Package 'insurancerating'

November 1, 2019

Type Package

Title Analytic Insurance Rating Techniques

Version 0.4.3

Author Martin Haringa

Maintainer Martin Haringa <mtharinga@gmail.com>

Description Methods for insurance rating. It provides a data driven strategy for the construction of insurance tariff classes. This strategy is based on the work by Antonio and Valdez (2012) <doi:10.1007/s10182-011-0152-7>.
The package also adds functionality showing additional lines for the reference categories in the levels of the coefficients in the output of a generalized linear regression analysis.
In addition it implements a procedure determining the level of a factor with the largest exposure, and thereafter changing the base level of the factor to this level.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports classInt, data.table, evtree, ggplot2, lubridate, mgcv, stringr

Depends R (>= 3.3)

Suggests testthat, scales

NeedsCompilation no

Repository CRAN

Date/Publication 2019-11-01 16:40:02 UTC

R topics documented:

`autoplot.insurancerating` *Automatically create a ggplot for objects obtained from construct_tariff_classes()*

Description

Takes an object produced by `construct_tariff_classes()`, and plots the predicted claim frequency. In addition the constructed tariff classes are shown.

Usage

```
## S3 method for class 'insurancerating'
autoplot(x, conf_int = FALSE,
  clusters = TRUE, color_gam = "steelblue", color_splits = "grey50",
  xstep = NULL, add_points = FALSE, size_points = 1,
  color_points = "black", rotate_labels = FALSE,
  remove_outliers = NULL)
```

Arguments

`x`	an object as produced by `construct_tariff_classes()`
`conf_int`	determines whether 95% confidence intervals will be plotted. The default is `conf_int = FALSE`
`clusters`	numerical vector with splits as produced by `construct_tariff_classes()`
`color_gam`	a color can be specified either by name (e.g.: "red") or by hexadecimal code (e.g. : "#FF1234") (default is "steelblue")
`color_splits`	change the color of the splits in the graph ("grey50" is default)
`xstep`	set step size for labels horizontal axis
`add_points`	add observed frequency/severity points for each level of the variable for which tariff classes are constructed
`size_points`	size for points (1 is default)
`color_points`	change the color of the points in the graph ("black" is default)
`rotate_labels`	rotate x-labels 45 degrees (this might be helpful for overlapping x-labels)
`remove_outliers`	do not show observations above this number in the plot. This might be helpful for outliers.

Value

a ggplot object

Author(s)

Martin Haringa

Examples

```
library(ggplot2)
x <- construct_tariff_classes(MTPL, nclaims, age_policyholder, exposure)
autoplot(x)
```

`biggest_reference` *Set reference group to the group with largest exposure*

Description

This function specifies the first level of a factor to the level with the largest exposure. Levels of factors are sorted using an alphabetic ordering. If the factor is used in a regression context, then the first level will be the reference. For insurance applications it is common to specify the reference level to the level with the largest exposure.

Usage

```
biggest_reference(x, weight)
```

Arguments

`x`	an unordered factor
`weight`	a vector containing weights (e.g. exposure). Should be numeric.

Value

a factor of the same length as x

Author(s)

Martin Haringa

References

Kaas, Rob & Goovaerts, Marc & Dhaene, Jan & Denuit, Michel. (2008). Modern Actuarial Risk Theory: Using R. doi:10.1007/978-3-540-70998-5.

Examples

```
## Not run:
library(dplyr)
df <- chickwts %>%
mutate_if(is.character, as.factor) %>%
mutate_if(is.factor, list(~biggest_reference(., weight)))

## End(Not run)
```

`construct_tariff_classes` *Construct insurance tariff classes*

Description

The function provides an interface to finding class intervals for continuous numerical variables. The goal is to bin the continuous factors such that categorical risk factors result which capture the effect of the covariate on the response in an accurate way, while being easy to use in a generalized linear model (GLM).

Usage

```
construct_tariff_classes(data, nclaims, x, exposure, amount = NULL,
  pure_premium = NULL, model = "frequency", alpha = 0,
  niterations = 10000, ntrees = 200, seed = 1, round_x = NULL)
```

Arguments

`data`	data.frame of an insurance portfolio
`nclaims`	column in `data` with number of claims
`x`	column in `data` with continuous risk factor
`exposure`	column in `data` with exposure
`amount`	column in `data` with claim amount
`pure_premium`	column in `data` with pure premium
`model`	choose either 'frequency', 'severity' or 'burning' (model = 'frequency' is default). See details section.
`alpha`	complexity parameter. The complexity parameter (alpha) is used to control the number of tariff classes. Higher values for `alpha` render less tariff classes. (`alpha` = 0 is default).
`niterations`	in case the run does not converge, it terminates after a specified number of iterations defined by niterations.
`ntrees`	the number of trees in the population.
`seed`	an numeric seed to initialize the random number generator (for reproducibility).
`round_x`	round elements in column `x` to multiple of `round_x`. This gives a speed enhancement for data containing many levels for `x`.

Details

The function provides an interface to finding class intervals for continuous numerical variables in the following three types of models: claim frequency, claim severity or burning cost model. The 'frequency' specification uses a Poisson GAM for fitting the number of claims. The logarithm of the exposure is included as an offset, such that the expected number of claims is proportional to the exposure. The 'severity' specification uses a lognormal GAM for fitting the average cost of a claim. The average cost of a claim is defined as the ratio of the claim amount and the number of claims. The number of claims is included as a weight. The 'burning' specification uses a lognormal GAM for fitting the pure premium of a claim. The pure premium is obtained by multiplying the estimated frequency and the estimated severity of claims. The word burning cost is used here as equivalent of risk premium and pure premium.

