Package ‘varbin’

October 12, 2018

**Type**  Package

**Title**  Optimal Binning of Continuous and Categorical Variables

**Version**  0.2.1

**Date**  2018-09-21

**Author**  Daniel Safai

**Maintainer**  Daniel Safai <danielsafai@gmail.com>

**Description**  Tool for easy and efficient discretization of continuous and categorical data. The package calculates the most optimal binning of a given explanatory variable with respect to a user-specified target variable. The purpose is to assign a unique Weight-of-Evidence value to each of the calculated binpoints in order to recode the original variable. The package allows users to impose certain restrictions on the functional form on the resulting binning while maximizing the overall information value in the original data. The package is well suited for logistic scoring models where input variables may be subject to restrictions such as linearity by e.g. regulatory authorities. An excellent source describing in detail the development of scorecards, and the role of Weight-of-Evidence coding in credit scoring is (Siddiqi 2006, ISBN: 978–0-471–75451–0). The package utilizes the discrete nature of decision trees and Isotonic Regression to accommodate the trade-off between flexible functional forms and maximum information value.

**License**  GPL (>= 2)

**Imports**  rpart

**Encoding**  UTF-8

**LazyData**  true

**RoxygenNote**  6.1.0

**NeedsCompilation**  no

**Repository**  CRAN

**Date/Publication**  2018-10-12 13:50:03 UTC
R topics documented:

```
varbin . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
varbin.convert . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
varbin.factor . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
varbin.kink . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
varbin.monotonic . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
varbin.plot . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
```

