qq** TRUE or FALSE (default), provide a Q-Q plot of residuals
- **eqv** TRUE or FALSE (default), perform Levene’s test for equality of variances
- **phMethod** 'none', 'gamesHowell' or 'tukey', which post-hoc tests to provide; 'none' shows no post-hoc tests, 'gamesHowell' shows Games-Howell post-hoc tests where no equivalence of variances is assumed, and 'tukey' shows Tukey post-hoc tests where equivalence of variances is assumed
- **phMeanDif** TRUE (default) or FALSE, provide mean differences for post-hoc tests
- **phSig** TRUE (default) or FALSE, provide significance levels for post-hoc tests
- **phTest** TRUE or FALSE (default), provide test results (t-value and degrees of freedom) for post-hoc tests
- **phFlag** TRUE or FALSE (default), flag significant post-hoc comparisons
- **formula** (optional) the formula to use, see the examples

**Details**

For convenience, this method allows specifying multiple dependent variables, resulting in multiple independent tests.

Note that the Welch’s ANOVA is the same procedure as the Welch’s independent samples t-test.

**Value**

A results object containing:

- `results$anova` a table of the test results
- `results$desc` a table containing the group descriptives
- `results$assump$norm` a table containing the normality tests
- `results$assump$eqv` a table of equality of variances tests
- `results$plots` an array of groups of plots
- `results$postHoc` an array of post-hoc tables

Tables can be converted to data frames with `asDF` or `as.data.frame`. For example:

```r
 results$anova$asDF
```

```r
 as.data.frame(results$anova)
```

**Examples**

```r
 data('ToothGrowth')
 dat <- ToothGrowth
 dat$dose <- factor(dat$dose)

 anovaOneW(formula = len ~ dose, data = dat)
```

# # ONE-WAY ANOVA
anovaRM

Repeated Measures ANOVA

Description

The Repeated Measures ANOVA is used to explore the relationship between a continuous dependent variable and one or more categorical explanatory variables, where one or more of the explanatory variables are 'within subjects' (where multiple measurements are from the same subject). Additionally, this analysis allows the inclusion of covariates, allowing for repeated measures ANCOVAs as well.

Usage

anovaRM(data, rm = list(list(label = "RM Factor 1", levels = list("Level 1", "Level 2"))), rmCells = NULL, bs = NULL, cov = NULL, effectSize = NULL, depLabel = "Dependent", rmTerms = NULL, bsTerms = NULL, ss = "3", spherTests = FALSE, spherCorr = list("none"), leveneTest = FALSE, contrasts = NULL, postHoc = NULL, postHocCorr = list("tukey"), emMeans = list(list()), emmPlots = TRUE, emmTables = FALSE, emmWeights = TRUE, ciWidthEmm = 95, emmPlotData = FALSE, emmPlotError = "ci", groupSumm = FALSE)

Arguments

data  the data as a data frame
rm  a list of lists, where each list describes the label (as a string) and the levels (as vector of strings) of a particular repeated measures factor
rmCells  a list of lists, where each list describes a repeated measure (as a string) from data defined as measure and the particular combination of levels from rm that it belongs to (as a vector of strings) defined as cell
bs  a vector of strings naming the between subjects factors from data
cov  a vector of strings naming the covariates from data. Variables must be numeric
effectSize  one or more of 'eta', 'partEta', or 'omega'; use eta^2, partial eta^2, and omega^2 effect sizes, respectively
**Details**

This analysis requires that the data be in ‘wide format’, where each row represents a subject (as opposed to long format, where each measurement of the dependent variable is represented as a row).

A non-parametric equivalent of the repeated measures ANOVA also exists; the Friedman test. However, it has the limitation of only being able to test a single factor.

**Value**

A results object containing:

```
results$rmTable a table
results$bsTable a table
```
results$assump$spherTable  a table
results$assump$leveneTable  a table
results$contrasts  an array of tables
results$postHoc  an array of tables
results$emm  an array of the estimated marginal means plots + tables
results$groupSummary  a summary of the groups

Tables can be converted to data frames with asDF or as.data.frame. For example:

results$rmTable$asDF
as.data.frame(results$rmTable)

Examples

## Not run:
data('bugs', package = 'jmv')

anovaRM(
data = bugs,
rm = list(
  list(
    label = 'Frightening',
    levels = c('Low', 'High'))),
rmCells = list(
  list(
    measure = 'LDLF',
    cell = 'Low'),
  list(
    measure = 'LDHF',
    cell = 'High'))),
rmTerms = list('Frightening'))

#  # REPEATED MEASURES ANOVA
#  # Within Subjects Effects
#  # -----------------------------------------------
#  # Sum of Squares  df  Mean Square  F  p
#  # -----------------------------------------------
#  # Frightening  126  1  126.11  44.2  < .001
#  # Residual  257  90  2.85
#  # -----------------------------------------------
#  # Note. Type 3 Sums of Squares
#  #
#  #
#  # Between Subjects Effects
#  # -----------------------------------------------
#  # Sum of Squares  df  Mean Square  F  p
anovaRMNP

Repeated Measures ANOVA (Non-parametric)

Description

The Friedman test is used to explore the relationship between a continuous dependent variable and a categorical explanatory variable, where the explanatory variable is 'within subjects' (where multiple measurements are from the same subject). It is analogous to Repeated Measures ANOVA, but with the advantage of being non-parametric, and not requiring the assumptions of normality or homogeneity of variances. However, it has the limitation that it can only test a single explanatory variable at a time.

Usage

anovaRMNP(data, measures, pairs = FALSE, desc = FALSE, plots = FALSE, plotType = "means")

Arguments

data the data as a data frame
measures a vector of strings naming the repeated measures variables
pairs TRUE or FALSE (default), perform pairwise comparisons
desc TRUE or FALSE (default), provide descriptive statistics
plots TRUE or FALSE (default), provide a descriptive plot
plotType 'means' (default) or 'medians', the error bars to use in the plot

Value

A results object containing:

results$table a table of the Friedman test results
results$comp a table of the pairwise comparisons
results$desc a table containing the descriptives
results$plot a descriptives plot

Tables can be converted to data frames with asDF or as.data.frame. For example:
results$table$asDF
Examples

data('bugs', package = 'jmv')

anovaRMNP(bugs, measures = vars(LDLF, LDHF, HDLF, HDHF))

# # REPEATED MEASURES ANOVA (NON-PARAMETRIC)
# # Friedman
# # ------------------------
# # X²   df    p
# # ------------------------
# #  55.8  3    < .001
# # ------------------------
#

bugs
data sets

Description
data sets

Author(s)
Ryan, Wilde & Crist (2013)

References

http://faculty.kutztown.edu/rryan/RESEARCH/PUBS/Ryan,%20Wilde,%20%26%20Crist%202013%20Web%20exp%20vs%20lab.pdf

CFA
Confirmatory Factor Analysis

Description

Confirmatory Factor Analysis
Usage

cfa(data, factors = list(list(label = "Factor 1", vars = list())),
    resCov, miss = "fiml", constrain = "facVar", estTest = TRUE,
    ci = FALSE, ciWidth = 95, stdEst = FALSE, factCovEst = TRUE,
    factInterceptEst = FALSE, resCovEst = FALSE,
    resInterceptEst = FALSE, fitMeasures = list("cfi", "tli", "rmsea"),
    modelTest = TRUE, pathDiagram = FALSE, corRes = FALSE,
    hlCorRes = 0.1, mi = FALSE, hlMI = 3)

Arguments

data the data as a data frame
factors a list containing named lists that define the label of the factor and the vars that belong to that factor
resCov a list of lists specifying the residual covariances that need to be estimated
miss 'listwise' or 'fiml', how to handle missing values; 'listwise' excludes a row from all analyses if one of its entries is missing, 'fiml' uses a full information maximum likelihood method to estimate the model.
constrain 'facVar' or 'facInd', how to contrain the model; 'facVar' fixes the factor variances to one, 'facInd' fixes each factor to the scale of its first indicator.
estTest TRUE (default) or FALSE, provide 'Z' and 'p' values for the model estimates
ci TRUE or FALSE (default), provide a confidence interval for the model estimates
ciWidth a number between 50 and 99.9 (default: 95) specifying the confidence interval width that is used as "ci"
stdEst TRUE or FALSE (default), provide a standardized estimate for the model estimates
factCovEst TRUE (default) or FALSE, provide estimates for the factor (co)variances
factInterceptEst TRUE or FALSE (default), provide estimates for the factor intercepts
resCovEst TRUE (default) or FALSE, provide estimates for the residual (co)variances
resInterceptEst TRUE or FALSE (default), provide estimates for the residual intercepts
fitMeasures one or more of 'cfi', 'tli', 'srmr', 'rmsea', 'aic', or 'bic'; use CFI, TLI, SRMR, RMSEA + 90% confidence interval, adjusted AIC, and BIC model fit measures, respectively
modelTest TRUE (default) or FALSE, provide a chi-square test for exact fit that compares the model with the perfect fitting model
pathDiagram TRUE or FALSE (default), provide a path diagram of the model
corRes TRUE or FALSE (default), provide the residuals for the observed correlation matrix (i.e., the difference between the expected correlation matrix and the observed correlation matrix)
hlCorRes a number (default: 0.1), highlight values in the 'corRes' table above this value
mi TRUE or FALSE (default), provide modification indices for the parameters not included in the model
hlMI a number (default: 3), highlight values in the 'modIndices' tables above this value
Value

