

Testing package with devtools

Emmanuelle Comets

02/02/2021

Notes before release

- package
 - **TODO** automatically load libraries when loading *npde* ()
 - ggplot2 not loaded, gridExtra, mclust not loaded
 - check: mclust
 - compilation: compiled package as npde 3.0
 - * version number 3.0 because reference profiles included
 - * lots of error messages for “no visible binding for global variable” in variables related to ggplot2 (Undefined global functions or variables) => hope this does not block us for CRAN
- TODO *npde website* (bookdown, Eco)
- code **TODO** for 3.1
 - add an option to remove all output when required and remove warnings if requested
 - plot for a NpdeData object
 - * change index labels to allow duplicate time points
- **TODO** simplify output for *show()* function (v3.0 or v3.1 ?)

Setup, loading libraries

Install package in development mode

```
dev_mode() # development mode

## v Dev mode: ON
install.packages(pkgs=file.path(compileDir,"npde_3.2.tar.gz"),repos=NULL)

## Installing package into '/home/eco/R-dev'
## (as 'lib' is unspecified)
library(npde)

if(FALSE) {
  source(file.path(workDir,"npde","R","aaa_generics.R"))
  source(file.path(workDir,"npde","R","npdeControl.R"))
  source(file.path(workDir,"npde","R","NpdeData.R"))
  source(file.path(workDir,"npde","R","NpdeSimData.R"))
  source(file.path(workDir,"npde","R","NpdeData-methods.R"))
  source(file.path(workDir,"npde","R","NpdeRes.R"))
  source(file.path(workDir,"npde","R","NpdeRes-methods.R"))
  source(file.path(workDir,"npde","R","NpdeObject.R"))
  source(file.path(workDir,"npde","R","NpdeObject-methods.R"))
```

```

# Computational functions from the package
source(file.path(workDir,"npde","R","compute_distribution.R"))
source(file.path(workDir,"npde","R","compute_npde.R"))
source(file.path(workDir,"npde","R","compute_pd.R"))
source(file.path(workDir,"npde","R","compute_ploq.R"))
source(file.path(workDir,"npde","R","mainNpde.R"))
source(file.path(workDir,"npde","R","npde.R"))

# Plot functions from the package - kept, some changes
source(file.path(workDir,"npde","R","plotNpde-methods.R"))
source(file.path(workDir,"npde","R","plotNpde-plotFunctions.R"))
source(file.path(workDir,"npde","R","plotNpde-binningPI.R")) # renamed from plotNpde-unitFunctionsPI.R

# Distribution plots
source(file.path(workDir,"npde","R","plotNpde-auxDistPlot.R")) # renamed from plotNpde-unitFunctionsAux.R
source(file.path(workDir,"npde","R","plotNpde-distributionPlot.R")) # extracted "main" for distribution

# Scatterplots
source(file.path(workDir,"npde","R","plotNpde-auxScatter.R")) # replaces plotNpde-unitFunctionsRef.R
source(file.path(workDir,"npde","R","plotNpde-auxScatterPlot.R"))
source(file.path(workDir,"npde","R","plotNpde-scatterplot.R")) # replaces plotNpde-meanProf.R
source(file.path(workDir,"npde","R","plotNpde-covplot.R"))
}

}

```

Running examples from documentation

Checking help files work

OK

Computing npde for theophylline data

Checking that the results are the same when we use data frames versus files on disk: OK

Missing data (lines with MDV=1) removed from the data for plots (also removed from simulated data for approx.pi=TRUE, as per documentation the simulated and observed data must have compatible dimensions).

```

## -----
## Distribution of npde :
##      nb of obs: 120
##      mean= 0.0668   (SE= 0.095 )
##      variance= 1.074   (SE= 0.14 )
##      skewness= 0.511
##      kurtosis= 0.2912
## -----
## Statistical tests (adjusted p-values):
##      t-test          : 1
##      Fisher variance test : 1
##      SW test of normality : 0.00819 **
##      Global test         : 0.00819 **
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1

