Package ‘tidyvpc’

March 10, 2022

Type  Package
Title  VPC Percentiles and Prediction Intervals
Version  1.3.0
Date  2022-03-08

Description  Perform a Visual Predictive Check (VPC), while accounting for
stratification, censoring, and prediction correction. Using piping from
‘magrittr’, the intuitive syntax gives users a flexible and powerful method
to generate VPCs using both traditional binning and a new binless approach
Regression (AQR) and Locally Estimated Scatterplot Smoothing (LOESS)
prediction correction.

URL  https://github.com/certara/tidyvpc

BugReports  https://github.com/certara/tidyvpc/issues

Depends  R (>= 3.5.0),

Imports  data.table (>= 1.9.8), magrittr, quantreg (>= 5.51), rlang (>=
0.3.0), methods, mcv, classInt, ggplot2, stats, fastDummies

Suggests  cluster, dplyr, KernSmooth, knitr, shiny, remotes, vpc,
rmardown, testthat (>= 2.1.0)

License  MIT + file LICENSE

LazyData  true

Encoding  UTF-8

VignetteBuilder  knitr

RoxygenNote  7.1.2

NeedsCompilation  no

Author  Olivier Barriere [aut],
    Benjamin Rich [aut],
    James Craig [aut, cre],
    Samer Mouksassi [aut],
    Kris Jamsen [ctb],
    Certara USA, Inc. [cph, fnd]
Maintainer James Craig <james.craig@certara.com>
Repository CRAN
Date/Publication 2022-03-10 22:10:02 UTC

R topics documented:

bininfo ................................................. 2
binless .................................................. 3
binning .................................................. 6
binningfunctions ....................................... 8
censoring ............................................... 9
check_order .......................................... 10
generics .............................................. 11
nopredcorrect ......................................... 11
npde .................................................... 12
observed .............................................. 13
obs_cat_data .......................................... 14
obs_data .............................................. 15
plot.tidyvpcobj ....................................... 16
predcorrect ........................................... 17
print.tidyvpcobj ...................................... 18
simulated ............................................. 19
sim_cat_data .......................................... 20
sim_data ............................................. 20
stratify ............................................... 21
vpcstats .............................................. 22

Index 24

bininfo Obtain information about the bins from a tidyvpcobj

Description

Obtain information about the bins from a tidyvpcobj

Usage

bininfo(o, ...)

## S3 method for class 'tidyvpcobj'
bininfo(o, by.strata = o$bin.by.strata, ...)
Arguments

- `o` An object.
- `...` Additional arguments.

`by.strata` Should the calculations be done by strata? Defaults to what was specified when the binning was done.

Value

A `data.table` containing the following columns:

- `nobs`: Number of observed data points in the bin
- `xmedian`: Median x-value of the observed data points in the bin
- `xmean`: Mean x-value of the observed data points in the bin
- `xmax`: Maximum x-value of the observed data points in the bin
- `xmin`: Minimum x-value of the observed data points in the bin
- `xmid`: Value halfway between `xmin` and `xmax`. x-value of the observed data points in the bin
- `xleft`: Value halfway between the minimum x-value of the current bin and the maximum x-value of the previous bin to the left (for the left-most bin, it is the minimum x-value).
- `xright`: Value halfway between the maximum x-value of the current bin and the minimum x-value of the next bin to the right (for the right-most bin, it is the maximum x-value).
- `xcenter`: Value halfway between `xleft` and `xright`.

In addition, if stratification was performed, the stratification columns will be included as well.

Methods (by class)

- `tidyvpcobj`: Method for `tidyvpcobj`.

---

### binless

**Perform binless Visual Predictive Check (VPC)**

**Description**

Use this function in place of traditional binning methods to derive VPC. For continuous VPC, this is obtained using additive quantile regression (`quantreg::rqss()` and LOESS for pcVPC. While for categorical VPC, this is obtained using a generalized additive model (`gam(family = "binomial")`).
Usage

binless(o, ...)

