

Package ‘radiant.model’

June 15, 2018

Type Package

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

Version 0.9.5

Date 2018-6-11

Description The Radiant Model menu includes interfaces for linear and logistic regression, naive Bayes, neural networks, classification and regression trees, model evaluation, collaborative filtering, decision analysis, and simulation. The application extends the functionality in radiant.data.

Depends R (>= 3.4.0), radiant.data (>= 0.9.5)

Imports radiant.basics (>= 0.9.5), shiny (>= 1.1.0), nnet (>= 7.3.12), NeuralNetTools (>= 1.5.1), sandwich (>= 2.3.4), car (>= 2.1.3), ggplot2 (>= 2.2.1), gridExtra (>= 2.0.0), data.tree (>= 0.7.4), stringr (>= 1.1.0), lubridate (>= 1.7.2), tidyr (>= 0.8.0), dplyr (>= 0.7.5), rlang (>= 0.2.0), magrittr (>= 1.5), DiagrammeR (>= 1.0.0), import (>= 1.1.0), psych (>= 1.8.4), e1071 (>= 1.6.8), rpart (>= 4.1.11), rstudioapi (>= 0.7), yaml, methods

Suggests testthat (>= 2.0.0)

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

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

License AGPL-3 | file LICENSE

LazyData true

Encoding UTF-8

RoxygenNote 6.0.1

NeedsCompilation no

Author Vincent Nijs [aut, cre]

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

Repository CRAN

Date/Publication 2018-06-15 07:52:57 UTC

R topics documented:

auc	3
catalog	4
confint_robust	5
confusion	5
crs	6
crtree	7
direct_marketing	9
dtree	9
dtree_parser	10
dvd	11
evalbin	11
evalreg	12
find_max	13
find_min	14
houseprices	14
ideal	15
logistic	15
minmax	16
movie_contract	17
nb	17
nn	18
plot.confusion	19
plot.crs	20
plot.crtree	21
plot.dtree	22
plot.evalbin	23
plot.evalreg	24
plot.logistic	24
plot.model.predict	26
plot.nb	27
plot.nb.predict	28
plot.nn	29
plot.regress	30
plot.repeater	31
plot.simulater	32
predict.crtree	33
predict.logistic	34
predict.nb	35
predict.nn	36
predict.regress	37
predict_model	38
print.crtree.predict	39
print.logistic.predict	39
print.nb.predict	40
print.nn.predict	40
print.regress.predict	41

print_predict_model	41
radiant.model	42
radiant.model-deprecated	42
radiant.model_viewer	43
radiant.model_window	43
ratings	44
regress	44
render.DiagrammeR	45
repeater	46
scaledf	47
sdw	48
sensitivity	48
sensitivity.dtree	49
simulater	50
sim_cleaner	51
sim_cor	52
sim_splitter	53
sim_summary	53
store.crs	54
store.model	54
store.model.predict	55
store.nb.predict	56
summary.confusion	57
summary.crs	57
summary.crtree	58
summary.dtree	59
summary.evalbin	60
summary.evalreg	61
summary.logistic	61
summary.nb	63
summary.nn	63
summary.regress	64
summary.repeater	65
summary.simulater	66
test_specs	66
var_check	67
write.coeff	68

Index **69**

auc

Area Under the Curve (AUC)

Description

Area Under the Curve (AUC)

Usage

```
auc(pred, rvar, lev)
```

Arguments

pred	Prediction or predictor
rvar	Response variable
lev	The level in the response variable defined as success

Details

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

Value

AUC statistic

See Also

[evalbin](#) to calculate results
[summary.evalbin](#) to summarize results
[plot.evalbin](#) to plot results

Examples

```
auc(runif(20000), dvd$buy, "yes")  
auc(ifelse(dvd$buy == "yes", 1, 0), dvd$buy, "yes")
```

catalog

Catalog sales for men's and women's apparel

Description

Catalog sales for men's and women's apparel

Usage

```
data(catalog)
```

Format

A data frame with 200 rows and 5 variables

Details

Description provided in `attr(catalog, "description")`

confint_robust	<i>Confidence interval for robust estimators</i>
----------------	--

Description

Confidence interval for robust estimators

Usage

```
confint_robust(object, level = 0.95, dist = "norm", vcov = NULL, ...)
```

Arguments

object	A fitted model object
level	The confidence level required
dist	Distribution to use ("norm" or "t")
vcov	Covariance matrix generated by, e.g., sandwich::vcovHC
...	Additional argument(s) for methods

Details

Wrapper for confint with robust standard errors. See <http://stackoverflow.com/a/3820125/1974918>

confusion	<i>Confusion matrix</i>
-----------	-------------------------

Description

Confusion matrix

Usage

```
confusion(dataset, pred, rvar, lev = "", cost = 1, margin = 2,
  train = "All", data_filter = "", ...)
```

Arguments

dataset	Dataset
pred	Predictions or predictors
rvar	Response variable
lev	The level in the response variable defined as success
cost	Cost for each connection (e.g., email or mailing)

margin	Margin on each customer purchase
train	Use data from training ("Training"), validation ("Validation"), both ("Both"), or all data ("All") to evaluate model evalbin
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")
...	further arguments passed to or from other methods

Details

Confusion matrix and additional metrics to evaluate binary classification models. See <https://radiant-rstats.github.io/docs/model/evalbin.html> for an example in Radiant

Value

A list of results

See Also

[summary.confusion](#) to summarize results

[plot.confusion](#) to plot results

Examples

```
data.frame(buy = dvd$buy, pred1 = runif(20000), pred2 = ifelse(dvd$buy == "yes", 1, 0)) %>%
  confusion(c("pred1", "pred2"), "buy") %>%
  str()
```

crs

Collaborative Filtering

Description

Collaborative Filtering

Usage

```
crs(dataset, id, prod, pred, rate, data_filter = "")
```

Arguments

dataset	Dataset
id	String with name of the variable containing user ids
prod	String with name of the variable with product ids
pred	Products to predict for
rate	String with name of the variable with product ratings
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "training == 1")

Details

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

Value

A data.frame with the original data and a new column with predicted ratings

See Also

[summary.crs](#) to summarize results

[plot.crs](#) to plot results if the actual ratings are available

Examples

```
crs(ratings, id = "Users", prod = "Movies", pred = c("M6", "M7", "M8", "M9", "M10"),
    rate = "Ratings", data_filter = "training == 1") %>% str()
```

 crtree

Classification and regression trees based on the rpart package

Description

Classification and regression trees based on the rpart package

Usage

```
crtree(dataset, rvar, evar, type = "", lev = "", wts = "None",
        minsplit = 2, minbucket = round(minsplit/3), cp = 0.001, nodes = NA,
        K = 10, seed = 1234, split = "gini", prior = NA, adjprob = TRUE,
        cost = NA, margin = NA, check = "", data_filter = "")
```

Arguments

dataset	Dataset
rvar	The response variable in the model
evar	Explanatory variables in the model
type	Model type (i.e., "classification" or "regression")
lev	The level in the response variable defined as <code>_success_</code>
wts	Weights to use in estimation
minsplit	The minimum number of observations that must exist in a node in order for a split to be attempted.
minbucket	the minimum number of observations in any terminal <leaf> node. If only one of minbucket or minsplit is specified, the code either sets minsplit to minbucket*3 or minbucket to minsplit/3, as appropriate.

cp	Minimum proportion of root node deviance required for split (default = 0.00001)
nodes	Maximum size of tree in number of nodes to return. If equal to NA no pruning is done
K	Number of folds use in cross-validation
seed	Random seed used for cross-validation
split	Splitting criterion to use (i.e., "gini" or "information")
prior	Adjust the initial probability for the selected level (e.g., set to .5 in unbalanced samples)
adjprob	Setting a prior will rescale the predicted probabilities. Set adjprob to TRUE to adjust the probabilities back to their original scale after estimation
cost	Cost for each connection (e.g., email or mailing)
margin	Margin on each customer purchase
check	Optional estimation parameters (e.g., "standardize")
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

