Package 'radiant.model'

February 16, 2018

Type Package

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

Version 0.9.0.9

Date 2018-2-16

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 (>= 3.3.0),
radiant.data (>= 0.9.0)

Imports radiant.basics (>= 0.9.0),
shiny (>= 1.0.5),
nnet (>= 7.3.12),
NeuralNetTools (>= 1.4.0),
sandwich (>= 2.3.4),
car (>= 2.1.3),
ggplot2 (>= 2.2.1),
gridExtra (>= 2.0.0),
data.tree (>= 0.7.3),
stringr (>= 1.1.0),
pryr (>= 0.1.2),
lubridate (>= 1.7.1),
tidyr (>= 0.7.2),
dplyr (>= 0.7.4),
rlang (>= 0.1.6),
magrittr (>= 1.5),
DiagrammeR (>= 0.9.2),
import (>= 1.1.0),
psych (>= 1.6.6),
e1071 (>= 1.6.8),
rpart (>= 4.1.10),
rstudioapi (>= 0.7),
yaml,
methods

Suggests testthat (>= 2.0.0)

URL https://github.com/radiant-rstats/radiant.model

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

License AGPL-3 | file LICENSE

LazyData true

RoxygenNote 6.0.1

R topics documented:

---

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(nrow(mtcars)), mtcars$vs, 1)
```

---

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")

cf *Movie ratings*

Description

Movie ratings

Usage

```
data(cf)
```

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(df,"description")

confint_robust *Confidence interval for robust estimators*

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` *Confusion matrix*

Description

Confusion matrix

Usage

```
confusion(dataset, pred, rvar, lev = "", cost = 1, margin = 2,
  train = "", 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
`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

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

crs *Collaborative Filtering*

Description

Collaborative Filtering

Usage

```
crs(dataset, id, prod, pred, rate, 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
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

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 name (string). This can be a dataframe in the global environment or an element in an r_data list from Radiant
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 _success_
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

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

direct_marketing *Direct marketing data*

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 *Create a decision tree*

Description

Create a decision tree

Usage

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

Arguments

yl	A yaml string or a list (e.g., from yaml::yaml.load_file())
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

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(yl)
```

Arguments

yl	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 *Data on DVD sales*

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` *Model evalbin*

Description

Model evalbin

Usage

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

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

```
result <- evalbin("titanic", c("age","fare"), "survived")
```

evalreg *Model evalreg*

Description

Model evalreg

Usage

```
evalreg(dataset, pred, rvar, train = "", 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
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., "price > 10000")

Details

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

find_max *Find maxium value of a vector*

Description

Find maxium value of a vector

Usage

```
find_max(var, val = "")
```

Arguments

var	Variable to find the maximum for
val	Variable to find the value for at the maxium of var

Value

Value of val at the maximum of var

---

find_min *Find minimum value of a vector*

Description

Find minimum value of a vector

Usage

```
find_min(var, val = "")
```

Arguments

var	Variable to find the minimum for
val	Variable to find the value for at the maxium of var

Value

Value of val at the minimum of var

---

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 *Ideal data for linear regression*

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 *Logistic regression*

Description

Logistic regression

Usage

```
logistic(dataset, rvar, evar, lev = "", int = "", wts = "None",
  check = "", ci_type, 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
rvar	The response variable in the model
evar	Explanatory variables in the model
lev	The level in the response variable defined as _success_
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 explicitely 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

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

---

minmax *Calculate min and max before standardization*

Description

Calculate min and max before standardization

Usage

```
minmax(dat)
```

Arguments

dat	Data frame

Value

Data frame min and max attributes

nb *Naive Bayes using e1071::naiveBayes*

Description

Naive Bayes using e1071::naiveBayes

Usage

```
nb(dataset, rvar, evar, laplace = 0, 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
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

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

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

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an r_data list from Radiant
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 _success_
size	Number of units (nodes) in the hidden layer
decay	Paramater decay
wts	Weights to use in estimation
seed	Random seed to use as the starting point
check	Optional estimation parameters ("standardize" is the default)
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/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

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

plot.confusion *Plot method for the confusion matrix*

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

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

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

x	Return value from crtree
plots	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
orient	Plot orientation for tree: LR for vertical and TD for horizontal
width	Plot width in pixels for tree (default is "900px")
labs	Use factor labels in plot (TRUE) or revert to default letters used by tree (FALSE)
dec	Decimal places to round results to
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 https://radiant-rstats.github.io/docs/model/crtree.html for an example in Radiant. The standard tree plot used by 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 <- crtree("titanic", "survived", c("pclass","sex"), lev = "Yes")
plot(result)
result <- crtree("diamonds", "price", c("carat","clarity", "cut"))
plot(result, plots = "prune")
result <- crtree("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

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 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 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