## S3 method for class 'tidyvpcobj'
binless(
  o,
  optimize = TRUE,
  optimization.interval = c(0, 7),
  loess.ypc = FALSE,
  lambda = NULL,
  span = NULL,
  sp = NULL,
  ...
)

Arguments

o A tidyvpcobj.
...
otherwise arguments to include will be ignored.
optimize Logical indicating whether smoothing parameters should be optimized using AIC.
optimization.interval Numeric vector of length 2 specifying the min/max range of smoothing parameter for optimization. Only applicable if optimize = TRUE.
loess.ypc Logical indicating LOESS prediction corrected VPC. Must first use predcorrect, if specifying loess.ypc = TRUE. Only applicable to continuous VPC.
lambda Numeric vector of length 3 specifying lambda values for each quantile. If stratified, specify a data frame with given strata represented the column name, and value specified as a numeric vector of length 3. See below examples. Only applicable to continuous VPC with optimize = FALSE.
span Numeric between 0,1 specifying smoothing parameter for LOESS prediction correction. Only applicable for continuous VPC with loess.ypc = TRUE and optimize = FALSE.
sp List of smoothing parameters applied to mgcv::gam(). Elements of list must be in the same order as unique values of DV. If one or more stratification variables present, the order of sp should be specified as unique combination of strata + DV, in ascending order. See below examples. Only applicable for categorical VPC, if optimize = FALSE.

Value

For continuous VPC, updates tidyvpcobj with additive quantile regression fits for observed and simulated data for quantiles specified in the qpred argument of vpcstats(). If the optimize = TRUE argument is specified, the resulting tidyvpcobj will contain optimized lambda values according to AIC. For prediction corrected VPC (pcVPC), specifying loess.ypc = TRUE will return optimized span value for LOESS smoothing. For categorical VPC, updates tidyvpcobj with fits obtained by gam(family="binomial") for observed and simulated data for each category of DV.
binless

(in each stratum if stratify defined). If optimize = TRUE argument is specified, the resulting tidyvpcobj will contain optimized sp values according to AIC.

See Also

observed simulated censoring predcorrect stratify binning vpcstats

Examples

```r
require(magrittr)
require(data.table)

obs_data <- obs_data[MDV == 0]
sim_data <- sim_data[MDV == 0]

vpc <- observed(obs_data, y = DV, x = TIME) %>%
  simulated(sim_data, y = DV) %>%
  binless() %>%
  vpcstats()

# Binless example with LOESS prediction correction
obs_data$PRED <- sim_data[REP == 1, PRED]

vpc <- observed(obs_data, y = DV, x = TIME) %>%
  simulated(sim_data, y = DV) %>%
  predcorrect(pred = PRED) %>%
  binless(optimize = TRUE, loess.ypc = TRUE) %>%
  vpcstats()

# Binless example with user specified lambda values stratified on "GENDER" with 2 levels ("M", "F"), 10%, 50%, 90% quantiles.
lambda_strat <- data.table(  
  GENDER_M = c(3,5,2),  
  GENDER_F = c(1,3,4)
)

vpc <- observed(obs_data, y = DV, x = TIME) %>%
  simulated(sim_data, y = DV) %>%
  stratify(~ GENDER) %>%
  binless(optimize = FALSE, lambda = lambda_strat) %>%
  vpcstats(qpred = c(0.1, 0.5, 0.9))

# Binless example for categorical DV with optimized smoothing
vpc <- observed(obs_cat_data, x = agemonths, yobs = zlencat) %>%
  simulated(sim_cat_data, ysim = DV) %>%
  stratify(~ Country_ID_code) %>%
  binless() %>%
  vpcstats(vpc.type = "cat", quantile.type = 6)
```
# Binless example for categorical DV with user specified sp values
user_sp <- list(
    Country1_prob0 = 100,
    Country1_prob1 = 3,
    Country1_prob2 = 4,
    Country2_prob0 = 90,
    Country2_prob1 = 3,
    Country2_prob2 = 4,
    Country3_prob0 = 55,
    Country3_prob1 = 3,
    Country3_prob2 = 200)

vpc <- observed(obs_cat_data, x = agemonths, yobs = zlencat) %>%
    simulated(sim_cat_data, ysim = DV) %>%
    stratify(~ Country_ID_code) %>%
    binless(optimize = FALSE, sp = user_sp) %>%
    vpcstats(vpc.type = "categorical", conf.level = 0.9, quantile.type = 6)

---

**binning**

### Description

This function executes binning methods available in classInt i.e. "jenks", "kmeans", "sd", "pretty", "pam", "kmeans", "hclust", "bclust", "fisher", and "dpih". You may also bin directly on x-variable or alternatively specify "centers" or "breaks". For explanation of binning methods see classIntervals.