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

Value

A list with all variables defined in crtree as an object of class tree

See Also

[summary.crtree](#) to summarize results

[plot.crtree](#) to plot results

[predict.crtree](#) for prediction

Examples

```
crtree(titanic, "survived", c("pclass", "sex"), lev = "Yes") %>% summary()
result <- crtree(titanic, "survived", c("pclass", "sex")) %>% summary()
result <- crtree(diamonds, "price", c("carat", "clarity"), type = "regression") %>% str()
```

direct_marketing	<i>Direct marketing data</i>
------------------	------------------------------

Description

Direct marketing data

Usage

```
data(direct_marketing)
```

Format

A data frame with 1,000 rows and 12 variables

Details

Description provided in `attr(direct_marketing, "description")`

dtree	<i>Create a decision tree</i>
-------	-------------------------------

Description

Create a decision tree

Usage

```
dtree(y1, opt = "max", base = character(0))
```

Arguments

y1	A yaml string or a list (e.g., from <code>yaml::yaml.load_file()</code>)
opt	Find the maximum ("max") or minimum ("min") value for each decision node
base	List of variable definitions from a base tree used when calling a sub-tree

Details

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

Value

A list with the initial tree and the calculated tree

See Also

[summary.dtree](#) to summarize results

[plot.dtree](#) to plot results

[sensitivity.dtree](#) to plot results

Examples

```
yaml::as.yaml(movie_contract) %>% cat()  
dtree(movie_contract, opt = "max") %>% summary(output = TRUE)
```

dtree_parser

Parse yaml input for dtree to provide (more) useful error messages

Description

Parse yaml input for dtree to provide (more) useful error messages

Usage

```
dtree_parser(y1)
```

Arguments

y1 A yaml string

Details

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

Value

An updated yaml string or a vector messages to return to the users

See Also

[dtree](#) to calculate tree

[summary.dtree](#) to summarize results

[plot.dtree](#) to plot results

dvd	<i>Data on DVD sales</i>
-----	--------------------------

Description

Data on DVD sales

Usage

```
data(dvd)
```

Format

A data frame with 20,000 rows and 4 variables

Details

Binary purchase response to coupon value. Description provided in `attr(dvd,"description")`

evalbin	<i>Evaluate the performance of different (binary) classification models</i>
---------	---

Description

Evaluate the performance of different (binary) classification models

Usage

```
evalbin(dataset, pred, rvar, lev = "", qnt = 10, cost = 1, margin = 2,
        train = "All", data_filter = "")
```

Arguments

dataset	Dataset
pred	Predictions or predictors
rvar	Response variable
lev	The level in the response variable defined as success
qnt	Number of bins to create
cost	Cost for each connection (e.g., email or mailing)
margin	Margin on each customer purchase
train	Use data from training ("Training"), validation ("Validation"), both ("Both"), or all data ("All") to evaluate model evalbin
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

Evaluate different (binary) classification models based on predictions. See <https://radiant-rstats.github.io/docs/model/evalbin.html> for an example in Radiant

Value

A list of results

See Also

[summary.evalbin](#) to summarize results

[plot.evalbin](#) to plot results

Examples

```
data.frame(buy = dvd$buy, pred1 = runif(20000), pred2 = ifelse(dvd$buy == "yes", 1, 0)) %>%
  evalbin(c("pred1", "pred2"), "buy") %>%
  str()
```

 evalreg

Evaluate the performance of different regression models

Description

Evaluate the performance of different regression models

Usage

```
evalreg(dataset, pred, rvar, train = "All", data_filter = "")
```

Arguments

dataset	Dataset
pred	Predictions or predictors
rvar	Response variable
train	Use data from training ("Training"), validation ("Validation"), both ("Both"), or all data ("All") to evaluate model evalreg
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "training == 1")

Details

Evaluate different regression models based on predictions. See <https://radiant-rstats.github.io/docs/model/evalreg.html> for an example in Radiant

Value

A list of results

See Also

[summary.evalreg](#) to summarize results

[plot.evalreg](#) to plot results

Examples

```
data.frame(price = diamonds$price, pred1 = rnorm(3000), pred2 = diamonds$price) %>%  
  evalreg(pred = c("pred1", "pred2"), "price") %>%  
  str()
```

find_max

Find maximum value of a vector

Description

Find maximum value of a vector

Usage

```
find_max(x, y)
```

Arguments

x Variable to find the maximum for
y Variable to find the value for at the maximum of var

Details

Find the value of y at the maximum value of x

Value

Value of val at the maximum of var

Examples

```
find_min(1:10, 20:30)
```

find_min *Find minimum value of a vector*

Description

Find minimum value of a vector

Usage

```
find_min(x, y)
```

Arguments

x Variable to find the minimum for
y Variable to find the value for at the maximum of var

Details

Find the value of y at the minimum value of x

Value

Value of val at the minimum of var

Examples

```
find_min(1:10, 20:30)
```

houseprices *Houseprices*

Description

Houseprices

Usage

```
data(houseprices)
```

Format

A data frame with 128 home sales and 6 variables

Details

Description provided in attr(houseprices, "description")

ideal	<i>Ideal data for linear regression</i>
-------	---

Description

Ideal data for linear regression

Usage

```
data(ideal)
```

Format

A data frame with 1,000 rows and 4 variables

Details

Description provided in `attr(ideal, "description")`

logistic	<i>Logistic regression</i>
----------	----------------------------

Description

Logistic regression

Usage

```
logistic(dataset, rvar, evar, lev = "", int = "", wts = "None",
  check = "", ci_type, data_filter = "")
```

Arguments

dataset	Dataset
rvar	The response variable in the model
evar	Explanatory variables in the model
lev	The level in the response variable defined as <code>_success_</code>
int	Interaction term to include in the model
wts	Weights to use in estimation
check	Use "standardize" to see standardized coefficient estimates. Use "stepwise-backward" (or "stepwise-forward", or "stepwise-both") to apply step-wise selection of variables in estimation. Add "robust" for robust estimation of standard errors (HC1)

ci_type	To use the profile-likelihood (rather than Wald) for confidence intervals use "profile". For datasets with more than 5,000 rows the Wald method will be used, unless "profile" is explicitly set
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

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

Value

A list with all variables defined in logistic as an object of class logistic

See Also

`summary.logistic` to summarize the results
`plot.logistic` to plot the results
`predict.logistic` to generate predictions
`plot.model.predict` to plot prediction output

Examples

```
logistic(titanic, "survived", c("pclass", "sex"), lev = "Yes") %>% summary()
logistic(titanic, "survived", c("pclass", "sex")) %>% str()
```

minmax

Calculate min and max before standardization

Description

Calculate min and max before standardization

Usage

```
minmax(dataset)
```

Arguments

dataset Data frame

Value

Data frame min and max attributes

movie_contract	<i>Movie contract decision tree</i>
----------------	-------------------------------------

Description

Movie contract decision tree

Usage

```
data(movie_contract)
```

Format

A nested list for decision and chance nodes, probabilities and payoffs

Details

Use decision analysis to create a decision tree for an actor facing a contract decision

nb	<i>Naive Bayes using e1071::naiveBayes</i>
----	--

Description

Naive Bayes using e1071::naiveBayes

Usage

```
nb(dataset, rvar, evar, laplace = 0, data_filter = "")
```

Arguments

dataset	Dataset
rvar	The response variable in the logit (probit) model
evar	Explanatory variables in the model
laplace	Positive double controlling Laplace smoothing. The default (0) disables Laplace smoothing.
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

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

Value

A list with all variables defined in `nb` as an object of class `nb`

See Also

[summary.nb](#) to summarize results

[plot.nb](#) to plot results

[predict.nb](#) for prediction

Examples

```
nb(titanic, "survived", c("pclass", "sex", "age")) %>% summary()
nb(titanic, "survived", c("pclass", "sex", "age")) %>% str()
```

 nn

Neural Networks

Description

Neural Networks

Usage

```
nn(dataset, rvar, evar, type = "classification", lev = "", size = 1,
    decay = 0.5, wts = "None", seed = NA, check = "standardize",
    data_filter = "")
```

Arguments

<code>dataset</code>	Dataset
<code>rvar</code>	The response variable in the model
<code>evar</code>	Explanatory variables in the model
<code>type</code>	Model type (i.e., "classification" or "regression")
<code>lev</code>	The level in the response variable defined as <code>_success_</code>
<code>size</code>	Number of units (nodes) in the hidden layer
<code>decay</code>	Parameter decay
<code>wts</code>	Weights to use in estimation
<code>seed</code>	Random seed to use as the starting point
<code>check</code>	Optional estimation parameters ("standardize" is the default)
<code>data_filter</code>	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

