

Package ‘radiant.basics’

July 30, 2019

Type Package

Title Basics Menu for Radiant: Business Analytics using R and Shiny

Version 1.0.0

Date 2019-7-29

Description The Radiant Basics menu includes interfaces for probability calculation, central limit theorem simulation, comparing means and proportions, goodness-of-fit testing, cross-tabs, and correlation. The application extends the functionality in radiant.data.

Depends R (>= 3.4.0), radiant.data (>= 1.0.0)

Imports ggplot2 (>= 2.2.1), gridExtra (>= 2.0.0), scales (>= 0.4.0), dplyr (>= 0.8.3), tidyr (>= 0.8.2), magrittr (>= 1.5), shiny (>= 1.2.0), psych (>= 1.8.3.3), import (>= 1.1.0)

Suggests testthat (>= 2.0.0), pkgdown (>= 1.1.0)

URL <https://github.com/radiant-rstats/radiant.basics>,
<https://radiant-rstats.github.io/radiant.basics>,
<https://radiant-rstats.github.io/docs>

BugReports <https://github.com/radiant-rstats/radiant.basics/issues>

License AGPL-3 | file LICENSE

LazyData true

Encoding UTF-8

Language en-US

RoxygenNote 6.1.1

NeedsCompilation no

Author Vincent Nijs [aut, cre]

Maintainer Vincent Nijs <radiant@rady.ucsd.edu>

Repository CRAN

Date/Publication 2019-07-30 04:40:06 UTC

R topics documented:

clt	2
compare_means	3
compare_props	4
consider	5
correlation	5
cross_tabs	6
demand_uk	7
goodness	7
newspaper	8
plot.clt	9
plot.compare_means	9
plot.compare_props	10
plot.correlation	11
plot.cross_tabs	12
plot.goodness	13
plot.prob_binom	14
plot.prob_chisq	15
plot.prob_disc	15
plot.prob_expo	16
plot.prob_fdist	17
plot.prob_lnorm	17
plot.prob_norm	18
plot.prob_pois	19
plot.prob_tdist	19
plot.prob_unif	20
plot.single_mean	21
plot.single_prop	22
print.rcorr	23
prob_binom	23
prob_chisq	24
prob_disc	24
prob_expo	25
prob_fdist	26
prob_lnorm	27
prob_norm	27
prob_pois	28
prob_tdist	29
prob_unif	30
radiant.basics	30
radiant.basics_viewer	31
radiant.basics_window	32
salary	32
single_mean	33
single_prop	34
summary.compare_means	35
summary.compare_props	36

summary.correlation	37
summary.cross_tabs	38
summary.goodness	39
summary.prob_binom	40
summary.prob_chisq	40
summary.prob_disc	41
summary.prob_expo	42
summary.prob_fdist	42
summary.prob_lnorm	43
summary.prob_norm	44
summary.prob_pois	44
summary.prob_tdist	45
summary.prob_unif	46
summary.single_mean	46
summary.single_prop	47

 clt

Central Limit Theorem simulation

Description

Central Limit Theorem simulation

Usage

```
clt(dist, n = 100, m = 100, norm_mean = 0, norm_sd = 1,
    binom_size = 10, binom_prob = 0.2, unif_min = 0, unif_max = 1,
    expo_rate = 1)
```

Arguments

dist	Distribution to simulate
n	Sample size
m	Number of samples
norm_mean	Mean for the normal distribution
norm_sd	Standard deviation for the normal distribution
binom_size	Size for the binomial distribution
binom_prob	Probability for the binomial distribution
unif_min	Minimum for the uniform distribution
unif_max	Maximum for the uniform distribution
expo_rate	Rate for the exponential distribution

Details

See <https://radiant-rstats.github.io/docs/basics/clt.html> for an example in Radiant

Value

A list with the name of the Distribution and a matrix of simulated data

Examples

```
clt("Uniform", 10, 10, unif_min = 10, unif_max = 20)
```

<code>compare_means</code>	<i>Compare sample means</i>
----------------------------	-----------------------------

Description

Compare sample means

Usage

```
compare_means(dataset, var1, var2, samples = "independent",
  alternative = "two.sided", conf_lev = 0.95, comb = "",
  adjust = "none", test = "t", data_filter = "")
```

Arguments

<code>dataset</code>	Dataset
<code>var1</code>	A numeric variable or factor selected for comparison
<code>var2</code>	One or more numeric variables for comparison. If <code>var1</code> is a factor only one variable can be selected and the mean of this variable is compared across (factor) levels of <code>var1</code>
<code>samples</code>	Are samples independent ("independent") or not ("paired")
<code>alternative</code>	The alternative hypothesis ("two.sided", "greater" or "less")
<code>conf_lev</code>	Span of the confidence interval
<code>comb</code>	Combinations to evaluate
<code>adjust</code>	Adjustment for multiple comparisons ("none" or "bonf" for Bonferroni)
<code>test</code>	t-test ("t") or Wilcox ("wilcox")
<code>data_filter</code>	Expression entered in, e.g., <code>Data > View</code> to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

