Package 'xpose'

January 7, 2025

Type Package

Title Diagnostics for Pharmacometric Models

Version 0.4.19

Description Diagnostics for non-linear mixed-effects (population)
models from 'NONMEM' <https://www.iconplc.com/solutions/technologies/nonmem/>.
'xpose' facilitates data import, creation of numerical run summary
and provide 'ggplot2'-based graphics for data exploration and model
diagnostics.

Depends R (>= 3.5.0), ggplot2 (>= 3.1.0)

Imports dplyr (>= 0.8.0), ggforce (>= 0.2.0), grDevices, purrr (>=
0.3.0), readr (>= 2.1.0), rlang (>= 0.3.0), stringr (>= 1.4.0),
tibble (>= 2.1.0), tidyr (>= 0.8.0), utils, stats, vpc (>=
1.1.0)

Suggests here, gridExtra, rmarkdown, knitr, testthat, plotly, webshot,
mvtnorm

License LGPL-3

URL https://uupharmacometrics.github.io/xpose/,
https://github.com/UUPharmacometrics/xpose/

BugReports https://github.com/UUPharmacometrics/xpose/issues/

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RoxygenNote 7.3.2

NeedsCompilation no

Author Benjamin Guiastrennec [aut, cre, cph],
Andrew C. Hooker [aut, cph],
Sebastian Ueckert [aut, cph],
Mike K. Smith [ctb],
Mats O. Karlsson [aut, cph]

Maintainer Benjamin Guiastrennec <guiastrennec@gmail.com>

Contents

Index **69**

---

amt_vs_idv *Compartment kinetics*

Description

Plot of the change in compartment amounts over the independent variable

Usage

```
amt_vs_idv(
  xpdb,
  mapping = NULL,
  group = "ID",
  drop_fixed = TRUE,
  type = "l",
  title = "Compartments amount vs. @x | @run",
  subtitle = "Ofv: @ofv",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  facets,
  .problem,
  quiet,
  ...
)
```

Arguments

xpdb	An xpose database object.
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. `point_color`).
group	Grouping variable to be used for lines.
drop_fixed	Should columns that only have a single unique value (i.e. fixed) be dropped.
type	String setting the type of plot to be used. Can be points 'p', line 'l', smooth 's' and text 't' or any combination of the four.
title	Plot title. Use NULL to remove.
subtitle	Plot subtitle. Use NULL to remove.
caption	Page caption. Use NULL to remove.
tag	Plot identification tag. Use NULL to remove.
log	String assigning logarithmic scale to axes, can be either '', 'x', y' or 'xy'.

facets	Either a character string to use `facet_wrap_paginate` or a formula to use `facet_grid_paginate`.
.problem	The $problem number to be used. By default returns the last estimation problem.
quiet	Logical, if `FALSE` messages are printed to the console.
...	Any additional aesthetics to be passed on xplot_scatter.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is layer_option where layer is one of the names defined in the list below and option is any option supported by this layer e.g. point_color = 'blue', smooth_method = 'lm', etc.

- point: options to `geom_point`
- line: options to `geom_line`
- guide: options to `geom_abline`
- smooth: options to `geom_smooth`
- text: options to `geom_text`
- xscale: options to `scale_x_continuous` or `scale_x_log10`
- yscale: options to `scale_y_continuous` or `scale_y_log10`

Faceting

Every xpose plot function has built-in faceting functionalities. Faceting arguments are passed to the functions `facet_wrap_paginate` when the `facets` argument is a character string (e.g. `facets = c('SEX', 'MED1')`) or `facet_grid_paginate` when facets is a formula (e.g. `facets = SEX~MED1`). All xpose plot functions accept all the arguments for the `facet_wrap_paginate` and `facet_grid_paginate` functions e.g. `dv_vs_ipred(xpdb_ex_pk,facets = SEX~MED1, ncol = 3, nrow = 3, page = 1, margins = TRUE, labeller = 'label_both')`.

Faceting options can either be defined in plot functions (e.g. `dv_vs_ipred(xpdb_ex_pk,facets = 'SEX')`) or assigned globally to an xpdb object via the xp_theme (e.g. `xpdb <- update_themes(xpdb_ex_pk, xp_theme = list(facets = 'SEX'))`). In the latter example all plots generate from this xpdb will automatically be stratified by 'SEX'.

By default, some plot functions use a custom stratifying variable named 'variable', e.g. eta_distrib(). When using the `facets` argument, 'variable' needs to be added manually e.g. `facets = c('SEX', 'variable')` or `facets = c('SEX', 'variable')`, but is optional, when using the `facets` argument in xp_theme variable is automatically added whenever needed.

Template titles

Template titles can be used to create highly informative diagnostics plots. They can be applied to any plot title, subtitle, caption and tag. Template titles are defined via a single string containing key variables staring with a '@' (e.g. '@ofv') which will be replaced by their actual value when rendering the plot. For example ''@run, @nobs observations in @nind subjects'' would become ''run001, 1022 observations in 74 subjects''. The available key variables are listed under `template_titles`.

See Also

xplot_scatter

Examples

```
amt_vs_idv(xpdb_ex_pk, nrow = 2, ncol = 1)
```

---

data_opt	*Create options for data import*

Description

Provide a list of options to the general plotting functions such as xplot_scatter in order to create appropriate data input for ggplot2.

Usage

```
data_opt(
  .problem = NULL,
  .subprob = NULL,
  .method = NULL,
  .source = "data",
  simtab = FALSE,
  filter = NULL,
  tidy = FALSE,
  index_col = NULL,
  value_col = NULL,
  post_processing = NULL
)
```

Arguments

.problem	The problem to be used, by default returns the last one.
.subprob	The subproblem to be used, by default returns the last one.
.method	The estimation method to be used, by default returns the last one.
.source	Define the location of the data in the xpdb. Should be either 'data' to use the output tables or the name of an output file attached to the xpdb.
simtab	Only used when 'data' is defined as the source and '.problem' is default. Should the data be coming from an estimation or a simulation table.
filter	A function used to filter the data e.g. filter = function(x) x[x$TIME > 20,] where x is the data.
tidy	Logical, whether the data should be transformed to tidy data.
index_col	Only used when 'tidy' is defined a TRUE and value_col is NULL. Column names to use as index when tidying the data.

value_col	Only used when 'tidy' is defined a TRUE and index_col is NULL. Column names to be stacked when tidying the data.
post_processing	
	A function used to modify the data after it has been tidied up e.g. post_processing = function(x) dplyr::mutate(.data = x, variable = as.factor(.$variable)) where x is the tidy data.

See Also

xplot_distrib xplot_qq xplot_scatter

Examples

```
data_opt(.problem = 1, .source = 'data', simtab = TRUE)
```

---

distrib_plot   *Distribution plots of ETA and parameters*

Description

Histograms and density plots of the ETA and parameter values.

Usage

```
prm_distrib(
  xpdb,
  mapping = NULL,
  drop_fixed = TRUE,
  type = "hr",
  title = "Parameter distribution | @run",
  subtitle = "Based on @nind individuals",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  guide = FALSE,
  facets,
  .problem,
  quiet,
  ...
)

eta_distrib(
  xpdb,
  mapping = NULL,
  drop_fixed = TRUE,
  type = "hr",
```

```
      title = "Eta distribution | @run",
      subtitle = "Based on @nind individuals, Eta shrink: @etashk",
      caption = "@dir",
      tag = NULL,
      log = NULL,
      guide = FALSE,
      facets,
      .problem,
      quiet,
      ...
    )

    res_distrib(
      xpdb,
      mapping = NULL,
      res = "CWRES",
      type = "hr",
      title = "@x distribution | @run",
      subtitle = "Based on @nobs observations",
      caption = "@dir",
      tag = NULL,
      log = NULL,
      guide = FALSE,
      facets,
      .problem,
      quiet,
      ...
    )

    cov_distrib(
      xpdb,
      mapping = NULL,
      drop_fixed = TRUE,
      type = "hr",
      title = "Continuous covariates distribution | @run",
      subtitle = "Based on @nind individuals",
      caption = "@dir",
      tag = NULL,
      log = NULL,
      guide = FALSE,
      facets,
      .problem,
      quiet,
      ...
    )
```

Arguments

xpdb   An xpose database object.

mapping	List of aesthetics mappings to be used for the xpose plot (e.g. `point_color`).
drop_fixed	Should columns that only have a single unique value (i.e. fixed) be dropped.
type	String setting the type of plot to be used. Can be histogram 'h', density 'd', rug 'r' or any combination of the three.
title	Plot title. Use `NULL` to remove.
subtitle	Plot subtitle. Use `NULL` to remove.
caption	Page caption. Use `NULL` to remove.
tag	Plot identification tag. Use `NULL` to remove.
log	String assigning logarithmic scale to axes, can be either '', 'x', y' or 'xy'.
guide	Should the guide (e.g. reference distribution) be displayed.
facets	Either a character string to use [facet_wrap_paginate](#) or a formula to use [facet_grid_paginate](#).
.problem	The $problem number to be used. By default returns the last estimation problem.
quiet	Logical, if `FALSE` messages are printed to the console.
...	Any additional aesthetics to be passed on `xplot_scatter`.
res	Only used for `res_distrib`. Defines the type of residual to be used. Default is "CWRES".

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is layer_option where layer is one of the names defined in the list below and option is any option supported by this layer e.g. histogram_fill = 'blue', rug_sides = 'b', etc.

- histogram: options to `geom_histogram`
- density: options to `geom_density`
- rug: options to `geom_rug`
- xscale: options to `scale_x_continuous` or `scale_x_log10`
- yscale: options to `scale_y_continuous` or `scale_y_log10`

Faceting

Every xpose plot function has built-in faceting functionalities. Faceting arguments are passed to the functions [facet_wrap_paginate](#) when the `facets` argument is a character string (e.g. `facets = c('SEX', 'MED1')`) or [facet_grid_paginate](#) when facets is a formula (e.g. `facets = SEX~MED1`). All xpose plot functions accept all the arguments for the [facet_wrap_paginate](#) and [facet_grid_paginate](#) functions e.g. dv_vs_ipred(xpdb_ex_pk, facets = SEX~MED1, ncol = 3, nrow = 3, page = 1, margins = TRUE, labeller = 'label_both').

Faceting options can either be defined in plot functions (e.g. dv_vs_ipred(xpdb_ex_pk, facets = 'SEX')) or assigned globally to an xpdb object via the xp_theme (e.g. xpdb <- update_themes(xpdb_ex_pk, xp_theme = list(facets = 'SEX'))). In the latter example all plots generate from this xpdb will automatically be stratified by 'SEX'.

By default, some plot functions use a custom stratifying variable named 'variable', e.g. eta_distrib(). When using the `facets` argument, 'variable' needs to be added manually e.g. facets = c('SEX', 'variable') or facets = c('SEX', 'variable'), but is optional, when using the `facets` argument in xp_theme variable is automatically added whenever needed.

Template titles

Template titles can be used to create highly informative diagnostics plots. They can be applied to any plot title, subtitle, caption and tag. Template titles are defined via a single string containing key variables staring with a '@' (e.g. '@ofv') which will be replaced by their actual value when rendering the plot. For example ''@run, @nobs observations in @nind subjects'' would become ''run001, 1022 observations in 74 subjects''. The available key variables are listed under `template_titles`.

See Also

`xplot_distrib`

Examples

```
# Histogram of parameters
prm_distrib(xpdb_ex_pk, type = 'h')

# Density plot of etas with a rug
eta_distrib(xpdb_ex_pk, type = 'dr')

# Histogram of different residuals
res_distrib(xpdb_ex_pk, type = 'hr', res = c('IWRES', 'CWRES'))

# Density plot of continuous covariates
cov_distrib(xpdb_ex_pk, type = 'd')
```

---

dv_vs_pred   *Observations plotted against model predictions*

Description

Plot of observations (DV) vs population predictions (PRED), individual predictions (IPRED) or conditional population predictions (CPRED).

