

# Package ‘ggPMX’

June 6, 2019

**Title** 'ggplot2' Based Tool to Facilitate Diagnostic Plots for NLME Models

**Description** At Novartis, we aimed at standardizing the set of diagnostic plots used for modeling activities in order to reduce the overall effort required for generating such plots. For this, we developed a guidance that proposes an adequate set of diagnostics and a toolbox, called 'ggPMX' to execute them. 'ggPMX' is a toolbox that can generate all diagnostic plots at a quality sufficient for publication and submissions using few lines of code.

**Version** 0.9.4

**Maintainer** Amine Gasseem <contact@ag-study.com>

**URL** <https://github.com/ggPMXdevelopment/ggPMX>

**BugReports** <https://github.com/ggPMXdevelopment/ggPMX/issues>

**Depends** R (>= 3.2.3)

**Imports** data.table, yaml, R6, gtable, ggplot2 (>= 2.2.0), magrittr, stringr, assertthat, checkmate, GGally, zoo, knitr, rmarkdown

**License** GPL-2

**LazyData** true

**Suggests** testthat, xtable

**VignetteBuilder** knitr

**NeedsCompilation** no

**RoxygenNote** 6.0.1

**Author** Amine Gasseem [cre, aut],  
Bruno Bieth [aut],  
Irina Baltcheva [aut],  
Thomas Dumortier [aut],  
Christian Bartels [aut],  
Souvik Bhattacharya [aut],  
Inga Ludwig [aut],  
Ines Paule [aut],  
Didier Renard [aut],  
Qing Xi Ooi [ctr],  
Novartis Pharma AG [cph]

Repository CRAN

Date/Publication 2019-06-06 12:00:03 UTC

## R topics documented:

abbrev	4
add_draft	4
distrib	5
eta_cov	6
eta_cov_plot	7
eta_distribution_plot	10
eta_pairs	13
FacetWrapPaginate	14
facet_wrap_paginate	14
getPmxOption	15
get_abbrev	16
get_cats	16
get_conts	17
get_covariates	17
get_data	18
get_occ	18
get_plot	19
get_plot_config	20
get_strats	20
ggPMX	21
gtable_remove_grobs	21
individual	22
input_finegrid	23
is.pmx_gpar	23
load_config	24
load_data_set	24
load_source	25
l_left_join	25
n_pages	26
parse_mlxtran	26
pk_occ	27
pk_pd	27
plots	28
plot_names	28
plot_pmx	29
plot_pmx.distrib	29
plot_pmx.eta_cov	30
plot_pmx.eta_pairs	31
plot_pmx.individual	31
plot_pmx.pmx_dens	32
plot_pmx.pmx_gpar	33
plot_pmx.pmx_qq	33

plot_pmx.residual . . . . .	34
plot_shrink . . . . .	35
pmx . . . . .	35
pmxOptions . . . . .	38
pmx_bloq . . . . .	39
pmx_comp_shrink . . . . .	39
pmx_config . . . . .	40
pmx_copy . . . . .	41
pmx_cov . . . . .	42
pmx_dens . . . . .	43
pmx_endpoint . . . . .	44
pmx_filter . . . . .	45
pmx_get_configs . . . . .	46
pmx_gpar . . . . .	46
pmx_plot . . . . .	47
pmx_plot_cats . . . . .	48
pmx_plot_eta_matrix . . . . .	49
pmx_plot_individual . . . . .	51
pmx_plot_iwres_dens . . . . .	54
pmx_plot_vpc . . . . .	56
pmx_qq . . . . .	58
pmx_qq_plot . . . . .	60
pmx_register_plot . . . . .	62
pmx_report . . . . .	62
pmx_report_template . . . . .	65
pmx_settings . . . . .	66
pmx_sim . . . . .	67
pmx_theme . . . . .	68
pmx_update . . . . .	69
pmx_vpc . . . . .	70
pmx_vpc_bin . . . . .	71
pmx_vpc_ci . . . . .	71
pmx_vpc_obs . . . . .	72
pmx_vpc_pi . . . . .	73
pmx_vpc_rug . . . . .	74
print.abbreviation . . . . .	74
print.configs . . . . .	75
print.pmxClass . . . . .	75
print.pmxConfig . . . . .	76
print.pmx_gpar . . . . .	76
read_input . . . . .	77
read_mlx_ind_est . . . . .	77
read_mlx_par_est . . . . .	78
read_mlx_pred . . . . .	78
residual . . . . .	79
residual_scatter . . . . .	80
set_abbrev . . . . .	83
set_data . . . . .	84

set_plot . . . . .	85
theophylline . . . . .	86
wrap_formula . . . . .	86
[.pmx_gpar . . . . .	87

<b>Index</b>	<b>88</b>
--------------	-----------

---

abbrev	<i>Give the whole abbreviation definition</i>
--------	---

---

### Description

Give the whole abbreviation definition

### Usage

```
abbrev(param)
```

### Arguments

param	abbreviation term
-------	-------------------

### Value

character abbreviation definition

### Examples

```
abbrev("VPC")
```

---

add_draft	<i>Add draft layer annotation</i>
-----------	-----------------------------------

---

### Description

This function adds the word draft to certain graphics.

### Usage

```
add_draft(label = "DRAFT", size = 10, colour = "grey50", x = Inf,
  y = -Inf, ...)
```

**Arguments**

label	draft layer default to DRAFT
size	size of the annotation
colour	color of the annotation default to grey50
x	numeric x coordinate of the draft label
y	numeric y coordinate of the draft label
...	extra parameters to geom text used to annotate the draft

**Value**

ggplot2 annotation

---

distrib *creates a graphic distribution object*

---

**Description**

creates a graphic distribution object

**Usage**

```
distrib(labels, is.shrink, type = c("box", "hist"), is.jitter = FALSE,
        jitter = NULL, facets = NULL, histogram = NULL, shrink = NULL,
        dname = NULL, ...)
```

**Arguments**

labels	list of texts/titles used within the plot
is.shrink	logical if TRUE add shrinkage layer
type	box for boxplot or histogram
is.jitter	logical if TRUE add jitter operator for points
jitter	list set jitter parameter
facets	list set the facet setting in case of histogram plot
histogram	list histogram graphical parameters
shrink	list list of parameters to tune the shrinkage
dname	name of dataset to be used
...	others graphics arguments passed to <a href="#">pmx_gpar</a> internal object.

## Details

**labels** is a list that contains:

- **title:** plot title default "EBE distribution"
- **subtitle:** plot subtitle default empty
- **x:** x axis label default to "Etas"
- **y:** y axis label default to empty
- **legend:** legend title default to "random Effect"

**shrink** is a list that contains:

- **fun:** shrinkage function can be sd or var
- **size:** shrinkage text size
- **color:** shrinkage text color
- **vjust:** shrinkage position vertical adjustment

## Value

distrib object

## See Also

Other plot\_pmx: [eta\\_cov](#), [eta\\_pairs](#), [individual](#), [plot\\_pmx.distrib](#), [plot\\_pmx.eta\\_cov](#), [plot\\_pmx.eta\\_pairs](#), [plot\\_pmx.individual](#), [plot\\_pmx.pmx\\_dens](#), [plot\\_pmx.pmx\\_gpar](#), [plot\\_pmx.pmx\\_qq](#), [plot\\_pmx.residual](#), [plot\\_pmx](#)

---

eta\_cov

*This creates an ETA covariance matrix which can be used to define the co-relation between the parameters and its shrinkage..*

---

## Description

This creates an ETA covariance matrix which can be used to define the co-relation between the parameters and its shrinkage..

## Usage

```
eta_cov(labels, type = c("cats", "conts"), dname = NULL,
        show.correl = TRUE, correl = NULL, facets = NULL, point = NULL,
        covariates = NULL, ...)
```

**Arguments**

labels	list of texts/titles used within the plot
type	box for cats or conts
dname	name of dataset to be used
show.correl	logical if TRUE add correlation to the plot
correl	list correl geom text graphical parameter
facets	list faceting graphical parameter
point	list geom point graphical parameter
covariates	pmxCOVObject <a href="#">pmx_cov</a>
...	others graphics arguments passed to <a href="#">pmx_gpar</a> internal object.

**Details**

**labels** is a list that contains:

- **title:** plot title default "EBE vs. covariates"
- **x:** x axis label default to "Etas"
- **y:** y axis label default to empty

**Value**

eta\_cov object

**See Also**

Other plot\_pmx: [distrib](#), [eta\\_pairs](#), [individual](#), [plot\\_pmx.distrib](#), [plot\\_pmx.eta\\_cov](#), [plot\\_pmx.eta\\_pairs](#), [plot\\_pmx.individual](#), [plot\\_pmx.pmx\\_dens](#), [plot\\_pmx.pmx\\_gpar](#), [plot\\_pmx.pmx\\_qq](#), [plot\\_pmx.residual](#), [plot\\_pmx](#)

---

eta\_cov\_plot

*Eta Covariates plots*

---

**Description**

Eta Covariates plots

Relationships between (ETA) and categorical covariates

Relationships between (ETA) and continuous covariates

**Usage**

```
dummy(dname, show.correl, correl, point, facets, filter, strat.facet,
      strat.color, trans, pmxgpar, labels, axis.title, axis.text, ranges, is.smooth,
      smooth, is.band, band, is.draft, draft, is.identity_line, identity_line,
      scale_x_log10, scale_y_log10, color.scales)
```

```
pmx_plot_eta_cats(ctr, ...)
```

```
pmx_plot_eta_conts(ctr, ...)
```

**Arguments**

dname	character name of dataset to be used
show.correl	logical if TRUE add correlation to the plot
correl	list correl geom text graphical parameter
point	list geom point graphical parameter
facets	list facetting graphical parameter
<b>pmx_update parameters</b>	
filter	expression filter which will be applied to plotting data.
strat.facet	formula optional stratification parameter by facetting. This split plot by strats(each strat in a facet)
strat.color	character optional stratification parameter by grouping. This will split the plot by group (color) of strat.
trans	character define the transformation to apply on x or y or both variables
pmxgpar	a object of class pmx_gpar possibly the output of the <b>pmx_gpar: Shared basic graphics parameters</b>
labels	list list containing plot and/or axis labels: title, subtitle, x , y
axis.title	list containing element_text attributes to customize the axis title. (similar to ggplot2 axis.title theme)
axis.text	list containing element_text attributes to customize the axis text (similar to ggplot2 axis.text theme)
ranges	list limits of x/y ranges
is.smooth	logical if set to TRUE add smooth layer
smooth	list geom_smooth graphical/smoothing fun parameters
is.band	logical if TRUE add horizontal band
band	list horizontal band parameters. geom_hline graphical parameters.
is.draft	logical if TRUE add draft layer
draft	list draft layer parameters. geom_text graphical parameters.
is.identity_line	logical if TRUE add an identity line
identity_line	listgeom_abline graphical parameters.



scale\_x\_log10 logical if TRUE use log10 scale for x axis.  
 scale\_y\_log10 logical if TRUE use log10 scale for y axis.  
 color.scales list define scales parameter in case of strat.color [pmx\\_settings](#)  
 ctr pmx controller  
 ... others graphics parameters passed :
 

- [pmx\\_gpar](#) internal function to customize shared graphical parameters
- [eta\\_cov](#) generic object for eta/covariates plots.
- [pmx\\_update](#) function.

**eta\_cov parameters**

### Value

ggplot2 object

### Examples

```

# basic use -----
ctr <- theophylline()
ctr %>% pmx_plot_eta_cats
ctr %>% pmx_plot_eta_conts

# update graphical parameter -----

## update labels
ctr %>% pmx_plot_eta_cats(
  labels = list(title = "New eta cats title")
)

## remove draft
ctr %>% pmx_plot_eta_cats(is.draft = FALSE)

## change text color line
ctr %>% pmx_plot_eta_conts(
  correl=list(colour="magenta")
)

## set covariates custom labels

ctr %>% pmx_plot_eta_conts(
  covariates=pmx_cov(values=list("WT0", "AGE0"),
    labels=list("Weight", "Age"))
)

## set effects and covaraites custom labels

```

```
ctr <- theophylline( settings = pmx_settings(
  effects=list( levels=c("ka", "V", "Cl"),
               labels=c("Concentration","Volume","Clearance")
             )
)
)
)
ctr %>% pmx_plot_eta_concs(
  covariates=pmx_cov(values=list("WT0","AGE0"),
                    labels=list("Weight","Age"))
)
```

---

eta\_distribution\_plot *Eta distribution plots*

---

### Description

Eta distribution plots

Eta Distribution boxplot

Eta Distribution histogram plot

### Usage

```
eta_distribution_plot(jitter, type, dname, is.shrink, shrink, is.jitter,
  histogram, filter, strat.facet, facets, strat.color, trans, pmxgpar, labels,
  axis.title, axis.text, ranges, is.smooth, smooth, is.band, band, is.draft,
  draft, is.identity_line, identity_line, scale_x_log10, scale_y_log10,
  color.scales, ...)
```

```
pmx_plot_eta_box(ctr, ...)
```

```
pmx_plot_eta_hist(ctr, ...)
```

### Arguments

jitter	list set jitter parameter
type	box for boxplot or histogram
dname	name of dataset to be used
is.shrink	logical if TRUE add shrinkage layer
shrink	list list of parameters to tune the shrinkage
is.jitter	logical if TRUE add jitter operator for points

histogram	list histogram graphical parameters <b>pmx_update parameters</b>
filter	expression filter which will be applied to plotting data.
strat.facet	formula optional stratification parameter by facetting. This split plot by strats(each strat in a facet)
facets	list facet_wrap parameters.
strat.color	character optional stratification parameter by grouping. This will split the plot by group (color) of strat.
trans	character define the transformation to apply on x or y or both variables
pmxgpar	a object of class pmx_gpar possibly the output of the <b>pmx_gpar: Shared basic graphics parameters</b>
labels	list list containing plot and/or axis labels: title, subtitle, x , y
axis.title	list containing element_text attributes to customize the axis title. (similar to ggplot2 axis.title theme)
axis.text	list containing element_text attributes to customize the axis text (similar to ggplot2 axis.text theme)
ranges	list limits of x/y ranges
is.smooth	logical if set to TRUE add smooth layer
smooth	list geom_smooth graphical/smoothing fun parameters
is.band	logical if TRUE add horizontal band
band	list horizontal band parameters. geom_hline graphical parameters.
is.draft	logical if TRUE add draft layer
draft	list draft layer parameters. geom_text graphical parameters.
is.identity_line	logical if TRUE add an identity line
identity_line	listgeom_abline graphical parameters.
scale_x_log10	logical if TRUE use log10 scale for x axis.
scale_y_log10	logical if TRUE use log10 scale for y axis.
color.scales	list define scales parameter in case of strat.color <a href="#">pmx_settings</a>
...	others graphics parameters passed : <ul style="list-style-type: none"> <li>• <a href="#">pmx_gpar</a> internal function to customize shared graphical parameters</li> <li>• <a href="#">distrib</a> generic object for distribution plots (histogram/boxplot).</li> <li>• <a href="#">pmx_update</a> function.</li> </ul> <b>distrib parameters</b>
ctr	pmx controller

