

Package ‘ggPMX’

June 23, 2017

Title to be determined

Description A ggplot2-based tool to facilitate diagnostic plots for NLME models.

Version 0.5.1

URL <http://www.github.com/someuser/somerepo>

BugReports <http://www.github.com/someuser/somerepo/issues>

Depends R (>= 3.2.3)

Imports data.table,
yaml,
ggthemes,
R6,
gtable,
ggplot2 (>= 2.2.0),
magrittr,
stringr,
assertthat,
checkmate

License file LICENSE

LazyData true

Suggests rmarkdown,
knitr,
testthat

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

R topics documented:

abbrev	2
add_draft	3
configs	3
distrib	4
FacetWrapPaginate	4
facet_wrap_paginate	5
getPmxOption	6

get_data	7
get_plot	7
get_plot_config	8
ggPMX	9
individual	9
is.pmx_gpar	10
load_config	10
load_data_set	11
load_source	11
l_left_join	12
n_pages	12
plot_names	13
plot_pmx	13
plot_pmx.distrib	14
plot_pmx.individual	14
plot_pmx.pmx_gpar	15
plot_pmx.residual	15
pmx	16
pmxOptions	17
pmx_filter	18
pmx_gpar	18
pmx_mlx	19
pmx_theme	20
pmx_update	20
print.configs	21
print.pmxClass	21
print.pmxConfig	22
print.pmx_gpar	22
read_input	23
read_mlx_ind_est	23
read_mlx_par_est	24
read_mlx_pred	24
residual	25
set_plot	25
shrinkage	26
shrinkage_layer	27
[.pmx_gpar	27

Index **28**

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

Description

Give the whole abbreviation definition

Usage

abbrev(param)

Arguments

param abbreviation term

Value

characater abbreviation defintion

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, color)
```

Arguments

label draft layer default to DRAFT
size size of the annotation
color color of the annotation default to grey50

Value

ggplot2 annotation

Examples

```
add_draft("DRAFT", size = 5, color = "grey50")
```

configs	<i>List configurations</i>
---------	----------------------------

Description

List configurations

Usage

```
configs(sys = "mlx")
```

Arguments

sys can be mlx, if missed all configurations will be listed

Value

names of the config

Examples

```
configs()
```

distrib	<i>creates a graphic distribution object</i>
---------	----------------------------------------------

Description

creates a graphic distribution object

Usage

```
distrib(labels, has.jitter = TRUE, jitter = list(shape = 2, color =
  "grey50", width = 0.1), facets = list(scales = "free", nrow = 3),
  type = c("box", "hist"), has.shrink = FALSE, dname = NULL, ...)
```

Arguments

labels	list of texts/titles used within the plot
jitter	list set jitter parameter
facets	list set the facet setting in case of histogram plot
type	box for boxplot or histogram
dname	name of dataset to be used
...	others graphics arguments passed to pmx_gpar internal object.

Value

distrib object

See Also

Other plot_pmx: [individual](#), [plot_pmx.distrib](#), [plot_pmx.individual](#), [plot_pmx.pmx_gpar](#), [plot_pmx.residual](#), [plot_pmx](#), [residual](#)

FacetWrapPaginate	<i>Extend facet_wrap using ggproto</i>
-------------------	----------------------------------------

Description

Extend facet_wrap using ggproto

 facet_wrap_paginate *Split facet_wrap over multiple plots*

Description

This extension to [facet_wrap](#) will allow you to split a faceted plot over multiple pages. You define a number of rows and columns per page as well as the page number to plot, and the function will automatically only plot the correct panels. Usually this will be put in a loop to render all 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	Either a formula or character vector. Use either a one sided formula, $\sim a + b$, 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 labeller() . See label_value 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", "bottom", "left", "right")</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.

Examples

```
# Calculate the number of pages with 9 panels per page
n_pages <- ceiling(
  length(levels(diamonds$cut)) * length(levels(diamonds$clarity)) / 9
)

# Draw each page
for (i in seq_len(n_pages)) {
  ggplot(diamonds) +
    geom_point(aes(carat, price), alpha = 0.1) +
    facet_wrap_paginate(~cut:clarity, ncol = 3, nrow = 3, page = i)
}
```

getPmxOption

Get ggPMX Option

Description

Get ggPMX Option

Usage

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

Arguments

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

Examples

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

## End(Not run)
```

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

Description

Get controller data set

Usage

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

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_plot_config](#), [get_plot](#), [plot_names](#), [pmx_mlx](#), [pmx_update](#), [pmx](#), [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_data](#), [get_plot_config](#), [plot_names](#), [pmx_mlx](#), [pmx_update](#), [pmx](#), [set_plot](#)