A results object containing:

- `results$factorLoadings` a table containing the factor loadings
- `results$factorEst$factorCov` a table containing factor covariances estimates
- `results$factorEst$factorIntercept` a table containing factor intercept estimates
- `results$resEst$resCov` a table containing residual covariances estimates
- `results$resEst$resIntercept` a table containing residual intercept estimates
- `results$modelFit$test` a table containing the chi-square test for exact fit
- `results$modelFit$fitMeasures` a table containing fit measures
- `results$modelPerformance$corRes` a table containing residuals for the observed correlation matrix
- `results$modelPerformance$modIndices` a group
- `results$pathDiagram` an image containing the model path diagram
- `results$modelSyntax` the lavaan syntax used to fit the model

Tables can be converted to data frames with `asDF` or `as.data.frame`. For example:

```
results$factorLoadings$asDF
as.data.frame(results$factorLoadings)
```

Examples

```r
data <- lavaan::HolzingerSwineford1939

jmv::cfa(
data = data, factors = list(
  list(label="Visual", vars=c("x1", "x2", "x3")),
  list(label="Textual", vars=c("x4", "x5", "x6")),
  list(label="Speed", vars=c("x7", "x8", "x9")),
  resCov = NULL)
#
# CONFIRMATORY FACTOR ANALYSIS
#
# Factor Loadings
# -----------------------------------------------------------------
# Factor Indicator Estimate  SE     Z    p
# -----------------------------------------------------------------
# Visual  x1    0.900  0.0832 10.81 < .001
# x2     0.498  0.0808  6.16 < .001
# x3     0.656  0.0776  8.46 < .001
# Textual x4    0.990  0.0567 17.46 < .001
# x5     1.102  0.0626 17.60 < .001
# x6     0.917  0.0538 17.05 < .001
# Speed  x7    0.619  0.0743  8.34 < .001
# x8     0.731  0.0755  9.68 < .001
# x9     0.670  0.0775  8.64 < .001
# -----------------------------------------------------------------
```
## FACTOR ESTIMATES

### Factor Covariances

<table>
<thead>
<tr>
<th>Estimate</th>
<th>SE</th>
<th>Z</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visual</td>
<td>Visual</td>
<td>1.000</td>
<td>a</td>
</tr>
<tr>
<td>Textual</td>
<td>0.459</td>
<td>0.0635</td>
<td>7.22</td>
</tr>
<tr>
<td>Speed</td>
<td>0.471</td>
<td>0.0862</td>
<td>5.46</td>
</tr>
<tr>
<td>Textual</td>
<td>1.000</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>Speed</td>
<td>0.283</td>
<td>0.0715</td>
<td>3.96</td>
</tr>
<tr>
<td>Speed</td>
<td>1.000</td>
<td>a</td>
<td></td>
</tr>
</tbody>
</table>

* a fixed parameter

## MODEL FIT

### Test for Exact Fit

<table>
<thead>
<tr>
<th>X²</th>
<th>df</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>85.3</td>
<td>24</td>
<td>&lt; .001</td>
</tr>
</tbody>
</table>

### Fit Measures

<table>
<thead>
<tr>
<th>CFI</th>
<th>TLI</th>
<th>RMSEA</th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.931</td>
<td>0.896</td>
<td>0.0921</td>
<td>0.0714</td>
<td>0.114</td>
</tr>
</tbody>
</table>

---

**Description**

The X² test of association (not to be confused with the X² goodness of fit) is used to test whether two categorical variables are independent or associated. If the p-value is low, it suggests the variables are not independent, and that there is a relationship between the two variables.

**Usage**

```r
contTables(data, rows, cols, counts = NULL, layers = NULL, 
chiSq = TRUE, chiSqCorr = FALSE, likeRat = FALSE, fisher = FALSE, 
contCoef = FALSE, phiCra = FALSE, logOdds = FALSE, odds = FALSE, 
```
relRisk = FALSE, ci = TRUE, ciWidth = 95, gamma = FALSE, taub = FALSE, obs = TRUE, exp = FALSE, pcRow = FALSE, pcCol = FALSE, pcTot = FALSE, formula)

Arguments

data the data as a data frame
rows the variable to use as the rows in the contingency table (not necessary when providing a formula, see the examples)
cols the variable to use as the columns in the contingency table (not necessary when providing a formula, see the examples)
counts the variable to use as the counts in the contingency table (not necessary when providing a formula, see the examples)
layers the variables to use to split the contingency table (not necessary when providing a formula, see the examples)
chiSq TRUE (default) or FALSE, provide \(X^2\)
chiSqCorr TRUE or FALSE (default), provide \(X^2\) with continuity correction
likeRat TRUE or FALSE (default), provide the likelihood ratio
fisher TRUE or FALSE (default), provide Fisher’s exact test
contCoef TRUE or FALSE (default), provide the contingency coefficient
phiCra TRUE or FALSE (default), provide Phi and Cramer’s V
logOdds TRUE or FALSE (default), provide the log odds ratio (only available for 2x2 tables)
ods TRUE or FALSE (default), provide the odds ratio (only available for 2x2 tables)
relRisk TRUE or FALSE (default), provide the relative risk (only available for 2x2 tables)
ci TRUE or FALSE (default), provide confidence intervals for the comparative measures

ciWidth a number between 50 and 99.9 (default: 95), width of the confidence intervals to provide
gamma TRUE or FALSE (default), provide gamma
taub TRUE or FALSE (default), provide Kendall’s tau-b
obs TRUE or FALSE (default), provide the observed counts
exp TRUE or FALSE (default), provide the expected counts
pcRow TRUE or FALSE (default), provide row percentages
pcCol TRUE or FALSE (default), provide column percentages
pcTot TRUE or FALSE (default), provide total percentages
formula (optional) the formula to use, see the examples
contTables

Value
A results object containing:

- `results$freqs` a table of proportions
- `results$chiSq` a table of $X^2$ test results
- `results$odds` a table of comparative measures
- `results$nom` a table of the ‘nominal’ test results
- `results$gamma` a table of the gamma test results
- `results$taub` a table of the Kendall’s tau-b test results

Tables can be converted to data frames with `asDF` or `as.data.frame`. For example:

```r
results$freqs$asDF
as.data.frame(results$freqs)
```

Examples

data('HairEyeColor')
dat <- as.data.frame(HairEyeColor)

contTables(formula = Freq ~ Hair:Eye, dat)

# CONTINGENCY TABLES
# Contingency Tables
# -----------------------------------------------------
# Hair Brown Blue Hazel Green Total
# -----------------------------------------------------
# Black 68 20 15 5 108
# Brown 119 84 54 29 286
# Red 26 17 14 14 71
# Blond 7 94 10 16 127
# Total 220 215 93 64 592
# -----------------------------------------------------

# X² Tests
# ----------------------------------------
# Value df p
# ----------------------------------------
# $X^2$ 138 9 < .001
# N 592
# ----------------------------------------

# Alternatively, omit the left of the formula (‘Freq’) if each row
# represents a single observation:

contTables(formula = ~ Hair:Eye, dat)
Description

McNemar test

Usage

\[
\text{contTablesPaired}(\text{data, rows, cols, counts} = \text{NULL, chiSq} = \text{TRUE, chiSqCorr} = \text{FALSE, exact} = \text{FALSE, pcRow} = \text{FALSE, pcCol} = \text{FALSE, formula})
\]

Arguments

data  
the data as a data frame

rows  
the variable to use as the rows in the contingency table (not necessary when providing a formula, see the examples)

cols  
the variable to use as the columns in the contingency table (not necessary when providing a formula, see the examples)

counts  
the variable to use as the counts in the contingency table (not necessary when providing a formula, see the examples)

chiSq  
TRUE (default) or FALSE, provide \(X^2\)

chiSqCorr  
TRUE or FALSE (default), provide \(X^2\) with continuity correction

exact  
TRUE or FALSE (default), provide an exact log odds ratio (requires exact2x2 to be installed)

pcRow  
TRUE or FALSE (default), provide row percentages

pcCol  
TRUE or FALSE (default), provide column percentages

formula  
(optional) the formula to use, see the examples

Value

A results object containing:

\[
\begin{align*}
\text{results$fREQs} & \quad \text{a proportions table} \\
\text{results$test} & \quad \text{a table of test results}
\end{align*}
\]