**Value**

ggplot2 object

**Examples**

```

# ***** basic use ***** -----

ctr <- theophylline()
## boxplot variation
p <- ctr %>% pmx_plot_eta_box
## histogram variation
p <- ctr %>% pmx_plot_eta_hist()

# update graphical parameter -----

## add jitter
ctr %>%
  pmx_plot_eta_hist(is.jitter = TRUE, jitter = list(alpha = 0.4, color = "red"))

## remove shrinkage
ctr %>%
  pmx_plot_eta_hist(is.shrink = FALSE)

## update histogram graphical parameters
ctr %>%
  pmx_plot_eta_hist(
    histogram = list(
      color = NA,
      position = "fill",
      binwidth = 1 / 100)
  )

# stratification -----

## categorical stratification color parameter
ctr %>% pmx_plot_eta_hist(is.jitter=TRUE, strat.facet=~STUD, strat.color="SEX")
## categorical stratification facetting
ctr %>% pmx_plot_eta_hist(strat.facet = "SEX")
## using formula categorical stratification facetting
ctr %>% pmx_plot_eta_hist(strat.facet = STUD~SEX,
  shrink=list(hjust=0.5))

# subsetting -----

## select a set of random effect
ctr %>% pmx_plot_eta_hist(filter = EFFECT %in% c("ka", "Cl"))
## filter and stratify by facets
ctr %>% pmx_plot_eta_hist(
  filter = EFFECT %in% c("ka", "Cl"), strat.facet = "SEX"
)
ctr %>% pmx_plot_eta_hist(
  filter = EFFECT %in% c("ka", "Cl"), strat.facet = "SEX"
)

```

---

eta_pairs	<i>This creates an eta correlation which defines the relationship between parameters</i>
-----------	--

---

### Description

This creates an eta correlation which defines the relationship between parameters

### Usage

```
eta_pairs(title, dname = NULL, type.eta = c("mode", "mean"),
  text_color = "black", is.shrink = TRUE, is.smooth = TRUE,
  smooth = NULL, point = NULL, shrink = NULL, is.hline = FALSE,
  hline = NULL, ...)
```

### Arguments

title	character the plot title
dname	name of dataset to be used
type.eta	character type of eat can be 'mode' or 'mean'.'mode' by default
text_color	color of the correlation text in the upper matrix
is.shrink	logical if TRUE add shrinkage to the plot
is.smooth	logical if TRUE add smoothing to lower matrix plots
smooth	list geom_smooth graphical parameters
point	list geom_point graphical parameter
shrink	list shrinkage graphical parameter
is.hline	logical if TRUE add horizontal line to lower matrix plots
hline	list geom_hline graphical parameters
...	others graphics arguments passed to <a href="#">pmx_gpar</a> internal object.

### Value

ecorrel object

### See Also

Other plot\_pmx: [distrib](#), [eta\\_cov](#), [individual](#), [plot\\_pmx.distrib](#), [plot\\_pmx.eta\\_cov](#), [plot\\_pmx.eta\\_pairs](#), [plot\\_pmx.individual](#), [plot\\_pmx.pmx\\_dens](#), [plot\\_pmx.pmx\\_gpar](#), [plot\\_pmx.pmx\\_qq](#), [plot\\_pmx.residual](#), [plot\\_pmx](#)

---

FacetWrapPaginate      *Extend facet\_wrap using ggproto*

---

### Description

Extend facet\_wrap using ggproto

### Usage

FacetWrapPaginate

---

facet\_wrap\_paginate      *Split facet\_wrap over multiple plots*

---

### Description

This extension to `facet_wrap` will allow user to split a faceted plot over multiple pages. User define the specific number of rows and columns per page as well as the page number to plot, and the function will automatically plot in the correct panels. This will be rendered in a loop to plot pages one by one.

### Usage

```
facet_wrap_paginate(facets, nrow = NULL, ncol = NULL, scales = "fixed",
  shrink = TRUE, labeller = "label_value", as.table = TRUE,
  switch = NULL, drop = TRUE, dir = "h", strip.position = "top",
  page = 1)
```

### Arguments

facets	A set of variables or expressions quoted by <code>vars()</code> and defining faceting groups on the rows or columns dimension. The variables can be named (the names are passed to labeller). For compatibility with the classic interface, can also be a formula or character vector. Use either a one sided formula, <code>~a + b</code> , or a character vector, <code>c("a", "b")</code> .
nrow	Number of rows and columns.
ncol	Number of rows and columns
scales	Should scales be fixed ("fixed", the default), free ("free"), or free in one dimension ("free_x", "free_y")?
shrink	If TRUE, will shrink scales to fit output of statistics, not raw data. If FALSE, will be range of raw data before statistical summary.

labeller	A function that takes one data frame of labels and returns a list or data frame of character vectors. Each input column corresponds to one factor. Thus there will be more than one with formulae of the type $\sim cyl + am$ . Each output column gets displayed as one separate line in the strip label. This function should inherit from the "labeller" S3 class for compatibility with <code>labeller()</code> . See <code>label_value()</code> for more details and pointers to other options.
as.table	If TRUE, the default, the facets are laid out like a table with highest values at the bottom-right. If FALSE, the facets are laid out like a plot with the highest value at the top-right.
switch	By default, the labels are displayed on the top and right of the plot. If "x", the top labels will be displayed to the bottom. If "y", the right-hand side labels will be displayed to the left. Can also be set to "both".
drop	If TRUE, the default, all factor levels not used in the data will automatically be dropped. If FALSE, all factor levels will be shown, regardless of whether or not they appear in the data.
dir	Direction: either "h" for horizontal, the default, or "v", for vertical.
strip.position	By default, the labels are displayed on the top of the plot. Using <code>strip.position</code> it is possible to place the labels on either of the four sides by setting <code>strip.position = c("top", "bot</code>
page	The page to draw

**Note**

If either `ncol` or `nrow` is NULL this function will fall back to the standard `facet_wrap` functionality.

---

getPmxOption

*Get ggPMX Option*

---

**Description**

Get ggPMX Option

**Usage**

```
getPmxOption(name, default = NULL)
```

**Arguments**

name	Name of an option to get.
default	Value to be returned if the option is not currently set.

**Examples**

```
## Not run:
pmxOptions(myOption = 10)
getPmxOption("myOption")

## End(Not run)
```

---

get_abbrev	<i>Get abbreviation definition by key</i>
------------	---

---

**Description**

Get abbreviation definition by key

**Usage**

```
get_abbrev(ctr, param)
```

**Arguments**

ctr	pmxClass controller
param	abbreviation term

**Value**

character abbreviation definition

---

get_cats	<i>Get category covariates</i>
----------	--------------------------------

---

**Description**

Get category covariates

**Usage**

```
get_cats(ctr)
```

**Arguments**

ctr	the controller object
-----	-----------------------

**Value**

a character vector

**See Also**

Other pmxclass: [get\\_confs](#), [get\\_covariates](#), [get\\_data](#), [get\\_occ](#), [get\\_plot\\_config](#), [get\\_plot](#), [get\\_strats](#), [plot\\_names](#), [plots](#), [pmx\\_update](#), [set\\_data](#), [set\\_plot](#)



---

get_conts	<i>Get continuous covariates</i>
-----------	----------------------------------

---

**Description**

Get continuous covariates

**Usage**

```
get_conts(ctr)
```

**Arguments**

ctr                    the controller object

**Value**

a character vector

**See Also**

Other pmxclass: [get\\_cats](#), [get\\_covariates](#), [get\\_data](#), [get\\_occ](#), [get\\_plot\\_config](#), [get\\_plot](#), [get\\_strats](#), [plot\\_names](#), [plots](#), [pmx\\_update](#), [set\\_data](#), [set\\_plot](#)

---

get_covariates	<i>Get covariates variables</i>
----------------	---------------------------------

---

**Description**

Get covariates variables

**Usage**

```
get_covariates(ctr)
```

**Arguments**

ctr                    the controller object

**Value**

a character vector

**See Also**

Other pmxclass: [get\\_cats](#), [get\\_conts](#), [get\\_data](#), [get\\_occ](#), [get\\_plot\\_config](#), [get\\_plot](#), [get\\_strats](#), [plot\\_names](#), [plots](#), [pmx\\_update](#), [set\\_data](#), [set\\_plot](#)

---

get_data	<i>Get controller data set</i>
----------	--------------------------------

---

**Description**

Get controller data set

**Usage**

```
get_data(ctr, data_set = c("estimates", "predictions", "eta", "finegrid",  
  "input", "sim", "individual"))
```

**Arguments**

ctr	the controller object
data_set	the data set name

**Value**

a data.table of the named data set if available.

**See Also**

Other pmxclass: [get\\_cats](#), [get\\_confs](#), [get\\_covariates](#), [get\\_occ](#), [get\\_plot\\_config](#), [get\\_plot](#), [get\\_strats](#), [plot\\_names](#), [plots](#), [pmx\\_update](#), [set\\_data](#), [set\\_plot](#)

---

get_occ	<i>Get controller occasional covariates</i>
---------	---

---

**Description**

Get controller occasional covariates

**Usage**

```
get_occ(ctr)
```

**Arguments**

ctr	the controller object
-----	-----------------------

**Value**

a character vector

**See Also**

Other pmxclass: [get\\_cats](#), [get\\_conts](#), [get\\_covariates](#), [get\\_data](#), [get\\_plot\\_config](#), [get\\_plot](#), [get\\_strats](#), [plot\\_names](#), [plots](#), [pmx\\_update](#), [set\\_data](#), [set\\_plot](#)

---

get_plot	<i>Get plot object</i>
----------	------------------------

---

**Description**

Get plot object

**Usage**

```
get_plot(ctr, nplot, npage = NULL)
```

**Arguments**

ctr	pmxClass controller object
nplot	character the plot name
npage	integer or integer vector, set page number in case of multi pages plot

**Value**

ggplot object

**See Also**

Other pmxclass: [get\\_cats](#), [get\\_conts](#), [get\\_covariates](#), [get\\_data](#), [get\\_occ](#), [get\\_plot\\_config](#), [get\\_strats](#), [plot\\_names](#), [plots](#), [pmx\\_update](#), [set\\_data](#), [set\\_plot](#)

**Examples**

```
library(ggPMX)
ctr <- theophylline()
p1 <- ctr %>% get_plot("iwres_ipred")
## get all pages or some pages
p2 <- ctr %>% get_plot("individual")
## returns one page of individual plot
p2 <- ctr %>% get_plot("individual",npage=1)
p3 <- ctr %>% get_plot("individual",npage=c(1,3))
## get distribution plot
pdistri <- ctr %>% get_plot("eta_hist")
```

---

get_plot_config	<i>Get the plot config by name</i>
-----------------	------------------------------------

---

**Description**

Get the plot config by name

**Usage**

```
get_plot_config(ctr, pname)
```

**Arguments**

ctr	the controller object
pname	the plot name

**Value**

the config object

**See Also**

Other pmxclass: [get\\_cats](#), [get\\_confs](#), [get\\_covariates](#), [get\\_data](#), [get\\_occ](#), [get\\_plot](#), [get\\_strats](#), [plot\\_names](#), [plots](#), [pmx\\_update](#), [set\\_data](#), [set\\_plot](#)

**Examples**

```
ctr <- theophylline()
ctr %>% set_plot("IND", pname = "indiv1")
ctr %>% get_plot_config("distr1")
```

---

get_strats	<i>Get extra stratification variables</i>
------------	---

---

**Description**

Get extra stratification variables

**Usage**

```
get_strats(ctr)
```

**Arguments**

ctr	the controller object
-----	-----------------------

**Value**

a character vector

**See Also**

Other pmxclass: [get\\_cats](#), [get\\_confs](#), [get\\_covariates](#), [get\\_data](#), [get\\_occ](#), [get\\_plot\\_config](#), [get\\_plot](#), [plot\\_names](#), [plots](#), [pmx\\_update](#), [set\\_data](#), [set\\_plot](#)

---

ggPMX	<i>ggPMX: A ggplot2 toolbox for Nonlinear Mixed-Effect Model graphical</i>
-------	--

---

**Description**

This package aims to generate diagnostic plots in a standard way. The tool reads data from many sources (MONOLIX, NONMEM, others) and generates standard graphics that can be easily integrated in a single report.

**Details**

- Get data from different system and create a data source
- Plot many plots using the generic plot method [plot\\_pmx](#).

For support, feedback or bug reports, please reach out to <[ggPMX\\_ORG@dl.mgd.novartis.com](mailto:ggPMX_ORG@dl.mgd.novartis.com)>.

**Version History**

**Jan 11 2017, 0.0.0** Init ggPMX from Novartis rtemplate.

**Feb 06 2017, 0.3.0** Import version 0.3.0 of package.

---

<code>gtable_remove_grobs</code>	<i>Remove named elements from gtable</i>
----------------------------------	--

---

**Description**

Remove named elements from gtable

**Usage**

```
gtable_remove_grobs(table, names, ...)
```

**Arguments**

<code>table</code>	The table from which grobs should be removed
<code>names</code>	A character vector of the grob names (as listed in <code>table\$layout</code> ) that should be removed
<code>...</code>	Other parameters passed through to <code>gtable_filter</code> .

---

individual	<i>This function can be used to obtain individual prediction and compare with observed data and population prediction for each individual separately</i>
------------	--

---

### Description

This function can be used to obtain individual prediction and compare with observed data and population prediction for each individual separately

### Usage

```
individual(labels, facets = NULL, dname = NULL, ipred_line = NULL,
  pred_line = NULL, point = NULL, bloq = NULL, is.legend, use.finegrid,
  ...)
```

### Arguments

labels	plot texts. labels, axis,
facets	list facets settings nrow/ncol
dname	name of dataset to be used
ipred_line	list some pred line geom properties aesthetics
pred_line	list some ipred line geom properties aesthetics
point	list some point geom properties aesthetics
bloq	pmxBLOQ object created by <a href="#">pmx_bloq</a>
is.legend	logical if TRUE add a legend
use.finegrid	logical if FALSE use predictions data set
...	others graphics arguments passed to <a href="#">pmx_gpar</a> internal object.