### Usage

```r
binning(o, ...)
```

## S3 method for class 'tidyvpcobj'

```r
binning(
    o,
    bin,
    data = o$data,
    xbin = "xmedian",
    centers,
    breaks,
    nbins,
    altx,
    stratum = NULL,
    by.strata = TRUE,
    ...
)
```
Arguments

- **o**: A tidyvpcobj.
- **...**: Other arguments to include.
- **bin**: Character string indicating binning method or unquoted variable name if binning on x-variable.
- **data**: Observed data supplied in `observed()` function.
- **xbin**: Character string indicating midpoint type for binning.
- **centers**: Numeric vector of centers for binning. Use `bin = "centers"`, if supplying centers.
- **breaks**: Numeric vector of breaks for binning. Use `bin = "breaks"`, if supplying breaks.
- **nbins**: Numeric number indicating the number of bins to use.
- **altx**: Unquoted variable name in observed data for alternative x-variable binning.
- **stratum**: List indicating the name of stratification variable and level, if using different binning methods by strata.
- **by.strata**: Logical indicating whether binning should be performed by strata.

Value

Updates tidyvpcobj with data.frame containing bin information including left/right boundaries and midpoint, as specified in xbin argument.

See Also

- `observed`
- `simulated`
- `censoring`
- `predcorrect`
- `stratify`
- `binless`
- `vpcstats`

Examples

```r
require(magrittr)

# Binning on x-variable NTIME
vpc <- observed(obs_data, x=TIME, y=DV) %>%
  simulated(sim_data, y=DV) %>%
  binning(bin = NTIME) %>%
  vpcstats()

# Binning using ntile and xmean for midpoint
vpc <- observed(obs_data, x=TIME, y=DV) %>%
  simulated(sim_data, y=DV) %>%
  binning(bin = "ntile", nbins = 8, xbin = "xmean") %>%
  vpcstats()

# Binning using centers
vpc <- observed(obs_data, x=TIME, y=DV) %>%
  simulated(sim_data, y=DV) %>%
  binning(bin = "centers", centers = c(1,3,5,7)) %>%
  vpcstats()
```
# Different Binning for each level of Strata

```r
vpc <- observed(obs_data, x=TIME, y=DV) %>%
simulated(sim_data, y=DV) %>%
stratify(~ GENDER) %>%
binning(stratum = list(GENDER = "M"), bin = "jenks", nbins = 5, by.strata = TRUE) %>%
binning(stratum = list(GENDER = "F"), bin = "pam", nbins = 4, by.strata = TRUE) %>%
vpcstats()
```

# Binning Categorical DV using rounded time variable

```r
vpc <- observed(obs_cat_data, x = agemonths, y = zlencat ) %>%
simulated(sim_cat_data, y = DV) %>%
binning(bin = round(agemonths, 0)) %>%
vpcstats(vpc.type = "categorical")
```

## binningfunctions

**Different functions that perform binning.**

### Description

Different functions that perform binning.

### Usage

- `cut_at(breaks)`
- `nearest(centers)`
- `bin_by_ntile(nbins)`
- `bin_by_eqcut(nbins)`
- `bin_by_pam(nbins)`
- `bin_by_classInt(style, nbins = NULL)`

### Arguments

- `breaks`: A numeric vector of values that designate cut points between bins.
- `centers`: A numeric vector of values that designate the center of each bin.
- `nbins`: The number of bins to split the data into.
- `style`: A binning style (see `classIntervals` for details).