Tables can be converted to data frames with asDF or as.data.frame. For example:

\[
\begin{align*}
\text{results$fREQs$asDF} \\
\text{as.data.frame(results$fREQs)}
\end{align*}
\]
Examples

dat <- data.frame(
  '1st survey' = c('Approve', 'Approve', 'Disapprove', 'Disapprove'),
  '2nd survey' = c('Approve', 'Disapprove', 'Approve', 'Disapprove'),
  'Counts' = c(794, 150, 86, 570),
  check.names=FALSE)

contTablesPaired(formula = Counts ~ `1st survey`:`2nd survey`, data = dat)

# # PAIRED SAMPLES CONTINGENCY TABLES
# # Contingency Tables
# # ------------------------------------------------
# # 1st survey Approve Disapprove Total
# # ------------------------------------------------
# # Approve 794 150 944
# # Disapprove 86 570 656
# # Total 880 720 1600
# # ------------------------------------------------
#
# # McNemar Test
# # -----------------------------------------------------
# # Value df p
# # -----------------------------------------------------
# # X² 17.4 1 < .001
# # X² continuity correction 16.8 1 < .001
# # -----------------------------------------------------
#
# Alternatively, omit the left of the formula ('Counts') from the
# formula if each row represents a single observation:

contTablesPaired(formula = ~ `1st survey`:`2nd survey`, data = dat)

---

corrMatrix	Correlation Matrix

description

Correlation matrices are a way to examine linear relationships between two or more continuous variables.

Usage

corrMatrix(data, vars, pearson = TRUE, spearman = FALSE,
  kendall = FALSE, sig = TRUE, flag = FALSE, n = FALSE,
ci = FALSE, ciWidth = 95, plots = FALSE, plotDens = FALSE, plotStats = FALSE, hypothesis = "corr")

Arguments

data       the data as a data frame
vars       a vector of strings naming the variables to correlate in data
pearson    TRUE (default) or FALSE, provide Pearson’s R
spearman   TRUE or FALSE (default), provide Spearman’s rho
kendall    TRUE or FALSE (default), provide Kendall’s tau-b
sig        TRUE (default) or FALSE, provide significance levels
flag       TRUE or FALSE (default), flag significant correlations
n          TRUE or FALSE (default), provide the number of cases
ci         TRUE or FALSE (default), provide confidence intervals
ciWidth    a number between 50 and 99.9 (default: 95), the width of confidence intervals to provide
plots      TRUE or FALSE (default), provide a correlation matrix plot
plotDens   TRUE or FALSE (default), provide densities in the correlation matrix plot
plotStats  TRUE or FALSE (default), provide statistics in the correlation matrix plot
hypothesis one of 'corr' (default), 'pos', 'neg' specifying the alternative hypothesis; correlated, correlated positively, correlated negatively respectively.

Details

For each pair of variables, a Pearson’s r value indicates the strength and direction of the relationship between those two variables. A positive value indicates a positive relationship (higher values of one variable predict higher values of the other variable). A negative Pearson’s r indicates a negative relationship (higher values of one variable predict lower values of the other variable, and vice-versa). A value of zero indicates no relationship (whether a variable is high or low, does not tell us anything about the value of the other variable).

More formally, it is possible to test the null hypothesis that the correlation is zero and calculate a p-value. If the p-value is low, it suggests the correlation co-efficient is not zero, and there is a linear (or more complex) relationship between the two variables.

Value

A results object containing:

results$matrix       a correlation matrix table
results$plot         a correlation matrix plot

Tables can be converted to data frames with asDF or as.data.frame. For example:
results$matrix$asDF
as.data.frame(results$matrix)

Examples

## Not run:
data('mtcars')
corrMatrix(mtcars, vars = vars(mpg, cyl, disp, hp))

#
# CORRELATION MATRIX
#
# Correlation Matrix
# --------------------------------------------------------------
# mpg cyl disp hp
# --------------------------------------------------------------
# mpg Pearson's r -0.852 -0.848 -0.776
# p-value < .001 < .001 < .001
#
# cyl Pearson's r 0.902 0.832
# p-value < .001 < .001
#
# disp Pearson's r 0.791
# p-value < .001
#
# hp Pearson's r
# p-value
# --------------------------------------------------------------
#
## End(Not run)

descriptives

**Descriptives**

Descriptives are an assortment of summarising statistics, and visualizations which allow exploring the shape and distribution of data. It is good practice to explore your data with descriptives before proceeding to more formal tests.

Usage

```r

descriptives(data, vars, splitBy = NULL, freq = FALSE, hist = FALSE, dens = FALSE, bar = FALSE, barCounts = FALSE, box = FALSE, violin = FALSE, dot = FALSE, dotType = "jitter", qq = FALSE, n = TRUE, missing = TRUE, mean = TRUE, median = TRUE, mode = FALSE, sum = FALSE, sd = FALSE, variance = FALSE, range = FALSE, min = TRUE, max = TRUE, se = FALSE, skew = FALSE, kurt = FALSE, sw = FALSE, quart = FALSE, pcEqGr = FALSE, pcNEqGr = 4, formula)
```
Arguments

data the data as a data frame
vars a vector of strings naming the variables of interest in data
splitBy a vector of strings naming the variables used to split vars
freq TRUE or FALSE (default), provide frequency tables (nominal, ordinal variables only)
hist TRUE or FALSE (default), provide histograms (continuous variables only)
dens TRUE or FALSE (default), provide density plots (continuous variables only)
bar TRUE or FALSE (default), provide bar plots (nominal, ordinal variables only)
barCounts TRUE or FALSE (default), add counts to the bar plots
box TRUE or FALSE (default), provide box plots (continuous variables only)
vio TRUE or FALSE (default), provide violin plots (continuous variables only)
dot TRUE or FALSE (default), provide dot plots (continuous variables only)
dotType 
qq TRUE or FALSE (default), provide Q-Q plots (continuous variables only)
n TRUE (default) or FALSE, provide the sample size
missing TRUE (default) or FALSE, provide the number of missing values
mean TRUE (default) or FALSE, provide the mean
median TRUE (default) or FALSE, provide the median
mode TRUE or FALSE (default), provide the mode
sum TRUE or FALSE (default), provide the sum
sd TRUE or FALSE (default), provide the standard deviation
variance TRUE or FALSE (default), provide the variance
range TRUE or FALSE (default), provide the range
min TRUE or FALSE (default), provide the minimum
max TRUE or FALSE (default), provide the maximum
se TRUE or FALSE (default), provide the standard error
skew TRUE or FALSE (default), provide the skewness
kurt TRUE or FALSE (default), provide the kurtosis
sw TRUE or FALSE (default), provide Shapiro-Wilk p-value
quart TRUE or FALSE (default), provide quartiles
pcEqGr TRUE or FALSE (default), provide quantiles
pcNEqGr an integer (default: 4) specifying the number of equal groups
formula (optional) the formula to use, see the examples

Value

A results object containing:
### Descriptives

- `results$descriptives`: a table of the descriptive statistics
- `results$frequencies`: an array of frequency tables
- `results$plots`: an array of descriptive plots

Tables can be converted to data frames with `asDF` or `as.data.frame`. For example:

```r
results$descriptives$asDF
as.data.frame(results$descriptives)
```

### Examples

```r
## Not run:
data('mtcars')
dat <- mtcars

# frequency tables can be provided for factors
dat$gear <- as.factor(dat$gear)

descriptives(dat, vars = vars(mpg, cyl, disp, gear), freq = TRUE)
```

```
# DESCRIPTIVES
# -------------------------------------------
# mpg cyl disp gear
# -------------------------------------------
# N 32 32 32 32
# Missing 0 0 0 0
# Mean 20.1 6.19 231 3.69
# Median 19.2 6.00 196 4.00
# Minimum 10.4 4.00 71.1 3
# Maximum 33.9 8.00 472 5
# -------------------------------------------

# FREQUENCIES
# Frequencies of gear
# --------------------
# Levels Counts
# --------------------
# 3 15
# 4 12
# 5 5
# --------------------

# splitting by a variable
descriptives(formula = disp + mpg ~ cyl, dat,
             median=F, min=F, max=F, n=F, missing=F)
```
# providing histograms
descriptives(formula = mpg ~ cyl, dat, hist=T,
    median=F, min=F, max=F, n=F, missing=F)

# splitting by multiple variables
descriptives(formula = mpg ~ cyl:gear, dat,
    median=F, min=F, max=F, missing=F)

## End(Not run)

---

## efa

### Exploratory Factor Analysis

**Description**

Exploratory Factor Analysis

**Usage**

```r
efa(data, vars, nFactorMethod = "parallel", nFactors = 1,
    minEigen = 0, extraction = "minres", rotation = "oblimin",
    hideLoadings = 0.3, sortLoadings = FALSE, screePlot = FALSE,
    eigen = FALSE, factorCor = FALSE, factorSummary = FALSE,
    modelFit = FALSE, kmo = FALSE, bartlett = FALSE)
```