### Value

individual fit object

### See Also

[plot\\_pmx.individual](#)

Other plot\_pmx: [distrib](#), [eta\\_cov](#), [eta\\_pairs](#), [plot\\_pmx.distrib](#), [plot\\_pmx.eta\\_cov](#), [plot\\_pmx.eta\\_pairs](#), [plot\\_pmx.individual](#), [plot\\_pmx.pmx\\_dens](#), [plot\\_pmx.pmx\\_gpar](#), [plot\\_pmx.pmx\\_qq](#), [plot\\_pmx.residual](#), [plot\\_pmx](#)

---

input_finegrid	<i>Merge input and finegrid data sets</i>
----------------	---

---

**Description**

Merge input and finegrid data sets

**Usage**

```
input_finegrid(input, finegrid)
```

**Arguments**

input	data.table input data set
finegrid	data.table finegrid data set

**Value**

data.table

---

is.pmx_gpar	<i>Check if an object is a pmx_gpar class</i>
-------------	---

---

**Description**

Check if an object is a pmx\_gpar class

**Usage**

```
is.pmx_gpar(x)
```

**Arguments**

x	pmx_gpar object
---	-----------------

**Value**

logical returns TRUE if it is a pmx\_gpar object

---

load_config	<i>Obtain the data source config</i>
-------------	--------------------------------------

---

**Description**

Obtain the data source config

**Usage**

```
load_config(x, sys = c("mlx", "nm", "mlx18"))
```

**Arguments**

x	the config name.
sys	can be mlx,nm,...

**Value**

a list :data configuration object

---

load_data_set	<i>Load data set</i>
---------------	----------------------

---

**Description**

Load data set

**Usage**

```
load_data_set(x, path, sys, ...)
```

**Arguments**

x	data set config
path	character path to the directory
sys	character mlx or nm
...	extra parameter passed to special readers

**Value**

data.table



---

load_source	<i>Load all/or some source data set</i>
-------------	---

---

**Description**

Load all/or some source data set

**Usage**

```
load_source(sys, path, dconf, ...)
```

**Arguments**

sys	type cane mlx/nom
path	character directory path containing all sources.
dconf	configuration object
...	any extra parameters for readers

**Value**

list of data.table

---

l_left_join	<i>Merge 2 lists</i>
-------------	----------------------

---

**Description**

left join , the first list is updated by the second one

**Usage**

```
l_left_join(base_list, overlay_list, recursive = TRUE)
```

**Arguments**

base_list	list to update
overlay_list	list used to update the first list
recursive	logical if TRUE do the merge in depth

**Value**

list

---

n_pages	<i>Determine the number of pages in a paginated facet plot</i>
---------	--

---

**Description**

This is a simple helper that returns the number of pages it takes to plot all panels when using `facet_wrap_paginate`. It partially builds the plot so depending on the complexity of your plot it might take some time to calculate...

**Usage**

```
n_pages(plot)
```

**Arguments**

plot            A ggplot object using either `facet_wrap_paginate` or `facet_grid_paginate`

**Value**

If the plot uses using either `facet_wrap_paginate` or `facet_grid_paginate` it returns the total number of pages. Otherwise it returns NULL

---

parse_mlxtran	<i>Parse MONOLIX mlxtran file</i>
---------------	-----------------------------------

---

**Description**

Parse MONOLIX mlxtran file

**Usage**

```
parse_mlxtran(file_name)
```

**Arguments**

file\_name        absolute path to mlxtran file

**Value**

list key/values to initialize ggPMX controller

---

pk_occ	<i>Creates pmx controller using monlix data having Occasional variable</i>
--------	--

---

**Description**

Creates pmx controller using monlix data having Occasional variable

**Usage**

```
pk_occ()
```

**Value**

pmx controller

**Examples**

```
## Not run:  
pk_occ()  
  
## End(Not run)
```

---

pk_pd	<i>Creates pkpd pmx controller using package internal data</i>
-------	--

---

**Description**

Creates pkpd pmx controller using package internal data

**Usage**

```
pk_pd(code = "3")
```

**Arguments**

code                    can be 3 or 4

---

plots	<i>Get plots description</i>
-------	------------------------------

---

**Description**

Get plots description

**Usage**

```
plots(ctr)
```

**Arguments**

ctr                   pmxClass controller object

**Value**

data.frame of plots

**See Also**

Other pmxclass: [get\\_cats](#), [get\\_confs](#), [get\\_covariates](#), [get\\_data](#), [get\\_occ](#), [get\\_plot\\_config](#), [get\\_plot](#), [get\\_strats](#), [plot\\_names](#), [pmx\\_update](#), [set\\_data](#), [set\\_plot](#)

---

plot_names	<i>Get plot names</i>
------------	-----------------------

---

**Description**

Get plot names

**Usage**

```
plot_names(ctr)
```

**Arguments**

ctr                   pmxClass controller object

**Value**

list of plot names

**See Also**

Other pmxclass: [get\\_cats](#), [get\\_confs](#), [get\\_covariates](#), [get\\_data](#), [get\\_occ](#), [get\\_plot\\_config](#), [get\\_plot](#), [get\\_strats](#), [plots](#), [pmx\\_update](#), [set\\_data](#), [set\\_plot](#)

---

plot_pmx	<i>This is a generic plot method that produces all plots by default described in pmx model evaluation guidance.</i>
----------	---

---

### Description

This is a generic plot method that produces all plots by default described in pmx model evaluation guidance.

### Usage

```
plot_pmx(x, dx, ...)
```

### Arguments

x	object to plot
dx	data.table , plot source data
...	extra argument (not used)

### See Also

[pmx\\_gpar](#).

Other plot\_pmx: [distrib](#), [eta\\_cov](#), [eta\\_pairs](#), [individual](#), [plot\\_pmx.distrib](#), [plot\\_pmx.eta\\_cov](#), [plot\\_pmx.eta\\_pairs](#), [plot\\_pmx.individual](#), [plot\\_pmx.pmx\\_dens](#), [plot\\_pmx.pmx\\_gpar](#), [plot\\_pmx.pmx\\_qq](#), [plot\\_pmx.residual](#)

---

plot_pmx.distrib	<i>Plot EBE distribution</i>
------------------	------------------------------

---

### Description

Plot EBE distribution

### Usage

```
## S3 method for class 'distrib'
plot_pmx(x, dx, ...)
```

### Arguments

x	distribution object
dx	data set
...	not used for the moment

**Value**

ggplot2 plot

**See Also**

[distrib](#)

Other plot\_pmx: [distrib](#), [eta\\_cov](#), [eta\\_pairs](#), [individual](#), [plot\\_pmx.eta\\_cov](#), [plot\\_pmx.eta\\_pairs](#), [plot\\_pmx.individual](#), [plot\\_pmx.pmx\\_dens](#), [plot\\_pmx.pmx\\_gpar](#), [plot\\_pmx.pmx\\_qq](#), [plot\\_pmx.residual](#), [plot\\_pmx](#)

---

plot_pmx.eta_cov	<i>This plots an ETA covariance matrix which can be used to define the co-relation between the parameters and its shrinkage</i>
------------------	---

---

**Description**

This plots an ETA covariance matrix which can be used to define the co-relation between the parameters and its shrinkage

**Usage**

```
## S3 method for class 'eta_cov'
plot_pmx(x, dx, ...)
```

**Arguments**

x	eta_cov object
dx	data set
...	not used for the moment

**Value**

ggplot2 plot

**See Also**

[eta\\_cov](#)

Other plot\_pmx: [distrib](#), [eta\\_cov](#), [eta\\_pairs](#), [individual](#), [plot\\_pmx.distrib](#), [plot\\_pmx.eta\\_pairs](#), [plot\\_pmx.individual](#), [plot\\_pmx.pmx\\_dens](#), [plot\\_pmx.pmx\\_gpar](#), [plot\\_pmx.pmx\\_qq](#), [plot\\_pmx.residual](#), [plot\\_pmx](#)

---

plot\_pmx.eta\_pairs      *Plot random effect correlation plot*

---

**Description**

Plot random effect correlation plot

**Usage**

```
## S3 method for class 'eta_pairs'
plot_pmx(x, dx, ...)
```

**Arguments**

x	distribution object
dx	data set
...	not used for the moment

**Value**

ggpairs plot

**See Also**

[distrib](#)

Other plot\_pmx: [distrib](#), [eta\\_cov](#), [eta\\_pairs](#), [individual](#), [plot\\_pmx.distrib](#), [plot\\_pmx.eta\\_cov](#), [plot\\_pmx.individual](#), [plot\\_pmx.pmx\\_dens](#), [plot\\_pmx.pmx\\_gpar](#), [plot\\_pmx.pmx\\_qq](#), [plot\\_pmx.residual](#), [plot\\_pmx](#)

---

plot\_pmx.individual      *This function can be used to plot individual prediction and compare with observed data and population prediction for each individual separately*

---

**Description**

This function can be used to plot individual prediction and compare with observed data and population prediction for each individual separately

**Usage**

```
## S3 method for class 'individual'
plot_pmx(x, dx, ...)
```

**Arguments**

x	individual object
dx	data set
...	not used for the moment

**Value**

a list of ggplot2

**See Also**

Other plot\_pmx: [distrib](#), [eta\\_cov](#), [eta\\_pairs](#), [individual](#), [plot\\_pmx.distrib](#), [plot\\_pmx.eta\\_cov](#), [plot\\_pmx.eta\\_pairs](#), [plot\\_pmx.pmx\\_dens](#), [plot\\_pmx.pmx\\_gpar](#), [plot\\_pmx.pmx\\_qq](#), [plot\\_pmx.residual](#), [plot\\_pmx](#)

---

plot_pmx.pmx_dens	<i>This function plot EBE versus covariates using qq plots</i>
-------------------	--

---

**Description**

This function plot EBE versus covariates using qq plots

**Usage**

```
## S3 method for class 'pmx_dens'
plot_pmx(x, dx, ...)
```

**Arguments**

x	eta_cov object
dx	data set
...	not used for the moment

**Value**

ggplot2 plot

**See Also**

[eta\\_cov](#)

Other plot\_pmx: [distrib](#), [eta\\_cov](#), [eta\\_pairs](#), [individual](#), [plot\\_pmx.distrib](#), [plot\\_pmx.eta\\_cov](#), [plot\\_pmx.eta\\_pairs](#), [plot\\_pmx.individual](#), [plot\\_pmx.pmx\\_gpar](#), [plot\\_pmx.pmx\\_qq](#), [plot\\_pmx.residual](#), [plot\\_pmx](#)



---

plot\_pmx.pmx\_gpar      *The ggPMX base plot function*

---

### Description

This function should be called internally by other plots to set general settings like , smoothing, add band, labelling, theming,...

### Usage

```
## S3 method for class 'pmx_gpar'
plot_pmx(gpar, p)
```

### Arguments

gpar	object of pmx_gpar type
p	plot

### Value

ggplot2 object

### See Also

Other plot\_pmx: [distrib](#), [eta\\_cov](#), [eta\\_pairs](#), [individual](#), [plot\\_pmx.distrib](#), [plot\\_pmx.eta\\_cov](#), [plot\\_pmx.eta\\_pairs](#), [plot\\_pmx.individual](#), [plot\\_pmx.pmx\\_dens](#), [plot\\_pmx.pmx\\_qq](#), [plot\\_pmx.residual](#), [plot\\_pmx](#)

---

plot\_pmx.pmx\_qq      *This function plot EBE versus covariates using qq plots*

---

### Description

This function plot EBE versus covariates using qq plots

### Usage

```
## S3 method for class 'pmx_qq'
plot_pmx(x, dx, ...)
```

### Arguments

x	pmx_qq object
dx	data set
...	not used for the moment

**Value**

ggplot2 plot

**See Also**

[eta\\_cov](#)

Other plot\_pmx: [distrib](#), [eta\\_cov](#), [eta\\_pairs](#), [individual](#), [plot\\_pmx.distrib](#), [plot\\_pmx.eta\\_cov](#), [plot\\_pmx.eta\\_pairs](#), [plot\\_pmx.individual](#), [plot\\_pmx.pmx\\_dens](#), [plot\\_pmx.pmx\\_gpar](#), [plot\\_pmx.residual](#), [plot\\_pmx](#)

---

plot_pmx.residual	<i>This function plots residual for each observed value by finding the difference between observed and predicted points. It also fits a distribution to the residual value.</i>
-------------------	---

---

**Description**

This function plots residual for each observed value by finding the difference between observed and predicted points. It also fits a distribution to the residual value.

**Usage**

```
## S3 method for class 'residual'
plot_pmx(x, dx, ...)
```

**Arguments**

x	residual object
dx	data set
...	not used for the moment

**Value**

ggplot2 object

**See Also**

[residual](#)

Other plot\_pmx: [distrib](#), [eta\\_cov](#), [eta\\_pairs](#), [individual](#), [plot\\_pmx.distrib](#), [plot\\_pmx.eta\\_cov](#), [plot\\_pmx.eta\\_pairs](#), [plot\\_pmx.individual](#), [plot\\_pmx.pmx\\_dens](#), [plot\\_pmx.pmx\\_gpar](#), [plot\\_pmx.pmx\\_qq](#), [plot\\_pmx](#)

---

plot_shrink	<i>Plot shrink in eta matrix</i>
-------------	----------------------------------

---

**Description**

Plot shrink in eta matrix

**Usage**

```
plot_shrink(x, shrink.dx, shrink)
```

**Arguments**

x	pmx_gpar object
shrink.dx	data.table of shrinkage
shrink	list graphical parameter

**Value**

ggplot2 object

---

pmx	<i>Create a pmx object</i>
-----	----------------------------

---

**Description**

Create a pmx object from a data source

Create a controller from mlxtran file

**Usage**

```
pmx(config, sys = c("mlx", "nm"), directory, input, dv, dvid, cats = NULL,
     conts = NULL, occ = NULL, strats = NULL, settings = NULL,
     endpoint = NULL, sim = NULL, bloq = NULL)
```

```
pmx_mlx(config, directory, input, dv, dvid, cats, conts, occ, strats, settings,
        endpoint, sim, bloq)
```

```
pmx_mlxtran(file_name, config = "standing", call = FALSE, endpoint, ...)
```

**Arguments**

config	Can be either : The complete path for the configuration file, the name of configuration within the built-in list of configurations, or a configuration object.
sys	the system name can be MLX/NM
directory	character modelling output directory.
input	character complete path to the modelling input file
dv	character the name of measurable variable used in the input modelling file
dvid	<i>[Optional]</i> character observation type parameter. This is mandatory in case of multiple endpoint (PKPD).
cats	<i>[Optional]</i> character vector of categorical covariates
conts	<i>[Optional]</i> character vector of continuous covariates
occ	<i>[Optional]</i> character occasional covariate variable name
strats	<i>[Optional]</i> character extra stratification variables
settings	<i>[Optional]</i> pmxSettingsClass <a href="#">pmx_settings</a> shared between all plots
endpoint	pmxEndpointClass or integer or character default to NULL of the endpoint code. <a href="#">pmx_endpoint</a>
sim	pmxSimClass default to NULL. <a href="#">pmx_sim</a>
bloq	pmxBLOQClass default to NULL. <a href="#">pmx_bloq</a>
file_name	character mlxtran file path.
call	logical if TRUE the result is the parameters parsed
...	extra arguments passed to pmx_mlx.

