# Package 'smbinning'

October 14, 2022

**Title** Scoring Modeling and Optimal Binning

**Version** 0.9

**Author** Herman Jopia

**Maintainer** Herman Jopia <hjopia@gmail.com>

**Description** A set of functions to build a scoring model from beginning to end, leading the user
to follow an efficient and organized development process, reducing significantly the time
spent on data exploration, variable selection, feature engineering, binning and model selection
among other recurrent tasks.
The package also incorporates monotonic and customized binning, scaling capabilities that
transforms logistic coefficients into points for a better business understanding and
calculates and visualizes classic performance metrics of a classification model.

**Depends** R (>= 3.2.0),sqldf,partykit,Formula

**Imports** gsubfn

**License** GPL (>= 2)

**LazyData** true

**RoxygenNote** 6.1.1

**Encoding** UTF-8

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-04-01 10:50:03 UTC

## R topics documented:

---

smbinning                          *Optimal Binning for Scoring Modeling*

---

### Description

**Optimal Binning** categorizes a numeric characteristic into bins for ulterior usage in scoring modeling. This process, also known as *supervised discretization*, utilizes Recursive Partitioning to categorize the numeric characteristic.

The especific algorithm is Conditional Inference Trees which initially excludes missing values (NA) to compute the cutpoints, adding them back later in the process for the calculation of the *Information Value*.

### Usage

```
smbinning(df, y, x, p = 0.05)
```

### Arguments

| | |
|---|---|
| df | A data frame. |
| y | Binary response variable (0,1). Integer (int) is required. Name of y must not have a dot. Name "default" is not allowed. |
| x | Continuous characteristic. At least 5 different values. Value Inf is not allowed. Name of x must not have a dot. |
| p | Percentage of records per bin. Default 5% (0.05). This parameter only accepts values greater that 0.00 (0%) and lower than 0.50 (50%). |

### Value

The command smbinning generates and object containing the necessary info and utilities for binning. The user should save the output result so it can be used with smbinning.plot, smbinning.sql, and smbinning.gen.

## Examples

```
# Load library and its dataset
library(smbinning) # Load package and its data

# Example: Optimal binning
result=smbinning(df=smbsimdf1,y="fgood",x="cbs1") # Run and save result
result$ivtable # Tabulation and Information Value
result$iv # Information value
result$bands # Bins or bands
result$ctree # Decision tree
```

---

smbinning.custom *Customized Binning*

---

## Description

It gives the user the ability to create customized cutpoints.

## Usage

```
smbinning.custom(df, y, x, cuts)
```

## Arguments

| | |
|---|---|
| df | A data frame. |
| y | Binary response variable (0,1). Integer (int) is required. Name of y must not have a dot. Name "default" is not allowed. |
| x | Continuous characteristic. At least 5 different values. Value Inf is not allowed. Name of x must not have a dot. |
| cuts | Vector with the cutpoints selected by the user. It does not have a default so user must define it. |

## Value

The command smbinning.custom generates and object containing the necessary info and utilities for binning. The user should save the output result so it can be used with smbinning.plot, smbinning.sql, and smbinning.gen.

## Examples

```
# Load library and its dataset
library(smbinning) # Load package and its data

# Custom cutpoints using percentiles (20% each)
cbs1cuts=as.vector(quantile(smbsimdf1$cbs1, probs=seq(0,1,0.2), na.rm=TRUE)) # Quantiles
cbs1cuts=cbs1cuts[2:(length(cbs1cuts)-1)] # Remove first (min) and last (max) values
```

```
# Example: Customized binning
result=smbinning.custom(df=smbsimdf1,y="fgood",x="cbs1",cuts=cbs1cuts) # Run and save
result$ivtable # Tabulation and Information Value
```

---

smbinning.eda                    *Exploratory Data Analysis (EDA)*

---

### Description

It shows basic statistics for each characteristic in a data frame. The report includes:

- Field: Field name.
- Type: Factor, numeric, integer, other.
- Recs: Number of records.
- Miss: Number of missing records.
- Min: Minimum value.
- Q25: First quartile. It splits off the lowest 25% of data from the highest 75%.
- Q50: Median or second quartile. It cuts data set in half.
- Avg: Average value.
- Q75: Third quartile. It splits off the lowest 75% of data from the highest 25%.
- Max: Maximum value.
- StDv: Standard deviation of a sample.
- Neg: Number of negative values.
- Pos: Number of positive values.
- OutLo: Number of outliers. Records below `Q25-1.5*IQR`, where `IQR=Q75-Q25`.
- OutHi: Number of outliers. Records above `Q75+1.5*IQR`, where `IQR=Q75-Q25`.

### Usage

```
smbinning.eda(df, rounding = 3, pbar = 1)
```

### Arguments

| | |
|---|---|
| df | A data frame. |
| rounding | Optional parameter to define the decimal points shown in the output table. Default is 3. |
| pbar | Optional parameter that turns on or off a progress bar. Default value is 1. |

### Value

The command `smbinning.eda` generates two data frames that list each characteristic with basic statistics such as extreme values and quartiles; and also percentages of missing values and outliers, among others.

## Examples

```
# Load library and its dataset
library(smbinning) # Load package and its data

# Example: Exploratory data analysis of dataset
smbinning.eda(smbsimdf1,rounding=3)$eda # Table with basic statistics
smbinning.eda(smbsimdf1,rounding=3)$edapct # Table with basic percentages
```

---

smbinning.factor        *Binning on Factor Variables*

---

## Description

It generates a table with relevant metrics for all the categories of a given factor variable.

## Usage

```
smbinning.factor(df, y, x, maxcat = 10)
```

## Arguments

| | |
|---|---|
| df | A data frame. |
| y | Binary response variable (0,1). Integer (int) is required. Name of y must not have a dot. |
| x | A factor variable with at least 2 different values. Labesl with commas are not allowed. |
| maxcat | Specifies the maximum number of categories. Default value is 10. Name of x must not have a dot. |

## Value

The command smbinning.factor generates and object containing the necessary info and utilities for binning. The user should save the output result so it can be used with smbinning.plot, smbinning.sql, and smbinning.gen.factor.

## Examples

```
# Load library and its dataset
library(smbinning) # Load package and its data

# Binning a factor variable
result=smbinning.factor(smbsimdf1,x="inc",y="fgood", maxcat=11)
result$ivtable
```

---

smbinning.factor.custom

*Customized Binning on Factor Variables*

---

**Description**

It gives the user the ability to combine categories and create new attributes for a given characteristic. Once these new attribues are created in a list (called `groups`), the funtion generates a table for the uniques values of a given factor variable.

**Usage**

```
smbinning.factor.custom(df, y, x, groups)
```

**Arguments**

| | |
|---|---|
| df | A data frame. |
| y | Binary response variable (0,1). Integer (`int`) is required. Name of y must not have a dot. |
| x | A factor variable with at least 2 different values. Value `Inf` is not allowed. |
| groups | Specifies customized groups created by the user. Name of x must not have a dot. |

**Value**

The command `smbinning.factor.custom` generates an object containing the necessary information and utilities for binning. The user should save the output result so it can be used with `smbinning.plot`, `smbinning.sql`, and `smbinning.gen.factor`.

**Examples**

```
# Load library and its dataset
library(smbinning) # Load package and its data

# Example: Customized binning for a factor variable
# Notation: Groups between double quotes
result=smbinning.factor.custom(
  smbsimdf1,x="inc",
  y="fgood",
  c("'W01','W02'",         # Group 1
    "'W03','W04','W05'",   # Group 2
    "'W06','W07'",         # Group 3
    "'W08','W09','W10'")) # Group 4
result$ivtable
```

---

smbinning.factor.gen     *Utility to generate a new characteristic from a factor variable*

---

**Description**

It generates a data frame with a new predictive characteristic from a factor variable after applying
smbinning.factor or smbinning.factor.custom.

