# 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 CRAN install.packages('xpose')
- From github (development version) library(devtools) install\_github('UUPharmacometrics/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)</pre> 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. tables, output files, the parsed model code, general options and plot themes.

#### **BASIC GOF**

Accepted plot types: l, p, s, t Layer names: quide, 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

advanced options



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



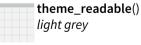
grd\_vs\_iteration(xpdb) (.grd file required) prm\_vs\_iteration(xpdb) (.ext file required)

## Customize plots

#### **THEMES**

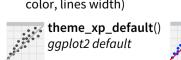
The xpdb objects contain two types of themes:

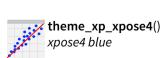
gg\_theme: sets plot background and text properties



theme\_bw2() black and white

• xp\_theme: sets default aesthetics values (e.g. points color, lines width)





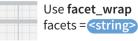
New themes can be applied globally: xpdb <- update\_themes(xpdb, gg\_theme, xp\_theme)</pre> Or locally in each plot function: dv\_vs\_ipred(xpdb, gg\_theme, xp\_theme)

#### **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\_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 SPROBLEM
- 2. Import and index tables (compatible with FIRSTONLY option, .csv and compressed (.zip) files)
- 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)</pre>

#### **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 %>% mutate(TAD = TIME %% 24) %>%  $dv_vs_idv(aes(x = TAD))$ 

built-in xpdb example

#### **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
- list\_vars(xpdb) display data variables
- **summary**(xpdb) display run summary
- prm\_table(xpdb) display parameter table

### 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

The file and dir arguments can contain template titles' keywords. Also handles plots with multiple pages. xpose\_save(plot, file = '@run\_@plotfun.pdf', dir)