PK/PD, Exposure-Response - Continuous

Your Name Here

09 July, 2020

Overview

This document contains exploratory plots for continuous PD data as well as the R code that generates these graphs. The plots presented here are based on simulated data (see: PKPD Datasets). Data specifications can be accessed on Datasets and Rmarkdown template to generate this page can be found on Rmarkdown-Template. You may also download the Multiple Ascending Dose PK/PD dataset for your reference (download dataset).

Setup

```
library(ggplot2)
library(dplyr)
library(tidyr)
library(xgxr)

#flag for labeling figures as draft
status = "DRAFT"

## ggplot settings
xgx_theme_set()

#directories for saving individual graphs
dirs = list(
  parent_dir= tempdir(),
  rscript_dir  = "./",
  rscript_name = "Example.R",
  results_dir  = "./",
  filename_prefix   = "",
  filename     = "Example.png")
```

Load Dataset

```
#load dataset
pkpd_data <- read.csv("https://raw.githubusercontent.com/Novartis/xgxr/master/data_create/raw/nonlinear

DOSE_CMT = 1
PK_CMT = 5
PD_CMT = 4
SS_PROFDAY = 0 # steady state prof day
PD_PROFDAYS <- c(0, 2, 4, 6)
TAU = 244 # time between doses, units should match units of TIME, e.g. 24 for QD, 12 for BID, 7*24 for

#ensure dataset has all the necessary columns
pkpd_data = pkpd_data %>%
```

```
  mutate(TIME = TIM2, NOMTIME = NT, EVID = 0, CENS = 0, DOSE = MGKG,     TRTACT = TRT, LIDV_NORM = LIDV/

#create a factor for the treatment variable for plotting
pkpd_data = pkpd_data %>%
  arrange(DOSE) %>%
  mutate(TRTACT_low2high = factor(TRTACT, levels = unique(TRTACT)),
         TRTACT_high2low = factor(TRTACT, levels = rev(unique(TRTACT))))

#create pk and pd datasets
pk_data <- pkpd_data %>%
  filter(CMT==PK_CMT)

pd_data <- pkpd_data %>%
  filter(CMT==PD_CMT)

#create wide pkpd dataset for plotting PK vs PD
pkpd_data_wide <- pd_data %>%
  select(ID, NOMTIME, PD = LIDV) %>%
  right_join(pk_data) %>%
  rename(CONC = LIDV)%>%
  filter(!is.na(PD))%>%
  filter(!is.na(CONC))

#perform NCA, for additional plots
NCA = pk_data %>%
  group_by(ID, DOSE) %>%
  filter(!is.na(LIDV)) %>%
  summarize(AUC_0 = ifelse(length(LIDV[NOMTIME > 0 & NOMTIME <= TAU]) > 1,
                           caTools::trapz(TIME[NOMTIME > 0 & NOMTIME <= TAU],
                                   LIDV[NOMTIME > 0 & NOMTIME <= TAU]),
                           NA),
            Cmax_0     = ifelse(length(LIDV[NOMTIME > 0 & NOMTIME <= TAU]) > 1,
                           max(LIDV[NOMTIME > 0 & NOMTIME <= TAU]),
                           NA),
            AUC_tau = ifelse(length(LIDV[NOMTIME > (SS_PROFDAY-1)*24 &
                                         NOMTIME <= ((SS_PROFDAY-1)*24 + TAU)]) > 1,
                             caTools::trapz(TIME[NOMTIME > (SS_PROFDAY-1)*24 &
                                                 NOMTIME <= ((SS_PROFDAY-1)*24 + TAU)],
                                    LIDV[NOMTIME > (SS_PROFDAY-1)*24 &
                                           NOMTIME <= ((SS_PROFDAY-1)*24 + TAU)]),
                             NA),
            Cmax_tau     = ifelse(length(LIDV[NOMTIME > (SS_PROFDAY-1)*24 &
                                              NOMTIME <= ((SS_PROFDAY-1)*24 + TAU)]) > 1,
                             max(LIDV[NOMTIME > (SS_PROFDAY-1)*24 &
                                        NOMTIME <= ((SS_PROFDAY-1)*24 + TAU)]),
                             NA),
            SEX      = SEX[1], #this part just keeps the SEX and WEIGHTB covariates
            WEIGHTB  = WEIGHTB[1]) %>%
  gather(PARAM, VALUE,-c(ID, DOSE, SEX, WEIGHTB)) %>%
  ungroup() %>%
  mutate(VALUE_NORM = VALUE/DOSE,
         PROFDAY = ifelse(PARAM %in% c("AUC_0", "Cmax_0"), 1, SS_PROFDAY))
```

```r
#add response data at day 1 and steady state to NCA for additional plots
NCA <- pd_data %>% subset(PROFDAY %in% c(1, SS_PROFDAY),) %>%
  select(ID, PROFDAY, DAY_label, PD = LIDV, TRTACT_low2high, TRTACT_high2low) %>%
  merge(NCA, by = c("ID","PROFDAY"))

#units and labels
time_units_dataset = "hours"
time_units_plot    = "days"
trtact_label       = "Dose"
dose_units         = unique((pkpd_data %>% filter(CMT == DOSE_CMT) )$LIDV_UNIT) %>% as.character()
dose_label         = paste0("Dose (", dose_units, ")")
conc_units         = unique(pk_data$LIDV_UNIT) %>% as.character()
conc_label         = paste0("Concentration (", conc_units, ")")
concnorm_label     = paste0("Normalized Concentration (", conc_units, ")/", dose_units)
AUC_units          = paste0("h.", conc_units)
pd_units           = unique(pd_data$LIDV_UNIT) %>% as.character()
pd_label           = paste0("Continuous PD Marker (", pd_units, ")")
```

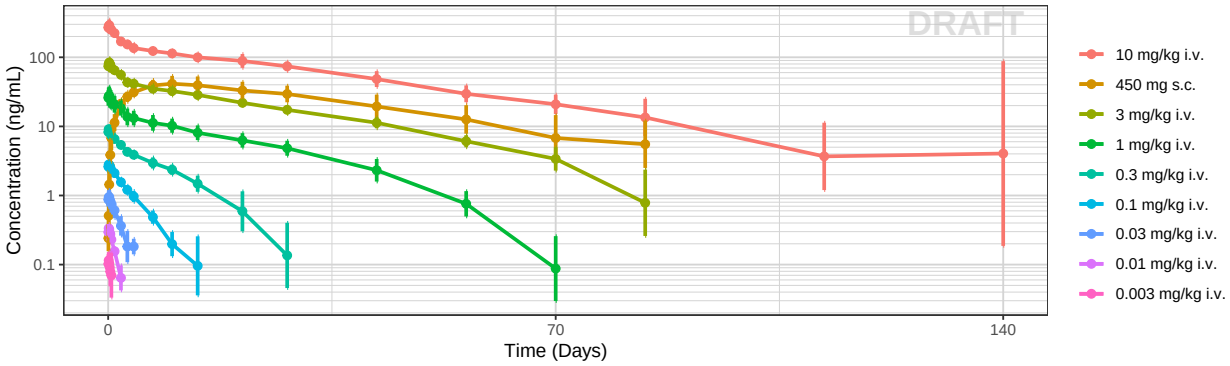
Provide an overview of the data

Summarize the data in a way that is easy to visualize the general trend of PD over time and between doses. Using summary statistics can be helpful, e.g. Mean +/- SE, or median, 5th & 95th percentiles. Consider either coloring by dose or faceting by dose. Depending on the amount of data one graph may be better than the other.

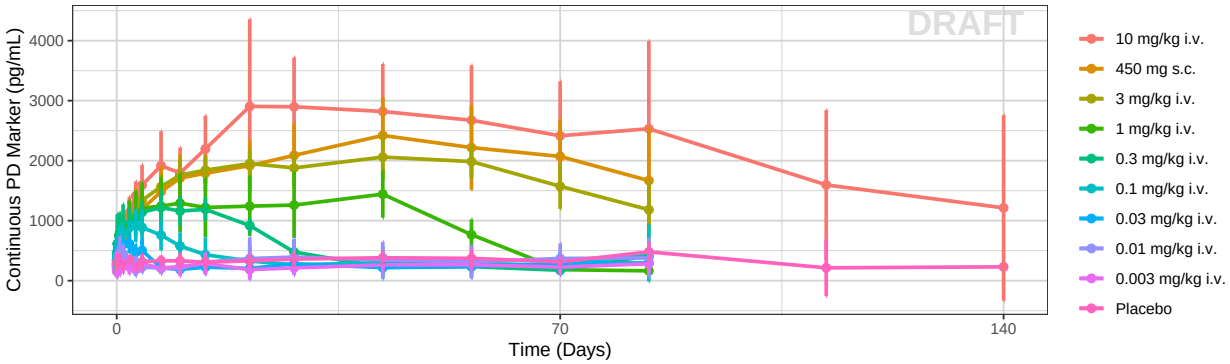
PK and PD marker over time, colored by Dose, mean (95% CI) percentiles by nominal time

Observe the overall shape of the average profiles. Does the effect appear to increase and decrease quickly on a short time scale, or does is occur over a longer time scale? Do the PK and PD profiles appear to be on the same time scale, or does the PD seem delayed compared to the PK? Is there clear separation between the profiles for different doses? Does the effect appear to increase with increasing dose? Do you detect a saturation of the effect?

```r
#PK data
gg <- ggplot(data = pk_data,
             aes(x = NOMTIME,y = LIDV, color = TRTACT_high2low, fill = TRTACT_high2low))
gg <- gg + xgx_stat_ci(conf_level = .95)
gg <- gg + xgx_annotate_status(status)
gg <- gg + xgx_scale_x_time_units(units_dataset = time_units_dataset,
                                  units_plot    = time_units_plot)
gg <- gg + guides(color = guide_legend(""), fill = guide_legend(""))
gg <- gg + xgx_scale_y_log10()
gg <- gg + labs(y = conc_label)
print(gg)
```

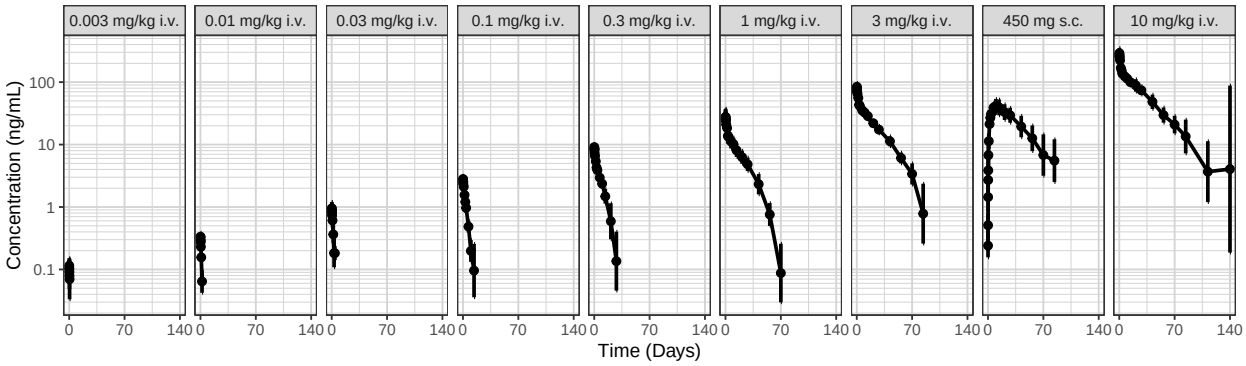



```
#PD data
gg %+% (data = pd_data) + scale_y_continuous() + labs(y = pd_label)
```

