

# Package ‘insurancerating’

May 31, 2019

**Type** Package

**Title** Analytic Insurance Rating Techniques

**Version** 0.4.2

**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, evtree, ggplot2, mgcv, stringr

**Depends** R (>= 3.3)

**Suggests** testthat, scales

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-05-31 20:00:03 UTC

## R topics documented:

autoplot.insurancerating . . . . .	2
biggest_reference . . . . .	3
construct_tariff_classes . . . . .	4
fisher . . . . .	6
get_splits . . . . .	7
MTPL . . . . .	7
rating_factors . . . . .	8

---

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, add_points = FALSE, size_points = 1,
         color_points = "black")
```

### Arguments

<code>x</code>	an object as produced by <code>construct_tariff_classes()</code>
<code>conf.int</code>	determines whether 95% confidence intervals will be plotted. The default is <code>conf.int = FALSE</code>
<code>clusters</code>	numerical vector with splits as produced by <code>construct_tariff_classes()</code>
<code>color_gam</code>	a color can be specified either by name (e.g.: "red") or by hexadecimal code (e.g. : "#FF1234") (default is "steelblue")
<code>color_splits</code>	change the color of the splits in the graph ("grey50" is default)
<code>xstep</code>	set step size for horizontal axis (default is 10)
<code>add_points</code>	add observed frequency/severity points for each level of the variable for which tariff classes are constructed
<code>size_points</code>	size for points (1 is default)
<code>color_points</code>	change the color of the points in the graph ("black" is default)

### 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, amount = NULL,
  pure_premium = NULL, model = "frequency", alpha = 0,
  niterations = 10000, ntrees = 200, seed = 1)
```

### 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).

### 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	<i>Get splits from partykit object</i>
------------	--

---

**Description**

Get splits from partykit object

**Usage**

```
get_splits(x)
```

**Arguments**

x                    A party object.

---

MTPL	<i>Ages of 32,731 policyholders in a Motor Third Party Liability (MTPL) portfolio.</i>
------	--

---

**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	<i>Include reference group in regression output</i>
----------------	---

---

**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

\*Topic **datasets**

MTPL, [7](#)

`autoplot.insurancerating`, [2](#)

`biggest_reference`, [3](#)

`construct_tariff_classes`, [4](#)

`fisher`, [6](#)

`get_splits`, [7](#)

MTPL, [7](#)

`rating_factors`, [8](#)