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

Value

A list with all variables defined in nn as an object of class nn

See Also

[summary.nn](#) to summarize results

[plot.nn](#) to plot results

[predict.nn](#) for prediction

Examples

```
nn(titanic, "survived", c("pclass", "sex"), lev = "Yes") %>% summary()
nn(titanic, "survived", c("pclass", "sex")) %>% str()
nn(diamonds, "price", c("carat", "clarity"), type = "regression") %>% summary()
```

plot.confusion	<i>Plot method for the confusion matrix</i>
----------------	---

Description

Plot method for the confusion matrix

Usage

```
## S3 method for class 'confusion'
plot(x, vars = c("kappa", "index", "ROME", "AUC"),
     scale_y = TRUE, size = 13, ...)
```

Arguments

x	Return value from confusion
vars	Measures to plot, i.e., one or more of "TP", "FP", "TN", "FN", "total", "TPR", "TNR", "precision", "accuracy", "kappa", "profit", "index", "ROME", "contact", "AUC"
scale_y	Free scale in faceted plot of the confusion matrix (TRUE or FALSE)
size	Font size used
...	further arguments passed to or from other methods

Details

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

See Also

[confusion](#) to generate results

[summary.confusion](#) to summarize results

Examples

```
data.frame(buy = dvd$buy, pred1 = runif(20000), pred2 = ifelse(dvd$buy == "yes", 1, 0)) %>%  
  confusion(c("pred1", "pred2"), "buy") %>%  
  plot()
```

plot.crs

Plot method for the crs function

Description

Plot method for the crs function

Usage

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

Arguments

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

Details

Plot that compares actual to predicted ratings. See <https://radiant-rstats.github.io/docs/model/crs.html> for an example in Radiant

See Also

[crs](#) to generate results

[summary.crs](#) to summarize results

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

Description

Plot method for the crtree function

Usage

```
## S3 method for class 'crtree'  
plot(x, plots = "tree", orient = "LR", width = "900px",  
      labs = TRUE, dec = 2, shiny = FALSE, custom = FALSE, ...)
```

Arguments

<code>x</code>	Return value from crtree
<code>plots</code>	Plots to produce for the specified rpart tree. "tree" shows a tree diagram. "prune" shows a line graph to evaluate appropriate tree pruning. "imp" shows a variable importance plot
<code>orient</code>	Plot orientation for tree: LR for vertical and TD for horizontal
<code>width</code>	Plot width in pixels for tree (default is "900px")
<code>labs</code>	Use factor labels in plot (TRUE) or revert to default letters used by tree (FALSE)
<code>dec</code>	Decimal places to round results to
<code>shiny</code>	Did the function call originate inside a shiny app
<code>custom</code>	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org for options.
<code>...</code>	further arguments passed to or from other methods

Details

See <https://radiant-rstats.github.io/docs/model/crtree.html> for an example in Radiant. The standard tree plot used by the rpart package can be generated by `plot.rpart(result$model)`. See [plot.rpart](#) for additional details.

See Also

[crtree](#) to generate results
[summary.crtree](#) to summarize results
[predict.crtree](#) for prediction

Examples

```

result <- ctree(titanic, "survived", c("pclass", "sex"), lev = "Yes")
plot(result)
result <- ctree(diamonds, "price", c("carat", "clarity", "cut"))
plot(result, plots = "prune")
result <- ctree(dvd, "buy", c("coupon", "purch", "last"), cp = .01)
plot(result, plots = "imp")

```

plot.dtree

Plot method for the dtree function

Description

Plot method for the dtree function

Usage

```

## S3 method for class 'dtree'
plot(x, symbol = "$", dec = 2, final = FALSE,
     orient = "LR", width = "900px", ...)

```

Arguments

x	Return value from dtree
symbol	Monetary symbol to use (\$ is the default)
dec	Decimal places to round results to
final	If TRUE plot the decision tree solution, else the initial decision tree
orient	Plot orientation: LR for vertical and TD for horizontal
width	Plot width in pixels (default is "900px")
...	further arguments passed to or from other methods

Details

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

See Also

[dtree](#) to generate the result
[summary.dtree](#) to summarize results
[sensitivity.dtree](#) to plot results

Examples

```

dtree(movie_contract, opt = "max") %>% plot()
dtree(movie_contract, opt = "max") %>% plot(final = TRUE, orient = "TD")

```

plot.evalbin *Plot method for the evalbin function*

Description

Plot method for the evalbin function

Usage

```
## S3 method for class 'evalbin'  
plot(x, plots = c("lift", "gains"), size = 13,  
      shiny = FALSE, custom = FALSE, ...)
```

Arguments

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

Details

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

See Also

[evalbin](#) to generate results
[summary.evalbin](#) to summarize results

Examples

```
data.frame(buy = dvd$buy, pred1 = runif(20000), pred2 = ifelse(dvd$buy == "yes", 1, 0)) %>%  
  evalbin(c("pred1", "pred2"), "buy") %>%  
  plot()
```

plot.evalreg *Plot method for the evalreg function*

Description

Plot method for the evalreg function

Usage

```
## S3 method for class 'evalreg'  
plot(x, vars = c("Rsqr", "RMSE", "MAE"), ...)
```

Arguments

x	Return value from evalreg
vars	Measures to plot, i.e., one or more of "Rsqr", "RMSE", "MAE"
...	further arguments passed to or from other methods

Details

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

See Also

[evalreg](#) to generate results
[summary.evalreg](#) to summarize results

Examples

```
data.frame(price = diamonds$price, pred1 = rnorm(3000), pred2 = diamonds$price) %>%  
  evalreg(pred = c("pred1", "pred2"), "price") %>%  
  plot()
```

plot.logistic *Plot method for the logistic function*

Description

Plot method for the logistic function

Usage

```
## S3 method for class 'logistic'  
plot(x, plots = "", conf_lev = 0.95, intercept = FALSE,  
      nrobs = -1, shiny = FALSE, custom = FALSE, ...)
```

Arguments

x	Return value from logistic
plots	Plots to produce for the specified GLM model. Use "" to avoid showing any plots (default). "dist" shows histograms (or frequency bar plots) of all variables in the model. "scatter" shows scatter plots (or box plots for factors) for the response variable with each explanatory variable. "coef" provides a coefficient plot
conf_lev	Confidence level to use for coefficient and odds confidence intervals (.95 is the default)
intercept	Include the intercept in the coefficient plot (TRUE or FALSE). FALSE is the default
nrobs	Number of data points to show in scatter plots (-1 for all)
shiny	Did the function call originate inside a shiny app
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org for options.
...	further arguments passed to or from other methods

Details

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

See Also

[logistic](#) to generate results
[plot.logistic](#) to plot results
[predict.logistic](#) to generate predictions
[plot.model.predict](#) to plot prediction output

Examples

```
result <- logistic(titanic, "survived", c("pclass", "sex"), lev = "Yes")  
plot(result, plots = "coef")
```

plot.model.predict *Plot method for model.predict functions*

Description

Plot method for model.predict functions

Usage

```
## S3 method for class 'model.predict'
plot(x, xvar = "", facet_row = ".",
     facet_col = ".", color = "none", conf_lev = 0.95, ...)
```

Arguments

x	Return value from predict functions (e.g., predict.regress)
xvar	Variable to display along the X-axis of the plot
facet_row	Create vertically arranged subplots for each level of the selected factor variable
facet_col	Create horizontally arranged subplots for each level of the selected factor variable
color	Adds color to a scatter plot to generate a heat map. For a line plot one line is created for each group and each is assigned a different color
conf_lev	Confidence level to use for prediction intervals (.95 is the default)
...	further arguments passed to or from other methods

See Also

[predict.regress](#) to generate predictions

[predict.logistic](#) to generate predictions

Examples

```
regress(diamonds, "price", c("carat", "clarity")) %>%
  predict(pred_cmd = "carat = 1:10") %>%
  plot(xvar = "carat")
logistic(titanic, "survived", c("pclass", "sex", "age"), lev = "Yes") %>%
  predict(pred_cmd = c("pclass = levels(pclass)", "sex = levels(sex)", "age = 0:100")) %>%
  plot(xvar = "age", color = "sex", facet_col = "pclass")
```