Subsequently, evolutionary trees are used as a technique to bin the resulting GAM estimates into risk homogeneous categories. This method is based on the work by Henckaerts et al. (2018). See Grubinger et al. (2014) for more details on the various parameters that control aspects of the evtree fit.

Value

A list with components

`splits`	vector with boundaries of the constructed tariff classes
`prediction`	data frame with the predicted claim frequency for each element of vector `x`
`x`	name of variable for which tariff classes are constructed
`tariff_classes`	values in vector `x` coded according to which constructed tariff class they fall
`model`	either 'frequency' or 'severity'
`data`	data frame with original data aggregated on the level of the variable for which tariff classes are constructed

Author(s)

Martin Haringa

References

Henckaerts, R., Antonio, K., Clijsters, M. and Verbelen, R. (2018). A data driven binning strategy for the construction of insurance tariff classes. Scandinavian Actuarial Journal, 2018:8, 681-705. doi:10.1080/03461238.2018.1429300.

Antonio, K. and Valdez, E. A. (2012). Statistical concepts of a priori and a posteriori risk classification in insurance. Advances in Statistical Analysis, 96(2):187–224. doi:10.1007/s10182-011-0152-7.

Grubinger, T., Zeileis, A., and Pfeiffer, K.-P. (2014). evtree: Evolutionary learning of globally optimal classification and regression trees in R. Journal of Statistical Software, 61(1):1–29. doi:10.18637/jss.v061.i01.

Wood, S.N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. Journal of the Royal Statistical Society (B) 73(1):3-36. doi:10.1111/j.1467-9868.2010.00749.x.

Examples

```
construct_tariff_classes(MTPL, nclaims, age_policyholder, exposure)
```

`fisher` *Fisher's natural breaks classification*

Description

The function provides an interface to finding class intervals for continuous numerical variables, for example for choosing colours for plotting maps.

Usage

```
fisher(vec, n = 7, diglab = 2)
```

Arguments

`vec`	a continuous numerical variable
`n`	number of classes required (n = 7 is default)
`diglab`	number of digits (n = 2 is default)

Details

The "fisher" style uses the algorithm proposed by W. D. Fisher (1958) and discussed by Slocum et al. (2005) as the Fisher-Jenks algorithm. This function is adopted from the classInt package.

Value

Vector with clustering

Author(s)

Martin Haringa

References

Bivand, R. (2018). classInt: Choose Univariate Class Intervals. R package version 0.2-3. https://CRAN.R-project.org/package=classInt

Fisher, W. D. 1958 "On grouping for maximum homogeneity", Journal of the American Statistical Association, 53, pp. 789–798. doi: 10.1080/01621459.1958.10501479.

get_splits *Get splits from partykit object*

Description

Get splits from partykit object

Usage

```
get_splits(x)
```

Arguments

x	A party object.

MTPL *Ages of 32,731 policyholders in a Motor Third Party Liability (MTPL) portfolio.*

Description

A dataset containing the age, number of claims, and exposure of almost 33,000 policyholders

Usage

```
MTPL
```

Format

A data frame with 32,731 rows and 4 variables:

age_policyholder age of policyholder, in years.

nclaims number of claims.

exposure exposure, for example, if a vehicle is insured as of July 1 for a certain year, then during that year, this would represent an exposure of 0.5 to the insurance company.

amount claim amount in Euros.

Author(s)

Martin Haringa

Source

The data is derived from the portfolio of a large Dutch motor insurance company.

`period_to_months` *Split period to months*

Description

The function splits rows with a time period longer than one month to multiple rows with a time period of exactly one month each. Values in numeric columns (e.g. exposure or premium) are divided over the months proportionately.

Usage

```
period_to_months(df, begin, end, ...)
```

Arguments

`df`	data.frame
`begin`	column in `df` with begin dates
`end`	column in `df` with end dates
`...`	numeric columns in df to split

Details

In insurance portfolios it is common that rows relate to periods longer than one month. This is for example problematic in case exposures per month are desired.

Since insurance premiums are constant over the months, and do not depend on the number of days per month, the function assumes that each month has the same number of days (i.e. 30).

Value

data.frame with same columns as in df, and one extra column called id

Author(s)

Martin Haringa

Examples

```
library(lubridate)
portfolio <- data.frame(
begin1 = ymd(c("2014-01-01", "2014-01-01")),
end = ymd(c("2014-03-14", "2014-05-10")),
termination = ymd(c("2014-03-14", "2014-05-10")),
exposure = c(0.2025, 0.3583),
premium =  c(125, 150))
period_to_months(portfolio, begin1, end, premium, exposure)
```

rating_factors *Include reference group in regression output*

Description

This extracts coefficients in terms of the original levels of the coefficients rather than the coded variables.

Usage

```
rating_factors(model, colname = "estimate", exponentiate = TRUE)
```

Arguments

model	a (generalized) linear model fit
colname	name of column with estimates. Defaults to "estimate".
exponentiate	Logical indicating whether or not to exponentiate the the coefficient estimates. Defaults to TRUE.

Details

This function is adopted from the dummy.coefstats function. Our adoption prints a data.frame as output. Categorical variables should be changed to factors in the data.frame used to fit the (generalized) linear model.

Value

data.frame

Author(s)

Martin Haringa

Examples

```
g1 <- glm(nclaims ~ age_policyholder, family = "poisson", data = MTPL)
rating_factors(g1)
```

Index