Package 'distreg.vis'

February 21, 2019

Type Package

Title Framework for the Visualization of Distributional Regression Models

Version 1.1.0

Maintainer Stanislaus Stadlmann <stadlmann@uni-goettingen.de>

Depends R (>= 3.3.0)

Imports stats, utils, methods, shiny (>= 1.0.3), bamlss (>= 0.1-2), gamlss (>= 5.0-6), gamlss.dist (>= 5.1-0), ggplot2 (>= 2.2.1), rhandsontable (>= 0.3.4), magrittr (>= 1.5), mvtnorm (>= 1.0-6), viridis (>= 0.4.0), RColorBrewer (>= 1.1-2), formatR (>= 1.5)

Suggests testthat, gridExtra, glogis

Description Functions for visualizing distributional regression models fitted using the 'gamlss' or 'bamlss' R package. The core of the package consists of a 'shiny' application, where the model results can be interactively explored and visualized.

License GPL-3

LazyData TRUE

URL https://github.com/Stan125/distreg.vis

BugReports https://github.com/Stan125/distreg.vis/issues

RoxygenNote 6.1.1

NeedsCompilation no

Author Stanislaus Stadlmann [cre, aut] (<https://orcid.org/0000-0001-6542-6342>)

Repository CRAN

Date/Publication 2019-02-21 10:20:02 UTC

R topics documented:

distreg.vis 2
dists 2

model_fam_data 3
moments 3
plot_dist 4
plot_moments 5
preds 6
search_funs 7
vis 7

Index **8**

`distreg.vis` `distreg.vis` *package*

Description

Framework for the visualization of distributional regression models

---

`dists` *Information about supported and not yet supported distribution families*

Description

A dataset containing all of bamlss' exported and gamlss.dist families. This is the backbone of the package; whether you can use a family or not depends on this dataset.

Usage

```
dists
```

Format

An object of class `data.frame` with 123 rows and 9 columns.

model_fam_data *Create a dataset to fit models with all possible families in bamlss*

Description

Create a dataset to fit models with all possible families in bamlss

Usage

```
model_fam_data(nrow = 500, seed = 1408, fam_name = "NO")
```

Arguments

nrow	Number of observations of the exported dataset.
seed	The seed which should be used, for reproducibility.
fam_name	The name of the distribution family to which the first dimension of the uniform distribution should be transformed to.

Details

This function creates a 3-dimensional uniform distribution (with support from 0 to 1) which has a cross-correlation of 0.5. Then the first dimension is transformed into a specified distribution (argument fam_name) via Inverse Transform Sampling https://en.wikipedia.org/wiki/Inverse_transform_sampling. The other two dimensions are transformed into a normal distribution (norm2) and a binomial distribution (binomial1, for testing categorical explanatory covariates). This procedure ensures that there is a dependency structure of the transformed first distribution and the other two.

Value

A data.frame with columns for differently distributed data.

moments *Return expected first two moments of a distribution, given the predicted parameters*

Description

This is basically a wrapper for pred.bamlss and pred.gamlss with the added ability to compute special figures that are functions of parameters as well

Usage

```
moments(par, fam_name, what = "mean", ex_fun = NULL)
```

Arguments

par	Parameters of the modeled distribution in a data.frame form. Can be Output of preds, for example.
fam_name	Name of the used family in character form. Can be one of distreg.vis::dists$dist_name. All gamlss.dist and exported bamlss families are supported. To obtain the family from a model in character form, use fam_obtainer.
what	One of "mean", "upperlimit", "lowerlimit". If it is mean (which is also the default), then the mean of the parameter samples is calculated. 2.5 for lowerlimit and upperlimit, respectively.
ex_fun	An external function function(par) {...} which calculates a measure, which dependency from a certain variable is of interest.

---

plot_dist *Plot predicted bamlss distribution families with ggplot2*

Description

This function plots the parameters of a predicted distribution (e.g. obtained through preds) with ggplot2. You can use all implemented families in bamlss except the cox family.

Usage

```
plot_dist(model, pred_params, palette = "default", type = "pdf",
  rug = FALSE)
```

Arguments

model	A fitted bamlss object.
pred_params	A data.frame with rows for every model prediction and columns for every predicted parameter of the distribution. Is easily obtained with the distreg.vis function preds.
palette	The colour palette used for colouring the plot. You can use any of the ones supplied in scale_fill_brewer though I suggest you use one of the qualitative ones: Accent, Dark2, etc. Since 0.5.0 "viridis" is included, to account for colour blindness. If you want to do 3D plots, the accepted palettes are one of: "default"(viridis), "Blues", "Greens", "OrRd", "Purples", "Spectral", "RdYlBu", "RdYlGn".
type	Do you want the probability distribution function ("pdf") or the cumulative distribution function ("cdf")?
rug	If TRUE, creates a rug plot

Value

A ggplot2 object.

