# Package 'insurancerating'

February 7, 2019

**Type** Package

**Title** Analytic Insurance Rating Techniques

**Version** 0.4.0

**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, ggplot2, mgcv, rpart

**Depends** R (>= 3.3)

**Suggests** testthat

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-02-06 23:11:02 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 = 10)
```

## 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 horizontal axis (default is `10`) |

## 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, funs(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,
  approximation = TRUE, cp = 0)
```

## 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 |
| `approximation` | if TRUE, elements in `nclaims` and `exposure` are aggregated to the level of unique elements in `x`. Approximation should be used for large insurance portfolios to avoid excessive computation times (default is TRUE). |
| `cp` | complexity parameter. The complexity parameter (cp) is used to control the number of tariff classes. Higher values for `cp` render less tariff classes. (`cp` = 0 is default). |

## Details

Poisson GAMs are used 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. Subsequently, regression trees are used as a technique to bin the resulting GAM estimates into risk homogeneous categories. This method is based on the work by Antonio and Valdez (2012).

## 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 |

## Author(s)

Martin Haringa

## References

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.

Therneau, T. and Atkinson, B. (2018). rpart: Recursive Partitioning and Regression Trees. R package version 4.1-13. https://CRAN.R-project.org/package=rpart

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)
```

## Arguments

| | |
|---|---|
| vec | a continuous numerical variable |
| n | number of classes required (n = 7 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.

---

`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.

`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.

## Value

data.frame

## Author(s)

Martin Haringa

## Examples

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

# Index