Package ‘vpc’

August 27, 2018

Title Create Visual Predictive Checks
Version 1.1.0
Date 2018-08-27
Author Ron Keizer <ronkeizer@gmail.com>
Maintainer Ron Keizer <ronkeizer@gmail.com>
Description Visual predictive checks are a commonly used diagnostic plot in pharmacometrics, showing how certain statistics (percentiles) for observed data compare to those same statistics for data simulated from a model. The package can generate VPCs for continuous, categorical, censored, and (repeated) time-to-event data.

Depends R (>= 3.1.0)
Imports classInt, dplyr, reshape2, MASS, survival, ggplot2, readr
License MIT + file LICENSE
LazyData true
URL https://github.com/ronkeizer/vpc
Suggests knitr, testit
RoxygenNote 6.0.1
NeedsCompilation no
Repository CRAN
Date/Publication 2018-08-27 21:00:03 UTC

R topics documented:

  vpc-package .................................................. 2
  add_noise .................................................... 3
  add_sim_index_number ....................................... 3
  add_stratification .......................................... 4
  auto_bin ..................................................... 4
  bin_data ..................................................... 5
  check_stratification_columns_available .................. 5
  compute_kaplan ............................................. 6
Index 27

vpc-package VPC package

Description
Create Visual Predictive Checks in R

Author(s)
Ron Keizer <ronkeizer@gmail.com>
**add_noise**  
*Add noise / residual error to data*

**Description**
Add noise / residual error to data

**Usage**
```
add_noise(x, ruv = list(proportional = 0, additive = 0, exponential = 0))
```

**Arguments**
- `x`: data
- `ruv`: list describing the magnitude of errors. List arguments: "proportional", "additive", "exponential".

**Examples**
```
library(dplyr)
ipred <- c(10, 8, 6, 4, 2, 0) %>% add_noise(ruv = list(proportional = 0.1, additive = 0.2))
```

---

**add_sim_index_number**  
*Add sim index number*

**Description**
Add simulation index number to simulation when not present

**Usage**
```
add_sim_index_number(sim, id = "id", sim_label = "sim")
```

**Arguments**
- `sim`: a data.frame containing the simulation data
- `id`: character specifying the column name in the data.frame
- `sim_label`: label to indicate simulation index (if available)
add_stratification  
Adds stratification to data set

Description
Adds stratification to data set

Usage
add_stratification(dat, strat, verbose = FALSE)

Arguments
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat</td>
<td>data.frame</td>
</tr>
<tr>
<td>strat</td>
<td>vector of stratification variables</td>
</tr>
<tr>
<td>verbose</td>
<td>verbosity (‘TRUE’ or ‘FALSE’)</td>
</tr>
</tbody>
</table>

auto_bin  
Calculate appropriate bin separators for vpc

Description
This function calculates bin separators either using R’s native binning approaches available in the classInt library such as ‘kmeans’, ‘jenks’, ‘pretty’ etc. Alternatively, a custom approach is available which is based on finding the nadirs in the density functions for the independent variable. Default approach is k-means clustering.

Usage
auto_bin(dat, type = "kmeans", n_bins = 8, verbose = FALSE)

Arguments
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat</td>
<td>data frame</td>
</tr>
<tr>
<td>type</td>
<td>auto-binning type: &quot;density&quot;, &quot;time&quot;, or &quot;data&quot;</td>
</tr>
<tr>
<td>n_bins</td>
<td>number of bins to use. For &quot;density&quot; the function might not return a solution with the exact number of bins.</td>
</tr>
<tr>
<td>verbose</td>
<td>show warnings and other messages (TRUE or FALSE)</td>
</tr>
</tbody>
</table>

Value
A vector of bin separators
**bin_data**

*Function to bin data based on a vector of bin separators, e.g. for use in VPC*

**Description**

Function to bin data based on a vector of bin separators, e.g. for use in VPC

**Usage**

```r
bin_data(x, bins = c(0, 3, 5, 7), idv = "time", labeled = F)
```

**Arguments**

- `x`: data
- `bins`: numeric vector specifying bin separators
- `idv`: variable in the data specifies the independent variable (e.g. "time")
- `labeled`: whether a labeled factor instead of integers should be returned