Index 8

```
varbin

Description
Optimal binning of numerical variable

Usage
```
varbin(df, x, y, p=0.05, custom_vec=NA)
```

Arguments
```
df        A data frame
x          String. Name of continuous variable in data frame.
y          String. Name of binary response variable (0,1) in data frame.
p          Percentage of records per bin. Default 5 pct. (0.05). This parameter only accepts
            values greater than 0.00 (0 pct.) and lower than 0.50 (50 pct.).
custom_vec Numerical input vector with custom cutpoints. E.g. custom_vec=c(20, 50, 75) for a variable
            representing age, will result in the cutpoints [<20, <50, <75, >=75]. NA results in
            the default unrestricted (most optimal) binning.
```

Value
The command varbin generates a data frame with necessary info and utilities for binning. The user
should save the output result so it can be used with e.g. varbin.plot, or varbin.convert.

Examples
```
# Set seed and generate data
set.seed(1337)
target <- as.numeric(runif(10000, 0, 1)<0.2)
age <- round(rnorm(10000, 40, 15), 0)
age[age<20] <- round(rnorm(sum(age<20), 40, 5), 0)
age[age>95] <- round(rnorm(sum(age>95), 40, 5), 0)
inc <- round(rnorm(10000, 100000, 10000), 0)
educ <- sample(c("MSC", "BSC", "SELF", "PHD", "OTHER"), 10000, replace=TRUE)
```
varbin.convert

\[
\text{df} \leftarrow \text{data.frame}(\text{target=target, age=age, inc=inc, educ=educ})
\]

# Perform unrestricted binning
\[
\text{result} \leftarrow \text{varbin(df, "age", "target")}
\]

# Perform custom binning
\[
\text{result2} \leftarrow \text{varbin(df, "age", "target", custom_vec=c(30,40,60,75))}
\]

---

Description

Generate new variable based on constructed binnings

Usage

\[
\text{varbin.convert(df, ivTable, x)}
\]

Arguments

- \textbf{df} \quad \text{A data frame}
- \textbf{ivTable} \quad \text{Output from either \textsc{varbin}, \textsc{varbin.factor}, \textsc{varbin.monotonic} or \textsc{varbin.kink.}}
- \textbf{x} \quad \text{String. Name of variable in data frame for which binnings should be applied.}

Value

The command varbin.convert appends a new variable named "WoE_[x]" to the data frame. The new variable consist of the Weight of Evidence values from the resulting binning.

Examples

# Set seed and generate data
\[
\text{set.seed(1337)}
\]
\[
\text{target} \leftarrow \text{as.numeric(runif(10000, 0, 1)<0.2)}
\]
\[
\text{age} \leftarrow \text{round(rnorm(10000, 40, 15), 0)}
\]
\[
\text{age[age<20]} \leftarrow \text{round(rnorm(sum(age<20), 40, 5), 0)}
\]
\[
\text{age[age>95]} \leftarrow \text{round(rnorm(sum(age>95), 40, 5), 0)}
\]
\[
\text{inc} \leftarrow \text{round(rnorm(10000, 100000, 10000), 0)}
\]
\[
\text{educ} \leftarrow \text{sample(c("MSC", "BSC", "SELF", "PHD", "OTHER"), 10000, replace=TRUE)}
\]
\[
\text{df} \leftarrow \text{data.frame(target=target, age=age, inc=inc, educ=educ)}
\]

# Split train/test
\[
\text{df_train} \leftarrow \text{df[1:5000, ]}
\]
\[
\text{df_test} \leftarrow \text{df[5001:nrow(df), ]}
\]

# Perform unrestricted binnings
\[
\text{result} \leftarrow \text{varbin.factor(df_train, "educ", "target")}
\]
\[
\text{result2} \leftarrow \text{varbin(df_train, "age", "target")}
\]
varbin.factor

Description
Binning of categorical variable

Usage
varbin.factor(df, x, y, custom_vec=NA)

Arguments
df A data frame
x String. Name of factor variable in data frame.
y String. Name of binary response variable (0,1) in data frame.
custom_vec Character input vector with custom cutpoints. E.g. custom_vec=c("STUDENT", "UNEMP,RETIRED", "EMPLOYED") for a variable representing occupation, will result in the cutpoints ["STUDENT", "UNEMP,RETIRED", "EMPLOYED"]. NA results in default binning (no binning) i.e. the cutpoints ["STUDENT", "UNEMP", "RETIRED", "EMPLOYED"] corresponding to the levels of the factor variable.

Value
The command varbin generates a data frame with necessary info and utilities for binning. The user should save the output result so it can be used with e.g. varbin.plot, or varbin.convert.

Examples
# Set seed and generate data
set.seed(1337)
target <- as.numeric(runif(10000, 0, 1)<0.2)
age <- round(rnorm(10000, 40, 15), 0)
age[age<20] <- round(rnorm(sum(age<20), 40, 5), 0)
age[age>95] <- round(rnorm(sum(age>95), 40, 5), 0)
inc <- round(rnorm(10000, 100000, 10000), 0)
educ <- sample(c("MSC", "BSC", "SELF", "PHD", "OTHER"), 10000, replace=TRUE)
df <- data.frame(target=target, age=age, inc=inc, educ=educ)

# Perform unrestricted binning
result <- varbin.factor(df, "educ", "target")

# Perform custom binning
result2 <- varbin.factor(df, "educ", "target", custom_vec=c("MSC,BSC,PHD", "SELF", "OTHER"))
Description

Impose global/local extremum i.e. a kink restriction on binning of numerical variable (if possible)

Usage

varbin.kink(df, x, y, p=0.05)

Arguments

df  A data frame

x   String. Name of continuous variable in data frame.

y   String. Name of binary response variable (0,1) in data frame.

p   Percentage of records per bin. Default 5 pct. (0.05). This parameter only accepts values greater than 0.00 (0 pct.) and lower than 0.50 (50 pct.).

Value

The command varbin.kink generates a data frame with necessary info and utilities for a variable where the binnings are restricted such that the functional form is characterized by having a global/local minimum/maximum i.e. a kink. The function will not work for variables where both a monotonically in- or decreasing functional form can’t be imposed. The user should save the output result so it can be used with e.g. varbin.plot, or varbin.convert.

Examples

# Set seed and generate data
set.seed(1337)
target <- as.numeric(runif(10000, 0, 1)<0.2)
age <- round(rnorm(10000, 40, 15), 0)
age[age<20] <- round(rnorm(sum(age<20), 40, 5), 0)
age[age>95] <- round(rnorm(sum(age>95), 40, 5), 0)
inc <- round(rnorm(10000, 100000, 10000), 0)
educ <- sample(c("MSC", "BSC", "SELF", "PHD", "OTHER"), 10000, replace=TRUE)
df <- data.frame(target=target, age=age, inc=inc, educ=educ)

# Perform restricted binning - note the kink shape of the WoE values in the output
result <- varbin.kink(df, "inc", "target")
Description

Monotonically in- or decreasing restriction on binning of numerical variable

Usage

varbin.monotonic(df, x, y, p=0.05, increase=F, decrease=F, auto=T)

Arguments

- df : A data frame
- x : String. Name of continuous variable in data frame.
- y : String. Name of binary response variable (0,1) in data frame.
- p : Percentage of records per bin. Default 5 pct. (0.05). This parameter only accepts values greater than 0.00 (0 pct.) and lower than 0.50 (50 pct.).
- increase : Logical (TRUE/FALSE). Whether to force an increasing monotonic functional form (if possible)
- decrease : Logical (TRUE/FALSE). Whether to force a decreasing monotonic functional form (if possible)
- auto : Logical (TRUE/FALSE). Whether to choose which of the two above is most optimal

Value

The command varbin generates a data frame with necessary info and utilities for a monotonically in- or decreasing functional form restriction imposed to the binning. The user should save the output result so it can be used with e.g. varbin.plot, or varbin.convert.

Examples

# Set seed and generate data
set.seed(1337)
target <- as.numeric(runif(10000, 0, 1)<0.2)
age <- round(rnorm(10000, 40, 15), 0)
age[age<20] <- round(rnorm(sum(age<20), 40, 5), 0)
age[age>95] <- round(rnorm(sum(age>95), 40, 5), 0)
inc <- round(rnorm(10000, 100000, 10000), 0)
educ <- sample(c("MSC", "BSC", "SELF", "PHD", "OTHER"), 10000, replace=TRUE)
df <- data.frame(target=target, age=age, inc=inc, educ=educ)

# Perform monotonically restricted binning
result <- varbin.monotonic(df, "inc", "target")
Description

Generate simple plot to visualize binning results

Usage

`varbin.plot(ivTable)`

Arguments

`ivTable`  
Output from either varbin, varbin.factor, varbin.monotonic or varbin.kink.

Value

The command varbin.plot generates a simple plot with the Weight of Evidence values on the y-axis and the cutpoints/binnings on the x-axis. Gives a nice overview of the functional form and the relationship between the explanatory variable and the dependent variable.

Examples

```r
# Set seed and generate data
set.seed(1337)
target <- as.numeric(runif(10000, 0, 1)<0.2)
age <- round(rnorm(10000, 40, 15), 0)
age[age<20] <- round(rnorm(sum(age<20), 40, 5), 0)
age[age>95] <- round(rnorm(sum(age>95), 40, 5), 0)
inc <- round(rnorm(10000, 100000, 10000), 0)
educ <- sample(c("MSC", "BSC", "SELF", "PHD", "OTHER"), 10000, replace=TRUE)
df <- data.frame(target=target, age=age, inc=inc, educ=educ)

# Perform restricted binning
result <- varbin.kink(df, "inc", "target")
# Plot result
varbin.plot(result)
```
Index

varbin, 2
varbin.convert, 3
varbin.factor, 4
varbin.kink, 5
varbin.monotonic, 6
varbin.plot, 7