Package ‘math3190package’

January 2, 2026

Title Package for MATH 3190 - Fundamentals of Data Science at SUU

Version 0.0.0.9000

Description This package contains functions and data files needed for MATH 3190 - Fundamentals of Data Science taught at Southern Utah University.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.2),
tidyverse

Imports boot,
broom,
caret,
dslabs,
GGally,
patchwork,
nnet,
e1071,
naivebayes,
MASS,
glmnet,
glmnetUtils,
faraway,
car,
pls,
umap,
ggfortify,
tsibble,
ggtime,
tseries,
forecast,
reticulate,
DBI,
RMySQL

LazyData true

Author Rick Brown [aut, cre]

Maintainer Rick Brown `<richardbrown1@suu.edu>`

Contents

`baseball` *baseball Data Set*

Description

This data set contains information for 337 baseball players.

Usage

```
baseball
```

Format

A tibble with 337 rows and 28 variables:

salary The salary in $1000s.

average Batting average of the player.

obp On base percentage of the player

runs Number of runs scored.

hits Number of hits in total.

doubles Number of doubles hit.

triples Number of triples hit.

homeruns Number of homeruns hit.

rbis Number of runs batted in.

walks Number of times walked.

sos Number of strikeouts.

sbs Number of stolen bases.

errors Number of errors committed.

freeagent Factor indicating whether the player is a free agent or is eligible for free agency.

arbitration Factor indicating whether the player has arbitration or is eligible for arbitration.

runsperso Number of runs per strikeout (runs/sos).

hitsperso Number of hits per strikeout (hits/sos).

hrsperso Number of homeruns per strikeout (homeruns/sos).

rbisperso Number of rbis per strikeout (rbis/sos).

walksperso Number of walks per strikeout (walks/sos).

obppererror On base percentage per error (obp/errors).

runspererror Number of runs scored per error (runs/errors).

hitspererror Number of hits per error (hits/errors).

hrspererror Number of homeruns per error (homeruns/errors).

sospererror Number of strikeouts per error (sos/errors).

sbsobp Number of stolen bases times on base percentage (sbs*obp). \itemsbsrunsNumber of stolen bases times number of runs scored (sbsruns).*

sbshits Number of stolen bases times number of hits (sbs*hits).

`births` *births Data Set*

Description

Data from 1995-1997 for a study hat examined pregnancies that resulted in the birth of twins. Births were classified as preterm with intervention (induced labor or cesarean), preterm without procedures, or term/post-term. Researchers also classified the pregnancies by the level of prenatal medical care the mother received (inadequate, adequate, or intensive). The data set consists of 278 cases (rows) with two columns indicating the level of prenatal care and type of birth for each set of twins.

Usage

```
births
```

Format

A tibble with 278 rows and 2 variables:

prenatal A factor indicating the prenatal care the mother received: Adequate, Inadequate, or Intensive.

type A factor indicating the classification of the birth: "Preterm (induced or cesarean)", "Preterm (without procedures)", and "Term or post-term"

bootstrapping — *Outputs and restyles an object from the* `boot()` *function from the boot package.*

Description

This function relies on the boot function. It generates R bootstrap replicates of a statistic applied to data. Both parametric and nonparametric resampling are possible. For the nonparametric bootstrap, possible resampling methods are the ordinary bootstrap, the balanced bootstrap, antithetic resampling, and permutation. For nonparametric multi-sample problems stratified resampling is used: this is specified by including a vector of strata in the call to boot. Importance resampling weights may be specified.

Usage

```
bootstrapping(data, statistic, R, ...)
```

Arguments

`data`	The data as a vector, matrix or data frame. If it is a matrix or data frame then each row is considered as one multivariate observation.
`statistic`	A function which when applied to data returns a vector containing the statistic(s) of interest. When sim = "parametric", the first argument to statistic must be the data. For each replicate a simulated dataset returned by ran.gen will be passed. In all other cases statistic must take at least two arguments. The first argument passed will always be the original data. The second will be a vector of indices, frequencies or weights which define the bootstrap sample. Further, if predictions are required, then a third argument is required which would be a vector of the random indices used to generate the bootstrap predictions. Any further arguments can be passed to statistic through the ... argument.
`R`	The number of bootstrap replicates. Usually this will be a single positive integer. For importance resampling, some resamples may use one set of weights and others use a different set of weights. In this case R would be a vector of integers where each component gives the number of resamples from each of the rows of weights.
`...`	See more documentation in the boot function.