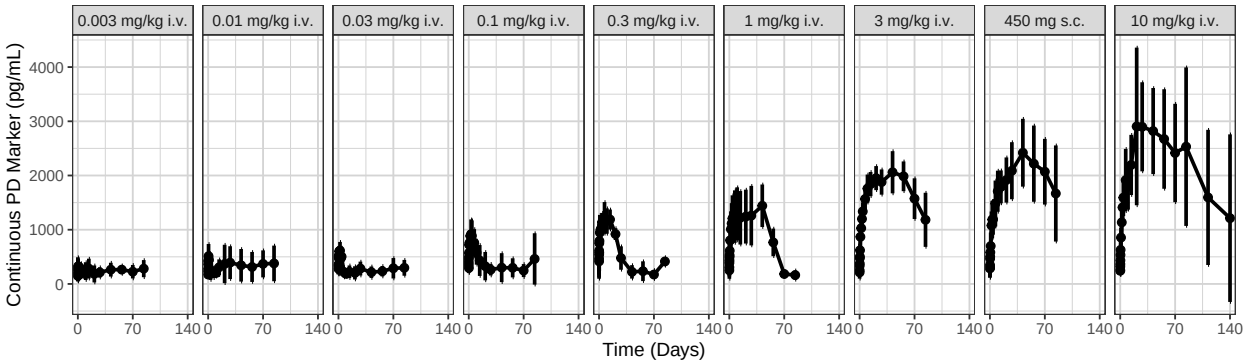


PK and PD marker over time, faceted by Dose, mean (95% CI) by nominal time

```
#PK data
gg <- ggplot(data = pk_data,
             aes(x = NOMTIME,y = LIDV))
gg <- gg + xgx_stat_ci(conf_level = .95)
gg <- gg + xgx_scale_x_time_units(units_dataset = time_units_dataset,
                                  units_plot    = time_units_plot)
gg <- gg + guides(color = guide_legend(""),fill = guide_legend(""))
gg <- gg + facet_grid(~TRTACT_low2high)
gg <- gg + xgx_scale_y_log10()
gg <- gg + labs(y = conc_label)
print(gg)
```




```
#PD data
gg %+% (data = pd_data %>% subset(DOSE>0)) + scale_y_continuous() + labs(y = pd_label)
```

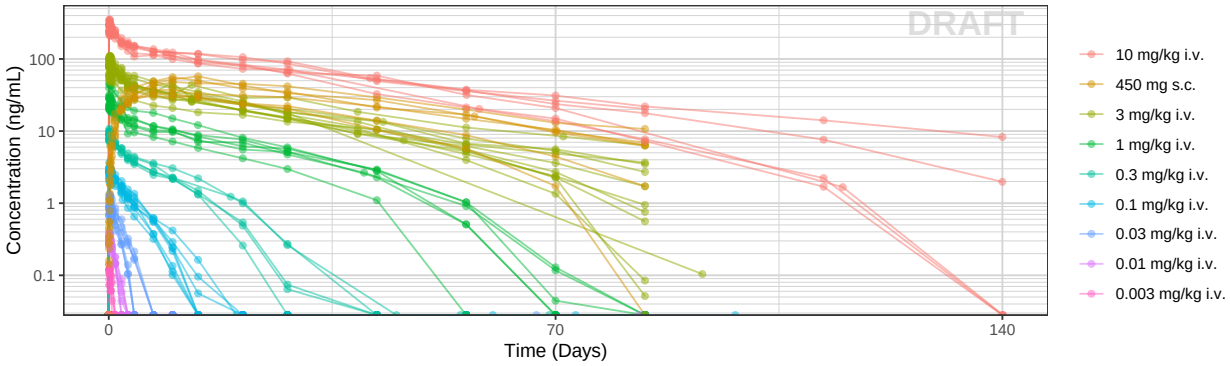


Explore variability

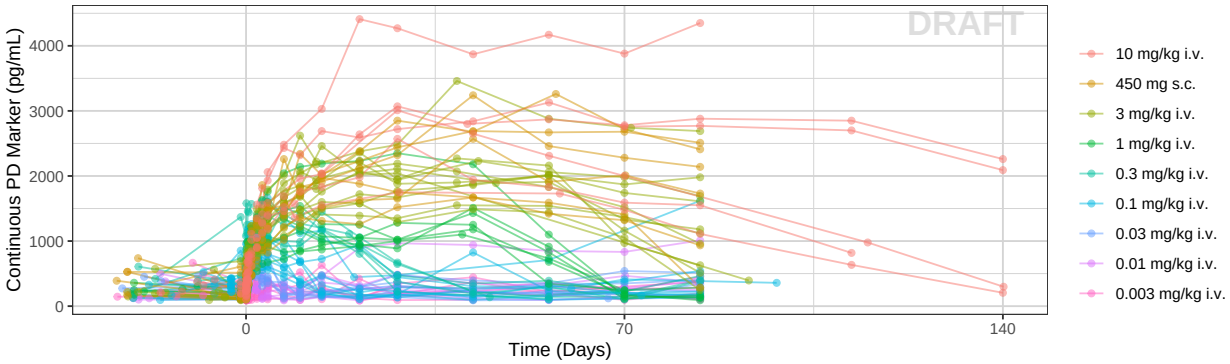
Use spaghetti plots to visualize the extent of variability between individuals. The wider the spread of the profiles, the higher the between subject variability. Distinguish different doses by color, or separate into different panels. If coloring by dose, do the individuals in the different dose groups overlap across doses? Dose there seem to be more variability at higher or lower concentrations?

PK and PD marker over time, colored by Dose, dots & lines grouped by individuals

```
#PK data
gg <- ggplot(data = pk_data,
             aes(x = TIME, y = LIDV,group = ID, color = factor(TRTACT_high2low)))
gg <- gg + xgx_annotate_status(status)
gg <- gg + geom_line(alpha = 0.5)
gg <- gg + geom_point(alpha = 0.5)
gg <- gg + guides(color = guide_legend(""), fill = guide_legend(""))
gg <- gg + xgx_scale_x_time_units(units_dataset = time_units_dataset,
                                  units_plot    = time_units_plot)
gg <- gg + geom_point(data = pk_data %>% subset(CENS == 1,), color = "red", shape = 8,alpha = 0.5)
gg <- gg + xgx_scale_y_log10()
gg <- gg + labs(y = conc_label)
print(gg)
```

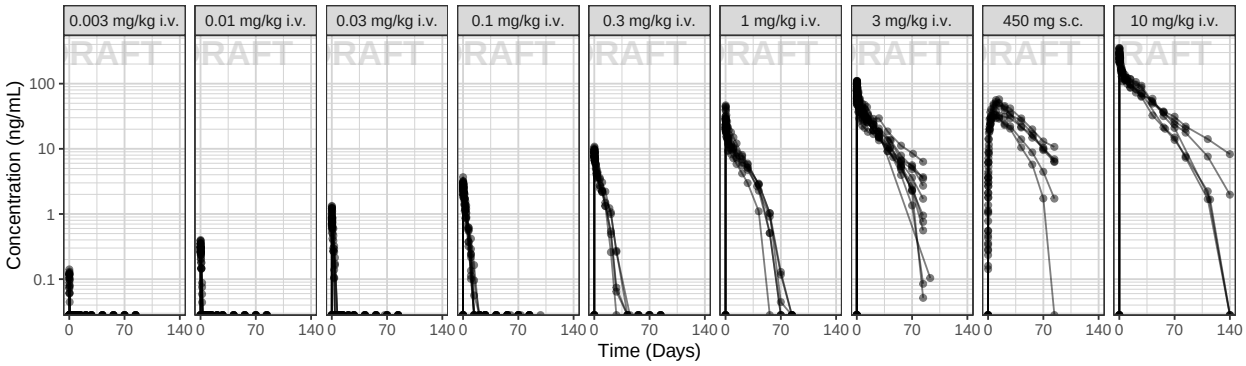



```
#PD data
gg %+% (data = pd_data %>% subset(DOSE>0)) + scale_y_continuous() + labs(y = pd_label)
```

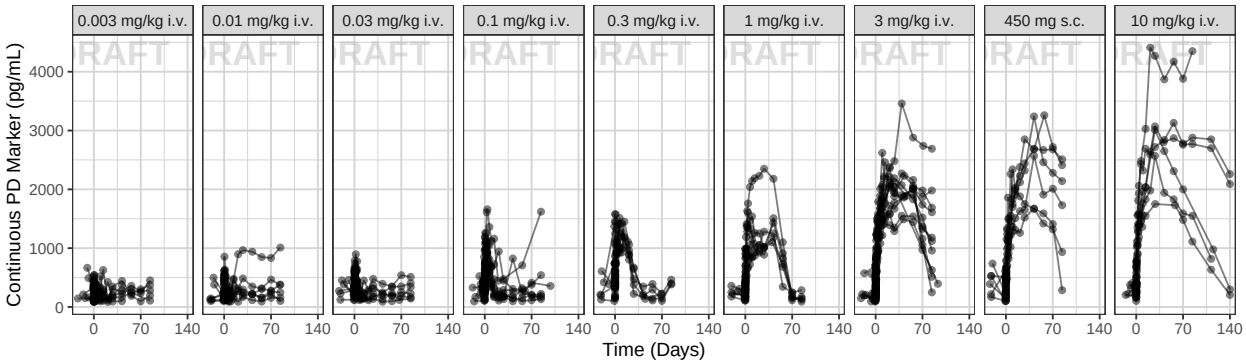


PK and PD marker over time, faceted by Dose, dots & lines grouped by individuals

```
#PK data
gg <- ggplot(data = pk_data, aes(x = TIME, y = LIDV, group = ID))
gg <- gg + xgx_annotate_status(status)
gg <- gg + geom_line(alpha = 0.5)
gg <- gg + geom_point(alpha = 0.5)
gg <- gg + guides(color = guide_legend(""), fill = guide_legend(""))
gg <- gg + xgx_scale_x_time_units(units_dataset = time_units_dataset,
                                  units_plot    = time_units_plot)
gg <- gg + facet_grid(~TRTACT_low2high)
gg <- gg + geom_point(data = pk_data %>% subset(CENS==1,), color="red", shape=8, alpha = 0.5)
gg <- gg + xgx_scale_y_log10()
gg <- gg + labs(y = conc_label)
print(gg)
```




```
#PD data
gg %+% (data = pd_data %>% subset(DOSE>0)) + scale_y_continuous() + labs(y = pd_label)
```



Explore Exposure-Response Relationship

Plot PD marker against concentration. Do you see any relationship? Does response increase (decrease) with increasing dose? Are you able to detect a plateau or emax (emin) on the effect?

