Package ‘monobin’

October 13, 2022

Title Monotonic Binning for Credit Rating Models

Version 0.2.4

Maintainer Andrija Djurovic <djandrija@gmail.com>

Description

Performs monotonic binning of numeric risk factor in credit rating models (PD, LGD, EAD) development. All functions handle both binary and continuous target variable.
Functions that use isotonic regression in the first stage of binning process have an additional feature for correction of minimum percentage of observations and minimum target rate per bin.
Additionally, monotonic trend can be identified based on raw data or, if known in advance, forced by functions' argument. Missing values and other possible special values are treated separately from so-called complete cases.

License GPL (>= 3)

URL https://github.com/andrija-djurovic/monobin

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends dplyr, Hmisc, R (>= 2.10)

NeedsCompilation no

Author Andrija Djurovic [aut, cre]

Repository CRAN

Date/Publication 2022-07-21 09:30:08 UTC

R topics documented:

`cum.bin` *Monotonic binning based on maximum cumulative target rate (MAPA)*

Description

`cum.bin` implements monotonic binning based on maximum cumulative target rate. This algorithm is known as MAPA (Monotone Adjacent Pooling Algorithm).

Usage

```
cum.bin(
  x,
  y,
  sc = c(NA, NaN, Inf, -Inf),
  sc.method = "together",
  g = 15,
  y.type = NA,
  force.trend = NA
)
```

Arguments

`x` Numeric vector to be binned.

`y` Numeric target vector (binary or continuous).

`sc` Numeric vector with special case elements. Default values are `c(NA, NaN, Inf, -Inf)`. Recommendation is to keep the default values always and add new ones if needed. Otherwise, if these values exist in `x` and are not defined in the `sc` vector, function will report the error.

`sc.method` Define how special cases will be treated, all together or in separate bins. Possible values are `"together"`, `"separately"`.

`g` Number of starting groups. Default is 15.

`y.type` Type of `y`, possible options are `"bina"` (binary) and `"cont"` (continuous). If default value (`NA`) is passed, then algorithm will identify if `y` is 0/1 or continuous variable.

`force.trend` If the expected trend should be forced. Possible values: `"i"` for increasing trend (`y` increases with increase of `x`), `"d"` for decreasing trend (`y` decreases with decrease of `x`). Default value is `NA`. If the default value is passed, then trend will be identified based on the sign of the Spearman correlation coefficient between `x` and `y` on complete cases.

Value

The command `cum.bin` generates a list of two objects. The first object, data frame `summary.tbl` presents a summary table of final binning, while `x.trans` is a vector of discretized values. In case of single unique value for `x` or `y` in complete cases (cases different than special cases), it will return data frame with info.

Examples

```
suppressMessages(library(monobin))
data(gcd)
amount.bin <- cum.bin(x = gcd$amount, y = gcd$qual)
amount.bin[[1]]
gcd$amount.bin <- amount.bin[[2]]
gcd %>% group_by(amount.bin) %>% summarise(n = n(), y.avg = mean(qual))
#increase default number of groups (g = 20)
amount.bin.1 <- cum.bin(x = gcd$amount, y = gcd$qual, g = 20)
amount.bin.1[[1]]
#force trend to decreasing
cum.bin(x = gcd$amount, y = gcd$qual, g = 20, force.trend = "d")[[1]]
```

gcd *Excerpt from German Credit Data*

Description

The German Credit Data contains data on 20 variables and the classification whether an applicant is considered a Good or a Bad credit risk for 1000 loan applicants. Only 3 numeric variables are extracted (Duration of Credit (month), Credit Amount and Age (years)) along with good/bad indicator (Creditability) and renamed as: qual (Creditability), maturity (Duration of Credit (month)), age (Age (years)), amount (Credit Amount).

Usage

```
gcd
```

Format

An object of class `data.frame` with 1000 rows and 4 columns.

