Package 'walrus'

October 12, 2022

Type Package

Title Robust Statistical Methods

Version 1.0.5

Date 2022-09-07

Author Jonathon Love, Patrick Mair

Maintainer Jonathon Love <jon@thon.cc>

Description A toolbox of common robust statistical tests, including robust descriptives, robust t-tests, and robust ANOVA. It is also available as a module for 'jamovi' (see <https://www.jamovi.org> for more information). Walrus is based on the WRS2 package by Patrick Mair, which is in turn based on the scripts and work of Rand Wilcox. These analyses are described in depth in the book 'Introduction to Robust Estimation & Hypothesis Testing'.

License GPL-3

Encoding UTF-8

Imports WRS2, ggplot2, jmvcore (>= 2.2), R6

Suggests MASS

URL https://github.com/jamovi/walrus

BugReports https://github.com/jamovi/walrus/issues

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2022-09-08 03:50:02 UTC

R topics documented:

`walrus-package` *Walrus*

Description

A toolbox of common robust statistical tests, including robust descriptives, robust t-tests, and robust ANOVA. It is also available as a module for 'jamovi' (see www.jamovi.org for more information). Walrus is based on the `WRS2` package by Patrick Mair, which is in turn based on the scripts and work of Rand Wilcox. These analyses are described in depth in the book Introduction to Robust Estimation & Hypothesis Testing.

Details

Box & Violin Plots	`rplots()`
Robust Descriptives	`rdesc()`
Robust Independent Samples T-Test	`rttestIS()`
Robust Paired Samples T-Test	`rttestIS()`
Robust ANOVA	`ranova()`

```
Ravi:
    "Should we create a logo for walrus?"

Jonathon:
    "Yeah, I guess. Maybe a walrus, or a
    skewed distribution?
    Bonus points if it somehow contains both."

Ravi gets bonus points
```

See Also

Useful links:

- `https://github.com/jamovi/walrus`
- Report bugs at `https://github.com/jamovi/walrus/issues`

ranova *Robust ANOVA*

Description

Robust Analysis of Variance

Usage

```
ranova(
  data,
  dep,
  factors = NULL,
  method = "trim",
  ph = FALSE,
  tr = 0.2,
  est = "mom",
  nboot = 599,
  dist = "proj"
)
```

Arguments

data	the data as a data frame
dep	a string naming the dependent variable from data; the variable must be numeric
factors	a vector of strings naming the fixed factors from data
method	'median', 'trim' (default) or 'boot'; the method to use, median, trimmed means, or bootstrapped
ph	TRUE or FALSE (default), provide post hoc tests
tr	a number between 0 and 0.5, (default: 0.2), the proportion of measurements to trim from each end, when using the trim and bootstrap methods
est	'onestep', 'mom' (default) or 'median', the M-estimator to use; One-step, Modified one-step or Median respectively
nboot	a number (default: 599) specifying the number of bootstrap samples to use when using the bootstrap method
dist	'maha' or 'proj' (default), whether to use Mahalanobis or Projection distances respectively

Value

A results object containing:

results$main	the table of ANOVA results
results$phs	the table of posthoc tests

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('goggles', package='WRS2')

ranova(goggles,
       dep = 'attractiveness',
       factors = c('gender', 'alcohol'),
       ph = TRUE)

#
#  ROBUST ANOVA
#
#  Robust ANOVA
#  -----------------------------------
#                      Q        p
#  -----------------------------------
#    gender             1.67    0.209
#    alcohol           48.28    0.001
#    gender:alcohol    26.26    0.001
#  -----------------------------------
#   Note. Method of trimmed means,
#   trim level 0.2
#
#
#  POST HOC TESTS
#
#  Post Hoc Tests - gender
#  ------------------------------------------------------------
#                       psi-hat   p        Lower    Upper
#  ------------------------------------------------------------
#    Female    Male     10.0      0.209    -6.00    26.0
#  ------------------------------------------------------------
#
#
#  Post Hoc Tests - alcohol
#  ------------------------------------------------------------------
#                           psi-hat   p         Lower    Upper
#  ------------------------------------------------------------------
#    None       2 Pints     -3.33      0.611    -20.5     13.8
#    None       4 Pints     35.83     < .001     19.3     52.3
#    2 Pints    4 Pints     39.17     < .001     22.5     55.9
#  ------------------------------------------------------------------
#
```

rdesc *Robust Descriptives*

Description

Robust Descriptives

Usage

```
rdesc(
  data,
  vars,
  splitBy = NULL,
  mean = TRUE,
  trim = TRUE,
  tr = 0.2,
  win = FALSE,
  wl = 0.2,
  mest = FALSE,
  bend = 1.28,
  med = FALSE
)
```

Arguments

data	the data as a data frame
vars	a vector of strings naming the variables in data of interest
splitBy	a string naming the variable in data to split the data by
mean	TRUE (default) or FALSE, provide a 'normal' arithmetic mean
trim	TRUE (default) or FALSE, provide a trimmed mean
tr	a number between 0 and 0.5 (default: 0.2); the proportion of measurements to trim from each end when producing trimmed means
win	TRUE or FALSE (default), provide a 'Winsorized' mean
wl	a number between 0 and 0.5 (default: 0.2); the level of 'winsorizing' when producing winsorized means
mest	TRUE or FALSE (default), provide an 'M-estimated' value
bend	a number (default: 1.28), the bending constant to use when using M-estimators
med	TRUE or FALSE (default), provide medians

Value

A results object containing:

results$table	the table of descriptives

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('eurosoccer', package='WRS2')

SpainGermany <- subset(eurosoccer, eurosoccer$League == 'Spain' | eurosoccer$League == 'Germany')
SpainGermany <- droplevels(SpainGermany)

walrus::rdesc(
    data = SpainGermany,
    vars = "GoalsGame",
    splitBy = "League",
    med = TRUE)

#
#  ROBUST DESCRIPTIVES
#
#  Robust Descriptives
#  ------------------------------------------------------------
#                                                        SE
#  ------------------------------------------------------------
#    GoalsGame    Germany    Mean            1.46     0.105
#                            Trimmed mean    1.45    0.1341
#                            Median          1.43    0.1599
#
#                 Spain      Mean            1.45     0.101
#                            Trimmed mean    1.33    0.0601
#                            Median          1.30    0.0766
#  ------------------------------------------------------------
#
```

