

# Package ‘mob’

October 13, 2022

**Title** Monotonic Optimal Binning

**Version** 0.4.2

**Description** Generate the monotonic binning and perform the woe (weight of evidence) transformation for the logistic regression used in the consumer credit scorecard development. The woe transformation is a piecewise transformation that is linear to the log odds. For a numeric variable, all of its monotonic functional transformations will converge to the same woe transformation.

**License** GPL (>= 2)

**URL** <https://github.com/statcompute/mob>

**Author** WenSui Liu

**Maintainer** WenSui Liu <liuwensui@gmail.com>

**Depends** R (>= 3.3.3)

**Imports** stats, gbm, Rborist

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-07-31 04:30:07 UTC

## R topics documented:

arb_bin . . . . .	2
bad_bin . . . . .	2
batch_bin . . . . .	3
batch_woe . . . . .	4
cal_woe . . . . .	4
gbm_bin . . . . .	5
hmeq . . . . .	6
iso_bin . . . . .	7
kmn_bin . . . . .	7

pool_bin . . . . .	8
qcut . . . . .	9
qtl_bin . . . . .	9
rng_bin . . . . .	10

<b>Index</b>	<b>11</b>
--------------	-----------

---

arb_bin	<i>Monotonic binning based on decision tree model</i>
---------	---

---

### Description

The function `arb_bin` implements the monotonic binning based on the decision tree.

### Usage

```
arb_bin(x, y)
```

### Arguments

<code>x</code>	A numeric vector
<code>y</code>	A numeric vector with 0/1 binary values

### Value

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

### Examples

```
data(hmeq)
arb_bin(hmeq$DEROG, hmeq$BAD)
```

---

bad_bin	<i>Monotonic binning by quantile with cases Y = 1</i>
---------	---

---

### Description

The function `bad_bin` implements the quantile-based monotonic binning by the iterative discretization based on cases with  $Y = 1$ .

### Usage

```
bad_bin(x, y)
```

**Arguments**

x                    A numeric vector  
 y                    A numeric vector with 0/1 binary values

**Value**

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

**Examples**

```
data(hmeq)
bad_bin(hmeq$DEROG, hmeq$BAD)
```

---

batch_bin	<i>Apply monotonic binning to all vectors in dataframe</i>
-----------	--

---

**Description**

The function batch\_bin applies multiple binning algorithms in batch to each vector in the dataframe.

**Usage**

```
batch_bin(y, xs, method = 1)
```

**Arguments**

y                    A numeric vector with 0/1 binary values.  
 xs                   A dataframe with numeric vectors to discretize.  
 method              A integer from 1 to 7 referring to implementations below: 1. Implementation of iso\_bin() 2. Implementation of qtl\_bin() 3. Implementation of bad\_bin() 4. Implementation of rng\_bin() 5. Implementation of gbm\_bin() 6. Implementation of kmn\_bin() 7. Implementation of arb\_bin()

**Value**

A list of binning outcomes with 2 dataframes: bin\_sum: A dataframe of binning summary. bin\_out: A list of binning output from binning functions, e.g. qtl\_bin().

**Examples**

```
data(hmeq)
batch_bin(hmeq$BAD, hmeq[, c('DEROG', 'DELINQ')])
```

---

batch_woe	<i>Apply WoE transformations to vectors in dataframe</i>
-----------	--

---

### Description

The function `batch_woe` applies WoE transformations to vectors in the dataframe.

### Usage

```
batch_woe(xs, bin_out)
```

### Arguments

<code>xs</code>	A dataframe with numeric vectors to discretize.
<code>bin_out</code>	A binning output from the function <code>batch_bin()</code> .

### Value

A dataframe with identical headers as the input `xs`. However, values of each variable have been transformed to WoE values.

### Examples

```
data(hmeq)
bin_out <- batch_bin(hmeq$BAD, hmeq[, c('DEROG', 'DELINQ')])$bin_out
head(batch_woe(hmeq[, c('DEROG', 'DELINQ')], bin_out))
```

---

cal_woe	<i>Perform WoE transformation of a numeric variable</i>
---------	---

---

### Description

The function `cal_woe` applies the WoE transformation to a numeric vector based on the binning outcome from a binning function, e.g. `qtl_bin()` or `iso_bin()`.

### Usage

```
cal_woe(x, bin)
```

### Arguments

<code>x</code>	A numeric vector that will be transformed to WoE values.
<code>bin</code>	A list with the binning outcome from the binning function, e.g. <code>qtl_bin()</code> or <code>iso_bin()</code>

**Value**

A numeric vector with WoE transformed values.

**Examples**

```
data(hmeq)
bin_out <- qtl_bin(hmeq$DEROG, hmeq$BAD)
cal_woe(hmeq$DEROG[1:10], bin_out)
```

---

gbm\_bin

*Monotonic binning based on generalized boosted model*

---

**Description**

The function `gbm_bin` implements the monotonic binning based on the generalized boosted model (GBM).

**Usage**

```
gbm_bin(x, y)
```

**Arguments**

`x` A numeric vector

`y` A numeric vector with 0/1 binary values

**Value**

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

**Examples**

```
data(hmeq)
gbm_bin(hmeq$DEROG, hmeq$BAD)
```

---

hmeq

*Credit attributes of 5,960 home equity loans*

---

### **Description**

A dataset containing characteristics and delinquency information for 5,960 home equity loans.