Warning: Even if you don't see an Emax, that doesn't mean there isn't one. Be very careful about using linear models for Dose-Response or Exposure-Response relationships. Extrapolation outside of the observed dose range could indicate a higher dose is always better (even if it isn't).

```
gg <- ggplot(data = pkpd_data_wide %>% subset(PROFDAY == SS_PROFDAY,), aes(x = CONC,y = PD))
gg <- gg + xgx_annotate_status(status)
gg <- gg + geom_point(aes(color = TRTACT_high2low))
gg <- gg + geom_point(data = pkpd_data_wide %>% subset(CENS==1,), color="red", shape = 8)
gg <- gg + labs(x = conc_label , y = pd_label)
gg <- gg + xgx_scale_x_log10()
gg <- gg + guides(color = guide_legend(""))
print(gg)
```

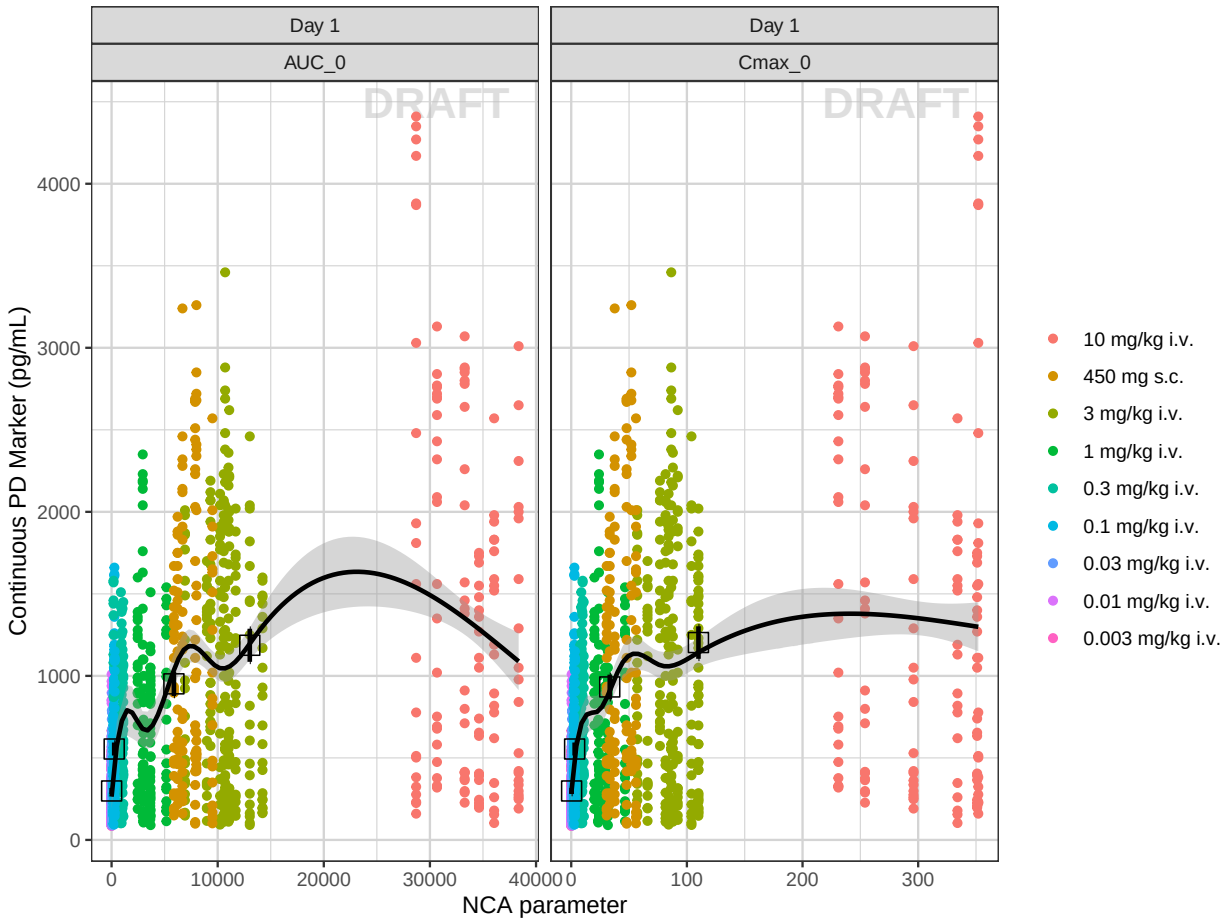
```
## Error in seq.default(from = best$lmin, to = best$lmax, by = best$lstep): 'from' must be of length 1
```

```r
data_to_plot <- pkpd_data_wide %>% subset(PROFDAY %in% PD_PROFDAYS,)

gg <- ggplot(data = data_to_plot,
             aes(x = CONC, y = PD, color = TRTACT_high2low))
gg <- gg + xgx_annotate_status(status)
gg <- gg + geom_point()
gg <- gg + geom_point(data = data_to_plot %>% subset(CENS==1,), color="red", shape = 8)
gg <- gg + guides(color = guide_legend(""), fill = guide_legend(""))
gg <- gg + labs(x = conc_label , y = pd_label)
gg <- gg + xgx_scale_x_log10()
gg + facet_grid(~DAY_label)
```

```
## Error: Faceting variables must have at least one value
```

Plotting AUC vs response instead of concentration vs response may make more sense in some situations. For example, when there is a large delay between PK and PD it would be diffcult to relate the time-varying concentration with the response. If rich sampling is only done at a particular point in the study, e.g. at steady state, then the AUC calculated on the rich profile could be used as the exposure variable for a number of PD visits. If PK samples are scarce, average Cmin could also be used as the exposure metric.

```
NCA_plot = NCA %>%
  group_by(PARAM) %>%
  mutate(VALUE_QUART = cut(VALUE, quantile(VALUE, na.rm=TRUE), na.rm=TRUE, include.lowest = TRUE)) %>%
  group_by(PARAM, VALUE_QUART) %>%
  mutate(VALUE_MIDPOINT = median(VALUE))

gg <- ggplot(data = NCA_plot, aes(x = VALUE, y = PD))
gg <- gg + xgx_annotate_status(status)
gg <- gg + geom_point(aes(color = TRTACT_high2low)) + geom_smooth(color = "black")
gg <- gg + xgx_stat_ci(aes(x = VALUE_MIDPOINT, y = PD), geom = "errorbar")
gg <- gg + xgx_stat_ci(aes(x = VALUE_MIDPOINT, y = PD), geom = "point", shape = 0, size = 4)
gg <- gg + guides(color = guide_legend(""), fill = guide_legend(""))
gg <- gg + labs(color = trtact_label, x = "NCA parameter", y = pd_label)
gg <- gg + facet_wrap(~DAY_label + PARAM, scales = "free_x")
print(gg)
```

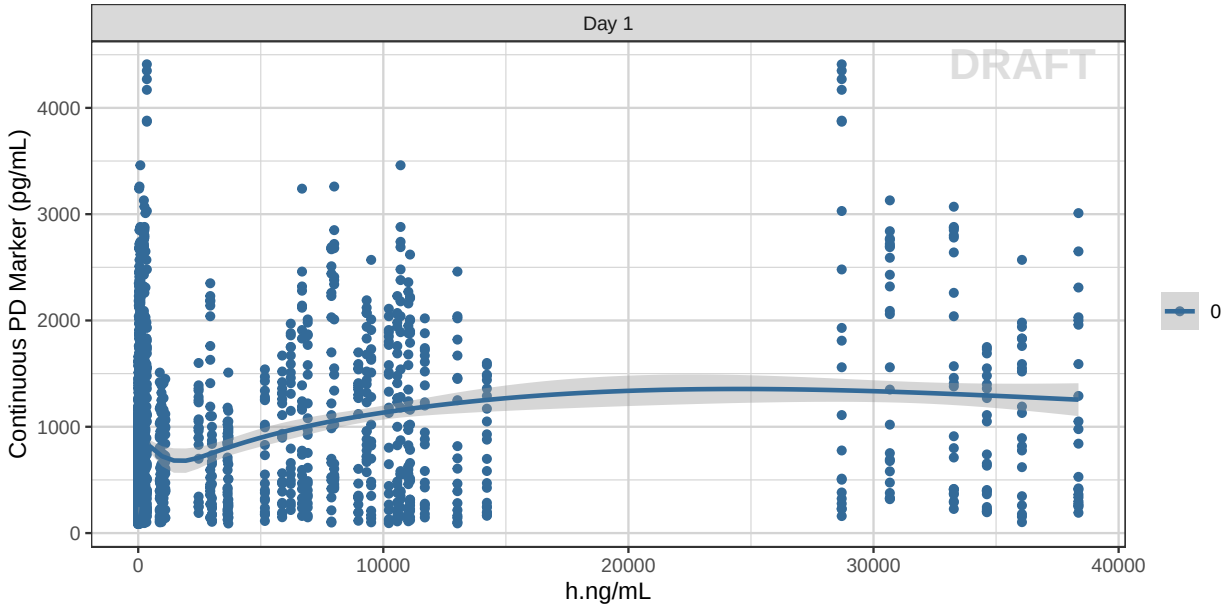


Explore covariate effects on Exposure-Response Relationship

Stratify exposure-response plots by covariates of interest to explore whether any key covariates impact response independent of exposure.

```
gg <- ggplot(data = NCA_plot, aes(x = VALUE, y = PD))
gg <- gg + xgx_annotate_status(status)
```

```
gg <- gg + geom_point(aes(color = SEX)) + geom_smooth(aes(color = SEX))
gg <- gg + guides(color = guide_legend(""), fill = guide_legend(""))
gg <- gg + labs(x = AUC_units , y = pd_label)
gg + facet_grid(.~DAY_label)
```



Individual response vs exposure hysteresis plots

Using geom_path you can create hysteresis plots of response vs exposure. Including details like arrows or colors can be helpful to indicate the direction of time.