---

rplots *Box & Violin Plots*

Description

Box & Violin Plots

Usage

```
rplots(
  data,
  vars,
  splitBy = NULL,
```

```
  violin = TRUE,
  boxplot = FALSE,
  dot = TRUE,
  dotType = "stack"
)
```

Arguments

data	the data as a data frame
vars	a vector of strings naming the variables in data of interest
splitBy	a string naming the variable in data to split the data by
violin	TRUE (default) or FALSE, provide violin plots
boxplot	TRUE or FALSE (default), provide box plots
dot	TRUE (default) or FALSE, plot each measurement as a dot
dotType	'jitter' or 'stack' (default); whether data dots are jittered or stacked

Value

A results object containing:

results$plots	an array of images

Examples

```
data('eurosoccer', package='WRS2')

# violin plots

walrus::rplots(
    data = eurosoccer,
    vars = "GoalsGame",
    splitBy = "League")


# box plots

walrus::rplots(
    data = eurosoccer,
    vars = "GoalsGame",
    splitBy = "League",
    violin = FALSE,
    boxplot = TRUE,
    dot = FALSE)
```

`rttestIS` *Robust Independent Samples T-Test*

Description

Robust Independent Samples T-Test

Usage

```
rttestIS(
  data,
  deps,
  group,
  yuen = TRUE,
  tr = 0.2,
  mest = FALSE,
  method = "mom",
  yuenbt = FALSE,
  nboot = 599,
  md = FALSE,
  ci = FALSE,
  es = FALSE,
  esci = FALSE
)
```

Arguments

`data`	the data as a data frame
`deps`	a vector of strings naming the dependent variables in `data`
`group`	a string naming the grouping variable in `data`; must have 2 levels
`yuen`	`TRUE` (default) or `FALSE`, use the Yuen's trim method
`tr`	a number between 0 and 0.5, (default: 0.2), the proportion of measurements to trim from each end, when using the trim and bootstrap methods
`mest`	`TRUE` or `FALSE` (default), use an M-estimator
`method`	`'onestep'`, `'mom'` (default) or `'median'`, the M-estimator to use; One-step, Modified one-step or Median respectively
`yuenbt`	`TRUE` or `FALSE` (default), use the Yuen's bootstrap method
`nboot`	a number (default: 599) specifying the number of bootstrap samples to use when using the bootstrap method
`md`	`TRUE` or `FALSE` (default), provide the mean difference
`ci`	`TRUE` or `FALSE` (default), provide a 95% confidence interval on the mean difference
`es`	`TRUE` or `FALSE` (default), provide the effect-size
`esci`	`TRUE` or `FALSE` (default), provide a 95% confidence interval on the effect-size

Value

A results object containing:

`results$ttest`	the table of t-test results

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('eurosoccer', package='WRS2')

SpainGermany <- subset(eurosoccer, eurosoccer$League == 'Spain' | eurosoccer$League == 'Germany')
SpainGermany <- droplevels(SpainGermany)

rttestIS(SpainGermany,
         dep = 'GoalsScored',
         group = 'League',
         yuen = TRUE,
         mest = TRUE)

#
#  ROBUST INDEPENDENT SAMPLES T-TEST
#
#  Robust Independent Samples T-Test
#  ------------------------------------------------------
#                                 t         df      p
#  ------------------------------------------------------
#    GoalsScored    Yuen's test     0.297    17.3    0.770
#                   M-estimator    -0.933            0.993
#  ------------------------------------------------------
#
```

`rttestPS` *Robust Paired Samples T-Test*

Description

Robust Paired Samples T-Test

Usage

```
rttestPS(data, pairs, tr = 0.2, md = FALSE, es = FALSE, ci = FALSE)
```

Arguments

data	the data as a data frame
pairs	a list of lists specifying the pairs of measurement in data
tr	a number between 0 and 0.5, (default: 0.2), the proportion of measurements to trim from each end, when using the trim and bootstrap methods
md	TRUE or FALSE (default), provide means and standard errors
es	TRUE or FALSE (default), provide effect sizes
ci	TRUE or FALSE (default), provide confidence intervals

Value

A results object containing:

results$ttest	the table of t-test results

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(anorexia, package='MASS')
anorexiaFT <- subset(anorexia, subset = Treat == "FT")

rttestPS(anorexiaFT,
         pairs = list(
             list(i1 = 'Prewt', i2 = 'Postwt')))

#
#  ROBUST PAIRED SAMPLES T-TEST
#
#  Robust Paired Samples T-Test
#  ---------------------------------------------
#                       t        df      p
#  ---------------------------------------------
#    Prewt    Postwt    -3.83    10.0    0.003
#  ---------------------------------------------
#
```

Index