

Package: webr (via r-universe)

September 15, 2024

Type Package

Title Data and Functions for Web-Based Analysis

Version 0.1.6

Imports moonBook, ggplot2, shiny, stringr, sjlabelled, flextable, magrittr, rrtable, dplyr, tibble, purrr, rlang, tidyr, tidyselect, psych, grid, ztable, ggforce, scales, vcd

URL <https://github.com/cardiomoon/webr>

BugReports <https://github.com/cardiomoon/webr/issues>

Description Several analysis-related functions for the book entitled "Web-based Analysis without R in Your Computer"(written in Korean, ISBN 978-89-5566-185-9) by Keon-Woong Moon. The main function plot.htest() shows the distribution of statistic for the object of class 'htest'.

Depends R (>= 2.10)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

VignetteBuilder knitr

Suggests testthat, knitr, rmarkdown

Repository <https://cardiomoon.r-universe.dev>

RemoteUrl <https://github.com/cardiomoon/webr>

RemoteRef HEAD

RemoteSha 6e411bdc671f061a554512e614fdcc424e780dc4

Contents

BiVar	2
ContinuousVar	3

cox.stuart.test	3
extractLabels	4
freqSummary	4
freqTable	5
gg_color_hue	5
GroupVar	6
langchoice1	6
makeSub	7
makeSubColor	7
mychisq.test	7
numSummary	8
numSummaryTable	9
PieDonut	9
plot.htest	12
renew_dic	13
replaceWithLabels	13
runs.test	14
transparent	14
x2result	15
x2summary	15
x2Table	16
Index	17

 BiVar

Extract bivariate variables

Description

Extract bivariate variables

Usage

BiVar(df)

Arguments

df a data.frame

ContinuousVar	<i>Extract continuous variables</i>
---------------	-------------------------------------

Description

Extract continuous variables

Usage

```
ContinuousVar(df)
```

Arguments

df	a data.frame
----	--------------

cox.stuart.test	<i>Cox-Stuart test for trend analysis The Cox-Stuart test is defined as a little powerful test (power equal to 0.78), but very robust for the trend analysis. It is therefore applicable to a wide variety of situations, to get an idea of the evolution of values obtained. The proposed method is based on the binomial distribution. This function was written by Tommaso Martino<todoslogos@gmail.com> (See 'References')</i>
-----------------	--

Description

Cox-Stuart test for trend analysis The Cox-Stuart test is defined as a little powerful test (power equal to 0.78), but very robust for the trend analysis. It is therefore applicable to a wide variety of situations, to get an idea of the evolution of values obtained. The proposed method is based on the binomial distribution. This function was written by Tommaso Martino<todoslogos@gmail.com> (See 'References')

Usage

```
cox.stuart.test(x)
```

Arguments

x	A numeric vector
---	------------------

Value

A list with class "htest"

References

Original code: <http://statistic-on-air.blogspot.kr/2009/08/trend-analysis-with-cox-stuart-test-in.html>

Examples

```
customers = c(5, 9, 12, 18, 17, 16, 19, 20, 4, 3, 18, 16, 17, 15, 14)
cox.stuart.test(customers)
```

extractLabels	<i>Extract labels</i>
---------------	-----------------------

Description

Extract labels

Usage

```
extractLabels(x)
```

Arguments

x	a vector
---	----------

freqSummary	<i>Make table summarizing frequency</i>
-------------	---

Description

Make table summarizing frequency

Usage

```
freqSummary(x, digits = 1, lang = "en")
```

Arguments

x	A vector
digits	integer indicating the number of decimal places
lang	Language. choices are one of c("en","kor")

Examples

```
require(moonBook)
freqSummary(acs$Dx)
#freqSummary(acs$smoking, lang="kor")
```

freqTable	<i>Make flextable summarizing frequency</i>
-----------	---

Description

Make flextable summarizing frequency

Usage

```
freqTable(  
  x,  
  digits = 1,  
  lang = getOption("freqTable.lang", "en"),  
  vanilla = FALSE,  
  ...  
)
```

Arguments

x	A vector
digits	integer indicating the number of decimal places
lang	Language. choices are one of c("en","kor")
vanilla	Logical. Whether make vanilla table or not
...	Further arguments to paseed to the df2flextable function

Value

An object of class flextable

Examples

```
require(moonBook)  
freqTable(acs$Dx)  
#freqTable(acs$smoking, lang="kor", vanilla=TRUE, fontsize=12)
```

gg_color_hue	<i>Make default palette</i>
--------------	-----------------------------