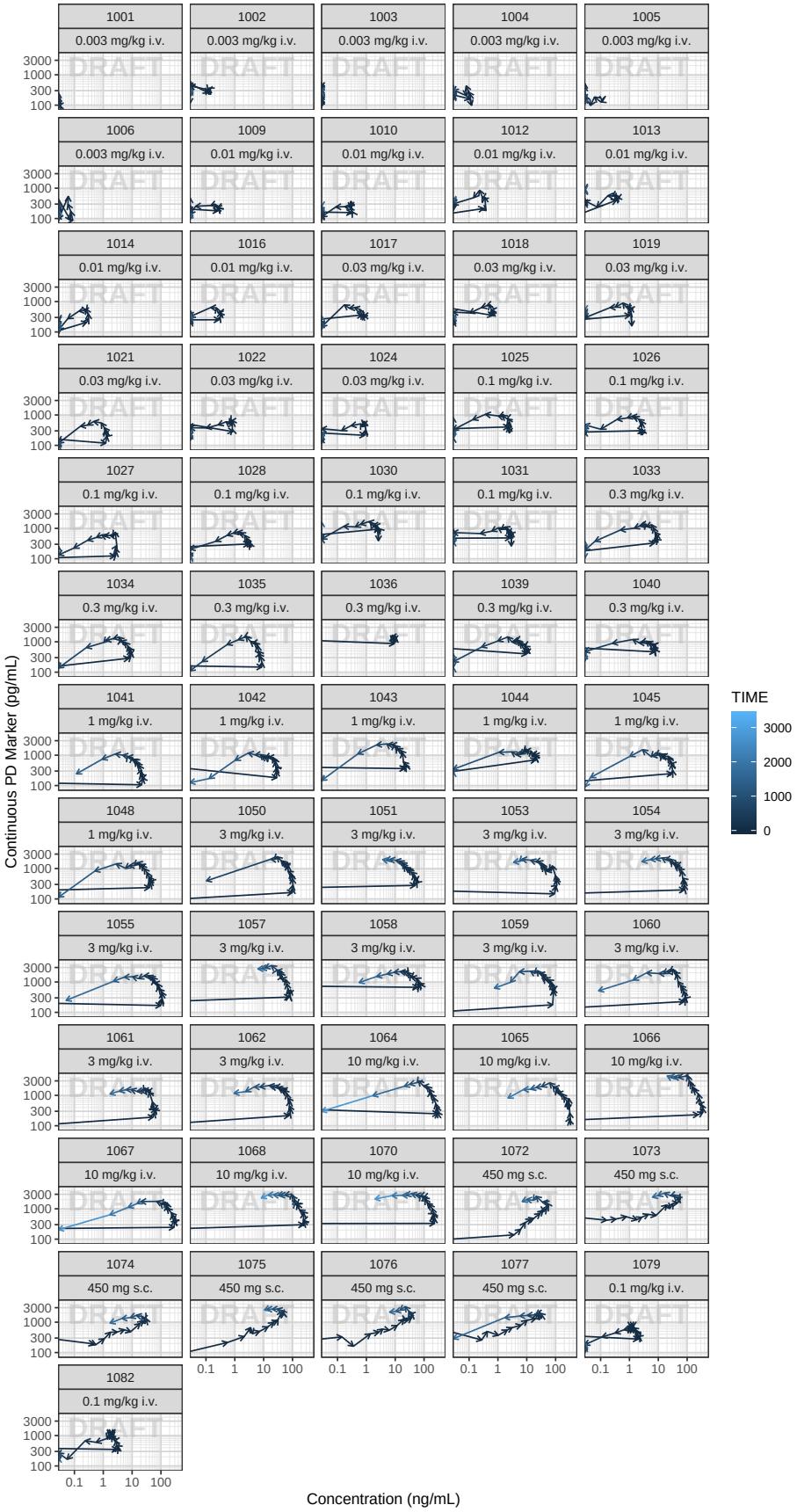
If most of the arrows point up and to the right or down and to the left, this indicates a positive relationship between exposure and response (i.e. increasing exposure -> increasing response). Arrows pointing down and to the right or up and to the left indicate a negative relationship.

In a hysteresis plot, you want to determine whether the path is curving, and if so in what direction. If you detect a curve in the hysteresis plot, this indicates there is a delay between the exposure and the response. Normally, a clockwise turn indicates that increasing exposure is associated with (a delayed) increasing response, while a counter clockwise turn indicates increasing concentration gives (a delayed) decreasing response.

In the plots below, most of the hysteresis paths follow a counter clockwise turn, with most arrows pointing up and to the right or down and to the left, indicating the effect increases in a delayed manner with increasing concentration.

```
pkpd_data_wide <- pkpd_data_wide %>% arrange(ID, TIME)

gg <- ggplot(data = pkpd_data_wide, aes(x = CONC, y = PD, color = TIME))
gg <- gg + xgx_annotate_status(status)
gg <- gg + geom_path(arrow = arrow(length = unit(0.15,"cm")))
gg <- gg + labs(x = conc_label , y = pd_label)
gg <- gg + xgx_scale_x_log10()
gg <- gg + xgx_scale_y_log10()
gg <- gg + theme(panel.grid.minor.x = ggplot2::element_line(color = rgb(0.9,0.9,0.9)),
                 panel.grid.minor.y = ggplot2::element_line(color = rgb(0.9,0.9,0.9)))
```

```
gg + facet_wrap(~ID + TRTACT_low2high, ncol = 5)
```


1001
0.003 mg/kg i.v.
1002
0.003 mg/kg i.v.
1003
0.003 mg/kg i.v.
1004
0.003 mg/kg i.v.
1005
0.003 mg/kg i.v.
1006
0.003 mg/kg i.v.
1009
0.01 mg/kg i.v.
1010
0.01 mg/kg i.v.
1012
0.01 mg/kg i.v.
1013
0.01 mg/kg i.v.
1014
0.01 mg/kg i.v.
1016
0.01 mg/kg i.v.
1017
0.03 mg/kg i.v.
1018
0.03 mg/kg i.v.
1019
0.03 mg/kg i.v.
1021
0.03 mg/kg i.v.
1022
0.03 mg/kg i.v.
1024
0.03 mg/kg i.v.
1025
0.1 mg/kg i.v.
1026
0.1 mg/kg i.v.
1027
0.1 mg/kg i.v.
1028
0.1 mg/kg i.v.
1030
0.1 mg/kg i.v.
1031
0.1 mg/kg i.v.
1033
0.3 mg/kg i.v.
1034
0.3 mg/kg i.v.
1035
0.3 mg/kg i.v.
1036
0.3 mg/kg i.v.
1039
0.3 mg/kg i.v.
1040
0.3 mg/kg i.v.
1041
1 mg/kg i.v.
1042
1 mg/kg i.v.
1043
1 mg/kg i.v.
1044
1 mg/kg i.v.
1045
1 mg/kg i.v.
1048
1 mg/kg i.v.
1050
3 mg/kg i.v.
1051
3 mg/kg i.v.
1053
3 mg/kg i.v.
1054
3 mg/kg i.v.
1055
3 mg/kg i.v.
1057
3 mg/kg i.v.
1058
3 mg/kg i.v.
1059
3 mg/kg i.v.
1060
3 mg/kg i.v.
1061
3 mg/kg i.v.
1062
3 mg/kg i.v.
1064
10 mg/kg i.v.
1065
10 mg/kg i.v.
1066
10 mg/kg i.v.
1067
10 mg/kg i.v.
1068
10 mg/kg i.v.
1070
10 mg/kg i.v.
1072
450 mg s.c.
1073
450 mg s.c.
1074
450 mg s.c.
1075
450 mg s.c.
1076
450 mg s.c.
1077
450 mg s.c.
1079
0.1 mg/kg i.v.
1082
0.1 mg/kg i.v.
DRAFT
3000
1000
300
100
0.1
1
10
100
Continuous PD Marker (pg/mL)
Concentration (ng/mL)
TIME
3000
2000
1000
0

R Session Info

```
sessionInfo()
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Red Hat Enterprise Linux
##
## Matrix products: default
## BLAS/LAPACK: /CHBS/apps/EB/software/imkl/2019.1.144-gompi-2019a/compilers_and_libraries_2019.1.144/li
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8
##  [2] LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8
##  [4] LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8
##  [6] LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8
##  [8] LC_NAME=C
##  [9] LC_ADDRESS=C
## [10] LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8
## [12] LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils
## [5] datasets  methods   base
##
## other attached packages:
## [1] xgxr_1.0.7    tidyr_1.0.0   dplyr_0.8.3
## [4] ggplot2_3.2.1
##
## loaded via a namespace (and not attached):
##   [1] nlme_3.1-142        bitops_1.0-6
##   [3] matrixStats_0.55.0  fs_1.3.1
##   [5] usethis_1.5.1       devtools_2.2.1
##   [7] RColorBrewer_1.1-2  httr_1.4.1
##   [9] rprojroot_1.3-2     rstan_2.19.2
##  [11] tools_3.6.1         backports_1.1.5
##  [13] R6_2.4.0            DT_0.9
##  [15] rpart_4.1-15        mgcv_1.8-31
##  [17] Hmisc_4.3-0         lazyeval_0.2.2
##  [19] colorspace_1.4-1    nnet_7.3-12
##  [21] withr_2.1.2         tidyselect_0.2.5
##  [23] gridExtra_2.3       prettyunits_1.0.2
##  [25] processx_3.4.1      curl_4.2
##  [27] compiler_3.6.1      cli_1.1.0
##  [29] binom_1.1-1         htmlTable_1.13.2
##  [31] desc_1.2.0          labeling_0.3
##  [33] caTools_1.17.1.2    scales_1.0.0
##  [35] checkmate_1.9.4     callr_3.3.2
##  [37] stringr_1.4.0       digest_0.6.22
##  [39] StanHeaders_2.19.0  foreign_0.8-72
```

```
##  [41] rmarkdown_1.16      base64enc_0.1-3
##  [43] pkgconfig_2.0.3     htmltools_0.4.0
##  [45] sessioninfo_1.1.1   fastmap_1.0.1
##  [47] learnr_0.10.0       htmlwidgets_1.5.1
##  [49] rlang_0.4.1         rstudioapi_0.10
##  [51] shiny_1.4.0         jsonlite_1.6
##  [53] acepack_1.4.1       inline_0.3.15
##  [55] RCurl_1.95-4.12     magrittr_1.5
##  [57] Formula_1.2-3       loo_2.1.0
##  [59] Matrix_1.2-17       Rcpp_1.0.3
##  [61] munsell_0.5.0       lifecycle_0.1.0
##  [63] stringi_1.4.3       yaml_2.2.0
##  [65] plyr_1.8.4          pkgbuild_1.0.6
##  [67] grid_3.6.1          parallel_3.6.1
##  [69] promises_1.1.0      crayon_1.3.4
##  [71] lattice_0.20-38     splines_3.6.1
##  [73] pander_0.6.3        zeallot_0.1.0
##  [75] knitr_1.25          ps_1.3.0
##  [77] pillar_1.4.2        markdown_1.1
##  [79] reshape2_1.4.3      codetools_0.2-16
##  [81] stats4_3.6.1        pkgload_1.0.2
##  [83] glue_1.3.1          evaluate_0.14
##  [85] latticeExtra_0.6-28 data.table_1.12.6
##  [87] remotes_2.1.0       png_0.1-7
##  [89] vctrs_0.2.0         httpuv_1.5.2
##  [91] testthat_2.3.0      gtable_0.3.0
##  [93] purrr_0.3.3         assertthat_0.2.1
##  [95] xfun_0.10           mime_0.7
##  [97] xtable_1.8-4        later_1.0.0
##  [99] rsconnect_0.8.15    survival_3.1-7
## [101] tibble_2.1.3        memoise_1.1.0
## [103] cluster_2.1.0       ellipsis_0.3.0
```

title: “PK/PD, Exposure-Response - Continuous” author: “Your Name Here” date: “09 July, 2020” output: html_document: toc: true toc_float: true code_folding: hide —

Overview

This document contains exploratory plots for continuous PD data as well as the R code that generates these graphs. The plots presented here are based on simulated data (see: PKPD Datasets). Data specifications can be accessed on Datasets and Rmarkdown template to generate this page can be found on Rmarkdown-Template. You may also download the Multiple Ascending Dose PK/PD dataset for your reference (download dataset).