```

```

## -----
## Object of class NpdeObject
## -----
## --- Component data -----
## -
## Object of class NpdeData
##     Structured data: Conc ~ Time | ID
## This object has the following components:
##     data: data
##     with 12 subjects
##     120 observations
## The data has the following components
##     X: Time (hr)
##     Y: Conc (mg/L)
##     missing data: mdv (1=missing)
##     LOQ: NA
## -----
## --- Component results -----
## -
## Object of class NpdeRes
##     containing the following elements:
##     predictions (ypred)
##     prediction discrepancies (pd)
##     normalised prediction distribution errors (npde)
##     completed responses (ycomp) for censored data
##     decorrelated responses (ydobs)
##     the dataframe has 120 non-missing observations and 132 lines.
## First 10 lines of results, removing missing observations:
##     ypred ycomp pd ydobs npde
## 2 2.923864 2.84 0.550 -0.05124648 0.1256613
## 3 4.682299 6.57 0.850 1.96398150 2.0537489
## 4 6.264357 10.50 0.990 2.56602650 2.3263479
## 5 6.986255 9.66 0.980 0.41616411 0.5244005
## 6 6.511039 8.58 0.930 0.28430866 0.2533471
## 7 5.895675 8.36 0.960 0.54879386 0.6744898
## 8 5.064736 7.47 0.970 1.79335938 1.6448536
## 9 4.302909 6.89 0.990 0.80506269 0.7721932
## 10 3.294020 5.94 0.995 1.91537662 1.7506861
## 11 1.168743 3.28 0.995 3.25535923 2.5758293
## -
## Distribution of npde :
##     nb of obs: 120
##     mean= 0.0668 (SE= 0.095 )
##     variance= 1.074 (SE= 0.14 )
##     skewness= 0.511
##     kurtosis= 0.2912
## -
## Statistical tests (adjusted p-values):
##     t-test : 1
##     Fisher variance test : 1
##     SW test of normality : 0.00819 **
##     Global test : 0.00819 **
## -

```

```

## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## -----
##      ypred ycomp   pd      ydobs      npde
## 1     NaN  NaN  NaN       NA       NA
## 2 2.923864 2.84 0.55 -0.05124648 0.1256613
## 3 4.682299 6.57 0.85 1.96398150 2.0537489
## 4 6.264357 10.50 0.99 2.56602650 2.3263479
## 5 6.986255 9.66 0.98 0.41616411 0.5244005
## 6 6.511039 8.58 0.93 0.28430866 0.2533471

```

Same results with data frame and with data on disk: TRUE

Show and print function :

- modified so that print(theofit) shows the results of gof.test applied to the object
- show displays the first 10 lines of the results dataframe
- **TODO** simplify output for *show()* function (v3.0 or v3.1 ?)

```

## Object of class NpdeObject
## -----
## ----- Component data -----
## -----
## Object of class NpdeData
## Structured data: Conc ~ Time | ID
## This object has the following components:
##   data: data
##   with 12 subjects
##   120 observations
## The data has the following components
##   X: Time (hr)
##   Y: Conc (mg/L)
##   missing data: mdv (1=missing)
##   LOQ: NA
## -----
## ----- Component results -----
## -----
## Object of class NpdeRes
## containing the following elements:
##   predictions (ypred)
##   prediction discrepancies (pd)
##   normalised prediction distribution errors (npde)
##   completed responses (ycomp) for censored data
##   decorrelated responses (ydobs)
## the dataframe has 120 non-missing observations and 132 lines.
## First 10 lines of results, removing missing observations:
##      ypred ycomp   pd      ydobs      npde
## 2 2.923864 2.84 0.550 -0.05124648 0.1256613
## 3 4.682299 6.57 0.850 1.96398150 2.0537489
## 4 6.264357 10.50 0.990 2.56602650 2.3263479
## 5 6.986255 9.66 0.980 0.41616411 0.5244005
## 6 6.511039 8.58 0.930 0.28430866 0.2533471
## 7 5.895675 8.36 0.960 0.54879386 0.6744898
## 8 5.064736 7.47 0.970 1.79335938 1.6448536
## 9 4.302909 6.89 0.990 0.80506269 0.7721932
## 10 3.294020 5.94 0.995 1.91537662 1.7506861

```

```

## 11 1.168743 3.28 0.995 3.25535923 2.5758293

## Object of class NpdeObject
## -----
## ----- Data -----
## -----
## Object of class NpdeData
##   longitudinal data
##   Structured data: Conc ~ Time | ID
##   predictor: Time (hr)
## Dataset characteristics:
##   number of subjects:      12
##   number of non-missing observations: 120
##   average/min/max nb obs: 10.00 / 10 / 10
##   LOQ:      NA
## First 10 lines of data:
##   index ID  Time  Conc mdv
## 1 1 0.00  NA 1
## 2 1 0.25 2.84 0
## 3 1 0.57 6.57 0
## 4 1 1.12 10.50 0
## 5 1 2.02 9.66 0
## 6 1 3.82 8.58 0
## 7 1 5.10 8.36 0
## 8 1 7.03 7.47 0
## 9 1 9.05 6.89 0
## 10 1 12.12 5.94 0
##
## Summary of original data:
##   vector of predictor Time
##   Min. 1st Qu. Median Mean 3rd Qu. Max.
##   0.000 0.595 3.530 5.895 9.000 24.650
##   vector of response Conc
##   Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
##   0.850 3.513 5.665 5.447 7.325 11.400 12
## -----
## ----- Key options -----
## -----
## Methods
##   compute normalised prediction discrepancies (npd): yes
##   compute normalised prediction distribution errors (npde): yes
##   method for decorrelation: Cholesky decomposition (upper triangular)
##   method to treat censored data: Impute pd* and compute y* as F-1(pd*)
## Input/output
##   verbose (prints a message for each new subject): FALSE
##   save the results to a file, save graphs: FALSE
## -----
## ----- Results -----
## -----
## Object of class NpdeRes
##   resulting from a call to npde or autonpde
##   containing the following elements:
##   predictions (ypred)
##   Min. 1st Qu. Median Mean 3rd Qu. Max.