### **Usage**

hmeq

### **Format**

A data frame with 5960 rows and 13 variables:

**BAD** indicator of applicant defaulted on loan or seriously delinquent

**LOAN** Amount of the loan request, in dollar

**MORTDUE** Amount due on existing mortgage, in dollar

**VALUE** Value of current property, in dollar

**REASON** DebtCon = debt consolidation; HomeImp = home improvement

**JOB** Occupational categories

**YOJ** Years at present job

**DEROG** Number of major derogatory reports

**DELINQ** Number of delinquent credit lines

**CLAGE** Age of oldest credit line in months

**NINQ** Number of recent credit inquiries

**CLNO** Number of credit lines

**DEBTINC** Debt-to-income ratio

### **Source**

<http://www.creditriskanalytics.net/datasets-private2.html>

---

iso_bin	<i>Monotonic binning based on isotonic regression</i>
---------	---

---

**Description**

The function `iso_bin` implements the monotonic binning based on the isotonic regression.

**Usage**

```
iso_bin(x, y)
```

**Arguments**

<code>x</code>	A numeric vector
<code>y</code>	A numeric vector with 0/1 binary values

**Value**

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

**Examples**

```
data(hmeq)
iso_bin(hmeq$DEROG, hmeq$BAD)
```

---

kmn_bin	<i>Monotonic binning based on k-means clustering</i>
---------	--

---

**Description**

The function `kmn_bin` implements the monotonic binning based on the k-means clustering

**Usage**

```
kmn_bin(x, y)
```

**Arguments**

<code>x</code>	A numeric vector
<code>y</code>	A numeric vector with 0/1 binary values

**Value**

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

**Examples**

```
data(hmeq)
knn_bin(hmeq$DEROG, hmeq$BAD)
```

---

pool_bin	<i>Monotonic binning for the pool data</i>
----------	--

---

**Description**

The function `pool_bin` implements the monotonic binning for the pool data based on the generalized boosted model (GBM).

**Usage**

```
pool_bin(x, num, den, log = FALSE)
```

**Arguments**

<code>x</code>	A numeric vector
<code>num</code>	A numeric vector with integer values for numerators to calculate bad rates
<code>den</code>	A numeric vector with integer values for denominators to calculate bad rates
<code>log</code>	A logical constant either TRUE or FALSE. The default is FALSE

**Value**

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

**Examples**

```
data(hmeq)
df <- rbind(Reduce(rbind,
  lapply(split(hmeq, floor(hmeq$CLAGE)),
    function(d) data.frame(AGE = unique(floor(d$CLAGE)),
                          NUM = sum(d$BAD),
                          DEN = nrow(d)))),
  data.frame(AGE = NA,
             NUM = sum(hmeq[is.na(hmeq$CLAGE), ]$BAD),
             DEN = nrow(hmeq[is.na(hmeq$CLAGE), ])))
pool_bin(df$AGE, df$NUM, df$DEN, log = TRUE)
```



---

qcut	<i>Discretizing a numeric vector</i>
------	--------------------------------------

---

**Description**

The function `qcut` discretizes a numeric vector into `N` pieces based on quantiles.

**Usage**

```
qcut(x, n)
```

**Arguments**

<code>x</code>	A numeric vector.
<code>n</code>	An integer indicating the number of categories to discretize.

**Value**

A numeric vector to divide the vector `x` into `n` categories.

**Examples**

```
x <- 1:10
# [1] 1 2 3 4 5 6 7 8 9 10
v <- qcut(1:10, 4)
# [1] 3 5 8
findInterval(x, sort(c(v, -Inf, Inf)), left.open = TRUE)
# [1] 1 1 1 2 2 3 3 3 4 4
```

---

qtl_bin	<i>Monotonic binning by quantile</i>
---------	--------------------------------------

---

**Description**

The function `qtl_bin` implements the quantile-based monotonic binning by the iterative discretization

**Usage**

```
qtl_bin(x, y)
```

**Arguments**

<code>x</code>	A numeric vector
<code>y</code>	A numeric vector with 0/1 binary values

**Value**

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

**Examples**

```
data(hmeq)
qtl_bin(hmeq$DEROG, hmeq$BAD)
```

---

rng\_bin

*Monotonic binning by quantile based on value range*

---

**Description**

The function `rng_bin` implements the quantile-based monotonic binning by the iterative discretization based on the equal-width range of values.

**Usage**

```
rng_bin(x, y)
```

**Arguments**

`x`                    A numeric vector  
`y`                    A numeric vector with 0/1 binary values

**Value**

A list of binning outcomes, including a numeric vector with cut points and a dataframe with binning summary

**Examples**

```
data(hmeq)
rng_bin(hmeq$DEROG, hmeq$BAD)
```

# Index

## \* datasets

hmeq, 6

arb\_bin, 2

bad\_bin, 2

batch\_bin, 3

batch\_woe, 4

cal\_woe, 4

gbm\_bin, 5

hmeq, 6

iso\_bin, 7

knn\_bin, 7

pool\_bin, 8

qcut, 9

qtl\_bin, 9

rng\_bin, 10