Source

https://online.stat.psu.edu/stat857/node/215/

iso.bin *Three-stage monotonic binning procedure*

Description

`iso.bin` implements three-stage monotonic binning procedure. The first stage is isotonic regression used to achieve the monotonicity, while the remaining two stages are possible corrections for minimum percentage of observations and target rate.

Usage

```
iso.bin(
  x,
  y,
  sc = c(NA, NaN, Inf, -Inf),
  sc.method = "together",
  y.type = NA,
  min.pct.obs = 0.05,
  min.avg.rate = 0.01,
  force.trend = NA
)
```

Arguments

`x` Numeric vector to be binned.

`y` Numeric target vector (binary or continuous).

`sc` Numeric vector with special case elements. Default values are `c(NA, NaN, Inf, -Inf)`. Recommendation is to keep the default values always and add new ones if needed. Otherwise, if these values exist in `x` and are not defined in the `sc` vector, function will report the error.

`sc.method` Define how special cases will be treated, all together or in separate bins. Possible values are `"together"`, `"separately"`.

`y.type` Type of `y`, possible options are `"bina"` (binary) and `"cont"` (continuous). If default value (`NA`) is passed, then algorithm will identify if `y` is 0/1 or continuous variable.

`min.pct.obs` Minimum percentage of observations per bin. Default is 0.05 or minimum 30 observations.

`min.avg.rate` Minimum `y` average rate. Default is 0.01 or minimum 1 bad case for `y` 0/1.

`force.trend` If the expected trend should be forced. Possible values: `"i"` for increasing trend (`y` increases with increase of `x`), `"d"` for decreasing trend (`y` decreases with decrease of `x`). Default value is `NA`. If the default value is passed, then trend will be identified based on the sign of the Spearman correlation coefficient between `x` and `y` on complete cases.

Details

The corrections of isotonic regression results present an important step in credit rating model development. The minimum percentage of observation is capped to minimum 30 observations per bin, while target rate for binary target is capped to 1 bad case.

Value

The command `iso.bin` generates a list of two objects. The first object, data frame `summary.tbl` presents a summary table of final binning, while `x.trans` is a vector of discretized values. In case of single unique value for `x` or `y` of complete cases (cases different than special cases), it will return data frame with info.

Examples

```
suppressMessages(library(monobin))
data(gcd)
age.bin <- iso.bin(x = gcd$age, y = gcd$qual)
age.bin[[1]]
table(age.bin[[2]])
# force increasing trend
iso.bin(x = gcd$age, y = gcd$qual, force.trend = "i")[[1]]

#stage by stage example
#inputs
x <- gcd$age #risk factor
y <- gcd$qual #binary dependent variable
min.pct.obs <- 0.05 #minimum percentage of observations per bin
min.avg.rate <- 0.01 #minimum percentage of defaults per bin
#stage 1: isotonic regression
db <- data.frame(x, y)
db <- db[order(db$x), ]
cc.sign <- sign(cor(db$y, db$x, method = "spearman", use = "complete.obs"))
iso.r <- isoreg(x = db$x, y = cc.sign * db$y)
db$y.hat <- iso.r$yf
db.s0 <- db %>%
   group_by(bin = y.hat) %>%
   summarise(no = n(),
 y.sum = sum(y),
 y.avg = mean(y),
 x.avg = mean(x),
 x.min = min(x),
 x.max = max(x))
db.s0
#stage 2: merging based on minimum percentage of observations
db.s1 <- db.s0
thr.no <- ceiling(ifelse(nrow(db) * min.pct.obs < 30, 30, nrow(db) * min.pct.obs))
thr.no #threshold for minimum number of observations per bin
repeat {
 if (nrow(db.s1) == 1) {break}
 values <- db.s1[, "no"]
 if (all(values >= thr.no)) {break}
 gap <- min(which(values < thr.no))
 if (gap == nrow(db.s1)) {
db.s1$bin[(gap - 1):gap] <- db.s1$bin[(gap - 1)]
} else {
db.s1$bin[gap:(gap + 1)] <- db.s1$bin[gap + 1]
}
 db.s1 <- db.s1 %>%
    group_by(bin) %>%
    mutate(
y.avg = weighted.mean(y.avg, no),
x.avg = weighted.mean(x.avg, no)) %>%
    summarise(
no = sum(no),
y.sum = sum(y.sum),
y.avg = unique(y.avg),
x.avg = unique(x.avg),
x.min = min(x.min),
x.max = max(x.max))
}
db.s1
#stage 3: merging based on minimum percentage of bad cases
db.s2 <- db.s1
thr.nb <- ceiling(ifelse(nrow(db) * min.avg.rate < 1, 1, nrow(db) * min.avg.rate))
thr.nb #threshold for minimum number of observations per bin
#already each bin has more bad cases than selected threshold hence no need for further merging
all(db.s2$y.sum > thr.nb)
#final result
db.s2
#result of the iso.bin function (formatting and certain metrics has been added)
iso.bin(x = gcd$age, y = gcd$qual)[[1]]
```