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

Description

Plot method for the nb function

Usage

```
## S3 method for class 'nb'  
plot(x, plots = "correlations", lev = "All levels",  
      nrobs = 1000, ...)
```

Arguments

<code>x</code>	Return value from <code>nb</code>
<code>plots</code>	Plots to produce for the specified model. Use "" to avoid showing any plots. Use "vimp" for variable importance or "correlations" to examine conditional independence
<code>lev</code>	The level(s) in the response variable used as the basis for plots (defaults to "All levels")
<code>nrobs</code>	Number of data points to show in scatter plots (-1 for all)
<code>...</code>	further arguments passed to or from other methods

Details

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

See Also

`nb` to generate results
`summary.nb` to summarize results
`predict.nb` for prediction

Examples

```
result <- nb(titanic, "survived", c("pclass", "sex"))  
plot(result)  
result <- nb(titanic, "pclass", c("sex", "age"))  
plot(result)
```

plot.nb.predict *Plot method for nb.predict function*

Description

Plot method for nb.predict function

Usage

```
## S3 method for class 'nb.predict'  
plot(x, xvar = "", facet_row = ".", facet_col = ".",  
     color = ".class", ...)
```

Arguments

x	Return value from predict function predict.nb
xvar	Variable to display along the X-axis of the plot
facet_row	Create vertically arranged subplots for each level of the selected factor variable
facet_col	Create horizontally arranged subplots for each level of the selected factor variable
color	Adds color to a scatter plot to generate a heat map. For a line plot one line is created for each group and each is assigned a different color
...	further arguments passed to or from other methods

See Also

[predict.nb](#) to generate predictions

Examples

```
result <- nb(titanic, "survived", c("pclass", "sex", "age"))  
pred <- predict(  
  result,  
  pred_cmd = c("pclass = levels(pclass)", "sex = levels(sex)", "age = seq(0, 100, 20)")  
)  
plot(pred, xvar = "age", facet_col = "sex", facet_row = "pclass")  
pred <- predict(result, pred_data = titanic)  
plot(pred, xvar = "age", facet_col = "sex")
```

plot.nn *Plot method for the nn function*

Description

Plot method for the nn function

Usage

```
## S3 method for class 'nn'  
plot(x, plots = "garson", size = 12, nrobs = -1,  
     shiny = FALSE, custom = FALSE, ...)
```

Arguments

x	Return value from <code>nn</code>
plots	Plots to produce for the specified Neural Network model. Use "" to avoid showing any plots (default). Options are "olden" or "garson" for importance plots, or "net" to depict the network structure
size	Font size used
nrobs	Number of data points to show in scatter plots (-1 for all)
shiny	Did the function call originate inside a shiny app
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org for options.
...	further arguments passed to or from other methods

Details

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

See Also

`nn` to generate results
`summary.nn` to summarize results
`predict.nn` for prediction

Examples

```
result <- nn(titanic, "survived", c("pclass", "sex"), lev = "Yes")  
plot(result, plots = "net")  
plot(result, plots = "olden")
```

 plot.regress

Plot method for the regress function

Description

Plot method for the regress function

Usage

```
## S3 method for class 'regress'
plot(x, plots = "", lines = "", conf_lev = 0.95,
      intercept = FALSE, nrobs = -1, shiny = FALSE, custom = FALSE, ...)
```

Arguments

x	Return value from regress
plots	Regression plots to produce for the specified regression model. Enter "" to avoid showing any plots (default). "dist" to shows histograms (or frequency bar plots) of all variables in the model. "correlations" for a visual representation of the correlation matrix selected variables. "scatter" to show scatter plots (or box plots for factors) for the response variable with each explanatory variable. "dashboard" for a series of six plots that can be used to evaluate model fit visually. "resid_pred" to plot the explanatory variables against the model residuals. "coef" for a coefficient plot with adjustable confidence intervals. "leverage" to show leverage plots for each explanatory variable
lines	Optional lines to include in the select plot. "line" to include a line through a scatter plot. "loess" to include a polynomial regression fit line. To include both use c("line", "loess")
conf_lev	Confidence level used to estimate confidence intervals (.95 is the default)
intercept	Include the intercept in the coefficient plot (TRUE, FALSE). FALSE is the default
nrobs	Number of data points to show in scatter plots (-1 for all)
shiny	Did the function call originate inside a shiny app
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org for options.
...	further arguments passed to or from other methods

Details

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

See Also

[regress](#) to generate the results
[summary.regress](#) to summarize results
[predict.regress](#) to generate predictions

Examples

```
result <- regress(diamonds, "price", c("carat", "clarity"))
plot(result, plots = "coef", conf_lev = .99, intercept = TRUE)
## Not run:
plot(result, plots = "dist")
plot(result, plots = "scatter", lines = c("line", "loess"))
plot(result, plots = "resid_pred", lines = "line")
plot(result, plots = "dashboard", lines = c("line", "loess"))

## End(Not run)
```

plot.repeater	<i>Plot repeated simulation</i>
---------------	---------------------------------

Description

Plot repeated simulation

Usage

```
## S3 method for class 'repeater'
plot(x, bins = 20, shiny = FALSE, custom = FALSE, ...)
```

Arguments

x	Return value from repeater
bins	Number of bins used for histograms (1 - 50)
shiny	Did the function call originate inside a shiny app
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org for options.
...	further arguments passed to or from other methods

See Also

[repeater](#) to run a repeated simulation
[summary.repeater](#) to summarize results from repeated simulation

plot.simulater *Plot method for the simulater function*

Description

Plot method for the simulater function

Usage

```
## S3 method for class 'simulater'  
plot(x, bins = 20, shiny = FALSE, custom = FALSE, ...)
```

Arguments

x	Return value from simulater
bins	Number of bins used for histograms (1 - 50)
shiny	Did the function call originate inside a shiny app
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org for options.
...	further arguments passed to or from other methods

Details

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

See Also

[simulater](#) to generate the result
[summary.simulater](#) to summarize results

Examples

```
simdat <- simulater(  
  const = "cost 3",  
  norm = "demand 2000 1000",  
  discrete = "price 5 8 .3 .7",  
  form = "profit = demand * (price - cost)",  
  seed = 1234  
)  
plot(simdat, bins = 25)
```

predict.crtree *Predict method for the crtree function*

Description

Predict method for the crtree function

Usage

```
## S3 method for class 'crtree'
predict(object, pred_data = NULL, pred_cmd = "",
        conf_lev = 0.95, se = FALSE, dec = 3, ...)
```

Arguments

object	Return value from crtree
pred_data	Provide the dataframe to generate predictions (e.g., titanic). The dataset must contain all columns used in the estimation
pred_cmd	Generate predictions using a command. For example, 'pclass = levels(pclass)' would produce predictions for the different levels of factor 'pclass'. To add another variable, create a vector of prediction strings, (e.g., c('pclass = levels(pclass)', 'age = seq(0,100,20)')
conf_lev	Confidence level used to estimate confidence intervals (.95 is the default)
se	Logical that indicates if prediction standard errors should be calculated (default = FALSE)
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

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

See Also

[crtree](#) to generate the result
[summary.crtree](#) to summarize results

Examples

```
result <- crtree(titanic, "survived", c("pclass", "sex"), lev = "Yes")
predict(result, pred_cmd = "pclass = levels(pclass)")
result <- crtree(titanic, "survived", "pclass", lev = "Yes")
predict(result, pred_data = titanic) %>% head()
```

predict.logistic *Predict method for the logistic function*

Description

Predict method for the logistic function

Usage

```
## S3 method for class 'logistic'
predict(object, pred_data = NULL, pred_cmd = "",
        conf_lev = 0.95, se = TRUE, interval = "confidence", dec = 3, ...)
```

Arguments

object	Return value from logistic
pred_data	Provide the dataframe to generate predictions (e.g., titanic). The dataset must contain all columns used in the estimation
pred_cmd	Generate predictions using a command. For example, 'pclass = levels(pclass)' would produce predictions for the different levels of factor 'pclass'. To add another variable, create a vector of prediction strings, (e.g., c('pclass = levels(pclass)', 'age = seq(0,100,20)')
conf_lev	Confidence level used to estimate confidence intervals (.95 is the default)
se	Logical that indicates if prediction standard errors should be calculated (default = FALSE)
interval	Type of interval calculation ("confidence" or "none"). Set to "none" if se is FALSE
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

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

See Also

[logistic](#) to generate the result
[summary.logistic](#) to summarize results
[plot.logistic](#) to plot results
[plot.model.predict](#) to plot prediction output

Examples

```

result <- logistic(titanic, "survived", c("pclass", "sex"), lev = "Yes")
predict(result, pred_cmd = "pclass = levels(pclass)")
logistic(titanic, "survived", c("pclass", "sex"), lev = "Yes") %>%
  predict(pred_cmd = "sex = c('male', 'female')")
logistic(titanic, "survived", c("pclass", "sex"), lev = "Yes") %>%
  predict(pred_data = titanic)

```

predict.nb	<i>Predict method for the nb function</i>
------------	---

Description

Predict method for the nb function

Usage

```

## S3 method for class 'nb'
predict(object, pred_data = NULL, pred_cmd = "",
        pred_names = "", dec = 3, ...)