### Value

Each of these functions returns a function of a single numeric vector ‘x’ that assigns each value of ‘x’ to a bin.
**Examples**

```r
x <- c(rnorm(10, 1, 1), rnorm(10, 3, 2), rnorm(20, 5, 3))
centers <- c(1, 3, 5)
nearest(c(centers))(x)

breaks <- c(2, 4)
cut_at(breaks)(x)

bin_by_eqcut(nbins=4)(x)
bin_by_ntile(nbins=4)(x)

bin_by_pam(nbins=4)(x)
bin_by_classInt("pretty", nbins=4)(x)
```

**censoring**

*Censoring observed data for Visual Predictive Check (VPC)*

**Description**

Specify censoring variable or censoring value for VPC.

**Usage**

```r
censoring(o, ...)
```

### S3 method for class `tidyvpcobj`

```r
censoring(o, blq, lloq, alq, uloq, data = o$data, ...)
```

**Arguments**

- `o` A `tidyvpcobj`.
- `...` Other arguments to include.
- `blq` `blq` variable if present in observed data.
- `lloq` Numeric value or numeric variable in data indicating the upper limit of quantification.
- `alq` Logical variable indicating above limit of quantification.
- `uloq` Numeric value or numeric variable in data indicating the upper limit of quantification.
- `data` Observed data supplied in `observed()` function.
check_order

Perform a consistency check on observed and simulated data

Description

This function performs a simple consistency check on an observed and simulated dataset to make sure they are consistent with respect to ordering as required by the other functions used in the VPC calculation.

Usage

check_order(obs, sim, tol = 1e-05)

Arguments

  obs, sim A `data.frame` with 2 columns (see Details).
  tol A tolerance for comparing time values.
Details

The consistency check is performed by comparing a combination of unique subject identifier (ID) and time. Both data.frame objects must be given with those in positions 1 and 2, respectively.

Value

The number of replicates contained in 'sim'.

See Also

observed, simulated.

Examples

```r
require(data.table)
check_order(obs_data[, .(ID, TIME)], sim_data[, .(ID, TIME)])
```

describe two functions to calculate the statistics that are plotted in a VPC. They would typically be chained together using the "pipe" operator (see Examples).

Arguments

- `o` A tidyvpcobj.
- `...` Additional arguments.

Description

Optional function to use indicating no pred correction for VPC.
Usage

npredcorrect(o, 

## S3 method for class 'tidyvpcobj'
npredcorrect(o, 

Arguments

o A tidyvpcobj.

... Other arguments to include.

npde Normalized Prediction Distribution Errors

Description

Normalized Prediction Distribution Errors

Usage

npde(o, 

## S3 method for class 'tidyvpcobj'
npde(o, id, data = o$data, smooth = FALSE, 

Arguments

o A tidyvpcobj.

... Additional arguments.

id A vector of IDs. Used to associate observations (y) that originate from the same individual. Evaluated in the data.frame data.

data A data.frame.

smooth Should a uniform random perturbation be used to smooth the pd/pde values?

References


Examples

```r
require(magrittr)
require(ggplot2)

obs <- obs_data[MDV==0]
sim <- sim_data[MDV==0]

npde <- observed(obs, x=NULL, y=DV) %>%
  simulated(sim, y=DV) %>%
  npde(id=ID)

vpc <- observed(npde$npdeobs, x=epred, y=npde) %>%
  simulated(npde$npdesim, y=npde) %>%
  binning("eqcut", nbins=10) %>%
  vpcstats()

plot(vpc) +
  labs(x="Simulation-based Population Prediction", y="Normalized Prediction Distribution Error")
```

---

**observed**

*Specify observed dataset and variables for VPC*

**Description**

The observed function is the first function in the vpc piping chain and is used for specifying observed data and variables for VPC. Note: Observed data must not contain missing DV and may require filtering MDV == 0 before generating VPC.