mdt.bin *Monotonic binning driven by decision tree*

Description

`mdt.bin` implements monotonic binning driven by decision tree. As a splitting metric for continuous target algorithm uses sum of squared errors, while for the binary target Gini index is used.

Usage

```
mdt.bin(
  x,
  y,
  g = 50,
  sc = c(NA, NaN, Inf, -Inf),
  sc.method = "together",
  y.type = NA,
  min.pct.obs = 0.05,
  min.avg.rate = 0.01,
  force.trend = NA
)
```

Arguments

`x` Numeric vector to be binned.

`y` Numeric target vector (binary or continuous).

`g` Number of splitting groups for each node. Default is 50.

`sc` Numeric vector with special case elements. Default values are `c(NA, NaN, Inf, -Inf)`. Recommendation is to keep the default values always and add new ones if needed. Otherwise, if these values exist in `x` and are not defined in the `sc` vector, function will report the error.

`sc.method` Define how special cases will be treated, all together or in separate bins. Possible values are `"together"`, `"separately"`.

`y.type` Type of `y`, possible options are `"bina"` (binary) and `"cont"` (continuous). If default value (`NA`) is passed, then algorithm will identify if `y` is 0/1 or continuous variable.

`min.pct.obs` Minimum percentage of observations per bin. Default is 0.05 or minimum 30 observations.

`min.avg.rate` Minimum `y` average rate. Default is 0.01 or minimum 1 bad case for y 0/1.

`force.trend` If the expected trend should be forced. Possible values: `"i"` for increasing trend (`y` increases with increase of `x`), `"d"` for decreasing trend (`y` decreases with decrease of `x`). Default value is `NA`. If the default value is passed, then trend will be identified based on the sign of the Spearman correlation coefficient between `x` and `y` on complete cases.

Value

The command `mdt.bin` generates a list of two objects. The first object, data frame `summary.tbl` presents a summary table of final binning, while `x.trans` is a vector of discretized values. In case of single unique value for `x` or `y` in complete cases (cases different than special cases), it will return data frame with info.

Examples

```
suppressMessages(library(monobin))
data(gcd)
amt.bin <- mdt.bin(x = gcd$amount, y = gcd$qual)
amt.bin[[1]]
table(amt.bin[[2]])
#force decreasing trend
mdt.bin(x = gcd$amount, y = gcd$qual, force.trend = "d")[[1]]
```

`ndr.bin` *Four-stage monotonic binning procedure including regression with nested dummies*

Description

`ndr.bin` implements extension of three-stage monotonic binning procedure (`iso.bin`) with step of regression with nested dummies as fourth stage. The first stage is isotonic regression used to achieve the monotonicity. The next two stages are possible corrections for minimum percentage of observations and target rate, while the last regression stage is used to identify statistically significant cut points.