```

Arguments

object	Return value from nb
pred_data	Provide the dataframe to generate predictions (e.g., titanic). The dataset must contain all columns used in the estimation
pred_cmd	Generate predictions using a command. For example, 'pclass = levels(pclass)' would produce predictions for the different levels of factor 'pclass'. To add another variable, create a vector of prediction strings, (e.g., c('pclass = levels(pclass)', 'age = seq(0,100,20)')
pred_names	Names for the predictions to be stored. If one name is provided, only the first column of predictions is stored. If empty, the level in the response variable of the nb model will be used
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

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

See Also

[nb](#) to generate the result
[summary.nb](#) to summarize results

Examples

```

result <- nb(titanic, "survived", c("pclass", "sex", "age"))
predict(result, pred_data = titanic)
predict(result, pred_data = titanic, pred_names = c("Yes", "No"))
predict(result, pred_cmd = "pclass = levels(pclass)")
result <- nb(titanic, "pclass", c("survived", "sex", "age"))
predict(result, pred_data = titanic)
predict(result, pred_data = titanic, pred_names = c("1st", "2nd", "3rd"))
predict(result, pred_data = titanic, pred_names = "")

```

predict.nn

Predict method for the nn function

Description

Predict method for the nn function

Usage

```

## S3 method for class 'nn'
predict(object, pred_data = NULL, pred_cmd = "", dec = 3,
  ...)

```

Arguments

object	Return value from nn
pred_data	Provide the dataframe to generate predictions (e.g., diamonds). The dataset must contain all columns used in the estimation
pred_cmd	Generate predictions using a command. For example, 'pclass = levels(pclass)' would produce predictions for the different levels of factor 'pclass'. To add another variable, create a vector of prediction strings, (e.g., c('pclass = levels(pclass)', 'age = seq(0,100,20)')
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

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

See Also

[nn](#) to generate the result
[summary.nn](#) to summarize results

Examples

```
result <- nn(titanic, "survived", c("pclass", "sex"), lev = "Yes")
predict(result, pred_cmd = "pclass = levels(pclass)")
result <- nn(diamonds, "price", "carat:color", type = "regression")
predict(result, pred_cmd = "carat = 1:3")
predict(result, pred_data = diamonds) %>% head()
```

predict.regress *Predict method for the regress function*

Description

Predict method for the regress function

Usage

```
## S3 method for class 'regress'
predict(object, pred_data = NULL, pred_cmd = "",
        conf_lev = 0.95, se = TRUE, interval = "confidence", dec = 3, ...)
```

Arguments

object	Return value from regress
pred_data	Provide the dataframe to generate predictions (e.g., diamonds). The dataset must contain all columns used in the estimation
pred_cmd	Command used to generate data for prediction
conf_lev	Confidence level used to estimate confidence intervals (.95 is the default)
se	Logical that indicates if prediction standard errors should be calculated (default = FALSE)
interval	Type of interval calculation ("confidence" or "prediction"). Set to "none" if se is FALSE
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

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

See Also

[regress](#) to generate the result
[summary.regress](#) to summarize results
[plot.regress](#) to plot results

Examples

```

result <- regress(diamonds, "price", c("carat", "clarity"))
predict(result, pred_cmd = "carat = 1:10")
predict(result, pred_cmd = "clarity = levels(clarity)")
result <- regress(diamonds, "price", c("carat", "clarity"), int = "carat:clarity")
predict(result, pred_data = diamonds) %>% head()

```

predict_model

Predict method for model functions

Description

Predict method for model functions

Usage

```

predict_model(object, pfun, mclass, pred_data = NULL, pred_cmd = "",
  conf_lev = 0.95, se = FALSE, dec = 3, ...)

```

Arguments

object	Return value from regress
pfun	Function to use for prediction
mclass	Model class to attach
pred_data	Dataset to use for prediction
pred_cmd	Command used to generate data for prediction (e.g., 'carat = 1:10')
conf_lev	Confidence level used to estimate confidence intervals (.95 is the default)
se	Logical that indicates if prediction standard errors should be calculated (default = FALSE)
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

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

print.crtree.predict *Print method for predict.crtree*

Description

Print method for predict.crtree

Usage

```
## S3 method for class 'crtree.predict'  
print(x, ..., n = 10)
```

Arguments

x	Return value from prediction method
...	further arguments passed to or from other methods
n	Number of lines of prediction results to print. Use -1 to print all lines

print.logistic.predict
Print method for logistic.predict

Description

Print method for logistic.predict

Usage

```
## S3 method for class 'logistic.predict'  
print(x, ..., n = 10)
```

Arguments

x	Return value from prediction method
...	further arguments passed to or from other methods
n	Number of lines of prediction results to print. Use -1 to print all lines

print.nb.predict *Print method for predict.nb*

Description

Print method for predict.nb

Usage

```
## S3 method for class 'nb.predict'  
print(x, ..., n = 10)
```

Arguments

x	Return value from prediction method
...	further arguments passed to or from other methods
n	Number of lines of prediction results to print. Use -1 to print all lines

print.nn.predict *Print method for predict.nn*

Description

Print method for predict.nn

Usage

```
## S3 method for class 'nn.predict'  
print(x, ..., n = 10)
```

Arguments

x	Return value from prediction method
...	further arguments passed to or from other methods
n	Number of lines of prediction results to print. Use -1 to print all lines

`print.regress.predict` *Print method for predict.regress*

Description

Print method for predict.regress

Usage

```
## S3 method for class 'regress.predict'  
print(x, ..., n = 10)
```

Arguments

x	Return value from prediction method
...	further arguments passed to or from other methods
n	Number of lines of prediction results to print. Use -1 to print all lines

`print_predict_model` *Print method for the model prediction*

Description

Print method for the model prediction

Usage

```
print_predict_model(x, ..., n = 10, header = "")
```

Arguments

x	Return value from prediction method
...	further arguments passed to or from other methods
n	Number of lines of prediction results to print. Use -1 to print all lines
header	Header line

<code>radiant.model</code>	<i>radiant.model</i>
----------------------------	----------------------

Description

`radiant.model`
Launch `radiant.model` in the default browser

Usage

```
radiant.model()
```

Details

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

Examples

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

<code>radiant.model-deprecated</code>	<i>Deprecated function(s) in the radiant.model package</i>
---------------------------------------	--

Description

These functions are provided for compatibility with previous versions of `radiant`. They will eventually be removed.