Examples

```
## Not run:
library(ggPMX)
ctr <- pmx_mlx("standing")
## get all pages or some pages
p1 <- ctr %>% get_plot("indiv")
p2 <- ctr %>% get_plot("indiv", napge=1)
p3 <- ctr %>% get_plot("indiv", napge=c(1,3))
## get distribution plot
pdistri <- ctr %>% get_plot("distri")

## End(Not run)
```

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_data](#), [get_plot](#), [plot_names](#), [pmx_mlx](#), [pmx_update](#), [pmx](#), [set_plot](#)

Examples

```
## Not run:
pmxOptions(work_dir = WORK_DIR)
ctr <- pmx_mlx(config = "standing")
ctr %>% set_plot("IND", pname = "indiv1")
get_plot_config("distr1")

## End(Not run)
```


ggPMX

ggPMX: A ggplot2 toolbox for Nonlinear Mixed-Effect Model graphical

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>.

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.

individual

Create individual fits object

Description

Create individual fits object

Usage

```
individual(labels, has.curve = TRUE, facets = list(ncol = 3, nrow = 4),
  dname = NULL, ...)
```

Arguments

labels	plot tesxts. labels, axis,
has.curve	logical if TRUE add predictions lines
facets	list facets settings nrow/ncol
dname	name of dataset to be used
...	others graphics arguments passed to pmx_gpar internal object.

Value

individual fit object

See Also

[plot_pmx.individual](#)

Other plot_pmx: [distrib](#), [plot_pmx.distrib](#), [plot_pmx.individual](#), [plot_pmx.pmx_gpar](#), [plot_pmx.residual](#), [plot_pmx](#), [residual](#)

Examples

```
## get individual plot using package default data set
library(ggPMX)
ctr <- pmx_mlx(config = "standing")
## display the first page of the individual plot
ctr %>% get_plot("indiv",1)
```

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>Get data source config</i>
-------------	-------------------------------

Description

Get data source config

Usage

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

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

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, include, exclude)
```

Arguments

sys	type cane mlx/nom
path	character directory path containing all sources.
dconf	configuration object
include	list of data sets to load, if missed (the default behavior) load all sources.

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](#) and [facet_grid_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

Examples

```
p <- ggplot(diamonds) +
  geom_point(aes(carat, price), alpha = 0.1) +
  facet_wrap_paginate(~cut:clarity, ncol = 3, nrow = 3, page = 1)
n_pages(p)
```

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_data](#), [get_plot_config](#), [get_plot](#), [pmx_mlx](#), [pmx_update](#), [pmx](#), [set_plot](#)

plot_pmx	<i>Generic plot method for pmx objects</i>
----------	--------------------------------------------

Description

Generic plot method for pmx objects

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](#), [individual](#), [plot_pmx.distrib](#), [plot_pmx.individual](#), [plot_pmx.pmx_gpar](#), [plot_pmx.residual](#), [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

Value

ggplot2 plot

See Also

[distrib](#)

Other plot_pmx: [distrib](#), [individual](#), [plot_pmx.individual](#), [plot_pmx.pmx_gpar](#), [plot_pmx.residual](#), [plot_pmx](#), [residual](#)

plot_pmx.individual	<i>Plot individual fits</i>
---------------------	-----------------------------

Description

Plot individual fits

Usage

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

Arguments

x	individual object
dx	data set
include	list of individual ID to plot, if missing all ID will be plotted

Value

a list of ggplot2

See Also

Other plot_pmx: [distrib](#), [individual](#), [plot_pmx.distrib](#), [plot_pmx.pmx_gpar](#), [plot_pmx.residual](#), [plot_pmx](#), [residual](#)

plot_pmx.pmx_gpar	<i>The ggPMX base plot function</i>
-------------------	-------------------------------------

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](#), [individual](#), [plot_pmx.distrib](#), [plot_pmx.individual](#), [plot_pmx.residual](#), [plot_pmx](#), [residual](#)

plot_pmx.residual	<i>Plot residual object</i>
-------------------	-----------------------------

Description

Plot residual object

Usage

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

Arguments

x	residual object
dx	data set

Value

ggplot2 object

See Also

[residual](#)

Other plot_pmx: [distrib](#), [individual](#), [plot_pmx.distrib](#), [plot_pmx.individual](#), [plot_pmx.pmx_gpar](#), [plot_pmx](#), [residual](#)

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

Description

Create a pmx object from a data source

Usage