---

**check_stratification_columns_available**

*Check whether stratification columns are available*

**Description**

Check whether stratification columns are available

**Usage**

```r
check_stratification_columns_available(data, stratify, type = "observation")
```

**Arguments**

- `data`: ‘data.frame’ with observation or simulation data
- `stratify`: vector of stratification columns
- `type`: either ‘observation’ or ‘simulation’
compute_kaplan  Compute Kaplan-Meier statistics

**Description**

Compute Kaplan-Meier statistics

**Usage**

```r
compute_kaplan(dat, strat = "strat", reverse_prob = FALSE, ci = NULL)
```

**Arguments**

- `dat`: data.frame with events
- `strat`: vector of stratification variables
- `reverse_prob`: reverse the probability (i.e. return ‘1-probability’)?
- `ci`: confidence interval to calculate, numeric vector of length 2

compute_kmmc  Compute KMMC statistics

**Description**

Kaplan-Meier Mean Covariate plots are a simulation-based diagnostic to study the influence of covariates and identify potential model misspecification.

**Usage**

```r
compute_kmmc(dat, strat = NULL, reverse_prob = FALSE, kmmc = "DOSE")
```

**Arguments**

- `dat`: data.frame with events
- `strat`: vector of stratification variables
- `reverse_prob`: reverse the probability (i.e. return ‘1-probability’)?
- `kmmc`: variable to create the KMMC plot for.
**create_vpc_theme**

Create new vpc theme

**Usage**

create_vpc_theme(...)  

**Arguments**

... pass arguments to 'new_vpc_theme'

---

**define_data_columns**

Define data column defaults for various softwares

**Description**

Define data column defaults for various softwares

**Usage**

define_data_columns(sim, obs, sim_cols, obs_cols, software_type)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sim</td>
<td>simulated data</td>
</tr>
<tr>
<td>obs</td>
<td>observed data</td>
</tr>
</tbody>
</table>
| sim_cols       | list for mapping simulation data columns, e.g. `list(dv = "DV", id = "ID", idv =  
|                | "TIME", pred="PRED")`                                                      |
| obs_cols       | list for mapping observation data columns, e.g. `list(dv = "DV", id = "ID", idv =  
|                | "TIME", pred="PRED")`                                                      |
| software_type  | software type, one of 'nonmem', 'phoenix', 'PKPDsim'                       |
**draw_params_mvr**  
*Draw parameters from multivariate distribution*

**Description**
Draw parameters from multivariate distribution

**Usage**
draw_params_mvr(ids, n_sim, theta, omega_mat, par_names = NULL)

**Arguments**
- `ids` vector of id numbers
- `n_sim` number of simulations
- `theta` theta vector
- `omega_mat` omega matrix
- `par_names` parameter names vector

**loq_perc**  
*Calculate percentiles below / above lloq / uloq*

**Description**
Calculate percentiles below / above lloq / uloq

**Usage**
loq_perc(x, limit = 1, cens = "left")

**Arguments**
- `x` data
- `limit` censoring limit
- `cens` censoring direction (left/right)
new_vpc_theme

Create a customized VPC theme

Description

Create a customized VPC theme

Usage

new_vpc_theme(update = NULL)

Arguments

update list containing the plot elements to be updated. Run ‘new_vpc_theme()’ with no arguments to show an overview of available plot elements.