Usage

```
ann(...)
```

Arguments

... Parameters to be passed to the updated functions

Details

`ann` is now a synonym for `nn`

`radiant.model_viewer` *Launch radiant.model in the Rstudio viewer*

Description

Launch `radiant.model` in the Rstudio viewer

Usage

```
radiant.model_viewer()
```

Details

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

Examples

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

`radiant.model_window` *Launch radiant.model in an Rstudio window*

Description

Launch `radiant.model` in an Rstudio window

Usage

```
radiant.model_window()
```

Details

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

Examples

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

ratings	<i>Movie ratings</i>
---------	----------------------

Description

Movie ratings

Usage

```
data(ratings)
```

Format

A data frame with 110 rows and 4 variables

Details

Use collaborative filtering to create recommendations based on ratings from existing users. Description provided in `attr(ratings, "description")`

regress	<i>Linear regression using OLS</i>
---------	------------------------------------

Description

Linear regression using OLS

Usage

```
regress(dataset, rvar, evar, int = "", check = "", data_filter = "")
```

Arguments

dataset	Dataset
rvar	The response variable in the regression
evar	Explanatory variables in the regression
int	Interaction terms to include in the model
check	Use "standardize" to see standardized coefficient estimates. Use "stepwise-backward" (or "stepwise-forward", or "stepwise-both") to apply step-wise selection of variables in estimation. Add "robust" for robust estimation of standard errors (HC1)
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

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

Value

A list of all variables used in the regress function as an object of class regress

See Also

[summary.regress](#) to summarize results

[plot.regress](#) to plot results

[predict.regress](#) to generate predictions

Examples

```
regress(diamonds, "price", c("carat", "clarity"), check = "standardize") %>% summary()  
regress(diamonds, "price", c("carat", "clarity")) %>% str()
```

render.DiagrammeR	<i>Method to render DiagrammeR plots</i>
-------------------	--

Description

Method to render DiagrammeR plots

Usage

```
## S3 method for class 'DiagrammeR'  
render(object, ...)
```

Arguments

object	DiagrammeR plot
...	Additional arguments

repeater	<i>Repeated simulation</i>
----------	----------------------------

Description

Repeated simulation

Usage

```
repeater(dataset, nr = 12, vars = "", grid = "", sum_vars = "",
  byvar = "sim", fun = "sum", form = "", seed = NULL, name = "")
```

Arguments

dataset	Return value from the simulator function
nr	Number times to repeat the simulation
vars	Variables to use in repeated simulation
grid	Character vector of expressions to use in grid search for constants
sum_vars	(Numeric) variables to summaries
byvar	Variable(s) to group data by before summarizing
fun	Functions to use for summarizing
form	A character vector with the formula to apply to the summarized data
seed	Seed for the repeated simulation
name	Deprecated argument

See Also

[summary.repeater](#) to summarize results from repeated simulation

[plot.repeater](#) to plot results from repeated simulation

Examples

```
simdat <- simulator(
  const = c("var_cost 5", "fixed_cost 1000"),
  norm = "E 0 100;",
  discrete = "price 6 8 .3 .7;",
  form = c(
    "demand = 1000 - 50*price + E",
    "profit = demand*(price-var_cost) - fixed_cost",
    "profit_small = profit < 100"
  ),
  seed = 1234
)

repmat <- repeater(
```

```
simdat,  
nr = 12,  
vars = c("E", "price"),  
sum_vars = "profit",  
byvar = "sim",  
form = "profit_365 = profit < 36500",  
seed = 1234,  
)  
  
head(repdatt)  
summary(repdatt)  
plot(repdatt)
```

scaledf

Center or standardize variables in a data frame

Description

Center or standardize variables in a data frame

Usage

```
scaledf(dataset, center = TRUE, scale = TRUE, sf = 2, wts = NULL,  
        calc = TRUE)
```

Arguments

dataset	Data frame
center	Center data (TRUE or FALSE)
scale	Scale data (TRUE or FALSE)
sf	Scaling factor (default is 2)
wts	Weights to use (default is NULL for no weights)
calc	Calculate mean and sd or use attributes attached to dat

Value

Scaled data frame

See Also

[copy_attr](#) to copy attributes from a training to a validation dataset

sdw *Standard deviation of weighted sum of variables*

Description

Standard deviation of weighted sum of variables

Usage

```
sdw(...)
```

Arguments

... A matched number of weights and stocks

Value

A vector of standard deviation estimates

sensitivity *Method to evaluate sensitivity of an analysis*

Description

Method to evaluate sensitivity of an analysis

Usage

```
sensitivity(object, ...)
```

Arguments

object Object of relevant class for which to evaluate sensitivity
... Additional arguments

See Also

[sensitivity.dtree](#) to plot results

sensitivity.dtree *Evaluate sensitivity of the decision tree*

Description

Evaluate sensitivity of the decision tree

Usage

```
## S3 method for class 'dtree'  
sensitivity(object, vars = NULL, decs = NULL,  
            shiny = FALSE, custom = FALSE, ...)
```

Arguments

object	Return value from dtree
vars	Variables to include in the sensitivity analysis
decs	Decisions to include in the sensitivity analysis
shiny	Did the function call originate inside a shiny app
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org for options.
...	Additional arguments

Details

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

See Also

[dtree](#) to generate the result
[plot.dtree](#) to summarize results
[summary.dtree](#) to summarize results

Examples

```
dtree(movie_contract, opt = "max") %>%  
  sensitivity(  
    vars = "legal fees 0 100000 10000",  
    decs = c("Sign with Movie Company", "Sign with TV Network"),  
    custom = FALSE  
  )
```

 simulater

Simulate data for decision analysis

Description

Simulate data for decision analysis

Usage

```
simulater(const = "", lnorm = "", norm = "", unif = "", discrete = "",
  binom = "", pois = "", sequ = "", grid = "", data = NULL,
  form = "", seed = NULL, nexact = FALSE, ncorr = NULL, name = "",
  nr = 1000, dataset = NULL)
```

Arguments

const	A character vector listing the constants to include in the analysis (e.g., c("cost = 3", "size = 4"))
lnorm	A character vector listing the log-normally distributed random variables to include in the analysis (e.g., "demand 2000 1000" where the first number is the log-mean and the second is the log-standard deviation)
norm	A character vector listing the normally distributed random variables to include in the analysis (e.g., "demand 2000 1000" where the first number is the mean and the second is the standard deviation)
unif	A character vector listing the uniformly distributed random variables to include in the analysis (e.g., "demand 0 1" where the first number is the minimum value and the second is the maximum value)
discrete	A character vector listing the random variables with a discrete distribution to include in the analysis (e.g., "price 5 8 .3 .7" where the first set of numbers are the values and the second set the probabilities)
binom	A character vector listing the random variables with a binomial distribution to include in the analysis (e.g., "crash 100 .01") where the first number is the number of trials and the second is the probability of success)
pois	A character vector listing the random variables with a poisson distribution to include in the analysis (e.g., "demand 10") where the number is the lambda value (i.e., the average number of events or the event rate)
sequ	A character vector listing the start and end for a sequence to include in the analysis (e.g., "trend 1 100 1"). The number of 'steps' is determined by the number of simulations
grid	A character vector listing the start, end, and step for a set of sequences to include in the analysis (e.g., "trend 1 100 1"). The number of rows in the expanded will over ride the number of simulations
data	Dataset to be used in the calculations

form	A character vector with the formula to evaluate (e.g., "profit = demand * (price - cost)")
seed	Optional seed used in simulation
nexact	Logical to indicate if normally distributed random variables should be simulated to the exact specified values
ncorr	A string of correlations used for normally distributed random variables. The number of values should be equal to one or to the number of combinations of variables simulated
name	Deprecated argument
nr	Number of simulations
dataset	Data list from previous simulation. Used by repeater function

Details

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

Value

A data.frame with the simulated data

See Also

[summary.simulator](#) to summarize results

[plot.simulator](#) to plot results

Examples

```
simulator(
  const = "cost 3",
  norm = "demand 2000 1000",
  discrete = "price 5 8 .3 .7",
  form = "profit = demand * (price - cost)",
  seed = 1234
) %>% str()
```

sim_cleaner

Clean input command string

Description

Clean input command string

Usage

```
sim_cleaner(x)
```

Arguments

x Input string

Value

Cleaned string

sim_cor	<i>Simulate correlated normally distributed data</i>
---------	--

Description

Simulate correlated normally distributed data

Usage

```
sim_cor(n, rho, means, sds, exact = FALSE)
```

Arguments

n The number of values to simulate (i.e., the number of rows in the simulated data)

rho A vector of correlations to apply to the columns of the simulated data. The number of values should be equal to one or to the number of combinations of variables to be simulated

means A vector of means. The number of values should be equal to the number of variables to simulate

sds A vector of standard deviations. The number of values should be equal to the number of variables to simulate

exact A logical that indicates if the inputs should be interpreted as population of sample characteristics

