Package ‘radiant.basics’

January 10, 2023

Type Package

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

Version 1.5.0

Date 2023-1-9

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 (>= 4.0.0), radiant.data (>= 1.5.0)

Imports ggplot2 (>= 2.2.1), scales (>= 0.4.0), dplyr (>= 1.0.7), tidyr (>= 0.8.2), magrittr (>= 1.5), shiny (>= 1.7.1), psych (>= 1.8.3.3), import (>= 1.1.0), lubridate (>= 1.7.4), polycor (>= 0.7.10), patchwork (>= 1.0.0), rlang (>= 1.0.6)

Suggests testthat (>= 2.0.0), pkgdown (>= 1.1.0), markdown (>= 1.3)


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

License AGPL-3 | file LICENSE

LazyData true

Encoding UTF-8

Language en-US

RoxygenNote 7.2.3

NeedsCompilation no

Author Vincent Nijs [aut, cre]

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

Repository CRAN

Date/Publication 2023-01-10 08:50:05 UTC
R topics documented:

clt ................................................................. 3
compare_means ............................................... 4
compare_props ............................................. 6
consider ...................................................... 7
cor2df .......................................................... 7
correlation ................................................... 8
cross_tabs .................................................... 9
demand_uk .................................................... 10
goodness ..................................................... 10
newspaper ..................................................... 11
plot.clt ........................................................ 12
plot.compare_means ....................................... 12
plot.compare_props ........................................ 13
plot.correlation ........................................... 14
plot.cross_tabs ............................................. 15
plot.goodness .............................................. 16
plot.prob_binom ........................................... 17
plot.prob_chisq ........................................... 18
plot.prob_disc ............................................. 18
plot.prob_expo ............................................ 19
plot.prob_fdist .......................................... 20
plot.prob_lnorm .......................................... 21
plot.prob_norm ............................................ 21
plot.prob_pois ............................................ 22
plot.prob_tdist .......................................... 23
plot.prob_unif ............................................ 24
plot.single_mean ......................................... 24
plot.single_prop ......................................... 25
print.rcorr ................................................ 26
prob_binom ................................................ 27
prob_chisq ................................................ 28
prob_disc .................................................. 29
prob_expo .................................................. 30
prob_fdist .................................................. 31
prob_lnorm ................................................ 32
prob_norm ................................................... 33
prob_pois .................................................... 34
prob_tdist .................................................. 35
prob_unif ................................................... 36
radiant.basics ............................................ 37
radiant.basics_viewer .................................... 37
radiant.basics_window ................................... 38
salary ......................................................... 38
single_mean ............................................... 39
single_prop ................................................ 40
summary.compare_means ................................ 41
Central Limit Theorem simulation

Description
Central Limit Theorem simulation

Usage
```r
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
compare_means

- **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](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**

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

---

**Description**

Compare sample means

**Usage**

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

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

**Details**

See [https://radiant-rstats.github.io/docs/basics/compare_means.html](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**

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

Compare sample proportions across groups

Description

Compare sample proportions across groups

Usage

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

Arguments

- `dataset` Dataset
- `var1` A grouping variable to split the data for comparisons
- `var2` The variable to calculate proportions for
- `levs` The factor level selected for the proportion comparison
- `alternative` The alternative hypothesis ("two.sided", "greater" or "less")
- `conf_lev` Span of the confidence interval
- `comb` Combinations to evaluate
- `adjust` Adjustment for multiple comparisons ("none" or "bonf" for Bonferroni)
- `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")
- `envir` Environment to extract data from

Details

See [https://radiant-rstats.github.io/docs/basics/compare_props.html](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

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

---

### consider

**Car brand consideration**

**Description**

Car brand consideration

**Usage**

```r
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")

---

### cor2df

**Store a correlation matrix as a (long) data.frame**

**Description**

Store a correlation matrix as a (long) data.frame

**Usage**

```r
cor2df(object, labels = c("label1", "label2"), ...)
```

**Arguments**

- `object` : Return value from `correlation`
- `labels` : Column names for the correlation pairs
- `...` : further arguments passed to or from other methods
correlation

