Package 'radiant.basics'

June 30, 2016

Type Package

Title Basic analysis menu for Radiant. Builds on the radiant.data package

Version 0.5.2

Date 2016-6-29

Description Basic analysis menu for Radiant.

Depends R (>= 3.2.0),
radiant.data (>= 0.5.2)

Imports ggplot2 (>= 2.0.0),
gridExtra (>= 2.0.0),
scales (>= 0.3.0),
dplyr (>= 0.5),
tibble (>= 1.0),
tidyr (>= 0.4.1),
magrittr (>= 1.5),
broom (>= 0.4.0),
shiny (>= 0.13.2.9003),
psych (>= 1.5.8),
import (>= 1.1.0),
methods

Suggests devtools (>= 1.8.0),
testthat (>= 0.10.0),
covr (>= 1.2.0)

URL https://github.com/radiant-rstats/radiant.basics, http://vnijs.github.io/radiant/

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

License AGPL-3 | file LICENSE

LazyData true

RoxygenNote 5.0.1

R topics documented:

compare_means *Compare means for two or more variables*

Description

Compare means for two or more variables

Usage

```
compare_means(dataset, var1, var2, samples = "independent",
  alternative = "two.sided", conf_lev = 0.95, comb = "",
  adjust = "none", test = "t", dec = 3, data_filter = "")
```

Arguments

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an r_data list from Radiant
var1	A numeric variable or factor selected for comparison
var2	One or more numeric variables for comparison. If var1 is a factor only one variable can be selected and the mean of this variable is compared across (factor) levels of va1r
samples	Are samples independent ("independent") or not ("paired")
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)
test	t-test ("t") or Wilcox ("wilcox")
dec	Number of decimals to show
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 http://vnijs.github.io/radiant/quant/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

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

compare_props *Compare proportions across groups*

Description

Compare proportions across groups

Usage

```
compare_props(dataset, var1, var2, levs = "", alternative = "two.sided",
  conf_lev = 0.95, comb = "", adjust = "none", dec = 3,
  data_filter = "")
```

Arguments

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an r_data list from Radiant
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)
dec	Number of decimals to show
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 http://vnijs.github.io/radiant/quant/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

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

consider *Car brand consideration*

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")

correlation *Calculate correlations for two or more variables*

Description

Calculate correlations for two or more variables

Usage

```
correlation(dataset, vars, method = "pearson", dec = 2, data_filter = "")
```

Arguments

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an r_data list from Radiant
vars	Variables to include in the analysis
method	Type of correlations to calculate. Options are "pearson", "spearman", and "kendall". "pearson" is the default
dec	Number of decimals to show
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 http://vnijs.github.io/radiant/quant/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

```
result <- correlation("diamonds", c("price","carat"))
result <- correlation("diamonds", c("price","carat","clarity"))
result <- correlation("diamonds", "price:table")
result <- diamonds %>% correlation("price:table")
```

---

cross_tabs *Evaluate associations between categorical variables*

Description

Evaluate associations between categorical variables

Usage

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

Arguments

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an r_data list from Radiant
var1	A categorical variable
var2	Another categorical variable
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 http://vnijs.github.io/radiant/quant/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

```
result <- cross_tabs("newspaper", "Income", "Newspaper")
result <- newspaper %>% cross_tabs("Income", "Newspaper")
```

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 wtih a hypothesized distribution*

Description

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

Usage

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

Arguments

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an r_data list from Radiant
var	A categorical variable
p	Hypothesized distribution as a number, fraction, or numeric vector. If unspecified, defaults to an even distribution
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 http://vnijs.github.io/radiant/quant/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

```
result <- goodness("newspaper", "Income")
```

---

newspaper *Newspaper readership*

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.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 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 opion 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 http://vnijs.github.io/radiant/quant/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, ...)
```

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
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/quant/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, ...)
```

Arguments

x	Return value from correlation
...	further arguments passed to or from other methods.

Details

See http://vnijs.github.io/radiant/quant/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","clarity"))
plot(result)
diamonds %>% correlation("price:clarity") %>% plot
```

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, ...)
```

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
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/quant/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"))
newspaper %>% cross_tabs("Income", "Newspaper") %>% plot(c("observed","expected"))
```

---

plot.goodness *Plot method for the goodness function*

Description

Plot method for the goodness function

Usage

```
## S3 method for class 'goodness'
plot(x, check = "", shiny = 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))
shiny	Did the function call originate inside a shiny app
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/quant/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"))
newspaper %>% goodness("Income") %>% plot(c("observed","expected"))
```

---

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

Description

Plot method for the probability calculator function (binomial)

Usage

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

Arguments

x	Return value from prob_binom
type	Probabilities or values
shiny	Did the function call originate inside a shiny app
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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", shiny = FALSE, ...)
```

Arguments

x	Return value from prob_chisq
type	Probabilities or values
shiny	Did the function call originate inside a shiny app
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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

Description

Plot method for the probability calculator function (discrete)

Usage

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

Arguments

x	Return value from prob_disc
type	Probabilities or values
shiny	Did the function call originate inside a shiny app
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/quant/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)
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

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

Arguments

x	Return value from prob_expo
type	Probabilities or values
shiny	Did the function call originate inside a shiny app
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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", shiny = FALSE, ...)
```

Arguments

x	Return value from prob_fdist
type	Probabilities or values
shiny	Did the function call originate inside a shiny app
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

`plot.prob_norm` *Plot method for the probability calculator (normal)*

Description

Plot method for the probability calculator (normal)

Usage

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

Arguments

`x`	Return value from `prob_norm`
`type`	Probabilities or values
`shiny`	Did the function call originate inside a shiny app
`...`	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

`plot.prob_pois` *Plot method for the probability calculator function (Poisson distribution)*

Description

Plot method for the probability calculator function (Poisson distribution)

Usage

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

Arguments

`x`	Return value from `prob_pois`
`type`	Probabilities or values
`shiny`	Did the function call originate inside a shiny app
`...`	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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", shiny = FALSE, ...)
```