Value

A data.frame with the simulated data

Examples

```
sim <- sim_cor(100, .74, c(0, 10), c(1, 5), exact = TRUE)
cor(sim)
sim_summary(sim)
```

sim_splitter	<i>Split input command string</i>
--------------	-----------------------------------

Description

Split input command string

Usage

```
sim_splitter(x, symbol = " ")
```

Arguments

x	Input string
symbol	Symbol used to split the command string

Value

Split input command string

sim_summary	<i>Print simulation summary</i>
-------------	---------------------------------

Description

Print simulation summary

Usage

```
sim_summary(dataset, dc = get_class(dataset), fun = "", dec = 4)
```

Arguments

dataset	Simulated data
dc	Variable classes
fun	Summary function to apply
dec	Number of decimals to show

See Also

[simulater](#) to run a simulation

[repeater](#) to run a repeated simulation

Examples

```

simulater(
  const = "cost 3",
  norm = "demand 2000 1000",
  discrete = "price 5 8 .3 .7",
  form = c("profit = demand * (price - cost)", "profit5K = profit > 5000"),
  seed = 1234
) %>% sim_summary()

```

store.crs

Deprecated: Store method for the crs function

Description

Deprecated: Store method for the crs function

Usage

```

## S3 method for class 'crs'
store(dataset, object, name, ...)

```

Arguments

dataset	Dataset
object	Return value from crs
name	Name to assign to the dataset
...	further arguments passed to or from other methods

Details

Return recommendations See <https://radiant-rstats.github.io/docs/model/crs.html> for an example in Radiant

store.model

Store residuals from a model

Description

Store residuals from a model

Usage

```

## S3 method for class 'model'
store(dataset, object, name = "residuals", ...)

```

Arguments

dataset	Dataset to append residuals to
object	Return value from a model function
name	Variable name(s) assigned to model residuals
...	Additional arguments

Details

The store method for objects of class "model". Adds model residuals to the dataset while handling missing values and filters. See <https://radiant-rstats.github.io/docs/model/regress.html> for an example in Radiant

Examples

```
regress(diamonds, rvar = "price", evar = c("carat", "cut"), data_filter = "price > 1000") %>%
  store(diamonds, ., name = "resid") %>%
  head()
```

store.model.predict *Store predicted values generated in model functions*

Description

Store predicted values generated in model functions

Usage

```
## S3 method for class 'model.predict'
store(dataset, object, name = "prediction", ...)
```

Arguments

dataset	Dataset to add predictions to
object	Return value from model function
name	Variable name(s) assigned to predicted values
...	Additional arguments

Details

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

Examples

```
regress(diamonds, rvar = "price", evar = c("carat", "cut")) %>%
  predict(pred_data = diamonds) %>%
  store(diamonds, ., name = c("pred", "pred_low", "pred_high")) %>%
  head()
```

store.nb.predict	<i>Store predicted values generated in the nb function</i>
------------------	--

Description

Store predicted values generated in the nb function

Usage

```
## S3 method for class 'nb.predict'
store(dataset, object, name = "pred_nb", ...)
```

Arguments

dataset	Dataset to add predictions two
object	Return value from model function
name	Variable name(s) assigned to predicted values. If empty, the levels of the response variable will be used
...	Additional arguments

Details

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

Examples

```
result <- nb(titanic, rvar = "survived", evar = c("pclass", "sex", "age"))
pred <- predict(result, pred_data = titanic)
titanic <- store(titanic, pred, name = c("Yes", "No"))
```

summary.confusion *Summary method for the confusion matrix*

Description

Summary method for the confusion matrix

Usage

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

Arguments

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

Details

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

See Also

[confusion](#) to generate results
[plot.confusion](#) to visualize result

Examples

```
data.frame(buy = dvd$buy, pred1 = runif(20000), pred2 = ifelse(dvd$buy == "yes", 1, 0)) %>%  
  confusion(c("pred1", "pred2"), "buy") %>%  
  summary()
```

summary.crs *Summary method for Collaborative Filter*

Description

Summary method for Collaborative Filter

Usage

```
## S3 method for class 'crs'  
summary(object, n = 36, dec = 2, ...)
```

Arguments

object	Return value from crs
n	Number of lines of recommendations to print. Use -1 to print all lines
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

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

See Also

[crs](#) to generate the results
[plot.crs](#) to plot results if the actual ratings are available

Examples

```
crs(ratings, id = "Users", prod = "Movies", pred = c("M6", "M7", "M8", "M9", "M10"),
    rate = "Ratings", data_filter = "training == 1") %>% summary()
```

summary.crtree	<i>Summary method for the crtree function</i>
----------------	---

Description

Summary method for the crtree function

Usage

```
## S3 method for class 'crtree'
summary(object, prn = TRUE, cptab = FALSE,
        modsum = FALSE, ...)
```

Arguments

object	Return value from crtree
prn	Print tree in text form
cptab	Print the cp table
modsum	Print the model summary
...	further arguments passed to or from other methods

Details

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

See Also

[crtree](#) to generate results
[plot.crtree](#) to plot results
[predict.crtree](#) for prediction

Examples

```
result <- crtree(titanic, "survived", c("pclass", "sex"), lev = "Yes")
summary(result)
result <- crtree(diamonds, "price", c("carat", "color"), type = "regression")
summary(result)
```

summary.dtree

*Summary method for the dtree function***Description**

Summary method for the dtree function

Usage

```
## S3 method for class 'dtree'
summary(object, input = TRUE, output = FALSE, dec = 2,
  ...)
```

Arguments

object	Return value from simulater
input	Print decision tree input
output	Print decision tree output
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

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

See Also

[dtree](#) to generate the results
[plot.dtree](#) to plot results
[sensitivity.dtree](#) to plot results

Examples

```
dtree(movie_contract, opt = "max") %>% summary(input = TRUE)
dtree(movie_contract, opt = "max") %>% summary(input = FALSE, output = TRUE)
```

summary.evalbin	<i>Summary method for the evalbin function</i>
-----------------	--

Description

Summary method for the evalbin function

Usage

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

Arguments

object	Return value from <code>evalbin</code>
prn	Print full table of measures per model and bin
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

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

See Also

`evalbin` to summarize results

`plot.evalbin` to plot results

Examples

```
data.frame(buy = dvd$buy, pred1 = runif(20000), pred2 = ifelse(dvd$buy == "yes", 1, 0)) %>%
  evalbin(c("pred1", "pred2"), "buy") %>%
  summary()
```

summary.evalreg	<i>Summary method for the evalreg function</i>
-----------------	--

Description

Summary method for the evalreg function

Usage

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

Arguments

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

Details

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

See Also

`evalreg` to summarize results

`plot.evalreg` to plot results

Examples

```
data.frame(price = diamonds$price, pred1 = rnorm(3000), pred2 = diamonds$price) %>%  
  evalreg(pred = c("pred1", "pred2"), "price") %>%  
  summary()
```

summary.logistic	<i>Summary method for the logistic function</i>
------------------	---

Description

Summary method for the logistic function

Usage

```
## S3 method for class 'logistic'
summary(object, sum_check = "", conf_lev = 0.95,
        test_var = "", dec = 3, ...)
```

Arguments

object	Return value from <code>logistic</code>
sum_check	Optional output. "vif" to show multicollinearity diagnostics. "confint" to show coefficient confidence interval estimates. "odds" to show odds ratios and confidence interval estimates.
conf_lev	Confidence level to use for coefficient and odds confidence intervals (.95 is the default)
test_var	Variables to evaluate in model comparison (i.e., a competing models Chi-squared test)
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

See <https://radiant-rstats.github.io/docs/model/logistic.html> for an example in Radian

See Also

`logistic` to generate the results
`plot.logistic` to plot the results
`predict.logistic` to generate predictions
`plot.model.predict` to plot prediction output

Examples

```
result <- logistic(titanic, "survived", "pclass", lev = "Yes")
summary(result, test_var = "pclass")
res <- logistic(titanic, "survived", c("pclass", "sex"), int = "pclass:sex", lev = "Yes")
summary(res, sum_check = c("vif", "confint", "odds"))
titanic %>% logistic("survived", c("pclass", "sex", "age"), lev = "Yes") %>% summary("vif")
```

summary.nb	<i>Summary method for the nb function</i>
------------	---

Description

Summary method for the nb function

Usage

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

Arguments

object	Return value from nb
dec	Decimals
...	further arguments passed to or from other methods