Details

Return the correlation matrix as a (long) data.frame. See https://radiant-rstats.github.io/docs/basics/correlation.html for an example in Radiant

---

**correlation**

*Calculate correlations for two or more variables*

Description

Calculate correlations for two or more variables

Usage

```r
correlation(
    dataset,
    vars = "",
    method = "pearson",
    hcor = FALSE,
    hcor_se = FALSE,
    data_filter = "",
    envir = parent.frame()
)
```

Arguments

- `dataset`: Dataset
- `vars`: Variables to include in the analysis. Default is all but character and factor variables with more than two unique values are removed
- `method`: Type of correlations to calculate. Options are "pearson", "spearman", and "kendall". "pearson" is the default
- `hcor`: Use polycor::hetcor to calculate the correlation matrix
- `hcor_se`: Calculate standard errors when using polycor::hetcor
- `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")
- `envir`: Environment to extract data from

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
cross_tabs

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 = "",
  envir = parent.frame()
)

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., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")
envir Environment to extract data from

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
goodness

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

Demand in the UK

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

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

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 = "",
  envir = parent.frame()
)
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")
- **envir**: Environment to extract data from

Details

See [https://radiant-rstats.github.io/docs/basics/goodness.html](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

```r
goodness(newspaper, "Income") %>% str()
goodness(newspaper, "Income", p = c(3 / 4, 1 / 4)) %>% str()
table(select(newspaper, Income)) %>% goodness(tab = .)
```

newspaper

<table>
<thead>
<tr>
<th>newspaper</th>
<th>Newspaper readership</th>
</tr>
</thead>
</table>

Description

Newspaper readership

Usage

data(newspaper)

Format

A data frame with 580 rows and 2 variables
plot.compare_means

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

Arguments
- `x`: Return value from `clt`
- `stat`: Statistic to use (sum or mean)
- `bins`: Number of bins to use
- `...`: Further arguments passed to or from other methods

Details
See [https://radiant-rstats.github.io/docs/basics/clt.html](https://radiant-rstats.github.io/docs/basics/clt.html) for an example in Radiant

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

Arguments

x
  Return value from compare_means
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 https://ggplot2.tidyverse.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 compare_props
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 [https://ggplot2.tidyverse.org/](https://ggplot2.tidyverse.org/) for options.

... further arguments passed to or from other methods

Details

See [https://radiant-rstats.github.io/docs/basics/compare_props.html](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

```r
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

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

Arguments

- `x` Return value from `correlation`
- `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 the x and y variables in a 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](https://radiant-rstats.github.io/docs/basics/correlation.html) for an example in Radiant
plot.cross_tabs

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
x  Return value from cross_tabs
check  Show plots 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 "row_perc", "col_perc", and "perc" for row, column, and table percentages respectively
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 https://ggplot2.tidyverse.org/ for options.
...
  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

x Return value from goodness
check Show plots for 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., \((o - e)^2 / e\)), and "dev_std" for the standardized differences between the observed and expected frequencies (i.e., \((o - e) / \sqrt{e}\))
fillcol Color used for bar plots
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 https://ggplot2.tidyverse.org/ for options.
...

Details

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

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

See Also

prob_binom to calculate results
summary.prob_binom to summarize results

Examples

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

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

See Also
prob_chisq to calculate results
summary.prob_chisq to summarize results

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

See Also

- `prob_disc` to calculate results
- `summary.prob_disc` to summarize results

Examples

```r
result <- prob_disc(v = 1:6, p = c(2 / 6, 2 / 6, 1 / 12, 1 / 12, 1 / 12, 1 / 12), pub = 0.95)
plot(result, type = "probs")
```

plot.prob_expo  
*Plot method for the probability calculator (Exponential distribution)*

Description

Plot method for the probability calculator (Exponential distribution)

Usage