**Arguments**

- **data**: the data as a data frame
- **vars**: a vector of strings naming the variables of interest in `data`
- **nFactorMethod**: 'parallel' (default), 'eigen' or 'fixed', the way to determine the number of factors
- **nFactors**: an integer (default: 1), the number of factors in the model
- **minEigen**: a number (default: 0), the minimal eigenvalue for a factor to be included in the model
- **extraction**: 'minres' (default), 'ml', or 'pa' use respectively 'minimum residual', 'maximum likelihood', or 'principal axis' as the factor extraction method
- **rotation**: 'none', 'varimax', 'quartimax', 'promax', 'oblimin' (default), or 'simplimax', the rotation to use in estimation
- **hideLoadings**: a number (default: 0.3), hide factor loadings below this value
- **sortLoadings**: TRUE or FALSE (default), sort the factor loadings by size
- **screePlot**: TRUE or FALSE (default), show scree plot
- **eigen**: TRUE or FALSE (default), show eigenvalue table
- **factorCor**: TRUE or FALSE (default), show factor correlations
linReg

factorSummary TRUE or FALSE (default), show factor summary
modelFit TRUE or FALSE (default), show model fit measures and test
kmo TRUE or FALSE (default), show Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy (MSA) results
bartlett TRUE or FALSE (default), show Bartlett’s test of sphericity results

Value
A results object containing:

results$text a preformatted

Examples

data('iris')
efa(iris, vars = vars(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width))

# EXPLORATORY FACTOR ANALYSIS
# Factor Loadings
# ------------------------------------------------
# 1 2 Uniqueness
# ------------------------------------------------
# Sepal.Length 0.993 0.10181
# Sepal.Width 0.725 0.42199
# Petal.Length 0.933 0.00483
# Petal.Width 0.897 0.07088
# ------------------------------------------------
# Note. 'oblimin' rotation was used
#

linReg Linear Regression

Description
Linear regression is used to explore the relationship between a continuous dependent variable, and one or more continuous and/or categorical explanatory variables. Other statistical methods, such as ANOVA and ANCOVA, are in reality just forms of linear regression.

Usage
linReg(data, dep, covs = NULL, factors = NULL, blocks = list(list())), refLevels = NULL, intercept = "refLevel", r = TRUE, r2 = TRUE,
linReg

r2Adj = FALSE, aic = FALSE, bic = FALSE, rmse = FALSE,
modelTest = FALSE, anova = FALSE, ci = FALSE, ciWidth = 95,
stdEst = FALSE, ciStdEst = FALSE, ciWidthStdEst = 95,
coefPlot = FALSE, norm = FALSE, qqPlot = FALSE, resPlots = FALSE,
durbin = FALSE, collin = FALSE, cooks = FALSE,
emMeans = list(list()), ciEmm = TRUE, ciWidthEmm = 95,
emmPlots = TRUE, emmTables = FALSE, emmWeights = TRUE)

Arguments

data the data as a data frame
dep the dependent variable from data, variable must be numeric
covs the covariates from data
factors the fixed factors from data
blocks a list containing vectors of strings that name the predictors that are added to the model. The elements are added to the model according to their order in the list
refLevels a list of lists specifying reference levels of the dependent variable and all the factors
intercept 'refLevel' (default) or 'grandMean', coding of the intercept. Either creates contrast so that the intercept represents the reference level or the grand mean
r TRUE (default) or FALSE, provide the statistical measure R for the models
r2 TRUE (default) or FALSE, provide the statistical measure R-squared for the models
r2Adj TRUE or FALSE (default), provide the statistical measure adjusted R-squared for the models
aic TRUE or FALSE (default), provide Akaike’s Information Criterion (AIC) for the models
bic TRUE or FALSE (default), provide Bayesian Information Criterion (BIC) for the models
rmse TRUE or FALSE (default), provide RMSE for the models
modelTest TRUE (default) or FALSE, provide the model comparison between the models and the NULL model
anova TRUE or FALSE (default), provide the omnibus ANOVA test for the predictors
ci TRUE or FALSE (default), provide a confidence interval for the model coefficients
ciWidth a number between 50 and 99.9 (default: 95) specifying the confidence interval width
stdEst TRUE or FALSE (default), provide a standardized estimate for the model coefficients
ciStdEst TRUE or FALSE (default), provide a confidence interval for the model coefficient standardized estimates
ciWidthStdEst a number between 50 and 99.9 (default: 95) specifying the confidence interval width
linReg

coefPlot TRUE or FALSE (default), provide a coefficient plot where for each predictor the estimated coefficient and confidence intervals are plotted.
norm TRUE or FALSE (default), perform a Shapiro-Wilk test on the residuals
qqPlot TRUE or FALSE (default), provide a Q-Q plot of residuals
resPlots TRUE or FALSE (default), provide residual plots where the dependent variable and each covariate is plotted against the standardized residuals.
durbin TRUE or FALSE (default), provide results of the Durbin- Watson test for autocorrelation
collin TRUE or FALSE (default), provide VIF and tolerance collinearity statistics
cooks TRUE or FALSE (default), provide summary statistics for the Cook’s distance
emMeans a formula containing the terms to estimate marginal means for, supports up to three variables per term
ciEmm TRUE (default) or FALSE, provide a confidence interval for the estimated marginal means
ciWidthEmm a number between 50 and 99.9 (default: 95) specifying the confidence interval width for the estimated marginal means
emmPlots TRUE (default) or FALSE, provide estimated marginal means plots
emmTables TRUE or FALSE (default), provide estimated marginal means tables
emmWeights TRUE (default) or FALSE, weigh each cell equally or weigh them according to the cell frequency

Value

A results object containing:

results$modelFit a table
results$modelComp a table
results$models an array of model specific results

Tables can be converted to data frames with asDF or as.data.frame. For example:

results$modelFit$asDF
as.data.frame(results$modelFit)

Examples

data(’Prestige’, package=’carData’)

linReg(data = Prestige, dep = income,
    covs = vars(education, prestige, women),
    blocks = list(list(’education’, ’prestige’, ’women’)))

# # LINEAR REGRESSION #
logLinear

Log-Linear Regression

Description

Log-Linear Regression

Usage

logLinear(data, factors = NULL, counts = NULL, blocks = list(list())), refLevels = NULL, modelTest = FALSE, dev = TRUE, aic = TRUE, bic = FALSE, pseudoR2 = list("r2mf"), omni = FALSE, ci = FALSE, ciWidth = 95, RR = FALSE, ciRR = FALSE, ciWidthRR = 95, emMeans = list(list())), ciEmm = TRUE, ciWidthEmm = 95, emmPlots = TRUE, emmTables = FALSE, emmWeights = TRUE)

Arguments

data the data as a data frame
factors a vector of strings naming the factors from data
counts a string naming a variable in data containing counts, or NULL if each row represents a single observation
blocks a list containing vectors of strings that name the predictors that are added to the model. The elements are added to the model according to their order in the list
logLinear

refLevels  a list of lists specifying reference levels of the dependent variable and all the factors
modelTest  TRUE or FALSE (default), provide the model comparison between the models and the NULL model
dev  TRUE (default) or FALSE, provide the deviance (or -2LogLikelihood) for the models
aic  TRUE (default) or FALSE, provide Aikaike’s Information Criterion (AIC) for the models
bic  TRUE or FALSE (default), provide Bayesian Information Criterion (BIC) for the models
pseudoR2  one or more of 'r2mf', 'r2cs', or 'r2n'; use McFadden’s, Cox & Snell, and Nagelkerke pseudo-R², respectively
omni  TRUE or FALSE (default), provide the omnibus likelihood ratio tests for the predictors
ci  TRUE or FALSE (default), provide a confidence interval for the model coefficient estimates
ciWidth  a number between 50 and 99.9 (default: 95) specifying the confidence interval width
RR  TRUE or FALSE (default), provide the exponential of the log-rate ratio estimate, or the rate ratio estimate
ciRR  TRUE or FALSE (default), provide a confidence interval for the model coefficient rate ratio estimates
ciWidthRR  a number between 50 and 99.9 (default: 95) specifying the confidence interval width
emMeans  a list of lists specifying the variables for which the estimated marginal means need to be calculate. Supports up to three variables per term.
ciEmm  TRUE (default) or FALSE, provide a confidence interval for the estimated marginal means
ciWidthEmm  a number between 50 and 99.9 (default: 95) specifying the confidence interval width for the estimated marginal means
emmPlots  TRUE (default) or FALSE, provide estimated marginal means plots
emmTables  TRUE or FALSE (default), provide estimated marginal means tables
emmWeights  TRUE (default) or FALSE, weigh each cell equally or weigh them according to the cell frequency

Value

A results object containing:

results$modelFit             a table
results$modelComp            a table
results$models               an array of model specific results
Tables can be converted to data frames with asDF or as.data.frame. For example:
results$modelFit$asDF
as.data.frame(results$modelFit)

