

Package ‘tidyvpc’

September 29, 2020

Type Package

Title VPC Percentiles and Prediction Intervals

Version 1.1.0

Date 2020-09-28

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 Jansen 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/certara/tidyvpc>

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

Depends R (>= 3.5.0), data.table (>= 1.9.8), magrittr, quantreg (>= 5.51)

Imports rlang (>= 0.3.0), methods, classInt, ggplot2

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

License MIT + file LICENSE

LazyData true

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 7.1.1

NeedsCompilation no

Author Olivier Barriere [aut],
Benjamin Rich [aut],
James Craig [aut, cre],
Samer Mouksassi [aut],
Kris Jansen [ctb]

Maintainer James Craig <jameswbcraig@gmail.com>

Repository CRAN

Date/Publication 2020-09-29 07:00:03 UTC

R topics documented:

bininfo	2
binless	3
binning	5
binningfunctions	7
censoring	8
check_order	9
generics	10
nopredcorrect	11
observed	11
obs_data	12
plot.tidyvpcobj	13
predcorrect	14
print.tidyvpcobj	16
runShinyVPC	16
simulated	17
sim_data	18
stratify	18
vpstats	19
Index	21

bininfo	<i>Obtain information about the bins from a VPC object.</i>
---------	---

Description

Obtain information about the bins from a VPC object.

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`: 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

binless

Description

Perform binless Visual Predictive Check (VPC)

Usage

```
binless(o, ...)

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

Arguments

<code>o</code>	tidyvpc object
<code>...</code>	other arguments
<code>qpred</code>	numeric vector of length 3 specifying quantiles (lower, median, upper) i.e. <code>c(0.1, 0.5, 0.9)</code>
<code>optimize</code>	logical indicating whether lambda and span should be optimized using AIC
<code>optimization.interval</code>	numeric vector of length 2 specifying interval for lambda optimization
<code>conf.level</code>	numeric confidence level for binless fit
<code>loess.ypc</code>	logical indicating loess precision corrected. Must first use <code>predcorrect()</code> if <code>loess.ypc = TRUE</code>
<code>lambda</code>	numeric vector of length 3 specifying lambda values for each quantile
<code>span</code>	numeric number between 0,1 specyng smoothing paramter for loess prediction corrected

Details

Use this function in subsitute 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

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

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

```

binning

binning

Description

Binning methods for Visual Predictive Check (VPC)

Usage

```

binning(o, ...)

## S3 method for class 'tidyvpcobj'
binning(
  o,
  bin,
  data = o$data,
  xbin = "xmedian",
  centers,
  breaks,
  nbins,
  altx,
  stratum = NULL,
  by.strata = TRUE,
  ...
)

```

Arguments

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

Details

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

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

```
# 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) %>%
  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()

```

binningfunctions	<i>Different functions that perform binning.</i>
------------------	--

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 <code>?classInt::classIntervals</code> 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

```
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)

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

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

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

```
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()
```

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's must be given with those in positions 1 and 2 repectively.

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)])
```

generics

Perform a Visual Predictive Check (VPC) computation

Description

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

Arguments

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

nopredcorrect	<i>nopredcorrect</i>
---------------	----------------------

Description

No pred correction for Visual Predictive Check (VPC)

Usage

```
nopredcorrect(o, ...)

## S3 method for class 'tidyvpobj'
nopredcorrect(o, ...)
```

Arguments

<code>o</code>	tidyvpobj
<code>...</code>	other arguments to include

Details

Optional function to use indicating no pred correction for VPC.

observed	<i>observed</i>
----------	-----------------

Description

Specify observed dataset and variables for VPC

Usage

```
observed(o, ...)

## S3 method for class 'data.frame'
observed(
  o,
  x,
  yobs,
  pred = NULL,
  blq = NULL,
  lloq = -Inf,
  alq = NULL,
  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. Note: Observed data must not contain missing DV and may require subsetting MDV == 0 before generating 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