Details

This function creates a theme that customizes how the VPC looks, i.e. colors, fills, transparencies, linetypes an sizes, etc. The following arguments can be specified in the input list:

- obs_color: color for observations points
- obs_size: size for observation points
- obs_median_color: color for median observation line
- obs_median_linetype: linetype for median observation line
- obs_median_size: size for median observation line
- obs_ci_fill: color for observation CI fill
- obs_ci_color: color for observation CI lines
- obs_ci_linetype: linetype for observation CI lines
- obs_ci_size: size for observations CI lines
- sim_pi_fill: fill color for simulated prediction interval areas
- sim_pi_alpha: transparency for simulated prediction interval areas
- sim_pi_color: color for simulated prediction interval lines
- sim_pi_linetype: linetype for simulated prediction interval lines
- sim_pi_size: size for simulated prediction interval lines
- sim_median_fill: fill color for simulated median area
- sim_median_alpha: transparency for simulated median area
- sim_median_color: color for simulated median line
- sim_median_linetype: linetype for simulated median line
- sim_median_size: size for simulated median line
- bin_separators_color: color for bin separator lines, NA for don’t plot
- bin_separators_location: where to plot bin separators ("t" for top, "b" for bottom)
- loq_color: color of line showing limit of quantification
Value

A list with vpc theme specifiers

Examples

```r
date <- c(0, 24)
t <- date[1]
dose <- 120
cl <- 0.345
vc <- 1.75

vpc:::pk_iv_1cmt(t = t, t_inf = 1, tau = 24, dose = 120, CL = 0.345, Vc = 1.75, ruv = NULL)
```

pk_iv_1cmt

Simulate PK data from a 1-compartment iv model

Description

Simulate PK data from a 1-compartment iv model

Usage

```
pk_iv_1cmt(t, t_inf, tau, dose, cl, vc, ruv)
```

Arguments

- **t**: Time after dose
- **t_inf**: Infusion length
- **tau**: Dosing interval
- **dose**: Dose
- **cl**: Clearance
- **vc**: Volume of distribution
- **ruv**: Residual variability

Value

A vector of predicted values, with or without added residual variability

Examples

```
dat1 <- vpc:::pk_iv_1cmt(t = c(0:22), tau = 24, dose = 120, CL = 5, Vc = 50)
dat2 <- vpc:::pk_iv_1cmt(t = c(0:22), tau = 24, dose = 120, CL = 5, Vc = 50, ruv = list(proportional = 0.1, additive = 0.1))
```
Simulate PK data from a 1-compartment oral model

**Usage**

```r
pk_oral_1cmt(t, tau = 24, dose = 120, ka = 1, ke = 1, cl = 10, ruv = NULL)
```

**Arguments**

- `t` Time after dose
- `tau` Dosing interval
- `dose` Dose
- `ka` Absorption rate
- `ke` Elimination rate
- `cl` Clearance
- `ruv` Residual variability

**Value**

A vector of predicted values, with or without added residual variability

**Examples**

```r
dat1 <- vpc:::pk_oral_1cmt(t = c(0:72), tau = 24, dose = 120, ka = 1, ke = 1, cl = 10)
dat2 <- vpc:::pk_oral_1cmt(t = c(0:72), tau = 24, dose = 120, ka = 1, ke = 1, cl = 10, ruv = list(proportional = 0.1, additive = 0.1))
```

**Description**

This function performs no parsing of data, it just plots the already calculated statistics generated using one of the `vpc` functions.
quantile_cens

Usage

```
plot_vpc(db, show = NULL, vpc_theme = NULL, smooth = TRUE,
log_x = FALSE, log_y = FALSE, xlab = NULL, ylab = NULL,
title = NULL, verbose = FALSE)
```

Arguments

- `db` object created using the `vpc` function
- `show` what to show in VPC (obs_dv, obs_ci, pi, pi_as_area, pi_ci, obs_median, sim_median, sim_median_ci)
- `vpc_theme` theme to be used in VPC. Expects list of class vpc_theme created with function vpc_theme()
- `smooth` “smooth” the VPC (connect bin midpoints) or show bins as rectangular boxes. Default is TRUE.
- `log_x` Boolean indicating whether x-axis should be shown as logarithmic. Default is FALSE.
- `log_y` Boolean indicating whether y-axis should be shown as logarithmic. Default is FALSE.
- `xlab` label for x axis
- `ylab` label for y axis
- `title` title
- `verbose` verbosity (T/F)