See https://radiant-rstats.github.io/docs/basics/compare_means.html for an example in Radiant

Value

A list of all variables defined in the function as an object of class `compare_means`

See Also

`summary.compare_means` to summarize results
`plot.compare_means` to plot results

Examples

```
compare_means(diamonds, "cut", "price") %>% str()
```

<code>compare_props</code>	<i>Compare sample proportions across groups</i>
----------------------------	---

Description

Compare sample proportions across groups

Usage

```
compare_props(dataset, var1, var2, levs = "",
              alternative = "two.sided", conf_lev = 0.95, comb = "",
              adjust = "none", data_filter = "")
```

Arguments

<code>dataset</code>	Dataset
<code>var1</code>	A grouping variable to split the data for comparisons
<code>var2</code>	The variable to calculate proportions for
<code>levs</code>	The factor level selected for the proportion comparison
<code>alternative</code>	The alternative hypothesis ("two.sided", "greater" or "less")
<code>conf_lev</code>	Span of the confidence interval
<code>comb</code>	Combinations to evaluate
<code>adjust</code>	Adjustment for multiple comparisons ("none" or "bonf" for Bonferroni)
<code>data_filter</code>	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

See https://radiant-rstats.github.io/docs/basics/compare_props.html for an example in Radiant

Value

A list of all variables defined in the function as an object of class `compare_props`

See Also

`summary.compare_props` to summarize results
`plot.compare_props` to plot results

Examples

```
compare_props(titanic, "pclass", "survived") %>% str()
```

<code>consider</code>	<i>Car brand consideration</i>
-----------------------	--------------------------------

Description

Car brand consideration

Usage

```
data(consider)
```

Format

A data frame with 1000 rows and 2 variables

Details

Survey data of consumer purchase intentions. Description provided in `attr(consider,"description")`

<code>correlation</code>	<i>Calculate correlations for two or more variables</i>
--------------------------	---

Description

Calculate correlations for two or more variables

Usage

```
correlation(dataset, vars = "", method = "pearson", data_filter = "")
```

Arguments

<code>dataset</code>	Dataset
<code>vars</code>	Variables to include in the analysis. Default is all but character and factor variables with more than two unique values are removed
<code>method</code>	Type of correlations to calculate. Options are "pearson", "spearman", and "kendall". "pearson" is the default
<code>data_filter</code>	Expression entered in, e.g., <code>Data > View</code> to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

See <https://radiant-rstats.github.io/docs/basics/correlation.html> for an example in Radiant

Value

A list with all variables defined in the function as an object of class `compare_means`

See Also

`summary.correlation` to summarize results

`plot.correlation` to plot results

Examples

```
correlation(diamonds, c("price", "carat")) %>% str()
correlation(diamonds, "x:z") %>% str()
```

cross_tabs

Evaluate associations between categorical variables

Description

Evaluate associations between categorical variables

Usage

```
cross_tabs(dataset, var1, var2, tab = NULL, data_filter = "")
```

Arguments

dataset	Dataset (i.e., a data.frame or table)
var1	A categorical variable
var2	A categorical variable
tab	Table with frequencies as alternative to dataset
data_filter	Expression entered in, e.g., <code>Data > View</code> to filter the dataset in Radiant. The expression should be a string (e.g., <code>"price > 10000"</code>)

Details

See https://radiant-rstats.github.io/docs/basics/cross_tabs.html for an example in Radiant

Value

A list of all variables used in `cross_tabs` as an object of class `cross_tabs`

See Also

`summary.cross_tabs` to summarize results

`plot.cross_tabs` to plot results

Examples

```
cross_tabs(newspaper, "Income", "Newspaper") %>% str()
table(select(newspaper, Income, Newspaper)) %>% cross_tabs(tab = .)
```

demand_uk	<i>Demand in the UK</i>
-----------	-------------------------

Description

Demand in the UK

Usage

```
data(demand_uk)
```

Format

A data frame with 1000 rows and 2 variables

Details

Survey data of consumer purchase intentions. Description provided in `attr(demand_uk,"description")`

goodness	<i>Evaluate if sample data for a categorical variable is consistent with a hypothesized distribution</i>
----------	--

Description

Evaluate if sample data for a categorical variable is consistent with a hypothesized distribution

Usage

```
goodness(dataset, var, p = NULL, tab = NULL, data_filter = "")
```


Arguments

dataset	Dataset
var	A categorical variable
p	Hypothesized distribution as a number, fraction, or numeric vector. If unspecified, defaults to an even distribution
tab	Table with frequencies as alternative to dataset
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

See <https://radiant-rstats.github.io/docs/basics/goodness.html> for an example in Radiant