Examples

data('mtcars')

tab <- table(gear=mtcars$gear, cyl=mtcars$cyl)
dat <- as.data.frame(tab)

logLinear(data = dat, factors = vars(gear, cyl), counts = Freq,
blocks = list(list("gear", "cyl", c("gear", "cyl"))),
refLevels = list(
    list(var="gear", ref="3"),
    list(var="cyl", ref="4")))

# LOG-LINEAR REGRESSION
#
# Model Fit Measures
# ---------------------------------------
# Model Deviance AIC R² McF
# ---------------------------------------
# 1 4.12e-10 41.4 1.000
# ---------------------------------------
#
# MODEL SPECIFIC RESULTS
#
# MODEL 1
#
# Model Coefficients
# ------------------------------------------------------------------
# Predictor Estimate SE  Z  p
# ------------------------------------------------------------------
# Intercept  -4.71e-16 1.00  -4.71e-16 1.000
# gear:
# 4 3 2.079 1.06 1.961 0.050
# 5 3 0.693 1.22 0.566 0.571
# cyl:
# 6 4 0.693 1.22 0.566 0.571
# 8 4 2.485 1.04 2.387 0.017
# gear:cyl:
# (4 3):(6 4) -1.386 1.37 -1.012 0.311
# (5 3):(6 4) -1.386 1.73 -0.800 0.423
# (4 3):(8 4) -26.867 42247.17 -6.36e-4 0.999
# (5 3):(8 4) -2.485 1.44 -1.722 0.085
# ------------------------------------------------------------------
**logRegBin**  

*Binomial Logistic Regression*

**Description**

Binomial Logistic Regression

**Usage**

```r
logRegBin(data, dep, covs = NULL, factors = NULL,  
blocks = list(list()), refLevels = NULL, modelTest = FALSE,  
dev = TRUE, aic = TRUE, bic = FALSE, pseudoR2 = list("r2mf"),  
omi = FALSE, ci = FALSE, ciWidth = 95, OR = FALSE,  
ciOR = FALSE, ciWidthOR = 95, emMeans = list(list()),  
emEmm = TRUE, ciWidthEmm = 95, emmPlots = TRUE,  
emmTables = FALSE, emmWeights = TRUE, class = FALSE, acc = FALSE,  
spec = FALSE, sens = FALSE, auc = FALSE, rocPlot = FALSE,  
cutOff = 0.5, cut0ffPlot = FALSE, collin = FALSE,  
boxTidwell = FALSE, cooks = FALSE)
```

**Arguments**

- `data`: the data as a data frame
- `dep`: a string naming the dependent variable from data, variable must be a factor
- `covs`: a vector of strings naming the covariates from data
- `factors`: a vector of strings naming the fixed factors from data
- `blocks`: a list containing vectors of strings that name the predictors that are added to the model. The elements are added to the model according to their order in the list
- `refLevels`: a list of lists specifying reference levels of the dependent variable and all the factors
- `modelTest`: TRUE or FALSE (default), provide the model comparison between the models and the NULL model
- `dev`: TRUE (default) or FALSE, provide the deviance (or -2LogLikelihood) for the models
- `aic`: TRUE (default) or FALSE, provide Aikaike’s Information Criterion (AIC) for the models
- `bic`: TRUE or FALSE (default), provide Bayesian Information Criterion (BIC) for the models
- `pseudoR2`: one or more of 'r2mf', 'r2cs', or 'r2n'; use McFadden’s, Cox & Snell, and Nagelkerke pseudo-R², respectively
- `omni`: TRUE or FALSE (default), provide the omnibus likelihood ratio tests for the predictors
- `ci`: TRUE or FALSE (default), provide a confidence interval for the model coefficient estimates
ciWidth
a number between 50 and 99.9 (default: 95) specifying the confidence interval width

OR
TRUE or FALSE (default), provide the exponential of the log-odds ratio estimate, or the odds ratio estimate

ciOR
TRUE or FALSE (default), provide a confidence interval for the model coefficient odds ratio estimates

ciWidthOR
a number between 50 and 99.9 (default: 95) specifying the confidence interval width

emMeans
a list of lists specifying the variables for which the estimated marginal means need to be calculate. Supports up to three variables per term.

ciEmm
TRUE (default) or FALSE, provide a confidence interval for the estimated marginal means

ciWidthEmm
a number between 50 and 99.9 (default: 95) specifying the confidence interval width for the estimated marginal means

emmPlots
TRUE (default) or FALSE, provide estimated marginal means plots

emmTables
TRUE (default), provide estimated marginal means tables

emmWeights
TRUE (default) or FALSE, weigh each cell equally or weigh them according to the cell frequency

class
TRUE or FALSE (default), provide a predicted classification table (or confusion matrix)

acc
TRUE or FALSE (default), provide the predicted accuracy of outcomes grouped by the cut-off value

spec
TRUE or FALSE (default), provide the predicted specificity of outcomes grouped by the cut-off value

sens
TRUE or FALSE (default), provide the predicted sensitivity of outcomes grouped by the cut-off value

auc
TRUE or FALSE (default), provide the area under the ROC curve (AUC)

rocPlot
TRUE or FALSE (default), provide a ROC curve plot

cutOff
TRUE or FALSE (default), set a cut-off used for the predictions

cutOffPlot
TRUE or FALSE (default), provide a cut-off plot

collin
TRUE or FALSE (default), provide VIF and tolerance collinearity statistics

boxTidwell
TRUE or FALSE (default), provide Box-Tidwell test for linearity of the logit

cooks
TRUE or FALSE (default), provide summary statistics for the Cook’s distance

Value
A results object containing:

results$modelFit a table
results$modelComp a table
results$models an array of model specific results
Tables can be converted to data frames with \textit{asDF} or \texttt{as.data.frame}. For example:

\begin{verbatim}
results$modelFit$asDF
as.data.frame(results$modelFit)
\end{verbatim}

\textbf{Examples}

\begin{verbatim}
data('birthwt', package='MASS')

dat <- data.frame(
    low = factor(birthwt$low),
    age = birthwt$age,
    bwt = birthwt$bwt)

logRegBin(data = dat, dep = low,
    covs = vars(age, bwt),
    blocks = list(list("age", "bwt")),
    refLevels = list(list(var="low", ref="0")))

# # BINOMIAL LOGISTIC REGRESSION
# # Model Fit Measures
# # ---------------------------------------
# # Model Deviance AIC R²-\textit{McF}
# # ---------------------------------------
# # 1 4.97e-7 6.00 1.000
# # ---------------------------------------
# # # MODEL SPECIFIC RESULTS
# # # MODEL 1
# # # Model Coefficients
# # # ---------------------------------------
# # # Predictor Estimate SE Z p
# # # ---------------------------------------
# # Intercept 2974.73225 218237.2 0.0136 0.989
# # age -0.00653 482.7 -1.35e-5 1.000
# # bwt -1.18532 87.0 -0.0136 0.989
# # ---------------------------------------
# # Note. Estimates represent the log odds of "low = 1"
# # vs. "low = 0"
# #
\end{verbatim}
Description

Multinomial Logistic Regression

Usage

logRegMulti(data, dep, covs = NULL, factors = NULL,
blocks = list(list()), refLevels = NULL, modelTest = FALSE,
dev = TRUE, aic = TRUE, bic = FALSE, pseudoR2 = list("r2mf"),
onmi = FALSE, ci = FALSE, ciWidth = 95, OR = FALSE,
OR = FALSE, ciWidthOR = 95, emMeans = list(list()),
OR = TRUE, ciWidthEmm = 95, emmPlots = TRUE,
emm tables = FALSE, emmWeights = TRUE)

Arguments

data the data as a data frame

dep a string naming the dependent variable from data, variable must be a factor
covs a vector of strings naming the covariates from data
factors a vector of strings naming the fixed factors from data
blocks a list containing vectors of strings that name the predictors that are added to the

model. The elements are added to the model according to their order in the list
refLevels a list of lists specifying reference levels of the dependent variable and all the
factors
modelTest TRUE or FALSE (default), provide the model comparison between the models and
the NULL model
dev TRUE (default) or FALSE, provide the deviance (or -2LogLikelihood) for the models
aic TRUE (default) or FALSE, provide Aikaike’s Information Criterion (AIC) for the models
bic TRUE or FALSE (default), provide Bayesian Information Criterion (BIC) for the models
pseudoR2 one or more of 'r2mf', 'r2cs', or 'r2n'; use McFadden’s, Cox & Snell, and
Nagelkerke pseudo-R², respectively
omni TRUE or FALSE (default), provide the omnibus likelihood ratio tests for the pre-
dictors

OR TRUE or FALSE (default), provide a confidence interval for the model coefficient
estimates
ci a number between 50 and 99.9 (default: 95) specifying the confidence interval
width

OR TRUE or FALSE (default), provide the exponential of the log-odds ratio estimate,
or the odds ratio estimate
ciOR TRUE or FALSE (default), provide a confidence interval for the model coefficient
odds ratio estimates
**logRegMulti**

- **ciWidthOR**: a number between 50 and 99.9 (default: 95) specifying the confidence interval width.
- **emMeans**: a list of lists specifying the variables for which the estimated marginal means need to be calculated. Supports up to three variables per term.
- **ciEmm**: TRUE (default) or FALSE, provide a confidence interval for the estimated marginal means.
- **ciWidthEmm**: a number between 50 and 99.9 (default: 95) specifying the confidence interval width for the estimated marginal means.
- **emmPlots**: TRUE (default) or FALSE, provide estimated marginal means plots.
- **emmTables**: TRUE or FALSE (default), provide estimated marginal means tables.
- **emmWeights**: TRUE (default) or FALSE, weigh each cell equally or weigh them according to the cell frequency.

