Package ‘tidyvpc’

March 26, 2020

Type Package
Title VPC Percentiles and Prediction Intervals
Version 1.0.0
Date 2020-02-17
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 Jamsen et al. (2018) <doi:10.1002/psp4.12319> with Additive Quantile Regression (AQR) and Locally Estimated Scatterplot Smoothing (LOESS) prediction correction.

URL https://github.com/jameswcraig/tidyvpc
BugReports https://github.com/jameswcraig/tidyvpc/issues
Depends R (>= 3.5.0), data.table (>= 1.9.8), magrittr, quantreg (>= 5.51)
Imports rlang (>= 0.3.0), methods
Suggests cluster, classInt, KernSmooth, ggplot2, shiny, remotes, vpc, knitr, rmarkdown
License MIT + file LICENSE
LazyData true
Encoding UTF-8
VignetteBuilder knitr
RoxygenNote 7.0.2.9000
NeedsCompilation no
Author Olivier Barriere [aut], Benjamin Rich [aut], James Craig [aut, cre], Samer Mouksassi [aut], Kris Jamsen [ctb]
Maintainer James Craig <jameswcraigm@gmail.com>
**bininfo**

Obtain information about the bins from a VPC object.

**Description**

Obtain information about the bins from a VPC object.

**Usage**

```r
bininfo(o, ...)
```

## S3 method for class 'tidyvpcobj'

```r
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`: the number of observed data points in the bin
- `xmedian`: the median x-value of the observed data points in the bin
- `xmean`: the mean x-value of the observed data points in the bin
- `xmax`: the maximum x-value of the observed data points in the bin
- `xmin`: the minimum x-value of the observed data points in the bin
- `xmid`: the value halfway between `xmin` and `xmax`. x-value of the observed data points in the bin
- `xleft`: the 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`: the 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`: the 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**

**Description**

Perform binless Visual Predictive Check (VPC)

**Usage**

```r
binless(o, ...)
```

```r
## S3 method for class 'tidyvpcobj'
binless(
o,
qpred = c(0.05, 0.5, 0.95),
opimize = TRUE,
optimization.interval = c(0, 7),
conf.level = 0.95,
loess.ypc = FALSE,
lambda = NULL,
span = NULL,
...)
```

```
Arguments

- `tidyvpc` object
- `...` other arguments
- `qpred` numeric vector of length 3 specifying quantiles (lower, median, upper) i.e. `c(0.1, 0.5, 0.9)`
- `optimize` logical indicating whether lambda and span should be optimized using AIC
- `optimization.interval` numeric vector of length 2 specifying interval for lambda optimization
- `conf.level` numeric confidence level for binless fit
- `loess.ypc` logical indicating loess precision corrected. Must first use `predcorrect()` if `loess.ypc = TRUE`
- `lambda` numeric vector of length 3 specifying lambda values for each quantile
- `span` numeric number between 0.1 specifying smoothing parameter for loess prediction corrected

Details

Use this function in substitute of traditional binning methods to derive VPC using additive quantile regression and loess for pcVPC.

Value

Updates `tidyvpcobj` with additive quantile regression fits for observed and simulated data for quantiles specified in `qpred` argument. If `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.

See Also

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

Examples

```r
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(qpred = c(0.1, 0.5, 0.9), optimize = FALSE, lambda = lambda_strat) %>%
  vpcstats()

---

### Description

Binning methods for Visual Predictive Check (VPC)

### 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` tidyvpc object
- `...` Other arguments to include
- `bin` Character string indicating binning method or unquoted variable name if binning on x-variable.
Observed data supplied in observed() function

Character string indicating midpoint type for binning

Numeric vector of centers for binning. Use bin = "centers" if supplying centers

Numeric vector of breaks for binning. Use bin = "breaks" if supplying breaks

Numeric number indicating the number of bins to use

Unquoted variable name in observed data for alternative x-variable binning

List indicating the name of stratification variable and level if using different binning methods by strata

Logical indicating whether binning should be performed by strata

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

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

observed simulated censoring predcorrect stratify binless vpcstats

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

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

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
censoring

description

censoring observed data for Visual Predictive Check (VPC)

usage

censoring(o, ...)

## S3 method for class 'tidyvpcobj'
censoring(o, blq, lloq, alq, uloq, data = o$data, ...)

arguments

o tidyvpc object
...
... Other arguments to include
blq blq variable if present in observed data
lloq lloq variable if present in observed data. Use numeric to specify lloq value
alq logical variable indicating above limit of quantification
uloq number or numeric variable in data indicating the upper limit of quantification
data observed data supplied in observed() function

details

Specify censoring variables or censoring value for VPC using this function

value

Updates obs data.frame in tidyvpcobj with censored values for observed data which includes lloq and uloq specified values for lower/upper limit of quantification. Logicals for blq and alq are returned which indicate whether the DV value lies below/above limit of quantification.

see also

observed simulated stratify predcorrect binning binless vpcstats
Examples

```r
obs_data <- as.data.table(tidyvpc::obs_data)
sim_data <- as.data.table(tidyvpc::sim_data)

vpc <- observed(obs_data, x=TIME, y=DV) %>%
  simulated(sim_data, y=DV) %>%
  censoring(blq=(DV < 50), lloq=50) %>%
  binning(bin = "pam", nbins = 5) %>%
  vpcstats()

#Using LLOQ variable in data with different values of LLOQ by Study:
obs_data$LLOQ <- obs_data[, ifelse(STUDY == "Study A", 50, 25)]

vpc <- observed(obs_data, x=TIME, y=DV) %>%
  simulated(sim_data, y=DV) %>%
  censoring(blq=(DV < LLOQ), lloq=LLOQ) %>%
  stratify(~ STUDY) %>%
  binning(bin = "kmeans", nbins = 4) %>%
  vpcstats()
```

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

```r
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`'s must be given with those in positions 1 and 2 respectively.