**Details**

pmx\_mlx is a wrapper to mlx for the MONOLIX system ( sys="mlx" )

pmx\_mlxtran parses mlxtran file and guess [pmx\\_mlx](#) arguments. In case of multi endpoint the first endpoint is selected. You can though set the endpoint through the same argument. When you set call=TRUE, no controller is created but only the parameters parsed by mlxtran. This can be very helpful, in case you would like to customize parameters (adding settings vi [pmx\\_settings](#), [chnag](#) [eth](#) [edefault](#) [endpoint](#).)

**Value**

pmxClass controller object.

**Examples**

```
## Example to create the controller using theophylline data
theophylline <- file.path(system.file(package = "ggPMX"), "testdata",
                          "theophylline")
WORK_DIR <- file.path(theophylline, "Monolix")
input_file <- file.path(theophylline, "data_pk.csv")
```

```

## using only mandatory variables
ctr <- pmx(
  sys="mlx",
  config = "standing",
  directory = WORK_DIR,
  input = input_file,
  dv = "Y",
  dvid ="DVID"
)
## Using covariates
ctr <- pmx(
  sys="mlx",
  config = "standing",
  directory = WORK_DIR,
  input = input_file,
  dv = "Y",
  dvid ="DVID",
  cats=c("SEX"),
  conts=c("WT0", "AGE0"),
  strats="STUD"
)
## using settings parameter
ctr <- pmx(
  sys="mlx",
  config = "standing",
  directory = WORK_DIR,
  input = input_file,
  dv = "Y",
  dvid ="DVID",
  settings=list(is.draft=FALSE)
)

## using mlxtran file
mlxtran_file <-
  file.path(system.file(package = "ggPMX"),
    "testdata", "1_popPK_model", "project.mlxtran")
pmx_mlxtran(mlxtran_file)

## mlxtran , call =TRUE to get the pmx_mlx argument parsed by pmx_mlxtran
params <- pmx_mlxtran(mlxtran_file,call=TRUE)

str(params)
# $ directory: chr results_pathile
# $ input    : chr observation file path
# $ dv       : chr "DV"
# $ cats     : chr [1:4] "SEX" "RACE" "DISE" "ILOW"
# $ conts    : chr [1:4] "AGE0" "WT0" "HT0" "TRT"
# $ occ      : chr "ISS"
# $ dvid     : chr "YTYPE"
# $ endpoint :List of 5
# ..$ code   : chr "1"
# ..$ label  : chr ""

```

```
# ..$ unit      : chr ""
# ..$ file.code: chr "1"
# ..$ trans     : NULL
# ..- attr(*, "class")= chr "pmxEndpointClass"
# $ config     : chr "standing"
```

---

pmxOptions

*This function can be used to set ggPMX options*

---

## Description

getPmxOption retrieves the value of a ggPMX option. ggPMXOptions sets the value of ggPMX options; it can also be used to return a list of all currently-set ggPMX options.

## Usage

```
pmxOptions(...)
```

## Arguments

... Options to set, with the form name = value.

## Details

There is a global option set, which is available by default.

## Options used in ggPMX

- **template\_dir**: path to template directory

## Examples

```
## Not run:
pmxOptions(template_dir=PATH_TO_CUSTOM_CONFIGURATION)

## End(Not run)
```

---

pmx\_bloq                      *Creates BLOQ object attributes*

---

### Description

Creates BLOQ object attributes

### Usage

```
pmx_bloq(cens = "CENS", limit = "LIMIT", colour = "pink", size = 2,
         alpha = 0.9, show = TRUE, ...)
```

### Arguments

cens	character the censoring column name
limit	character the limit column name (optional)
colour	character the color of the geom
size	numeric the size of the geom
alpha	numeric the alpha of the geom
show	logical if FALSE remove all censory observations
...	any other graphical parameter

### Details

To define that a measurement is censored, the observation data set should include a CENSORING column ( default to 'CENS' ) and put 1 for lower limit or -1 for upper limit. Optionally, data set can contain have a limit column ( default to 'LIMIT' ) column to set the other limit.

---

pmx\_comp\_shrink              *Compute Shrinkage*

---

### Description

Compute Shrinkage

### Usage

```
pmx_comp_shrink(ctr, fun = c("sd", "var"), strat.facet, strat.color, filter,
                ...)
```

**Arguments**

ctr	pmxClass controller object
fun	character can be sd or var
strat.facet	formula optional stratification parameter
strat.color	character optional stratification parameter
filter	optional filter which will be applied to plotting data
...	others parameters not used for the moment

**Value**

data.table

---

pmx_config	<i>This function can be used to define the pmx configuration used in plots. e.g. Monolox/Nonmem</i>
------------	---

---

**Description**

This function can be used to define the pmx configuration used in plots. e.g. Monolox/Nonmem

**Usage**

```
pmx_config(sys = "mlx", inputs, plots, ...)
```

**Arguments**

sys	character system used , monolix,nonmem,...
inputs	character path to the inputs settings file (yaml format)
plots	character path to the inputs settings file (yaml format)
...	extra arguments not used

**Details**

To create a controller user can create a pmxConfig object using

- either an input template file
- or a plot template file
- or both.

By default the 'standing' configuration will be used.

**Value**

pmxConfig object



**Examples**

```
# ***** Create a controller using custom plot configuration ***** -----

library(ggPMX)
theophylline <- file.path(
  system.file(package = "ggPMX"), "testdata",
  "theophylline"
)
WORK_DIR <- file.path(theophylline, "Monolix")
input_file <- file.path(theophylline, "data_pk.csv")

# create a controller with a custom plots template
ctr <- pmx_mlx(
  config = pmx_config(
    plots=file.path( system.file(package = "ggPMX"),"examples/plots.yaml"),
    inputs = system.file(package = "ggPMX","examples/custom_inputs.yaml")
  ),
  directory = WORK_DIR,
  input = input_file,
  dv = "Y",
  dvid = "DVID",
  cats = c("SEX"),
  conts = c("WT0", "AGE0"),
  strats = "STUD"
)

## get the list of plots
ctr %>% plots
ctr %>% get_plot("custom_res_time")
ctr %>% get_plot("custom_npde_time")
```

---

p<sub>mx</sub>\_copy

*Creates a deep copy of the controller*


---

**Description**

Creates a deep copy of the controller

**Usage**

```
pmx_copy(ctr, keep_globals = FALSE, ...)
```

**Arguments**

ctr	p <sub>mx</sub> Class object
keep_globals	logical if TRUE we keep the global parameters changed by p <sub>mx</sub> _settings
...	extra parameters passed to p <sub>mx</sub> _settings

**Details**

The controller is an 'R6' object, it behaves like a reference object. Some functions ( methods) can have a side effect on the controller and modify it internally. Technically speaking we talk about chaining not piping here. However , using pmx\_copy user can work on a copy of the controller.

By default the copy don't keep global parameters setted using pmx\_settings.

**Value**

an object of pmxClass

**Examples**

```
ctr <- theophylline()
cctr <- ctr %>% pmx_copy
## Any change in the ctr has no side effect in the ctr and vice versa
```

---

pmx\_cov

*Select/Map covariates using human labels*

---

**Description**

Select/Map covariates using human labels

**Usage**

```
pmx_cov(values, labels = NULL)
```

**Arguments**

values	list of covariates to use to create the plot
labels	list of covariates facets labels

**Details**

In case of 'pmx\_plot\_eta\_cats' and 'pmx\_plot\_eta\_conts' you can customize the covariates and covaraites labels using 'pmx\_cov'.

**Value**

pmxCOVobject object

---

pmx_dens	<i>Creates a density plot object</i>
----------	--------------------------------------

---

### Description

Creates a density plot object

### Usage

```
pmx_dens(x, labels, dname = NULL, xlim = 3, var_line = NULL,  
         snd_line = NULL, vline = NULL, ...)
```

### Arguments

x	character variable name to sample
labels	list of texts/titles used within the plot
dname	name of dataset to be used
xlim	numeric x axis limits
var_line	list variable density graphics parameters
snd_line	list normal density graphics parameters
vline	list vertical line graphics parameters
...	others graphics arguments passed to <a href="#">pmx_gpar</a> internal object.

### Details

**labels** is a list that contains:

- **title:** plot title default "IWRES density plot"
- **x:** x axis label default to "Etas"
- **y:** y axis label default to empty

**var\_line** is a list that contains:

- **linetype:** default to 1
- **color:** default to black
- **size:** default to 1

**snd\_line** is a list that contains:

- **linetype:** default to 2
- **color:** default to black
- **size:** default to 1

**vline** is a list that contains:

- **linetype:** default to 3
- **color:** default to black
- **size:** default to 1

---

pmx_endpoint	<i>Creates pmx endpoint object</i>
--------------	------------------------------------

---

### Description

Creates pmx endpoint object

### Usage

```
pmx_endpoint(code, label = "", unit = "", file.code = code,
             trans = NULL)
```

### Arguments

code	character endpoint code : used to filter observations DVID==code.
label	character endpoint label: used to set title and axis labels
unit	character endpoint unit : used to set title and axis labels
file.code	character endpoint file code : used to set predictions and finegrid files extensions in case using code parameter is not enough.
trans	list Transformation parameter not used yet.

### Details

In case of multiple endpoints, pkpd case for example, we need to pass endpoint to the pmx call. Internally , ggPMX will filter the observations data set to keep only rows satisfying DVID==code. The code is also used to find the right predictions and or finegrid files. ggPMX use the configuration file to find the path of the predictions file (like the single endpoint case) and then filter the right file using the code parameter.

For example:

- predictions{code}.txt for mlx16
- predictions{code}.txt and y{code}\_residual for mlx18

For some tricky examples the code parameter is not enough to find the files. In that case the file.code parameter is used to distinguish the endpoint files.

### Examples

```
## Use file.code parameter
pk_pd_path <- file.path(system.file(package = "ggPMX"), "testdata", "pk_pd")

WORK_DIR <- file.path(pk_pd_path, "RESULTS")

ep <- pmx_endpoint(
  code="4",
  file.code="2"
```

```
)  
  
input_file <- file.path(pk_pd_path, "pk_pd.csv")  
  
ctr <- pmx_mlx(  
  config = "standing",  
  directory = WORK_DIR,  
  input = input_file,  
  dv = "dv",  
  dvid = "dvid",  
  cats = "sex",  
  conts = "wt",  
  endpoint = ep  
)  
  
## using mlxtran  
  
ep <- pmx_endpoint(  
  code="3",  
  file.code="1"  
)  
  
mlxtran_file <- file.path(pk_pd_path, "pk_pd.mlxtran")  
ctr <- pmx_mlxtran(mlxtran_file,endpoint=ep)
```

---

pmx\_filter

*filter data in a pmx controller*

---

## Description

filter data in a pmx controller

## Usage

```
pmx_filter(ctr, data_set = c("estimates", "predictions", "eta", "finegrid",  
  "shrink", "input", "individual"), pmx_exp)
```

## Arguments

ctr	A controller. An object of 'pmxClass'
data_set	A data_set within the controller to apply a filter to.
pmx_exp	A filter expression

## Value

Returns a pmx controller with a filtered data set.

**Examples**

```
## example of global filter
ctr <- theophylline()
ctr %>% pmx_filter(data_set = "prediction", ID == 5 & TIME <2)
ctr %>% get_data("prediction")
```

---

pmx\_get\_configs      *Get List of built-in configurations*

---

**Description**

Get List of built-in configurations

**Usage**

```
pmx_get_configs(sys = "mlx")
```

**Arguments**

sys                      can be mlx, by default all configurations will be listed

**Value**

names of the config

**Examples**

```
pmx_get_configs()
```

---

pmx\_gpar                      *Handling pmx Graphical parameters*

---

**Description**

Handling pmx Graphical parameters

**Usage**

```
pmx_gpar(labels, axis.title, axis.text, ranges, is.smooth, smooth, is.band,
band, is.draft, draft, discrete, is.identity_line, identity_line,
scale_x_log10, scale_y_log10, color.scales, is.legend)
```

**Arguments**

labels	list of labels, like title, subtitle, x , y
axis.title	list or element_text (same as ggplot2 axis.title theme)
axis.text	list or element_text (same as ggplot2 axis.text theme)
ranges	limits of x/y ranges
is.smooth	logical if set to TRUE add smooth layer
smooth	smooth layer parameters
is.band	logical if TRUE add horizontal band
band	horizontal band parameters
is.draft	logical if TRUE add draft layer
draft	draft layer parameters
discrete	logical if TRUE x axis is discrete(FALSE by default)
is.identity_line	logical if TRUE add y=x line
identity_line	list y=x aes properties
scale_x_log10	logical if TRUE add scale_x_log10 layer
scale_y_log10	logical if TRUE add scale_y_log10 layer
color.scales	list define scales parameter in case of strat.color <a href="#">pmx_settings</a>
is.legend	logical if TRUE x axis is discrete(FALSE by default)

**Details**

This object contains all general graphic settings. It used internally by all pmx\_plot(generic function) to set the default behavior.

**Value**

An object of class "pmx\_gpar".