**Value**

A results object containing:

```r
results$modelFit        a table
results$modelComp       a table
results$models         an array of model specific results
```

Tables can be converted to data frames with `asDF` or `as.data.frame`. For example:

```r
results$modelFit$asDF
as.data.frame(results$modelFit)
```

**Examples**

```r
data('birthwt', package='MASS')

dat <- data.frame(
  race = factor(birthwt$race),
  age = birthwt$age,
  low = factor(birthwt$low))

logRegMulti(data = dat, dep = race,
            covs = age, factors = low,
            blocks = list(list("age", "low")),
            refLevels = list(
              list(var="race", ref="1"),
              list(var="low", ref="0")))
```

# MULTINOMIAL LOGISTIC REGRESSION
#
# Model Fit Measures
# _______________________
# Model  Deviance  AIC  R^2-McF
# MODEL SPECIFIC RESULTS

# MODEL 1

# Model Coefficients
# ---------------------------------------------------------------
# race Predictor Estimate SE Z p
# ---------------------------------------------------------------
# 2 - 1 Intercept 0.8155 1.1186 0.729 0.466
# age -0.1038 0.0487 -2.131 0.033
# low:
# 1 0 0.7527 0.4700 1.601 0.109
# 3 - 1 Intercept 1.0123 0.7798 1.298 0.194
# age -0.0663 0.0324 -2.047 0.041
# low:
# 1 0 0.5677 0.3522 1.612 0.107
# ---------------------------------------------------------------

logRegOrd

**Ordinal Logistic Regression**

**Description**

Ordinal Logistic Regression

**Usage**

```r
logRegOrd(data, dep, covs = NULL, factors = NULL, 
blocks = list(list())), refLevels = NULL, modelTest = FALSE, 
dev = TRUE, aic = TRUE, bic = FALSE, pseudoR2 = list("r2mf"), 
omic = FALSE, thres = FALSE, ci = FALSE, ciWidth = 95, 
OR = FALSE, ciOR = FALSE, ciWidthOR = 95)
```

**Arguments**

- `data` the data as a data frame
- `dep` a string naming the dependent variable from `data`, variable must be a factor
- `covs` a vector of strings naming the covariates from `data`
- `factors` a vector of strings naming the fixed factors from `data`
blocks
a list containing vectors of strings that name the predictors that are added to the model. The elements are added to the model according to their order in the list.

refLevels
a list of lists specifying reference levels of the dependent variable and all the factors.

modelTest
TRUE or FALSE (default), provide the model comparison between the models and the NULL model.

dev
TRUE (default) or FALSE, provide the deviance (or -2LogLikelihood) for the models.

aic
TRUE (default) or FALSE, provide Aikaike’s Information Criterion (AIC) for the models.

bic
TRUE or FALSE (default), provide Bayesian Information Criterion (BIC) for the models.

pseudoR2
one or more of 'r2ml', 'r2cs', or 'r2n'; use McFadden’s, Cox & Snell, and Nagelkerke pseudo-R², respectively.

omni
TRUE or FALSE (default), provide the omnibus likelihood ratio tests for the predictors.

thres
TRUE or FALSE (default), provide the thresholds that are used as cut-off scores for the levels of the dependent variable.

ci
TRUE or FALSE (default), provide a confidence interval for the model coefficient estimates.

ciWidth
a number between 50 and 99.9 (default: 95) specifying the confidence interval width.

OR
TRUE or FALSE (default), provide the exponential of the log-odds ratio estimate, or the odds ratio estimate.

ciOR
TRUE or FALSE (default), provide a confidence interval for the model coefficient odds ratio estimates.

ciWidthOR
a number between 50 and 99.9 (default: 95) specifying the confidence interval width.

Value
A results object containing:

```r
results$modelFit
results$modelComp
results$models
```

Tables can be converted to data frames with asDF or as.data.frame. For example:

```r
results$modelFit$asDF
as.data.frame(results$modelFit)
```
Examples

```r
set.seed(1337)

y <- factor(sample(1:3, 100, replace = TRUE))
x1 <- rnorm(100)
x2 <- rnorm(100)

df <- data.frame(y=y, x1=x1, x2=x2)

logRegOrd(data = df, dep = y,
  covs = vars(x1, x2),
  blocks = list(list("x1", "x2")))
```

### ORDINAL LOGISTIC REGRESSION

### Model Fit Measures

<table>
<thead>
<tr>
<th>Model</th>
<th>Deviance</th>
<th>AIC</th>
<th>R²-McF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>218</td>
<td>226</td>
<td>5.68e-4</td>
</tr>
</tbody>
</table>

### MODEL SPECIFIC RESULTS

### MODEL 1

### Model Coefficients

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Estimate</th>
<th>SE</th>
<th>Z</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>0.0579</td>
<td>0.193</td>
<td>0.300</td>
<td>0.764</td>
</tr>
<tr>
<td>x2</td>
<td>0.0330</td>
<td>0.172</td>
<td>0.192</td>
<td>0.848</td>
</tr>
</tbody>
</table>

mancova MANCOVA

Description

Multivariate Analysis of (Co)Variance (MANCOVA) is used to explore the relationship between multiple dependent variables, and one or more categorical and/or continuous explanatory variables.

Usage

```r
mancova(data, deps, factors = NULL, covs = NULL,
```
multivar = list("pillai", "wilks", "hotel", "roy"), boxM = FALSE, shapiro = FALSE, qqPlot = FALSE)

Arguments

- **data**: the data as a data frame
- **deps**: a string naming the dependent variable from data, variable must be numeric
- **factors**: a vector of strings naming the factors from data
- **covs**: a vector of strings naming the covariates from data
- **multivar**: one or more of 'pillai', 'wilks', 'hotel', or 'roy'; use Pillai’s Trace, Wilks' Lambda, Hotelling's Trace, and Roy's Largest Root multivariate statistics, respectively
- **boxM**: TRUE or FALSE (default), provide Box’s M test
- **shapiro**: TRUE or FALSE (default), provide Shapiro-Wilk test
- **qqPlot**: TRUE or FALSE (default), provide a Q-Q plot of multivariate normality

Value

A results object containing:

- `results$multivar`: a table
- `results$univar`: a table
- `results$assump$boxM`: a table
- `results$assump$shapiro`: a table
- `results$assump$qqPlot`: an image

Tables can be converted to data frames with `asDF` or `as.data.frame`. For example:

```r
results$multivar$asDF
as.data.frame(results$multivar)
```

Examples

data('iris')

mancova(data = iris,
        deps = vars(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width),
        factors = Species)

#
# MANCOVA
#
# Multivariate Tests
# -----------------------------------
#   value  F   df1  df2   p
# --------------------------
# Species Pillai's Trace 1.19 53.5   8 290 < .001
# Wilks' Lambda 0.0234 199 8 288 < .001
# Hotelling's Trace 32.5 581 8 286 < .001
# Roy's Largest Root 32.2 1167 4 145 < .001

# Univariate Tests
# -----------------------------------------------------------------------------------------------
# Dependent Variable    Sum of Squares  df  Mean Square  F       p
# -----------------------------------------------------------------------------------------------
# Species Sepal.Length  63.21  2  31.6061  119.3 < .001
#          Sepal.Width   11.34  2   5.6725  49.2  < .001
#          Petal.Length  437.10  2 218.5514 1180.2 < .001
#          Petal.Width   80.41  2  40.2067  960.0 < .001
# Residuals Sepal.Length  38.96 147  0.2650
#          Sepal.Width   16.96 147  0.1154
#          Petal.Length  27.22 147  0.1852
#          Petal.Width   6.16 147  0.0419
# -----------------------------------------------------------------------------------------------

pca  

Principal Component Analysis

Description
Principal Component Analysis

Usage
pca(data, vars, nFactorMethod = "parallel", nFactors = 1,
    minEigen = 1, rotation = "varimax", hideLoadings = 0.3,
    sortLoadings = FALSE, screePlot = FALSE, eigen = FALSE,
    factorCor = FALSE, factorSummary = FALSE, kmo = FALSE,
    bartlett = FALSE)