```r
## 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](https://radiant-rstats.github.io/docs/basics/prob_calc.html) for an example in Radiant
plot.prob_fdist

See Also

prob_expo to calculate results
summary.prob_expo to summarize results

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

See Also

prob_fdist to calculate results
summary.prob_fdist to summarize results

Examples

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

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_lnorm to calculate results

plot.prob_lnorm to plot results

Examples

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

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](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

```r
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

```r
## 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](https://radiant-rstats.github.io/docs/basics/prob_calc.html) for an example in Radiant
plot.prob_tdist

See Also

prob_pois to calculate results
summary.prob_pois to summarize results

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

See Also

prob_tdist to calculate results
summary.prob_tdist to summarize results

Examples

result <- prob_tdist(df = 10, ub = 2.228)
plot(result, type = "values")
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](https://radiant-rstats.github.io/docs/basics/prob_calc.html) for an example in Radiant

See Also

`prob_unif` to calculate results

`summary.prob_unif` to summarize results

Examples

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

Description

Plot method for the single_mean function

Usage

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

```
plot.single_prop

Arguments

x  Return value from \texttt{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 (\texttt{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 \url{https://ggplot2.tidyverse.org/} for options.

...  further arguments passed to or from other methods

Details

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

See Also

\texttt{single_mean} to generate the result
\texttt{summary.single_mean} to summarize results

Examples

result <- single_mean(diamonds, "price", comp_value = 3500)
plot(result, plots = c("hist", "simulate"))
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 [https://ggplot2.tidyverse.org/](https://ggplot2.tidyverse.org/) for options.
- **...**: further arguments passed to or from other methods

Details

See [https://radiant-rstats.github.io/docs/basics/single_prop.html](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

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

---

**Description**

Print method for the correlation function

**Usage**

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

**Arguments**

- **x**: Return value from `correlation`
- **...**: further arguments passed to or from other methods
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

See Also

summary.prob_binom to summarize results
plot.prob_binom to plot results

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**

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

**Arguments**

- `df`  
  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](https://radiant-rstats.github.io/docs/basics/prob_calc.html) for an example in Radiant

**See Also**

- `summary.prob_chisq` to summarize results
- `plot.prob_chisq` to plot results

**Examples**

```r
prob_chisq(df = 1, ub = 3.841)
```
**prob_disc**

*Probability calculator for a discrete distribution*

### Description

Probability calculator for a discrete distribution

### Usage

```r
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](https://radiant-rstats.github.io/docs/basics/prob_calc.html) for an example in Radiant

### See Also

- `summary.prob_disc` to summarize results
- `plot.prob_disc` to plot results

### Examples

```r
prob_disc(v = 1:6, p = 1 / 6, pub = 0.95)
prob_disc(v = 1:6, p = c(2 / 6, 2 / 6, 1 / 12, 1 / 12, 1 / 12, 1 / 12), pub = 0.95)
```
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

See Also

summary.prob_expo to summarize results
plot.prob_expo to plot results

Examples

prob_expo(rate = 1, ub = 2.996)
**Description**

Probability calculator for the F-distribution

**Usage**

```r
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](https://radiant-rstats.github.io/docs/basics/prob_calc.html) for an example in Radiant

**See Also**

- `summary.prob_fdist` to summarize results
- `plot.prob_fdist` to plot results

**Examples**

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

See Also

summary.prob_lnorm to summarize results
plot.prob_lnorm to plot results

Examples

prob_lnorm(meanlog = 0, sdlog = 1, lb = 0, ub = 1)
**Description**

Probability calculator for the normal distribution

**Usage**

```r
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](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**

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

See Also

summary.prob_pois to summarize results
plot.prob_pois to plot results

Examples

prob_pois(lambda = 1, ub = 3)
Probability calculator for the t-distribution

Usage

prob_tdist(df, lb = NA, ub = NA, p1lb = NA, pub = NA, dec = 3)