Arguments

x	Return value from prob_tdist
type	Probabilities or values
shiny	Did the function call originate inside a shiny app
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

plot.prob_unif *Plot method for the probability calculator (uniform)*

Description

Plot method for the probability calculator (uniform)

Usage

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

Arguments

x	Return value from prob_unif
type	Probabilities or values
shiny	Did the function call originate inside a shiny app
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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 opion 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 http://vnijs.github.io/radiant/quant/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 opion 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 http://vnijs.github.io/radiant/quant/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("diamonds","clarity", lev = "IF", comp_value = 0.05)
plot(result, plots = c("hist", "simulate"))
result <- single_prop("titanic","pclass", lev = "1st")
plot(result, plots = c("hist","simulate"))
```

prob_binom *Probability calculator for the binomial distribution (binomial)*

Description

Probability calculator for the binomial distribution (binomial)

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 http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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

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 http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

prob_disc — *Probability calculator for the discrete distribution (discrete)*

Description

Probability calculator for the discrete distribution (discrete)

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 http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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 http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

prob_tdist *Probability calculator for the t distribution*

Description

Probability calculator for the t distribution

Usage

```
prob_tdist(df, mean = 0, stdev = 1, lb = NA, ub = NA, plb = NA,
  pub = NA, dec = 3)
```

Arguments

df	Degrees of freedom
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 http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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	Minmum value
max	Maximum value
lb	Lower bound
ub	Upper bound
plb	Lower probability bound
pub	Upper probability bound
dec	Number of decimals to show

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

radiant.basics *radiant.basics*

Description

radiant.basics

Launch Radiant in the default browser

Usage

```
radiant.basics()
```

Details

See http://vnijs.github.io/radiant for documentation and tutorials

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 *Compare a sample mean to a population mean*

Description

Compare a sample mean to a population mean

Usage

```
single_mean(dataset, var, comp_value = 0, alternative = "two.sided",
  conf_lev = 0.95, dec = 3, data_filter = "")
```

Arguments

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an r_data list from Radiant
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
dec	Number of decimals to show
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 http://vnijs.github.io/radiant/quant/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")
```

---

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, dec = 3, data_filter = "")
```

Arguments

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an r_data list from Radiant
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
dec	Number of decimals to show
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 http://vnijs.github.io/radiant/quant/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

```
result <- single_prop("diamonds","cut")
result <- single_prop("diamonds","clarity", lev = "IF", comp_value = 0.05)
```

`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, ...)
```

Arguments

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

Details

See http://vnijs.github.io/radiant/quant/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)
result <- diamonds %>% tbl_df %>% compare_means("x","y")
summary(result)
result <- diamonds %>% tbl_df %>% group_by(cut) %>% compare_means("x",c("x","y"))
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, ...)
```

Arguments

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

Details

See http://vnijs.github.io/radiant/quant/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)
titanic %>% compare_props("pclass", "survived") %>% summary
```

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, ...)
```

Arguments

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

Details

See http://vnijs.github.io/radiant/quant/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","clarity"))
summary(result, cutoff = .3)
diamonds %>% correlation("price:clarity") %>% summary
```

---

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 = "", ...)
```

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)
...	further arguments passed to or from other methods.

Details

See http://vnijs.github.io/radiant/quant/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"))
newspaper %>% cross_tabs("Income", "Newspaper") %>% summary("observed")
```

---

summary.goodness *Summary method for the goodness function*

Description

Summary method for the goodness function

Usage

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

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)
...	further arguments passed to or from other methods.

Details

See http://vnijs.github.io/radiant/quant/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"))
newspaper %>% goodness("Income", "1/3 2/3") %>% summary("observed")
```

summary.prob_binom *Summary method for the probability calculator function*

Description

Summary method for the probability calculator function

Usage

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

Arguments

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

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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

Description

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

Usage

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

Arguments

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

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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

Description

Summary method for the probability calculator function (discrete)

Usage

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

Arguments

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

Details

See http://vnijs.github.io/radiant/quant/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 function (Exponential distribution)*

Description

Summary method for the probability calculator function (Exponential distribution)

Usage

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

Arguments

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

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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

Description

Summary method for the probability calculator function (F-distribution)

Usage

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

Arguments

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

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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

Description

Summary method for the probability calculator function (normal)

Usage

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

Arguments

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

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

summary.prob_pois *Summary method for the probability calculator function (Poisson distribution)*

Description

Summary method for the probability calculator function (Poisson distribution)

Usage

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

Arguments

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

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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

Description

Summary method for the probability calculator function (t-distribution)

Usage

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

Arguments

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

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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

Description

Summary method for the probability calculator function (uniform)

Usage

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

Arguments

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

Details

See http://vnijs.github.io/radiant/quant/prob_calc.html for an example in Radiant

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, ...)
```

Arguments

object	Return value from single_mean
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/quant/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, ...)
```

Arguments

`object`	Return value from `single_prop`
`...`	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/quant/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("diamonds","clarity", lev = "IF", comp_value = 0.05)
summary(result)
diamonds %>% single_prop("clarity", lev = "IF", comp_value = 0.05) %>% summary
```

Index