**Usage**

```
smbinning.factor.gen(df, ivout, chrname = "NewChar")
```

**Arguments**

| | |
|---|---|
| df | Dataset to be updated with the new characteristic. |
| ivout | An object generated after smbinning.factor or smbinning.factor.custom. |
| chrname | Name of the new characteristic. |

**Value**

A data frame with the binned version of the original characteristic.

**Examples**

```
# Load library and its dataset
library(smbinning) # Load package and its data
pop=smbsimdf1 # Set population
train=subset(pop,rnd<=0.7) # Training sample

# Binning a factor variable on training data
result=smbinning.factor(train,x="home",y="fgood")

# Example: Append new binned characteristic to population
pop=smbinning.factor.gen(pop,result,"g1home")

# Split training
train=subset(pop,rnd<=0.7) # Training sample

# Check new field counts
table(train$g1home)
table(pop$g1home)
```

---

smbinning.gen *Utility to generate a new characteristic from a numeric variable*

---

### Description

It generates a data frame with a new predictive characteristic after applying smbinning or smbinning.custom.

### Usage

```
smbinning.gen(df, ivout, chrname = "NewChar")
```

### Arguments

df           Dataset to be updated with the new characteristic.

ivout        An object generated after smbinning or smbinning.custom.

chrname      Name of the new characteristic.

### Value

A data frame with the binned version of the original characteristic.

### Examples

```
# Load library and its dataset
library(smbinning) # Load package and its data
pop=smbsimdf1 # Set population
train=subset(pop,rnd<=0.7) # Training sample

# Binning application for a numeric variable
result=smbinning(df=train,y="fgood",x="dep") # Run and save result

# Generate a dataset with binned characteristic
pop=smbinning.gen(pop,result,"g1dep")

# Check new field counts
table(pop$g1dep)
```

---

smbinning.logitrank *Logistic Regression Ranking*

---

### Description

It runs all the possible logistic models for a given set of characteristics (chr) and then rank them from highest to lowest performance based on AIC. Important Note: This function may take time depending on the datset size and number of variables used in it. The user should run it at the end of the modeling process once variables have been pre-selected in previous steps.

## Usage

```
smbinning.logitrank(y, chr, df)
```

## Arguments

| | |
|---|---|
| y | Binary dependent variable. |
| chr | Vector with the characteristics (independent variables). |
| df | Data frame. |

## Value

The command smbinning.logitrank returns a table with the combination of characteristics and their corresponding AIC and deviance. The table is ordered by AIC from lowest (best) to highest.

## Examples

```
# Load library and its dataset
library(smbinning) # Load package and its data

# Example: Best combination of characteristics
smbinning.logitrank(y="fgood",chr=c("chr1","chr2","chr3"),df=smbsimdf3)
```

---

| smbinning.metrics | *Performance Metrics for a Classification Model* |
|---|---|

---

## Description

It computes the classic performance metrics of a scoring model, including AUC, KS and all the relevant ones from the classification matrix at a specific threshold or cutoff.

## Usage

```
smbinning.metrics(dataset, prediction, actualclass, cutoff = NA,
  report = 1, plot = "none", returndf = 0)
```

## Arguments

| | |
|---|---|
| dataset | Data frame. |
| prediction | Classifier. A value generated by a classification model (Must be numeric). |
| actualclass | Binary variable (0/1) that represents the actual class (Must be numeric). |
| cutoff | Point at wich the classifier splits (predicts) the actual class (Must be numeric). If not specified, it will be estimated by using the maximum value of Youden J (Sensitivity+Specificity-1). If not found in the data frame, it will take the closest lower value. |
| report | Indicator defined by user. 1: Show report (Default), 0: Do not show report. |

| plot | Specifies the plot to be shown for overall evaluation. It has three options: 'auc' shows the ROC curve, 'ks' shows the cumulative distribution of the actual class and its maximum difference (KS Statistic), and 'none' (Default). |
|------|---|
| returndf | Option for the user to save the data frame behind the metrics. 1: Show data frame, 0: Do not show (Default). |

## Value

The command `smbinning.metrics` returns a report with classic performance metrics of a classification model.