**Usage**

```r
observed(o, ...)
```

---

## S3 method for class 'data.frame'
observed(
  o,
  x,
  yobs,
  pred = NULL,
  blq = NULL,
  lloq = -Inf,
  alq = NULL,
  uloq = Inf,
  ...
)
```
Arguments

- **o**: A data.frame of observation data.
- **x**: Numeric x-variable, typically named TIME.
- **yobs**: Numeric y-variable, typically named DV.
- **pred**: Population prediction variable, typically named PRED.
- **blq**: Logical variable indicating below limit of quantification.
- **lloq**: Number or numeric variable in data indicating the lower limit of quantification.
- **alq**: Logical variable indicating above limit of quantification.
- **uloq**: Number or numeric variable in data indicating the upper limit of quantification.

Value

A tidyvpcobj containing both original data and observed data formatted with `x` and `y` variables as specified in function. Resulting data is of class data.frame and data.table.

See Also

simulated censoring stratify predcorrect binning binless vpcstats

Examples

```r
obs_data <- obs_data[MDV == 0]
sim_data <- sim_data[MDV == 0]

vpc <- observed(obs_data, x=TIME, y=DV)
```

---

### obs_cat_data

**Example observed data with categorical DV**

Description

An observed dataset with 3 levels of categorical DV.

Usage

obs_cat_data

Format

A data frame with 4014 rows and 4 variables:

- **PID_code**: Subject identifier
- **agemonths**: Time
- **zlencat**: Categorical DV with the 3 levels
- **Country_ID_code**: Country code for stratification
Example observed data with continuous DV

Source

Certara University

Description

An observed dataset from a hypothetical PK model, altered to include NTIME, GROUP, GENDER.

Usage

obs_data

Format

A data.table with 600 rows and 7 variables:

- **ID**: Subject identifier
- **TIME**: Time
- **DV**: Concentration of drug
- **AMT**: Amount of dosage initially administered at DV = 0, TIME = 0
- **DOSE**: Dosage amount
- **MDV**: Dummy indicating missing dependent variable value
- **NTIME**: Nominal Time
- **GENDER**: Character variable indicating subject’s gender ("M", "F")
- **STUDY**: Character variable indicating study type ("Study A", "Study B")

Source

simple_data
plot.tidyvpcobj  
Plot a tidyvpcobj

Description
Use ggplot2 graphics to plot and customize the appearance of VPC.

Usage

```r
## S3 method for class 'tidyvpcobj'
plot(
  x,
  facet = FALSE,
  show.points = TRUE,
  show.boundaries = TRUE,
  show.stats = !is.null(x$stats),
  show.binning = isFALSE(show.stats),
  xlab = NULL,
  ylab = NULL,
  color = c("red", "blue", "red"),
  linetype = c("dotted", "solid", "dashed"),
  point.alpha = 0.4,
  point.size = 1,
  point.shape = "circle-fill",
  point.stroke = 1,
  ribbon.alpha = 0.1,
  legend.position = "top",
  facet.scales = "free",
  custom.theme = "ggplot2::theme_bw",
  ...
)
```

Arguments

- `x` A tidyvpcobj.
- `facet` Set to TRUE to facet plot by quantile (continuous VPC) or category (categorical VPC).
- `show.points` Should the observed data points be plotted?
- `show.boundaries` Should the bin boundary be displayed?
- `show.stats` Should the VPC stats be displayed?
- `show.binning` Should the binning be displayed by coloring the observed data points by bin?
- `xlab` A character label for the x-axis.
- `ylab` A character label for the y-axis.
- `color` A character vector of colors for the percentiles, from low to high.
linetype: A character vector of line type for the percentiles, from low to high.
point.alpha: Numeric value specifying transparency of points.
point.size: Numeric value specifying size of point.
point.shape: Character one of "circle", "circle-fill", "diamond", "diamond-fill", "square", "square-fill", "triangle"). Defaults to "circle-fill".
point.stroke: Numeric value specifying size of point stroke.
ribbon.alpha: Numeric value specifying transparency of ribbon.
legend.position: A character string specifying the position of the legend. Options are "top", "bottom", "left", "right".
facet.scales: A character string specifying the scales argument to use for faceting. Options are "free", "fixed".
custom.theme: A character string specifying theme from ggplot2 package.
...: Further arguments can be specified but are ignored.

Value

A ggplot object.

See Also

ggplot

tidyvpcobj

Description

Specify prediction variable for pcVPC.

Usage

predcorrect(o, ...)

## S3 method for class 'tidyvpcobj'
predcorrect(o, pred, data = o$data, ..., log = FALSE)

Arguments

o: A tidyvpcobj.
...: Other arguments to include.
pred: Prediction variable in observed data.
data: Observed data supplied in observed() function.
log: Logical indicating whether DV was modeled in logarithmic scale.
Value

Updates tidyvpcobj with required information to performing prediction correction, which includes the predcor logical indicating whether prediction corrected VPC is to be performed, the predcor.log logical indicating whether the DV is on a log-scale, and the pred prediction column from the original data.

See Also

observed simulated censoring stratify binning binless vpcstats

Examples