See Also

`sim_data, vpc_cens, vpc_tte, vpc_cat`

Examples

```r
## See vpc.ronkeizer.com for more documentation and examples

library(vpc)
vpc_db <- vpc(sim = simple_data$sim, obs = simple_data$obs, vpcdb = TRUE)
plot_vpc(vpc_db, title = "My new vpc", x = "Custom x label")
```

quantile_cens

Calculate quantiles respecting the censored data

Description

Calculate quantiles respecting the censored data

Usage

```
quantile_cens(x, p = 0.5, limit = 1, cens = "left")
```
Arguments

- `x`: data
- `p`: quantile
- `limit`: censoring limit
- `cens`: censoring direction (left/right)

**read_table_nm**  
*NONMEM output table import function*

**Description**

Quickly import NONMEM output tables into R. Function taken from `modelviz` package by Benjamin Guiastrennec. When both `skip` and `header` are `NULL`, `read_table_nm` will automatically detect the optimal settings to import the tables. When more than one files are provided for a same NONMEM run, they will be combined into a single `data.frame`.

**Usage**

```r
read_table_nm(file = NULL, skip = NULL, header = NULL,
              rm_duplicates = FALSE, nonmem_tab = TRUE)
```

**Arguments**

- `file`: full file name
- `skip`: number of lines to skip before reading data
- `header`: logical value indicating whether the file contains the names of the variables as its first line
- `rm_duplicates`: logical value indicating whether duplicated columns should be removed
- `nonmem_tab`: logical value indicating to the function whether the file is a table or a nonmem additional output file.

**Value**

A `data.frame`

**Examples**

```r
## Not run:
data <- read_table_nm(file = '..', models/pk/sdtab101')

## End(Not run)
**replace_list_elements**  
*Replace list elements by name*

**Description**  
Replace list elements by name

**Usage**  
```r
replace_list_elements(list, replacement)
```

**Arguments**
- `list`  
  original list
- `replacement`  
  replacement list

**Details**  
Finds and replaces list elements by name and throws an error if an element is not available in the original list. This is a local duplicate of the PKPDmisc copy for the VPC package to reduce dependency on PKPDmisc at this time.

**Examples**  
```r
## Not run:
list <- list(ipred = "ipred", dv = "dv", idv = "idv", pred = "pred")
replacement <- list(dv = "conc", idv = "time")
list <- replace_list_elements(list, replacement)

## End(Not run)
```

---

**rtte_obs_nm**  
*Simulated RTTE data (1x)*

**Description**  
An example dataset with simulated repeated time-to-event data

**Usage**  
```r
rtte_obs_nm
```

**Format**  
An object of class `data.frame` with 573 rows and 6 columns.
### rtte_sim_nm

**Simulated RTTE data (100x)**

#### Description
An example dataset with simulated repeated time-to-event data (100 simulations)

#### Usage
 rtte_sim_nm

#### Format
An object of class `data.frame` with 2000000 rows and 7 columns.

---

### show_default

**Defaults for show argument**

#### Description
Defaults for show argument

#### Usage
 show_default

#### Format
An object of class `list` of length 11.

---

### show_default_tte

**Defaults for show argument for TTE VPC**

#### Description
Defaults for show argument for TTE VPC

#### Usage
 show_default_tte

#### Format
An object of class `list` of length 11.
simple_data  

*Description*

A small rich dataset

*Usage*

`simple_data`

*Format*

An object of class `list` of length 2.

*Details*

A list containing the obs and sim data for an example dataset to run a simple vpc.

*Examples*

```r
## Not run:
vpc(simple_data$sim, simple_data$obs)
## End(Not run)
```

---

sim_data  

*Description*

Simulate data based on a model and parameter distributions

*Usage*