Usage

```
ndr.bin(
  x,
  y,
  sc = c(NA, NaN, Inf, -Inf),
  sc.method = "together",
  y.type = NA,
  min.pct.obs = 0.05,
  min.avg.rate = 0.01,
  p.val = 0.05,
  force.trend = NA
)
```

Arguments

`x` Numeric vector to be binned.

`y` Numeric target vector (binary or continuous).

`sc` Numeric vector with special case elements. Default values are `c(NA, NaN, Inf, -Inf)`. Recommendation is to keep the default values always and add new ones if needed. Otherwise, if these values exist in `x` and are not defined in the `sc` vector, function will report the error.

`sc.method` Define how special cases will be treated, all together or separately. Possible values are `"together"`, `"separately"`.

`y.type` Type of `y`, possible options are `"bina"` (binary) and `"cont"` (continuous). If default value is passed, then algorithm will identify if `y` is 0/1 or continuous variable.

`min.pct.obs` Minimum percentage of observations per bin. Default is 0.05 or 30 observations.

`min.avg.rate` Minimum `y` average rate. Default is 0.05 or 30 observations.

`p.val` Threshold for p-value of regression coefficients. Default is 0.05. For a binary target binary logistic regression is estimated, whereas for a continuous target, linear regression is used.

`force.trend` If the expected trend should be forced. Possible values: `"i"` for increasing trend (`y` increases with increase of `x`), `"d"` for decreasing trend (`y` decreases with decrease of `x`). Default value is `NA`. If the default value is passed, then trend will be identified based on the sign of the Spearman correlation coefficient between `x` and `y` on complete cases.

Value

The command `ndr.bin` generates a list of two objects. The first object, data frame `summary.tbl` presents a summary table of final binning, while `x.trans` is a vector of discretized values. In case of single unique value for `x` or `y` of complete cases (cases different than special cases), it will return data frame with info.

See Also

`iso.bin` for three-stage monotonic binning procedure.

Examples

```
suppressMessages(library(monobin))
data(gcd)
age.bin <- ndr.bin(x = gcd$age, y = gcd$qual)
age.bin[[1]]
table(age.bin[[2]])
#linear regression example
amount.bin <- ndr.bin(x = gcd$amount, y = gcd$qual, y.type = "cont", p.val = 0.05)
#create nested dummies
db.reg <- gcd[, c("qual", "amount")]
db.reg$amount.bin <- amount.bin[[2]]
amt.s <- db.reg %>%
      group_by(amount.bin) %>%
      summarise(qual.mean = mean(qual),
    amt.min = min(amount))
mins <- amt.s$amt.min
for (i in 2:length(mins)) {
 level.l <- mins[i]
 nd <- ifelse(db.reg$amount < level.l, 0, 1)
 db.reg <- cbind.data.frame(db.reg, nd)
 names(db.reg)[ncol(db.reg)] <- paste0("dv_", i)
 }
reg.f <- paste0("qual ~ dv_2 + dv_3")
lrm <- lm(as.formula(reg.f), data = db.reg)
lr.coef <- data.frame(summary(lrm)$coefficients)
lr.coef
cumsum(lr.coef$Estimate)
#check
as.data.frame(amt.s)
diff(amt.s$qual.mean)
```

pct.bin *Monotonic binning based on percentiles*

Description

`pct.bin` implements percentile-based monotonic binning by the iterative discretization.

Usage

```
pct.bin(
  x,
  y,
  sc = c(NA, NaN, Inf, -Inf),
  sc.method = "together",
  g = 15,
  y.type = NA,
  woe.trend = TRUE,
  force.trend = NA
)
```

Arguments

x — Numeric vector to be binned.

y — Numeric target vector (binary or continuous).

sc — Numeric vector with special case elements. Default values are c(NA, NaN, Inf, -Inf). Recommendation is to keep the default values always and add new ones if needed. Otherwise, if these values exist in x and are not defined in the sc list some statistics cannot be calculated properly.

sc.method — Define how special cases will be treated, all together or in separate bins. Possible values are "together", "separately".

g — Number of starting groups. Default is 15.