Value

A list of all variables used in goodness as an object of class goodness

See Also

`summary.goodness` to summarize results
`plot.goodness` to plot results

Examples

```
goodness(newspaper, "Income") %>% str()
table(select(newspaper, Income)) %>% goodness(tab = .)
```

newspaper	<i>Newspaper readership</i>
-----------	-----------------------------

Description

Newspaper readership

Usage

```
data(newspaper)
```

Format

A data frame with 580 rows and 2 variables

Details

Newspaper readership data for 580 consumers. Description provided in `attr(newspaper,"description")`

`plot.clt`*Plot method for the Central Limit Theorem simulation*

Description

Plot method for the Central Limit Theorem simulation

Usage

```
## S3 method for class 'clt'  
plot(x, stat = "sum", bins = 15, ...)
```

Arguments

<code>x</code>	Return value from <code>clt</code>
<code>stat</code>	Statistic to use (sum or mean)
<code>bins</code>	Number of bins to use
<code>...</code>	further arguments passed to or from other methods

Details

See <https://radiant-rstats.github.io/docs/basics/clt.html> for an example in Radiant

Examples

```
clt("Uniform", 100, 100, unif_min = 10, unif_max = 20) %>% plot()
```

`plot.compare_means` *Plot method for the compare_means function*

Description

Plot method for the `compare_means` function

Usage

```
## S3 method for class 'compare_means'  
plot(x, plots = "scatter", shiny = FALSE,  
      custom = FALSE, ...)
```

Arguments

x	Return value from <code>compare_means</code>
plots	One or more plots ("bar", "density", "box", or "scatter")
shiny	Did the function call originate inside a shiny app
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org for options.
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/compare_means.html for an example in Radiant

See Also

`compare_means` to calculate results
`summary.compare_means` to summarize results

Examples

```
result <- compare_means(diamonds, "cut", "price")  
plot(result, plots = c("bar", "density"))
```

`plot.compare_props` *Plot method for the compare_props function*

Description

Plot method for the `compare_props` function

Usage

```
## S3 method for class 'compare_props'  
plot(x, plots = "bar", shiny = FALSE,  
      custom = FALSE, ...)
```

Arguments

x	Return value from <code>compare_props</code>
plots	One or more plots of proportions ("bar" or "dodge")
shiny	Did the function call originate inside a shiny app
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org for options.
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/compare_props.html for an example in Radiant

See Also

`compare_props` to calculate results
`summary.compare_props` to summarize results

Examples

```
result <- compare_props(titanic, "pclass", "survived")
plot(result, plots = c("bar", "dodge"))
```

`plot.correlation` *Plot method for the correlation function*

Description

Plot method for the correlation function

Usage

```
## S3 method for class 'correlation'
plot(x, nrobs = -1, jit = c(0, 0), dec = 2,
     ...)
```

Arguments

x	Return value from <code>correlation</code>
nrobs	Number of data points to show in scatter plots (-1 for all)
jit	A numeric vector that determines the amount of jittering to apply to scatter plot. Default is 0. Use, e.g., 0.3 to add some jittering
dec	Number of decimals to show
...	further arguments passed to or from other methods.

Details

See <https://radiant-rstats.github.io/docs/basics/correlation.html> for an example in Radiant

See Also

`correlation` to calculate results

`summary.correlation` to summarize results

Examples

```
result <- correlation(diamonds, c("price", "carat", "table"))
plot(result)
```

`plot.cross_tabs` *Plot method for the cross_tabs function*

Description

Plot method for the `cross_tabs` function

Usage

```
## S3 method for class 'cross_tabs'
plot(x, check = "", shiny = FALSE,
     custom = FALSE, ...)
```

Arguments

<code>x</code>	Return value from <code>cross_tabs</code>
<code>check</code>	Show plots for variables <code>var1</code> and <code>var2</code> . "observed" for the observed frequencies table, "expected" for the expected frequencies table (i.e., frequencies that would be expected if the null hypothesis holds), "chi_sq" for the contribution to the overall chi-squared statistic for each cell (i.e., $(o - e)^2 / e$), "dev_std" for the standardized differences between the observed and expected frequencies (i.e., $(o - e) / \sqrt{e}$), and "row_perc", "col_perc", and "perc" for row, column, and table percentages respectively
<code>shiny</code>	Did the function call originate inside a shiny app
<code>custom</code>	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org for options.
<code>...</code>	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/cross_tabs.html for an example in Radiant

See Also

`cross_tabs` to calculate results

`summary.cross_tabs` to summarize results

Examples

```
result <- cross_tabs(newspaper, "Income", "Newspaper")
plot(result, check = c("observed", "expected", "chi_sq"))
```

plot.goodness

Plot method for the goodness function

Description

Plot method for the goodness function

Usage

```
## S3 method for class 'goodness'
plot(x, check = "", fillcol = "blue",
     shiny = FALSE, custom = FALSE, ...)
```