```
evalbin("titanic", "age", "survived") %>% plot
evalbin("titanic", c("age","fare"), "survived") %>% plot
evalbin("titanic", c("age","fare"), "survived") %>% summary
```

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("Rsq", "RMSE", "MAE"), ...)
```

Arguments

x	Return value from evalreg
vars	Measures to plot, i.e., one or more of "Rsq", "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

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,
  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. "dashboard" is a series of four plots used to visually evaluate model. "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
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 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 colour
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="pclass=levels(pclass), sex=levels(sex), age=seq(0,100,20)") %>%
  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, ...)
```

Arguments

x	Return value from nb
...	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 colour
...	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="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, shiny = FALSE,
  custom = FALSE, ...)
```

Arguments

x	Return value from nn
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
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 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 = c("olden","net"))
```

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, 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
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 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)
plot(result, plots = "dist")
## Not run:
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 *Plot repeated simulation*

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 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

---

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

```
result <- simulater(
  const = "cost 3",
  norm = "demand 2000 1000",
  discrete = "price 5 8 .3 .7",
  form = "profit = demand * (price - cost)"
)
plot(result, 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 = "", pred_cmd = "",
  conf_lev = 0.95, se = FALSE, dec = 3, ...)
```

Arguments

object	Return value from crtree
pred_data	Provide the name of a 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 use a ',' (e.g., '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 = "", pred_cmd = "",
  conf_lev = 0.95, se = TRUE, dec = 3, ...)
```

Arguments

object	Return value from logistic
pred_data	Provide the name of a 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 use a ',' (e.g., '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/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 *Predict method for the nb function*

Description

Predict method for the nb function

Usage

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

Arguments

object	Return value from nb
pred_data	Provide the name of a 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 use a ',' (e.g., '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(result, pred_data = "titanic", pred_names = NA)
```

---

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 = "", pred_cmd = "",
  conf_lev = 0.95, se = FALSE, dec = 3, ...)
```

Arguments

object	Return value from nn
pred_data	Provide the name of a 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 use a ',' (e.g., '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/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 = "", pred_cmd = "",
  conf_lev = 0.95, se = TRUE, dec = 3, ...)
```

Arguments

object	Return value from regress
pred_data	Name of the dataset to use for prediction
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)
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 = c("carat:clarity"))
dpred <<- getdata("diamonds") %>% slice(1:10)
predict(result, pred_data = "dpred")
rm(dpred, envir = .GlobalEnv)
```

---

predict_model *Predict method for model functions*

Description

Predict method for model functions

Usage

```
predict_model(object, pfun, mclass, pred_data = "", 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	Name of the dataset to use for prediction
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)
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

radiant.model *radiant.model*

Description

radiant.model

Launch radiant.model in 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)
```

radiant.model-deprecated *Deprecated function(s) in the radiant.model package*

Description

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

Usage

```
regression(...)
```

Arguments

...	Parameters to be passed to the updated functions

Details

regression is now a synonym for regress
glm_reg is now a synonym for logistic
performance is now a synonym for evalbin
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)
```

regress *Linear regression using OLS*

Description

Linear regression using OLS

Usage

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

```
result <- regress("diamonds", "price", c("carat","clarity"))
result <- regress("diamonds", "price", c("carat","clarity"), check = "standardize")
```

---

render.DiagrammeR *Method to render DiagrammeR plots*

Description

Method to render DiagrammeR plots

Usage

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

Arguments

object	DiagrammeR plot
...	Additional arguments

repeater *Repeated simulation*

Description

Repeated simulation

Usage

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

Arguments

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	Name for data.frame with the repeated simulation data (optional)
sim	Return value from the simulater function (data.frame or data.frame name)

Examples

```
result <- simulater(
  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
)

repeater(
  nr = 12,
  vars = c("E","price"),
  sum_vars = "profit",
  byvar = "sim",
  form = "profit_365 = profit < 36500",
  seed = 1234,
  sim = result
) %>% head
```

scaledf *Center or standardize variables in a data frame*

Description

Center or standardize variables in a data frame

Usage

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

Arguments

dat	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 traning 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

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 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.
...	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

`simulater` *Simulate data for decision analysis*

Description

Simulate data for decision analysis

Usage

```
simulater(const = "", lnorm = "", norm = "", unif = "", discrete = "",
  binom = "", sequ = "", grid = "", data = "", form = "",
  seed = NULL, nexact = FALSE, ncorr = NULL, name = "", nr = 1000,
  dat = 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)
`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`	Name of a 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	Name used to store the simulated data (optional)
nr	Number of simulations
dat	Data list from previous simulation. Used by repeater function

Details

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

Value

A data.frame with the simulated data

See Also

summary.simulater to summarize results

plot.simulater to plot results

Examples

```
result <- simulater(
  const = "cost 3",
  norm = "demand 2000 1000",
  discrete = "price 5 8 .3 .7",
  form = "profit = demand * (price - cost)",
)
```