Value

This function returns an object of class boot3150 that works much like an object of class boot from the `boot` package.

Examples

```
med <- function(x, i){median(x[i])}
sample_data <- rnorm(100)
boot_out <- bootstrapping(sample_data, med, 1000)
boot_out
```

`boot_ci` *Outputs and restyles an object from the* `boot.ci()` *function from the boot package.*

Description

This function relies on the boot.ci function. This function generates 5 different types of equi-tailed two-sided nonparametric confidence intervals. These are the first order normal approximation, the basic bootstrap interval, the studentized bootstrap interval, the bootstrap percentile interval, and the adjusted bootstrap percentile (BCa) interval. All or a subset of these intervals can be generated.

Usage

```
boot_ci(boot_out, conf = 0.95, type = "all", ...)
```

Arguments

`boot_out`	An object of class "boot" or "boot3150" containing the output of a bootstrap calculation.
`conf`	A scalar or vector containing the confidence level(s) of the required interval(s).
`type`	A vector of character strings representing the type of intervals required. The value should be any subset of the values c("norm","basic", "stud", "perc", "bca") or simply "all" which will compute all five types of intervals.
`...`	See more documentation in the boot.ci function.

Value

This function returns an object of class boot3150ci that works much like an object of class bootci from the `boot` package.

Examples

```
med <- function(x, i){median(x[i])}
sample_data <- rnorm(100)
boot_out <- bootstrapping(sample_data, med, 1000)
bootci_out <- boot_ci(boot_out)
```

`cars99` *cars99 Data Set*

Description

This data set contains information on 109 vechicles from 1999.

Usage

```
cars99
```

Format

A tibble with 109 rows and 11 variables:

Model The vechicle model name.

CityMPG The miles per gallon (MPG) for the vehicle in the city.

HwyMPG The miles per gallon (MPG) for the vehicle on the highway

FuelCap The fuel capacity of the vehicle in gallons.

Weight The weight of the vehicle in lbs.

FrontWt The front weight of the vehicle.

Accel0_30 The time it takes, in seconds, for the vehicle to accelerate from 0 to 30 mph.

Accel0_60 The time it takes, in seconds, for the vehicle to accelerate from 0 to 60 mph.

QtrMile The time it takes, in seconds, for the vehicle to travel a quarter of a mile.

`classifier_plot` *Plots the Points and Predicted Region for a Classifier Model*

Description

This function plots the points and predicted region for a classifier model of class "naive_bayes", "lda", or "svm". This function can support up to three predictor variables: two of which can be quantitative and one can be categorical.

Usage

```
classifier_plot(classifier_model, data, res = 200, sv_diamonds = TRUE)
```

Arguments

`classifier_model`	The classifier model of class "naive_bayes", "lda", or "svm".
`data`	The dataset that should be plotted. This need not be the dataset on which the classifier was trained.
`res`	The resolution of the plot. Larger values will make the lines look straighter and make the grid more fine, but take longer.
`sv_diamonds`	A boolean indicating whether the support vectors should be plotted with diamonds around them when plotting a support vector machine classifier.

Value

This function returns a ggplot object plot. It can be added onto like any ggplot object.

Examples

```
data(mtcars)
mtcars2 <- mtcars |> mutate(vs = factor(vs), am = factor(am))
classifier <- MASS::lda(vs ~ mpg + disp + am, data = mtcars2)
classifier_plot(classifier, data = mtcars2)
plot_tukey(t)
```

`class_data_f2019` *class_data_f2019 Data Set*

Description

This data set contains information on 42 students from Fall of 2019.

Usage

```
class_data_f2019
```

Format

A tibble with 42 rows and 7 variables:

level A factor indicating the class level the student is: Freshman, Sophomore, Junior, Senior, or Graduate.

major A character indicating the major of the student.

sex A factor indicating the sex of the student: F for female or M for male.

ski A factor indicating downhill preference: Ski, Snowboard, or Neither.

penny A factor indicating preference regarding the penny: Abolish, Retain or No Answer.

speed An integer indicating the fastest speed the student had driven in a vehicle (in mph).

sleep A numeric variable indicating how long the student slept the night before.

`diseases` *diseases Data Set*

Description

This data set contains information for each state and Washington, D.C. about the number of reported cases of AIDS, syphilis, and tuberculosis.

Usage

```
diseases
```

Format

A tibble with 51 rows and 4 variables:

State A charcter vector indicating the state.

AIDS The number of reporeted AIDS cases.