Usage

```
dv_vs_ipred(
  xpdb,
  mapping = NULL,
  group = "ID",
  type = "pls",
  title = "@y vs. @x | @run",
  subtitle = "Ofv: @ofv, Eps shrink: @epsshk",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  guide = TRUE,
  facets,
  .problem,
```

```
  quiet,
  ...
)

dv_vs_pred(
  xpdb,
  mapping = NULL,
  group = "ID",
  type = "pls",
  title = "@y vs. @x | @run",
  subtitle = "Ofv: @ofv",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  guide = TRUE,
  facets,
  .problem,
  quiet,
  ...
)
```

Arguments

xpdb	An xpose database object.
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. `point_color`).
group	Grouping variable to be used for lines.
type	String setting the type of plot to be used. Can be points 'p', line 'l', smooth 's' and text 't' or any combination of the four.
title	Plot title. Use `NULL` to remove.
subtitle	Plot subtitle. Use `NULL` to remove.
caption	Page caption. Use `NULL` to remove.
tag	Plot identification tag. Use `NULL` to remove.
log	String assigning logarithmic scale to axes, can be either '', 'x', y' or 'xy'.
guide	Enable guide display (e.g. unity line).
facets	Either a character string to use [facet_wrap_paginate](#) or a formula to use [facet_grid_paginate](#).
.problem	The $problem number to be used. By default returns the last estimation problem.
quiet	Logical, if `FALSE` messages are printed to the console.
...	Any additional aesthetics to be passed on xplot_scatter.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is layer_option where layer is one of the names defined in the list below and option is any option supported by this layer e.g. point_color = 'blue', smooth_method = 'lm', etc.

- point: options to `geom_point`

- line: options to `geom_line`

- guide: options to `geom_abline`

- smooth: options to `geom_smooth`

- text: options to `geom_text`

- xscale: options to `scale_x_continuous` or `scale_x_log10`

- yscale: options to `scale_y_continuous` or `scale_y_log10`

Faceting

Every xpose plot function has built-in faceting functionalities. Faceting arguments are passed to the functions facet_wrap_paginate when the `facets` argument is a character string (e.g. `facets = c('SEX', 'MED1')`) or facet_grid_paginate when facets is a formula (e.g. `facets = SEX~MED1`). All xpose plot functions accept all the arguments for the facet_wrap_paginate and facet_grid_paginate functions e.g. `dv_vs_ipred(xpdb_ex_pk, facets = SEX~MED1, ncol = 3, nrow = 3, page = 1, margins = TRUE, labeller = 'label_both')`.

Faceting options can either be defined in plot functions (e.g. `dv_vs_ipred(xpdb_ex_pk, facets = 'SEX')`) or assigned globally to an xpdb object via the xp_theme (e.g. `xpdb <- update_themes(xpdb_ex_pk, xp_theme = list(facets = 'SEX'))`). In the latter example all plots generate from this xpdb will automatically be stratified by 'SEX'.

By default, some plot functions use a custom stratifying variable named 'variable', e.g. `eta_distrib()`. When using the `facets` argument, 'variable' needs to be added manually e.g. `facets = c('SEX', 'variable')` or `facets = c('SEX', 'variable')`, but is optional, when using the `facets` argument in `xp_theme` variable is automatically added whenever needed.

Template titles

Template titles can be used to create highly informative diagnostics plots. They can be applied to any plot title, subtitle, caption and tag. Template titles are defined via a single string containing key variables staring with a '@' (e.g. '@ofv') which will be replaced by their actual value when rendering the plot. For example ''@run, @nobs observations in @nind subjects'' would become ''run001, 1022 observations in 74 subjects''. The available key variables are listed under template_titles.

See Also

xplot_scatter

Examples

```
dv_vs_pred(xpdb_ex_pk)

dv_vs_ipred(xpdb_ex_pk)
```

---

get_code *Access model code*

Description

Access model code from an xpdb object.

Usage

```
get_code(xpdb, .problem = NULL)
```

Arguments

xpdb	An xpose_data object from which the model code will be extracted.
.problem	The problem to be used, in addition, problem 0 is attributed to general output (e.g. NM-TRAN warnings in NONMEM). By default returns the entire code.

Value

A tibble of the parsed model.

See Also

[xpose_data](#), [read_nm_model](#)

Examples

```
parsed_model <- get_code(xpdb_ex_pk)
parsed_model
```

---

get_data *Access model output table data*

Description

Access model output table data from an xpdb object.

Usage

```
get_data(xpdb, table = NULL, .problem = NULL, quiet)
```

Arguments

xpdb	An xpose_data object from which the model output file data will be extracted.
table	Name of the output table to be extracted from the xpdb e.g. 'sdtab001'. Alternative to the '.problem' argument.
.problem	Accesses all tables from the specified problem. Alternative to the 'table' argument.
quiet	Logical, if FALSE messages are printed to the console.

Value

By default returns data from the last estimation problem. If only simulation problems are present then the data from last simulation will be returned instead. Object returned as tibble for single tables/problems or a named list for multiple tables/problems.

See Also

list_data, xpose_data, read_nm_tables

Examples

```
# By table name
sdtab <- get_data(xpdb_ex_pk, 'sdtab001')
sdtab

# By problem
tables <- get_data(xpdb_ex_pk, .problem = 1)
tables

# Tip to list available tables in the xpdb
print(xpdb_ex_pk)
```

---

get_file   *Access model output file data*

Description

Access model output file data from an xpdb object.

Usage

```
get_file(
  xpdb,
  file = NULL,
  ext = NULL,
  .problem = NULL,
  .subprob = NULL,
```

```
  .method = NULL,
  quiet
)
```

Arguments

xpdb	An xpose_data object from which the model output file data will be extracted.
file	Full name of the file to be extracted from the xpdb e.g. 'run001.phi'. Alternative to the 'ext' argument.
ext	Extension of the file to be extracted from the xpdb e.g. 'phi'. Alternative to the 'file' argument.
.problem	The problem to be used, by default returns the last one for each file.
.subprob	The subproblem to be used, by default returns the last one for each file.
.method	The estimation method to be used (e.g. 'foce', 'imp', 'saem'), by default returns the last one for each file.
quiet	Logical, if FALSE messages are printed to the console.

Value

A tibble for single file or a named list for multiple files.

See Also

[list_files](#), [xpose_data](#), [read_nm_files](#)

Examples

```
# Single file (returns a tibble)
ext_file <- get_file(xpdb_ex_pk, file = 'run001.ext')
ext_file

# Multiple files (returns a list)
files <- get_file(xpdb_ex_pk, file = c('run001.ext', 'run001.phi'))
files

# Tip to list available files in the xpdb
print(xpdb_ex_pk)
```

---

get_prm   *Access model parameters*

Description

Access model parameter estimates from an xpdb object.

Usage

```
get_prm(
  xpdb,
  .problem = NULL,
  .subprob = NULL,
  .method = NULL,
  digits = 4,
  transform = TRUE,
  show_all = FALSE,
  quiet
)
```

Arguments

xpdb	An xpose_data object from which the model output file data will be extracted.
.problem	The problem to be used, by default returns the last one for each file.
.subprob	The subproblem to be used, by default returns the last one for each file.
.method	The estimation method to be used, by default returns the last one for each file
digits	The number of significant digits to be displayed.
transform	Should diagonal OMEGA and SIGMA elements be transformed to standard deviation and off diagonal elements be transformed to correlations.
show_all	Logical, whether the 0 fixed off-diagonal elements should be removed from the output.
quiet	Logical, if FALSE messages are printed to the console.

Value

A tibble for single problem/subprob or a named list for multiple problem|subprob.

See Also

prm_table

Examples

```
# Store the parameter table
prm <- get_prm(xpdb_ex_pk, .problem = 1)

# Display parameters to the console
prm_table(xpdb_ex_pk, .problem = 1)
```

---

get_special   *Access special model data*

Description

Access special model data from an xpdb object.

Usage

```
get_special(xpdb, .problem = NULL, quiet)
```

Arguments

xpdb	An xpose_data object from which the special data will be extracted.
.problem	The problem to be used, by default returns the last one.
quiet	Logical, if FALSE messages are printed to the console.

Value

A list.

See Also

[list_special](#), [xpose_data](#)

Examples

```
special <- get_summary(xpdb_ex_pk)
special
```

---

get_summary   *Access model summary data*

Description

Access model summary data from an xpdb object.

Usage

```
get_summary(xpdb, .problem = NULL, .subprob = NULL, only_last = FALSE)
```

Arguments

xpdb	An xpose_data object from which the summary data will be extracted.
.problem	The .problem to be used, by default returns the last one for each label.
.subprob	The subproblem to be used, by default returns the last one for each label.
only_last	Logical, if TRUE only the last record for each label is returned in case of multiple problem and/or subproblem. If FALSE all values are returned.

Value

A tibble of model summary.

See Also

[xpose_data](#), [template_titles](#), [summary.xpose_data](#)

Examples

```
run_summary <- get_summary(xpdb_ex_pk)
run_summary
```

---

gg_themes   *An additional set of themes for ggplot2*

Description

An additional set of complete ggplot2 themes intended to make ggplot2 more readable when used in presentation or publications. These themes also bring the legend_position option without having to call the ggplot2 theme() function to modify a complete theme.

- theme_bw2: Black and white theme inspired by a theme from Gunnar Yngman.
- theme_readable: Light grey theme, with dimmed background and grid lines intended to bring the focus on the data.

Usage

```
theme_bw2(base_size = 11, base_family = "", legend_position = "right")

theme_readable(base_size = 11, base_family = "", legend_position = "right")
```

Arguments

base_size	Base font size.
base_family	Base font family.
legend_position	
	The position of legends defined as 'none', 'left', 'right', 'bottom', 'top', or a two-element numeric vector.

Examples

```
# With the gg_theme theme_readable() (default)
dv_vs_ipred(xpdb_ex_pk, facets = 'SEX')

# With the gg_theme theme_bw2()
xpdb_ex_pk %>%
 update_themes(gg_theme = theme_bw2()) %>%
 dv_vs_ipred(facets = 'SEX')
```

---

ind_plots	*Observations, individual predictions and population predictions plot-ted against the independent variable for every individual*

Description

Observations (DV), individual predictions (IPRED) and population predictions (PRED) plotted against the independent variable for every individual

Usage

```
ind_plots(
  xpdb,
  mapping = NULL,
  group = "variable",
  type = "lp",
  title = "Individual plots | @run",
  subtitle = "Ofv: @ofv, Eps shrink: @epsshk",
  caption = "@dir | Page @page of @lastpage",
  tag = NULL,
  log = NULL,
  facets,
  .problem,
  quiet,
  color = c("grey60", "deepskyblue4", "deepskyblue3"),
  point_alpha = c(0.8, 0, 0),
  line_linetype = c("blank", "solid", "55"),
  ...
)
```

Arguments

xpdb	An xpose database object.
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. `point_color`).
group	Grouping variable to be used for lines.
type	String setting the type of plot to be used. Can be points 'p', line 'l', smooth 's' and text 't' or any combination of the four.

title	Plot title. Use NULL to remove.
subtitle	Plot subtitle. Use NULL to remove.
caption	Page caption. Use NULL to remove.
tag	Plot identification tag. Use NULL to remove.
log	String assigning logarithmic scale to axes, can be either '', 'x', y' or 'xy'.
facets	Either a character string to use `facet_wrap_paginate` or a formula to use `facet_grid_paginate`.
.problem	The $problem number to be used. By default returns the last estimation problem.
quiet	Logical, if FALSE messages are printed to the console.
color	Changes the **lines, points and text** color. Should be a vector of 3 values (i.e. DV, IPRED, PRED). This color argument is a special case in xpose as it applies to three different layers (geom_line, geom_point and geom_text). This special case is due to the fact that in ggplot2 it is not possible to have two different color scales for different layers.
point_alpha	Points alpha, should be a vector of 3 values (i.e. DV, IPRED, PRED).
line_linetype	Lines linetype, should be a vector of 3 values (i.e. DV, IPRED, PRED).
...	Any additional aesthetics to be passed on xplot_scatter.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is layer_option where layer is one of the names defined in the list below and option is any option supported by this layer e.g. point_color = 'blue', smooth_method = 'lm', etc.