Arguments
data  

vars  

nFactorMethod  'parallel' (default), 'eigen' or 'fixed', the way to determine the number

nFactors  

minEigen  

rotation  'none', 'varimax' (default), 'quartimax', 'promax', 'oblimin', or 'simplicial',

    the rotation to use in estimation
hideLoadings  a number (default: 0.3), hide loadings below this value
sortLoadings  TRUE or FALSE (default), sort the factor loadings by size
screePlot  TRUE or FALSE (default), show scree plot
eigen  TRUE or FALSE (default), show eigenvalue table
factorCor  TRUE or FALSE (default), show factor correlations
factorSummary  TRUE or FALSE (default), show factor summary
kmo  TRUE or FALSE (default), show Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy (MSA) results
bartlett  TRUE or FALSE (default), show Bartlett’s test of sphericity results

Value
A results object containing:

results$loadings  a table
results$factorStats$factorSummary  a table
results$factorStats$factorCor  a table
results$modelFit$fit  a table
results$assump$bartlett  a table
results$assump$kmo  a table
results$eigen$initEigen  a table
results$eigen$screePlot  an image

Tables can be converted to data frames with asDF or as.data.frame. For example:
results$loadings$asDF
as.data.frame(results$loadings)

Examples

data('iris')

pca(iris, vars = vars(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width))

#  # PRINCIPAL COMPONENT ANALYSIS
#  
#  # Component Loadings
#  # ---------------------------------------------------------------
#  #                  1      Uniqueness
#  # ---------------------------------------------------------------
#  # Sepal.Length  0.890  0.2076
#  # Sepal.Width  -0.460  0.7883
#  # Petal.Length  0.992  0.0168
#  # Petal.Width  0.965  0.0688
#  # ---------------------------------------------------------------
#  # Note. 'varimax' rotation was used


# propTest2

## Description

The Binomial test is used to test the Null hypothesis that the proportion of observations match some expected value. If the p-value is low, this suggests that the Null hypothesis is false, and that the true proportion must be some other value.

## Usage

```r
propTest2(data, vars, areCounts = FALSE, testValue = 0.5,
    hypothesis = "notequal", ci = FALSE, ciWidth = 95, bf = FALSE,
    priorA = 1, priorB = 1, ciBayes = FALSE, ciBayesWidth = 95,
    postPlots = FALSE)
```

## Arguments

- `data` - the data as a data frame
- `vars` - a vector of strings naming the variables of interest in `data`
- `areCounts` - TRUE or FALSE (default), the variables are counts
- `testValue` - a number (default: 0.5), the value for the null hypothesis
- `hypothesis` - 'notequal' (default), 'greater' or 'less', the alternative hypothesis
- `ci` - TRUE or FALSE (default), provide confidence intervals
- `ciWidth` - a number between 50 and 99.9 (default: 95), the confidence interval width
- `bf` - TRUE or FALSE (default), provide Bayes factors
- `priorA` - a number (default: 1), the beta prior 'a' parameter
- `priorB` - a number (default: 1), the beta prior 'b' parameter
- `ciBayes` - TRUE or FALSE (default), provide Bayesian credible intervals
- `ciBayesWidth` - a number between 50 and 99.9 (default: 95), the credible interval width
- `postPlots` - TRUE or FALSE (default), provide posterior plots

## Value

A results object containing:

- `results$table` - a table of the proportions and test results
- `results$postPlots` - an array of the posterior plots
Tables can be converted to data frames with asDF or \texttt{as.data.frame}. For example:
\begin{verbatim}
results$table$asDF
as.data.frame(results$table)
\end{verbatim}

\section*{Examples}

\begin{verbatim}
## Not run:
dat <- data.frame(x=c(8, 15))

propTest2(dat, vars = x, areCounts = TRUE)

# # PROPORTION TEST (2 OUTCOMES)
# # Binomial Test
# # -------------------------------
# # Level Count Total Proportion p
# # -------------------------------
# # x 1 8 23 0.348 0.210
# # 2 15 23 0.652 0.210
# # -------------------------------
# # Note. Ha is proportion != 0.5
#
## End(Not run)
\end{verbatim}

\section*{propTestN \hspace{1cm} Proportion Test (N Outcomes)}

\subsection*{Description}

The $X^2$ Goodness of fit test (not to be confused with the $X^2$ test of independence), tests the Null hypothesis that the proportions of observations match some expected proportions. If the p-value is low, this suggests that the Null hypothesis is false, and that the true proportions are different to those tested.

\subsection*{Usage}

\begin{verbatim}
propTestN(data, var, counts = NULL, expected = FALSE, ratio = NULL, formula)
\end{verbatim}

\subsection*{Arguments}

\begin{itemize}
  \item \texttt{data} \hspace{1cm} the data as a data frame
  \item \texttt{var} \hspace{1cm} the variable of interest in data (not necessary when using a formula, see the examples)
  \item \texttt{counts} \hspace{1cm} the counts in data
\end{itemize}
propTestN

expected: TRUE or FALSE (default), whether expected counts should be displayed
ratio: a vector of numbers: the expected proportions
formula: (optional) the formula to use, see the examples

Value

A results object containing:

```
results$props  a table of the proportions
results$tests  a table of the test results
```

Tables can be converted to data frames with asDF or as.data.frame. For example:
```
results$props$asDF
as.data.frame(results$props)
```

Examples

```
data('HairEyeColor')
dat <- as.data.frame(HairEyeColor)

propTestN(formula = Freq ~ Eye, data = dat, ratio = c(1,1,1,1))
```

```
# PROPORTION TEST (N OUTCOMES)
#
# Proportions
# -------------------------
# Level  Count  Proportion
# -------------------------
# Brown  220  0.372
# Blue  215  0.363
# Hazel  93  0.157
# Green  64  0.108
# -------------------------
#
# X² Goodness of Fit
# ---------------------
# X²  df  p
# ---------------------
# 133  3  < .001
# ---------------------
```
Reliability Analysis

Usage

reliability(data, vars, alphaScale = TRUE, omegaScale = FALSE, meanScale = FALSE, sdScale = FALSE, corPlot = FALSE, alphaItems = FALSE, omegaItems = FALSE, meanItems = FALSE, sdItems = FALSE, itemRestCor = FALSE, revItems = NULL)

Arguments

data: the data as a data frame
vars: a vector of strings naming the variables of interest in data
alphaScale: TRUE (default) or FALSE, provide Cronbach’s alpha
omegaScale: TRUE or FALSE (default), provide McDonald’s omega
meanScale: TRUE or FALSE (default), provide the mean
sdScale: TRUE or FALSE (default), provide the standard deviation
corPlot: TRUE or FALSE (default), provide a correlation plot
alphaItems: TRUE or FALSE (default), provide what the Cronbach’s alpha would be if the item was dropped
omegaItems: TRUE or FALSE (default), provide what the McDonald’s omega would be if the item was dropped
meanItems: TRUE or FALSE (default), provide item means
sdItems: TRUE or FALSE (default), provide item standard deviations
itemRestCor: TRUE or FALSE (default), provide item-rest correlations
revItems: a vector containing strings naming the variables that are reverse scaled

Value

A results object containing:

- results$scale: a table
- results$items: a table
- results$corPlot: an image

Tables can be converted to data frames with asDF or as.data.frame. For example:
results$scale$asDF
as.data.frame(results$scale)

Examples

data('iris')

reliability(iris, vars = c('Sepal.Length', 'Sepal.Width', 'Petal.Length', 'Petal.Width'),
          omegaScale = TRUE)

# # RELIABILITY ANALYSIS
# # Scale Reliability Statistics
# # -----------------------------------------
# # Cronbach's alpha McDonald's omega
# # -----------------------------------------
# scale 0.708 0.848
# -----------------------------------------

---

ttestIS

**Independent Samples T-Test**

Description

The Student’s Independent samples t-test (sometimes called a two-samples t-test) is used to test the null hypothesis that two groups have the same mean. A low p-value suggests that the null hypothesis is not true, and therefore the group means are different.

