# Package 'radiant.model'

June 27, 2016

**Type** Package

**Title** Model estimation and evaluation menu for Radiant. Builds on the radiant.data package

**Version** 0.5.1

**Date** 2016-6-27

**Description** Modeling menu for Radiant.

**Depends** R (>= 3.2.0),
    radiant.data (>= 0.5.1)

**Imports** radiant.basics (>= 0.5.1),
    shiny (>= 0.13.2.9003),
    nnet (>= 7.3.11),
    NeuralNetTools (>= 1.4.0),
    sandwich (>= 2.3.4),
    car (>= 2.1.1),
    broom (>= 0.4.0),
    ggplot2 (>= 2.0.0),
    gridExtra (>= 2.0.0),
    data.tree (>= 0.1.9),
    yaml (>= 2.1.13),
    stringr (>= 1.0),
    pryr (>= 0.1.2),
    lubridate (>= 1.5.0),
    tidyr (>= 0.4.1),
    dplyr (>= 0.5),
    magrittr (>= 1.5),
    DiagrammeR(>= 0.8.3),
    import (>= 1.1.0),
    methods

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

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

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

**License** AGPL-3 | file LICENSE

**LazyData** true

**RoxygenNote** 5.0.1

# R **topics documented:**

| ann | *Artificial Neural Networks* |
|---|---|

## Description

Artificial Neural Networks

## Usage

```
ann(dataset, rvar, evar, type = "classification", lev = "", size = 1,
  decay = 0.5, wts = "None", seed = NA, check = "", dec = 3,
  data_filter = "")
```

## Arguments

| | |
|---|---|
| dataset | Dataset name (string). This can be a dataframe in the global environment or an element in an r_data list from Radiant |
| rvar | The response variable in the logit (probit) 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 output or estimation parameters. "vif" to show the multicollinearity diagnostics. "confint" to show coefficient confidence interval estimates. "odds" to show odds ratios and confidence interval estimates. "standardize" to output standardized coefficient estimates. "stepwise" to apply step-wise selection of variables |
| dec | Number of decimals to show |
| data_filter | Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000") |

**Details**

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

**Value**

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

**See Also**

[summary.ann](summary.ann) to summarize results

[plot.ann](plot.ann) to plot results

[predict.ann](predict.ann) for prediction

**Examples**

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

---

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

**Value**

AUC statistic

**See Also**

[evalbin](evalbin) to calculate results

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

[plot.evalbin](plot.evalbin) to plot results

## Examples

```
auc(mtcars$mpg, mtcars$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")

---

| confint_robust | *Confidence interval for robust estimators* |
|---|---|

---

## Description

Confidence interval for robust estimators

## Usage

```
confint_robust(object, parm, level = 0.95, vcov = NULL, ...)
```

## Arguments

| | |
|---|---|
| object | A fitted model object |
| parm | A specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered |
| level | The confidence level required |
| vcov | Covariance matrix generated by, e.g., sandwich::vcovHC |
| ... | Additional argument(s) for methods |

## Details

Wrapper for confint.default with robust standard errors. See [http://stackoverflow.com/a/3820125/1974918](http://stackoverflow.com/a/3820125/1974918)

---

confusion *Confusion matrix*

---

### Description

Confusion matrix

### Usage

```
confusion(dataset, pred, rvar, lev = "", margin = 1, cost = 1,
  train = "", method = "xtile", 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_ |
| margin | Margin on each customer purchase |
| cost | Cost for each connection (e.g., email or mailing) |
| train | Use data from training ("Training"), validation ("Validation"), both ("Both"), or all data ("All") to evaluate model evalbin |
| method | Use either ntile or xtile to split the data (default is xtile) |
| 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 <http://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")
```

---

crs                        *Collaborative Filtering*

---

### Description

Collaborative Filtering

### Usage

```
crs(dataset, id, prod, pred, rate, name = "pred", 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 |
| name | Name for the prediction variable |
| data_filter | Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000") |

### Details

See <http://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

---

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")
```

#### 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 |

#### Details

See http://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

| 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 http://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, margin = 1, cost = 1,
  train = ", method = "xtile", 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 |
| margin | Margin on each customer purchase |
| cost | Cost for each connection (e.g., email or mailing) |
| train | Use data from training ("Training"), validation ("Validation"), both ("Both"), or all data ("All") to evaluate model evalbin |
| method | Use either ntile or xtile to split the data (default is xtile) |
| data_filter | Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000") |

**Details**

See <http://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 <http://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                          *Generalized linear models (GLM)*

---

### Description

Generalized linear models (GLM)

### Usage