- point: options to geom_point
- line: options to geom_line
- guide: options to geom_abline
- smooth: options to geom_smooth
- text: options to geom_text
- xscale: options to scale_x_continuous or scale_x_log10
- yscale: options to scale_y_continuous or scale_y_log10

Faceting

Every xpose plot function has built-in faceting functionalities. Faceting arguments are passed to the functions `facet_wrap_paginate` when the facets argument is a character string (e.g. facets = c('SEX', 'MED1')) or `facet_grid_paginate` when facets is a formula (e.g. facets = SEX~MED1). All xpose plot functions accept all the arguments for the `facet_wrap_paginate` and `facet_grid_paginate` functions e.g. dv_vs_ipred(xpdb_ex_pk, facets = SEX~MED1, ncol = 3, nrow = 3, page = 1, margins = TRUE, labeller = 'label_both').

Faceting options can either be defined in plot functions (e.g. dv_vs_ipred(xpdb_ex_pk, facets = 'SEX')) or assigned globally to an xpdb object via the xp_theme (e.g. xpdb <- update_themes(xpdb_ex_pk, xp_theme = list(facets = 'SEX'))). In the latter example all plots generate from this xpdb will automatically be stratified by 'SEX'.

By default, some plot functions use a custom stratifying variable named 'variable', e.g. `eta_distrib()`. When using the `facets` argument, 'variable' needs to be added manually e.g. `facets = c('SEX', 'variable')` or `facets = c('SEX', 'variable')`, but is optional, when using the `facets` argument in `xp_theme` variable is automatically added whenever needed.

Template titles

Template titles can be used to create highly informative diagnostics plots. They can be applied to any plot title, subtitle, caption and tag. Template titles are defined via a single string containing key variables staring with a '@' (e.g. '@ofv') which will be replaced by their actual value when rendering the plot. For example ''@run, @nobs observations in @nind subjects'' would become ''run001, 1022 observations in 74 subjects''. The available key variables are listed under template_titles.

See Also

xplot_scatter

Examples

```
## Not run:
# Basic example
ind_plots(xpdb_ex_pk, page = 1,
          ncol = 2, nrow = 2)

## End(Not run)
```

---

irep	*Add simulation counter*

Description

Add a column containing a simulation counter (irep). A new simulation is counted everytime a value in x is lower than its previous value.

Usage

```
irep(x, quiet = FALSE)
```

Arguments

x	The column to be used for computing simulation number, usually the ID column.
quiet	Logical, if `FALSE` messages are printed to the console.

Examples

```
xpdb_ex_pk_2 <- xpdb_ex_pk %>%
 mutate(sim_id = irep(ID), .problem = 2)
```

---

list_nm_tables *List NONMEM output tables*

Description

List NONMEM output tables file names from a nm_model object.

Usage

```
list_nm_tables(nm_model = NULL)
```

Arguments

nm_model   An xpose nm_model object generated with [read_nm_model](#).

See Also

[read_nm_model](#), [read_nm_tables](#)

Examples

```
## Not run:
read_nm_model(file = 'run001.lst') %>%
  list_nm_tables()

## End(Not run)
```

---

list_vars *List available variables*

Description

Function listing all available variables in an xpdb object.

Usage

```
list_vars(xpdb, .problem = NULL)
```

Arguments

xpdb   An xpose_data object from which the model code will be extracted.

.problem   The problem to be used, by lists all available problems.

See Also

[set_var_types](#)

Examples

```
list_vars(xpdb_ex_pk)
```

---

list_xpdb   *List available datasets*

Description

Function providing a detailed listing of all available datasets in an xpdb object.

Usage

```
list_data(xpdb)

list_files(xpdb)

list_special(xpdb)
```

Arguments

xpdb   An xpose_data object to be evaluated

See Also

[get_data](#), [get_file](#), [get_special](#)

Examples

```
# List output tables data
list_data(xpdb_ex_pk)

# List output files data
list_files(xpdb_ex_pk)

# List special data
xpdb_ex_pk %>%
vpc_data(quiet = TRUE) %>%
list_special()
```

---

manual_nm_import   *Manually define nonmem tables to be imported*

Description

Manually provide names of the table files to be imported by xpose_data.

Usage

```
manual_nm_import(
  tab_names = c("sdtab", "mutab", "patab", "catab", "cotab", "mytab", "extra", "xptab",
    "cwtab"),
  tab_suffix = "",
  sim_suffix = "sim"
)
```

Arguments

tab_names	Provide the name of the tables to import e.g. 'sdtab', 'patab', 'cotab', 'catab' for NONMEM.
tab_suffix	Default is '', but can be changed to any character string to be used as suffix in the table names.
sim_suffix	Default is 'sim', but can be changed to any character string to be used as suffix in the simulation table names e.g. sdtab001sim.

Details

In order to be imported manually, table names must follow the following convention: `<tab_names><runno><tab/sim_suffi`
e.g. sdtab001sim. When the argument 'file' is used in xpose_data, the <runno> part is guessed
by taking the portion of the string starting by any digit and ending at the file extension e.g. `file =`
`run001a.mod` will guess <runno> as '001a'. If no valid <runno> can be guessed, xpose will return
an error. In this case it is advised to use the xpose_data argument 'runno' directly rather than 'file'
hence preventing xpose from having to guess <runno>.

Note that with manual table import xpose still reads in the NONMEM model file in order to generate
the run summary.

See Also

[xpose_data](#)

Examples

```
## Not run:
# Import all names specified by default as in xpose4
xpose_data(runno = '001', manual_import = manual_nm_import())

# Import a specific table name
```

```
xpose_data(runno = '001', manual_import = manual_nm_import(tab_names = 'mytab'))

## End(Not run)
```

---

minimization_plots   *Parameter value or gradient vs. iterations*

Description

Change of parameter value or gradient vs. iterations.

Usage

```
prm_vs_iteration(
  xpdb,
  mapping = NULL,
  group = "variable",
  type = "l",
  title = "Parameter @y vs. @x | @run",
 subtitle = "Method: @method, minimization time: @runtime\nTermination message: @term",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  guide = FALSE,
  facets,
  .problem,
  .subprob,
  .method,
  quiet,
  ...
)

grd_vs_iteration(
  xpdb,
  mapping = NULL,
  group = "variable",
  type = "l",
  title = "Gradient @y vs. @x | @run",
 subtitle = "Method: @method, minimization time: @runtime\nTermination message: @term",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  guide = FALSE,
  facets,
  .problem,
  .subprob,
  .method,
```

```
    quiet,
    ...
)
```

Arguments

xpdb	An xpose database object.
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. `point_color`).
group	Grouping variable to be used for lines.
type	String setting the type of plot to be used. Can be points 'p', line 'l', smooth 's' and text 't' or any combination of the four.
title	Plot title. Use `NULL` to remove.
subtitle	Plot subtitle. Use `NULL` to remove.
caption	Page caption. Use `NULL` to remove.
tag	Plot identification tag. Use `NULL` to remove.
log	String assigning logarithmic scale to axes, can be either '', 'x', y' or 'xy'.
guide	Enable guide display (e.g. unity line).
facets	Either a character string to use [facet_wrap_paginate](#) or a formula to use [facet_grid_paginate](#).
.problem	The $problem number to be used. By default returns the last estimation problem.
.subprob	The sub-problem number to be used. By default returns the last sub-problem associated with the selected problem.
.method	The estimation method to be used, by default returns the last one for each file
quiet	Logical, if `FALSE` messages are printed to the console.
...	Any additional aesthetics to be passed on `xplot_scatter`.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is layer_option where layer is one of the names defined in the list below and option is any option supported by this layer e.g. point_color = 'blue', smooth_method = 'lm', etc.

- point: options to `geom_point`
- line: options to `geom_line`
- guide: options to `geom_abline`
- smooth: options to `geom_smooth`
- text: options to `geom_text`
- xscale: options to `scale_x_continuous` or `scale_x_log10`
- yscale: options to `scale_y_continuous` or `scale_y_log10`

Faceting

Every xpose plot function has built-in faceting functionalities. Faceting arguments are passed to the functions [facet_wrap_paginate](#) when the `facets` argument is a character string (e.g. `facets = c('SEX', 'MED1')`) or [facet_grid_paginate](#) when facets is a formula (e.g. `facets = SEX~MED1`). All xpose plot functions accept all the arguments for the [facet_wrap_paginate](#) and [facet_grid_paginate](#) functions e.g. `dv_vs_ipred(xpdb_ex_pk,facets = SEX~MED1, ncol = 3, nrow = 3, page = 1, margins = TRUE, labeller = 'label_both')`.

Faceting options can either be defined in plot functions (e.g. `dv_vs_ipred(xpdb_ex_pk,facets = 'SEX')`) or assigned globally to an xpdb object via the `xp_theme` (e.g. `xpdb <- update_themes(xpdb_ex_pk, xp_theme = list(facets = 'SEX'))`). In the latter example all plots generate from this xpdb will automatically be stratified by 'SEX'.

By default, some plot functions use a custom stratifying variable named 'variable', e.g. `eta_distrib()`. When using the `facets` argument, 'variable' needs to be added manually e.g. `facets = c('SEX', 'variable')` or `facets = c('SEX', 'variable')`, but is optional, when using the `facets` argument in `xp_theme` variable is automatically added whenever needed.

Template titles

Template titles can be used to create highly informative diagnostics plots. They can be applied to any plot title, subtitle, caption and tag. Template titles are defined via a single string containing key variables staring with a '@' (e.g. '@ofv') which will be replaced by their actual value when rendering the plot. For example ''@run, @nobs observations in @nind subjects'' would become ''run001, 1022 observations in 74 subjects''. The available key variables are listed under [template_titles](#).

See Also

[xplot_scatter](#)

Examples

```
prm_vs_iteration(xpdb_ex_pk)

grd_vs_iteration(xpdb_ex_pk)
```

---

modify_xpdb   *Add, remove or rename variables in an xpdb*

Description

`mutate()` adds new variables and preserves existing ones. `select()` keeps only the listed variables; `rename()` keeps all variables.

Usage

```
## S3 method for class 'xpose_data'
mutate(.data, ..., .problem, .source, .where)

## S3 method for class 'xpose_data'
select(.data, ..., .problem, .source, .where)

## S3 method for class 'xpose_data'
rename(.data, ..., .problem, .source, .where)
```

Arguments

`.data`	An xpose database object.
`...`	Name-value pairs of expressions. Use `NULL` to drop a variable.
	These arguments are automatically quoted and evaluated in the context of the data frame. They support unquoting and splicing. See the dplyr vignette("programming") for an introduction to these concepts.
`.problem`	The problem from which the data will be modified
`.source`	The source of the data in the xpdb. Can either be 'data' or an output file extension e.g. 'phi'.
`.where`	A vector of element names to be edited in special (e.g. `.where = c('vpc_dat', 'aggr_obs')` with vpc).

Examples

```
# Mutate columns
xpdb_ex_pk %>%
 mutate(lnDV = log(DV),
        sim_count = irep(ID),
        .problem = 1) %>%
 dv_vs_idv(aes(y = lnDV))

# Rename/select columns
xpdb_ex_pk %>%
 select(ID:TAD, DV, EVID) %>%
 rename(TSLD = TAD) %>%
 dv_vs_idv(aes(x = TSLD))
```

---

pred_vs_idv	*Observations and model predictions plotted against the independent variable*

Description

Plot of observations (DV), individual model predictions (IPRED) and/or population predictions (PRED) plotted against the independent variable (IDV).