Details

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

See Also

[nb](#) to generate results
[plot.nb](#) to plot results
[predict.nb](#) for prediction

Examples

```
result <- nb(titanic, "survived", c("pclass", "sex", "age"))  
summary(result)
```

summary.nn	<i>Summary method for the nn function</i>
------------	---

Description

Summary method for the nn function

Usage

```
## S3 method for class 'nn'  
summary(object, prn = TRUE, ...)
```

Arguments

object	Return value from <code>nn</code>
prn	Print list of weights
...	further arguments passed to or from other methods

Details

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

See Also

`nn` to generate results
`plot.nn` to plot results
`predict.nn` for prediction

Examples

```
result <- nn(titanic, "survived", "pclass", lev = "Yes")
summary(result)
```

summary.regress	<i>Summary method for the regress function</i>
-----------------	--

Description

Summary method for the regress function

Usage

```
## S3 method for class 'regress'
summary(object, sum_check = "", conf_lev = 0.95,
        test_var = "", dec = 3, ...)
```

Arguments

object	Return value from <code>regress</code>
sum_check	Optional output. "rsme" to show the root mean squared error and the standard deviation of the residuals. "sumsquares" to show the sum of squares table. "vif" to show multicollinearity diagnostics. "confint" to show coefficient confidence interval estimates.
conf_lev	Confidence level used to estimate confidence intervals (.95 is the default)
test_var	Variables to evaluate in model comparison (i.e., a competing models F-test)
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

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

See Also

[regress](#) to generate the results
[plot.regress](#) to plot results
[predict.regress](#) to generate predictions

Examples

```
result <- regress(diamonds, "price", c("carat", "clarity"))
summary(result, sum_check = c("rmse", "sumsquares", "vif", "confint"), test_var = "clarity")
result <- regress(ideal, "y", c("x1", "x2"))
summary(result, test_var = "x2")
ideal %>% regress("y", "x1:x3") %>% summary()
```

summary.repeater	<i>Summarize repeated simulation</i>
------------------	--------------------------------------

Description

Summarize repeated simulation

Usage

```
## S3 method for class 'repeater'
summary(object, dec = 4, ...)
```

Arguments

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

See Also

[repeater](#) to run a repeated simulation
[plot.repeater](#) to plot results from repeated simulation

summary.simulater	<i>Summary method for the simulater function</i>
-------------------	--

Description

Summary method for the simulater function

Usage

```
## S3 method for class 'simulater'  
summary(object, dec = 4, ...)
```

Arguments

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

Details

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

See Also

[simulater](#) to generate the results
[plot.simulater](#) to plot results

Examples

```
simdat <- simulater(norm = "demand 2000 1000", seed = 1234)  
summary(simdat)
```

test_specs	<i>Add interaction terms to list of test variables if needed</i>
------------	--

Description

Add interaction terms to list of test variables if needed

Usage

```
test_specs(test_var, int)
```

Arguments

test_var	List of variables to use for testing for regress or logistic
int	Interaction terms specified

Details

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

Value

A vector of variables names to test

Examples

```
test_specs("a", c("a:b", "b:c"))
```

var_check	<i>Check if main effects for all interaction effects are included in the model</i>
-----------	--

Description

Check if main effects for all interaction effects are included in the model

Usage

```
var_check(ev, cn, intv = "")
```

Arguments

ev	List of explanatory variables provided to regress or logistic
cn	Column names for all explanatory variables in the dataset
intv	Interaction terms specified

Details

If ':' is used to select a range evar is updated. See <https://radiant-rstats.github.io/docs/model/regress.html> for an example in Radiant

Value

vars is a vector of right-hand side variables, possibly with interactions, iv is the list of explanatory variables, and intv are interaction terms

Examples

```
var_check("a:d", c("a","b","c","d"))
var_check(c("a", "b"), c("a", "b"), "a:c")
```

`write.coeff`*Write coefficient table for linear and logistic regression*

Description

Write coefficient table for linear and logistic regression

Usage

```
write.coeff(object, file = "", sort = FALSE, intercept = TRUE)
```

Arguments

<code>object</code>	A fitted model object of class <code>regress</code> or <code>logistic</code>
<code>file</code>	A character string naming a file. "" indicates output to the console
<code>sort</code>	Sort table by variable importance
<code>intercept</code>	Include the intercept in the output (TRUE or FALSE). TRUE is the default

Details

Write coefficients and importance scores to csv or return as a `data.frame`

Examples

```
regress(diamonds, rvar = "price", evar = c("carat", "clarity", "x"), check = "standardize") %>%
  write.coeff(sort = TRUE) %>%
  format_df(dec = 3)
```

Index

*Topic **datasets**

- catalog, 4
 - direct_marketing, 9
 - dvd, 11
 - houseprices, 14
 - ideal, 15
 - movie_contract, 17
 - ratings, 44
- ann (radiant.model-deprecated), 42
- auc, 3
- catalog, 4
- confint_robust, 5
- confusion, 5, 19, 20, 57
- copy_attr, 47
- crs, 6, 20, 54, 58
- crtree, 7, 21, 33, 58, 59
- direct_marketing, 9
- dtree, 9, 10, 22, 49, 59
- dtree_parser, 10
- dvd, 11
- evalbin, 4, 11, 23, 60
- evalreg, 12, 24, 61
- find_max, 13
- find_min, 14
- houseprices, 14
- ideal, 15
- logistic, 15, 25, 34, 62, 67
- minmax, 16
- movie_contract, 17
- nb, 17, 27, 35, 63
- nn, 18, 29, 36, 42, 64
- plot.confusion, 6, 19, 57
- plot.crs, 7, 20, 58
- plot.crtree, 8, 21, 59
- plot.dtree, 10, 22, 49, 59
- plot.evalbin, 4, 12, 23, 60
- plot.evalreg, 13, 24, 61
- plot.logistic, 16, 24, 25, 34, 62
- plot.model.predict, 16, 25, 26, 34, 62
- plot.nb, 18, 27, 63
- plot.nb.predict, 28
- plot.nn, 19, 29, 64
- plot.regress, 30, 37, 45, 65
- plot.repeater, 31, 46, 65
- plot.rpart, 21
- plot.simulater, 32, 51, 66
- predict.crtree, 8, 21, 33, 59
- predict.logistic, 16, 25, 26, 34, 62
- predict.nb, 18, 27, 28, 35, 63
- predict.nn, 19, 29, 36, 64
- predict.regress, 26, 31, 37, 45, 65
- predict_model, 38
- print.crtree.predict, 39
- print.logistic.predict, 39
- print.nb.predict, 40
- print.nn.predict, 40
- print.regress.predict, 41
- print_predict_model, 41
- radiant.model, 42
- radiant.model-deprecated, 42
- radiant.model-deprecated-package
(radiant.model-deprecated), 42
- radiant.model-package (radiant.model),
42
- radiant.model_viewer, 43
- radiant.model_window, 43
- ratings, 44
- regress, 30, 31, 37, 38, 44, 64, 65, 67
- render.DiagrammeR, 45
- repeater, 31, 46, 53, 65

scaledf, 47
sdw, 48
sensitivity, 48
sensitivity.dtree, 10, 22, 48, 49, 59
sim_cleaner, 51
sim_cor, 52
sim_splitter, 53
sim_summary, 53
simulator, 32, 50, 53, 59, 66
store.crs, 54
store.model, 54
store.model.predict, 55
store.nb.predict, 56
summary.confusion, 6, 20, 57
summary.crs, 7, 20, 57
summary.crtree, 8, 21, 33, 58
summary.dtree, 10, 22, 49, 59
summary.evalbin, 4, 12, 23, 60
summary.evalreg, 13, 24, 61
summary.logistic, 16, 34, 61
summary.nb, 18, 27, 35, 63
summary.nn, 19, 29, 36, 63
summary.regress, 31, 37, 45, 64
summary.repeater, 31, 46, 65
summary.simulator, 32, 51, 66

test_specs, 66

var_check, 67

write.coeff, 68