Arguments

<code>x</code>	Return value from <code>goodness</code>
<code>check</code>	Show plots for variable <code>var</code> . "observed" for the observed frequencies table, "expected" for the expected frequencies table (i.e., frequencies that would be expected if the null hypothesis holds), "chi_sq" for the contribution to the overall chi-squared statistic for each cell (i.e., $(o - e)^2 / e$), and "dev_std" for the standardized differences between the observed and expected frequencies (i.e., $(o - e) / \sqrt{e}$)
<code>fillcol</code>	Color used for bar plots
<code>shiny</code>	Did the function call originate inside a shiny app
<code>custom</code>	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org for options.
<code>...</code>	further arguments passed to or from other methods

Details

See <https://radiant-rstats.github.io/docs/basics/goodness> for an example in Radiant

See Also

goodness to calculate results
summary.goodness to summarize results

Examples

```
result <- goodness(newspaper, "Income")  
plot(result, check = c("observed", "expected", "chi_sq"))  
goodness(newspaper, "Income") %>% plot(c("observed", "expected"))
```

plot.prob_binom *Plot method for the probability calculator (binomial)*

Description

Plot method for the probability calculator (binomial)

Usage

```
## S3 method for class 'prob_binom'  
plot(x, type = "values", ...)
```

Arguments

x	Return value from prob_binom
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_binom(n = 10, p = 0.3, ub = 3)  
plot(result, type = "values")
```

plot.prob_chisq *Plot method for the probability calculator (Chi-squared distribution)*

Description

Plot method for the probability calculator (Chi-squared distribution)

Usage

```
## S3 method for class 'prob_chisq'
plot(x, type = "values", ...)
```

Arguments

x	Return value from prob_chisq
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_chisq(df = 1, ub = 3.841)
plot(result, type = "values")
```

plot.prob_disc *Plot method for the probability calculator (discrete)*

Description

Plot method for the probability calculator (discrete)

Usage

```
## S3 method for class 'prob_disc'
plot(x, type = "values", ...)
```

Arguments

x	Return value from prob_disc
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_disc(v = "5 6 7 8 9 10 11", p = ".1 .2 .3 .15 .1 .1 .05", prob = 0.95)
plot(result, type = "probs")
```

plot.prob_expo	<i>Plot method for the probability calculator (Exponential distribution)</i>
----------------	--

Description

Plot method for the probability calculator (Exponential distribution)

Usage

```
## S3 method for class 'prob_expo'
plot(x, type = "values", ...)
```

Arguments

x	Return value from prob_expo
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_expo(rate = 1, ub = 2.996)
plot(result, type = "values")
```

plot.prob.fdist *Plot method for the probability calculator (F-distribution)*

Description

Plot method for the probability calculator (F-distribution)

Usage

```
## S3 method for class 'prob.fdist'  
plot(x, type = "values", ...)
```

Arguments

x	Return value from prob.fdist
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob.fdist(df1 = 10, df2 = 10, ub = 2.978)  
plot(result, type = "values")
```

plot.prob.lnorm *Plot method for the probability calculator (log normal)*

Description

Plot method for the probability calculator (log normal)

Usage

```
## S3 method for class 'prob.lnorm'  
plot(x, type = "values", ...)
```

Arguments

x	Return value from prob.norm
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_lnorm(meanlog = 0, sdlog = 1, lb = 0, ub = 1)
plot(result, type = "values")
```

plot.prob_norm	<i>Plot method for the probability calculator (normal)</i>
----------------	--

Description

Plot method for the probability calculator (normal)

Usage

```
## S3 method for class 'prob_norm'
plot(x, type = "values", ...)
```

Arguments

x	Return value from prob_norm
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

See Also

prob_norm to calculate results
summary.prob_norm to summarize results

Examples

```
result <- prob_norm(mean = 0, stdev = 1, ub = 0)
plot(result)
```

plot.prob_pois *Plot method for the probability calculator (poisson)*

Description

Plot method for the probability calculator (poisson)

Usage

```
## S3 method for class 'prob_pois'  
plot(x, type = "values", ...)
```

Arguments

x	Return value from prob_pois
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_pois(lambda = 1, ub = 3)  
plot(result, type = "values")
```

plot.prob_tdist *Plot method for the probability calculator (t-distribution)*

Description

Plot method for the probability calculator (t-distribution)

Usage

```
## S3 method for class 'prob_tdist'  
plot(x, type = "values", ...)
```

Arguments

x	Return value from prob_tdist
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_tdist(df = 10, ub = 2.228)
plot(result, type = "values")
```

plot.prob_unif	<i>Plot method for the probability calculator (uniform)</i>
----------------	---

Description

Plot method for the probability calculator (uniform)

Usage

```
## S3 method for class 'prob_unif'
plot(x, type = "values", ...)
```

Arguments

x	Return value from prob_unif
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_unif(min = 0, max = 1, ub = 0.3)
plot(result, type = "values")
```

plot.single_mean *Plot method for the single_mean function*

Description

Plot method for the single_mean function

Usage