Usage

```
dv_vs_idv(
  xpdb,
  mapping = NULL,
  group = "ID",
  type = "pls",
  title = "@y vs. @x | @run",
  subtitle = "Ofv: @ofv",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  facets,
  .problem,
  quiet,
  ...
)

ipred_vs_idv(
  xpdb,
  mapping = NULL,
  group = "ID",
  type = "pls",
  facets,
  title = "@y vs. @x | @run",
  subtitle = "Ofv: @ofv, Eps shrink: @epsshk",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  .problem,
  quiet,
  ...
)

pred_vs_idv(
  xpdb,
  mapping = NULL,
  group = "ID",
  type = "pls",
  facets,
  title = "@y vs. @x | @run",
  subtitle = "Ofv: @ofv",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  .problem,
  quiet,
  ...
)
```

```
dv_preds_vs_idv(
  xpdb,
  mapping = NULL,
  group = "ID",
  type = "pls",
  facets,
  title = "Observations, Individual and Population Predictions vs. @x | @run",
  subtitle = "Ofv: @ofv, Eps shrink: @epsshk",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  .problem,
  quiet,
  ...
)
```

Arguments

xpdb	An xpose database object.
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. `point_color`).
group	Grouping variable to be used for lines.
type	String setting the type of plot to be used. Can be points 'p', line 'l', smooth 's' and text 't' or any combination of the four.
title	Plot title. Use `NULL` to remove.
subtitle	Plot subtitle. Use `NULL` to remove.
caption	Page caption. Use `NULL` to remove.
tag	Plot identification tag. Use `NULL` to remove.
log	String assigning logarithmic scale to axes, can be either ", 'x', y' or 'xy'.
facets	Either a character string to use `facet_wrap_paginate` or a formula to use `facet_grid_paginate`.
.problem	The $problem number to be used. By default returns the last estimation problem.
quiet	Logical, if `FALSE` messages are printed to the console.
...	Any additional aesthetics to be passed on `xplot_scatter`.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is layer_option where layer is one of the names defined in the list below and option is any option supported by this layer e.g. point_color = 'blue', smooth_method = 'lm', etc.

- point: options to `geom_point`
- line: options to `geom_line`
- guide: options to `geom_abline`
- smooth: options to `geom_smooth`

- text: options to `geom_text`

- xscale: options to `scale_x_continuous` or `scale_x_log10`

- yscale: options to `scale_y_continuous` or `scale_y_log10`

Faceting

Every xpose plot function has built-in faceting functionalities. Faceting arguments are passed to the functions [facet_wrap_paginate](#) when the `facets` argument is a character string (e.g. `facets = c('SEX', 'MED1')`) or [facet_grid_paginate](#) when facets is a formula (e.g. `facets = SEX~MED1`). All xpose plot functions accept all the arguments for the [facet_wrap_paginate](#) and [facet_grid_paginate](#) functions e.g. `dv_vs_ipred(xpdb_ex_pk,facets = SEX~MED1, ncol = 3, nrow = 3, page = 1, margins = TRUE, labeller = 'label_both')`.

Faceting options can either be defined in plot functions (e.g. `dv_vs_ipred(xpdb_ex_pk,facets = 'SEX')`) or assigned globally to an xpdb object via the xp_theme (e.g. `xpdb <- update_themes(xpdb_ex_pk, xp_theme = list(facets = 'SEX'))`). In the latter example all plots generate from this xpdb will automatically be stratified by 'SEX'.

By default, some plot functions use a custom stratifying variable named 'variable', e.g. `eta_distrib()`. When using the `facets` argument, 'variable' needs to be added manually e.g. `facets = c('SEX', 'variable')` or `facets = c('SEX', 'variable')`, but is optional, when using the `facets` argument in xp_theme variable is automatically added whenever needed.

Template titles

Template titles can be used to create highly informative diagnostics plots. They can be applied to any plot title, subtitle, caption and tag. Template titles are defined via a single string containing key variables staring with a '@' (e.g. '@ofv') which will be replaced by their actual value when rendering the plot. For example ''@run, @nobs observations in @nind subjects'' would become ''run001, 1022 observations in 74 subjects''. The available key variables are listed under [template_titles](#).

See Also

[xplot_scatter](#)

Examples

```
dv_vs_idv(xpdb_ex_pk)

ipred_vs_idv(xpdb_ex_pk)

pred_vs_idv(xpdb_ex_pk)

dv_preds_vs_idv(xpdb_ex_pk)
```

---

print.xpose_data *Print an xpose_data object*

Description

This function returns to the console a list of the files and options attached to an xpose_data object.

Usage

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

Arguments

x   An xpose_data object generated with xpose_data.

...   Ignored in this function

Examples

```
# Using the print function
print(xpdb_ex_pk)

# Or simply by writing the xpdb name
xpdb_ex_pk
```

---

print.xpose_plot *Draw an xpose_plot object*

Description

This function explicitly draw an xpose_plot and interprets keywords contained in labels.

Usage

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

Arguments

x   An xpose_plot object.

page   The page number to be drawn. Can be specified as vector or range of integer
               values.

...   Options to be passed on to the ggplot2 print method.

Examples

```
my_plot <- dv_vs_ipred(xpdb_ex_pk) +
             labs(title = 'A label with keywords: @nind individuals & @nobs observations')
# Using the print function
print(my_plot)

# Or simply by writting the plot object name
my_plot
```

---

prm_table   *Display a parameter estimates to the console*

Description

Display parameter estimates from an xpdb object to the console.

Usage

```
prm_table(
  xpdb,
  .problem = NULL,
  .subprob = NULL,
  .method = NULL,
  digits = 4,
  transform = TRUE,
  show_all = FALSE
)
```

Arguments

xpdb   An xpose_data object from which the model output file data will be extracted.

.problem   The problem to be used, by default returns the last one for each file.

.subprob   The subproblem to be used, by default returns the last one for each file.

.method   The estimation method to be used, by default returns the last one for each file

digits   The number of significant digits to be displayed.

transform   Should diagonal OMEGA and SIGMA elements be transformed to standard deviation and off diagonal elements be transformed to correlations.

show_all   Logical, whether the 0 fixed off-diagonal elements should be removed from the output.

See Also

[get_prm](#),

Examples

```
## Not run:
# Store the parameter table
prm <- get_prm(xpdb_ex_pk, .problem = 1)

# Display parameters to the console
prm_table(xpdb_ex_pk, .problem = 1)

## End(Not run)
```

---

qq_plot   *QQ plots of ETA and residuals*

Description

QQ plots of the ETA and model residuals.

Usage

```
prm_qq(
  xpdb,
  mapping = NULL,
  drop_fixed = TRUE,
  type = "p",
  title = "QQ plot of parameters | @run",
  subtitle = "Based on @nind individuals",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  guide = TRUE,
  facets,
  .problem,
  quiet,
  ...
)

eta_qq(
  xpdb,
  mapping = NULL,
  drop_fixed = TRUE,
  type = "p",
  title = "QQ plot of etas | @run",
  subtitle = "Based on @nind individuals, Eta shrink: @etashk",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  guide = TRUE,
```

```
    facets,
    .problem,
    quiet,
    ...
  )

  res_qq(
    xpdb,
    mapping = NULL,
    res = "CWRES",
    type = "p",
    title = "QQ plot of @sample | @run",
    subtitle = "Based on @nobs observations",
    caption = "@dir",
    tag = NULL,
    log = NULL,
    guide = TRUE,
    facets,
    .problem,
    quiet,
    ...
  )

  cov_qq(
    xpdb,
    mapping = NULL,
    drop_fixed = TRUE,
    type = "p",
    title = "QQ plot of continuous covariates | @run",
    subtitle = "Based on @nind individuals",
    caption = "@dir",
    tag = NULL,
    log = NULL,
    guide = TRUE,
    facets,
    .problem,
    quiet,
    ...
  )
```

Arguments

xpdb	An xpose database object.
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. point_color).
drop_fixed	Should columns that only have a single unique value (i.e. fixed) be dropped.
type	String setting the type of plot. Can only be points 'p'.
title	Plot title. Use NULL to remove.

subtitle	Plot subtitle. Use NULL to remove.
caption	Page caption. Use NULL to remove.
tag	Plot identification tag. Use NULL to remove.
log	String assigning logarithmic scale to axes, can be either ", 'x', y' or 'xy'.
guide	Should the guide (e.g. reference line) be displayed.
facets	Either a character string to use facet_wrap_paginate or a formula to use facet_grid_paginate.
.problem	The $problem number to be used. By default returns the last estimation problem.
quiet	Logical, if FALSE messages are printed to the console.
...	Any additional aesthetics to be passed on xplot_scatter.
res	Only used for res_qq. Defines the type of residual to be used. Default is "CWRES".

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is layer_option where layer is one of the names defined in the list below and option is any option supported by this layer e.g. point_color = 'blue', etc.

- point: options to geom_point

- guide: options to geom_abline

- xscale: options to scale_x_continuous or scale_x_log10

- yscale: options to scale_y_continuous or scale_y_log10

Faceting

Every xpose plot function has built-in faceting functionalities. Faceting arguments are passed to the functions facet_wrap_paginate when the facets argument is a character string (e.g. facets = c('SEX', 'MED1')) or facet_grid_paginate when facets is a formula (e.g. facets = SEX~MED1). All xpose plot functions accept all the arguments for the facet_wrap_paginate and facet_grid_paginate functions e.g. dv_vs_ipred(xpdb_ex_pk,facets = SEX~MED1, ncol = 3, nrow = 3, page = 1, margins = TRUE, labeller = 'label_both').

Faceting options can either be defined in plot functions (e.g. dv_vs_ipred(xpdb_ex_pk,facets = 'SEX')) or assigned globally to an xpdb object via the xp_theme (e.g. xpdb <- update_themes(xpdb_ex_pk, xp_theme = list(facets = 'SEX'))). In the latter example all plots generate from this xpdb will automatically be stratified by 'SEX'.

By default, some plot functions use a custom stratifying variable named 'variable', e.g. eta_distrib(). When using the facets argument, 'variable' needs to be added manually e.g. facets = c('SEX', 'variable') or facets = c('SEX', 'variable'), but is optional, when using the facets argument in xp_theme variable is automatically added whenever needed.

Template titles

Template titles can be used to create highly informative diagnostics plots. They can be applied to any plot title, subtitle, caption and tag. Template titles are defined via a single string containing key variables staring with a '@' (e.g. '@ofv') which will be replaced by their actual value when rendering the plot. For example ''@run, @nobs observations in @nind subjects'' would become ''run001, 1022 observations in 74 subjects''. The available key variables are listed under `template_titles`.

See Also

`xplot_distrib`

Examples

```
# QQ plot of parameters
prm_qq(xpdb_ex_pk)

# QQ plot of eta
eta_qq(xpdb_ex_pk)

# QQ plot of residuals
res_qq(xpdb_ex_pk, res = c('IWRES', 'CWRES'))

# QQ plot of continuous covariates
cov_qq(xpdb_ex_pk)
```

---

read_nm_files   *NONMEM output file import function*

Description

Quickly import NONMEM output files into R.

Usage

```
read_nm_files(
  runno = NULL,
  prefix = "run",
  ext = c(".ext", ".cor", ".cov", ".phi", ".grd", ".shk"),
  file = NULL,
  dir = NULL,
  quiet = FALSE
)
```

Arguments

runno	Run number to be evaluated.
prefix	Prefix of the model file names.
ext	A vector of the file extension to import. By default '.ext', '.cor', '.cov', '.phi', '.grd', '.shk' files are listed.
file	Names of the model output file to be imported. Alternative argument to prefix, runno and ext.
dir	Location of the model files.
quiet	Logical, if FALSE messages are printed to the console.

File path generation

The rules for model file names generation are as follow:

- with runno: the full path is generated as <dir>/<prefix><runno>.<ext> e.g. with dir = 'model/pk', prefix = 'run', runno = '001', ext = '.lst' the resulting path would be model/pk/run001.lst

- with file: the full path is generated as <dir>/<file> e.g. with dir = 'model/pk', file = 'run001.lst' the resulting path would also be model/pk/run001.lst. Note: in this case the file extension should be provided as part of the 'file' argument.

See Also

[xpose_data](), [read_nm_tables]()

Examples

```
## Not run:
# Using the `file` argument to import a model file:
ext_file <- read_nm_files(file = 'run001.ext', dir = 'models')

# Using the `runno` argument to import a model file:
ext_file <- read_nm_files(runno = '001', ext = '.ext', dir = 'models')

## End(Not run)
```

---

read_nm_model | *NONMEM model file parser*

Description

Parse NONMEM model files in R format

Usage

```
read_nm_model(
  runno = NULL,
  prefix = "run",
  ext = ".lst",
  file = NULL,
  dir = NULL,
  check_ext = TRUE
)
```

Arguments

runno	Run number to be used to generate model file name. Used in combination with `prefix` and `ext`.
prefix	Prefix to be used to generate model file name. Used in combination with `runno` and `ext`.
ext	Extension to be used to generate model file name. Should be one of '.lst' (default), '.out', '.res', '.mod' or '.ctl' for NONMEM.
file	Model file name (preferably a '.lst' file) containing the file extension. Alternative to `prefix`, `runno` and `ext` arguments.
dir	Location of the model files.
check_ext	Logical, if `TRUE` will provide an error message if the extension of the NONMEM input file is not one of '.lst', '.out', '.res', '.mod' or '.ctl' for NONMEM. If `FALSE` any file extension can be used.

