|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
|  | ggPMX Fast and Furious Tips  Dear **PMX colleagues**,  Here is a collection of all monthly Tips that were shared with you so far – for an easy reference.  April 2020 – VPC set up  **Question: Why VPC with ggPMX does not work despite having provided a simulation dataset?**    **Answer:** Please ensure that the ID numbers generated by simulx are the original ID numbers contained in your modeling dataset and that the ID column has the same name as in your modeling dataset!    The following R-code shows how to deal with the pre-requisite above:  **C:\Users\baltcir1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\5C3CEA67.tmp**  June 2020 – BLOQ  **Question: How to display BLOQ data with ggPMX**    **Answer:** In all diagnostic plots, ggPMX uses color-coding to display BLOQ data – by default in red - by specifying the censoring column in the controller. For individual plots, as in Monolix, a bar is displayed from the censored value specified in the data set and the associated limit. Note that the simulated BLOQ feature of Monolix is not currently implemented in ggPMX.  C:\Users\baltcir1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\5015280D.tmp    C:\Users\baltcir1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\664D82A3.tmp  September 2020 – Substitute TIME  **Question:**Is it possible to substitute TIME for another time metric on a subset of diagnostic plots? For example, how to generate NPDE plot with TAD on x-axis instead of TIME?    **Answer:** You can use the aess argument within the ggPMX plot function as follows:  C:\Users\baltcir1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9CDF37A9.tmp    The **aess**argument can be used with any other ggPMX plot function, except with pmx\_plot\_vpc().    Note that if you want to produce **all diagnostic plots** with the TAD instead of TIME, the best option is to create the controller using TAD as time variable:  C:\Users\baltcir1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\236DE09F.tmp  December 2020 – Plot subset of covariates  **Question:**How can I plot only a subset of the covariates in my diagnostic plots?    **Answer:**To get only a subset of covariates to be plotted in all graphs, you should **copy the controller** using the function **pmx\_copy()**. Below is the code using the theophilline built-in example:    C:\Users\baltcir1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\BFAD805B.tmp  Q1 2021  **Question:** I am displaying diagnostics stratified by categorical covariates, but I have too many strata and all stratified graphs on the same row do not look good. How can I specify the number of rows and columns for stratified diagnostics?    **Answer:** Row and column numbers can’t be specified in *pmx\_plot\_cats*. However, you can use the *filter* argument within the *pmx\_plot\_cats* function to plot a subset of categories as follows:  ctr %>%  pmx\_plot\_cats("dv\_pred", cats = "DoseStr",  filter = DoseStr %in% c("MAD 10 mg Crystalline", "MAD 30 mg Crystalline"))    Q2 2021  **Question:** How can I generate diagnostics reports for a series of models?    **Answer:**You can use the below script that loops over all models in a folder and creates diagnostics reports for each of them (including VPC).  # produce diagnostics reports for all models    format(Sys.time(),'%d-%b-%Y, %A, %H:%M')  initMlxR(path = '/CHBS/apps/Monolix/2019R2')  library(ggPMX)  MLXTRAN\_ALL = list.files(pathModels, pattern='\\.mlxtran$',recursive=TRUE)  print(paste0('Working in: ',pathModels))    as.data.frame(MLXTRAN\_ALL) %>% kable(caption='Models to rerun')  for (run\_cur in MLXTRAN\_ALL){  proj\_cur = file.path(pathModels,run\_cur)  print(proj\_cur)  if(file.exists(proj\_cur)){  model\_subfolder = sub("/.\*mlxtran", "", run\_cur)    # Simulation for VPCs will be saved in the following file:  mysimfilename = paste0(sub("/VAY736.mlxtran", "/", proj\_cur), "sim4VPC.csv")    ## Create simulated object using simulx  mysim <- simulx(project=proj\_cur, nrep=20)  ## Retrieve simulated dataset (assumed to be in y1)  simdata <- mysim$LIDV  ## Need to revert the original IDs as in modeling dataset for ggPMX  ## Rename IDs column to same name as in modeling dataset, e.g.  ## “id” in the example below  simdata2 <- simdata %>%  mutate(newId = as.numeric(as.character(id))) %>%  left\_join(., mysim$originalId) %>%  mutate(id = as.numeric(as.character(oriId))) %>%  select(-oriId, -newId) %>%  data.table::data.table()    # Create ggPMX controller:  ctr = pmx\_mlxtran(file\_name = proj\_cur,  sim = pmx\_sim(data = simdata2, irun ="rep", idv="TIME"))    # Create GoF report using custom template  ctr %>% pmx\_report(name = "GoF\_report",  save\_dir = file.path(pathModels, model\_subfolder),  template = file.path(pathScripts, "GoF\_report\_template.Rmd"),  format = "both", # report and plots  extension= "all", #.docx, .pdf, .html  footnote = TRUE)    # Create individual plots report  ctr %>% pmx\_report(name = "Individual\_fits",  save\_dir = file.path(pathModels, model\_subfolder),  template = file.path(pathScripts, "Individual\_fits\_report\_template.Rmd"),  format = "report", # report and plots  npage = NULL,  extension= "all", #.docx, .pdf, .html  footnote = TRUE)  }  }  ## System settings  sessionInfo()  format(Sys.time(),'%d-%b-%Y, %A, %H:%M')  ## END OF FILE  ggPMX Team  Sponsor: D. Renard  Team: B. Bieth, I. Baltcheva, S. Bhattacharya, M. Fidler  For internal use only and not to be shown, distributed or disseminated outside of Novartis |  |
|  |  |  |
|  | Novartis |  |
|  | | |