```

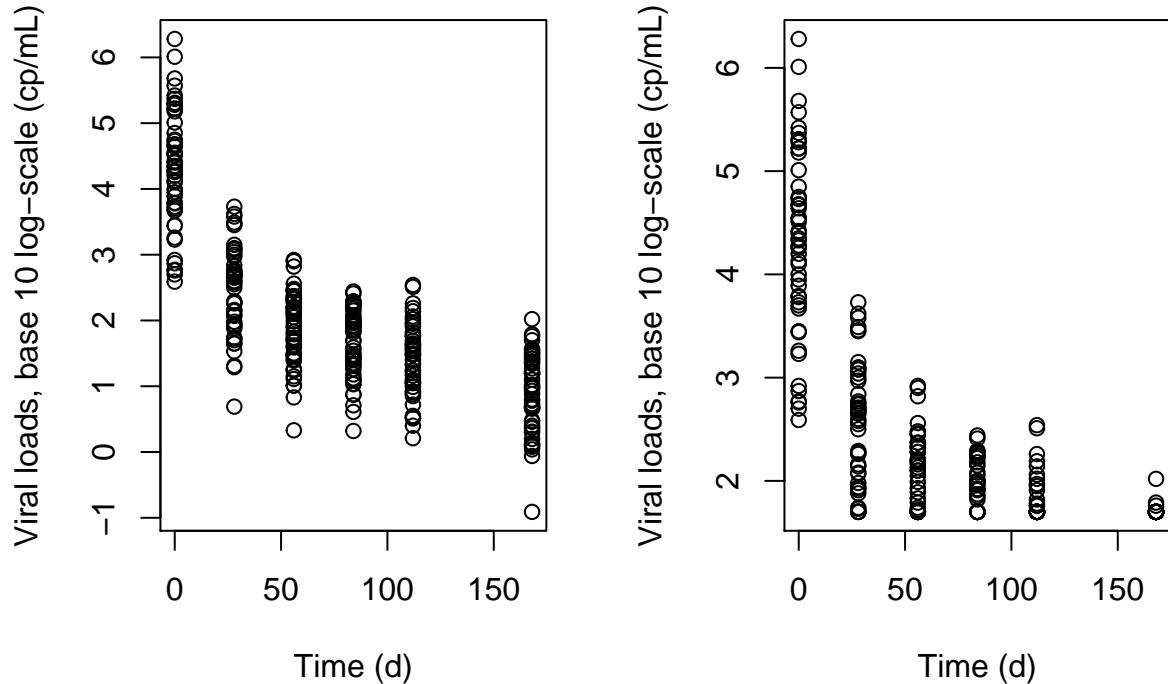
```

##  0.8897 4.0194 5.5705 5.5208 7.1204 10.0162
##      prediction discrepancies (pd)
##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
##  0.0500  0.2600  0.4300  0.4807  0.7100  0.9950
##      normalised prediction distribution errors (npde)
##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## -2.3263 -0.5102  0.0000  0.0668  0.5244  2.5758
##      completed responses (ycomp) for censored data
##      decorrelated responses (ydobs)
##      the dataframe has 120 non-missing observations and 132 lines.