Details

A NONMEM model output file (i.e. .lst, .out or .res) should preferably be provided to `read_nm_model` to allow for a more extensive xpose summary. However in some cases these output files may not contain the model code, thus preventing xpose from identifying the associated output tables names. In such cases xpose will attempt to read the associated model file (i.e. .mod or .ctl) instead to find the model code. Note: it is important that between the naming convention between the NONMEM output and the model file remains consistent e.g. run001.lst should be associated with run001.mod.

Value

A [tibble](#) of class model containing the following columns:

- problem: a numeric identifier for the $PROBLEM associated with the code.
- level: a unique numeric identifier to each subroutine block associated with the code.
- subroutine: a character identifier named after the 3 first letters of the subroutine name e.g. '$THETA' and '$TABLE' will become 'the' and 'tab' respectively. In addition all output from the .lst is labeled 'lst', the general nonmem output e.g. NM-TRAN messages are labelled 'oth'. With priors thp, tpv, omp, opd, sip, spd abbreviations are given to the THETAP, THETAPV, OMEGAP, etc.
- code: the code without comments or subroutine names e.g. '$THETA 0.5 ; TVCL' will return '0.5'.
- comment: the last comment of a record e.g. '0.5 ; Clearance (L/h) ; TVCL' will return 'TVCL'.

File path generation

The rules for model file names generation are as follow:

- with runno: the full path is generated as <dir>/<prefix><runno>.<ext> e.g. with dir
  = 'model/pk', prefix = 'run', runno = '001', ext = '.lst' the resulting path would be
  model/pk/run001.lst

- with file: the full path is generated as <dir>/<file> e.g. with dir = 'model/pk', file =
  'run001.lst' the resulting path would also be model/pk/run001.lst. Note: in this case
  the file extension should be provided as part of the 'file' argument.

See Also

xpose_data, read_nm_tables

Examples

```
## Not run:
# Using the `file` argument to import a model file:
nm_model <- read_nm_model(file = 'run001.lst', dir = 'models')

# Using the `runno` argument to import a model file:
nm_model <- read_nm_model(runno = '001', ext = '.lst', dir = 'models')

## End(Not run)
```

---

read_nm_tables *NONMEM output table import function*

Description

Quickly import NONMEM output tables into R. This function automatically detects the optimal
settings to import the tables from nonmem.

Usage

```
read_nm_tables(
  file = NULL,
  dir = NULL,
  combined = TRUE,
  rm_duplicates = TRUE,
  quiet = FALSE,
  simtab = NULL,
  ziptab = TRUE,
  ...
)
```

Arguments

`file`	A character vector of path to the files or a nm_table_list object created with `list_nm_tables`.
`dir`	Location of the model files.
`combined`	Logical value indicating whether multiple tables should be combined into a single one. If the number of rows does not match an error will be returned.
`rm_duplicates`	Logical value indicating whether duplicated columns should be removed.
`quiet`	Logical, if FALSE messages are printed to the console.
`simtab`	If TRUE only reads in simulation tables, if FALSE only reads estimation tables. Default NULL reads all tables.
`ziptab`	If TRUE search for the tables that have been compressed and renamed ´<file>.zip´.
`...`	Additional arguments to be passed to the [read_table](#) or [read_csv](#) functions.

Table format requirement

When using `read_nm_tables` with the `combined` argument set to FALSE an ID column must be present in all data tables. When `combined` is set to TRUE instead an ID column must be present in at least one table for each problem and for each 'firstonly' category. ID columns are required to properly combine/merge tables and removing NA records. If the ID column is missing from a table and `combined = FALSE` `read_nm_tables` will return the following warning: Unknown variables: `ID`. While the data is returned beware that NA records might be left in the data and the output should be checked carefully. If `combined = TRUE` `read_nm_tables` xpose is more strict and will return the following warning instead: Dropped `<tablenames>` due to missing required `ID` column..

Examples

```
## Not run:
# Import tables manually and return them as a list of individual tables
nm_tables <- read_nm_tables(file = c('sdtab001', 'patab001'),
                            dir = 'models', combined = FALSE)

# Import tables manually and return them as a single merged table
nm_tables <- read_nm_tables(file = c('sdtab001', 'patab001'),
                            dir = 'models', combined = TRUE)

# Import tables automatically (used internally by xpose_data())
nm_tables <- read_nm_model(file = 'run001.lst', dir = 'models') %>%
             list_nm_tables() %>%
             read_nm_tables()

# Passing arguments to readr via `...`
# (e.g. import columns as character and only first 10 rows)
nm_tables <- read_nm_tables(file = 'sdtab001', dir = 'models',
                            col_type = readr::cols(.default = 'c'),
                            n_max = 10)


## End(Not run)
```

---

res_vs_idv   *Residuals plotted against the independent variable*

Description

Model residuals plotted against the independent variable (IDV).

The residuals can be one of:

- RES: model residuals
- WRES: weighted model residuals
- CWRES: conditional weighted model residuals
- EWRES/ECWRES: Monte Carlo based model residuals
- NPDE: Normalized prediction distribution error

Usage

```
res_vs_idv(
  xpdb,
  mapping = NULL,
  res = "CWRES",
  group = "ID",
  type = "pls",
  title = "@y vs. @x | @run",
  subtitle = "Ofv: @ofv",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  guide = TRUE,
  facets,
  .problem,
  quiet,
  ...
)

absval_res_vs_idv(
  xpdb,
  mapping = NULL,
  res = "CWRES",
  group = "ID",
  type = "pls",
  title = "@y vs. @x  | @run",
  subtitle = "Ofv: @ofv",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  guide = FALSE,
```

```
    facets,
    .problem,
    quiet,
    ...
)
```

Arguments

xpdb	An xpose database object.
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. `point_color`).
res	Type of residual to be used. Default is "CWRES".
group	Grouping variable to be used for lines.
type	String setting the type of plot to be used. Can be points 'p', line 'l', smooth 's' and text 't' or any combination of the four.
title	Plot title. Use `NULL` to remove.
subtitle	Plot subtitle. Use `NULL` to remove.
caption	Page caption. Use `NULL` to remove.
tag	Plot identification tag. Use `NULL` to remove.
log	String assigning logarithmic scale to axes, can be either '', 'x', y' or 'xy'.
guide	Enable guide display (e.g. unity line).
facets	Either a character string to use [facet_wrap_paginate](#) or a formula to use [facet_grid_paginate](#).
.problem	The $problem number to be used. By default returns the last estimation problem.
quiet	Logical, if `FALSE` messages are printed to the console.
...	Any additional aesthetics to be passed on `xplot_scatter`.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is layer_option where layer is one of the names defined in the list below and option is any option supported by this layer e.g. point_color = 'blue', smooth_method = 'lm', etc.

- point: options to `geom_point`

- line: options to `geom_line`

- guide: options to `geom_abline`

- smooth: options to `geom_smooth`

- text: options to `geom_text`

- xscale: options to `scale_x_continuous` or `scale_x_log10`

- yscale: options to `scale_y_continuous` or `scale_y_log10`

Template titles

Template titles can be used to create highly informative diagnostics plots. They can be applied to any plot title, subtitle, caption and tag. Template titles are defined via a single string containing key variables staring with a '@' (e.g. '@ofv') which will be replaced by their actual value when rendering the plot. For example ''@run, @nobs observations in @nind subjects'' would become ''run001, 1022 observations in 74 subjects''. The available key variables are listed under template_titles.

See Also

xplot_scatter

Examples

```
# Standard residual
res_vs_idv(xpdb_ex_pk, res = c('IWRES', 'CWRES'))

# Absolute value of the residuals
absval_res_vs_idv(xpdb_ex_pk, res = 'CWRES')
```

---

res_vs_pred   *Residuals plotted against population predictions*

Description

Model residuals plotted against population predictions (PRED).

The residuals can be one of:

- RES: model residuals
- WRES: weighted model residuals
- CWRES: conditional weighted model residuals
- EWRES/ECWRES: Monte Carlo based model residuals
- NPDE: Normalized prediction distribution error

Usage

```
res_vs_pred(
  xpdb,
  mapping = NULL,
  res = "CWRES",
  group = "ID",
  type = "pls",
  title = "@y vs. @x | @run",
  subtitle = "Ofv: @ofv",
  caption = "@dir",
  tag = NULL,
```

```
  log = NULL,
  guide = TRUE,
  facets,
  .problem,
  quiet,
  ...
)

absval_res_vs_pred(
  xpdb,
  mapping = NULL,
  res = "CWRES",
  group = "ID",
  type = "pls",
  title = "@y vs. @x | @run",
  subtitle = "Ofv: @ofv",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  guide = FALSE,
  facets,
  .problem,
  quiet,
  ...
)
```

Arguments

xpdb	An xpose database object.
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. point_color).
res	Type of residual to be used. Default is "CWRES".
group	Grouping variable to be used for lines.
type	String setting the type of plot to be used. Can be points 'p', line 'l', smooth 's' and text 't' or any combination of the four.
title	Plot title. Use NULL to remove.
subtitle	Plot subtitle. Use NULL to remove.
caption	Page caption. Use NULL to remove.
tag	Plot identification tag. Use NULL to remove.
log	String assigning logarithmic scale to axes, can be either '', 'x', y' or 'xy'.
guide	Enable guide display (e.g. unity line).
facets	Either a character string to use facet_wrap_paginate or a formula to use facet_grid_paginate.
.problem	The $problem number to be used. By default returns the last estimation problem.
quiet	Logical, if FALSE messages are printed to the console.
...	Any additional aesthetics to be passed on xplot_scatter.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is layer_option where layer is one of the names defined in the list below and option is any option supported by this layer e.g. point_color = 'blue', smooth_method = 'lm', etc.

- point: options to `geom_point`

- line: options to `geom_line`

- guide: options to `geom_abline`

- smooth: options to `geom_smooth`

- text: options to `geom_text`

- xscale: options to `scale_x_continuous` or `scale_x_log10`

- yscale: options to `scale_y_continuous` or `scale_y_log10`

Faceting

Every xpose plot function has built-in faceting functionalities. Faceting arguments are passed to the functions [facet_wrap_paginate](#) when the `facets` argument is a character string (e.g. facets = c('SEX', 'MED1')) or [facet_grid_paginate](#) when facets is a formula (e.g. facets = SEX~MED1). All xpose plot functions accept all the arguments for the [facet_wrap_paginate](#) and [facet_grid_paginate](#) functions e.g. dv_vs_ipred(xpdb_ex_pk, facets = SEX~MED1, ncol = 3, nrow = 3, page = 1, margins = TRUE, labeller = 'label_both').

Faceting options can either be defined in plot functions (e.g. dv_vs_ipred(xpdb_ex_pk, facets = 'SEX')) or assigned globally to an xpdb object via the xp_theme (e.g. xpdb <- update_themes(xpdb_ex_pk, xp_theme = list(facets = 'SEX'))). In the latter example all plots generate from this xpdb will automatically be stratified by 'SEX'.

By default, some plot functions use a custom stratifying variable named 'variable', e.g. eta_distrib(). When using the `facets` argument, 'variable' needs to be added manually e.g. facets = c('SEX', 'variable') or facets = c('SEX', 'variable'), but is optional, when using the `facets` argument in xp_theme variable is automatically added whenever needed.

Template titles

Template titles can be used to create highly informative diagnostics plots. They can be applied to any plot title, subtitle, caption and tag. Template titles are defined via a single string containing key variables staring with a '@' (e.g. '@ofv') which will be replaced by their actual value when rendering the plot. For example ''@run, @nobs observations in @nind subjects'' would become ''run001, 1022 observations in 74 subjects''. The available key variables are listed under [template_titles](#).