```
## S3 method for class 'single_mean'  
plot(x, plots = "hist", shiny = FALSE,  
      custom = FALSE, ...)
```

Arguments

x	Return value from single_mean
plots	Plots to generate. "hist" shows a histogram of the data along with vertical lines that indicate the sample mean and the confidence interval. "simulate" shows the location of the sample mean and the comparison value (comp_value). Simulation is used to demonstrate the sampling variability in the data under the null-hypothesis
shiny	Did the function call originate inside a shiny app
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org for options.
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/single_mean.html for an example in Radiant

See Also

single_mean to generate the result
summary.single_mean to summarize results

Examples

```
result <- single_mean(diamonds, "price", comp_value = 3500)  
plot(result, plots = c("hist", "simulate"))
```

plot.single_prop *Plot method for the single_prop function*

Description

Plot method for the single_prop function

Usage

```
## S3 method for class 'single_prop'  
plot(x, plots = "bar", shiny = FALSE,  
      custom = FALSE, ...)
```

Arguments

x	Return value from single_prop
plots	Plots to generate. "bar" shows a bar chart of the data. The "simulate" chart shows the location of the sample proportion and the comparison value (comp_value). Simulation is used to demonstrate the sampling variability in the data under the null-hypothesis
shiny	Did the function call originate inside a shiny app
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org for options.
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/single_prop.html for an example in Radiant

See Also

single_prop to generate the result
summary.single_prop to summarize the results

Examples

```
result <- single_prop(titanic, "survived", lev = "Yes", comp_value = 0.5, alternative = "le  
plot(result, plots = c("bar", "simulate"))
```

```
print.rcorr
```

Print method for the correlation function

Description

Print method for the correlation function

Usage

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

Arguments

x	Return value from <code>correlation</code>
...	further arguments passed to or from other methods

```
prob_binom
```

Probability calculator for the binomial distribution

Description

Probability calculator for the binomial distribution

Usage

```
prob_binom(n, p, lb = NA, ub = NA, plb = NA, pub = NA, dec = 3)
```

Arguments

n	Number of trials
p	Probability
lb	Lower bound on the number of successes
ub	Upper bound on the number of successes
plb	Lower probability bound
pub	Upper probability bound
dec	Number of decimals to show

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
prob_binom(n = 10, p = 0.3, ub = 3)
```

`prob_chisq`*Probability calculator for the chi-squared distribution*

Description

Probability calculator for the chi-squared distribution

Usage

```
prob_chisq(df, lb = NA, ub = NA, plb = NA, pub = NA, dec = 3)
```

Arguments

<code>df</code>	Degrees of freedom
<code>lb</code>	Lower bound (default is 0)
<code>ub</code>	Upper bound (default is Inf)
<code>plb</code>	Lower probability bound
<code>pub</code>	Upper probability bound
<code>dec</code>	Number of decimals to show

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
prob_chisq(df = 1, ub = 3.841)
```

`prob_disc`*Probability calculator for a discrete distribution*

Description

Probability calculator for a discrete distribution

Usage

```
prob_disc(v, p, lb = NA, ub = NA, plb = NA, pub = NA, dec = 3)
```

Arguments

v	Values
p	Probabilities
lb	Lower bound on the number of successes
ub	Upper bound on the number of successes
plb	Lower probability bound
pub	Upper probability bound
dec	Number of decimals to show

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
prob_disc(v = "5 6 7 8 9 10 11", p = ".1 .2 .3 .15 .1 .1 .05", pub = 0.95)
```

prob_expo

Probability calculator for the exponential distribution

Description

Probability calculator for the exponential distribution

Usage

```
prob_expo(rate, lb = NA, ub = NA, plb = NA, pub = NA, dec = 3)
```

Arguments

rate	Rate
lb	Lower bound (default is 0)
ub	Upper bound (default is Inf)
plb	Lower probability bound
pub	Upper probability bound
dec	Number of decimals to show

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
prob_expo(rate = 1, ub = 2.996)
```

prob_fdist

Probability calculator for the F-distribution

Description

Probability calculator for the F-distribution

Usage

```
prob_fdist(df1, df2, lb = NA, ub = NA, plb = NA, pub = NA,  
           dec = 3)
```

Arguments

df1	Degrees of freedom
df2	Degrees of freedom
lb	Lower bound (default is 0)
ub	Upper bound (default is Inf)
plb	Lower probability bound
pub	Upper probability bound
dec	Number of decimals to show

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
prob_fdist(df1 = 10, df2 = 10, ub = 2.978)
```

prob_lnorm *Probability calculator for the log normal distribution*

Description

Probability calculator for the log normal distribution

Usage

```
prob_lnorm(meanlog, sdlog, lb = NA, ub = NA, plb = NA, pub = NA,  
           dec = 3)
```

Arguments

meanlog	Mean of the distribution on the log scale
sdlog	Standard deviation of the distribution on the log scale
lb	Lower bound (default is -Inf)
ub	Upper bound (default is Inf)
plb	Lower probability bound
pub	Upper probability bound
dec	Number of decimals to show

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
prob_lnorm(meanlog = 0, sdlog = 1, lb = 0, ub = 1)
```

prob_norm *Probability calculator for the normal distribution*

Description

Probability calculator for the normal distribution

Usage