```

Computing npde for viral load data

- For censored viral load data (20 or 50 cp/mL), we compute the npde using the different methods implemented in the package. We expect:
 - NS for yvir1, yvir20, yvir50 (complete datasets, and censored datasets with imputed BQL)
 - significant for ppred method
 - NS for omit method because we also omit BQL data in the simulations (so OK)
 - NS for ipred method



```

## -----
## Distribution of npde :
##   nb of obs: 300
##   mean= 0.03821   (SE= 0.053 )
##   variance= 0.8327   (SE= 0.068 )
##   skewness= -0.04464
##   kurtosis= -0.2207
## -----
## Statistical tests (adjusted p-values):
##   t-test           : 1
##   Fisher variance test : 0.0959 .
##   SW test of normality : 1

```

```

##   Global test      : 0.0959 .
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## -----
## 
## Distribution of npde :
##   nb of obs: 300
##   mean= -0.004501  (SE= 0.055 )
##   variance= 0.9146  (SE= 0.075 )
##   skewness= -0.04748
##   kurtosis= -0.04797
## -----
## Statistical tests (adjusted p-values):
##   t-test      : 1
##   Fisher variance test : 0.884
##   SW test of normality : 1
##   Global test      : 0.884
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## -----
## 
## Distribution of npde :
##   nb of obs: 300
##   mean= 0.00204  (SE= 0.055 )
##   variance= 0.9137  (SE= 0.075 )
##   skewness= 0.08765
##   kurtosis= 0.2513
## -----
## Statistical tests (adjusted p-values):
##   t-test      : 1
##   Fisher variance test : 0.868
##   SW test of normality : 1
##   Global test      : 0.868
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## -----
## 
## Distribution of npde :
##   nb of obs: 300
##   mean= -0.03544  (SE= 0.055 )
##   variance= 0.9018  (SE= 0.074 )
##   skewness= 0.04896
##   kurtosis= -0.05554
## -----
## Statistical tests (adjusted p-values):
##   t-test      : 1
##   Fisher variance test : 0.674
##   SW test of normality : 1
##   Global test      : 0.674
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## -----

```

```

## -----
## Distribution of npde :
##   nb of obs: 169
##   mean= 0.1433  (SE= 0.07 )
##   variance= 0.8186  (SE= 0.089 )
##   skewness= -0.03812
##   kurtosis= -0.3733
## -----
## Statistical tests (adjusted p-values):
##   t-test : 0.123
##   Fisher variance test : 0.247
##   SW test of normality : 1
##   Global test : 0.123
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## ----

## -----
## Distribution of npde :
##   nb of obs: 300
##   mean= 0.03101  (SE= 0.057 )
##   variance= 0.9715  (SE= 0.079 )
##   skewness= -0.006498
##   kurtosis= 0.8122
## -----
## Statistical tests (adjusted p-values):
##   t-test : 1
##   Fisher variance test : 1
##   SW test of normality : 0.00364 **
##   Global test : 0.00364 **
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## ----

## -----
## Distribution of npde :
##   nb of obs: 300
##   mean= 0.03058  (SE= 0.062 )
##   variance= 1.164  (SE= 0.095 )
##   skewness= 0.04433
##   kurtosis= -0.05092
## -----
## Statistical tests (adjusted p-values):
##   t-test : 1
##   Fisher variance test : 0.162
##   SW test of normality : 1
##   Global test : 0.162
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## -----

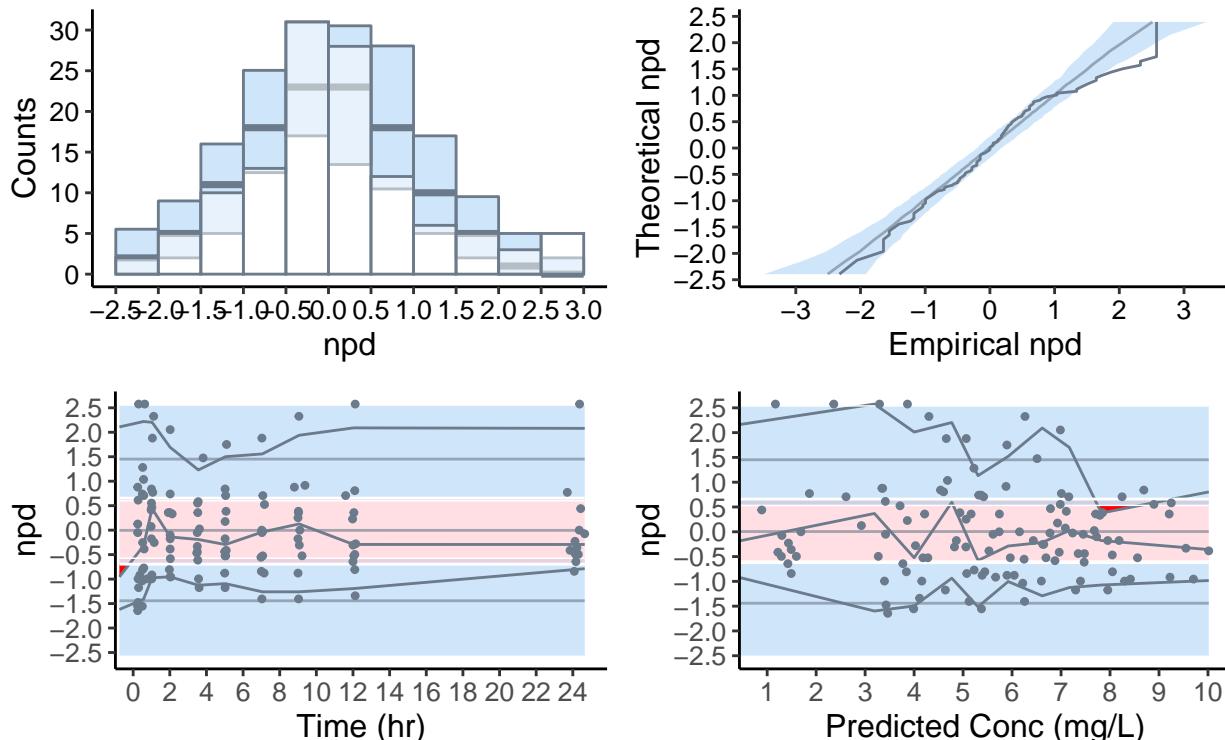
```