```
logistic(dataset, rvar, evar, lev = "", int = "", wts = "None",
  check = "", dec = 3, data_filter = "")
```

### Arguments

| | |
|---|---|
| dataset | Dataset name (string). This can be a dataframe in the global environment or an element in an r_data list from Radiant |
| rvar | The response variable in the logit (probit) 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 | Optional estimation parameters. "standardize" to output standardized coefficient estimates. "stepwise" to apply step-wise selection of variables |
| dec | Number of decimals to show |
| data_filter | Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000") |

### Details

See <http://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"))
```

---

plot.ann                    *Plot method for the ann function*

---

### Description

Plot method for the ann function

### Usage

```
## S3 method for class 'ann'
plot(x, shiny = FALSE, ...)
```

### Arguments

| | |
|---|---|
| x | Return value from [ann](#) |
| shiny | Did the function call originate inside a shiny app |
| ... | further arguments passed to or from other methods |

### Details

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

### See Also

[ann](#) to generate results

[summary.ann](#) to summarize results

[predict.ann](#) for prediction

### Examples

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

---

plot.confusion              *Plot method for the confusion matrix*

---

### Description

Plot method for the confusion matrix

### Usage

```
## S3 method for class 'confusion'
plot(x, scale_y = FALSE, shiny = FALSE, ...)
```

## Arguments

| | |
|---|---|
| x | Return value from [evalreg](#) |
| scale_y | Free scale in faceted plot of the confusion matrix (TRUE or FALSE) |
| shiny | Did the function call originate inside a shiny app |
| ... | further arguments passed to or from other methods |

## Details

See <http://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.crs | *Plot method for the crs function* |
|---|---|

---

## Description

Plot method for the crs function

## Usage

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

## Arguments

| | |
|---|---|
| x | Return value from [crs](#) |
| shiny | Did the function call originate inside a shiny app |
| ... | further arguments passed to or from other methods |

## Details

See <http://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.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,
  shiny = FALSE, ...)
```

### Arguments

| | |
|---|---|
| x | Return value from [dtree](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 |
| shiny | Did the function call originate inside a shiny app |
| ... | further arguments passed to or from other methods |

### Details

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

### See Also

[dtree](dtree) to generate the result

[summary.dtree](summary.dtree) to summarize 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"), shiny = FALSE, ...)
```

### Arguments

| | |
|---|---|
| x | Return value from [evalbin](evalbin) |
| plots | Plots to return |
| shiny | Did the function call originate inside a shiny app |
| ... | further arguments passed to or from other methods |

## Details

See <http://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", method = "xtile") %>% 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, shiny = FALSE, ...)
```

## Arguments

| | |
|---|---|
| x | Return value from [evalreg](#) |
| shiny | Did the function call originate inside a shiny app |
| ... | further arguments passed to or from other methods |

## Details

See <http://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). "hist" shows histograms 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 http://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.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). "hist" to show histograms 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 http://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 = "dashboard", lines = c("line","loess"))
plot(result, plots = "coef", conf_lev = .99, intercept = TRUE)
plot(result, plots = "hist")
plot(result, plots = "scatter", lines = c("line","loess"))
plot(result, plots = "correlations")
plot(result, plots = "resid_pred", lines = "line")
```

---

plot.repeater                *Plot repeated simulation*

---

## Description

Plot repeated simulation

## Usage

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

## Arguments

| x | Return value from [repeater](repeater) |
| shiny | Did the function call originate inside a shiny app |
| ... | 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, shiny = FALSE, ...)
```

## Arguments

| x | Return value from [simulater](simulater) |
| shiny | Did the function call originate inside a shiny app |
| ... | further arguments passed to or from other methods |

## Details

See <http://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)
```

---

| predict.ann | *Predict method for the ann function* |

---

## Description

Predict method for the ann function

## Usage

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

## Arguments

| | |
|---|---|
| object | Return value from [ann](#) |
| 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) |
| ... | further arguments passed to or from other methods |

## Details

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

## See Also

[ann](#) to generate the result

[summary.ann](#) to summarize results

## 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.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 = FALSE, ...)
```

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

## Details

See <http://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.model | *Predict method for model functions* |
|---|---|

## Description

Predict method for model functions

## Usage

```
## S3 method for class 'model'
predict(object, pfun, mclass, pred_data = "", pred_cmd = "",
  conf_lev = 0.95, se = FALSE, ...)
```

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

## Details

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

| 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, ...)
```

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

### Details

See <http://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)
```

---

print.ann.predict              *Print method for predict.ann*

---

### Description

Print method for predict.ann

### Usage

```
## S3 method for class 'ann.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.model.predict | *Print method for the model prediction* |
|---|---|