---

pmx\_plot

*Generic pmx plot*

---

**Description**

Generic pmx plot

**Usage**

```
pmx_plot(ctr, pname, ...)
```

**Arguments**

ctr	pmxClass pmx controller
pname	plot name
...	others graphics parameters passed : <ul style="list-style-type: none"> <li>• <a href="#">pmx_gpar</a> internal function to customize shared graphical parameters</li> <li>• <a href="#">pmx_qq</a> quantile-quantile plot object</li> <li>• <a href="#">pmx_update</a> function.</li> </ul>

---

pmx_plot_cats	<i>Generic pmx stratified plot</i>
---------------	------------------------------------

---

**Description**

Generic pmx stratified plot

**Usage**

```
pmx_plot_cats(ctr, pname, cats, chunk = "", print = TRUE, ...)
```

**Arguments**

ctr	pmxClass pmx controller
pname	plot name
cats	list of categorical variables. By default all of them
chunk	chunk name
print	logical if TRUE print plots otherwise the list of plots is returned
...	others graphics parameters passed : <ul style="list-style-type: none"> <li>• <a href="#">pmx_gpar</a> internal function to customize shared graphical parameters</li> <li>• <a href="#">pmx_qq</a> quantile-quantile plot object</li> <li>• <a href="#">pmx_update</a> function.</li> </ul>



---

pmx\_plot\_eta\_matrix     *Eta matrix plot*

---

## Description

Eta matrix plot

## Usage

```
pmx_plot_eta_matrix(ctr, title, dname, type.eta, text_color, is.shrink, shrink,
  point, is.smooth, smooth, is.hline, hline, filter, strat.facet, facets,
  strat.color, trans, pmxgpar, labels, axis.title, axis.text, ranges, is.band,
  band, is.draft, draft, is.identity_line, identity_line, scale_x_log10,
  scale_y_log10, color.scales, ...)
```

## Arguments

ctr	pmx controller
title	character the plot title
dname	name of dataset to be used
type.eta	character type of eat can be 'mode' or 'mean'. 'mode' by default
text_color	color of the correlation text in the upper matrix
is.shrink	logical if TRUE add shrinkage to the plot
shrink	list shrinkage graphical parameter
point	list geom_point graphical parameter
is.smooth	logical if TRUE add smoothing to lower matrix plots
smooth	list geom_smooth graphical parameters
is.hline	logical if TRUE add horizontal line to lower matrix plots
hline	list geom_hline graphical parameters
	<b>pmx_update parameters</b>
filter	expression filter which will be applied to plotting data.
strat.facet	formula optional stratification parameter by facetting. This split plot by strats(each strat in a facet)
facets	list facet_wrap parameters.
strat.color	character optional stratification parameter by grouping. This will split the plot by group (color) of strat.
trans	character define the transformation to apply on x or y or both variables
pmxgpar	a object of class pmx_gpar possibly the output of the <b>pmx_gpar: Shared basic graphics parameters</b>
labels	list list containing plot and/or axis labels: title, subtitle, x , y

axis.title	list containing element_text attributes to customize the axis title. (similar to ggplot2 axis.title theme)
axis.text	list containing element_text attributes to customize the axis text (similar to ggplot2 axis.text theme)
ranges	list limits of x/y ranges
is.band	logical if TRUE add horizontal band
band	list horizontal band parameters. geom_hline graphical parameters.
is.draft	logical if TRUE add draft layer
draft	list draft layer parameters. geom_text graphical parameters.
is.identity_line	logical if TRUE add an identity line
identity_line	listgeom_abline graphical parameters.
scale_x_log10	logical if TRUE use log10 scale for x axis.
scale_y_log10	logical if TRUE use log10 scale for y axis.
color.scales	list define scales parameter in case of strat.color <a href="#">pmx_settings</a>
...	others graphics parameters passed : <ul style="list-style-type: none"> <li>• <a href="#">pmx_gpar</a> internal function to customize shared graphical parameters</li> <li>• <a href="#">eta_pairs</a> ggPMX internal function for eta matrix plot.</li> <li>• <a href="#">pmx_update</a> function.</li> </ul>

**eta\_pairs parameters**

**Value**

ggplot2 object

**Examples**

```
# basic use -----
ctr <- theophylline()
p <- ctr %>% pmx_plot_eta_matrix

# update graphical parameter -----

## update labels
ctr %>% pmx_plot_eta_matrix(
  labels = list(title = "Eta matrix new title")
)

## remove draft
ctr %>% pmx_plot_eta_matrix(is.draft = FALSE)

## change text color line
```

```

ctr %>% pmx_plot_eta_matrix(
  text_color="red",
  shrink=list(mapping=aes(color="magenta"))
)

## custom point aes and static parameters
## we can customize any geom_point parameter
ctr %>% pmx_plot_eta_matrix(
  point = list(color = "blue", shape = 4)
)

# stratification -----

## IGNORE continuous stratification
ctr %>% pmx_plot_eta_matrix(strat.color = "WT0")
## IGNORE categorical stratification
ctr %>% pmx_plot_eta_matrix(strat.facet = "SEX")

# subsetting -----

## we can use any expression involving the data
ctr %>% pmx_plot_eta_matrix(filter = EFFECT%in% c("Cl","ka"))

```

---

pmx\_plot\_individual    *Individual plot*

---

## Description

Individual plot

## Usage

```

pmx_plot_individual(ctr, npage = 1, print = FALSE, dname, pred_line,
  ipred_line, point, is.legend, use.finegrid, bloq, filter, strat.facet, facets,
  strat.color, trans, pmxgpar, labels, axis.title, axis.text, ranges, is.smooth,
  smooth, is.band, band, is.draft, draft, is.identity_line, identity_line,
  scale_x_log10, scale_y_log10, color.scales, ...)

```

## Arguments

ctr	pmx controller
npage	integer page(s) to display , set npage to NULL
print	logical if TRUE the ouptut will be a print not a ggplot2. This is useful for rmarkdwon output to avoid verbose list index print.
dname	character name of dataset to be used. User can create his own dataset using <a href="#">set_data</a> and pass it as dname to be plotted.

<code>pred_line</code>	list some ipred line geom properties aesthetics
<code>ipred_line</code>	list some pred line geom properties aesthetics
<code>point</code>	list some point geom properties aesthetics
<code>is.legend</code>	logical if TRUE add a legend
<code>use.finegrid</code>	logical if FALSE use predictions data set
<code>bloq</code>	pmxBLOQ object created by <a href="#">pmx_bloq</a> .
<b>pmx_update parameters</b>	
<code>filter</code>	expression filter which will be applied to plotting data.
<code>strat.facet</code>	formula optional stratification parameter by facetting. This split plot by strats(each strat in a facet)
<code>facets</code>	list <code>facet_wrap</code> parameters.
<code>strat.color</code>	character optional stratification parameter by grouping. This will split the plot by group (color) of strat.
<code>trans</code>	character define the transformation to apply on x or y or both variables
<code>pmxgpar</code>	a object of class <code>pmx_gpar</code> possibly the output of the <b>pmx_gpar: Shared basic graphics parameters</b>
<code>labels</code>	list list containing plot and/or axis labels: title, subtitle, x , y
<code>axis.title</code>	list containing <code>element_text</code> attributes to customize the axis title. (similar to <code>ggplot2 axis.title</code> theme)
<code>axis.text</code>	list containing <code>element_text</code> attributes to customize the axis text (similar to <code>ggplot2 axis.text</code> theme)
<code>ranges</code>	list limits of x/y ranges
<code>is.smooth</code>	logical if set to TRUE add smooth layer
<code>smooth</code>	list <code>geom_smooth</code> graphical/smoothing fun parameters
<code>is.band</code>	logical if TRUE add horizontal band
<code>band</code>	list horizontal band parameters. <code>geom_hline</code> graphical parameters.
<code>is.draft</code>	logical if TRUE add draft layer
<code>draft</code>	list draft layer parameters. <code>geom_text</code> graphical parameters.
<code>is.identity_line</code>	logical if TRUE add an identity line
<code>identity_line</code>	list <code>geom_abline</code> graphical parameters.
<code>scale_x_log10</code>	logical if TRUE use log10 scale for x axis.
<code>scale_y_log10</code>	logical if TRUE use log10 scale for y axis.
<code>color.scales</code>	list define scales parameter in case of <code>strat.color</code> <a href="#">pmx_settings</a>
<code>...</code>	others graphics parameters passed : <ul style="list-style-type: none"> <li>• <a href="#">pmx_gpar</a> internal function to customize shared graphical parameters</li> <li>• <a href="#">individual</a> generic object for individual plots.</li> <li>• <a href="#">pmx_update</a> function.</li> </ul>
<b>individual parameters</b>	

**Value**

ggplot2 or list of ggplot2 objects

**Examples**

```
# basic use -----

ctr <- theophylline()
ctr %>% pmx_plot_individual(npage = 1)
## multiple pages
ctr %>% pmx_plot_individual(npage = c(1, 3))
## change faceting
ctr %>% pmx_plot_individual(facets = list(nrow = 5, ncol = 5), npage = 2)

# update graphical parameter -----

## update labels
ctr %>% pmx_plot_individual(
  labels = list(title = "Custom individual plot")
)

## remove draft
ctr %>% pmx_plot_individual(is.draft = FALSE)

## Customize ipred_line with any geom_line parameter
ctr %>% pmx_plot_individual(
  pred_line = list(color = "red", linetype = 20, alpha = 0.5)
)

## Customize ipred_line with any geom_line parameter
ctr %>% pmx_plot_individual(
  ipred_line = list(size = 5)
)

## Customize any geom_point parameter
ctr %>% pmx_plot_individual(
  point = list(aes(alpha = DV), color = "green", shape = 4)
)

## legend

p <- ctr %>% pmx_plot_individual(
  is.legend=TRUE,
  point=list(shape=20),
  pred_line=list(linetype=6),
  pred_line=list(linetype=3)
)

# # stratification -----
```

```

#
# ## continuous stratification
ctr %>% pmx_plot_individual(strat.color = "WT0")

# # subsetting -----
#
# ## we can use any expression involving the data
# ## filter and stratify
ctr %>% pmx_plot_individual(
  filter = SEX == 1, strat.facet = ~SEX,
  facets = list(nrow = 5, ncol = 5))

# # transformation -----
#
# ## apply a log transformation in y
ctr %>% pmx_plot_individual(trans = "log10_y")
# ## apply a custom transformation to normalize axis between 0 and 1

## get a list of parameter
p <- ctr %>% pmx_plot_individual(
  npage=NULL,
  point=list(shape=4,color='blue',size=10),
  facets = list(nrow = 5, ncol = 5),
  labels = list(title = "My individuals",x='my time',y='PD data')
)

```

---

pmx\_plot\_iwres\_dens     *IWRES density plot*

---

## Description

IWRES density plot

## Usage

```

pmx_plot_iwres_dens(ctr, dname, xlim, var_line, snd_line, vline, filter,
  strat.facet, facets, strat.color, trans, pmxgpar, labels, axis.title,
  axis.text, ranges, is.smooth, smooth, is.band, band, is.draft, draft,
  is.identity_line, identity_line, scale_x_log10, scale_y_log10, color.scales,
  ...)

```

## Arguments

ctr	pmx controller
dname	character name of dataset to be used. User can create his own dataset using <a href="#">set_data</a> and pass it as dname to be plotted.
xlim	numeric x axis limits

var_line	list variable density graphics parameters
snd_line	list normal density graphics parameters
vline	list vertical line graphics parameters
<b>pmx_update parameters</b>	
filter	expression filter which will be applied to plotting data.
strat.facet	formula optional stratification parameter by facetting. This split plot by strats(each strat in a facet)
facets	list facet_wrap parameters.
strat.color	character optional stratification parameter by grouping. This will split the plot by group (color) of strat.
trans	character define the transformation to apply on x or y or both variables
pmxgpar	a object of class pmx_gpar possibly the output of the
<b>pmx_gpar: Shared basic graphics parameters</b>	
labels	list list containing plot and/or axis labels: title, subtitle, x , y
axis.title	list containing element_text attributes to customize the axis title. (similar to ggplot2 axis.title theme)
axis.text	list containing element_text attributes to customize the axis text (similar to ggplot2 axis.text theme)
ranges	list limits of x/y ranges
is.smooth	logical if set to TRUE add smooth layer
smooth	list geom_smooth graphical/smoothing fun parameters
is.band	logical if TRUE add horizontal band
band	list horizontal band parameters. geom_hline graphical parameters.
is.draft	logical if TRUE add draft layer
draft	list draft layer parameters. geom_text graphical parameters.
is.identity_line	logical if TRUE add an identity line
identity_line	listgeom_abline graphical parameters.
scale_x_log10	logical if TRUE use log10 scale for x axis.
scale_y_log10	logical if TRUE use log10 scale for y axis.
color.scales	list define scales parameter in case of strat.color <a href="#">pmx_settings</a>
...	others graphics parameters passed : <ul style="list-style-type: none"> <li>• <a href="#">pmx_gpar</a> internal function to customize shared graphical parameters</li> <li>• <a href="#">pmx_dens</a> pmx density object.</li> <li>• <a href="#">pmx_update</a> function.</li> </ul>
<b>pmx_dens parameters</b>	

**Value**

ggplot2 or list of ggplot2 objects

pmx\_plot\_vpc

*VPC plot***Description**

VPC plot

**Usage**

```
pmx_plot_vpc(ctr, type, idv, obs, pi, ci, rug, bin, is.legend, dname, filter,
  strat.facet, facets, strat.color, trans, pmxgpar, labels, axis.title,
  axis.text, ranges, is.smooth, smooth, is.band, band, is.draft, draft,
  is.identity_line, identity_line, scale_x_log10, scale_y_log10, color.scales,
  ...)
```

**Arguments**

ctr	pmx controller
type	character can be either percentile or scatter
idv	character individual variable
obs	pmx_vpc_obs object observation layer <a href="#">pmx_vpc_obs</a>
pi	pmx_vpc_pi object percentile layer <a href="#">pmx_vpc_pi</a>
ci	pmx_vpc_ci object confidence interval layer <a href="#">pmx_vpc_ci</a>
rug	pmx_vpc_rug object rug layer <a href="#">pmx_vpc_rug</a>
bin	pmx_vpc_bin object <a href="#">pmx_vpc_bin</a>
is.legend	logical if TRUE add legend
dname	added for compatibility with other ggPMX plots
	<b>pmx_update parameters</b>
filter	expression filter which will be applied to plotting data.
strat.facet	formula optional stratification parameter by faceting. This split plot by strats(each strat in a facet)
facets	list facet_wrap parameters.
strat.color	character optional stratification parameter by grouping. This will split the plot by group (color) of strat.
trans	character define the transformation to apply on x or y or both variables
pmxgpar	a object of class pmx_gpar possibly the output of the
	<b>pmx_gpar: Shared basic graphics parameters</b>
labels	list list containing plot and/or axis labels: title, subtitle, x , y
axis.title	list containing element_text attributes to customize the axis title. (similar to ggplot2 axis.title theme)



axis.text	list containing element_text attributes to customize the axis text (similar to ggplot2 axis.text theme)
ranges	list limits of x/y ranges
is.smooth	logical if set to TRUE add smooth layer
smooth	list geom_smooth graphical/smoothing fun parameters
is.band	logical if TRUE add horizontal band
band	list horizontal band parameters. geom_hline graphical parameters.
is.draft	logical if TRUE add draft layer
draft	list draft layer parameters. geom_text graphical parameters.
is.identity_line	logical if TRUE add an identity line
identity_line	listgeom_abline graphical parameters.
scale_x_log10	logical if TRUE use log10 scale for x axis.
scale_y_log10	logical if TRUE use log10 scale for y axis.
color.scales	list define scales parameter in case of strat.color <a href="#">pmx_settings</a>
...	others graphics parameters passed : <ul style="list-style-type: none"> <li>• <a href="#">pmx_gpar</a> internal function to customize shared graphical parameters</li> <li>• <a href="#">pmx_vpc</a> pmx vpc object.</li> <li>• <a href="#">pmx_update</a> function.</li> </ul>

### **pmx\_vpc parameters**

### **Details**

You can use [pmx\\_vpc\\_bin](#) to set the bin parameters. In case of stratification, binning can be different for each strat level (case within\_strat equal to FALSE).