## Examples

```
# Load library and its dataset
library(smbinning) # Load package and its data

# Example: Metrics Credit Score 1
smbinning.metrics(dataset=smbsimdf1,prediction="cbs1",actualclass="fgood",
                   report=1) # Show report
smbinning.metrics(dataset=smbsimdf1,prediction="cbs1",actualclass="fgood",
                   cutoff=600, report=1) # User cutoff
smbinning.metrics(dataset=smbsimdf1,prediction="cbs1",actualclass="fgood",
                   report=0, plot="auc") # Plot AUC
smbinning.metrics(dataset=smbsimdf1,prediction="cbs1",actualclass="fgood",
                   report=0, plot="ks") # Plot KS

# Save table with all details of metrics
cbs1metrics=smbinning.metrics(
  dataset=smbsimdf1,prediction="cbs1",actualclass="fgood",
  report=0, returndf=1) # Save metrics details
```

---

smbinning.metrics.plot

*Visualization of a Classification Matrix*

---

## Description

It generates four plots after running and saving the output report from `smbinning.metrics`.

## Usage

```
smbinning.metrics.plot(df, cutoff = NA, plot = "cmactual")
```

## Arguments

| df | Data frame generated with `smbinning.metrics`. |
|------|---|
| cutoff | Value of the classifier that splits the data between positive (>=) and negative (<). |
| plot | Plot to be drawn. Options are: 'cmactual' (default),'cmactualrates','cmmodel','cmmodelrates'. |

## Examples

```
# Load library and its dataset
library(smbinning)
smbmetricsdf=smbinning.metrics(dataset=smbsimdf1, prediction="cbs1",
                               actualclass="fgood", returndf=1)

# Example 1: Plots based on optimal cutoff
smbinning.metrics.plot(df=smbmetricsdf,plot='cmactual')

# Example 2: Plots using user defined cutoff
smbinning.metrics.plot(df=smbmetricsdf,cutoff=600,plot='cmactual')
smbinning.metrics.plot(df=smbmetricsdf,cutoff=600,plot='cmactualrates')
smbinning.metrics.plot(df=smbmetricsdf,cutoff=600,plot='cmmodel')
smbinning.metrics.plot(df=smbmetricsdf,cutoff=600,plot='cmmodelrates')
```

---

smbinning.monotonic  *Monotonic Binning*

---

## Description

It gives the user the ability to impose a monotonic trend for good/bad rates per bin.

## Usage

```
smbinning.monotonic(df, y, x, p = 0.05)
```

## Arguments

| | |
|---|---|
| df | A data frame. |
| y | Binary response variable (0,1). Integer (int) is required. Name of y must not have a dot. Name "default" is not allowed. |
| x | Continuous characteristic. At least 5 different values. Value Inf is not allowed. Name of x must not have a dot. |
| p | Percentage of records per bin. Default 5% (0.05). |

## Value

The command smbinning.monotonic generates and object containing the necessary info and utilities for binning. The user should save the output result so it can be used with smbinning.plot, smbinning.sql, and smbinning.gen.

## Examples

```
# Load library and its dataset
library(smbinning) # Load package and its data

# Example 1: Monotonic Binning (Increasing Good Rate per Bin)
smbinning(df=smbsimdf2,y="fgood2",x="chr2",p=0.05)$ivtable # Run regular binning
```

```
smbinning.monotonic(df=smbsimdf2,y="fgood2",x="chr2",p=0.05)$ivtable # Run monotonic binning

# Example 2: Monotonic Binning (Decreasing Good Rate per Bin)
smbinning(df=smbsimdf2,y="fgood3",x="chr3",p=0.05)$ivtable # Run regular binning
smbinning.monotonic(df=smbsimdf2,y="fgood3",x="chr3",p=0.05)$ivtable # Run monotonic binning
```

---

smbinning.plot              *Plots after binning*

---

### Description

It generates plots for distribution, bad rate, and weight of evidence after running smbinning and saving its output.