See Also

[xplot_scatter](#)

Examples

```
# Standard residual
res_vs_pred(xpdb_ex_pk, res = c('IWRES', 'CWRES'))
```

```
# Absolute value of the residuals
absval_res_vs_pred(xpdb_ex_pk, res = 'CWRES')
```

---

set_vars   *Set variable type, label or units*

Description

Function designed to change the type, label or unit associated with variables.

Usage

```
set_var_types(xpdb, .problem = NULL, ..., auto_factor = TRUE, quiet)

set_var_labels(xpdb, .problem = NULL, ..., quiet)

set_var_units(xpdb, .problem = NULL, ..., quiet)
```

Arguments

xpdb	An xpose_data object.
.problem	The problem number to which the edits will be applied.
...	Specifications of the edits to be made to the xpdb index. Edits are made as type and variable pairs e.g. idv = 'TAD' will assign TAD to the type idv (independent variable).
auto_factor	With set_var_types only. If TRUE new columns assigned to the type 'catcov' will be converted to factor.
quiet	Logical, if FALSE messages are printed to the console.

Value

An xpose_data object

Recognized variable types

- a: Compartments' amount
- amt: Dose amount
- catcov: Categorical covariate
- contcov: Continuous covariate
- dv: Dependent variable
- dvid: DV identifier
- eta: Eta
- evid: Event identifier

- id: Subject identifier

- idv: Independent variable

- ipred: Individual model predictions

- mdv: Missing dependent variable

- na: Not attributed

- occ: Occasion flag

- param: Model parameter

- pred: Typical model predictions

- res: Residuals

See Also

list_vars

Examples

```
# Change variable type
xpdb_2 <- set_var_types(xpdb_ex_pk, .problem = 1, idv = 'TAD')

# Change labels
xpdb_2 <- set_var_labels(xpdb_2, .problem = 1, ALAG1 = 'Lag time', CL = 'Clearance', V = 'Volume')

# Change units
xpdb_2 <- set_var_units(xpdb_2, .problem = 1, ALAG1 = 'h', CL = 'L/h', V = 'L')
```

---

subset_xpdb *Subset datasets in an xpdb*

Description

Use `filter()` to select rows/cases where conditions are true. Unlike base subsetting, rows where the condition evaluates to NA are dropped. Use `slice()` to select row/cases by their position

Usage

```
## S3 method for class 'xpose_data'
filter(.data, ..., .problem, .source, .where)

## S3 method for class 'xpose_data'
slice(.data, ..., .problem, .source, .where)

## S3 method for class 'xpose_data'
distinct(.data, ..., .problem, .source, .where)
```

Arguments

`.data`	An xpose database object.
`...`	Name-value pairs of expressions. Use `NULL` to drop a variable.
	These arguments are automatically quoted and evaluated in the context of the data frame. They support unquoting and splicing. See the dplyr vignette("programming") for an introduction to these concepts.
`.problem`	The problem from which the data will be modified
`.source`	The source of the data in the xpdb. Can either be 'data' or an output file extension e.g. 'phi'.
`.where`	A vector of element names to be edited in special (e.g. `.where = c('vpc_dat', 'aggr_obs')` with vpc).

Examples

```
# Subset by condition
xpdb_ex_pk %>%
 filter(DV < 1, .problem = 1) %>%
 dv_vs_ipred()

# Subset by positions
xpdb_ex_pk %>%
 slice(1:100, .problem = 1) %>%
 dv_vs_ipred()

# Deduplicate rows
xpdb_ex_pk %>%
 distinct(TIME, .problem = 1) %>%
 dv_vs_ipred()
```

---

summarise_xpdb	*Group/ungroup and summarize variables in an xpdb*

Description

`group_by()` takes an existing table and converts it into a grouped table where operations are performed "by group". `ungroup()` removes grouping. `summarize()` reduces multiple values down to a single value.

Usage

```
## S3 method for class 'xpose_data'
group_by(.data, ..., .problem, .source, .where)

## S3 method for class 'xpose_data'
ungroup(x, ..., .problem, .source, .where)
```

```
## S3 method for class 'xpose_data'
summarise(.data, ..., .problem, .source, .where)

## S3 method for class 'xpose_data'
summarize(.data, ..., .problem, .source, .where)
```

Arguments

`.data`	An xpose database object.
`...`	Name-value pairs of expressions. Use NULL to drop a variable.
	These arguments are automatically quoted and evaluated in the context of the data frame. They support unquoting and splicing. See the dplyr vignette("programming") for an introduction to these concepts.
`.problem`	The problem from which the data will be modified
`.source`	The source of the data in the xpdb. Can either be 'data' or an output file extension e.g. 'phi'.
`.where`	A vector of element names to be edited in special (e.g. `.where = c('vpc_dat', 'aggr_obs')` with vpc).
x	Same as .data (used for consistency with dplyr functions).

Examples

```
# Create a distribution plot of Cmax
xpdb_ex_pk %>%
 group_by(ID, SEX, .problem = 1) %>%
 summarize(CMAX = max(DV), .problem = 1) %>%
 ungroup(.problem = 1) %>%
 xplot_distrib(aes(x = CMAX, density_fill = SEX), type = 'dr')
```

---

summary.xpose_data   *Summarizing xpose_data*

Description

This function returns a summary of an [xpose_data](#) to the console.

Usage

```
## S3 method for class 'xpose_data'
summary(object, .problem = NULL, ...)
```

Arguments

`object`	An xpose_data object generated with [xpose_data](#).
`.problem`	The problem to be used, by default returns the last one for each label.
`...`	Ignored in this function

Examples

```
summary(xpdb_ex_pk)
```

---

template_titles   *Template titles*

Description

Template titles can be used to create highly informative diagnostics plots. They can be applied to any plot title, subtitle, caption and the filename when saving with the [xpose_save](#) function.

Template titles are defined via a single string containing key variables staring with a @ (e.g. @ofv) which will be replaced by their actual value when rendering the plot. For example '@run, @nobs observations in @nind subjects' would become 'run001, 1022 observations in 74 subjects'

Many key variables are available:

@condn Condition number

@covtime Covariance matrix runtime

@data Model input data used

@descr Model description

@dir Model directory

@epsshk Epsilon shrinkage

@errors Run errors (e.g termination error)

@esampleseed ESAMPLE seed number (used in NPDE)

@etashk Eta shrinkage

@file Model file name

@label Model label

@method Estimation method or sim

@nesample Number of ESAMPLE (used in NPDE)

@nind Number of individuals

@nobs Number of observations

@nsig Number of significant digits

@nsim Number of simulations

@ofv Objective function value

@page and @lastpage Are respectively the page number and the number of the last page when faceting on multiple pages

@probn Problem number

@plotfun Name of the plot function

@ref Reference model

@run Model run name

@runtime Estimation/Sim runtime

@software Software used (e.g. NONMEM)

@simseed Simulation seed

@subroutine Differential equation solver

@timestart Run start time

@timestop Run stop time

@timeplot Time of the plot rendering

@term Termination message

@version Software version (e.g. 7.3)

@vpcci VPC confidence interval

@vpcdir VPC data directory

@vpclloq VPC lower limit of quantification

@vpcnsim Number of simulations for VPC

@vpcpi VPC prediction interval

@vpculoq VPC upper limit of quantification

@warnings Run warnings (e.g. boundary)

@x @y etc. Name of any ggplot2 variable used for mapping in an `aes()` type function

See Also

[xpose_save](#)

Examples

```
# Defined when creating a plot
dv_vs_ipred(xpdb_ex_pk,
            title = '@x vs. @y',
            subtitle = '@ofv, @nind subjects, @nobs obs.',
            caption = '@run, @descr')

# Any label can be modified later on
dv_vs_ipred(xpdb_ex_pk, aes(point_color = SEX,
                            line_color = SEX)) +
 labs(title = 'This runs is: @descr',
      color = 'Color scale for @run',
      x     = 'IPRED for @nind subjects',
      subtitle = NULL)
```

---

update_themes   *Create xpose theme*

Description

Create an xpose theme. This function will update the theme of an xpdb object. All plots generated with this xpdb will automatically use the defined xpose (xp_theme) and ggplot2 (gg_theme) themes.

Usage

```
update_themes(xpdb = NULL, gg_theme = NULL, xp_theme = NULL, quiet)
```

Arguments

xpdb	An xpose_data object generated with [xpose_data]().
gg_theme	A complete ggplot2 theme object (e.g. [theme_classic]()), a function returning a complete ggplot2 theme, or a change to the current gg_theme.
xp_theme	A complete xpose theme object (e.g. [theme_xp_default]()) or a list of modifications to the current xp_theme (e.g. list(point_color = 'red', line_linetype = 'dashed')).
quiet	Logical, if FALSE messages are printed to the console.

Examples

```
# Before default theme
dv_vs_ipred(xpdb_ex_pk, facets = 'SEX')

# Updating the gg_theme and xp_theme
xpdb_ex_pk %>%
  update_themes(gg_theme = theme(legend.position = 'top'),
                xp_theme = list(point_color = 'blue',
                                line_color  = 'blue')) %>%
  dv_vs_ipred(facets = 'SEX')
```

---

vpc   *Visual predictive checks*

Description

Generate visual predictive checks (VPC)

Usage

```
vpc(
  xpdb,
  vpc_type = NULL,
  mapping = NULL,
  smooth = TRUE,
  type = "alpr",
  title = "Visual predictive checks | @run",
  subtitle = "Number of simulations: @vpcnsim, confidence interval: @vpcci%",
  caption = "@vpcdir",
  tag = NULL,
  log = NULL,
  guide = TRUE,
  gg_theme,
  xp_theme,
  facets,
  quiet,
  area_fill = c("steelblue3", "grey60", "steelblue3"),
  line_linetype = c("93", "solid", "93"),
  ...
)
```

Arguments

xpdb	An xpose database object.
vpc_type	Only used when multiple vpc data are present in the same xpdb. The type of vpc to be created. Can be one of can be one of: 'continuous', 'categorical', 'censored' or 'time-to-event'.
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. point_color).
smooth	Should the bins be smoothed (connect bin midpoints, default) or shown as rectangular boxes.
type	String setting the type of plot to be used. Can be points 'p', line 'l', area 'a', rug 'r' and text 't' or any combination of the five.
title	Plot title. Use NULL to remove.
subtitle	Plot subtitle. Use NULL to remove.
caption	Page caption. Use NULL to remove.
tag	Plot identification tag. Use NULL to remove.
log	String assigning logarithmic scale to axes, can be either '', 'x', y' or 'xy'.
guide	Enable guide display in vpc continuous (e.g. lloq and uloq lines).
gg_theme	A complete ggplot2 theme object (e.g. [theme_classic](#)), a function returning a complete ggplot2 theme, or a change to the current gg_theme.
xp_theme	A complete xpose theme object (e.g. [theme_xp_default](#)) or a list of modifications to the current xp_theme (e.g. list(point_color = 'red', line_linetype = 'dashed')).

facets	Either a character string to use [facet_wrap](#) or a formula to use [facet_grid](#).
quiet	Logical, if `FALSE` messages are printed to the console.
area_fill	Shaded areas filling color, should be a vector of 3 values (i.e. low, med, high).
line_linetype	Lines linetype, should be a vector of 3 values (i.e. low, med, high).
...	any additional aesthetics.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is layer_option where layer is one of the names defined in the list below and option is any option supported by this layer e.g. point_color = 'blue', area_fill = 'green', etc.

- point: options to geom_point
- line: options to geom_line
- area: options to geom_ribbon (smooth = TRUE) or geom_rect (smooth = FALSE)
- rug: options to geom_rug
- text: options to geom_text
- guide: options to geom_hline
- xscale: options to scale_x_continuous or scale_x_log10
- yscale: options to scale_y_continuous or scale_y_log10

Faceting

Every xpose plot function has built-in faceting functionalities. Faceting arguments are passed to the functions [facet_wrap_paginate](#) when the facets argument is a character string (e.g. facets = c('SEX', 'MED1')) or [facet_grid_paginate](#) when facets is a formula (e.g. facets = SEX~MED1). All xpose plot functions accept all the arguments for the [facet_wrap_paginate](#) and [facet_grid_paginate](#) functions e.g. dv_vs_ipred(xpdb_ex_pk, facets = SEX~MED1, ncol = 3, nrow = 3, page = 1, margins = TRUE, labeller = 'label_both').