```
prob_norm(mean, stdev, lb = NA, ub = NA, plb = NA, pub = NA,  
          dec = 3)
```

Arguments

mean	Mean
stdev	Standard deviation
lb	Lower bound (default is -Inf)
ub	Upper bound (default is Inf)
plb	Lower probability bound
pub	Upper probability bound
dec	Number of decimals to show

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

See Also

`summary.prob_norm` to summarize results
`plot.prob_norm` to plot results

Examples

```
prob_norm(mean = 0, stdev = 1, ub = 0)
```

`prob_pois`*Probability calculator for the poisson distribution*

Description

Probability calculator for the poisson distribution

Usage

```
prob_pois(lambda, lb = NA, ub = NA, plb = NA, pub = NA, dec = 3)
```

Arguments

lambda	Rate
lb	Lower bound (default is 0)
ub	Upper bound (default is Inf)
plb	Lower probability bound
pub	Upper probability bound
dec	Number of decimals to show

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
prob_pois(lambda = 1, ub = 3)
```

prob_tdist	<i>Probability calculator for the t-distribution</i>
------------	--

Description

Probability calculator for the t-distribution

Usage

```
prob_tdist(df, lb = NA, ub = NA, plb = NA, pub = NA, dec = 3)
```

Arguments

df	Degrees of freedom
lb	Lower bound (default is -Inf)
ub	Upper bound (default is Inf)
plb	Lower probability bound
pub	Upper probability bound
dec	Number of decimals to show

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
prob_tdist(df = 10, ub = 2.228)
```

`prob_unif`*Probability calculator for the uniform distribution*

Description

Probability calculator for the uniform distribution

Usage

```
prob_unif(min, max, lb = NA, ub = NA, plb = NA, pub = NA,  
          dec = 3)
```

Arguments

<code>min</code>	Minimum value
<code>max</code>	Maximum value
<code>lb</code>	Lower bound (default = 0)
<code>ub</code>	Upper bound (default = 1)
<code>plb</code>	Lower probability bound
<code>pub</code>	Upper probability bound
<code>dec</code>	Number of decimals to show

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
prob_unif(min = 0, max = 1, ub = 0.3)
```

`radiant.basics`*radiant.basics*

Description

`radiant.basics`

Launch `radiant.basics` in the default browser

Usage

```
radiant.basics(state, ...)
```

Arguments

state Path to state file to load
... additional arguments to pass to shiny::runApp (e.g, port = 8080)

Details

See <https://radiant-rstats.github.io/docs> for documentation and tutorials

Examples

```
## Not run:  
radiant.basics()  
  
## End(Not run)
```

```
radiant.basics_viewer
```

Launch radiant.basics in the Rstudio viewer

Description

Launch radiant.basics in the Rstudio viewer

Usage

```
radiant.basics_viewer(state, ...)
```

Arguments

state Path to state file to load
... additional arguments to pass to shiny::runApp (e.g, port = 8080)

Details

See <https://radiant-rstats.github.io/docs> for documentation and tutorials

Examples

```
## Not run:  
radiant.basics_viewer()  
  
## End(Not run)
```

```
radiant.basics_window
```

Launch radiant.basics in an Rstudio window

Description

Launch radiant.basics in an Rstudio window

Usage

```
radiant.basics_window(state, ...)
```

Arguments

state	Path to state file to load
...	additional arguments to pass to shiny::runApp (e.g, port = 8080)

Details

See <https://radiant-rstats.github.io/docs> for documentation and tutorials

Examples

```
## Not run:  
radiant.basics_window()  
  
## End(Not run)
```

```
salary
```

Salaries for Professors

Description

Salaries for Professors

Usage

```
data(salary)
```

Format

A data frame with 397 rows and 6 variables

Details

2008-2009 nine-month salary for professors in a college in the US. Description provided in attr(salary,description")

single_mean	<i>Compare a sample mean to a population mean</i>
-------------	---

Description

Compare a sample mean to a population mean

Usage

```
single_mean(dataset, var, comp_value = 0, alternative = "two.sided",  
            conf_lev = 0.95, data_filter = "")
```

Arguments

dataset	Dataset
var	The variable selected for the mean comparison
comp_value	Population value to compare to the sample mean
alternative	The alternative hypothesis ("two.sided", "greater", or "less")
conf_lev	Span for the confidence interval
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

See https://radiant-rstats.github.io/docs/basics/single_mean.html for an example in Radiant

Value

A list of variables defined in `single_mean` as an object of class `single_mean`

See Also

`summary.single_mean` to summarize results
`plot.single_mean` to plot results

Examples

```
single_mean(diamonds, "price") %>% str()
```

single_prop	<i>Compare a sample proportion to a population proportion</i>
-------------	---

Description

Compare a sample proportion to a population proportion

Usage

```
single_prop(dataset, var, lev = "", comp_value = 0.5,
  alternative = "two.sided", conf_lev = 0.95, test = "binom",
  data_filter = "")
```