### Usage

```
smbinning.plot(ivout, option = "dist", sub = "")
```

### Arguments

| | |
|---|---|
| ivout | An object generated by binning. |
| option | Distribution ("dist"), Good Rate ("goodrate"), Bad Rate ("badrate"), and Weight of Evidence ("WoE"). |
| sub | Subtitle for the chart (optional). |

### Examples

```
# Load library and its dataset
library(smbinning)

# Example 1: Numeric variable (1 page, 4 plots)
result=smbinning(df=smbsimdf1,y="fgood",x="cbs1") # Run and save result
par(mfrow=c(2,2))
boxplot(smbsimdf1$cbs1~smbsimdf1$fgood,
        horizontal=TRUE, frame=FALSE, col="lightgray",main="Distribution")
mtext("Credit Score",3)
smbinning.plot(result,option="dist",sub="Credit Score")
smbinning.plot(result,option="badrate",sub="Credit Score")
smbinning.plot(result,option="WoE",sub="Credit Score")
par(mfrow=c(1,1))

# Example 2: Factor variable (1 plot per page)
result=smbinning.factor(df=smbsimdf1,y="fgood",x="inc",maxcat=11)
smbinning.plot(result,option="dist",sub="Income Level")
smbinning.plot(result,option="badrate",sub="Income Level")
smbinning.plot(result,option="WoE",sub="Income Level")
```

---

smbinning.psi *Population Stability Index*

---

### Description

Often models are developed using multiple periods in time for a number of reasons. For example, to avoid seasonality, to increase the size of the population, and some others. With a metrics like the Population Stability Index (PSI), users can check if there is a significant variation in the distribution of a certain feature by partition (usually time) using the first one as the reference.

### Usage

```
smbinning.psi(df, y, x)
```

### Arguments

| | |
|---|---|
| df | Data frame. |
| y | Column name the indicates the different partitions. |
| x | Feature to be evaluated in terms of stability (It must be factor). |

### Value

Three crosstabs by feature and period that show the frequency (psicnt), percentage (psipct) and PSI (psimg), and a plot for the analyzed characteristic.

### Examples

```
# Load library and its dataset
library(smbinning)

# Check stability for income
smbinning.psi(df=smbsimdf1,y="period",x="inc")
```

---

smbinning.scaling *Scaling*

---

### Description

It transforms the coefficients of a logistic regression into scaled points based on the following three parameters pre-selected by the analyst: PDO, Score, and Odds.

### Usage

```
smbinning.scaling(logitraw, pdo = 20, score = 720, odds = 99)
```

**Arguments**

| | |
|---|---|
| `logitraw` | Logistic regression (glm) that must have specified `family=binomial` and whose variables have been generated with `smbinning.gen` or `smbinning.factor.gen`. |
| `pdo` | Points to double the oods. |
| `score` | Score at which the desire odds occur. |
| `odds` | Desired odds at the selected `score`. |

**Value**

A scaled model from a logistic regression built with binned variables, the parameters used in the scaling process, the expected minimum and maximum score, and the original logistic model.