Faceting options can either be defined in plot functions (e.g. dv_vs_ipred(xpdb_ex_pk, facets = 'SEX')) or assigned globally to an xpdb object via the xp_theme (e.g. xpdb <- update_themes(xpdb_ex_pk, xp_theme = list(facets = 'SEX'))). In the latter example all plots generate from this xpdb will automatically be stratified by 'SEX'.

By default, some plot functions use a custom stratifying variable named 'variable', e.g. eta_distrib(). When using the facets argument, 'variable' needs to be added manually e.g. facets = c('SEX', 'variable') or facets = c('SEX', 'variable'), but is optional, when using the facets argument in xp_theme variable is automatically added whenever needed.

Template titles

Template titles can be used to create highly informative diagnostics plots. They can be applied to any plot title, subtitle, caption and tag. Template titles are defined via a single string containing key variables staring with a '@' (e.g. '@ofv') which will be replaced by their actual value when rendering the plot. For example ''@run, @nobs observations in @nind subjects'' would become ''run001, 1022 observations in 74 subjects''. The available key variables are listed under [template_titles](#).

See Also

vpc_data

Examples

```
xpdb_ex_pk %>%
 vpc_data(opt = vpc_opt(n_bins = 7)) %>%
 vpc()
```

---

vpc_data   *Visual predictive checks data*

Description

Generate visual predictive checks (VPC) data

Usage

```
vpc_data(
  xpdb,
  opt,
  stratify,
  vpc_type = c("continuous", "categorical", "censored", "time-to-event"),
  psn_folder = NULL,
  psn_bins = FALSE,
  obs_problem = NULL,
  sim_problem = NULL,
  quiet
)
```

Arguments

xpdb	An xpose database object.
opt	A list of options regarding binning, pi and ci computation. For more information see [vpc_opt](vpc_opt).
stratify	Either a character string or a formula to stratify the data. For 'categorical' vpcs the stratification fixed to the different categories.
vpc_type	A string specifying the type of VPC to be created, can be one of: 'continuous', 'categorical', 'censored' or 'time-to-event'.
psn_folder	Specify a PsN-generated VPC-folder.
psn_bins	Only used with argument psn_folder. If TRUE bins will be inputed from the PsN vpc_bins.txt file. If FALSE (default) bins will be re-calculated in R. Note that when psn_bins = TRUE only the first bin array will be used and applied to all panels as it is not currently possible to define per panel binning in xpose. In addition when psn_bins = TRUE is used along with vpc(smooth = FALSE) the observations lines may not be centered in the bins. Check the output carefully.

obs_problem	Alternative to the option 'psn_folder'. The $problem number to be used for observations. By default returns the last estimation problem.
sim_problem	Alternative to the option 'psn_folder'. The $problem number to be used for simulations. By default returns the last simulation problem.
quiet	Logical, if FALSE messages are printed to the console.

See Also

[vpc](#) [vpc_opt](#)

Examples

```
## Not run:
xpdb_ex_pk %>%
 vpc_data() %>%
 vpc()

## End(Not run)
```

---

vpc_opt   *Generate a list of options for VPC data generation*

Description

Provide a list of options to vpc_data function.

Usage

```
vpc_opt(
  bins = "jenks",
  n_bins = "auto",
  bin_mid = "mean",
  pred_corr = FALSE,
  pred_corr_lower_bnd = 0,
  pi = c(0.025, 0.975),
  ci = c(0.025, 0.975),
  lloq = NULL,
  uloq = NULL,
  rtte = FALSE,
  rtte_calc_diff = TRUE,
  events = NULL,
  kmmc = NULL,
  reverse_prob = FALSE,
  as_percentage = TRUE
)
```

Arguments

bins	Binning method, can be one of 'density', 'time', 'data', 'none', or one of the approaches available in classInterval() such as 'jenks' (default), 'pretty', or a numeric vector specifying the bin separators.
n_bins	When using the 'auto' binning method, what number of bins to aim for.
bin_mid	Specify how to is the mid bin value calculated, can be either 'mean' for the mean of all timepoints (default) or 'middle' to use the average of the bin boundaries.
pred_corr	Option reserved to continuous VPC. Logical, should a prediction correction (pcVPC) of the data be used.
pred_corr_lower_bnd	
	Option reserved to continuous VPC. Lower bound for the prediction-correction.
pi	Option reserved to continuous VPC. Simulated prediction interval to plot. Default is c(0.05, 0.95).
ci	Confidence interval around the percentiles to plot. Default is c(0.05, 0.95)
lloq	Number or NULL indicating lower limit of quantification. Default is NULL.
uloq	Number or NULL indicating upper limit of quantification. Default is NULL.
rtte	Option reserved to time-to-event VPC. Is the data repeated time-to-event (RTTE) TRUE or single time-to-event (TTE) FALSE.
rtte_calc_diff	Option reserved to time-to-event VPC. Should the time be recalculated? When simulating in NONMEM, you will probably need to set this to TRUE to recalculate the TIME to the relative time between events (unless you output the time difference between events and specify that as independent variable in the index.
events	Option reserved to time-to-event VPC. Numeric vector describing which events to show a VPC for when repeated TTE data, e.g. c(1:4). Default is NULL, which shows all events.
kmmc	Option reserved to time-to-event VPC. Either NULL for regular TTE VPC (default), or a variable name for a KMMC plot (e.g. 'WT').
reverse_prob	Option reserved to time-to-event VPC. Should the probability be reversed (i.e. plot 1-probability).
as_percentage	Should the Y-scale be in percent (0-100) TRUE (default), or standard (0-1) FALSE.

See Also

[vpc](#) [vpc_data](#)

Examples

```
vpc_opt()
```

---

xpdb_ex_pk　　　　　　　xpose_data *examples*

Description

Moxonidine xpose_data example.

Format

An xpose_data object

Source

[Moxonidine model]: Assumption Testing in Population Pharmacokinetic Models: Illustrated with an Analysis of Moxonidine Data from Congestive Heart Failure Patients. Journal of Pharmacokinetics and Biopharmaceutics. 26(2):207–246 (1998).

Examples

```
print(xpdb_ex_pk)
```

---

xplot_distrib　　　　　　*Default xpose distribution plot function*

Description

Manually generate distribution plots from an xpdb object.

Usage

```
xplot_distrib(
  xpdb,
  mapping = NULL,
  type = "hr",
  guide = FALSE,
  xscale = "continuous",
  yscale = "continuous",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  tag = NULL,
  plot_name = "density_plot",
  gg_theme,
  xp_theme,
  opt,
```

```
    quiet,
    ...
)
```

Arguments

xpdb	An xpose_data object generated with xpose_data.
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. point_color).
type	String setting the type of plot to be used. Can be histogram 'h', density 'd', rug 'r' or any combination of the three.
guide	Should the guide (e.g. reference distribution) be displayed.
xscale	Scale type for x axis (e.g. 'continuous', 'discrete', 'log10').
yscale	Scale type for y axis (e.g. 'continuous', 'discrete', 'log10').
title	Plot title. Use NULL to remove.
subtitle	Plot subtitle. Use NULL to remove.
caption	Page caption. Use NULL to remove.
tag	Plot identification tag. Use NULL to remove.
plot_name	Name to be used by xpose_save() when saving the plot.
gg_theme	A complete ggplot2 theme object (e.g. theme_classic), a function returning a complete ggplot2 theme, or a change to the current gg_theme.
xp_theme	A complete xpose theme object (e.g. theme_xp_default) or a list of modifications to the current xp_theme (e.g. list(point_color = 'red', line_linetype = 'dashed')).
opt	A list of options in order to create appropriate data input for ggplot2. For more information see data_opt.
quiet	Logical, if FALSE messages are printed to the console.
...	Any additional aesthetics.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is layer_option where layer is one of the names defined in the list below and option is any option supported by this layer e.g. histogram_fill = 'blue', rug_sides = 'b', etc.

- histogram: options to geom_histogram
- density: options to geom_density
- rug: options to geom_rug
- xscale: options to scale_x_continuous or scale_x_log10
- yscale: options to scale_y_continuous or scale_y_log10

Faceting

Every xpose plot function has built-in faceting functionalities. Faceting arguments are passed to the functions `facet_wrap_paginate` when the `facets` argument is a character string (e.g. `facets =` `c('SEX', 'MED1')`) or `facet_grid_paginate` when facets is a formula (e.g. `facets = SEX~MED1`). All xpose plot functions accept all the arguments for the `facet_wrap_paginate` and `facet_grid_paginate` functions e.g. `dv_vs_ipred(xpdb_ex_pk,facets = SEX~MED1, ncol = 3, nrow = 3, page = 1, margins` `= TRUE, labeller = 'label_both')`.

Faceting options can either be defined in plot functions (e.g. `dv_vs_ipred(xpdb_ex_pk,facets =` `'SEX')`) or assigned globally to an xpdb object via the `xp_theme` (e.g. `xpdb <- update_themes(xpdb_ex_pk,` `xp_theme = list(facets = 'SEX'))`). In the latter example all plots generate from this xpdb will automatically be stratified by 'SEX'.

By default, some plot functions use a custom stratifying variable named 'variable', e.g. `eta_distrib()`. When using the `facets` argument, 'variable' needs to be added manually e.g. `facets = c('SEX',` `'variable')` or `facets = c('SEX', 'variable')`, but is optional, when using the `facets` argument in `xp_theme` variable is automatically added whenever needed.

Template titles

Template titles can be used to create highly informative diagnostics plots. They can be applied to any plot title, subtitle, caption and tag. Template titles are defined via a single string containing key variables staring with a '@' (e.g. '@ofv') which will be replaced by their actual value when rendering the plot. For example ''@run, @nobs observations in @nind subjects'' would become ''run001, 1022 observations in 74 subjects''. The available key variables are listed under `template_titles`.

See Also

`xplot_scatter` `xplot_qq`

Examples

```
# A simple histogram
xplot_distrib(xpdb_ex_pk, aes(x = WT), type = 'hr')

# A simple density plot
xplot_distrib(xpdb_ex_pk, aes(x = CWRES), type = 'dr')
```

---

xplot_qq   *Default xpose QQ plot function*

Description

Manually generate QQ plots from an xpdb object.

Usage

```
xplot_qq(
  xpdb,
  mapping = NULL,
  type = "p",
  guide = FALSE,
  xscale = "continuous",
  yscale = "continuous",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  tag = NULL,
  plot_name = "qq_plot",
  gg_theme,
  xp_theme,
  opt,
  quiet,
  ...
)
```

Arguments

xpdb	An xpose_data object generated with xpose_data.
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. point_color).
type	String setting the type of plot to be used. Can only be points 'p'.
guide	Should the guide (e.g. reference line) be displayed.
xscale	Scale type for x axis (e.g. 'continuous', 'discrete', 'log10').
yscale	Scale type for y axis (e.g. 'continuous', 'discrete', 'log10').
title	Plot title. Use NULL to remove.
subtitle	Plot subtitle. Use NULL to remove.
caption	Page caption. Use NULL to remove.
tag	Plot identification tag. Use NULL to remove.
plot_name	Name to be used by xpose_save() when saving the plot.
gg_theme	A complete ggplot2 theme object (e.g. theme_classic), a function returning a complete ggplot2 theme, or a change to the current gg_theme.
xp_theme	A complete xpose theme object (e.g. theme_xp_default) or a list of modifications to the current xp_theme (e.g. list(point_color = 'red', line_linetype = 'dashed')).
opt	A list of options in order to create appropriate data input for ggplot2. For more information see data_opt.
quiet	Logical, if FALSE messages are printed to the console.
...	Any additional aesthetics.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is layer_option where layer is one of the names defined in the list below and option is any option supported by this layer e.g. point_color = 'blue', etc.