Description

Make default palette

Usage

```
gg_color_hue(n)
```

Arguments

n number of colors

GroupVar *Extract categorical variables*

Description

Extract categorical variables

Usage

```
GroupVar(df, max.ylev = 20)
```

Arguments

df a data.frame
max.ylev maximal length of unique values of categorical variables

langchoice1 *Select word*

Description

Select word

Usage

```
langchoice1(id, lang = "en")
```

Arguments

id data id
lang language. Possible choices are c("en","kor")

makeSub	<i>Make subtitle</i>
---------	----------------------

Description

Make subtitle

Usage

```
makeSub(x)
```

Arguments

x	An object of class "htest"
---	----------------------------

makeSubColor	<i>Make subcolors with main colors</i>
--------------	--

Description

Make subcolors with main colors

Usage

```
makeSubColor(main, no = 3)
```

Arguments

main	character. main colors
no	number of subcolors

mychisq.test	<i>My chisquare test</i>
--------------	--------------------------

Description

My chisquare test

Usage

```
mychisq.test(x)
```

Arguments

x	a table
---	---------

numSummary	<i>Numerical Summary</i>
------------	--------------------------

Description

Numerical Summary

Usage

```
numSummary(x, ..., digits = 2, lang = "en")
```

```
numSummary1(x, ..., digits = 2, lang = "en")
```

```
numSummary2(x, ..., digits = 2, lang = "en")
```

Arguments

x	A numeric vector or a data.frame or a grouped_df
...	further arguments to be passed
digits	integer indicating the number of decimal places
lang	Language. choices are one of c("en","kor")

Functions

- numSummary1: Numerical Summary of a data.frame or a vector
- numSummary2: Numerical Summary of a grouped_df

Examples

```
require(moonBook)
require(magrittr)
require(dplyr)
require(rtable)
require(webr)
require(tibble)
numSummary(acs)
numSummary(acs$age)
numSummary(acs, age, EF)
acs %>% group_by(sex) %>% numSummary(age, BMI)
acs %>% group_by(sex) %>% select(age) %>% numSummary
acs %>% group_by(sex) %>% select(age, EF) %>% numSummary
acs %>% group_by(sex, Dx) %>% select(age, EF) %>% numSummary
acs %>% group_by(sex, Dx) %>% select(age) %>% numSummary
#acs %>% group_by(sex, Dx) %>% numSummary(age, EF, lang="kor")
```

numSummaryTable	<i>Make a table showing numerical summary</i>
-----------------	---

Description

Make a table showing numerical summary

Usage

```
numSummaryTable(
  x,
  ...,
  lang = getOption("numSummaryTable.lang", "en"),
  vanilla = FALSE,
  add.rownames = NULL
)
```

Arguments

x	A grouped_df or a data.frame or a vector
...	further argument to be passed
lang	Language. choices are one of c("en","kor")
vanilla	Logical. Whether make vanilla table or not
add.rownames	Logical. Whether or not add rownames

Examples

```
require(moonBook)
require(dplyr)
numSummaryTable(acs)
numSummaryTable(acs$age)
acs %>% group_by(sex) %>% select(age) %>% numSummaryTable
acs %>% group_by(sex) %>% select(age,EF) %>% numSummaryTable
acs %>% group_by(sex,Dx) %>% select(age,EF) %>% numSummaryTable(vanilla=FALSE)
acs %>% group_by(sex,Dx) %>% numSummaryTable(age,EF,add.rownames=FALSE)
```

PieDonut	<i>Draw a PieDonut plot</i>
----------	-----------------------------

Description

Draw a PieDonut plot

Usage

```

PieDonut(
  data,
  mapping,
  start = getOption("PieDonut.start", 0),
  addPieLabel = TRUE,
  addDonutLabel = TRUE,
  showRatioDonut = TRUE,
  showRatioPie = TRUE,
  ratioByGroup = TRUE,
  showRatioThreshold = getOption("PieDonut.showRatioThreshold", 0.02),
  labelposition = getOption("PieDonut.labelposition", 2),
  labelpositionThreshold = 0.1,
  r0 = getOption("PieDonut.r0", 0.3),
  r1 = getOption("PieDonut.r1", 1),
  r2 = getOption("PieDonut.r2", 1.2),
  explode = NULL,
  selected = NULL,
  explodePos = 0.1,
  color = "white",
  pieAlpha = 0.8,
  donutAlpha = 1,
  maxx = NULL,
  showPieName = TRUE,
  showDonutName = FALSE,
  title = NULL,
  pieLabelSize = 4,
  donutLabelSize = 3,
  titlesize = 5,
  explodePie = TRUE,
  explodeDonut = FALSE,
  use.label = TRUE,
  use.labels = TRUE,
  family = getOption("PieDonut.family", "")
)