**Examples**

```
# Load library and its dataset
library(smbinning)

# Sampling
pop=smbsimdf1 # Population
train=subset(pop,rnd<=0.7) # Training sample

# Generate binning object to generate variables
smbcbs1=smbinning(train,x="cbs1",y="fgood")
smbcbinq=smbinning.factor(train,x="cbinq",y="fgood")
smbcblineut=smbinning.custom(train,x="cblineut",y="fgood",cuts=c(30,40,50))
smbpmt=smbinning.factor(train,x="pmt",y="fgood")
smbtob=smbinning.custom(train,x="tob",y="fgood",cuts=c(1,2,3))
smbdpd=smbinning.factor(train,x="dpd",y="fgood")
smbdep=smbinning.custom(train,x="dep",y="fgood",cuts=c(10000,12000,15000))
smbod=smbinning.factor(train,x="od",y="fgood")
smbhome=smbinning.factor(train,x="home",y="fgood")
smbinc=smbinning.factor.custom(
  train,x="inc",y="fgood",
  c("'W01','W02'","'W03','W04','W05'","'W06','W07'","'W08','W09','W10'"))

pop=smbinning.gen(pop,smbcbs1,"g1cbs1")
pop=smbinning.factor.gen(pop,smbcbinq,"g1cbinq")
pop=smbinning.gen(pop,smbcblineut,"g1cblineut")
pop=smbinning.factor.gen(pop,smbpmt,"g1pmt")
pop=smbinning.gen(pop,smbtob,"g1tob")
pop=smbinning.factor.gen(pop,smbdpd,"g1dpd")
pop=smbinning.gen(pop,smbdep,"g1dep")
pop=smbinning.factor.gen(pop,smbod,"g1od")
pop=smbinning.factor.gen(pop,smbhome,"g1home")
pop=smbinning.factor.gen(pop,smbinc,"g1inc")

# Resample
train=subset(pop,rnd<=0.7) # Training sample
test=subset(pop,rnd>0.7) # Testing sample

# Run logistic regression
```

```
f=fgood~g1cbs1+g1cbinq+g1cblineut+g1pmt+g1tob+g1dpd+g1dep+g1od+g1home+g1inc
modlogisticsmb=glm(f,data = train,family = binomial())
summary(modlogisticsmb)

# Example: Scaling from logistic parameters to points
smbscaled=smbinning.scaling(modlogisticsmb,pdo=20,score=720,odds=99)
smbscaled$logitscaled # Scaled model
smbscaled$minmaxscore # Expected minimum and maximum Score
smbscaled$parameters # Parameters used for scaling
summary(smbscaled$logitraw) # Extract of original logistic regression

# Example: Generate score from scaled model
pop1=smbinning.scoring.gen(smbscaled=smbscaled, dataset=pop)

# Example Generate SQL code from scaled model
smbinning.scoring.sql(smbscaled)
```

---

smbinning.scoring.gen   *Generation of Score and Its Weights*

---

## Description

After applying smbinning.scaling to the model, smbinning.scoring generates a data frame with the final Score and additional fields with the points assigned to each characteristic so the user can see how the final score is calculated. Example shown on smbinning.scaling section.

## Usage

```
smbinning.scoring.gen(smbscaled, dataset)
```

## Arguments

| | |
|---|---|
| smbscaled | Object generated using smbinning.scaling. |
| dataset | A data frame. |

## Value

The command smbinning.scoring generates a data frame with the final scaled Score and its corresponding scaled weights per characteristic.

---

smbinning.scoring.sql   *Generation of SQL Code After Scaled Model*

---

### Description

After applying smbinning.scaling to the model, smbinning.scoring.sql generates a SQL code
that creates and updates all variables present in the scaled model. Example shown on smbinning.scaling
section.

### Usage

```
smbinning.scoring.sql(smbscaled)
```

### Arguments

smbscaled          Object generated using smbinning.scaling.

### Value

The command smbinning.scoring.sql generates a SQL code to implement the model the model
in SQL.

---

smbinning.sql               *SQL Code*

---

### Description

It outputs a SQL code to facilitate the generation of new binned characetristic in a SQL environment.
User must define table and new characteristic name.

### Usage

```
smbinning.sql(ivout)
```

### Arguments

ivout              An object generated by smbinning.

### Value

A text with the SQL code for binning.

## Examples

```
# Load library and its dataset
library(smbinning)

# Example 1: Binning a numeric variable
result=smbinning(df=smbsimdf1,y="fgood",x="cbs1") # Run and save result
smbinning.sql(result)

# Example 2: Binning for a factor variable
result=smbinning.factor(df=smbsimdf1,x="inc",y="fgood",maxcat=11)
smbinning.sql(result)

# Example 3: Customized binning for a factor variable
result=smbinning.factor.custom(
  df=smbsimdf1,x="inc",y="fgood",
  c("'W01','W02'","'W03','W04','W05'",
    "'W06','W07'","'W08','W09','W10'"))
smbinning.sql(result)
```

---

smbinning.sumiv             *Information Value Summary*

---

## Description

It gives the user the ability to calculate, in one step, the IV for each characteristic of the dataset. This function also shows a progress bar so the user can see the status of the process.