#### Description

Print method for the model prediction

#### Usage

```
## S3 method for class 'model.predict'
print(x, ..., n = 10, header = "", lev = "")
```

#### 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 |
| lev | The level in the response variable defined as _success_ for classification models |

| 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 |

---

radiant.model                *radiant.model*

---

## Description

radiant.model

Launch Radiant in the default browser

## Usage

```
radiant.model()
```

## Details

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

---

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](regress)
glm_reg is now a synonym for [logistic](logistic)
performance is now a synonym for [evalbin](evalbin)

---

regress                *Linear regression using OLS*

---

## Description

Linear regression using OLS

## Usage

```
regress(dataset, rvar, evar, int = "", check = "", dec = 3,
  data_filter = "")
```

## Arguments

| | |
|---|---|
| dataset | Dataset name (string). This can be a dataframe in the global environment or an element in an r_data list from Radiant |
| rvar | The response variable in the regression |
| evar | Explanatory variables in the regression |
| int | Interaction terms to include in the model |
| check | "standardize" to see standardized coefficient estimates. "stepwise" to apply stepwise selection of variables in estimation |
| dec | Number of decimals to show |
| data_filter | Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000") |

## Details

See <http://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                                    *Repeat simulation*

---

**Description**

Repeat simulation

**Usage**

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

**Arguments**

| | |
|---|---|
| nr | Number times to repeat the simulation |
| vars | Variables to use in repeated simulation |
| grid | Expression 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 string with the formula to apply to the summarized data |
| seed | To repeat a simulation with the same randomly generated values enter a number into Random seed input box. |
| name | To save the simulated data for further analysis specify a name in the Sim name input box. You can then investigate the simulated data by choosing the specified name from the Datasets dropdown in any of the other Data tabs. |
| sim | Return value from the simulater function |

**Examples**

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

repeater(sim = result)
```

---

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, ...)
```

### Arguments

| | |
|---|---|
| object | Return value from [dtree](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 |
| ... | Additional arguments |

### Details

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

---

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 = "",
  name = "", nr = 1000, dat = NULL)
```

### Arguments

| | |
|---|---|
| const | A string listing the constants to include in the analysis (e.g., "cost = 3; size = 4") |
| lnorm | A string 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 string 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 string 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 string 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 string listing the random variables with a binomail 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 string 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 string 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 string with the formula to evaluate (e.g., "profit = demand * (price - cost)") |
| seed | To repeat a simulation with the same randomly generated values enter a number into Random seed input box. |
| name | To save the simulated data for further analysis specify a name in the Sim name input box. You can then investigate the simulated data by choosing the specified name from the Datasets dropdown in any of the other Data tabs. |
| nr | Number of simulations |
| dat | Data list from previous simulation. Used by repeater function |

## Details

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

## Value

A data.frame with the created variables

## See Also

[summary.simulater](summary.simulater) to summarize results

[plot.simulater](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_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.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 <http://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 <http://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 %>% head
```

---

store_ann     *Deprecated function to store predictions from an ANN*

---

### Description

Deprecated function to store predictions from an ANN

### Usage

```
store_ann(object, data = object$dataset, name = paste0("predict_ann"))
```

### Arguments

| | |
|---|---|
| object | Return value from predict.ann |
| data | Dataset name |
| name | Variable name assigned to the residuals or predicted values |

### Details

Use store.model.predict or store.model instead

---

store_crs *Store predicted values generated in the crs function*

---

### Description

Store predicted values generated in the crs function

### Usage

```
store_crs(pred, data, name = "pred_crs")
```

### Arguments

| | |
|---|---|
| pred | Return value from predict.nnet |
| data | Dataset name |
| name | Variable name assigned to the predicted values |

### Details

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

---

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.ann                    *Summary method for the ann function*

---

## Description

Summary method for the ann function

## Usage

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

## Arguments

| | |
|---|---|
| object | Return value from [ann](#) |
| ... | further arguments passed to or from other methods |

## Details

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

## See Also

[ann](#) to generate esults

[plot.ann](#) to plot results

[predict.ann](#) for prediction

### Examples

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

---

summary.crs                    *Summary method for Collaborative Filter*

---

### Description

Summary method for Collaborative Filter

### Usage

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

### Arguments

| | |
|---|---|
| object | Return value from [crs](#) |
| ... | further arguments passed to or from other methods |

### Details

See <http://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.dtree                  *Summary method for the dtree function*

---

### Description

Summary method for the dtree function

### Usage

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

### Arguments

| | |
|---|---|
| object | Return value from [simulater](#) |
| ... | further arguments passed to or from other methods |

**Details**

See <http://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

---

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 model evalbin results (default is TRUE) |
| ... | further arguments passed to or from other methods |

**Details**

See <http://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 <http://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 = "", ...)
```

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

**Details**

See <http://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.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 = "", ...)
```

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

**Details**

See <http://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 <http://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 <http://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 <http://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")
```

# Index