Setup

```
library(ggplot2)
library(dplyr)
library(tidyr)
library(xgxr)

#flag for labeling figures as draft
```

```
status = "DRAFT"

## ggplot settings
xgx_theme_set()

#directories for saving individual graphs
dirs = list(
  parent_dir= tempdir(),
  rscript_dir  = "./",
  rscript_name = "Example.R",
  results_dir  = "./",
  filename_prefix   = "",
  filename     = "Example.png")
```

Load Dataset

```
#load dataset
pkpd_data <- read.csv("https://raw.githubusercontent.com/Novartis/xgxr/master/data_create/raw/nonlinear

DOSE_CMT = 1
PK_CMT = 5
PD_CMT = 4
SS_PROFDAY = 6 # steady state prof day
PD_PROFDAYS <- c(0, 2, 4, 6)
TAU = 244 # time between doses, units should match units of TIME, e.g. 24 for QD, 12 for BID, 7*24 for

#ensure dataset has all the necessary columns
pkpd_data = pkpd_data %>%
  mutate(TIME = TIM2, NOMTIME = NT, EVID = 0, CENS = 0, DOSE = MGKG,     TRTACT = TRT, LIDV_NORM = LIDV

#create a factor for the treatment variable for plotting
pkpd_data = pkpd_data %>%
  arrange(DOSE) %>%
  mutate(TRTACT_low2high = factor(TRTACT, levels = unique(TRTACT)),
         TRTACT_high2low = factor(TRTACT, levels = rev(unique(TRTACT))))

#create pk and pd datasets
pk_data <- pkpd_data %>%
  filter(CMT==PK_CMT)

pd_data <- pkpd_data %>%
  filter(CMT==PD_CMT)

#create wide pkpd dataset for plotting PK vs PD
pkpd_data_wide <- pd_data %>%
  select(ID, NOMTIME, PD = LIDV) %>%
  right_join(pk_data) %>%
  rename(CONC = LIDV)%>%
  filter(!is.na(PD))%>%
  filter(!is.na(CONC))

#perform NCA, for additional plots
```

```
NCA = pk_data %>%
  group_by(ID, DOSE) %>%
  filter(!is.na(LIDV)) %>%
  summarize(AUC_0 = ifelse(length(LIDV[NOMTIME > 0 & NOMTIME <= TAU]) > 1,
                           caTools::trapz(TIME[NOMTIME > 0 & NOMTIME <= TAU],
                                   LIDV[NOMTIME > 0 & NOMTIME <= TAU]),
                           NA),
            Cmax_0     = ifelse(length(LIDV[NOMTIME > 0 & NOMTIME <= TAU]) > 1,
                           max(LIDV[NOMTIME > 0 & NOMTIME <= TAU]),
                           NA),
            AUC_tau = ifelse(length(LIDV[NOMTIME > (SS_PROFDAY-1)*24 &
                                         NOMTIME <= ((SS_PROFDAY-1)*24 + TAU)]) > 1,
                             caTools::trapz(TIME[NOMTIME > (SS_PROFDAY-1)*24 &
                                                 NOMTIME <= ((SS_PROFDAY-1)*24 + TAU)],
                                    LIDV[NOMTIME > (SS_PROFDAY-1)*24 &
                                           NOMTIME <= ((SS_PROFDAY-1)*24 + TAU)]),
                             NA),
            Cmax_tau     = ifelse(length(LIDV[NOMTIME > (SS_PROFDAY-1)*24 &
                                              NOMTIME <= ((SS_PROFDAY-1)*24 + TAU)]) > 1,
                             max(LIDV[NOMTIME > (SS_PROFDAY-1)*24 &
                                        NOMTIME <= ((SS_PROFDAY-1)*24 + TAU)]),
                             NA),
            SEX      = SEX[1], #this part just keeps the SEX and WEIGHTB covariates
            WEIGHTB  = WEIGHTB[1]) %>%
  gather(PARAM, VALUE,-c(ID, DOSE, SEX, WEIGHTB)) %>%
  ungroup() %>%
  mutate(VALUE_NORM = VALUE/DOSE,
         PROFDAY = ifelse(PARAM %in% c("AUC_0", "Cmax_0"), 1, SS_PROFDAY))

#add response data at day 1 and steady state to NCA for additional plots
NCA <- pd_data %>% subset(PROFDAY %in% c(1, SS_PROFDAY),) %>%
  select(ID, PROFDAY, DAY_label, PD = LIDV, TRTACT_low2high, TRTACT_high2low) %>%
  merge(NCA, by = c("ID","PROFDAY"))

#units and labels
time_units_dataset = "hours"
time_units_plot    = "days"
trtact_label       = "Dose"
dose_units         = unique((pkpd_data %>% filter(CMT == DOSE_CMT) )$LIDV_UNIT) %>% as.character()
dose_label         = paste0("Dose (", dose_units, ")")
conc_units         = unique(pk_data$LIDV_UNIT) %>% as.character()
conc_label         = paste0("Concentration (", conc_units, ")")
concnorm_label     = paste0("Normalized Concentration (", conc_units, ")/", dose_units)
AUC_units          = paste0("h.", conc_units)
pd_units           = unique(pd_data$LIDV_UNIT) %>% as.character()
pd_label           = paste0("Continuous PD Marker (", pd_units, ")")
```

Provide an overview of the data

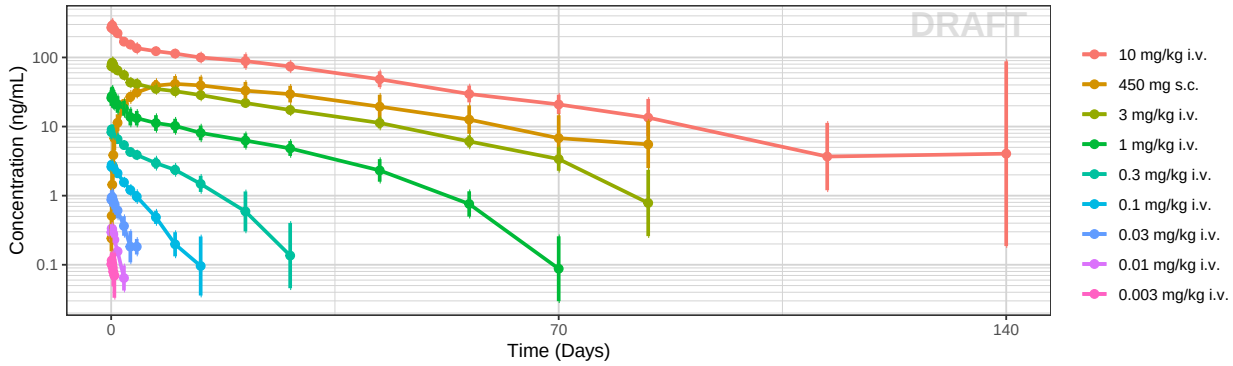
Summarize the data in a way that is easy to visualize the general trend of PD over time and between doses. Using summary statistics can be helpful, e.g. Mean +/- SE, or median, 5th & 95th percentiles. Consider either coloring by dose or faceting by dose. Depending on the amount of data one graph may be better than

the other.

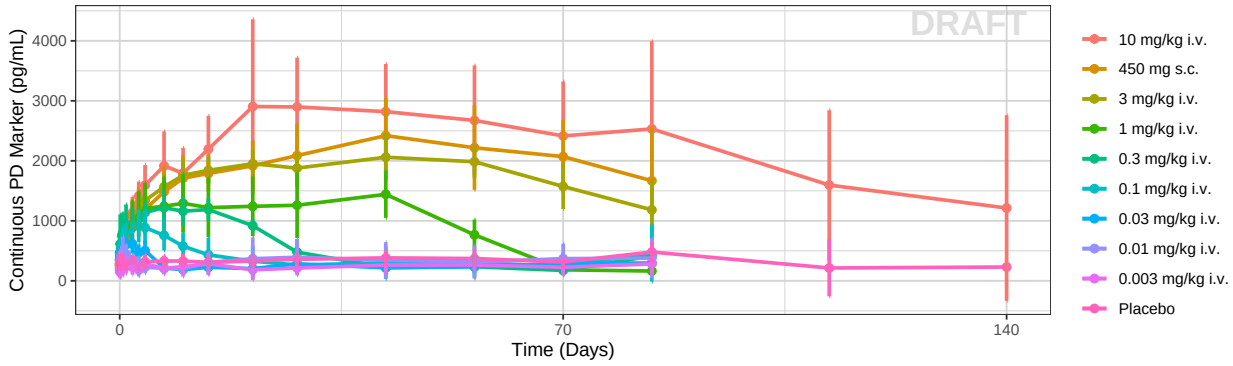
PK and PD marker over time, colored by Dose, mean (95% CI) percentiles by nominal time

Observe the overall shape of the average profiles. Does the effect appear to increase and decrease quickly on a short time scale, or does is occur over a longer time scale? Do the PK and PD profiles appear to be on the same time scale, or does the PD seem delayed compared to the PK? Is there clear separation between the profiles for different doses? Does the effect appear to increase with increasing dose? Do you detect a saturation of the effect?

```
#PK data
gg <- ggplot(data = pk_data,
             aes(x = NOMTIME,y = LIDV, color = TRTACT_high2low, fill = TRTACT_high2low))
gg <- gg + xgx_stat_ci(conf_level = .95)
gg <- gg + xgx_annotate_status(status)
gg <- gg + xgx_scale_x_time_units(units_dataset = time_units_dataset,
                                  units_plot    = time_units_plot)
gg <- gg + guides(color = guide_legend(""), fill = guide_legend(""))
gg <- gg + xgx_scale_y_log10()
gg <- gg + labs(y = conc_label)
print(gg)
```

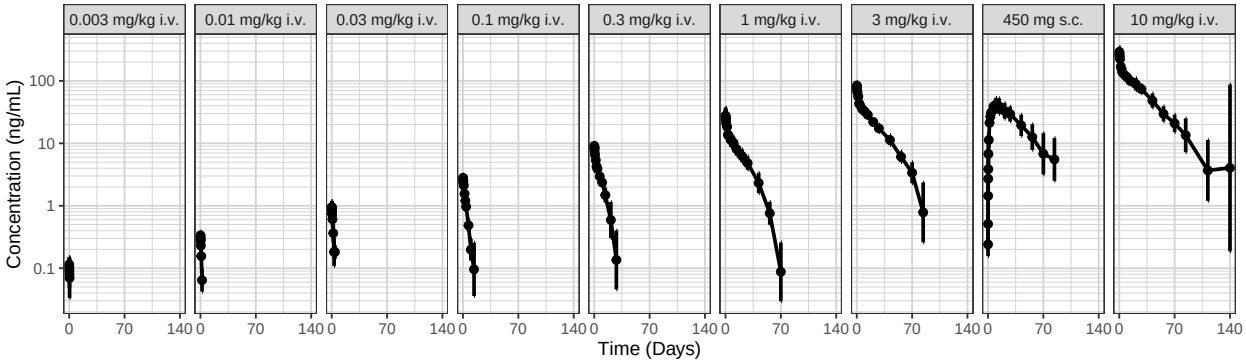



```
#PD data
gg %+% (data = pd_data) + scale_y_continuous() + labs(y = pd_label)
```