## Usage

```
smbinning.sumiv(df, y)
```

## Arguments

| | |
|---|---|
| df | A data frame. |
| y | Binary response variable (0,1). Integer (`int`) is required. Name of y must not have a dot. Name "default" is not allowed. |

## Value

The command `smbinning.sumiv` generates a table that lists each characteristic with its corresponding IV for those where the calculation is possible, otherwise it will generate a missing value (NA).

## Examples

```
# Load library and its dataset
library(smbinning)

# Test sample
test=subset(smbsimdf1,rnd>0.9) # Training sample
```

```
test$rnd=NULL

# Example: Information Value Summary
testiv=smbinning.sumiv(test,y="fgood")
testiv

# Example: Plot of Information Value Summary
smbinning.sumiv.plot(testiv)
```

---

smbinning.sumiv.plot    *Plot Information Value Summary*

---

## Description

It gives the user the ability to plot the Information Value by characteristic. The chart only shows characteristics with a valid IV. Example shown on smbinning.sumiv section.

## Usage

```
smbinning.sumiv.plot(sumivt, cex = 0.9)
```

## Arguments

| | |
|---|---|
| sumivt | A data frame saved after smbinning.sumiv. |
| cex | Optional parameter for the user to control the font size of the characteristics displayed on the chart. The default value is 0.9 |

## Value

The command smbinning.sumiv.plot returns a plot that shows the IV for each numeric and factor characteristic in the dataset.

---

smbsimdf1                    *Simulated Credit Data*

---

## Description

A simulated dataset where the target variable is fgood, which represents the binary status of default (0) and not default (1).

## Format

Data frame with 2,500 rows and 22 columns with 500 defaults.

## Details

- fgood: Default (0), Not Default (1).
- cbs1: Credit quality index (1-100).
- cbs2: Profitability index (1-100).
- cbinq: Number of inquiries.
- cbline: Number of credit lines.
- cbterm: Number of term loans.
- cblineut: Line utilization (0-100).
- cbtob: Number of years on file.
- cbdpd: Indicator of days past due on bureau (Yes, No).
- cbnew: Number of new loans.
- pmt: Type of payment (M: Manual, A: Autopay, P: Payroll).
- tob: Time on books (Years).
- dpd: Level of delinquency (No, Low, High).
- dep: Amount of deposits.
- dc: Number of transactions.
- od: Number of overdrafts.
- home: Home ownership indicator (Yes, No).
- inc: Level of income.
- dd: Number of electronic transfers.
- online: Indicator of online activity (Yes, No).
- rnd: Random number to select testing and training samples.
- period: Factor that indicates the year/month of the data (Based on rnd).

---

smbsimdf2                    *Monotonic Binning Sample Data*

---

## Description

A simulated dataset used to illustrate the application of monotonic binning.

## Format

Data frame with 2,500 rows and 6 columns.

## Details

- fgood1: Default (0), Not Default (1) for Numeric Variable 1.
- chr1: Numeric variable 1.
- fgood2: Default (0), Not Default (1) for Numeric Variable 2.
- chr2: Numeric variable 2.
- fgood3: Default (0), Not Default (1) for Numeric Variable 3.
- chr3: Numeric variable 3.

---

smbsimdf3                                 *Monotonic Binning Sample Data*

---

## Description

A simulated dataset used to illustrate the application of model ranking.

## Format

Data frame with 1,000 rows and 4 columns.

## Details

- fgood1: Default (0), Not Default (1) for Numeric Variable 1.
- chr1: Numeric variable 1.
- chr2: Numeric variable 2.
- chr3: Numeric variable 3.

# Index