### **Value**

ggplot2 or list of ggplot2 objects

### **See Also**

Other vpc: [pmx\\_vpc\\_bin](#), [pmx\\_vpc\\_ci](#), [pmx\\_vpc\\_obs](#), [pmx\\_vpc\\_pi](#), [pmx\\_vpc\\_rug](#), [pmx\\_vpc](#)

### **Examples**

```
library(ggPMX)

theo_path <- file.path(
  system.file(package = "ggPMX"), "testdata",
  "theophylline"
)
WORK_DIR <- file.path(theo_path, "Monolix")
input_file <- file.path(theo_path, "data_pk.csv")
```

```

vpc_file <- file.path(theo_path, "sim.csv")

ctr <- pmx_mlx(
  config = "standing",
  directory = WORK_DIR,
  input = input_file,
  dv = "Y",
  dvid = "dvid",
  cats = c("SEX"),
  conts = c("WT0", "AGE0"),
  strats = "STUD",
  settings = pmx_settings(
    use.labels=TRUE,
    cats.labels=list(
      SEX=c("0"="Male", "1"="Female")
    )
  ),
  sim = pmx_sim(
    file = vpc_file,
    irun = "rep",
    idv = "TIME"
  )
)

ctr %>% pmx_plot_vpc(
  strat.facet="SEX",
  facets=list(nrow=2),
  type="percentile",
  is.draft = FALSE,
  pi = pmx_vpc_pi(interval = c(0.1,0.9),
    median=list(color="green"),
    extreme= list(color="green")),
  obs = pmx_vpc_obs(color="blue", shape=18, size=2),
  ci = pmx_vpc_ci(interval = c(0.1,0.9),
    median=list(fill="pink")),
  bin=pmx_vpc_bin("kmeans", n=5)
)

ctr %>%
  pmx_plot_vpc(bin= pmx_vpc_bin(
    style = "fixed",
    fixedBreaks=c(-10,2, 5, 10,15,50))
  )

```

**Description**

This function creates a qq plot object

**Usage**

```
pmx_qq(x, labels, dname = NULL, point = NULL, xmax = TRUE,
       facets = NULL, is.reference_line = NULL, reference_line = NULL,
       is.shrink = NULL, shrink = NULL, is.hline = NULL, hline = NULL, ...)
```

**Arguments**

x	character variable name to sample
labels	list of texts/titles used within the plot
dname	name of dataset to be used
point	list geom_point attributes color, shape,...
xmax	logical if FALSE do not use max(aes(x)) as limits default to TRUE
facets	list
is.reference_line	logical if TRUE add reference line to the plot
reference_line	list geom_line attributes. Used only for pmx_plot_eta_qq
is.shrink	logical if TRUE add shrinkage to the plot
shrink	list shrinkage graphical parameter
is.hline	logical if TRUE add horizontal line y=0 ( TRUE by default)
hline	geom hline graphical parameters
...	others graphics arguments passed to <a href="#">pmx_gpar</a> internal object.

**Details**

**labels** is a list that contains:

- **title:** plot title default "EBE vs. covariates"
- **x:** x axis label default to "Etas"
- **y:** y axis label default to empty

**point** is a list that contains:

- **shape:** default to 1
- **color:** default to black
- **size:** default to 1

**Value**

pmx\_qq object

pmx\_qq\_plot

*Quantile-quantile plots***Description**

Quantile-quantile plots  
 Quantile-quantile plot of IWRES  
 Quantile-quantile plot of eta variables  
 Quantile-quantile plot of NPDE

**Usage**

```
pmx_qq_plot(dname, point, is.reference_line, reference_line, is.shrink, shrink,
  is.hline, hline, filter, strat.facet, facets, strat.color, trans, pmxgpar,
  labels, axis.title, axis.text, ranges, is.smooth, smooth, is.band, band,
  is.draft, draft, is.identity_line, identity_line, scale_x_log10,
  scale_y_log10, color.scales, ...)
```

```
pmx_plot_iwres_qq(ctr, ...)
```

```
pmx_plot_eta_qq(ctr, ...)
```

```
pmx_plot_npde_qq(ctr, ...)
```

**Arguments**

dname	name of dataset to be used
point	list geom_point parameters.
is.reference_line	logical if TRUE add reference line to the plot
reference_line	list geom_abline parameters.
is.shrink	logical if TRUE add shrinkage to the plot
shrink	list shrinkage graphical parameter (geom_text)
is.hline	logical if TRUE add horizontal line y=0 ( TRUE by default)
hline	list geom hline graphical parameters
	<b>pmx_update parameters</b>
filter	expression filter which will be applied to plotting data.
strat.facet	formula optional stratification parameter by facetting. This split plot by strats(each strat in a facet)
facets	list facet_wrap parameters.
strat.color	character optional stratification parameter by grouping. This will split the plot by group (color) of strat.

trans	character define the transformation to apply on x or y or both variables
pmxgpar	an object of class pmx_gpar
labels	list list containing plot and/or axis labels: title, subtitle, x , y
axis.title	list containing element_text attributes to customize the axis title. (similar to ggplot2 axis.title theme)
axis.text	list containing element_text attributes to customize the axis text (similar to ggplot2 axis.text theme)
ranges	list limits of x/y ranges
is.smooth	logical if set to TRUE add smooth layer
smooth	list geom_smooth graphical/smoothing fun parameters
is.band	logical if TRUE add horizontal band
band	list horizontal band parameters. geom_hline graphical parameters.
is.draft	logical if TRUE add draft layer
draft	list draft layer parameters. geom_text graphical parameters.
is.identity_line	logical if TRUE add an identity line
identity_line	listgeom_abline graphical parameters.
scale_x_log10	logical if TRUE use log10 scale for x axis.
scale_y_log10	logical if TRUE use log10 scale for y axis.
color.scales	list define scales parameter in case of strat.color <a href="#">pmx_settings</a>
...	others graphics parameters passed : <ul style="list-style-type: none"> <li>• <a href="#">pmx_gpar</a> internal function to customize shared graphical parameters</li> <li>• <a href="#">pmx_qq</a> quantile-quantile plot object.</li> <li>• <a href="#">pmx_update</a> function.</li> </ul>
	<b>pmx_qq parameters</b>
ctr	pmx controller

**Value**

ggplot2 object

**Examples**

```
# ***** basic use ***** -----

ctr <- theophylline()
ctr %>% pmx_plot_eta_qq
ctr %>% pmx_plot_npde_qq
ctr %>% pmx_plot_iwres_qq

# update graphical parameter -----
```

```

## add reference line
ctr %>% pmx_plot_npde_qq(reference_line=list(color="blue"))

## remove reference line
ctr %>% pmx_plot_eta_qq(reference_line=NULL)

# stratification -----

## categorical stratification color parameter
ctr %>% pmx_plot_iwres_qq(strat.facet="STUD", strat.color="SEX")
## categorical stratification facetting
ctr %>% pmx_plot_eta_qq(strat.facet = "SEX")

## do not use symmetric axis
ctr %>% pmx_plot_npde_qq(xmax=FALSE, reference_line=list())

```

---

pmx\_register\_plot      *Register plot*

---

### Description

Register plot

### Usage

```
pmx_register_plot(ctr, pp, pname = NULL)
```

### Arguments

ctr	pmxClass controller
pp	ggplot2 plot
pname	character plot nme

---

pmx\_report      *Generates ggpmX report from a pre-defined template*

---

### Description

Generates ggpmX report from a pre-defined template

**Usage**

```
pmx_report(contr, name, save_dir, format = c("both", "plots", "report"),
  template = "standing", footnote = format == "both", edit = FALSE, title,
  ...)
```

**Arguments**

contr	pmxClass controller
name	character The report name
save_dir	Output directory. A directory to write the results files to
format	character the result type, can be a standalone directory of plots or a report document as defined in the template (pdf, docx,..) ,or both
template	character ggPMX predefined template or the path to a custom rmarkdwon template. Use <a href="#">pmx_report_template</a> to get the list of available templates
footnote	logical TRUE to add a footnote to the generated plots. The default footnote is to add the path where the plot is saved.
edit	logical TRUE to edit the template immediately
title	character report title (optional)
...	extra parameters depending in the template used

**Details**

`pmx_report` uses pre-defined template `.Rmd` to generate the report. The idea is to pass the controller as a report argument using `knitr params artifact`.

**Examples**

```
library(ggPMX)
## list of templates
## ctr %>% pmx_report_template()

report_dir <- tempdir()
## case1: generate a single report
## We use default save dir,
ctr <- theophylline()
ctr %>% pmx_report(
  name = "my_report",
  save_dir = report_dir,
  format="report")

## case2: generate standalone plots
## Note here the use of a custom dir to save results
ctr <- theophylline()
```

```
ctr %>% pmx_report(
  name = "my_report",
  save_dir = report_dir,
  format="plots")

## case3: generate both : reports + plots
## by default add footnote
## Note , you can force footnote to FALSE using footnote parameter
ctr <- theophylline()
ctr %>% pmx_report(
  name = "my_report",
  save_dir = report_dir,
  format="both")

## case4 : generate standalone plots with footnotes
ctr <- theophylline()
ctr %>% pmx_report(
  name = "my_report",
  save_dir = report_dir,
  footnote=TRUE,
  format="plots")

## case6: dynamic edit
## uncomment to run
# ctr <- theophylline()
# ctr %>% pmx_report(
#   save_dir = file.path(report_dir,"case6"),
#   name = "my_report",
#   format="report",
#   edit = TRUE)

## case7 : use custom template file

ctr <- theophylline()
custom_template <-
  file.path( system.file(package = "ggPMX"),"examples","templates","custom_report.Rmd")
ctr %>% pmx_report(
  name="report2",
  save_dir = report_dir,
  template=custom_template,
  format="both"
)

## case7 : generate individual plots report

ctr <- theophylline()
ctr %>% pmx_report(
```



```
    name="report2",
    save_dir = report_dir,
    template="individual",
    format="both",
    npage=1:2
  )

## case8: misc example with complicated features
## see github issue : #179
ctr <- theophylline()
misc_template <-
  file.path( system.file(package = "ggPMX"), "examples", "templates", "misc.Rmd")
ctr %>% pmx_report(
  name="misc",
  save_dir = report_dir,
  template=misc_template,
  format="both"
)
```

---

pmx\_report\_template    *Gets build-in report templates*

---

## Description

Gets build-in report templates

## Usage

```
pmx_report_template()
```

## Value

list of templates names

## Examples

```
pmx_report_template()
```

---

pmx\_settings                      *Create controller global settings*

---

### Description

Create controller global settings

### Usage

```
pmx_settings(is.draft = TRUE, use.abbrev = FALSE, color.scales = NULL,
  cats.labels = NULL, use.labels = FALSE, use.titles = TRUE,
  effects = NULL, ...)
```

### Arguments

is.draft	logical if FALSE any plot is without draft annotation
use.abbrev	logical if TRUE use abbreviations mapping for axis names
color.scales	list list containing elements of scale_color_manual
cats.labels	list list of named vectors for each factor
use.labels	logical if TRUE replace factor named by cats.labels
use.titles	logical FALSE to generate plots without titles
effects	list list of effects levels and labels
...	extra parameter not used yet

### Value

pmxSettingsClass object

### Examples

```
library(ggPMX)
library(ggplot2)
ctr <- theophylline(
  settings=
    pmx_settings(
      color.scales=list(
        "Study",
        labels=c("Study 1", "Study 2"),
        values=c("1"="lightyellow", "2"="lightblue")),
      cats.labels=list(
        SEX=c("0"="M", "1"="F"),
        STUD=c("1"="Study 1", "2"="Study 2")
      ),
      use.abbrev=TRUE,
      is.draft=TRUE,
```

```

        use.labels=TRUE
      )
    )

    ctr %>%
      pmx_plot_npde_time(strat.color="STUD", strat.facet=~SEX)
    #
    #
    ctr %>%
      pmx_plot_eta_box(strat.color="STUD", strat.facet =~SEX)

    ctr %>% pmx_plot_eta_hist

```

---

pmx\_sim

*Create simulation object*


---

## Description

Create simulation object

## Usage

```
pmx_sim(file, irun, idv)
```

## Arguments

file	character path to the simulation file
irun	character name of the simulation column
idv	character name of the ind. variable

## Examples

```

library(ggPMX)

theo_path <- file.path(
  system.file(package = "ggPMX"), "testdata",
  "theophylline"
)
WORK_DIR <- file.path(theo_path, "Monolix")
input_file <- file.path(theo_path, "data_pk.csv")
vpc_file <- file.path(theo_path, "sim.csv")

ctr <- pmx_mlx(
  config = "standing",
  directory = WORK_DIR,

```

```

input = input_file,
dv = "Y",
dvid = "dvid",
cats = c("SEX"),
conts = c("WT0", "AGE0"),
strats = "STUD",
settings = pmx_settings(
  use.labels=TRUE,
  cats.labels=list(
    SEX=c("0"="Male", "1"="Female")
  )
),
sim = pmx_sim(
  file = vpc_file,
  irun = "rep",
  idv = "TIME"
)
)

ctr %>% pmx_plot_vpc(
  strat.facet="SEX",
  facets=list(nrow=2),
  type="percentile",
  is.draft = FALSE,
  pi = pmx_vpc_pi(interval = c(0.1,0.9),
    median=list(color="green"),
    extreme= list(color="green")),
  obs = pmx_vpc_obs(color="blue", shape=18, size=2),
  ci = pmx_vpc_ci(interval = c(0.1,0.9),
    median=list(fill="pink")),
  bin=pmx_vpc_bin("kmeans", n=5)
)

ctr %>%
  pmx_plot_vpc(bin= pmx_vpc_bin(
    style = "fixed",
    fixedBreaks=c(-10,2, 5, 10,15,50))
  )

```

---

pmx\_theme

*Define ggPMX theme*

---

## Description

This theme is a simple wrapper gdoc theme from ggthemes package.

**Usage**

```
pmx_theme(...)
```

**Arguments**

... can contain any valid argument of ggplot2 [theme](#) object.