- point: options to `geom_point`
- guide: options to `geom_abline`
- xscale: options to `scale_x_continuous` or `scale_x_log10`
- yscale: options to `scale_y_continuous` or `scale_y_log10`

Faceting

Every xpose plot function has built-in faceting functionalities. Faceting arguments are passed to the functions [facet_wrap_paginate](#) when the `facets` argument is a character string (e.g. `facets = c('SEX', 'MED1')`) or [facet_grid_paginate](#) when facets is a formula (e.g. `facets = SEX~MED1`). All xpose plot functions accept all the arguments for the [facet_wrap_paginate](#) and [facet_grid_paginate](#) functions e.g. `dv_vs_ipred(xpdb_ex_pk, facets = SEX~MED1, ncol = 3, nrow = 3, page = 1, margins = TRUE, labeller = 'label_both')`.

Faceting options can either be defined in plot functions (e.g. `dv_vs_ipred(xpdb_ex_pk, facets = 'SEX')`) or assigned globally to an xpdb object via the `xp_theme` (e.g. `xpdb <- update_themes(xpdb_ex_pk, xp_theme = list(facets = 'SEX'))`). In the latter example all plots generate from this xpdb will automatically be stratified by 'SEX'.

By default, some plot functions use a custom stratifying variable named 'variable', e.g. `eta_distrib()`. When using the `facets` argument, 'variable' needs to be added manually e.g. `facets = c('SEX', 'variable')` or `facets = c('SEX', 'variable')`, but is optional, when using the `facets` argument in `xp_theme` variable is automatically added whenever needed.

Template titles

Template titles can be used to create highly informative diagnostics plots. They can be applied to any plot title, subtitle, caption and tag. Template titles are defined via a single string containing key variables staring with a '@' (e.g. '@ofv') which will be replaced by their actual value when rendering the plot. For example ''@run, @nobs observations in @nind subjects'' would become ''run001, 1022 observations in 74 subjects''. The available key variables are listed under [template_titles](#).

See Also

[xplot_scatter](#) [xplot_distrib](#)

Examples

```
xplot_qq(xpdb_ex_pk, aes(sample = CWRES), guide = TRUE)
```

---

xplot_scatter   *Default xpose scatter plot function*

Description

Manually generate scatter plots from an xpdb object.

Usage

```
xplot_scatter(
  xpdb,
  mapping = NULL,
  group = "ID",
  type = "pls",
  guide = FALSE,
  xscale = "continuous",
  yscale = "continuous",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  tag = NULL,
  plot_name = "scatter_plot",
  gg_theme,
  xp_theme,
  opt,
  quiet,
  ...
)
```

Arguments

xpdb	An xpose_data object generated with xpose_data.
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. point_color).
group	Grouping variable to be used for lines.
type	String setting the type of plot to be used. Can be line 'l', point 'p', smooth 's' and text 't' or any combination of the four.
guide	Should the guide (e.g. unity line) be displayed.
xscale	Scale type for x axis (e.g. 'continuous', 'discrete', 'log10').
yscale	Scale type for y axis (e.g. 'continuous', 'discrete', 'log10').
title	Plot title. Use NULL to remove.
subtitle	Plot subtitle. Use NULL to remove.
caption	Page caption. Use NULL to remove.
tag	Plot identification tag. Use NULL to remove.
plot_name	Name to be used by xpose_save() when saving the plot.

gg_theme   A complete ggplot2 theme object (e.g. `theme_classic`), a function returning a
                 complete ggplot2 theme, or a change to the current gg_theme.

xp_theme   A complete xpose theme object (e.g. `theme_xp_default`) or a list of modifica-
                 tions to the current xp_theme (e.g. `list(point_color = 'red', line_linetype
                 = 'dashed')`).

opt   A list of options in order to create appropriate data input for ggplot2. For more
                 information see `data_opt`.

quiet   Logical, if `FALSE` messages are printed to the console.

...   Any additional aesthetics.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is
layer_option where layer is one of the names defined in the list below and option is any option
supported by this layer e.g. point_color = 'blue', smooth_method = 'lm', etc.

- point: options to `geom_point`
- line: options to `geom_line`
- guide: options to `geom_abline`
- smooth: options to `geom_smooth`
- text: options to `geom_text`
- xscale: options to `scale_x_continuous` or `scale_x_log10`
- yscale: options to `scale_y_continuous` or `scale_y_log10`

Faceting

Every xpose plot function has built-in faceting functionalities. Faceting arguments are passed to the
functions `facet_wrap_paginate` when the facets argument is a character string (e.g. facets =
c('SEX', 'MED1')) or `facet_grid_paginate` when facets is a formula (e.g. facets = SEX~MED1).
All xpose plot functions accept all the arguments for the `facet_wrap_paginate` and `facet_grid_paginate`
functions e.g. dv_vs_ipred(xpdb_ex_pk, facets = SEX~MED1, ncol = 3, nrow = 3, page = 1, margins
= TRUE, labeller = 'label_both').

Faceting options can either be defined in plot functions (e.g. dv_vs_ipred(xpdb_ex_pk, facets =
'SEX')) or assigned globally to an xpdb object via the xp_theme (e.g. xpdb <- update_themes(xpdb_ex_pk,
xp_theme = list(facets = 'SEX'))). In the latter example all plots generate from this xpdb will
automatically be stratified by 'SEX'.

By default, some plot functions use a custom stratifying variable named 'variable', e.g. eta_distrib().
When using the facets argument, 'variable' needs to be added manually e.g. facets = c('SEX',
'variable') or facets = c('SEX', 'variable'), but is optional, when using the facets argu-
ment in xp_theme variable is automatically added whenever needed.

Template titles

Template titles can be used to create highly informative diagnostics plots. They can be applied to
any plot title, subtitle, caption and tag. Template titles are defined via a single string containing key
variables staring with a '@' (e.g. '@ofv') which will be replaced by their actual value when render-
ing the plot. For example ''@run, @nobs observations in @nind subjects'' would become ''run001,
1022 observations in 74 subjects''. The available key variables are listed under `template_titles`.

See Also

[xplot_distrib](#) [xplot_qq](#)

Examples

```
xplot_scatter(xpdb_ex_pk, aes(x = IPRED, y = DV))
```

---

xpose_data *Import NONMEM output into R*

Description

Gather model outputs into a R database

Usage

```
xpose_data(
  runno = NULL,
  prefix = "run",
  ext = ".lst",
  file = NULL,
  dir = NULL,
  gg_theme = theme_readable,
  xp_theme = theme_xp_default(),
  simtab = NULL,
  manual_import = NULL,
  ignore = NULL,
  check_ext = TRUE,
  extra_files,
  quiet,
  ...
)
```

Arguments

runno	Run number to be used to generate model file name. Used in combination with `prefix` and `ext`.
prefix	Prefix to be used to generate model file name. Used in combination with `runno` and `ext`.
ext	Extension to be used to generate model file name. Should be one of '.lst' (default), '.out', '.res', '.mod' or '.ctl' for NONMEM.
file	Model file name (preferably a '.lst' file) containing the file extension. Alternative to `prefix`, `runno` and `ext` arguments.
dir	Location of the model files.

gg_theme	A complete ggplot2 theme object (e.g. theme_classic), or a function returning a complete ggplot2 theme.
xp_theme	A complete xpose theme object (e.g. theme_xp_default).
simtab	If TRUE only reads in simulation tables, if FALSE only reads estimation tables. Default NULL reads all tables. Option not compatible with manual_import.
manual_import	If NULL (default) the names of the output tables to import will be obtained from the model file. To manually import files as in previous versions of xpose, the check the function manual_nm_import.
ignore	Character vector be used to ignore the import/generation of: 'data', 'files', 'summary' or any combination of the three.
check_ext	Logical, if TRUE will provide an error message if the extension of the NONMEM input file is not one of '.lst', '.out', '.res', '.mod' or '.ctl' for NONMEM. If FALSE any file extension can be used.
extra_files	A vector of additional output file extensions to be imported. Default is '.ext', '.cov', '.cor', '.phi', '.grd' for NONMEM.
quiet	Logical, if FALSE messages are printed to the console.
...	Additional arguments to be passed to the read_nm_tables functions.

File path generation

The rules for model file names generation are as follow:

- with runno: the full path is generated as <dir>/<prefix><runno>.<ext> e.g. with dir = 'model/pk', prefix = 'run', runno = '001', ext = '.lst' the resulting path would be model/pk/run001.lst
- with file: the full path is generated as <dir>/<file> e.g. with dir = 'model/pk', file = 'run001.lst' the resulting path would also be model/pk/run001.lst. Note: in this case the file extension should be provided as part of the 'file' argument.

Table format requirement

When importing data, an ID column must be present in at least one table for each problem and for each 'firstonly' category. ID columns are required to properly combine/merge tables and removing NA records. If ID columns are missing xpose will return the following warning: Dropped `<tablenames>` due to missing required `ID` column.

Examples

```
## Not run:
# Using the `file` argument to point to the model file:
xpdb <- xpose_data(file = 'run001.lst', dir = 'models')

# Using the `runno` argument to point to the model file:
xpdb <- xpose_data(runno = '001', ext = '.lst', dir = 'models')

# Using the `extra_files` argument to import specific output files only:
xpdb <- xpose_data(file = 'run001.lst', dir = 'models', extra_files = c('.ext', '.phi'))
```

```
# Using `ignore` to disable import of tables and output files:
xpdb <- xpose_data(file = 'run001.lst', dir = 'models', ignore = c('data', 'files'))

# Using `simtab` to disable import of simulation tables
xpdb <- xpose_data(file = 'run001.lst', dir = 'models', simtab = FALSE)


## End(Not run)
```

---

xpose_save   *Save xpose plot*

Description

Built as a wrapper around [ggsave](#), this function facilitates the export of xpose plots.

Usage

```
xpose_save(
  plot = last_plot(),
  file = NULL,
  dir = NULL,
  device = NULL,
  scale = 1,
  width = 7,
  height = 6,
  units = c("in", "cm", "mm", "px"),
  dpi = 200,
  limitsize = TRUE,
  bg = NULL,
  ...
)
```

Arguments

plot	A xpose plot object.
file	A name with file extension (if device is NULL) to be given to the output file. Template variables such as @run (run number) and @plotfun (plot function) can be used to automatically name files e.g. file = '@run_@plotfun.pdf'.
dir	Directory under which the xpose plots will be saved. Template variables such as @dir can be used to generate template names.
device	Graphical device to use. Can be either be a device function (e.g. [png](#)), or one of 'eps', 'ps', 'tex' (pictex), 'pdf', 'jpeg', 'tiff', 'png', 'bmp', 'svg' or 'wmf' (windows only).
scale	Multiplicative scaling factor.

width, height, units

>   Plot size in in `units` ("in", "cm", "mm", or "px"). If not supplied, uses the size of current graphics device.

dpi   Plot resolution. Also accepts a string input: "retina" (320), "print" (300), or "screen" (72). Applies only to raster output types.

limitsize   When `TRUE` (the default), `xpose_save()` will not save images larger than 50x50 inches, to prevent the common error of specifying dimensions in pixels.

bg   Background color. If `NULL`, uses the `plot.background` fill value from the plot theme.

...   Other arguments passed on to the graphics device function, as specified by `device`.

Examples

```
## Not run:
xpdb_ex_pk %>%
 dv_vs_ipred() %>%
 xpose_save(file = file.path(tempdir(), "dv_vs_ipred_example.pdf"))

## End(Not run)
```

---

xp_themes   *A set of xpose themes*

Description

xpose themes are used to consistently apply a set of preference for the plot geoms (e.g. color scales, point size, etc.) whereas ggplot2 theme focus on the plot background, axes, titles etc.

- `theme_xp_default`: The default xp_theme in xpose
- `theme_xp_xpose4`: An xp_theme that makes xpose look like xpose4.

Usage

```
theme_xp_default()

theme_xp_xpose4()
```

Examples

```
# With the xp_theme theme_xp_default() (default)
dv_vs_ipred(xpdb_ex_pk, facets = 'SEX')

# With the xp_theme theme_xp_xpose4()
xpdb_ex_pk %>%
  update_themes(xp_theme = theme_xp_xpose4()) %>%
  dv_vs_ipred(facets = 'SEX')
```

Index