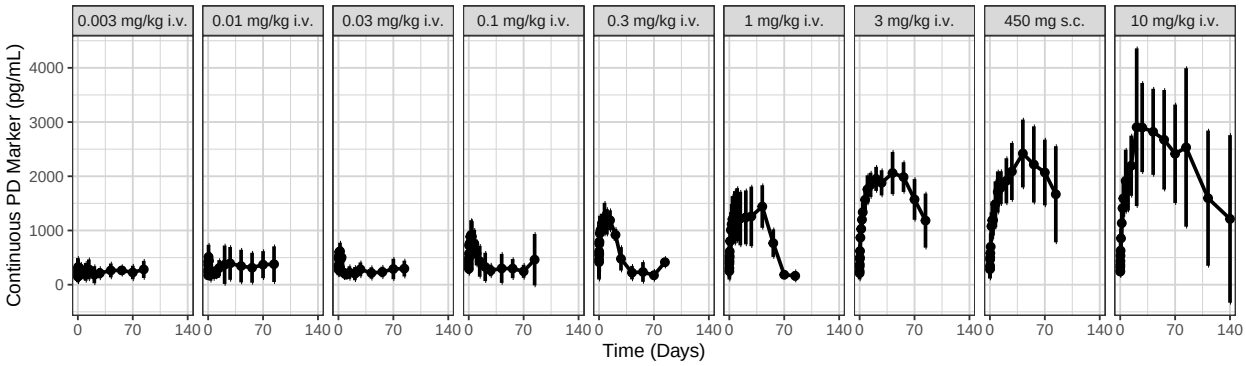


PK and PD marker over time, faceted by Dose, mean (95% CI) by nominal time

```
#PK data
gg <- ggplot(data = pk_data,
             aes(x = NOMTIME,y = LIDV))
gg <- gg + xgx_stat_ci(conf_level = .95)
gg <- gg + xgx_scale_x_time_units(units_dataset = time_units_dataset,
                                  units_plot    = time_units_plot)
gg <- gg + guides(color = guide_legend(""),fill = guide_legend(""))
gg <- gg + facet_grid(~TRTACT_low2high)
gg <- gg + xgx_scale_y_log10()
gg <- gg + labs(y = conc_label)
print(gg)
```




```
#PD data
gg %+% (data = pd_data %>% subset(DOSE>0)) + scale_y_continuous() + labs(y = pd_label)
```

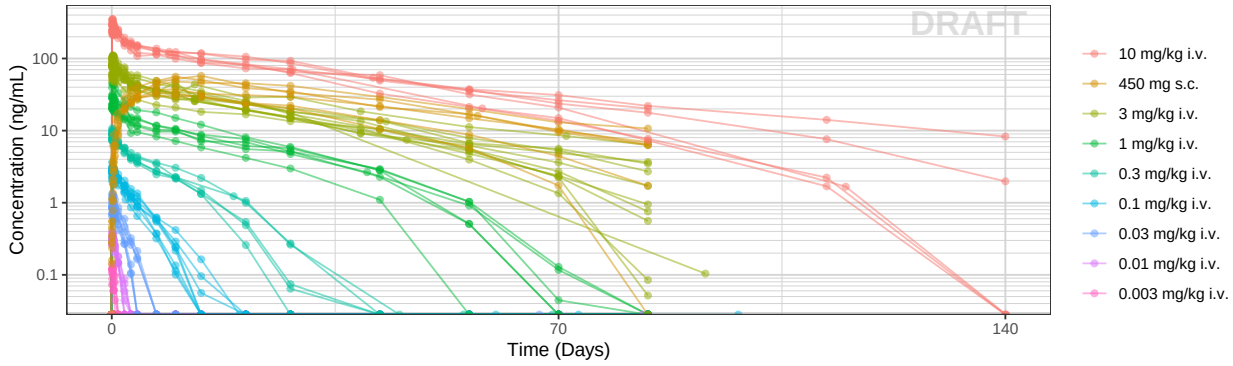


Explore variability

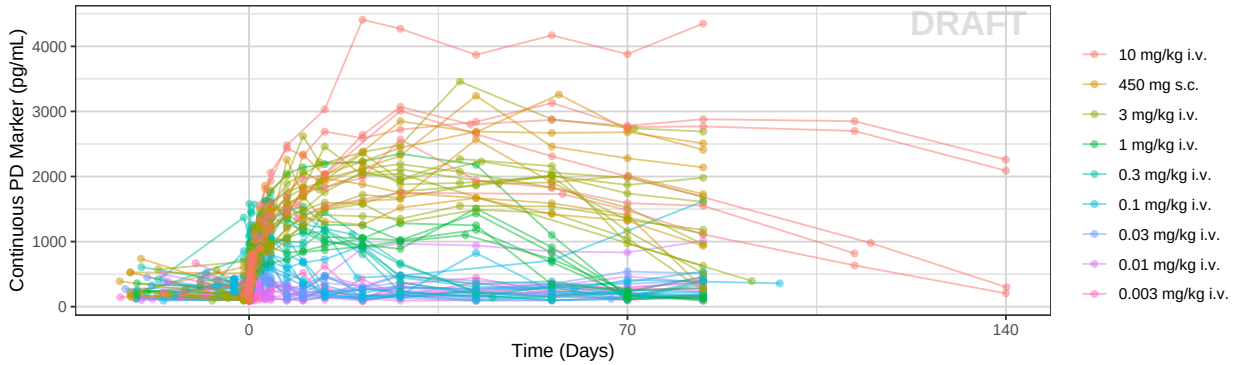
Use spaghetti plots to visualize the extent of variability between individuals. The wider the spread of the profiles, the higher the between subject variability. Distinguish different doses by color, or separate into different panels. If coloring by dose, do the individuals in the different dose groups overlap across doses? Dose there seem to be more variability at higher or lower concentrations?

PK and PD marker over time, colored by Dose, dots & lines grouped by individuals

```
#PK data
gg <- ggplot(data = pk_data,
             aes(x = TIME, y = LIDV,group = ID, color = factor(TRTACT_high2low)))
gg <- gg + xgx_annotate_status(status)
gg <- gg + geom_line(alpha = 0.5)
gg <- gg + geom_point(alpha = 0.5)
gg <- gg + guides(color = guide_legend(""), fill = guide_legend(""))
gg <- gg + xgx_scale_x_time_units(units_dataset = time_units_dataset,
                                  units_plot    = time_units_plot)
gg <- gg + geom_point(data = pk_data %>% subset(CENS == 1,), color = "red", shape = 8,alpha = 0.5)
gg <- gg + xgx_scale_y_log10()
gg <- gg + labs(y = conc_label)
print(gg)
```



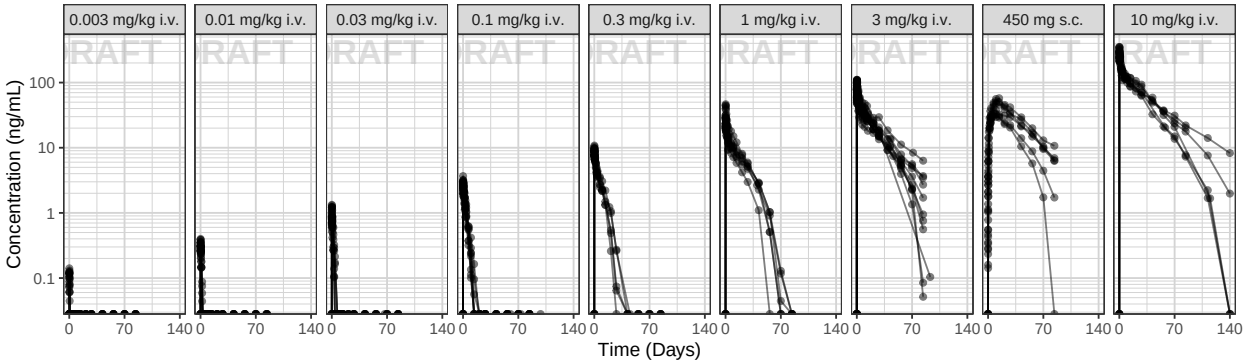

```
#PD data
gg %+% (data = pd_data %>% subset(DOSE>0)) + scale_y_continuous() + labs(y = pd_label)
```



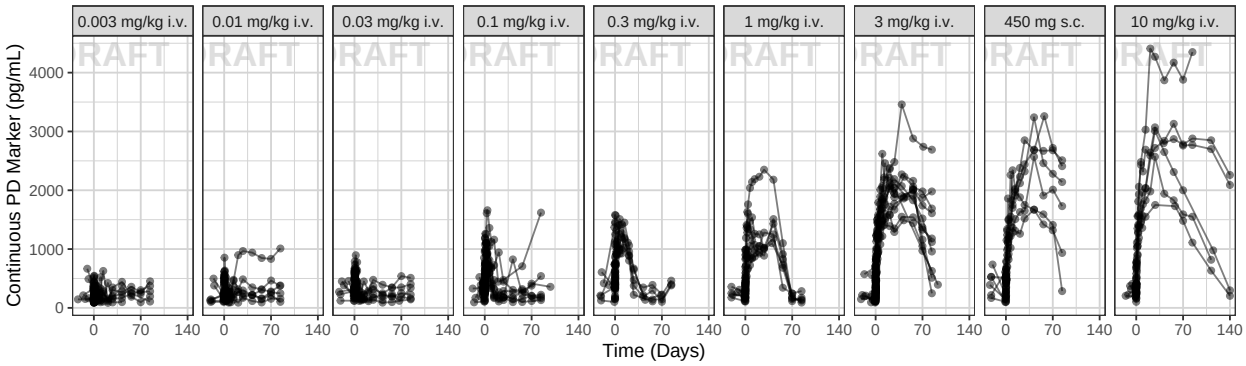
PK and PD marker over time, faceted by Dose, dots & lines grouped by individuals

```
#PK data
gg <- ggplot(data = pk_data, aes(x = TIME, y = LIDV, group = ID))
gg <- gg + xgx_annotate_status(status)
gg <- gg + geom_line(alpha = 0.5)
gg <- gg + geom_point(alpha = 0.5)
gg <- gg + guides(color = guide_legend(""), fill = guide_legend(""))
gg <- gg + xgx_scale_x_time_units(units_dataset = time_units_dataset,
```

```
                                  units_plot    = time_units_plot)
gg <- gg + facet_grid(~TRTACT_low2high)
gg <- gg + geom_point(data = pk_data %>% subset(CENS==1,), color="red", shape=8, alpha = 0.5)
gg <- gg + xgx_scale_y_log10()
gg <- gg + labs(y = conc_label)
print(gg)
```




```
#PD data
gg %+% (data = pd_data %>% subset(DOSE>0)) + scale_y_continuous() + labs(y = pd_label)
```



Explore Exposure-Response Relationship

Plot PD marker against concentration. Do you see any relationship? Does response increase (decrease) with increasing dose? Are you able to detect a plateau or emax (emin) on the effect?