---

sim_cleaner *Clean input command string*

Description

Clean input command string

Usage

```
sim_cleaner(x)
```

Arguments

x	Input string

Value

Cleaned string

sim_cor *Simulate correlated normally distributed data*

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

sim_splitter *Split input command string*

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 *Print simulation summary*

Description

Print simulation summary

Usage

```
sim_summary(dat, dc = getclass(dat), fun = "", dec = 4)
```

Arguments

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

store.crs *Store predicted values generated in the crs function*

Description

Store predicted values generated in the crs function

Usage

```
## S3 method for class 'crs'
store(object, name = "predict_cf", envir = parent.frame(),
  ...)
```

Arguments

object	Return value from crs
name	Name of the dataset to store
envir	Environment to assign 'new' dataset (optional). Used when an r_data list is not available
...	further arguments passed to or from other methods

Details

Store data frame with predictions in Radiant r_data list if available. 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(object, ..., name = "residuals")
```

Arguments

object	Return value from a model function
...	Additional arguments
name	Variable name(s) assigned to predicted values

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")) %>%
  store %>% 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(object, ..., data = attr(object, "pred_data"),
  name = "prediction")
```

Arguments

object	Return value from model function
...	Additional arguments
data	Data or dataset name (e.g., data = mtcars or data = "mtcars")
name	Variable name(s) assigned to predicted values

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(diamonds) %>%
  store(name = "pred, pred_low, pred_high") %>% head
```

---

store.nb.predict *Store predicted values generated in the nb function*

Description

Store predicted values generated in the nb function

Usage

```
## S3 method for class 'nb.predict'
store(object, ..., data = attr(object, "pred_data"),
  name = "")
```

Arguments

object	Return value from model function
...	Additional arguments
data	Data or dataset name (e.g., data = mtcars or data = "mtcars")
name	Variable name(s) assigned to predicted values. If empty, the levels of the response variable will be used

Details

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

Examples

```
result <- nb("titanic", "survived", c("pclass","sex","age"))
pred <- predict(result, pred_data = "titanic")
store(pred, data = titanic, name = "pred") %>% head
store(pred, data = titanic) %>% head
```

store_glm — *Deprecated function to store logistic regression residuals and predictions*

Description

Deprecated function to store logistic regression residuals and predictions

Usage

```
store_glm(object, data = object$dataset, type = "residuals",
  name = paste0(type, "_logit"))
```

Arguments

object	Return value from logistic or predict.logistic
data	Dataset name
type	Residuals ("residuals") or predictions ("predictions"). For predictions the dataset name must be provided
name	Variable name assigned to the residuals or predicted values

Details

Use store.model.predict or store.model instead

store_reg — *Deprecated function to store regression residuals and predictions*

Description

Deprecated function to store regression residuals and predictions

Usage

```
store_reg(object, data = object$dataset, type = "residuals",
  name = paste0(type, "_reg"))
```

Arguments

object	Return value from regress or predict.regress
data	Dataset name
type	Residuals ("residuals") or predictions ("predictions"). For predictions the dataset name must be provided
name	Variable name assigned to the residuals or predicted values

Details

Use store.model.predict or store.model instead

summary.confusion *Summary method for the confusion matrix*

Description

Summary method for the confusion matrix

Usage

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

Arguments

object	Return value from confusion
...	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

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

---

summary.crtree *Summary method for the crtree function*

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, ...)
```

Arguments

object	Return value from simulater
input	Print decision tree input
output	Print decision tree output
...	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

summary.evalbin *Summary method for the evalbin function*

Description

Summary method for the evalbin function

Usage

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

Arguments

object	Return value from evalbin
prn	Print full table of measures per model and bin
...	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

```
evalbin("titanic", "age", "survived") %>% summary
evalbin("titanic", c("age","fare"), "survived") %>% summary
```

---

summary.evalreg *Summary method for the evalreg function*

Description

Summary method for the evalreg function

Usage

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

Arguments

object	Return value from evalreg
...	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

summary.logistic *Summary method for the logistic function*

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 logistic
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 Radiant

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 *Summary method for the nb function*

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 *Summary method for the nn function*

Description

Summary method for the nn function

Usage

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

Arguments

object	Return value from nn
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 *Summary method for the regress function*

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 regress
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 *Summarize repeated simulation*

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

---

summary.simulater *Summary method for the simulater function*

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

```
result <- simulater(norm = "demand 2000 1000")
summary(result)
```

---

test_specs *Add interaction terms to list of test variables if needed*

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 *Check if main effects for all interaction effects are included in the model If ':' is used to select a range _evar_ is updated*

Description

Check if main effects for all interaction effects are included in the model If ':' is used to select a range _evar_ is updated

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 _dat_
intv	Interaction terms specified

Details

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

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

Details

Write coefficients and importance scores to csv

Examples

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

Index