```r
require(magrittr)

obs_data <- obs_data[MDV == 0]
sim_data <- sim_data[MDV == 0]

# Add PRED variable to observed data from first replicate of
# simulated data

obs_data$PRED <- sim_data[REP == 1, PRED]

vpc <- observed(obs_data, x=TIME, y=DV) %>%
  simulated(sim_data, y=DV) %>%
  binning(bin = NTIME) %>%
  predcorrect(pred=PRED) %>%
  vpcstats()

# For binless loess prediction corrected, use predcorrect() before
# binless() and set loess.ypc = TRUE

vpc <- observed(obs_data, x=TIME, y=DV) %>%
  simulated(sim_data, y=DV) %>%
  predcorrect(pred=PRED) %>%
  binless(loess.ypc = TRUE) %>%
  vpcstats()
```

Description

Print generic used to return information about VPC.
Usage

```r
## S3 method for class 'tidyvpcobj'
print(x, ...)
```

Arguments

- `x` An tidyvpcobj.
- `...` Further arguments can be specified but are ignored.

Value

Returns `x` invisibly.

---

**simulated**

*Specify simulated dataset and variables for VPC*

Description

The `simulated` function is used for specifying simulated input data and variables for VPC. Note: Simulated data must not contain missing DV and may require filtering `MDV == 0` before generating VPC. The ordering of observed and simulated data must also be consistent, with replicates in simulated data stacked on top of each other.

Usage

```r
simulated(o, ...)
```

## S3 method for class 'tidyvpcobj'

```r
simulated(o, data, ysim, ...)
```

Arguments

- `o` A tidyvpcobj.
- `...` Other arguments.
- `data` A `data.frame` of simulated data.
- `ysim` Numeric y-variable, typically named DV.

Value

A tidyvpcobj containing simulated dataset `sim` formatted with columns x, y, and repl, which indicates the replicate number. The column x is used from the observed() function. Resulting dataset is of class `data.frame` and `data.table`.

See Also

`observed censoring stratify predcorrect binning binless vpcstats`
Examples

```r
require(magrittr)

vpc <- observed(obs_data, x=TIME, y=DV) %>%
  simulated(sim_data, y=DV)
```

---

**sim_cat_data**  
*Example simulated data with categorical DV*

**Description**

A simulated dataset with the 3 levels of categorical DV across 100 replicates.

**Usage**

```r
sim_cat_data
```

**Format**

A data frame with 401400 rows and 4 variables:

- **PID_code** Subject identifier
- **IVAR** Time
- **DV** Categorical DV with 3 levels
- **Replicate** Replicate num for simulation

**Source**

Certara University

---

**sim_data**  
*Example simulated data with continuous DV*

**Description**

A simulated dataset from a hypothetical PK model with 100 replicates.

**Usage**

```r
sim_data
```
Stratification for Visual Predictive Check (VPC)

**Format**

A data.table with 60000 rows and 10 variables:

- **ID**  Subject identifier
- **REP** Replicate num for simulation
- **TIME**  Time
- **DV**  Concentration of drug
- **IPRED** Individual prediction variable
- **PRED** Population prediction variable
- **AMT** Amount of dosage initially administered at DV = 0, TIME = 0
- **DOSE** Dosage amount
- **MDV** Dummy indicating missing dependent variable value
- **NTIME** Nominal Time

**Source**

`simple_data`

**stratify**  

**Description**

Use to specify stratification variables for VPC.

**Usage**