wcov devrait marcher (avec les covariables données en numéro de colonne et pas avec leurs noms)

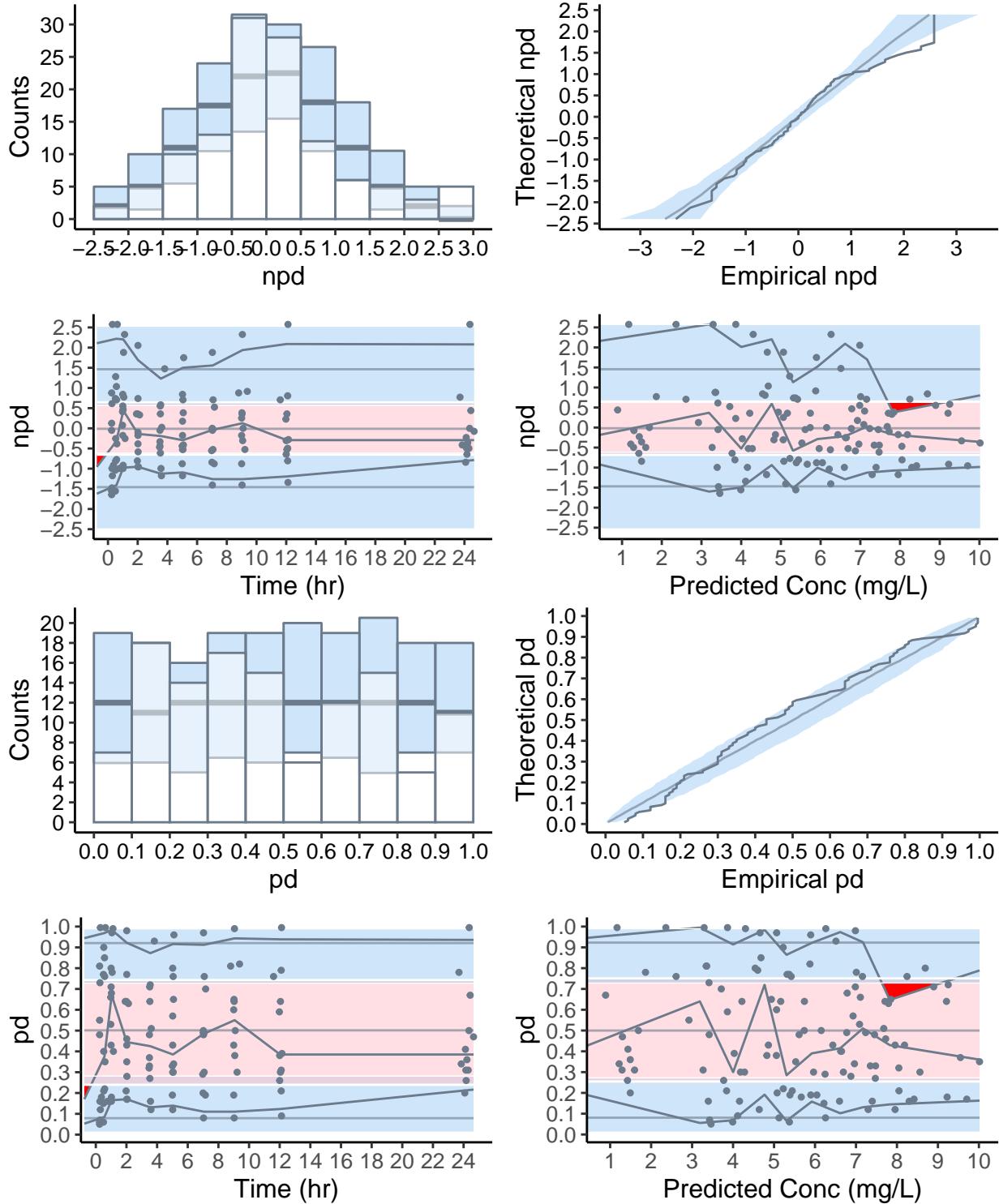
Test plots - default options

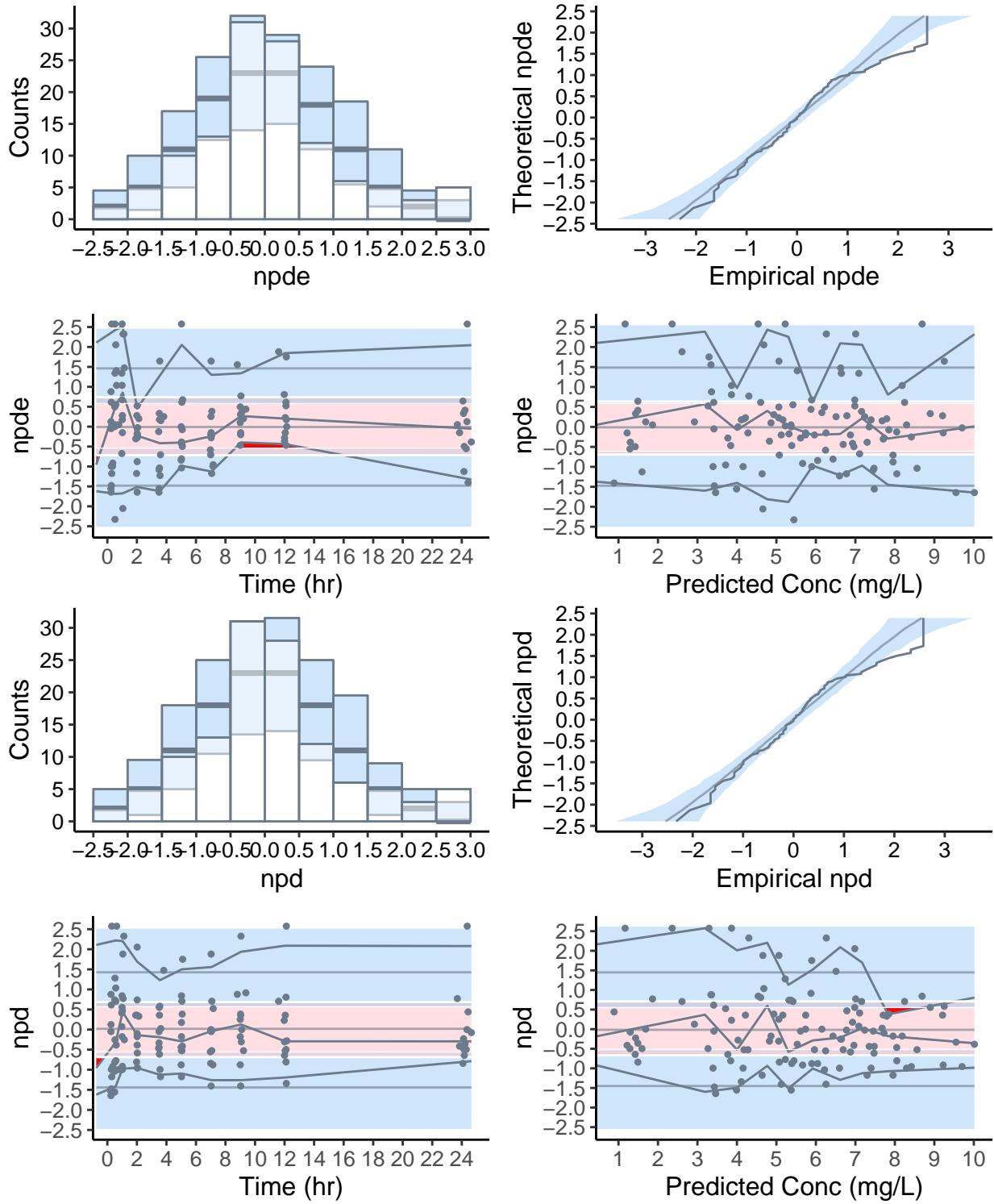
Default plots for theophylline data

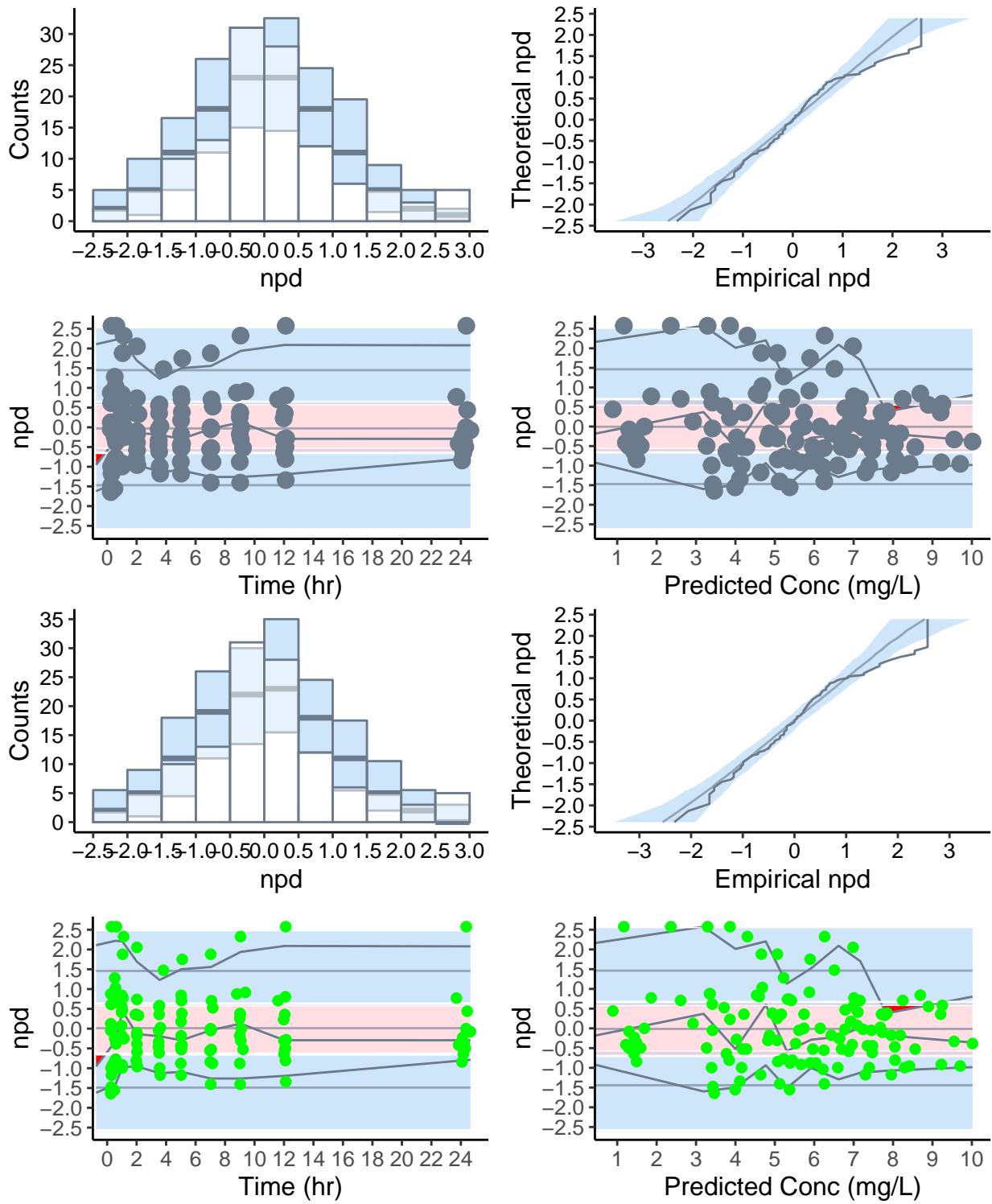
- **Change**
 - npd now default for diagnostic graphs (npde still default for tests)
 - **TODO** change output for test function
- **BUGS Romain TODOO**
 - **remain**
 - **solved**
 - * **title**
 - ajout de la condition plot.opt\$plot.default==FALSE) pour l'affichage des titres dans les plot
 - * axes in the histogram plot => add
 - **Eco check**
 - **solved X-range on PI** extend prediction band to min/max X (it seems to stop at the center of the bin) => option ???
 - **Note** :dans plotNpde-auxScatterPlot: 1er bin avec valeur minimal de la borne et dernier bin avec valeur maximal de la borne. Les plots sur me 1er et dernier bin ne s'arrêtent pas au milieu du bin.
 - **global options** not passing on to the functions
 - * **solved size**
 - **solved** normalement si l'utilisateur spécifie "size=2" sans spécifier size.pobs, size.pobs devrait passer à 1.5, et là ça ne semble rien changer
 - **solved** ni dans le graphe par défaut (ie les 4 graphes), ni individuellement
 - * **solved colours**
 - **solved** when only col is specified, it should affect other elements (but not sure which we had decided), here only affects histogram and col.lobs for qqplot
 - **solved col => col.pobs, col.lobs, col.pcens ?**
 - **solved col.lobs** ne change rien, devrait changer les lignes correspondant aux percentiles et les lignes dans l'histogramme