y.type — Type of y, possible options are "bina" (binary) and "cont" (continuous). If default value is passed, then algorithm will identify if y is 0/1 or continuous variable.

woe.trend — Applied only for a continuous target (y) as weights of evidence (WoE) trend check. Default is TRUE.

force.trend — If the expected trend should be forced. Possible values: "i" for increasing trend (y increases with increase of x), "d" for decreasing trend (y decreases with decrease of x). Default value is NA. If the default value is passed, algorithm will stop if perfect negative or positive correlation (Spearman) is achieved between average y and average x per bin. Otherwise, it will stop only if the forced trend is achieved.

Value

The command pct.bin generates a list of two objects. The first object, data frame summary.tbl presents a summary table of final binning, while x.trans is a vector of discretized values. In case of single unique value for x or y of complete cases (cases different than special cases), it will return data frame with info.

Examples

```
suppressMessages(library(monobin))
data(gcd)
#binary target
mat.bin <- pct.bin(x = gcd$maturity, y = gcd$qual)
mat.bin[[1]]
table(mat.bin[[2]])
#continuous target, separate groups for special cases
set.seed(123)
gcd$age.d <- gcd$age
gcd$age.d[sample(1:nrow(gcd), 10)] <- NA
gcd$age.d[sample(1:nrow(gcd), 3)] <- 9999999999
age.d.bin <- pct.bin(x = gcd$age.d,
    y = gcd$qual,
    sc = c(NA, NaN, Inf, -Inf, 9999999999),
   sc.method = "separately",
    force.trend = "d")
age.d.bin[[1]]
gcd$age.d.bin <- age.d.bin[[2]]
gcd %>% group_by(age.d.bin) %>% summarise(n = n(), y.avg = mean(qual))
```

sts.bin *Four-stage monotonic binning procedure with statistical test correction*

Description

`sts.bin` implements extension of the three-stage monotonic binning procedure (`iso.bin`) with final step of iterative merging of adjacent bins based on statistical test.

Usage

```
sts.bin(
  x,
  y,
  sc = c(NA, NaN, Inf, -Inf),
  sc.method = "together",
  y.type = NA,
  min.pct.obs = 0.05,
  min.avg.rate = 0.01,
  p.val = 0.05,
  force.trend = NA
)
```

Arguments

`x` Numeric vector to be binned.

`y` Numeric target vector (binary or continuous).

`sc` Numeric vector with special case elements. Default values are `c(NA, NaN, Inf, -Inf)`. Recommendation is to keep the default values always and add new ones if needed. Otherwise, if these values exist in `x` and are not defined in the `sc` vector, function will report the error.

`sc.method` Define how special cases will be treated, all together or in separate bins. Possible values are `"together"`, `"separately"`.

`y.type` Type of `y`, possible options are `"bina"` (binary) and `"cont"` (continuous). If default value (`NA`) is passed, then algorithm will identify if `y` is 0/1 or continuous variable.

`min.pct.obs` Minimum percentage of observations per bin. Default is 0.05 or minimum 30 observations.

`min.avg.rate` Minimum `y` average rate. Default is 0.01 or minimum 1 bad case for `y` 0/1.

`p.val` Threshold for p-value of statistical test. Default is 0.05. For binary target test of two proportion is applied, while for continuous two samples independent t-test.

`force.trend` If the expected trend should be forced. Possible values: `"i"` for increasing trend (`y` increases with increase of `x`), `"d"` for decreasing trend (`y` decreases with decrease of `x`). Default value is `NA`. If the default value is passed, then trend will be identified based on the sign of the Spearman correlation coefficient between `x` and `y` on complete cases.

Value

The command `sts.bin` generates a list of two objects. The first object, data frame `summary.tbl` presents a summary table of final binning, while `x.trans` is a vector of discretized values. In case of single unique value for `x` or `y` of complete cases (cases different than special cases), it will return data frame with info.

See Also

`iso.bin` for three-stage monotonic binning procedure.