Arguments

df Degrees of freedom
lb Lower bound (default is -Inf)
ub Upper bound (default is Inf)
p1lb 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_tdist to summarize results
plot.prob_tdist to plot results

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

min  Minimum value
max  Maximum value
lb   Lower bound (default = 0)
ub   Upper bound (default = 1)
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_unif to summarize results
plot.prob_unif to plot results

Examples

prob_unif(min = 0, max = 1, ub = 0.3)
radiant.basics

Description
Launch radiant.basics in the default web 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

```r
## 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

```r
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/](https://radiant-rstats.github.io/docs/) for documentation and tutorials

## Examples

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

## End(Not run)
```

---

**salary**  *Salaries for Professors*

## Description

Salaries for Professors

## Usage

```r
data(salary)
```

## Format

A data frame with 397 rows and 6 variables
single_mean

Details

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

---

single_mean  Compare a sample mean to a population mean

Description

Compare a sample mean to a population mean

Usage

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

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")

envir  Environment to extract data from

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**

**Compare a sample proportion to a population proportion**

**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 = "",
    envir = parent.frame()
)
```

**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` binomial 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")
- `envir` Environment to extract data from

**Details**

See [https://radiant-rstats.github.io/docs/basics/single_prop.html](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

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

---

**summary.compare_means**  
*Summary method for the compare_means function*

**Description**

Summary method for the compare_means function

**Usage**

```r
## 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](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**

```r
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

<table>
<thead>
<tr>
<th>argument</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>Return value from compare_props</td>
</tr>
<tr>
<td>show</td>
<td>Show additional output (i.e., chisq.value, df, and confidence interval)</td>
</tr>
<tr>
<td>dec</td>
<td>Number of decimals to show</td>
</tr>
<tr>
<td>...</td>
<td>further arguments passed to or from other methods</td>
</tr>
</tbody>
</table>

Details

See [https://radiant-rstats.github.io/docs/basics/compare_props.html](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

<table>
<thead>
<tr>
<th>argument</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>Return value from correlation</td>
</tr>
<tr>
<td>cutoff</td>
<td>Correlation cutoff to consider as statistically significant</td>
</tr>
<tr>
<td>covar</td>
<td>Include covariates in the correlation matrix</td>
</tr>
<tr>
<td>dec</td>
<td>Number of decimals to show</td>
</tr>
<tr>
<td>...</td>
<td>further arguments passed to or from other methods</td>
</tr>
</tbody>
</table>
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](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

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

Description

Summary method for the cross_tabs function

Usage

```r
## 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.
summary.goodness

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
Summary method for the probability calculator (binomial)

Description

Summary method for the probability calculator (binomial)

Usage

```r
## 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](https://radiant-rstats.github.io/docs/basics/prob_calc.html) for an example in Radiant

See Also

- `prob_binom` to calculate results
- `plot.prob_binom` to plot results

Examples

```r
result <- prob_binom(n = 10, p = 0.3, ub = 3)
summary(result, type = "values")
```
Summary method for the probability calculator (Chi-squared distribution)

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

### Usage
```r
## 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](https://radiant-rstats.github.io/docs/basics/prob_calc.html) for an example in Radiant

### See Also
- `prob_chisq` to calculate results
- `plot.prob_chisq` to plot results

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

Summary method for the probability calculator (discrete)

### Description
Summary method for the probability calculator (discrete)

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

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

See Also

`prob_disc` to calculate results

`plot.prob_disc` to plot results

Examples

```r
result <- prob_disc(v = 1:6, p = c(2 / 6, 2 / 6, 1 / 12, 1 / 12, 1 / 12, 1 / 12), 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

```r
## 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](https://radiant-rstats.github.io/docs/basics/prob_calc.html) for an example in Radiant
See Also

`prob_expo` to calculate results
`plot.prob_expo` to plot results

Examples