Examples

```
# Generating data
data_fam <- model_fam_data(fam_name = "BE")
# Fit model
library("gamlss")
beta_model <- gamlss(BE ~ norm2 + binomial1,
  data = data_fam, family = BE())
# Get 3 predictions
pred_df <- data_fam[sample(1:nrow(data_fam), 3), c("norm2", "binomial1")]
param_preds <- preds(beta_model, pred_df)
# Create pdf, cdf plots
plot_dist(beta_model, param_preds, rug = TRUE)
```

`plot_moments` *Plot function: Display the influence of a covariate*

Description

This function takes a dataframe of predictions with one row per prediction and one column for every explanatory variable. Then, those predictions are held constant while one specific variable is varied over it's whole range (min-max). Then, the constant variables with the varied interest variables are predicted and plotted against the expected value and the variance of the underlying distribution

Usage

```
plot_moments(model, int_var, pred_data, palette = "default",
  rug = FALSE, samples = FALSE, uncertainty = FALSE, ex_fun = NULL)
```

Arguments

`model`	A fitted model on which the plots are based.
`int_var`	The variable for which influences of the moments shall be graphically displayed. Has to be in character form.
`pred_data`	Combinations of covariate data, sometimes also known as "newdata", including the variable of interest, which will be ignored in later processing.
`palette`	See `plot_dist`
`rug`	Should the resulting plot be a rug plot?
`samples`	If the provided model is a bamlss model, should the moment values be "correctly" calculated, using the transformed samples? See details for details.
`uncertainty`	If `TRUE`, displays uncertainty measures about the covariate influences. Can only be `TRUE` if samples is also `TRUE`.
`ex_fun`	An external function `function(par) {...}` which calculates a measure, which dependency from a certain variable is of interest.

Examples

```
library("gamlss")
dat <- model_fam_data(fam_name = "LOGNO")
model <- gamlss(LOGNO ~ ps(norm2) + binomial1,
                ~ ps(norm2) + binomial1,
                data = dat, family = "LOGNO")
ndata <- dat[1:5, c("norm2", "binomial1")]

# Normal plot
plot_moments(model, int_var = "norm2", pred_data = ndata)

# Rug Plot
plot_moments(model, int_var = "norm2", pred_data = ndata, rug = TRUE)

# Using an external function
ineq <- function(par) {
  2 * pnorm((par[["sigma"]] / 2) * sqrt(2)) - 1
}
plot_moments(model, int_var = "norm2", pred_data = ndata, ex_fun = "ineq")
```

`preds` *Predict distributional parameters of a bamlss family with a bamlss model*

Description

This function takes a fitted model and a dataframe with explanatory variables and a column for the intercept to compute predicted parameters for the specified distribution.

Usage

```
preds(model, newdata, what = "mean")
```

Arguments

`model`	A fitted bamlss model object, created with `bamlss`.
`newdata`	A data.frame with explanatory variables as columns, and rows with the combinations you want to do predictions for. Furthermore, whether or not to include the intercept has to be specified via a logical variable `intercept`.
`what`	One of `"mean"` or `"samples"`. The default for bamlss models is "samples", while the default for gamlss models is "mean". This argument changes how the mean of the parameter is calculated. See details for details.

Value

A data.frame with one column for every distributional parameter and a row for every covariate combination that should be predicted.

Examples

```
# Generating data
data_fam <- model_fam_data(fam_name = "BE")
# Fit model
library("gamlss")
beta_model <- gamlss(BE ~ norm2 + binomial1,
  data = data_fam, family = BE())
# Get 3 predictions
pred_df <- data_fam[sample(1:nrow(data_fam), 3), c("binomial1", "norm2")]
param_preds <- preds(beta_model, pred_df)
```

---

search_funs *function Searcher*

Description

Function that looks for objects of class 'function' in the working directory.

Usage

```
search_funs()
```

---

vis *distreg.vis function*

Description

Function to call the distreg.vis Shiny App which represents the core of this package.

Usage

```
vis()
```

Examples

```
library("gamlss")
library("bamlss")
# A gamlss model
normal_gamlss <- gamlss(NO ~ binomial1 + ps(norm2),
                        sigma.formula = ~ binomial1 + ps(norm2),
                        data = model_fam_data(),
                        trace = FALSE)

# Start the App - only in interactive modes
if (interactive()) {
distreg.vis::vis()
}
```

Index

∗Topic **datasets**
dists, 2

bamlss, *6*

distreg.vis, 2
distreg.vis-package *(*distreg.vis*)*, 2
dists, 2

fam_obtainer, *4*

model_fam_data, 3
moments, 3

plot_dist, 4, *5*
plot_moments, 5
preds, *4*, 6

scale_fill_brewer, *4*
search_funs, 7

vis, 7