---

pmx_update	<i>Update plot object</i>
------------	---------------------------

---

**Description**

Update plot object

**Usage**

```
pmx_update(ctr, pname, strat.color = NULL, strat.facet = NULL,
  color.scales = NULL, filter = NULL, trans = NULL, ..., pmxgpar = NULL)
```

**Arguments**

ctr	pmxClass controller object
pname	character the plot name to update
strat.color	character optional stratification parameter
strat.facet	formula optional stratification parameter
color.scales	list can be used with strat.color to set scale_color_manual <a href="#">pmx_gpar</a> function.
filter	optional filter which will be applied to plotting data
trans	character define the transformation to apply on x or y or both variables
...	others graphical parameters given to set the plot
pmxgpar	a object of class pmx_gpar possibly the output of the

**Details**

**trans** is a transformation that user can apply to x, or y coordinates. The transformation is applied to the data before the plotting. This gives more flexibility to the user and also conserves all static positions like annotations ( draft specially)

For example:

var\_x apply variance to x coordinates the variance function

var\_xy apply variance to both This mechanism is applied internally to scale log.

**Value**

controller object with the plot updated

**See Also**

Other pmxclass: [get\\_cats](#), [get\\_conts](#), [get\\_covariates](#), [get\\_data](#), [get\\_occ](#), [get\\_plot\\_config](#), [get\\_plot](#), [get\\_strats](#), [plot\\_names](#), [plots](#), [set\\_data](#), [set\\_plot](#)

---

pmx_vpc	<i>Creates vpc object</i>
---------	---------------------------

---

**Description**

Creates vpc object

**Usage**

```
pmx_vpc(type = c("percentile", "scatter"), idv = "TIME",
  obs = pmx_vpc_obs(), pi = pmx_vpc_pi(), ci = pmx_vpc_ci(),
  rug = pmx_vpc_rug(), bin = pmx_vpc_bin(), labels = NULL,
  facets = NULL, is.legend = FALSE, dname = NULL, ...)
```

**Arguments**

type	character can be either percentile or scatter
idv	character individual variable
obs	pmx_vpc_obs object observation layer <a href="#">pmx_vpc_obs</a>
pi	pmx_vpc_pi object percentile layer <a href="#">pmx_vpc_pi</a>
ci	pmx_vpc_ci object confidence interval layer <a href="#">pmx_vpc_ci</a>
rug	pmx_vpc_rug object rug layer <a href="#">pmx_vpc_rug</a>
bin	pmx_vpc_bin object <a href="#">pmx_vpc_bin</a>
labels	list define title and axis labels
facets	is a list of parameters passed to <code>facet_wrap</code> in case of stratification
is.legend	logical if TRUE add legend
dname	added for compatibility with other ggPMX plots
...	extra parameters passed to base graphical parameters

**See Also**

Other vpc: [pmx\\_plot\\_vpc](#), [pmx\\_vpc\\_bin](#), [pmx\\_vpc\\_ci](#), [pmx\\_vpc\\_obs](#), [pmx\\_vpc\\_pi](#), [pmx\\_vpc\\_rug](#)

---

pmx\_vpc\_bin                      *Creates vpc bins*

---

### Description

Creates vpc bins

### Usage

```
pmx_vpc_bin(style, within_strat = FALSE, ...)
```

### Arguments

style	character style chosen on of the: "fixed", "sd", "equal", "pretty", "quantile", "kmeans", "hclust" or "jenks"
within_strat	logical if TRUE compute the bining for each strat level. By default t is false and bining are equal for all stratifications levels.
...	other classInt::classIntervals parameters excpet style and n

### Details

This is a warraper to

### See Also

Other vpc: [pmx\\_plot\\_vpc](#), [pmx\\_vpc\\_ci](#), [pmx\\_vpc\\_obs](#), [pmx\\_vpc\\_pi](#), [pmx\\_vpc\\_rug](#), [pmx\\_vpc](#)

---

pmx\_vpc\_ci                      *Sets vpc confidence interval layer*

---

### Description

Sets vpc confidence interval layer

### Usage

```
pmx_vpc_ci(show = c("all", "median"), interval = c(0.05, 0.95),
  method = c("ribbon", "rectangle"), median = list(fill = "#3388cc", alpha =
  0.3), extreme = list(fill = "#3388cc", alpha = 0.3))
```

**Arguments**

show	character how areas are displayed: <ul style="list-style-type: none"> <li>• <b>show="all"</b> areas will be displayed for each of the 3 percentiles.</li> <li>• <b>show="median"</b> Show only median area.</li> </ul>
interval	numeric quantiles values default to <code>c(.05, .95)</code>
method	character which areas are displayed: <ul style="list-style-type: none"> <li>• <b>method="ribbon"</b> areas are ribbons.</li> <li>• <b>method="rectangle"</b> areas are horizontal rectangles.</li> </ul>
median	list containing: <ul style="list-style-type: none"> <li>• <b>fill</b> character Color of the area representing the CI for the median. Default: "#3388cc".</li> <li>• <b>alpha</b> numeric Transparency of the area representing the PI for the median. Default=0.3.</li> </ul>
extreme	list containing: <ul style="list-style-type: none"> <li>• <b>fill</b> character Color of the area representing the CI for the extreme percentiles. Default: "#3388cc".</li> <li>• <b>alpha</b> numeric Transparency of the area representing the PI for the extreme percentiles. Default=0.3.</li> </ul>

**See Also**

Other vpc: [pmx\\_plot\\_vpc](#), [pmx\\_vpc\\_bin](#), [pmx\\_vpc\\_obs](#), [pmx\\_vpc\\_pi](#), [pmx\\_vpc\\_rug](#), [pmx\\_vpc](#)

---

pmx_vpc_obs	<i>Sets vpc observation layer</i>
-------------	-----------------------------------

---

**Description**

Sets vpc observation layer

**Usage**

```
pmx_vpc_obs(show = TRUE, color = "#000000", size = 1, alpha = 0.7,
            shape = 1)
```

**Arguments**

show	logical if TRUE show observation points
color	character Color of the observed endpoint values. Default: "#000000".
size	numeric Size of the observed endpoint values. Default: 1.
alpha	numeric Transparency of the observed endpoint values. Default: 0.7.
shape	numeric Shape of the observed endpoint values. Default: 1.



**See Also**

Other vpc: [pmx\\_plot\\_vpc](#), [pmx\\_vpc\\_bin](#), [pmx\\_vpc\\_ci](#), [pmx\\_vpc\\_pi](#), [pmx\\_vpc\\_rug](#), [pmx\\_vpc](#)

---

pmx_vpc_pi	<i>Sets vpc percentile layer</i>
------------	----------------------------------

---

**Description**

Sets vpc percentile layer

**Usage**

```
pmx_vpc_pi(show = c("all", "median"), interval = c(0.05, 0.95),
  median = list(color = "#000000", size = 1, alpha = 0.7, linetype = "solid"),
  extreme = list(color = "#000000", size = 1, alpha = 0.7, linetype =
    "solid"))
```

**Arguments**

show	charcater how lines are displayed: <ul style="list-style-type: none"> <li>• <b>show=all</b> lines will be displayed for each of the 3 percentiles.</li> <li>• <b>show=median</b> Show only median line.</li> </ul>
interval	numeric quantiles values default to c(.05, .95)
median	list containing: <ul style="list-style-type: none"> <li>• <b>color</b> charcater Color of the median percentile line. Default: "#000000".</li> <li>• <b>size</b> numeric Thickness of the median percentile line. Default: 1.</li> <li>• <b>alpha</b> numeric Transparency of the median percentile line. Default: 0.7.</li> <li>• <b>linetype</b> charcater Linetype of the median percentile line. Default: "solid".</li> </ul>
extreme	list containing: <ul style="list-style-type: none"> <li>• <b>color</b> charcater Color of the median percentile line. Default: "#000000".</li> <li>• <b>size</b> numeric Thickness of the median percentile line. Default: 1.</li> <li>• <b>alpha</b> numeric Transparency of the median percentile line. Default: 0.7.</li> <li>• <b>linetype</b> charcater Linetype of the median percentile line. Default: "solid"</li> </ul>

**See Also**

Other vpc: [pmx\\_plot\\_vpc](#), [pmx\\_vpc\\_bin](#), [pmx\\_vpc\\_ci](#), [pmx\\_vpc\\_obs](#), [pmx\\_vpc\\_rug](#), [pmx\\_vpc](#)

---

pmx\_vpc\_rug                      *Sets vpc rug layer*

---

### Description

Sets vpc rug layer

### Usage

```
pmx_vpc_rug(show = TRUE, color = "#000000", size = 1, alpha = 0.7)
```

### Arguments

show	logical	If TRUE show bin separators
color	character	Color of the rug. Default: "#000000".
size	numeric	Thickness of the rug. Default: 1.
alpha	numeric	Transparency of the rug. Default: 0.7.

### Details

When the vpc confidence interval layer method is rectangles we don't show rug separators.

### See Also

Other vpc: [pmx\\_plot\\_vpc](#), [pmx\\_vpc\\_bin](#), [pmx\\_vpc\\_ci](#), [pmx\\_vpc\\_obs](#), [pmx\\_vpc\\_pi](#), [pmx\\_vpc](#)

---

print.abbreviation              *S3 print abbreviation*

---

### Description

S3 print abbreviation

### Usage

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

### Arguments

x	object of class configs
...	pass additional options (not used presently)

### Value

print abbreviation

---

print.configs	<i>This function can be used to print configuration of the defined object using S3 method.</i>
---------------	--

---

**Description**

This function can be used to print configuration of the defined object using S3 method.

**Usage**

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

**Arguments**

x	object of class configs
...	pass additional options (not used presently)

**Value**

print result

---

print.pmxClass	<i>Print pmxClass object</i>
----------------	------------------------------

---

**Description**

Print pmxClass object

**Usage**

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

**Arguments**

x	pmxClass object
...	additinal arguments to pass to print

**Value**

print object to screen

print.pmxConfig      *S3 method print pmxConfig object*

---

**Description**

S3 method print pmxConfig object

**Usage**

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

**Arguments**

x                    pmxConfig object  
...                  additional arguments to pass to print (unused currently)

**Value**

invisible object

---

print.pmx\_gpar      *Print pmx\_gpar object*

---

**Description**

Print pmx\_gpar object

**Usage**

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

**Arguments**

x                    pmx\_gpar object  
...                  argument passed to print ( to satisfy generic)

**Value**

a character description of graphical parameters

---

read_input	<i>Read Modelling input data</i>
------------	----------------------------------

---

**Description**

Read Modelling input data

**Usage**

```
read_input(ipath, dv, dvid, cats = "", conts = "", strats = "",
           occ = "", endpoint = NULL, id = NULL, time = NULL)
```

**Arguments**

ipath	full path of the input file
dv	character the name of measurable variable used in the input modelling file
dvid	character observation type parameter
cats	<i>[Optional]</i> character vector of categorical covariates
conts	<i>[Optional]</i> character vector of continuous covariates
strats	<i>[Optional]</i> character extra stratification variables
occ	<i>[Optional]</i> character inter individual occasion variables
endpoint	integer null in case of a single endpoint otherwise the index of endpoints.
id	character the name of identifier variable used in the input modelling file.
time	character the name of time variable used in the input modelling file

**Value**

data.table well formatted containing modelling input data

---

read_mlx_ind_est	<i>Read MONOLIX individual parameters</i>
------------------	---

---

**Description**

Read MONOLIX individual parameters

**Usage**

```
read_mlx_ind_est(path, x, ...)
```

**Arguments**

path	character path to the file
x	dataset object
...	extra paramter not used

**Value**

data.table object

---

read_mlx_par_est	<i>Read MONOLIX parameter estimation file</i>
------------------	---

---

**Description**

Read MONOLIX parameter estimation file

**Usage**

```
read_mlx_par_est(path, x, ...)
```

**Arguments**

path	character path to the file
x	dataset object
...	extra parameter not used

**Value**

data.table object

---

read_mlx_pred	<i>Read MONOLIX model predictions</i>
---------------	---------------------------------------

---

**Description**

Read MONOLIX model predictions

**Usage**

```
read_mlx_pred(path, x, ...)
```

**Arguments**

path	character path to the file
x	dataset object
...	extra paramter not used

**Value**

data.table object

---

residual	<i>This function create a residual for each observed value and also generates a residual distribution</i>
----------	---

---

**Description**

This function create a residual for each observed value and also generates a residual distribution

**Usage**

```
residual(x, y, labels = NULL, point = NULL, is.hline = FALSE,
         hline = NULL, dname = NULL, facets = NULL, bloq = NULL, ...)
```

**Arguments**

x	x axis aesthetics
y	y axis aesthetics
labels	list that contain title,subtitle, axis labels
point	geom point graphical parameters
is.hline	logical if TRUE add horizontal line y=0 ( TRUE by default)
hline	geom hline graphical parameters
dname	name of dataset to be used
facets	list wrap facetting in case of strat.facet
bloq	pmxBLOQ object created by <a href="#">pmx_bloq</a>
...	others graphics arguments passed to <a href="#">pmx_gpar</a> internal object.

