Model Diagnostics with xpose : : CHEAT SHEET

The **xpose** package facilitates the creation of model diagnostics from NONMEM output. Inspired by **xpose4**, this new version is actively being redesigned around the popular **tidyverse** packages **ggplot2**, **dplyr** and **readr**.

Getting started

INSTALLATION

- **From github** *(development version)*
  library(devtools)
  install_github('UUPharmacometrics/xpose')
- **From CRAN** *(coming soon)*
  install.packages('xpose')

GETTING HELP

- Comprehensive documentation and examples are available at: uupharmacometrics.github.io/xpose/
- Use **?<function_name>** in R to access functions' help (e.g. ?xpose_data).

PLOT TYPE

Plot **type** is specified via a single string, where values: **a** (area), **d** (density), **h** (histogram), **l** (line), **p** (point), **r** (rug), **s** (smooth) and **t** (text) can be combined depending on the plot function.

dv_vs_ipred(xpdb_ex_pk, **type** = **'pls'**)
eta_distrib(xpdb_ex_pk, **type** = **'hdr'**)

PLOT LAYERS

All **ggplot2** functions can be used to add or modify **xpose** plot layers, mapping, labels, scales, annotations, etc.

plot <– **dv_vs_ipred**(xpdb_ex_pk)
plot + **geom_hline**(yintercept = 1)

PIPES

All **xpose** functions can be used with pipes (%>%)

xpdb_ex_pk %>%
filter(OCC == 3) %>%
dv_vs_ipred()

Acknowledgements: the development of xpose was jointly funded by Pfizer and Pharmetheus

Plot functions

The **xpdb** (xpose database) is a structured object containing the NONMEM output tables, output files, the parsed model code, general options and plot themes.

BASIC GOF

Accepted plot types: l, p, s, t
Layer names: guide, line, point, smooth, text, xscale, yscale

dv_vs_ipred(xpdb, guide = TRUE)
dv_vs_pred(xpdb, guide = TRUE)

res_vs_idv(xpdb, res = 'CWRES', guide = TRUE)
res_vs_pred(xpdb, res = 'CWRES', guide = TRUE)

absval_res_vs_idv(xpdb, res = 'CWRES')
absval_res_vs_pred(xpdb, res = 'CWRES')

dv_vs_idv(xpdb, group = 'ID')
ipred_vs_idv(xpdb, group = 'ID')
pred_vs_idv(xpdb, group = 'ID')

dv_preds_vs_idv(xpdb, group = 'ID')
display of DV, IPRED and PRED side by side

INDIVIDUAL PLOTS

Accepted plot types: l, p, s, t
Layer names: line, point, smooth, text, xscale, yscale

ind_plots(xpdb)

COMPARTMENT KINETICS

Accepted plot types: l, p, s, t
Layer names: line, point, smooth, text, xscale, yscale

amt_vs_idv(xpdb, group = ID)
uses A1, A2, …, columns by default

VISUAL PREDICTIVE CHECKS

Accepted plot types: a, l, p, r, t
Layer names: area, line, point, rug, text, xscale, yscale

compute vpc data in R — advanced options (e.g. bins, lloq, pi, ci, etc.)

vpc_data(xpdb, opt = **vpc_opt**(…), vpc_type, stratify, psn_folder) %>%
vpc(smooth) — plot the vpc

DISTRIBUTIONS

Accepted plot types: d, h, r
Layer names: density, histogram, rug, xscale, yscale

prm_distrib(xpdb)
eta_distrib(xpdb)
cov_distrib(xpdb)
res_distrib(xpdb, res = 'CWRES')

QQ PLOTS

Accepted plot types: p
Layer names: guide, point

prm_qq(xpdb, guide = TRUE)
eta_qq(xpdb, guide = TRUE)
cov_qq(xpdb, guide = TRUE)
res_qq(xpdb, res = 'CWRES', guide =TRUE)

MINIMIZATION DIAGNOSTICS

Accepted plot types: l, p, s, t
Layer names: line, point, smooth, text, xscale, yscale

grd_vs_iteration(xpdb) *(.grd file required)*
prm_vs_iteration(xpdb) *(.ext file required)*

Customize plots

THEMES

The xpdb objects contain two types of themes :

- **gg_theme**: sets plot background and text properties

theme_readable() *light grey*
theme_bw2() *black and white*

- **xp_theme**: sets default aesthetics values (e.g. points color, lines width)

theme_xp_default() *ggplot2 default*
theme_xp_xpose4() *xpose4 blue*

New themes can be applied globally:
xpdb <– **update_themes**(xpdb, gg_theme, xp_theme)
Or locally in each plot function:
dv_vs_ipred(xpdb, gg_theme, xp_theme)

PLOTS AESTHETICS

Arguments for aesthetics are composed of the target layer name (e.g. point, line) and the name of the argument in the format **<layer>_<argument>** (e.g. point_color = 'red', line_linetype = 'dashed', smooth_method = 'lm').

FACETING & PAGINATION

All **xpose** plot functions accept arguments for **facet_wrap** and **facet_grid** (e.g. facets, ncol, nrow, scales)

Use **facet_wrap** facets = **<string>**
Use **facet_grid** facets = **<formula>**

Pagination is enabled when the arguments **ncol** and **nrow** are both set. The argument **page** can then be used to output specific pages or a range of pages:
xpdb_ex_pk %>%
dv_vs_ipred(facets=MED1~OCC, **ncol=2**, **nrow=1**, **page=1:2**)

Data

IMPORT

Data import in **xpose** is structured as follows:
1. Read NONMEM **control stream** (.mod/.lst) to list table filenames for each $PROBLEM
2. Import and index tables (compatible with FIRSTONLY option, .csv and compressed (.zip) files)
3. Import NONMEM output files (.ext, .phi, .cov, etc.)
4. Summarize control stream

Runs can automatically be imported either by using the **file** or the **prefix**, **runno** and **ext** arguments.
xpdb <– **xpose_data**(dir, file, prefix, runno, ext)

EDIT

Data in the xpdb can be edited using **dplyr** functionalities
- **filter**(xpdb, …, .problem, .source)
  subset data based on logical condition(s)
- **mutate**(xpdb, …, .problem, .source)
  add, modify and remove columns
- **set_var_type**(xpdb, …, problem)
  assign or modify output tables' index

xpdb_ex_pk %>% — built-in xpdb example
mutate(TAD = TIME %% 24) %>% — e.g. generate and plot time after dose
dv_vs_idv(aes(x = TAD))

ACCESS

Access and extract data from an xpdb.
- **get_code**(xpdb, problem) *(parsed control stream)*
- **get_prm**(xpdb, problem) *(table of parameter estimates)*
- **get_file**(xpdb, ext, problem) *(parsed output files)*
- **get_data**(xpdb, problem) *(combined dataset)*
- **get_summary**(xpdb, problem) *(table of run summary)*

SUMMARY

- **print**(xpdb) or xpdb
  display xpdb structure
- **summary**(xpdb)
  display run summary
- **list_vars**(xpdb)
  display output tables column indexing

Template titles

Special **@<keywords>** can be used in plot labels. They are automatically replaced by their actual value in the run summary when plotting (e.g. title = 'ofv: @ofv' can give 'ofv: -1518.108'). Check **?template_titles** in R for a full list.

Save plots

By default, plots are saved in .pdf and the file named after the plot function used.
xpose_save(plot, file, dir, width, height)