Package 'cureplots'

October 30, 2024

Type Package

Title CURE (Cumulative Residual) Plots

Version 1.1.1

Description Creates 'ggplot2' Cumulative Residual (CURE) plots to check the goodness-of-fit of a count model; or the tables to create a customized version. A dataset of crashes in Washington state is available for illustrative purposes.

License AGPL (>= 3)

Encoding UTF-8

LazyData true

URL https://github.com/gbasulto/cureplots, https://gbasulto.github.io/cureplots/

BugReports https://github.com/gbasulto/cureplots/issues

Imports dplyr, ggplot2, glue

RoxygenNote 7.3.2

Depends R (>= 2.10)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Language en-US

NeedsCompilation no

Author Jonathan Wood [aut] (<https://orcid.org/0000-0003-0131-6384>), Guillermo Basulto-Elias [aut, cre] (<https://orcid.org/0000-0002-5205-2190>)

Maintainer Guillermo Basulto-Elias <basulto@iastate.edu>

Repository CRAN

Date/Publication 2024-10-30 18:30:02 UTC

Contents

`calculate_cure_dataframe` *Calculate CURE Dataframe*

Description

Calculate CURE Dataframe

Usage

```
calculate_cure_dataframe(covariate_values, residuals)
```

Arguments

`covariate_values`	name to be plot. With or without quotes.
`residuals`	Residuals.

Value

A data frame with five columns: independent variable, residuals, cumulative residuals, lower confidence interval limit, and upper confidence interval limit.

Examples

```
set.seed(2000)

## Define parameters
beta <- c(-1, 0.3, 3)

## Simulate independent variables
n <- 900
AADT <- c(runif(n, min = 2000, max = 150000))
nlanes <- sample(x = c(2, 3, 4), size = n, replace = TRUE)
LNAADT <- log(AADT)

## Simulate dependent variable
theta <- exp(beta[1] + beta[2] * LNAADT + beta[3] * nlanes)
y <- rpois(n, theta)

## Fit model
```

```
mod <- glm(y ~ LNAADT + nlanes, family = poisson)

## Calculate residuals
res <- residuals(mod, type = "response")

## Calculate CURE plot data
cure_df <- calculate_cure_dataframe(AADT, res)

head(cure_df)
```

`cure_plot` *CURE Plot*

Description

CURE Plot

Usage

```
cure_plot(x, covariate = NULL, n_resamples = 0)
```

Arguments

`x`	Either a data frame produced with `calculate_cure_dataframe`, in that case, the first column is used to produce CURE plot; or regression model for count data (e.g., Poisson) adjusted with `glm` or `gam`.
`covariate`	Required when `x` is model fit.
`n_resamples`	Number of resamples to overlay on CURE plot. Zero is the default.

Value

A CURE plot generated with **ggplot2**.

Examples

```
## basic example code

set.seed(2000)

## Define parameters
beta <- c(-1, 0.3, 3)

## Simulate independent variables
n <- 900
AADT <- c(runif(n, min = 2000, max = 150000))
nlanes <- sample(x = c(2, 3, 4), size = n, replace = TRUE)
LNAADT <- log(AADT)

## Simulate dependent variable
```

```
theta <- exp(beta[1] + beta[2] * LNAADT + beta[3] * nlanes)
y <- rpois(n, theta)

## Fit model
mod <- glm(y ~ LNAADT + nlanes, family = poisson)

## Calculate residuals
res <- residuals(mod, type = "response")

## Calculate CURE plot data
cure_df <- calculate_cure_dataframe(AADT, res)

head(cure_df)

## Providing CURE data frame
cure_plot(cure_df)

## Providing glm object
cure_plot(mod, "LNAADT")

## Providing glm object adding resamples cumulative residuals
cure_plot(mod, "LNAADT", n_resamples = 3)
```

`resample_residuals` *Resample residuals*

Description

Resample residuals to compute several cumulative residual curves. Receives the covariate values, residuals and number of samples and shuffles (i.e., samples without replacement a vector of the same size) the residuals, and returns a stacked data frame.

Usage

```
resample_residuals(covariate_values, residuals, n_resamples)
```

Arguments

`covariate_values`	Covariate values.
`residuals`	Residuals.
`n_resamples`	Number of times to sample the residuals.

Value

Data frame of stacked

Examples

```
library(cureplots)
library(ggplot2)
## basic example
set.seed(2000)
## Define parameters.
beta <- c(-1, 0.3, 3)
## Simulate independent variables
n <- 900
AADT <- c(runif(n, min = 2000, max = 150000))
nlanes <- sample(x = c(2, 3, 4), size = n, replace = TRUE)
LNAADT <- log(AADT)
## Simulate dependent variable
theta <- exp(beta[1] + beta[2] * LNAADT + beta[3] * nlanes)
y <- rpois(n, theta)
## Fit model
mod <- glm(y ~ LNAADT + nlanes, family = poisson)
## Calculate residuals
res <- residuals(mod, type = "response")
## Calculate CURE plot data
cure_df <- calculate_cure_dataframe(AADT, res)
resampled_residuals_tbl <- resample_residuals(AADT, res, n_resamples = 3)
ggplot(data = cure_df) +
  aes(AADT, cumres) +
  geom_line(
    data = resampled_residuals_tbl,
    aes(group = sample),
    col = "grey"
  ) +
  geom_line(color = "darkgreen", linewidth = 0.8) +
  geom_line(
    aes(y = lower),
    color = "magenta",
    linetype = "dashed",
    linewidth = 0.8) +
  geom_line(
    aes(y = upper),
    color = "blue",
    linetype = "dashed",
    linewidth = 0.8) +
  theme_bw()
```

`washington_roads` *Washington Road Crashes*

Description

Crashes on Washington primary roads from 2016, 2017, and 2018. Data acquired from Washington Department of Transportation through the Highway Safety Information System (HSIS).

Usage

```
washington_roads
```

Format

The data frame `washington_roads` has 1,501 rows and 9 columns:

ID Anonymized road ID. Factor.

Year Year. Integer.

AADT Annual Average Daily Traffic (AADT). Double.

Length Segment length in miles. Double.

Total_crashes Total crashes. Integer.

lnaadt Natural logarithm of AADT. Double.

lnlength Natural logarithm of length in miles. Double.

speed50 Indicator of whether the speed limit is 50 mph or greater. Binary.

ShouldWidth04 Indicator of whether the shoulder is 4 feet or wider. Binary.

Source

<https://highways.dot.gov/research/safety/hsis>

Index