```r
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**

```r
## 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](https://radiant-rstats.github.io/docs/basics/prob_calc.html) for an example in Radiant

See Also

`prob_fdist` to calculate results
`plot.prob_fdist` to plot results

Examples

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

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_lnorm to calculate results
plot.prob_lnorm to summarize results

Examples

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

Summary method for the probability calculator (normal)

Usage

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

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](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

```r
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

```r
## 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](https://radiant-rstats.github.io/docs/basics/prob_calc.html) for an example in Radiant
See Also

prob_pois to calculate results
plot.prob_pois to plot results

Examples

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

result <- prob_tdist(df = 10, ub = 2.228)
summary(result, type = "values")
### Summary method for the probability calculator (uniform)

#### Description
Summary method for the probability calculator (uniform)

#### Usage
```r
## 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](https://radiant-rstats.github.io/docs/basics/prob_calc.html) for an example in Radiant

#### See Also
- `prob_unif` to calculate results
- `plot.prob_unif` to plot results

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

### Summary method for the single_mean function

#### Description
Summary method for the single_mean function

#### Usage
```r
## 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](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

```r
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

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

Arguments

- `object`  
  Return value from `single_prop`
- `dec`  
  Number of decimals to show
- `...`  
  Further arguments passed to or from other methods

Details

See [https://radiant-rstats.github.io/docs/basics/single_prop.html](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 = "less")
summary(result)
Index

* datasets
  consider, 7
  demand_uk, 10
  newspaper, 11
  salary, 38

clt, 3, 12
compare_means, 4, 13, 41
compare_props, 6, 13, 14, 42
cor2df, 7
correlation, 7, 8, 14, 15, 26, 43
cross_tabs, 9, 15, 16, 43, 44
demand_uk, 10
goodness, 10, 16, 17, 44, 45
newspaper, 11

plot.clt, 12
plot.compare_means, 5, 12, 41
plot.compare_props, 7, 13, 42
plot.correlation, 9, 14, 43
plot.cross_tabs, 10, 15, 44
plot.goodness, 11, 16, 45
plot.prob_binom, 18, 27, 45
plot.prob_disc, 18, 29, 47
plot.prob_expo, 19, 30, 48
plot.prob_fdist, 20, 31, 48
plot.prob_lnorm, 21, 22, 32, 49
plot.prob_norm, 21, 33, 50
plot.prob_pois, 22, 34, 51
plot.prob_tdist, 23, 35, 51
plot.prob_unif, 24, 36, 52
plot.single_mean, 24, 39, 53
plot.single_prop, 25, 41, 54
print.rcorr, 26

prob_binom, 17, 27, 45
prob_chisq, 18, 28, 46
prob_disc, 19, 29, 47
prob_expo, 19, 30, 47, 48
prob_fdist, 20, 31, 48
prob_lnorm, 21, 32, 49
prob_norm, 21, 23, 33, 49, 50
prob_pois, 22, 23, 34, 50, 51
prob_tdist, 23, 35, 51
prob_unif, 24, 36, 52

radiant.basics, 37
radiant.basics_viewer, 37
radiant.basics_window, 38

salary, 38
single_mean, 25, 39, 53
single_prop, 26, 40, 53, 54
summary.compare_means, 5, 13, 41
summary.compare_props, 7, 14, 42
summary.correlation, 9, 15, 42
summary.cross_tabs, 10, 16, 43
summary.goodness, 11, 17, 44
summary.prob_binom, 17, 27, 45
summary.prob_chisq, 18, 28, 46
summary.prob_disc, 19, 29, 46
summary.prob_expo, 20, 30, 47
summary.prob_fdist, 20, 31, 48
summary.prob_lnorm, 32, 49
summary.prob_norm, 22, 33, 49
summary.prob_pois, 23, 34, 50
summary.prob_tdist, 23, 35, 51
summary.prob_unif, 24, 36, 52
summary.single_mean, 25, 39, 52
summary.single_prop, 26, 41, 53