Warning: Even if you don't see an Emax, that doesn't mean there isn't one. Be very careful about using linear models for Dose-Response or Exposure-Response relationships. Extrapolation outside of the observed dose range could indicate a higher dose is always better (even if it isn't).

```
gg <- ggplot(data = pkpd_data_wide %>% subset(PROFDAY == SS_PROFDAY,), aes(x = CONC,y = PD))
gg <- gg + xgx_annotate_status(status)
gg <- gg + geom_point(aes(color = TRTACT_high2low))
gg <- gg + geom_point(data = pkpd_data_wide %>% subset(CENS==1,), color="red", shape = 8)
gg <- gg + labs(x = conc_label , y = pd_label)
gg <- gg + xgx_scale_x_log10()
gg <- gg + guides(color = guide_legend(""))
print(gg)
```

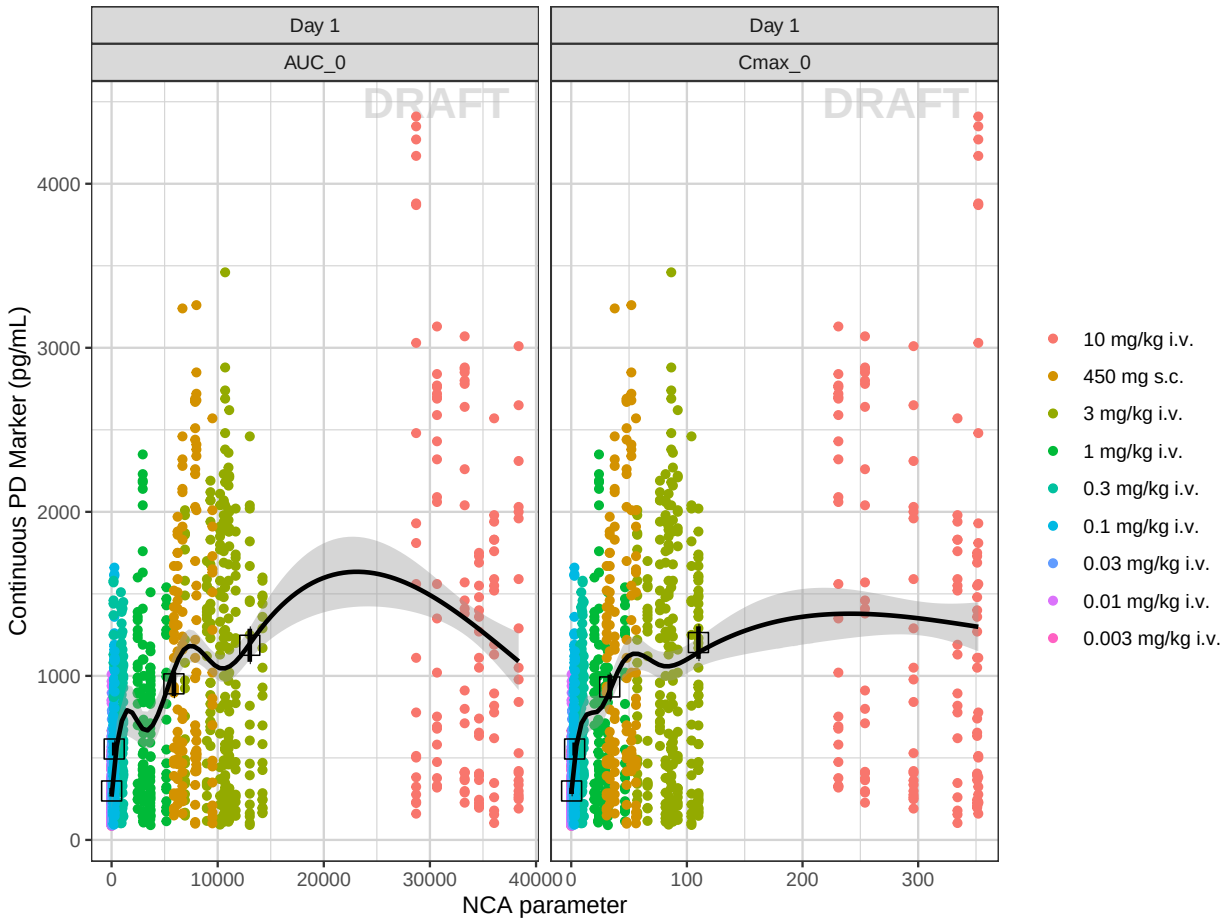
```
## Error in seq.default(from = best$lmin, to = best$lmax, by = best$lstep): 'from' must be of length 1
```

```r
data_to_plot <- pkpd_data_wide %>% subset(PROFDAY %in% PD_PROFDAYS,)

gg <- ggplot(data = data_to_plot,
             aes(x = CONC, y = PD, color = TRTACT_high2low))
gg <- gg + xgx_annotate_status(status)
gg <- gg + geom_point()
gg <- gg + geom_point(data = data_to_plot %>% subset(CENS==1,), color="red", shape = 8)
gg <- gg + guides(color = guide_legend(""), fill = guide_legend(""))
gg <- gg + labs(x = conc_label , y = pd_label)
gg <- gg + xgx_scale_x_log10()
gg + facet_grid(~DAY_label)
```

```
## Error: Faceting variables must have at least one value
```

Plotting AUC vs response instead of concentration vs response may make more sense in some situations. For example, when there is a large delay between PK and PD it would be diffcult to relate the time-varying concentration with the response. If rich sampling is only done at a particular point in the study, e.g. at steady state, then the AUC calculated on the rich profile could be used as the exposure variable for a number of PD visits. If PK samples are scarce, average Cmin could also be used as the exposure metric.

```r
NCA_plot = NCA %>%
  group_by(PARAM) %>%
  mutate(VALUE_QUART = cut(VALUE, quantile(VALUE, na.rm=TRUE), na.rm=TRUE, include.lowest = TRUE)) %>%
  group_by(PARAM, VALUE_QUART) %>%
  mutate(VALUE_MIDPOINT = median(VALUE))

gg <- ggplot(data = NCA_plot, aes(x = VALUE, y = PD))
gg <- gg + xgx_annotate_status(status)
gg <- gg + geom_point(aes(color = TRTACT_high2low)) + geom_smooth(color = "black")
gg <- gg + xgx_stat_ci(aes(x = VALUE_MIDPOINT, y = PD), geom = "errorbar")
gg <- gg + xgx_stat_ci(aes(x = VALUE_MIDPOINT, y = PD), geom = "point", shape = 0, size = 4)
gg <- gg + guides(color = guide_legend(""), fill = guide_legend(""))
gg <- gg + labs(color = trtact_label, x = "NCA parameter", y = pd_label)
gg <- gg + facet_wrap(~DAY_label + PARAM, scales = "free_x")
print(gg)
```

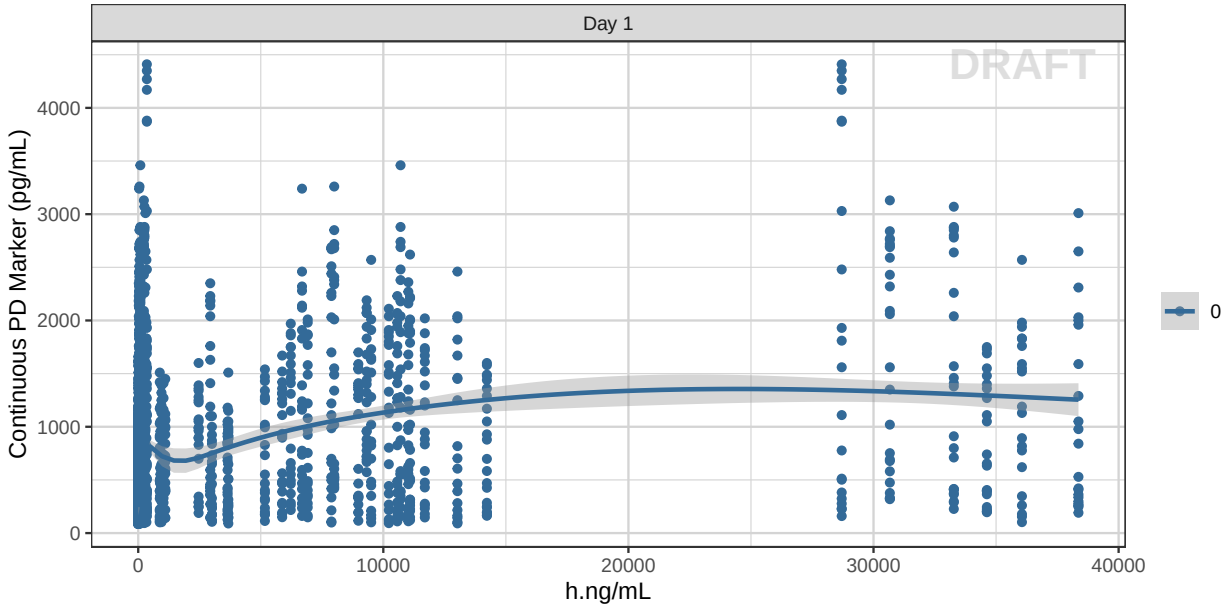


Explore covariate effects on Exposure-Response Relationship

Stratify exposure-response plots by covariates of interest to explore whether any key covariates impact response independent of exposure.

```r
gg <- ggplot(data = NCA_plot, aes(x = VALUE, y = PD))
gg <- gg + xgx_annotate_status(status)
```

```
gg <- gg + geom_point(aes(color = SEX)) + geom_smooth(aes(color = SEX))
gg <- gg + guides(color = guide_legend(""), fill = guide_legend(""))
gg <- gg + labs(x = AUC_units , y = pd_label)
gg + facet_grid(.~DAY_label)
```



Individual response vs exposure hysteresis plots

Using geom_path you can create hysteresis plots of response vs exposure. Including details like arrows or colors can be helpful to indicate the direction of time.