Arguments

dataset	Dataset
var	The variable selected for the proportion comparison
lev	The factor level selected for the proportion comparison
comp_value	Population value to compare to the sample proportion
alternative	The alternative hypothesis ("two.sided", "greater", or "less")
conf_lev	Span of the confidence interval
test	bionomial exact test ("binom") or Z-test ("z")
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

See https://radiant-rstats.github.io/docs/basics/single_prop.html for an example in Radiant

Value

A list of variables used in single_prop as an object of class single_prop

See Also

`summary.single_prop` to summarize the results

`plot.single_prop` to plot the results

Examples

```
single_prop(titanic, "survived") %>% str()
single_prop(titanic, "survived", lev = "Yes", comp_value = 0.5, alternative = "less") %>% s
```

```
summary.compare_means
```

Summary method for the compare_means function

Description

Summary method for the compare_means function

Usage

```
## S3 method for class 'compare_means'  
summary(object, show = FALSE, dec = 3, ...)
```

Arguments

object	Return value from compare_means
show	Show additional output (i.e., t.value, df, and confidence interval)
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/compare_means.html for an example in Radiant

See Also

compare_means to calculate results
plot.compare_means to plot results

Examples

```
result <- compare_means(diamonds, "cut", "price")  
summary(result)
```

```
summary.compare_props
```

Summary method for the compare_props function

Description

Summary method for the compare_props function

Usage

```
## S3 method for class 'compare_props'  
summary(object, show = FALSE, dec = 3, ...)
```

Arguments

object	Return value from compare_props
show	Show additional output (i.e., chisq.value, df, and confidence interval)
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/compare_props.html for an example in Radiant

See Also

compare_props to calculate results
plot.compare_props to plot results

Examples

```
result <- compare_props(titanic, "pclass", "survived")  
summary(result)
```

```
summary.correlation
```

Summary method for the correlation function

Description

Summary method for the correlation function

Usage

```
## S3 method for class 'correlation'  
summary(object, cutoff = 0, covar = FALSE,  
        dec = 2, ...)
```

Arguments

object	Return value from correlation
cutoff	Show only correlations larger than the cutoff in absolute value. Default is a cutoff of 0
covar	Show the covariance matrix (default is FALSE)
dec	Number of decimals to show
...	further arguments passed to or from other methods.

Details

See <https://radiant-rstats.github.io/docs/basics/correlation.html> for an example in Radiant

See Also

correlation to calculate results
plot.correlation to plot results

Examples

```
result <- correlation(diamonds, c("price", "carat", "table"))  
summary(result, cutoff = .3)
```

summary.cross_tabs *Summary method for the cross_tabs function*

Description

Summary method for the cross_tabs function

Usage

```
## S3 method for class 'cross_tabs'  
summary(object, check = "", dec = 2, ...)
```

Arguments

object	Return value from cross_tabs
check	Show table(s) for variables var1 and var2. "observed" for the observed frequencies table, "expected" for the expected frequencies table (i.e., frequencies that would be expected if the null hypothesis holds), "chi_sq" for the contribution to the overall chi-squared statistic for each cell (i.e., $(o - e)^2 / e$), "dev_std" for the standardized differences between the observed and expected frequencies (i.e., $(o - e) / \sqrt{e}$), and "dev_perc" for the percentage difference between the observed and expected frequencies (i.e., $(o - e) / e$)
dec	Number of decimals to show
...	further arguments passed to or from other methods.

Details

See https://radiant-rstats.github.io/docs/basics/cross_tabs.html for an example in Radiant

See Also

cross_tabs to calculate results
plot.cross_tabs to plot results

Examples

```
result <- cross_tabs(newspaper, "Income", "Newspaper")  
summary(result, check = c("observed", "expected", "chi_sq"))
```

summary.goodness *Summary method for the goodness function*

Description

Summary method for the goodness function

Usage

```
## S3 method for class 'goodness'
summary(object, check = "", dec = 2, ...)
```

Arguments

object	Return value from goodness
check	Show table(s) for the selected variable (var). "observed" for the observed frequencies table, "expected" for the expected frequencies table (i.e., frequencies that would be expected if the null hypothesis holds), "chi_sq" for the contribution to the overall chi-squared statistic for each cell (i.e., $(o - e)^2 / e$), "dev_std" for the standardized differences between the observed and expected frequencies (i.e., $(o - e) / \sqrt{e}$), and "dev_perc" for the percentage difference between the observed and expected frequencies (i.e., $(o - e) / e$)
dec	Number of decimals to show
...	further arguments passed to or from other methods.