Examples

```
suppressMessages(library(monobin))
data(gcd)
#binary target
maturity.bin <- sts.bin(x = gcd$maturity, y = gcd$qual)
maturity.bin[[1]]
tapply(gcd$qual, maturity.bin[[2]], function(x) c(length(x), sum(x), mean(x)))
prop.test(x = c(sum(gcd$qual[maturity.bin[[2]]%in%"01 (-Inf,8)"]),
       sum(gcd$qual[maturity.bin[[2]]%in%"02 [8,16)"])),
       n = c(length(gcd$qual[maturity.bin[[2]]%in%"01 (-Inf,8)"]),
       length(gcd$qual[maturity.bin[[2]]%in%"02 [8,16)"])),
       alternative = "less",
       correct = FALSE)$p.value
#continuous target
age.bin <- sts.bin(x = gcd$age, y = gcd$qual, y.type = "cont")
age.bin[[1]]
t.test(x = gcd$qual[age.bin[[2]]%in%"01 (-Inf,26)"],
    y = gcd$qual[age.bin[[2]]%in%"02 [26,35)"],
    alternative = "greater")$p.value
```

woe.bin *Four-stage monotonic binning procedure with WoE threshold*

Description

`woe.bin` implements extension of the three-stage monotonic binning procedure (`iso.bin`) with weights of evidence (WoE) threshold. The first stage is isotonic regression used to achieve the monotonicity. The next two stages are possible corrections for minimum percentage of observations and target rate, while the last stage is iterative merging of bins until WoE threshold is exceeded.

Usage

```
woe.bin(
  x,
  y,
  sc = c(NA, NaN, Inf, -Inf),
  sc.method = "together",
  y.type = NA,
  min.pct.obs = 0.05,
  min.avg.rate = 0.01,
  woe.gap = 0.1,
  force.trend = NA
)
```

Arguments

`x` Numeric vector to be binned.

`y` Numeric target vector (binary or continuous).

`sc` Numeric vector with special case elements. Default values are `c(NA, NaN, Inf, -Inf)`. Recommendation is to keep the default values always and add new ones if needed. Otherwise, if these values exist in `x` and are not defined in the `sc` vector, function will report the error.

`sc.method` Define how special cases will be treated, all together or in separate bins. Possible values are `"together"`, `"separately"`.

`y.type` Type of `y`, possible options are `"bina"` (binary) and `"cont"` (continuous). If default value (`NA`) is passed, then algorithm will identify if `y` is 0/1 or continuous variable.

`min.pct.obs` Minimum percentage of observations per bin. Default is 0.05 or minimum 30 observations.

`min.avg.rate` Minimum `y` average rate. Default is 0.01 or minimum 1 bad case for y 0/1.

`woe.gap` Minimum WoE gap between bins. Default is 0.1.

`force.trend` If the expected trend should be forced. Possible values: `"i"` for increasing trend (`y` increases with increase of `x`), `"d"` for decreasing trend (`y` decreases with decrease of `x`). Default value is `NA`. If the default value is passed, then trend will be identified based on the sign of the Spearman correlation coefficient between `x` and `y` on complete cases.

Value

The command `woe.bin` generates a list of two objects. The first object, data frame `summary.tbl` presents a summary table of final binning, while `x.trans` is a vector of discretized values. In case of single unique value for x or y of complete cases (cases different than special cases), it will return data frame with info.

See Also

iso.bin for three-stage monotonic binning procedure.

Examples

```
suppressMessages(library(monobin))
data(gcd)
amount.bin <- woe.bin(x = gcd$amount, y = gcd$qual)
amount.bin[[1]]
diff(amount.bin[[1]]$woe)
tapply(gcd$amount, amount.bin[[2]], function(x) c(length(x), mean(x)))
woe.bin(x = gcd$maturity, y = gcd$qual)[[1]]
woe.bin(x = gcd$maturity, y = gcd$qual, woe.gap = 0.5)[[1]]
```

Index