**Details**

Some parameters are a list of parameters :

**point** is a list that contains:

- **shape:** default to 1
- **color:** default to black
- **size:** default to 1

**labels** is a list that contains:

- **title:** plot title default to AES\_X versus AES\_Y
- **subtitle:** plot subtitle default empty
- **x:** x axis label default to AES\_X
- **y:** y axis label default to AES\_Y

### Value

a residual object

### See Also

[plot\\_pmx.residual](#)

---

residual_scatter	<i>Scatter residual plots</i>
------------------	-------------------------------

---

### Description

Scatter residual plots  
DV vs PRED plot  
DV vs IPRED plot  
IWRES vs IPRED plot  
|IWRES| vs IPRED plot  
IWRES vs TIME plot  
NPDE vs TIME plot  
NPDE vs PRED plot

### Usage

```
residual_scatter(point, is.hline, hline, dname, bloq, filter, strat.facet,  
  facets, strat.color, trans, pmxgpar, labels, axis.title, axis.text, ranges,  
  is.smooth, smooth, is.band, band, is.draft, draft, is.identity_line,  
  identity_line, scale_x_log10, scale_y_log10, color.scales, ...)
```

```
pmx_plot_dv_pred(ctr, ...)
```

```
pmx_plot_dv_ipred(ctr, ...)
```

```
pmx_plot_iwres_ipred(ctr, ...)
```

```
pmx_plot_abs_iwres_ipred(ctr, ...)
```

```
pmx_plot_iwres_time(ctr, ...)
```



```
pmx_plot_npde_time(ctr, ...)
```

```
pmx_plot_npde_pred(ctr, ...)
```

### Arguments

point	list geom_point graphical parameters.
is.hline	logical if TRUE add horizontal line y=0 ( TRUE by default).
hline	list geom_hline graphical parameters.
dname	character name of dataset to be used. User can create his own dataset using <a href="#">set_data</a> and pass it as dname to be plotted.
bloq	pmxBLOQ object created by <a href="#">pmx_bloq</a> .
	<b>pmx_update parameters</b>
filter	expression filter which will be applied to plotting data.
strat.facet	formula optional stratification parameter by facetting. This split plot by strats(each strat in a facet)
facets	list facet_wrap parameters.
strat.color	character optional stratification parameter by grouping. This will split the plot by group (color) of strat.
trans	character define the transformation to apply on x or y or both variables
pmxgpar	a object of class pmx_gpar possibly the output of the <b>pmx_gpar: Shared basic graphics parameters</b>
labels	list list containing plot and/or axis labels: title, subtitle, x , y
axis.title	list containing element_text attributes to customize the axis title. (similar to ggplot2 axis.title theme)
axis.text	list containing element_text attributes to customize the axis text (similar to ggplot2 axis.text theme)
ranges	list limits of x/y ranges
is.smooth	logical if set to TRUE add smooth layer
smooth	list geom_smooth graphical/smoothing fun parameters
is.band	logical if TRUE add horizontal band
band	list horizontal band parameters. geom_hline graphical parameters.
is.draft	logical if TRUE add draft layer
draft	list draft layer parameters. geom_text graphical parameters.
is.identity_line	logical if TRUE add an identity line
identity_line	listgeom_abline graphical parameters.
scale_x_log10	logical if TRUE use log10 scale for x axis.
scale_y_log10	logical if TRUE use log10 scale for y axis.
color.scales	list define scales parameter in case of strat.color <a href="#">pmx_settings</a>

```

...           others graphics parameters passed :
              • pmx_gpar internal function to customize shared graphical parameters
              • residual generic object for all residual (scatter) plots .
              • pmx_update function.

residual parameters

ctr          pmx controller

```

**Value**

ggplot2 object

**Examples**

```

# NOTES #####
# examples are availables for all residual plots:
# - pmx_plot_abs_iwres_ipred
# - pmx_plot_dv_ipred
# - pmx_plot_dv_pred
# - pmx_plot_iwres_ipred
# - pmx_plot_iwres_time
# - pmx_plot_npde_time

# basic use -----

ctr <- theophylline()
p <- ctr %>% pmx_plot_dv_pred()
## p is a ggplot2 object you can add any layer here
p + ggplot2::theme_minimal()

# update graphical parameter -----

## update labels
ctr %>% pmx_plot_dv_pred(
  labels = list(title = "DV versus PRED new title")
)

## remove draft
ctr %>% pmx_plot_dv_pred(is.draft = FALSE)

## remove horizontal line
ctr %>% pmx_plot_dv_pred(is.hline = FALSE)

## custom point aes and static parameters
## we can customize any geom_point parameter
ctr %>% pmx_plot_dv_pred(
  point = list(aes(alpha = DV), color = "green", shape = 4)
)

```

```

# stratification -----

## continuous stratification
ctr %>% pmx_plot_dv_pred(strat.color = "WT0")
## categorical stratification
ctr %>% pmx_plot_dv_pred(strat.facet = "SEX")
## using formula notation
ctr %>% pmx_plot_dv_pred(strat.facet = STUD~SEX)

# subsetting -----

## we can use any expression involving the data
ctr %>% pmx_plot_dv_pred(filter = DV > mean(DV) & PRED < median(PRED))
## filter and stratify
ctr %>% pmx_plot_dv_pred(filter = SEX == 1, strat.facet = ~SEX)

# transformation -----

## apply a log transformation in y
ctr %>% pmx_plot_dv_pred(trans = "log10_y")

```

---

set_abbrev	<i>update or add a new abbreviation</i>
------------	---

---

## Description

update or add a new abbreviation

## Usage

```
set_abbrev(ctr, ...)
```

## Arguments

ctr	pmxClass controller object
...	Options to set or add, with the form name = value.

## Examples

```

ctr <- theophylline()
ctr %>% set_abbrev("new_param"="new value")
ctr %>% get_abbrev("new_param")

```

---

set_data	<i>Set a controller data set</i>
----------	----------------------------------

---

### Description

Set a controller data set

### Usage

```
set_data(ctr, ...)
```

### Arguments

ctr	the controller object
...	a named list parameters (see example)

### Details

This function can be used to set an existing data set or to create a new one. The basic idea is to change the built-in data set (change the factor level names, change some rows values or apply any other data set operation) and use the new data set using the `dname` parameter of `pmx_plot` family functions.

### See Also

Other `pmxclass`: [get\\_cats](#), [get\\_conts](#), [get\\_covariates](#), [get\\_data](#), [get\\_occ](#), [get\\_plot\\_config](#), [get\\_plot](#), [get\\_strats](#), [plot\\_names](#), [plots](#), [pmx\\_update](#), [set\\_plot](#)

### Examples

```
ctr <- theophylline()
dx <- ctr %>% get_data("eta")
dx <- dx[,EFFECT:=factor(
  EFFECT,levels=c("ka", "V", "Cl"),
  labels=c("Concentration", "Volume", "Clearance"))]
## update existing data set
ctr %>% set_data(eta=dx)
## or create a new data set
ctr %>% set_data(eta_long=dx)
```

---

 set\_plot

 Create a new plot of the desired type
 

---

### Description

Create a new plot of the desired type

### Usage

```
set_plot(ctr, ptype = c("IND", "DIS", "SCATTER", "ETA_PAIRS", "ETA_COV",
  "PMX_QQ", "VPC", "PMX_DENS"), pname, use.defaults = TRUE, filter = NULL,
  strat.color = NULL, strat.facet = NULL, color.scales = NULL,
  trans = NULL, ...)
```

### Arguments

ctr	pmxClass controller object
ptype	plot type can be: <ul style="list-style-type: none"> <li>• "IND" Individual plot type: <a href="#">individual</a></li> <li>• "DIS" Distribution plot type : <a href="#">distrib</a></li> <li>• "SCATTER" Residual plot type : <a href="#">residual</a></li> </ul>
pname	plot name, if missing it will be created using function aesthetics
use.defaults	logical if FALSE do not use defaults defined in yaml init files
filter	optional filter which will be applied to plotting data
strat.color	character
strat.facet	formula define categorical stratification as formula
color.scales	list can be used with strat.color to set <code>scale_color_manual</code>
trans	listtransformation operator
...	other plot parameters to configure <a href="#">pmx_gpar</a> .

### Value

invisible ctr object

### See Also

Other pmxclass: [get\\_cats](#), [get\\_confs](#), [get\\_covariates](#), [get\\_data](#), [get\\_occ](#), [get\\_plot\\_config](#), [get\\_plot](#), [get\\_strats](#), [plot\\_names](#), [plots](#), [pmx\\_update](#), [set\\_data](#)

---

theophylline	<i>Creates pmx controller using theophylline data</i>
--------------	---

---

**Description**

Creates pmx controller using theophylline data

**Usage**

```
theophylline(settings = NULL, ...)
```

**Arguments**

settings	pmxSettings object
...	other parameters of pmx_mlx like endpoint

**Value**

pmx controller

**Examples**

```
## Not run:
theophylline()

## End(Not run)
```

---

wrap_formula	<i>merge facets formula with new formula</i>
--------------	--

---

**Description**

merge facets formula with new formula

**Usage**

```
wrap_formula(x, origin = "lfacet")
```

**Arguments**

x	formula object
origin	the origin formula default to ~lfacets

**Value**

formula object

---

[.pmx\_gpar                    *Method for subsetting "pmx\_gpar" objects*

---

**Description**

Method for subsetting "pmx\_gpar" objects

**Usage**

```
## S3 method for class 'pmx_gpar'  
x[index, ...]
```

**Arguments**

x	pmx_gpar object
index	can be character/integer of element
...	other parameter (not used just for generic)

**Value**

if exists the parameter description

# Index

## \*Topic **datasets**

FacetWrapPaginate, 14  
[.pmx\_gpar, 87

abbrev, 4  
add\_draft, 4

distrib, 5, 7, 11, 13, 22, 29–34, 85  
dummy(eta\_cov\_plot), 7

eta\_cov, 6, 6, 9, 13, 22, 29–34  
eta\_cov\_plot, 7  
eta\_distribution\_plot, 10  
eta\_pairs, 6, 7, 13, 22, 29–34, 50

facet\_wrap, 14  
facet\_wrap\_paginate, 14, 26  
FacetWrapPaginate, 14

get\_abbrev, 16  
get\_cats, 16, 17–21, 28, 70, 84, 85  
get\_conts, 16, 17, 17, 18–21, 28, 70, 84, 85  
get\_covariates, 16, 17, 17, 18–21, 28, 70, 84, 85  
get\_data, 16, 17, 18, 19–21, 28, 70, 84, 85  
get\_occ, 16–18, 18, 19–21, 28, 70, 84, 85  
get\_plot, 16–19, 19, 20, 21, 28, 70, 84, 85  
get\_plot\_config, 16–19, 20, 21, 28, 70, 84, 85  
get\_strats, 16–20, 20, 28, 70, 84, 85  
getPmxOption, 15  
ggPMX, 21  
ggPMX-package (ggPMX), 21  
gtable\_remove\_grobs, 21

individual, 6, 7, 13, 22, 29–34, 52, 85  
input\_finegrid, 23  
is.pmx\_gpar, 23

l\_left\_join, 25  
label\_value(), 15

labeller(), 15  
load\_config, 24  
load\_data\_set, 24  
load\_source, 25

n\_pages, 26

parse\_mlxtran, 26  
pk\_occ, 27  
pk\_pd, 27  
plot\_names, 16–21, 28, 28, 70, 84, 85  
plot\_pmx, 6, 7, 13, 21, 22, 29, 30–34  
plot\_pmx.distrib, 6, 7, 13, 22, 29, 29, 30–34  
plot\_pmx.eta\_cov, 6, 7, 13, 22, 29, 30, 30, 31–34  
plot\_pmx.eta\_pairs, 6, 7, 13, 22, 29, 30, 31, 32–34  
plot\_pmx.individual, 6, 7, 13, 22, 29–31, 31, 32–34  
plot\_pmx.pmx\_dens, 6, 7, 13, 22, 29–32, 32, 33, 34  
plot\_pmx.pmx\_gpar, 6, 7, 13, 22, 29–32, 33, 34  
plot\_pmx.pmx\_qq, 6, 7, 13, 22, 29–33, 33, 34  
plot\_pmx.residual, 6, 7, 13, 22, 29–34, 34, 80  
plot\_shrink, 35  
plots, 16–21, 28, 28, 70, 84, 85  
pmx, 35  
pmx\_bloq, 22, 36, 39, 52, 79, 81  
pmx\_comp\_shrink, 39  
pmx\_config, 40  
pmx\_copy, 41  
pmx\_cov, 7, 42  
pmx\_dens, 43, 55  
pmx\_endpoint, 36, 44  
pmx\_filter, 45  
pmx\_get\_configs, 46  
pmx\_gpar, 5, 7, 9, 11, 13, 22, 29, 43, 46, 48, 50, 52, 55, 57, 59, 61, 69, 79, 82, 85



- pmx\_mlx, [36](#)
- pmx\_mlx (pmx), [35](#)
- pmx\_mlxtran (pmx), [35](#)
- pmx\_plot, [47](#)
- pmx\_plot\_abs\_iwres\_ipred  
(residual\_scatter), [80](#)
- pmx\_plot\_cats, [48](#)
- pmx\_plot\_dv\_ipred (residual\_scatter), [80](#)
- pmx\_plot\_dv\_pred (residual\_scatter), [80](#)
- pmx\_plot\_eta\_box  
(eta\_distribution\_plot), [10](#)
- pmx\_plot\_eta\_cats (eta\_cov\_plot), [7](#)
- pmx\_plot\_eta\_conts (eta\_cov\_plot), [7](#)
- pmx\_plot\_eta\_hist  
(eta\_distribution\_plot), [10](#)
- pmx\_plot\_eta\_matrix, [49](#)
- pmx\_plot\_eta\_qq (pmx\_qq\_plot), [60](#)
- pmx\_plot\_individual, [51](#)
- pmx\_plot\_iwres\_dens, [54](#)
- pmx\_plot\_iwres\_ipred  
(residual\_scatter), [80](#)
- pmx\_plot\_iwres\_qq (pmx\_qq\_plot), [60](#)
- pmx\_plot\_iwres\_time (residual\_scatter),  
[80](#)
- pmx\_plot\_npde\_pred (residual\_scatter),  
[80](#)
- pmx\_plot\_npde\_qq (pmx\_qq\_plot), [60](#)
- pmx\_plot\_npde\_time (residual\_scatter),  
[80](#)
- pmx\_plot\_vpc, [56, 70–74](#)
- pmx\_qq, [48, 58, 61](#)
- pmx\_qq\_plot, [60](#)
- pmx\_register\_plot, [62](#)
- pmx\_report, [62](#)
- pmx\_report\_template, [63, 65](#)
- pmx\_settings, [9, 11, 36, 47, 50, 52, 55, 57,](#)  
[61, 66, 81](#)
- pmx\_sim, [36, 67](#)
- pmx\_theme, [68](#)
- pmx\_update, [9, 11, 16–21, 28, 48, 50, 52, 55,](#)  
[57, 61, 69, 82, 84, 85](#)
- pmx\_vpc, [57, 70, 71–74](#)
- pmx\_vpc\_bin, [56, 57, 70, 71, 72–74](#)
- pmx\_vpc\_ci, [56, 57, 70, 71, 71, 73, 74](#)
- pmx\_vpc\_obs, [56, 57, 70–72, 72, 73, 74](#)
- pmx\_vpc\_pi, [56, 57, 70–73, 73, 74](#)
- pmx\_vpc\_rug, [56, 57, 70–73, 74](#)
- pmxOptions, [38](#)
- print.abbreviation, [74](#)
- print.configs, [75](#)
- print.pmx\_gpar, [76](#)
- print.pmxClass, [75](#)
- print.pmxConfig, [76](#)
- read\_input, [77](#)
- read\_mlx\_ind\_est, [77](#)
- read\_mlx\_par\_est, [78](#)
- read\_mlx\_pred, [78](#)
- residual, [34, 79, 82, 85](#)
- residual\_scatter, [80](#)
- set\_abbrev, [83](#)
- set\_data, [16–21, 28, 51, 54, 70, 81, 84, 85](#)
- set\_plot, [16–21, 28, 70, 84, 85](#)
- theme, [69](#)
- theophylline, [86](#)
- vars(), [14](#)
- wrap\_formula, [86](#)