```

Arguments

<code>data</code>	A data.frame
<code>mapping</code>	Set of aesthetic mappings created by <code>aes</code> or <code>aes_</code> .
<code>start</code>	offset of starting point from 12 o'clock in radians
<code>addPieLabel</code>	A logical value. If TRUE, labels are added to the Pies
<code>addDonutLabel</code>	A logical value. If TRUE, labels are added to the Donuts
<code>showRatioDonut</code>	A logical value. If TRUE, ratios are added to the DonutLabels
<code>showRatioPie</code>	A logical value. If TRUE, ratios are added to the PieLabels
<code>ratioByGroup</code>	A logical value. If TRUE, ratios are calculated per group

showRatioThreshold	An integer. Threshold to show label as a ratio of total. default value is 0.02.
labelposition	A number indicating the label position
labelpositionThreshold	label position threshold. Default value is 0.1.
r0	Integer. start point of pie
r1	Integer. end point of pie
r2	Integer. end point of donut
explode	pies to explode
selected	donuts to explode
explodePos	explode position
color	color
pieAlpha	transparency of pie
donutAlpha	transparency of pie
maxx	maximum position of plot
showPieName	logical. Whether or not show Pie Name
showDonutName	logical. Whether or not show Pie Name
title	title of plot
pieLabelSize	integer. Pie label size
donutLabelSize	integer. Donut label size
titlesize	integer. Title size
explodePie	Logical. Whether or not explode pies
explodeDonut	Logical. Whether or not explode donuts
use.label	Logical. Whether or not use column label in case of labelled data
use.labels	Logical. Whether or not use value labels in case of labelled data
family	font family

Examples

```

require(moonBook)
require(ggplot2)
browser=c("MSIE","Firefox","Chrome","Safari","Opera")
share=c(50,21.9,10.8,6.5,1.8)
df=data.frame(browser,share)
PieDonut(df,aes(browser,count=share),r0=0.7,start=3*pi/2,labelpositionThreshold=0.1)

PieDonut(df,aes(browser,count=share),r0=0.7,explode=5,start=3*pi/2)
PieDonut(mtcars,aes(gear,carb),start=3*pi/2,explode=3,explodeDonut=TRUE,maxx=1.7)
PieDonut(mtcars,aes(carb,gear),r0=0)
PieDonut(acs,aes(smoking,Dx),title="Distribution of smoking status by diagnosis")
PieDonut(acs,aes(Dx,smoking),ratioByGroup=FALSE,r0=0)
PieDonut(acs,aes(Dx,smoking),selected=c(1,3,5,7),explodeDonut=TRUE)
PieDonut(acs,aes(Dx,smoking),explode=1,selected=c(2,4,6,8),labelposition=0,explodeDonut=TRUE)

```

```

PieDonut(acs,aes(Dx,smoking),explode=1)
PieDonut(acs,aes(Dx,smoking),explode=1,explodeDonut=TRUE,labelposition=0)
PieDonut(acs,aes(Dx,smoking),explode=1,explodePie=FALSE,explodeDonut=TRUE,labelposition=0)
PieDonut(acs,aes(Dx,smoking),selected=c(2,5,8), explodeDonut=TRUE,start=pi/2,labelposition=0)
PieDonut(acs,aes(Dx,smoking),r0=0.2,r1=0.9,r2=1.3,explode=1,start=pi/2,explodeDonut=TRUE)
PieDonut(acs,aes(Dx,smoking),r0=0.2,r1=0.9,r2=1.3,explode=1,start=pi/2,labelposition=0)
PieDonut(acs,aes(Dx,smoking),explode=1,start=pi,explodeDonut=TRUE,labelposition=0)
require(dplyr)
df=mtcars %>% group_by(gear,carb) %>% summarize(n=n())
PieDonut(df,aes(pies=gear,donuts=carb,count=n),ratioByGroup=FALSE)

```

plot.htest

Plotting distribution of statistic for object "htest"

Description

Plotting distribution of statistic for object "htest"

Usage

```

## S3 method for class 'htest'
plot(x, ...)