```r
sim_data
design = cbind(id = c(1, 1, 1), idv = c(0, 1, 2)),
model = function(x) { return(x$alpha + x$beta) },
theta, omega_mat, par_names, par_values = NULL, draw_iiv = "mvrnorm",
error = list(proportional = 0, additive = 0, exponential = 0), n = 100)
```
Arguments

design: a design dataset. See example
model: A function with the first argument the simulation design, i.e. a dataset with the columns ... The second argument to this function is a dataset with parameters for every individual. This can be supplied by the user, or generated by this sim_data if theta and omega_mat are supplied.
theta: vector of fixed effect parameters
omega_mat: vector of between subject random effects, specified as lower triangle
par_names: A character vector linking the parameters in the model to the variables in the dataset. See example.
par_values: parameter values
draw_iiv: draw between subject random effects?
error: see example
n: number of simulations to perform

Details
This function generates the simulated dependent values for use in the VPC plotting function.

Value
a vector of simulated dependent variables (for us in the VPC plotting function)

See Also
vpc

theme_empty

Empty ggplot2 theme

Description
Empty ggplot2 theme

Usage
theme_empty()

Examples
vpc(simple_data$sim, simple_data$obs) + theme_empty()
theme_plain  A nicer default theme for ggplot2

Description
A nicer default theme for ggplot2

Usage
theme_plain()

Examples
vpc(simple_data$sim, simple_data$obs) + theme_plain()

triangle_to_full  Lower to full triangle

Description
Convert the lower triangle of a covariance matrix to a full matrix object

Usage
triangle_to_full(vect)

Arguments

vect the lower triangle of a covariance matrix

vpc  VPC function

Description
Creates a VPC plot from observed and simulation data
Usage

vpc(sim, ...)

## Default S3 method:
vpc(sim, ...)

vpc_vpc(sim = NULL, obs = NULL, psn_folder = NULL, bins = "jenks", n_bins = "auto", bin_mid = "mean", obs_cols = NULL, sim_cols = NULL, software = "auto", show = NULL, stratify = NULL, pred_corr = FALSE, pred_corr_lower_bnd = 0, pi = c(0.05, 0.95), ci = c(0.05, 0.95), uloq = NULL, lloq = NULL, log_y = FALSE, log_y_min = 0.001, xlab = NULL, ylab = NULL, title = NULL, smooth = TRUE, vpc_theme = NULL, facet = "wrap", labeller = NULL, vpcdb = FALSE, verbose = FALSE, ...)

Arguments

sim
this is usually a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm. However it can also be an object like a nlmixr or xpose object

... Other arguments sent to other methods (like xpose or nlmixr): Note these arguments are not used in the default vpc and are ignored by the default method.

obs
a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm

psn_folder
instead of specifying "sim" and "obs", specify a PsN-generated VPC-folder

bins
either "density", "time", or "data", "none", or one of the approaches available in classInterval() such as "jenks" (default) or "pretty", or a numeric vector specifying the bin separators.

n_bins
when using the "auto" binning method, what number of bins to aim for

bin_mid
either "mean" for the mean of all timepoints (default) or "middle" to use the average of the bin boundaries.

obs_cols
observation dataset column names (list elements: "dv", "idv", "id", "pred")

sim_cols
simulation dataset column names (list elements: "dv", "idv", "id", "pred", "sim")

software
name of software platform using (e.g. nonmem, phoenix)

show
what to show in VPC (obs_dv, obs_ci, pi, pi_as_area, pi_ci, obs_median, sim_median, sim_median_ci)

stratify
character vector of stratification variables. Only 1 or 2 stratification variables can be supplied.

pred_corr
perform prediction-correction?

pred_corr_lower_bnd
lower bound for the prediction-correction

pi
