Package 'prettyglm'

June 24, 2021

Type Package

Title Pretty Summaries of Generalized Linear Model Coefficients

Version 0.1.0

Maintainer Jared Fowler <jared.fowler8@gmail.com>

Description One of the main advantages of using Generalised
Linear Models is their interpretability. The goal of 'prettyglm' is to
provide a set of functions which easily create beautiful coefficient
summaries which can readily be shared and explained.

License GPL-3

Depends R (>= 3.5.0)

Imports broom, car, dplyr, forcats, kableExtra, knitr, plotly,
stringr, tibble, tidycat, tidyr, tidyselect, vip

Suggests rmarkdown, testthat

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Author Jared Fowler [cre, aut]

Repository CRAN

Date/Publication 2021-06-24 07:40:05 UTC

R topics documented:

---

bank_data	*Bank marketing campaigns data set analysis*

Description

It is a dataset that describing Portugal bank marketing campaigns results. Conducted campaigns were based mostly on direct phone calls, offering bank client to place a term deposit. If after all marking afforts client had agreed to place deposit - target variable marked 'yes', otherwise 'no'

Usage

```
data(bank)
```

Format

An object of class `"data.frame"`

job Type of job

marital marital status

education education

default has credit in default?

housing has housing loan?

loan has personal loan?

age age

y has the client subscribed a term deposit? (binary: "yes","no")

Details

Sourse of the data https://archive.ics.uci.edu/ml/datasets/bank+marketing

References

This dataset is public available for research. The details are described in S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

Examples

```
data(bank)
head(bank_data)
```

---

clean_coefficients   *clean_coefficients*

Description

Processing to split out base levels and add variable importance to each term. Directly inspired by 'tidycat::tidy_categorical()', modified for use in prettyglm.

Usage

```
clean_coefficients(d = NULL, m = NULL)
```

Arguments

d	A data frame [tibble](#) output from [tidy.lm](#); with one row for each term in the regression, including column 'term'
m	A model object [glm](#)

Value

Expanded [tibble](#) from the version passed to 'd' including additional columns:

variable	The name of the variable that the regression term belongs to.
level	The level of the categorical variable that the regression term belongs to. Will be an the term name for numeric variables.

Author(s)

Jared Fowler, Guy J. Abel

See Also

[tidy.lm](#)

---

pretty_coefficients   *pretty_coefficients*

Description

Creates a pretty kable of model coefficients including coefficient base levels.

Usage

```
pretty_coefficients(
  model_object,
  relativity_transform = NULL,
  type_iii = NULL,
  conf.int = FALSE,
  return_data = FALSE
)
```

Arguments

model_object   Model object to create coefficient table for. Must be of type: glm, lm, linear_reg
                    or logistic_reg.

relativity_transform
                    String of the function to be applied to the model estimate to calculate the rel-
                    ativity, for example: 'exp(estimate)-1'. Default is for relativity to be excluded
                    from output.

type_iii   Type III statistical test to perform. Default is none. Options are 'Wald' or 'LR'.
                    Warning 'LR' can be computationally expensive. Test performed via Anova

conf.int   Set to TRUE to include confidence intervals in summary table. Warning, can be
                    computationally expensive.

return_data   Set to TRUE to return data.frame instead of creating kable.

Value

kable if return_data = FALSE. data.frame if return_data = TRUE.

Examples

```
library(dplyr)
library(prettyglm)
data('titanic')
columns_to_factor <- c('Pclass',
                       'Sex',
                       'Cabin',
                       'Embarked',
                       'Cabintype',
                       'Survived')
titanic  <- titanic  %>%
  dplyr::mutate_at(columns_to_factor, list(~factor(.)))
survival_model <- stats::glm(Survived ~
                               Pclass +
                               Sex +
                               Age +
                               Fare +
                               Embarked +
                               SibSp +
                               Parch +
                               Cabintype,
```

```
                                data = titanic,
                                family = binomial(link = 'logit'))
pretty_coefficients(survival_model)
```

---

pretty_relativities   *pretty_relativities*

Description

Creates a pretty html plot of model relativities including base Levels.

Usage

```
pretty_relativities(
  feature_to_plot,
  model_object,
  plot_approx_ci = TRUE,
  relativity_transform = "exp(estimate)-1",
  ordering = NULL,
  plot_factor_as_numeric = FALSE,
  width = 800,
  height = 500,
  return_data = FALSE,
  ylabel = "Relativity"
)
```

Arguments

feature_to_plot	A string of the variable to plot.
model_object	Model object to create coefficient table for. Must be of type: [glm,](#) [lm,](#) [linear_reg](#) or [logistic_reg](#)
plot_approx_ci	Set to TRUE to include confidence intervals in summary table. Warning, can be computationally expensive.
relativity_transform	String of the function to be applied to the model estimate to calculate the relativity, for example: 'exp(estimate)'. Default is for relativity to be 'exp(estimate)-1'.
ordering	Type III statistical test to perform. Default is none. Options are 'Wald' or 'LR'. Warning 'LR' can be computationally expensive. Test performed via [Anova](#)
plot_factor_as_numeric	Set to TRUE to return [data.frame](#) instead of creating [kable.](#)
width	Width of plot
height	Height of plot

return_data	Set to TRUE to return data set instead of plot
ylabel	Label for yaxis of relativity plot, some users may prefer to refer to this as log odds

Value

kable if return_data = FALSE. data.frame if return_data = TRUE.

Examples

```
library(dplyr)
library(prettyglm)
data('titanic')
columns_to_factor <- c('Pclass',
                       'Sex',
                       'Cabin',
                       'Embarked',
                       'Cabintype',
                       'Survived')
titanic  <- titanic  %>%
  dplyr::mutate_at(columns_to_factor, list(~factor(.)))
survival_model <- stats::glm(Survived ~
                               Pclass +
                               Sex +
                               Age +
                               Fare +
                               Embarked +
                               SibSp +
                               Parch +
                               Cabintype,
                             data = titanic,
                             family = binomial(link = 'logit'))
pretty_relativities(feature_to_plot = 'Pclass',
                    model_object = survival_model)
```

---

titanic	*Titanic Data*

Description

The sinking of the Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the widely considered "unsinkable" RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren't enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew. While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others. In this challenge, we ask you to build a predictive model that answers the question: "what sorts of people were more likely to survive?" using passenger data (ie name, age, gender, socio-economic class, etc).

Usage

```
data(titanic)
```

Format

An object of class "data.frame"

survival Survival

pclass Ticket class

sex Sex

Age Age in years

sibsp number of siblings / spouses

parch number of parents / children

ticket Ticket number

fare Passenger fare

cabin Cabin Number

cabintype Type of cabin

embarked Port of Embarkation

References

This data set sourced from https://www.kaggle.com/c/titanic/data?select=train.csv

Examples

```
data(titanic)
head(titanic)
```

Index