```
pmx(config, sys = c("mlx", "nm"), directory, input, dv, dvid, cats, conts,
    occ)
```

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/OTHERS
directory	where the files are located. This is an optional parameter by default pmw will look in work_dir pmx options: <code>getPmxOptions("work_dir")</code>
input	character complete path to the modelling input file
dv	character the name of measurable variable used in the input modelling file
dvid	character observation type parameter
cats	character vector of categorical covariates
conts	character vector of continuous covariates
occ	character occasional covariate variable name

Value

a pmxClass object

See Also

Other pmxclass: [get_data](#), [get_plot_config](#), [get_plot](#), [plot_names](#), [pmx_mlx](#), [pmx_update](#), [set_plot](#)

Examples

```
## Not run:

## Example to create the controller object:
workdir <- "~/SVN/ggPMX_doc/USER_EXAMPLES/MLX/PK_NO_COVARIATE"
WORK_DIR <- file.path(workdir, "RESULTS")
input_file <- file.path(workdir, "oral_data.txt")
ctr <- pmx(sys="mlx",
           config = "standing",
           directory = WORK_DIR, input = input_file, dv="Y")
## Better option is to use pmxOptions
pmxOptions(work_dir = WORK_DIR, input = input_file, dv = "Y")
## Now the latter call becomes
ctr <- pmx(sys="mlx", config = "standing")
## Or even simpler
ctr1 <- pmx_mlx("standing")

## End(Not run)
```

pmxOptions	<i>Get or set ggPMX options</i>
------------	---------------------------------

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

- **work_dir:** working directory containing input files like estimates.txt, finedgrid,...
- **input:** path to modelling input file
- **dv:** measurable variable name default to Y
- **cats:** *[Optional]* character covariate categorical variables
- **conts:** *[Optional]* covariate continuous variables
- **occ:** *[Optional]* occasional type variable
- **strats:** *[Optional]* extra variables names can be used for stratification

Examples

```
## Not run:
pmxOptions(work_dir = WORK_DIR, input = input_file, dv = "Y")

## End(Not run)
```

pmx_filter	<i>filter data in a pmx controller</i>
------------	----------------------------------------

Description

filter data in a pmx controller

Usage

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

```
## Not run:
## example of global filter
ctr <- pmx_mlx("standing")
ctr %>% pmx_filter(data_set = "prediction", ID == 5 & TIME <2)
ctr %>% get_data("prediction")

## End(Not run)
```

pmx_gpar	<i>Handling pmx Graphical parameters</i>
----------	------------------------------------------

Description

Handling pmx Graphical parameters

Usage

```
pmx_gpar(labels, axis.title = c(size = 12), axis.text = c(size = 14),
  ranges = NULL, has.smooth = FALSE, smooth = list(se = FALSE, linetype =
  2, size = 1.5, method = "loess"), has.band = FALSE, band = list(y = c(-2,
  2), linetype = 2, size = 0.5), is.draft = TRUE, draft = list(size = 20,
  label = "DRAFT", color = "grey50"), discrete = FALSE, ...)
```

Arguments

labels	list of labels, like title, subtitle, xlab , ylab
axis.title	list of axis title parameter : font size
axis.text	list of axis title parameter : font size
ranges	limits of x/y ranges
has.smooth	logical if set to TRUE add smooth layer
smooth	smooth layer parameters
has.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)
...	extra arguments (not used yet)

Details

This object contains all general graphic settings. It used internally by all pmx_plot(generic fuction) to set the default behavior.

Value

An object of class "pmx_gpar".

pmx_mlx	<i>Wrapper to pmx constructor</i>
---------	-----------------------------------

Description

Wrapper to pmx constructor

Usage

```
pmx_mlx(config, directory, input, dv, dvid, cats, conts, occ)
```

Arguments

config	the config name
directory	the data directory (working diectory)
input	the input file
dv	the dv parameter

Value

pmxClass object

See Also

Other pmxclass: [get_data](#), [get_plot_config](#), [get_plot](#), [plot_names](#), [pmx_update](#), [pmx](#), [set_plot](#)

pmx_theme	<i>Define ggPMX theme</i>
-----------	---------------------------

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, filter = NULL, ..., pmxgpar = NULL)
```

Arguments

ctr	pmxClass controller object
pname	character the plot name to update
filter	optional filter which will be applied to plotting data
...	others graphical parameters given to set the plot
pmxgpar	a object of class pmx_gpar possibly the output of the pmx_gpar function.