simulated prediction interval to plot. Default is c(0.05, 0.95),
ci  confidence interval to plot. Default is (0.05, 0.95)
uloq  Number or NULL indicating upper limit of quantification. Default is NULL.
lloq  Number or NULL indicating lower limit of quantification. Default is NULL.
log_y  Boolean indicating whether y-axis should be shown as logarithmic. Default is FALSE.
log_y_min  minimal value when using log_y argument. Default is 1e-3.
xlab  label for x axis
ylab  label for y axis
title  title
smooth  "smooth" the VPC (connect bin midpoints) or show bins as rectangular boxes. Default is TRUE.
vpc_theme  theme to be used in VPC. Expects list of class vpc_theme created with function vpc_theme()
facet  either "wrap", "columns", or "rows"
labeller  ggplot2 labeller function to be passed to underlying ggplot object
vpcdb  Boolean whether to return the underlying vpcdb rather than the plot
verbose  show debugging information (TRUE or FALSE)

Value

a list containing calculated VPC information (when vpcdb=TRUE), or a ggplot2 object (default)

See Also

sim_data, vpc_cens, vpc_tte, vpc_cat

Examples

## See vpc.ronkeizer.com for more documentation and examples
library(vpc)

# Basic commands:
vpc(sim = simple_data$sim, obs = simple_data$obs)
vpc(sim = simple_data$sim, obs = simple_data$obs, lloq = 20)
vpc_cat

VPC function for categorical

Description

Creates a VPC plot from observed and simulation data for categorical variables.

Usage

vpc_cat(sim = NULL, obs = NULL, psn_folder = NULL, bins = "jenks",
        n_bins = "auto", bin_mid = "mean", obs_cols = NULL, sim_cols = NULL,
        software = "auto", show = NULL, ci = c(0.05, 0.95), uloq = NULL,
        lloq = NULL, xlab = NULL, ylab = NULL, title = NULL, smooth = TRUE,
        vpc_theme = NULL, facet = "wrap", labeller = NULL, plot = TRUE,
        vpcdb = FALSE, verbose = FALSE)

Arguments

sim a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm
obs a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm
psn_folder instead of specifying "sim" and "obs", specify a PsN-generated VPC-folder
bins either "density", "time", or "data", "none", or one of the approaches available in classInterval() such as "jenks" (default) or "pretty", or a numeric vector specifying the bin separators.
n_bins when using the "auto" binning method, what number of bins to aim for
bin_mid either "mean" for the mean of all timepoints (default) or "middle" to use the average of the bin boundaries.
obs_cols observation dataset column names (list elements: "dv", "idv", "id", "pred")
sim_cols simulation dataset column names (list elements: "dv", "idv", "id", "pred")
software name of software platform using (e.g. nonmem, phoenix)
show what to show in VPC (obs_ci, pi, pi_as_area, pi_ci, obs_median, sim_median, sim_median_ci)
ci confidence interval to plot. Default is (0.05, 0.95)
uloq Number or NULL indicating upper limit of quantification. Default is NULL.
lloq Number or NULL indicating lower limit of quantification. Default is NULL.
xlab label for x-axis
ylab label for y-axis
title title
smooth  "smooth" the VPC (connect bin midpoints) or show bins as rectangular boxes. Default is TRUE.

vpc_theme theme to be used in VPC. Expects list of class vpc_theme created with function vpc_theme()

facet either "wrap", "columns", or "rows"

labeller ggplot2 labeller function to be passed to underlying ggplot object

plot Boolean indicating whether to plot the ggplot2 object after creation. Default is FALSE.

vpcdb boolean whether to return the underlying vpcdb rather than the plot

verbose show debugging information (TRUE or FALSE)

Value

a list containing calculated VPC information (when vpcdb=TRUE), or a ggplot2 object (default)

See Also

sim_data, vpc, vpc_tte, vpc_cens

Examples