Syphilis The number of reporeted syphilis cases.

Tuberculosis The number of reporeted tuberculosis cases.

`epilepsy` *epilepsy Data Set*

Description

This data set contains information on the number of seizures, which treatment, and the age of 59 patients with epilepsy.

Usage

```
epilepsy
```

Format

A tibble with 59 rows and 4 variables:

id The patient ID.

numseiz The number of seizures the patient had.

age The age of the patient

`idealwt` *idealwt Data Set*

Description

This data set contains weight information on 182 people (119 females and 63 males). Actual weights, ideal weights, and the difference between them are recorded.

Usage

```
idealwt
```

Format

A tibble with 182 rows and 4 variables:

sex A factor indicating the sex of the person: Female or Male.

actual The person's actual weight.

ideal The person's ideal weight.

diff The difference between the person's actual weight and their ideal weight. Negative values indicate that the person weighs less than what they consider ideal.

ilogit *Compute the Inverse Logit of a Value*

Description

A function that computes the inverse logit (the inverse log odds) of a value.

Usage

```
ilogit(x)
```

Arguments

`x`	A value for which that we want to find the inverse.

Examples

```
data(mtcars)
mod <- glm(vs ~ mpg, data = mtcars, family = binomial())
ilogit(predict(mod, data.frame(mpg = 20)))
```

influence_plots *Creates many diagnostic plots*

Description

This function creates many diagnostic plots for a given model. These plots include residual plots, a QQ plot, a leverage plot, a Cook's distance plot, a DfFits plot, and DfBetas plots for the intercept and all slopes.

Usage

```
influence_plots(model, missing_group = NULL)
```

Arguments

`model`	The model for which we would like these plots. This can be of class "lm" or "glm" with a binomial family.
`missing_group`	Used for multinomial regression to indicate which group is not being plotted.

Value

This function returns plots. The user must press "enter" or "return" to view subsequent plots.

Examples

```
mod <- lm(mpg ~ disp, data = mtcars)
influence_plots(mod)
```

interaction_plot *Two-way Interaction Plot with ggplot2*

Description

This function is similar to interaction.plot, but plots in ggplot2 instead of base R. It plots the mean (or other summary) of the response for two-way combinations of factors, thereby illustrating possible interactions.

Usage

```
interaction_plot(
  df,
  response,
  x_factor,
  group_factor,
  linewidth = 1,
  point_size = 2,
  .f = mean,
  ...
)
```

Arguments

`df`	The data frame that contains the response and factor variables.
`response`	The name of the numeric variable giving the response. Can be entered with or without quotes.
`x_factor`	The name of a factor whose levels will form the x axis. Can be entered with or without quotes.
`group_factor`	The name of a factor whose levels will split up the other variables. Can be entered with or without quotes.
`linewidth`	The linewidth of the lines plotted. Enter 0 if the lines should be omitted.
`point_size`	The size of the points plotted. Enter 0 if the points should be omitted.
`...`	Other parameters that can be passed to geom_point.

Value

This function returns a two-way interaction plot as a ggplot object. All the ggplot add-on functions can be added onto the object returned for more customization.

Examples

```
data(warpbreaks)
interaction_plot(warpbreaks, breaks, wool, tension)
interaction_plot(warpbreaks, "breaks", "wool", "tension")
```

`logistic_plots` *Creates many diagnostic plots for logistic or multinomial models*

Description

This function feeds into the `influence_plots()` function to create many diagnostic plots for a given logistic or multinomial model. These plots include residual plots, a leverage plot, a Cook's distance plot, a DfFits plot, and DfBetas plots for the intercept and all slopes.

Usage

```
logistic_plots(model)
```

Arguments

`model`	The model for which we would like these plots. This can be of class "glm" with a binomial family or of class "multinom" from the `nnet` library.
`missing_group`	Used for multinomial regression to indicate which group is not being plotted.

Value

This function returns plots. The user must press "enter" or "return" to view subsequent plots.

Examples

```
mod <- glm(vs ~ disp, data = mtcars, family = "binomial")
logistic_plots(mod)
```

`mlbsalaries` *mlbsalaries Data Set*

Description

This data set contains information on 877 MLB players from 2018.

Usage

```
mlbsalaries
```

Format

A tibble with 877 rows and 5 variables:

Rank The ranking of the player's salary with 1 being the most money earned.

Name A character vector containing the name of the player.

Team A character vector containing the team the player played for.