```r
stratify(o, ...) 
```

## S3 method for class 'tidyvpcobj'

```r
stratify(o, formula, data = o$data, ...)
```

**Arguments**

- **o**  A tidyvpcobj.
- **...**  Other arguments to include.
- **formula**  Formula for stratification.
- **data**  Observed data supplied in observed() function.

**Value**

Returns updated tidyvpcobj with stratification formula, stratification column(s), and strat.split datasets, which is obs split by unique levels of stratification variable(s). Resulting datasets are of class object data.frame and data.table.
See Also

observed simulated censoring predcorrect binning binless vpcstats

Examples

```r
require(magrittr)

vpc <- observed(obs_data, x=TIME, y=DV) %>%
    simulated(sim_data, y=DV) %>%
    stratify(~ GENDER) %>%
    binning(NTIME) %>%
    vpcstats()

# Example with 2-way stratification by GENDER and STUDY.

vpc <- vpc %>%
    stratify(~ GENDER + STUDY) %>%
    binning(bin = "centers", centers = c(1,3,5,7,10)) %>%
    vpcstats()
```

---

vpcstats

Compute VPC statistics

Description

Compute prediction interval statistics for VPC.

Usage

vpcstats(o, ...)

## S3 method for class 'tidyvpcobj'

vpcstats(
  o,
  vpc.type = c("continuous", "categorical"),
  qpred = c(0.05, 0.5, 0.95),
  ..., conf.level = 0.95,
  quantile.type = 7
)

Arguments

- `o` A tidyvpcobj.
- `...` Other arguments to include.
- `vpc.type` Character specifying type of VPC (e.g., "continuous" (Default) or "categorical").
apred Numeric vector of length 3 specifying quantile prediction interval. Only applicable for `vpc.type = "continuous"`.

conf.level Numeric specifying confidence level.

quantile.type Numeric indicating quantile type. See `quantile`.

Value

Updates `tidyvpcobj` with `stats.data.table` object, which contains the following columns:

- `bin`: Resulting bin value as specified in `binning()` function
- `xbin`: Midpoint x-value of the observed data points in the bin as specified in `xbin` argument of `binning()` function
- `qname`: Quantiles specified in `apred`. Only returned if `vpc.type = "continuous"`
- `pname`: Categorical probability names. Only returned if `vpc.type = "categorical"`
- `y`: Observed y value for the specified quantile
- `lo`: Lower bound of specified confidence interval for y value in simulated data
- `md`: Median y value in simulated data
- `hi`: Upper bound of specified confidence interval for y value in simulated data

See Also

`observed simulated censoring stratify binning binless predcorrect`
Index

* datasets
  obs_cat_data, 14
  obs_data, 15
  sim_cat_data, 20
  sim_data, 20
bin_by_classInt (binningfunctions), 8
bin_by_eqcut (binningfunctions), 8
bin_by_ntile (binningfunctions), 8
bin_by_pam (binningfunctions), 8
bininfo, 2
binless, 3, 7, 10, 14, 18, 19, 22, 23
binning, 5, 6, 10, 14, 18, 19, 22, 23
binningfunctions, 8
censoring, 5, 7, 9, 14, 18, 19, 22, 23
check_order, 10
classIntervals, 6, 8
cut_at (binningfunctions), 8
generics, 11
nearest (binningfunctions), 8
nopredcorrect, 11
npde, 12
obs_cat_data, 14
obs_data, 15
observed, 5, 7, 10, 11, 13, 18, 19, 22, 23
plot.tidyvpcobj, 16
predcorrect, 4, 5, 7, 10, 14, 17, 19, 22, 23
print.tidyvpcobj, 18
quantile, 23
sim_cat_data, 20
sim_data, 20
simple_data, 15, 21
simulated, 5, 7, 10, 11, 14, 18, 19, 22, 23
stratify, 5, 7, 10, 14, 18, 19, 21, 23
vpcstats, 5, 7, 10, 14, 18, 19, 22, 22