```
obs_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

```
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](#)

plot.tidyvpcobj	<i>plot</i>
-----------------	-------------

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

<code>x</code>	A tidyvpcobj object.
<code>...</code>	Further arguments can be specified but are ignored.
<code>show.points</code>	Should the observed data points be plotted?
<code>show.boundaries</code>	Should the bin boundary be displayed?
<code>show.stats</code>	Should the VPC stats be displayed?
<code>show.binning</code>	Should the binning be displayed by coloring the observed data points by bin?
<code>xlab</code>	A character label for the x-axis.
<code>ylab</code>	A character label for the y-axis.
<code>color</code>	A character vector of colors for the percentiles, from low to high.
<code>linetype</code>	A character vector of linetyps for the percentiles, from low to high.
<code>legend.position</code>	A character string specifying the position of the legend.
<code>facet.scales</code>	A character string specifying the 'scales' argument to use for facetting.
<code>custom.theme</code>	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

predcorrect

predcorrect

Description

Prediction corrected Visual Predictive Check (pcVPC)

Usage

```
predcorrect(o, ...)
```

```
## S3 method for class 'tidyvpcobj'
```

```
predcorrect(o, pred, data = o$data, ..., log = FALSE)
```

Arguments

<code>o</code>	tidyvpc object
<code>...</code>	Other arguments to include
<code>pred</code>	prediction variable in observed data
<code>data</code>	observed data supplied in <code>observed()</code> function
<code>log</code>	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

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

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) %>%
```

```
vpstats()
```

```
print.tidyvpcobj      Print a tidyvpcobj.
```

Description

Print a tidyvpcobj.

Usage

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

Arguments

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

Value

Returns x invisibly.

```
runShinyVPC            runShinyVPC
```

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](#)

<i>simulated</i>	<i>observed</i>
------------------	-----------------

Description

Specify simulated dataset and variables for VPC

Usage

```
simulated(o, ...)

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

Arguments

<code>o</code>	<code>tidyvpobj</code>
<code>...</code>	other arguments
<code>data</code>	<code>data.frame</code> or <code>data.table</code> of simulated data
<code>ysim</code>	numeric y-variable, typically named DV

Details

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 subsetting `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.

Value

A `tidyvpobj` 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	<i>Example simulated data from vpc package.</i>
----------	---

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 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	<i>stratify</i>
----------	-----------------

Description

Stratification for Visual Predictive Check (VPC)

Usage

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

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

Arguments

<code>o</code>	tidyvpc object
<code>...</code>	Other arguments to include
<code>formula</code>	formula for stratification
<code>data</code>	Observed data supplied in <code>observed()</code> 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) %>%
  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

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

<code>o</code>	tidyvpc object
<code>...</code>	Other arguments to include
<code>qpred</code>	Numeric vector of length 3 specifying quantile prediction interval
<code>conf.level</code>	Numeric specifying confidence level
<code>quantile.type</code>	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](#) [predcorrect](#)

Index

* datasets

obs_data, [12](#)

sim_data, [18](#)

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](#), [9](#), [12](#), [15](#), [17](#), [19](#), [20](#)

binning, [4](#), [5](#), [9](#), [12](#), [15](#), [17](#), [19](#), [20](#)

binningfunctions, [7](#)

censoring, [4](#), [6](#), [8](#), [12](#), [15](#), [17](#), [19](#), [20](#)

check_order, [9](#)

classIntervals, [6](#)

cut_at (binningfunctions), [7](#)

generics, [10](#)

nearest (binningfunctions), [7](#)

nopredcorrect, [11](#)

obs_data, [12](#)

observed, [4](#), [6](#), [9](#), [10](#), [11](#), [15](#), [17](#), [19](#), [20](#)

plot.tidyvpcobj, [13](#)

predcorrect, [4](#), [6](#), [9](#), [12](#), [14](#), [17](#), [19](#), [20](#)

print.tidyvpcobj, [16](#)

quantile, [20](#)

runShinyVPC, [16](#)

sim_data, [18](#)

simple_data, [13](#), [18](#)

simulated, [4](#), [6](#), [9](#), [10](#), [12](#), [15](#), [17](#), [19](#), [20](#)

stratify, [4](#), [6](#), [9](#), [12](#), [15](#), [17](#), [18](#), [20](#)

vpcstats, [4](#), [6](#), [9](#), [12](#), [15](#), [17](#), [19](#), [19](#)