Value

The number of replicates contained in `sim`. 

See Also

observed, simulated.

Examples

library(vpc)

exampleobs <- as.data.table(vpc::simple_data$obs)[MDV == 0]
examplesim <- as.data.table(vpc::simple_data$sim)[MDV == 0]

check_order(exampleobs[, .(ID, TIME)], examplesim[, .(ID, TIME)])
Details

Optional function to use indicating no pred correction for VPC.

Description

Specify observed dataset and variables for VPC

Usage

observed(o, ...)

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

Arguments

- **o**: data.frame or data.table of observation data
- **...**: other arguments
- **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

Details

The observed function is the first function in the vpc piping chain and is used for specifying observed data and variables for VPC

Value

A tidyvpcobj containing both original data and observed data formatted with x & 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
ox_data <- as.data.table(tidyvpc::obs_data)
sim_data <- as.data.table(tidyvpc::sim_data)

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

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

---

**obs_data**

*Example observed data from vpc package.*

**Description**

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

**Usage**

```r
obs_data
```

**Format**

A data frame 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`
Description

Plot a tidyvpcobj.

Usage

## S3 method for class 'tidyvpcobj'
plot(
  x,
  ...,
  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"),
  legend.position = "top",
  facet.scales = "free",
  custom.theme = "ggplot2::theme_bw"
)

Arguments

x A tidyvpcobj object.
...
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 linetyps for the percentiles, from low to high.
legend.position A character string specifying the position of the legend.
facet.scales A character string specifying the ‘scales’ argument to use for facetting.
custom.theme A Character string specifying theme from ggplot2 package
Details
Use ggplot2 graphics to plot and customize the appearance of VPC

Value
A ‘ggplot’ object.

See Also
ggplot

desc

Description
Prediction corrected Visual Predictive Check (pcVPC)

Usage
predcorrect(o, ...)

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

Arguments
o tidyvpc object
... 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

Details
Specify prediction variable for pcVPC

Value
Updates tidyvpcobj with required information to performing prediction correction which include predcor logical indicating whether prediction corrected VPC is to be performed, 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
obs_data <- as.data.table(tidyvpc::obs_data)
sim_data <- as.data.table(tidyvpc::sim_data)

# 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 a tidyvpcobj.

Usage

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

Arguments

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

Value

Returns `x` invisibly.
Description

Run Shiny app for tidyvpc

Usage

runShinyVPC()

Details

Use this function to run Shiny application to parameterize VPC from a GUI and generate corresponding tidyvpc code to derive VPC.

See Also

Shiny-VPC GitHub

Description

Specify simulated dataset and variables for VPC

Usage

simulated(o, ...)

## S3 method for class 'tidyvpcobj'
simulated(o, data, ysim, ...)

Arguments

- `o`: tidyvpcobj
- `...`: other arguments
- `data`: data.frame or data.table of simulated data
- `ysim`: numeric y-variable, typically named DV

Details

The simulated function is the second function in the vpc piping chain and is used for specifying simulated data and variables for VPC
sim_data

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

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

---

sim_data Example simulated data from vpc package.

Description

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

Usage

sim_data

Format

A data frame with 60000 rows and 10 variables:

ID Subect 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
Stratification for Visual Predictive Check (VPC)

Usage

stratify(o, ...)  

## S3 method for class 'tidyvpcobj'
stratify(o, formula, data = o$data, ...)

Arguments

- **o** tidyvpc object
- **...** Other arguments to include
- **formula** formula for stratification
- **data** Observed data supplied in observed() function

Details

specify stratification variables for VPC using this 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

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

# 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

---

Description

Compute VPC statistics

Usage

vpcstats(o, ...)

## S3 method for class 'tidyvpcobj'
vpcstats(
o,
qpred = c(0.05, 0.5, 0.95),
...,
conf.level = 0.95,
quantile.type = 7
)

Arguments

o tidyvpc object
...
Other arguments to include
qpred Numeric vector of length 3 specifying quantile prediction interval
conf.level Numeric specifying confidence level
quantile.type Numeric indicating quantile type. See quantile

Details

Compute predictional interval statistics for VPC

Value

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

- bin: the resulting bin value as specified in ‘binning()’ function
- xbin: the midpoint x-value of the observed data points in the bin as specified in ‘xbin’ argument of ‘binning()’ function
- qname: the quantiles specified in ‘qpred’
- y: the observed y value for the specified quantile
- lo: the lower bound of specified confidence interval for y value in simulated data
- md: the median y value in simulated data
- hi: the upper bound of specified confidence interval for y value in simulated data
See Also

observed simulated censoring stratify binning binless precorrect
Index

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