Usage

ttestIS(data, vars, group, students = TRUE, bf = FALSE,
        bfPrior = 0.707, welchs = FALSE, mann = FALSE,
        hypothesis = "different", norm = FALSE, qq = FALSE, eqv = FALSE,
        meanDiff = FALSE, effectSize = FALSE, ci = FALSE, ciWidth = 95,
        desc = FALSE, plots = FALSE, miss = "perAnalysis", formula)

Arguments

data the data as a data frame
vars the dependent variables (not necessary when using a formula, see the examples)
group the grouping variable with two levels (not necessary when using a formula, see the examples)
students TRUE (default) or FALSE, perform Student’s t-tests
bf TRUE or FALSE (default), provide Bayes factors
bfPrior a number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors
welchs TRUE or FALSE (default), perform Welch’s t-tests
mann TRUE or FALSE (default), perform Mann-Whitney U tests
hypothesis 'different' (default), 'oneGreater' or 'twoGreater', the alternative hypothesis; group 1 different to group 2, group 1 greater than group 2, and group 2 greater than group 1 respectively
norm TRUE or FALSE (default), perform Shapiro-Wilk tests of normality
qq TRUE or FALSE (default), provide Q-Q plots of residuals
eqv TRUE or FALSE (default), perform Levene’s tests for equality of variances
meanDiff TRUE or FALSE (default), provide means and standard errors
effectSize TRUE or FALSE (default), provide effect sizes
ci TRUE or FALSE (default), provide confidence intervals
ciWidth a number between 50 and 99.9 (default: 95), the width of confidence intervals
desc TRUE or FALSE (default), provide descriptive statistics
plots TRUE or FALSE (default), provide descriptive plots
miss 'perAnalysis' or 'listwise'. how to handle missing values; 'perAnalysis' excludes missing values for individual dependent variables, 'listwise' excludes a row from all analyses if one of its entries is missing.
formula (optional) the formula to use, see the examples

details
The Student’s independent t-test assumes that the data from each group are from a normal distribution, and that the variances of these groups are equal. If unwilling to assume the groups have equal variances, the Welch’s t-test can be used in its place. If one is additionally unwilling to assume the data from each group are from a normal distribution, the non-parametric Mann-Whitney U test can be used instead (However, note that the Mann-Whitney U test has a slightly different null hypothesis; that the distributions of each group is equal).

value
A results object containing:

results$ttest a table containing the t-test results
results$assum$norm a table containing the normality tests
results$assum$eqv a table containing the equality of variances tests
results$desc a table containing the group descriptives
results$plots an array of groups of plots

tables can be converted to data frames with asDF or as.data.frame. For example:
results$ttest$asDF
as.data.frame(results$ttest)
Examples

data('ToothGrowth')

ttestIS(formula = len ~ supp, data = ToothGrowth)

#  # INDEPENDENT SAMPLES T-TEST
#  # Independent Samples T-Test
#  # -----------------------------------------------
#  statistic  df    p
#  -----------------------------------------------
#  len    Student's t  1.92  58.0  0.060
#  # -----------------------------------------------
#

---

ttestOneS  

One Sample T-Test

Description

The Student’s One-sample t-test is used to test the null hypothesis that the true mean is equal to a particular value (typically zero). A low p-value suggests that the null hypothesis is not true, and therefore the true mean must be different from the test value.

Usage

ttestOneS(data, vars, students = TRUE, bf = FALSE, bfPrior = 0.707, wilcoxon = FALSE, testValue = 0, hypothesis = "dt", norm = FALSE, qq = FALSE, meanDiff = FALSE, effectSize = FALSE, ci = FALSE, ciWidth = 95, desc = FALSE, plots = FALSE, miss = "perAnalysis", mann = FALSE)

Arguments

data  the data as a data frame
vars  a vector of strings naming the variables of interest in data
students  TRUE (default) or FALSE, perform Student’s t-tests
bf  TRUE or FALSE (default), provide Bayes factors
bfPrior  a number between 0.5 and 2.0 (default 0.707), the prior width to use in calculating Bayes factors
wilcoxon  TRUE or FALSE (default), perform Wilcoxon signed rank tests
testValue  a number specifying the value of the null hypothesis
hypothesis  'dt' (default), 'gt' or 'lt', the alternative hypothesis; different to testValue, greater than testValue, and less than testValue respectively
norm TRUE or FALSE (default), perform Shapiro-wilk tests of normality
qq TRUE or FALSE (default), provide a Q-Q plot of residuals
meanDiff TRUE or FALSE (default), provide means and standard deviations
effectSize TRUE or FALSE (default), provide Cohen’s d effect sizes
ci TRUE or FALSE (default), provide confidence intervals for the mean difference
ciWidth a number between 50 and 99.9 (default: 95), the width of confidence intervals
desc TRUE or FALSE (default), provide descriptive statistics
plots TRUE or FALSE (default), provide descriptive plots
miss 'perAnalysis' or 'listwise', how to handle missing values; 'perAnalysis' excludes missing values for individual dependent variables, 'listwise' excludes a row from all analyses if one of its entries is missing.
mann deprecated

Details

The Student’s One-sample t-test assumes that the data are from a normal distribution – in the case that one is unwilling to assume this, the non-parametric Wilcoxon signed-rank can be used in it’s place (However, note that the Wilcoxon signed-rank has a slightly different null hypothesis; that the *median* is equal to the test value).

Value

A results object containing:

- results$ttest a table containing the t-test results
- results$normality a table containing the normality test results
- results$descriptives a table containing the descriptives
- results$plots an image of the descriptive plots
- results$qq an array of Q-Q plots

Tables can be converted to data frames with asDF or as.data.frame. For example:

results$ttest$asDF
as.data.frame(results$ttest)

Examples

data('ToothGrowth')
ttestOneS(ToothGrowth, vars = vars(len, dose))

# # ONE SAMPLE T-TEST
# # One Sample T-Test
# # --------------------------------------
**Description**

The Student’s paired samples t-test (sometimes called a dependent-samples t-test) is used to test the null hypothesis that the difference between pairs of measurements is equal to zero. A low p-value suggests that the null hypothesis is not true, and that the difference between the measurement pairs is not zero.

**Usage**

```r
ttestPS(data, pairs, students = TRUE, bf = FALSE, bfPrior = 0.707,
    wilcoxon = FALSE, hypothesis = "different", norm = FALSE,
    qq = FALSE, meanDiff = FALSE, effectSize = FALSE, ci = FALSE,
    ciWidth = 95, desc = FALSE, plots = FALSE, miss = "perAnalysis")
```

**Arguments**

- `data`: the data as a data frame
- `pairs`: a list of lists specifying the pairs of measurement in data
- `students`: TRUE (default) or FALSE, perform Student’s t-tests
- `bf`: TRUE or FALSE (default), provide Bayes factors
- `bfPrior`: a number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors
- `wilcoxon`: TRUE or FALSE (default), perform Wilcoxon signed rank tests
- `hypothesis`: ‘different’ (default), ‘oneGreater’ or ‘twoGreater’, the alternative hypothesis; measure 1 different to measure 2, measure 1 greater than measure 2, and measure 2 greater than measure 1 respectively
- `norm`: TRUE or FALSE (default), perform Shapiro-wilk normality tests
- `qq`: TRUE or FALSE (default), provide a Q-Q plot of residuals
- `meanDiff`: TRUE or FALSE (default), provide means and standard errors
- `effectSize`: TRUE or FALSE (default), provide effect sizes
- `ci`: TRUE or FALSE (default), provide confidence intervals
- `ciWidth`: a number between 50 and 99.9 (default: 95), the width of confidence intervals
- `desc`: TRUE or FALSE (default), provide descriptive statistics
plots TRUE or FALSE (default), provide descriptive plots
miss 'perAnalysis' or 'listwise', how to handle missing values; 'perAnalysis' excludes missing values for individual dependent variables, 'listwise' excludes a row from all analyses if one of its entries is missing

Details

The Student’s paired samples t-test assumes that pair differences follow a normal distribution – in the case that one is unwilling to assume this, the non-parametric Wilcoxon signed-rank can be used in it’s place (However, note that the Wilcoxon signed-rank has a slightly different null hypothesis; that the two groups of measurements follow the same distribution).

Value

A results object containing:

- results$ttest a table containing the t-test results
- results$norm a table containing the normality test results
- results$desc a table containing the descriptives
- results$plots an array of the descriptive plots

Tables can be converted to data frames with asDF or as.data.frame. For example:
results$ttest$asDF
as.data.frame(results$ttest)

Examples

```r
## Not run:
data('bugs', package = 'jmv')
ttestPS(bugs, pairs = list(
  list(i1 = 'LDLF', i2 = 'LDHF')))

# PAIRED SAMPLES T-TEST
#
# Paired Samples T-Test
# --------------------------------------------------------------
# statistic    df  p
# --------------------------------------------------------------
# LDLF  LDHF  Student's t -6.65  90.0 < .001
# --------------------------------------------------------------
```

## End(Not run)
Index

ancova, 2
ANOVA, 4
anovaNP, 7
anovaOneW, 8
anovaRM, 10
anovaRMNP, 13
as.data.frame, 4, 6, 7, 9, 12, 13, 16, 19, 20, 22, 25, 29, 32, 35, 37, 39, 41, 43, 45–47, 49, 51, 53
bugs, 14
cfa, 14
contTables, 17
contTablesPaired, 20
corrMatrix, 21
descriptives, 23
efa, 26
linReg, 27
logLinear, 30
logRegBin, 33
logRegMulti, 35
logRegOrd, 38
mancova, 40
pca, 42
propTest2, 44
propTestN, 45
reliability, 47
ttestIS, 48
ttestOneS, 50
ttestPS, 52