Details

See <https://radiant-rstats.github.io/docs/basics/goodness> for an example in Radiant

See Also

goodness to calculate results
 plot.goodness to plot results

Examples

```
result <- goodness(newspaper, "Income", c(.3, .7))
summary(result, check = c("observed", "expected", "chi_sq"))
goodness(newspaper, "Income", "1/3 2/3") %>% summary("observed")
```

summary.prob_binom *Summary method for the probability calculator (binomial)*

Description

Summary method for the probability calculator (binomial)

Usage

```
## S3 method for class 'prob_binom'  
summary(object, type = "values", ...)
```

Arguments

object	Return value from prob_binom
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_binom(n = 10, p = 0.3, ub = 3)  
summary(result, type = "values")
```

summary.prob_chisq *Summary method for the probability calculator (Chi-squared distribution)*

Description

Summary method for the probability calculator (Chi-squared distribution)

Usage

```
## S3 method for class 'prob_chisq'  
summary(object, type = "values", ...)
```

Arguments

object	Return value from prob_chisq
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_chisq(df = 1, ub = 3.841)
summary(result, type = "values")
```

summary.prob_disc *Summary method for the probability calculator (discrete)*

Description

Summary method for the probability calculator (discrete)

Usage

```
## S3 method for class 'prob_disc'
summary(object, type = "values", ...)
```

Arguments

object	Return value from prob_disc
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_disc(v = "5 6 7 8 9 10 11", p = ".1 .2 .3 .15 .1 .1 .05", pub = 0.95)
summary(result, type = "probs")
```

summary.prob_expo *Summary method for the probability calculator (exponential)*

Description

Summary method for the probability calculator (exponential)

Usage

```
## S3 method for class 'prob_expo'  
summary(object, type = "values", ...)
```

Arguments

object	Return value from prob_expo
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_expo(rate = 1, ub = 2.996)  
summary(result, type = "values")
```

summary.prob_fdist *Summary method for the probability calculator (F-distribution)*

Description

Summary method for the probability calculator (F-distribution)

Usage

```
## S3 method for class 'prob_fdist'  
summary(object, type = "values", ...)
```

Arguments

object	Return value from prob_fdist
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_fdist(df1 = 10, df2 = 10, ub = 2.978)
summary(result, type = "values")
```

summary.prob_lnorm *Summary method for the probability calculator (log normal)*

Description

Summary method for the probability calculator (log normal)

Usage

```
## S3 method for class 'prob_lnorm'
summary(object, type = "values", ...)
```

Arguments

object	Return value from prob_norm
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_lnorm(meanlog = 0, sdlog = 1, lb = 0, ub = 1)
summary(result, type = "values")
```

summary.prob_norm *Summary method for the probability calculator (normal)*

Description

Summary method for the probability calculator (normal)

Usage

```
## S3 method for class 'prob_norm'  
summary(object, type = "values", ...)
```

Arguments

object	Return value from prob_norm
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

See Also

prob_norm to calculate results
plot.prob_norm to plot results

Examples

```
result <- prob_norm(mean = 0, stdev = 1, ub = 0)  
summary(result)
```

summary.prob_pois *Summary method for the probability calculator (poisson)*

Description

Summary method for the probability calculator (poisson)

Usage

```
## S3 method for class 'prob_pois'  
summary(object, type = "values", ...)
```

Arguments

object	Return value from prob_pois
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_pois(lambda = 1, ub = 3)
summary(result, type = "values")
```

summary.prob_tdist *Summary method for the probability calculator (t-distribution)*

Description

Summary method for the probability calculator (t-distribution)

Usage

```
## S3 method for class 'prob_tdist'
summary(object, type = "values", ...)
```

Arguments

object	Return value from prob_tdist
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_tdist(df = 10, ub = 2.228)
summary(result, type = "values")
```

summary.prob_unif *Summary method for the probability calculator (uniform)*

Description

Summary method for the probability calculator (uniform)

Usage

```
## S3 method for class 'prob_unif'  
summary(object, type = "values", ...)
```

Arguments

object	Return value from prob_unif
type	Probabilities ("probs") or values ("values")
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/prob_calc.html for an example in Radiant

Examples

```
result <- prob_unif(min = 0, max = 1, ub = 0.3)  
summary(result, type = "values")
```

summary.single_mean

Summary method for the single_mean function

Description

Summary method for the single_mean function

Usage

```
## S3 method for class 'single_mean'  
summary(object, dec = 3, ...)
```

Arguments

object	Return value from single_mean
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/single_mean.html for an example in Radiant

See Also

`single_mean` to generate the results

`plot.single_mean` to plot results

Examples

```
result <- single_mean(diamonds, "price")
summary(result)
diamonds %>% single_mean("price") %>% summary()
```

```
summary.single_prop
```

Summary method for the single_prop function

Description

Summary method for the `single_prop` function

Usage

```
## S3 method for class 'single_prop'
summary(object, dec = 3, ...)
```

Arguments

<code>object</code>	Return value from <code>single_prop</code>
<code>dec</code>	Number of decimals to show
<code>...</code>	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/basics/single_prop.html for an example in Radiant

See Also

`single_prop` to generate the results

`plot.single_prop` to plot the results

Examples

```
result <- single_prop(titanic, "survived", lev = "Yes", comp_value = 0.5, alternative = "le")
summary(result)
```