Position A factor indicating what position the player played: SP for starting pitcher, 1B for first base, 2B for second base, 3B for third base, SS for shortstop, OF for outfield, RP for relief pitcher, and DH for designated hitter.

Salary A numeric vector containing the salary the player made for the 2018 season.

`multinom_summary`	*Outputs estimates, standard errors, test statistic, and p-values for a multinomial logistic regression object from the nnet library.*

Description

When given an nnet multinom object, this function will compute the test statistic and p-value for each term in the models and return a tibble with the coefficients, standard errors, test statistic, and p-values in a list with one list element per model.

Usage

```
multinom_summary(model)
```

Arguments

`model`	An object of type "multinom" "nnet"

Value

This function returns a list of tibbles with the coefficient estimates, the standard errors, the test statistics, and the p-values for the models comparing each category to the baseline. The AIC and deviance are also given as the last elements in the list.

Examples

```
mod <- mtcars |>
  mutate(carb = factor(carb)) |>
  multinom(carb ~ mpg + disp + hp, data = _)
multinom_summary(mod)
```

`nitrogen`	*nitrogen Data Set*

Description

Nitrogen content of trees in an orchard, the growing tips of 150 leaves are clipped from trees throughout the orchard. These leaves are ground to form one composite sample, which the researcher assays for percentage of nitrogen. Composite samples obtained from a random sample of 36 orchards throughout the state gave the nitrogen contents.

Usage

```
nitrogen
```

Format

A data frame with 36 rows 1 variable:

nitrogen The nitrogen content for the trees.

`plot.boot3150` *Plots a boot3150 object*

Description

When given a boot3150 object, this function plots the samples with both histograms and QQ plots.

Usage

```
## S3 method for class 'boot3150'
plot(x)
```

Arguments

`x`	An object of class boot3150

Value

This function plots the boot3150 object.

`plot_tukey` *Plots the Tukey HSD Confidence Intervals*

Description

This function plots the Tukey HSD confidence intervals when given a tibble outputted by the tukeyhsd function. If more than one variable, all will be plotted by default, but that can be adjusted with the `which` option.

Usage

```
plot_tukey(t, which = "all")
```

Arguments

`t`	The Tukey HSD tibble outputted by the tukeyhsd function.
`which`	Indicates which variable for which the graph should be created. The default is "all", but a variable name used in the model can also be entered (in quotes).

Value

This function returns plots. The user must press "enter" or "return" to view subsequent plots if there are any.

Examples

```
data(warpbreaks)
fm1 <- aov(breaks ~ wool + tension, data = warpbreaks)
t <- tukeyhsd(fm1, "tension", ordered = TRUE)
plot_tukey(t)
```

`print.boot3150` *Prints a boot3150 object*

Description

When given a boot3150 object, this function ensures it prints correctly.

Usage

```
## S3 method for class 'boot3150'
print(x)
```

Arguments

`x`	An object of class boot3150

Value

This function prints the output for a boot3150 object.

`print.boot3150ci` *Prints a boot3150ci object*

Description

When given a boot3150ci object, this function ensures it prints correctly.

Usage

```
## S3 method for class 'boot3150ci'
print(x)
```

Arguments

`x`	An object of class boot3150ci

Value

This function prints the output for a boot3150ci object.

`print.innerboot3150ci` *Prints a innerboot3150ci object*

Description

When given a innerboot3150ci object, this function ensures it prints correctly.

Usage

```
## S3 method for class 'innerboot3150ci'
print(x)
```

Arguments

`x`	An object of class innerboot3150ci

Value

This function prints the output for a innerboot3150ci object.

`residual_plots` *Creates residual and QQ plots*

Description

This function creates a residual and a QQ plot for a given model.

Usage

```
residual_plots(model, resid_type = "jackknife", smoother = TRUE, se = T)
```

Arguments

`model`	The model for which we would like these plots. This can be of class "lm" or "glm" with a binomial family.
`resid_type`	The type of residuals that should be used in the plots. The options are "jackknife", "raw", "standard", and "pearson". Jackknife (externally studentized) residuals are used by default for "lm" objects. Standardized Pearson residuals are used for "glm" objects regardless of what is entered here.
`smoother`	Logical indicating whether a loess smoother should be added to the residual plot.
`se`	Logical indicating whether the standard error bands should be added to the smoother.

Value

This function returns plots. The user must press "enter" or "return" to view subsequent plots.

Examples

```
mod <- lm(mpg ~ disp, data = mtcars)
residual_plots(mod)
```

Index