```r
## See vpc.ronkeizer.com for more documentation and examples
library(vpc)

# simple function to simulate categorical data for single individual
sim_id <- function(id = 1) {
  n <- 10
  logit <- function(x) exp(x) / (1+exp(x))
  data.frame(id = id, time = seq(1, n, length.out = n),
             dv = round(logit((1:n) - n/2 + rnorm(n, 0, 1.5))) )
}

## simple function to simulate categorical data for a trial
sim_trial <- function(i = 1, n = 20) { # function to simulate categorical data for a trial
  data.frame(sim = i, do.call("rbind", lapply(1:n, sim_id)))
}

## simulate single trial for 20 individuals
obs <- sim_trial(n = 20)

## simulate 200 trials of 20 individuals
sim <- do.call("rbind", lapply(1:200, sim_trial, n = 20))

## Plot categorical VPC
vpc_cat(sim = sim, obs = obs)
```
vpc_cens

VPC function for left- or right-censored data (e.g. BLOQ data)

Description

Creates a VPC plot from observed and simulation data for censored data. Function can handle both left- (below lower limit of quantification) and right-censored (above upper limit of quantification) data.

Usage

vpc_cens(sim = NULL, obs = NULL, psn_folder = NULL, bins = "jenks", n_bins = 8, bin_mid = "mean", obs_cols = NULL, sim_cols = NULL, software = "auto", show = NULL, stratify = NULL, stratify_color = NULL, ci = c(0.05, 0.95), uloq = NULL, lloq = NULL, plot = FALSE, xlab = "Time", ylab = "Probability of <LOQ", title = NULL, smooth = TRUE, vpc_theme = NULL, facet = "wrap", labeller = NULL, vpcdb = FALSE, verbose = FALSE)

Arguments

sim a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm

obs a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm

psn_folder instead of specifying "sim" and "obs", specify a PsN-generated VPC-folder

bins either "density", "time", or "data", or a numeric vector specifying the bin separators.

n_bins number of bins

bin_mid either "mean" for the mean of all timepoints (default) or "middle" to use the average of the bin boundaries.

obs_cols observation dataset column names (list elements: "dv", "idv", "id", "pred")

sim_cols simulation dataset column names (list elements: "dv", "idv", "id", "pred")

software name of software platform using (e.g. nonmem, phoenix)

show what to show in VPC (obs_ci, pi, pi_as_area, pi_ci, obs_median, sim_median, sim_median_ci)

stratify character vector of stratification variables. Only 1 or 2 stratification variables can be supplied.

stratify_color variable to stratify and color lines for observed data. Only 1 stratification variables can be supplied.

ci confidence interval to plot. Default is (0.05, 0.95)
uloq  Number or NULL indicating upper limit of quantification. Default is NULL.
1loq  Number or NULL indicating lower limit of quantification. Default is NULL.
plot Boolean indicating whether to plot the ggplot2 object after creation. Default is FALSE.
xlab  ylab as numeric vector of size 2
ylab  ylab as numeric vector of size 2
title title
smooth "smooth" the VPC (connect bin midpoints) or show bins as rectangular boxes. Default is TRUE.
vpc_theme theme to be used in VPC. Expects list of class vpc_theme created with function vpc_theme()
facet either "wrap", "columns", or "rows"
labeller ggplot2 labeller function to be passed to underlying ggplot object
vpcdb boolean whether to return the underlying vpcdb rather than the plot
verbose show debugging information (TRUE or FALSE)

Value
a list containing calculated VPC information, and a ggplot2 object

See Also
sim_data, vpc, vpc_tte, vpc_cat

Examples

## See vpc.ronkeizer.com for more documentation and examples
library(vpc)
vpc_cens(sim = simple_data$sim, obs = simple_data$obs, lloq = 30)
vpc_cens(sim = simple_data$sim, obs = simple_data$obs, uloq = 120)

vpc_tte | VPC function for time-to-event (survival) data

Description
This function can be used for either single time-to-event (TTE) or repeated time-to-event (RTTE) data.
vpc_tte

Usage

vpc_tte(sim = NULL, obs = NULL, psn_folder = NULL, rtte = FALSE,
rtte_calc_diff = TRUE, events = NULL, bins = FALSE, n_bins = 10,