Value

controller object with the plot updated

See Also

Other pmxclass: [get_data](#), [get_plot_config](#), [get_plot](#), [plot_names](#), [pmx_mlx](#), [pmx](#), [set_plot](#)

print.configs	<i>S3 method to print configs</i>
---------------	-----------------------------------

Description

S3 method to print configs

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
...	additional arguments to pass to print

Value

print object to screen

print.pmxConfig	<i>S3 method print pmxConfig object</i>
-----------------	-----------------------------------------

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	<i>Print pmx_gpar object</i>
----------------	------------------------------

Description

Print pmx_gpar object

Usage

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

Arguments

x	pmx_gpar object
...	argument passed to print (to statisfy 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 = NULL, dvid, covariates = "")
```

Arguments

ipath	full path of the input 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

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

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

Value

data.table object

residual	Create a residual object
----------	--------------------------

Description

Create a residual object

Usage

```
residual(x, y, labels = NULL, point = NULL, add_hline = TRUE,
         dname = NULL, ...)
```

Arguments

x	x axis aesthetics
y	y axis aesthetics
labels	list that contain title, subtitle, axis labels
point	geom point graphical parameters
add_hline	logical if TRUE add horizontal line y=0 (TRUE by default)
dname	name of dataset to be used
...	others graphics arguments passed to pmx_gpar internal object.

Value

a residual object

See Also

[plot_pmx.residual](#)

Other `plot_pmx`: [distrib](#), [individual](#), [plot_pmx.distrib](#), [plot_pmx.individual](#), [plot_pmx.pmx_gpar](#), [plot_pmx.residual](#), [plot_pmx](#)

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", "RES"), pname, filter = NULL, ...)
```

Arguments

ctr	pmxClass controller object
ptype	plot type can be: <ul style="list-style-type: none"> • "IND" Individual plot type: individual • "DIS" Distribution plot type : distrib • "RES" Residual plot type : residual
pname	plot name, if missing it will be created using function aesthetics
filter	optional filter which will be applied to plotting data
...	other plot parameters to configure pmx_gpar .

Value

invisible ctr object

See Also

Other pmxclass: [get_data](#), [get_plot_config](#), [get_plot](#), [plot_names](#), [pmx_mlx](#), [pmx_update](#), [pmx](#)

shrinkage	<i>Compute shrinkage</i>
-----------	--------------------------

Description

Compute shrinkage

Usage

```
shrinkage(estimates, eta, sys = "mlx", fun = c("sd", "var"))
```

Arguments

estimates	parameters estimation data
eta	individual estimation data
sys	data system
fun	can be either sd or var

Value

data.table computing the shrinkage by effect

shrinkage_layer	Create shrinkage layer
-----------------	------------------------

Description

Create shrinkage layer

Usage

```
shrinkage_layer(ctr, type = "hist", color = "red", size = 5,  
  pos_ = 0.75, fun = "sd", x_ = 0)
```

Arguments

ctr	controller
-----	------------

Value

ggplot2 layer

[.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, 4

[.pmx_gpar, 27

abbrev, 2

add_draft, 3

configs, 3

distrib, 4, 9, 13–16, 25, 26

facet_grid_paginate, 12

facet_wrap, 5

facet_wrap_paginate, 5, 12

FacetWrapPaginate, 4

get_data, 7, 7, 8, 13, 16, 19, 20, 26

get_plot, 7, 7, 8, 13, 16, 19, 20, 26

get_plot_config, 7, 8, 13, 16, 19, 20, 26

getPmxOption, 6

ggPMX, 9

ggPMX-package (ggPMX), 9

individual, 4, 9, 13–16, 25, 26

is.pmx_gpar, 10

l_left_join, 12

label_value, 5

labeller, 5

load_config, 10

load_data_set, 11

load_source, 11

n_pages, 12

plot_names, 7, 8, 13, 16, 19, 20, 26

plot_pmx, 4, 9, 13, 14–16, 25

plot_pmx.distrib, 4, 9, 13, 14, 15, 16, 25

plot_pmx.individual, 4, 9, 13, 14, 14, 15,
16, 25

plot_pmx.pmx_gpar, 4, 9, 13–15, 15, 16, 25

plot_pmx.residual, 4, 9, 13–15, 15, 25

pmx, 7, 8, 13, 16, 19, 20, 26

pmx_filter, 18

pmx_gpar, 4, 9, 13, 18, 20, 25, 26

pmx_mlx, 7, 8, 13, 16, 19, 20, 26

pmx_theme, 20

pmx_update, 7, 8, 13, 16, 19, 20, 26

pmxOptions, 17

print.configs, 21

print.pmx_gpar, 22

print.pmxClass, 21

print.pmxConfig, 22

read_input, 23

read_mlx_ind_est, 23

read_mlx_par_est, 24

read_mlx_pred, 24

residual, 4, 9, 13–16, 25, 26

set_plot, 7, 8, 13, 16, 19, 20, 25

shrinkage, 26

shrinkage_layer, 27

theme, 20