```

Arguments

x	object of class "htest"
...	further arguments to ggplot

Value

a ggplot or NULL

Examples

```

require(moonBook)
require(webr)
## chi-square test
x=chisq.test(table(mtcars$am,mtcars$cyl))
plot(x)

#Welch Two Sample t-test
x=t.test(mpg~am,data=mtcars)
plot(x)

x=t.test(BMI~sex,data=acs)
plot(x)

# F test to compare two variances

```

```

x=var.test(age~sex,data=acs,alternative="less")
plot(x)

# Paired t-test
x=t.test(iris$Sepal.Length,iris$Sepal.Width,paired=TRUE)
plot(x)

# One sample t-test
plot(t.test(acs$age,mu=63))

# Two sample t-test
x=t.test(age~sex, data=acs,conf.level=0.99,alternative="greater",var.equal=TRUE)
plot(x)

```

renew_dic

Renew dictionary Renew dictionary

Description

Renew dictionary Renew dictionary

Usage

```
renew_dic()
```

replaceWithLabels

replace vector with labels

Description

replace vector with labels

Usage

```
replaceWithLabels(x)
```

Arguments

x A vector

runs.test	<i>Runs test for randomness</i>
-----------	---------------------------------

Description

Runs test for randomness

Usage

```
runs.test(
  y,
  plot.it = FALSE,
  alternative = c("two.sided", "positive.correlated", "negative.correlated")
)
```

Arguments

y	A vector
plot.it	A logical. whether or not draw a plot
alternative	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".

Value

A list with class "htest" containing the following components: statistic,p-value,method and data.name

Examples

```
y=c(1,2,2,1,1,2,1,2)
runs.test(y)
y=c("A","B","B","A","A","B","A","B")
runs.test(y,alternative="p")
```

transparent	<i>Make transparent theme</i>
-------------	-------------------------------

Description

Make transparent theme

Usage

```
transparent(size = 0)
```

Arguments

size	border size. default value is 0
------	---------------------------------

x2result	<i>Extract x2 statistical result</i>
----------	--------------------------------------

Description

Extract x2 statistical result

Usage

```
x2result(x)
```

Arguments

x	a table
---	---------

x2summary	<i>Summarize chisquare result</i>
-----------	-----------------------------------

Description

Summarize chisquare result

Usage

```
x2summary(
  data = NULL,
  x = NULL,
  y = NULL,
  a,
  b,
  margin = 1,
  show.percent = TRUE,
  show.label = TRUE
)
```

Arguments

data	A data.frame
x	a column name
y	a column name
a	a vector
b	a vector
margin	numeric If 1 row percent, if 2 col percent
show.percent	logical
show.label	logical

Examples

```
require(moonBook)
x2summary(acs, sex, DM)
```

x2Table	<i>Make a chisquare result table</i>
---------	--------------------------------------

Description

Make a chisquare result table

Usage

```
x2Table(
  data,
  x,
  y,
  margin = 1,
  show.percent = TRUE,
  show.label = TRUE,
  show.stat = TRUE,
  vanilla = FALSE,
  fontsize = 12,
  ...
)
```

Arguments

data	A data.frame
x	a column name
y	a column name
margin	numeric If 1 row percent, if 2 col percent
show.percent	logical
show.label	logical
show.stat	logical
vanilla	logical whether or not make vanilla table
fontsize	A numeric
...	Further arguments to be passed to df2flextable()

Examples

```
require(moonBook)
x2Table(acs, sex, DM)
```


Index

BiVar, [2](#)

ContinuousVar, [3](#)
cox.stuart.test, [3](#)

extractLabels, [4](#)

freqSummary, [4](#)
freqTable, [5](#)

gg_color_hue, [5](#)
GroupVar, [6](#)

langchoice1, [6](#)

makeSub, [7](#)
makeSubColor, [7](#)
mychisq.test, [7](#)

numSummary, [8](#)
numSummary1 (numSummary), [8](#)
numSummary2 (numSummary), [8](#)
numSummaryTable, [9](#)

PieDonut, [9](#)
plot.htest, [12](#)

renew_dic, [13](#)
replaceWithLabels, [13](#)
runs.test, [14](#)

transparent, [14](#)

x2result, [15](#)
x2summary, [15](#)
x2Table, [16](#)