software = "auto", obs_cols = NULL, sim_cols = NULL, kmmc = NULL,
reverse_prob = FALSE, stratify = NULL, stratify_color = NULL,
ct = c(0.05, 0.95), plot = FALSE, xlab = "Time",
ylab = "Survival (%)", show = NULL, as_percentage = TRUE,
title = NULL, smooth = FALSE, vpc_theme = NULL, facet = "wrap",
labeller = NULL, verbose = FALSE, vpcdb = FALSE)

Arguments

sim a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm
obs a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm
psn_folder instead of specifying "sim" and "obs", specify a PsN-generated VPC-folder
rtte repeated time-to-event data? Default is FALSE (treat as single-event TTE)
rtte_calc_diff recalculate time (T/F)? When simulating in NONMEM, you will probably need to set this to TRUE to recalculate the TIME to relative times between events (unless you output the time difference between events and specify that as independent variable to the vpc_tte() function.

events numeric vector describing which events to show a VPC for when repeated TTE data, e.g. c(1:4). Default is NULL, which shows all events.
bins either "density", "time", or "data", or a numeric vector specifying the bin separators.
n_bins number of bins
software name of software platform using (e.g. nonmem, phoenix)
obs_cols observation dataset column names (list elements: "dv", "idv", "id", "pred")
sim_cols simulation dataset column names (list elements: "dv", "idv", "id", "pred", "sim")
kmmc either NULL (for regular TTE vpc, default), or a variable name for a KMMC plot (e.g. "WT")
reverse_prob reverse the probability scale (i.e. plot 1-probability)
stratify character vector of stratification variables. Only 1 or 2 stratification variables can be supplied.
stratify_color character vector of stratification variables. Only 1 stratification variable can be supplied, cannot be used in conjunction with 'stratify'.
ci confidence interval to plot. Default is (0.05, 0.95)
plot Boolean indicating whether to plot the ggplot2 object after creation. Default is FALSE.
xlab label for x-axis
ylab label for y-axis
show what to show in VPC (obs_ci, obs_median, sim_median, sim_median_ci)
as_percentage Show y-scale from 0-100 percent? TRUE by default, if FALSE then scale from 0-1.
title title
smooth "smooth" the VPC (connect bin midpoints) or show bins as rectangular boxes. Default is TRUE.
vpc_theme theme to be used in VPC. Expects list of class vpc_theme created with function vpc_theme()
facet either "wrap", "columns", or "rows"
labeller ggplot2 labeller function to be passed to underlying ggplot object
verbose TRUE or FALSE (default)
vpcdb Boolean whether to return the underlying vpcdb rather than the plot

Details
Creates a VPC plot from observed and simulation survival data

Value
a list containing calculated VPC information, and a ggplot2 object

See Also
sim_data, vpc, vpc_tte, vpc_cens

Examples
## See vpc-docs.ronkeizer.com for more documentation and examples.

## Example for repeated) time-to-event data
## with NONMEM-like data (e.g. simulated using a dense grid)
data(rtte_obs_nm)
data(rtte_sim_nm)

# treat RTTE as TTE, no stratification
vpc_tte(sim = rtte_sim_nm[rtte_sim_nm$sim <= 20,],
        obs = rtte_obs_nm,
        rtte = FALSE,
        sim_cols=list(dv = "dv", idv = "t"), obs_cols=list(idv = "t"))
Index

*Topic** datasets
- rtte_obs_nm, 14
- rtte_sim_nm, 15
- show_default, 15
- show_default_tte, 15
- simple_data, 16

add_noise, 3
add_sim_index_number, 3
add_stratification, 4
auto_bin, 4

bin_data, 5

check_stratification_columns_available, 5
compute_kaplan, 6
compute_kmmc, 6
create_vpc_theme, 7

define_data_columns, 7
draw_params_mvr, 8

loq_perc, 8

new_vpc_theme, 9

pk_iv_Qcmt, 10
pk_oral_Qcmt, 11
plot_vpc, 11

quantile_cens, 12

read_table_nm, 13, 19, 21, 23, 25
replace_list_elements, 14
rtte_obs_nm, 14
rtte_sim_nm, 15

show_default, 15
show_default_tte, 15
sim_data, 12, 16, 20, 22, 24, 26

simple_data, 16
theme_empty, 17
theme_plain, 18
triangle_to_full, 18

vpc, 17, 18, 22, 24, 26
vpc-package, 2
vpc_cat, 12, 20, 21, 24
vpc_cens, 12, 20, 22, 23, 26
vpc_tte, 12, 20, 22, 24, 26
vpc_vpc (vpc), 18