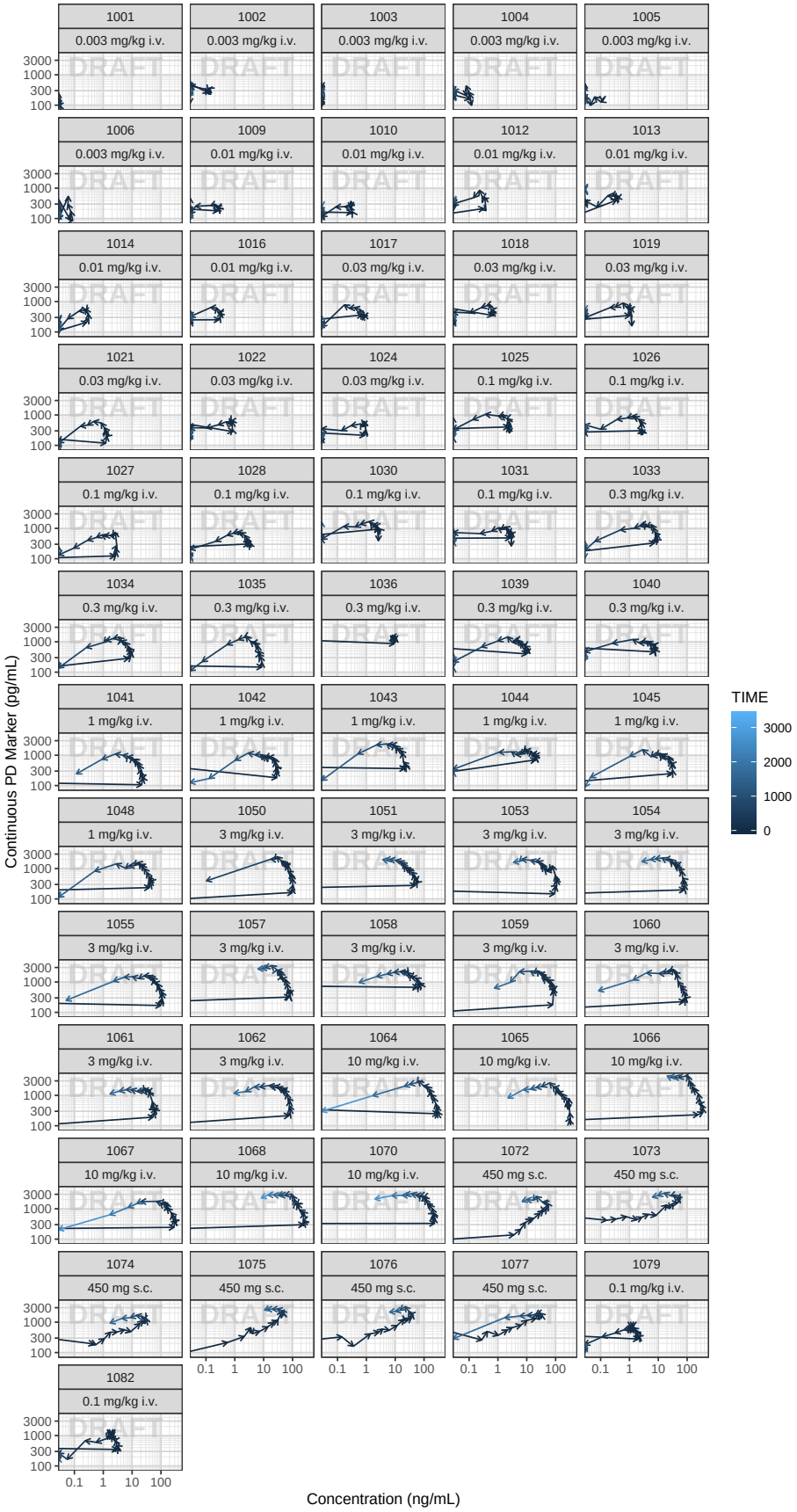
If most of the arrows point up and to the right or down and to the left, this indicates a positive relationship between exposure and response (i.e. increasing exposure -> increasing response). Arrows pointing down and to the right or up and to the left indicate a negative relationship.

In a hysteresis plot, you want to determine whether the path is curving, and if so in what direction. If you detect a curve in the hysteresis plot, this indicates there is a delay between the exposure and the response. Normally, a clockwise turn indicates that increasing exposure is associated with (a delayed) increasing response, while a counter clockwise turn indicates increasing concentration gives (a delayed) decreasing response.

In the plots below, most of the hysteresis paths follow a counter clockwise turn, with most arrows pointing up and to the right or down and to the left, indicating the effect increases in a delayed manner with increasing concentration.

```
pkpd_data_wide <- pkpd_data_wide %>% arrange(ID, TIME)

gg <- ggplot(data = pkpd_data_wide, aes(x = CONC, y = PD, color = TIME))
gg <- gg + xgx_annotate_status(status)
gg <- gg + geom_path(arrow = arrow(length = unit(0.15,"cm")))
gg <- gg + labs(x = conc_label , y = pd_label)
gg <- gg + xgx_scale_x_log10()
gg <- gg + xgx_scale_y_log10()
gg <- gg + theme(panel.grid.minor.x = ggplot2::element_line(color = rgb(0.9,0.9,0.9)),
                 panel.grid.minor.y = ggplot2::element_line(color = rgb(0.9,0.9,0.9)))
```

```
gg + facet_wrap(~ID + TRTACT_low2high, ncol = 5)
```


1001
0.003 mg/kg i.v.
1002
0.003 mg/kg i.v.
1003
0.003 mg/kg i.v.
1004
0.003 mg/kg i.v.
1005
0.003 mg/kg i.v.
1006
0.003 mg/kg i.v.
1009
0.01 mg/kg i.v.
1010
0.01 mg/kg i.v.
1012
0.01 mg/kg i.v.
1013
0.01 mg/kg i.v.
1014
0.01 mg/kg i.v.
1016
0.01 mg/kg i.v.
1017
0.03 mg/kg i.v.
1018
0.03 mg/kg i.v.
1019
0.03 mg/kg i.v.
1021
0.03 mg/kg i.v.
1022
0.03 mg/kg i.v.
1024
0.03 mg/kg i.v.
1025
0.1 mg/kg i.v.
1026
0.1 mg/kg i.v.
1027
0.1 mg/kg i.v.
1028
0.1 mg/kg i.v.
1030
0.1 mg/kg i.v.
1031
0.1 mg/kg i.v.
1033
0.3 mg/kg i.v.
1034
0.3 mg/kg i.v.
1035
0.3 mg/kg i.v.
1036
0.3 mg/kg i.v.
1039
0.3 mg/kg i.v.
1040
0.3 mg/kg i.v.
1041
1 mg/kg i.v.
1042
1 mg/kg i.v.
1043
1 mg/kg i.v.
1044
1 mg/kg i.v.
1045
1 mg/kg i.v.
1048
1 mg/kg i.v.
1050
3 mg/kg i.v.
1051
3 mg/kg i.v.
1053
3 mg/kg i.v.
1054
3 mg/kg i.v.
1055
3 mg/kg i.v.
1057
3 mg/kg i.v.
1058
3 mg/kg i.v.
1059
3 mg/kg i.v.
1060
3 mg/kg i.v.
1061
3 mg/kg i.v.
1062
3 mg/kg i.v.
1064
10 mg/kg i.v.
1065
10 mg/kg i.v.
1066
10 mg/kg i.v.
1067
10 mg/kg i.v.
1068
10 mg/kg i.v.
1070
10 mg/kg i.v.
1072
450 mg s.c.
1073
450 mg s.c.
1074
450 mg s.c.
1075
450 mg s.c.
1076
450 mg s.c.
1077
450 mg s.c.
1079
0.1 mg/kg i.v.
1082
0.1 mg/kg i.v.
DRAFT
3000
1000
300
100
0.1
1
10
100
Continuous PD Marker (pg/mL)
Concentration (ng/mL)
TIME
3000
2000
1000
0

R Session Info

```
sessionInfo()
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Red Hat Enterprise Linux
##
## Matrix products: default
## BLAS/LAPACK: /CHBS/apps/EB/software/imkl/2019.1.144-gompi-2019a/compilers_and_libraries_2019.1.144/li
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8
##  [2] LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8
##  [4] LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8
##  [6] LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8
##  [8] LC_NAME=C
##  [9] LC_ADDRESS=C
## [10] LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8
## [12] LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils
## [5] datasets  methods   base
##
## other attached packages:
## [1] xgxr_1.0.7    tidyr_1.0.0   dplyr_0.8.3
## [4] ggplot2_3.2.1
##
## loaded via a namespace (and not attached):
##   [1] nlme_3.1-142        bitops_1.0-6
##   [3] matrixStats_0.55.0  fs_1.3.1
##   [5] usethis_1.5.1       devtools_2.2.1
##   [7] RColorBrewer_1.1-2  httr_1.4.1
##   [9] rprojroot_1.3-2     rstan_2.19.2
##  [11] tools_3.6.1         backports_1.1.5
##  [13] R6_2.4.0            DT_0.9
##  [15] rpart_4.1-15        mgcv_1.8-31
##  [17] Hmisc_4.3-0         lazyeval_0.2.2
##  [19] colorspace_1.4-1    nnet_7.3-12
##  [21] withr_2.1.2         tidyselect_0.2.5
##  [23] gridExtra_2.3       prettyunits_1.0.2
##  [25] processx_3.4.1      curl_4.2
##  [27] compiler_3.6.1      cli_1.1.0
##  [29] binom_1.1-1         htmlTable_1.13.2
##  [31] desc_1.2.0          labeling_0.3
##  [33] caTools_1.17.1.2    scales_1.0.0
##  [35] checkmate_1.9.4     callr_3.3.2
##  [37] stringr_1.4.0       digest_0.6.22
##  [39] StanHeaders_2.19.0  foreign_0.8-72
```

```
##  [41] rmarkdown_1.16      base64enc_0.1-3
##  [43] pkgconfig_2.0.3     htmltools_0.4.0
##  [45] sessioninfo_1.1.1   fastmap_1.0.1
##  [47] learnr_0.10.0       htmlwidgets_1.5.1
##  [49] rlang_0.4.1         rstudioapi_0.10
##  [51] shiny_1.4.0         jsonlite_1.6
##  [53] acepack_1.4.1       inline_0.3.15
##  [55] RCurl_1.95-4.12     magrittr_1.5
##  [57] Formula_1.2-3       loo_2.1.0
##  [59] Matrix_1.2-17       Rcpp_1.0.3
##  [61] munsell_0.5.0       lifecycle_0.1.0
##  [63] stringi_1.4.3       yaml_2.2.0
##  [65] plyr_1.8.4          pkgbuild_1.0.6
##  [67] grid_3.6.1          parallel_3.6.1
##  [69] promises_1.1.0      crayon_1.3.4
##  [71] lattice_0.20-38     splines_3.6.1
##  [73] pander_0.6.3        zeallot_0.1.0
##  [75] knitr_1.25          ps_1.3.0
##  [77] pillar_1.4.2        markdown_1.1
##  [79] reshape2_1.4.3      codetools_0.2-16
##  [81] stats4_3.6.1        pkgload_1.0.2
##  [83] glue_1.3.1          evaluate_0.14
##  [85] latticeExtra_0.6-28 data.table_1.12.6
##  [87] remotes_2.1.0       png_0.1-7
##  [89] vctrs_0.2.0         httpuv_1.5.2
##  [91] testthat_2.3.0      gtable_0.3.0
##  [93] purrr_0.3.3         assertthat_0.2.1
##  [95] xfun_0.10           mime_0.7
##  [97] xtable_1.8-4        later_1.0.0
##  [99] rsconnect_0.8.15    survival_3.1-7
## [101] tibble_2.1.3        memoise_1.1.0
## [103] cluster_2.1.0       ellipsis_0.3.0
```