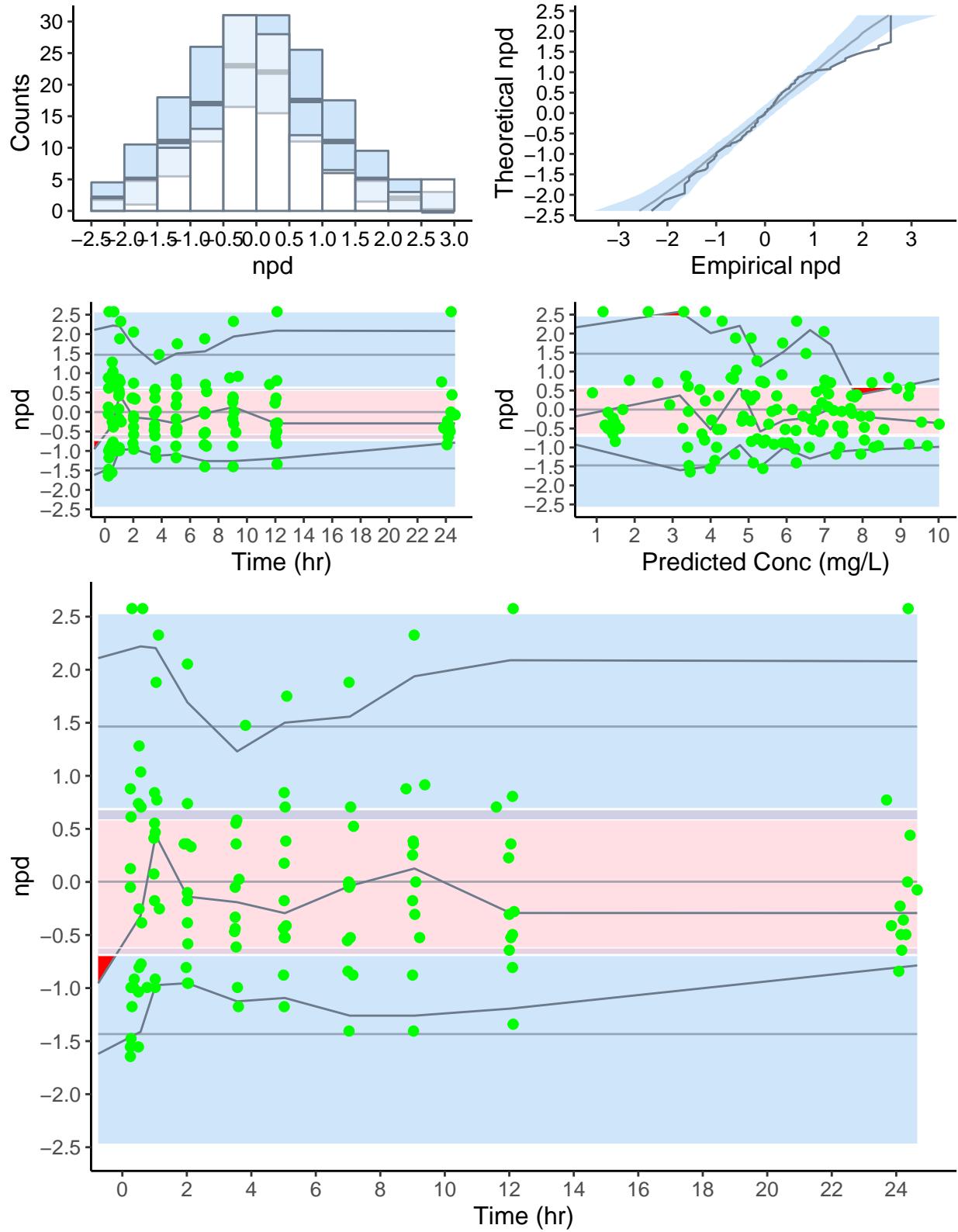


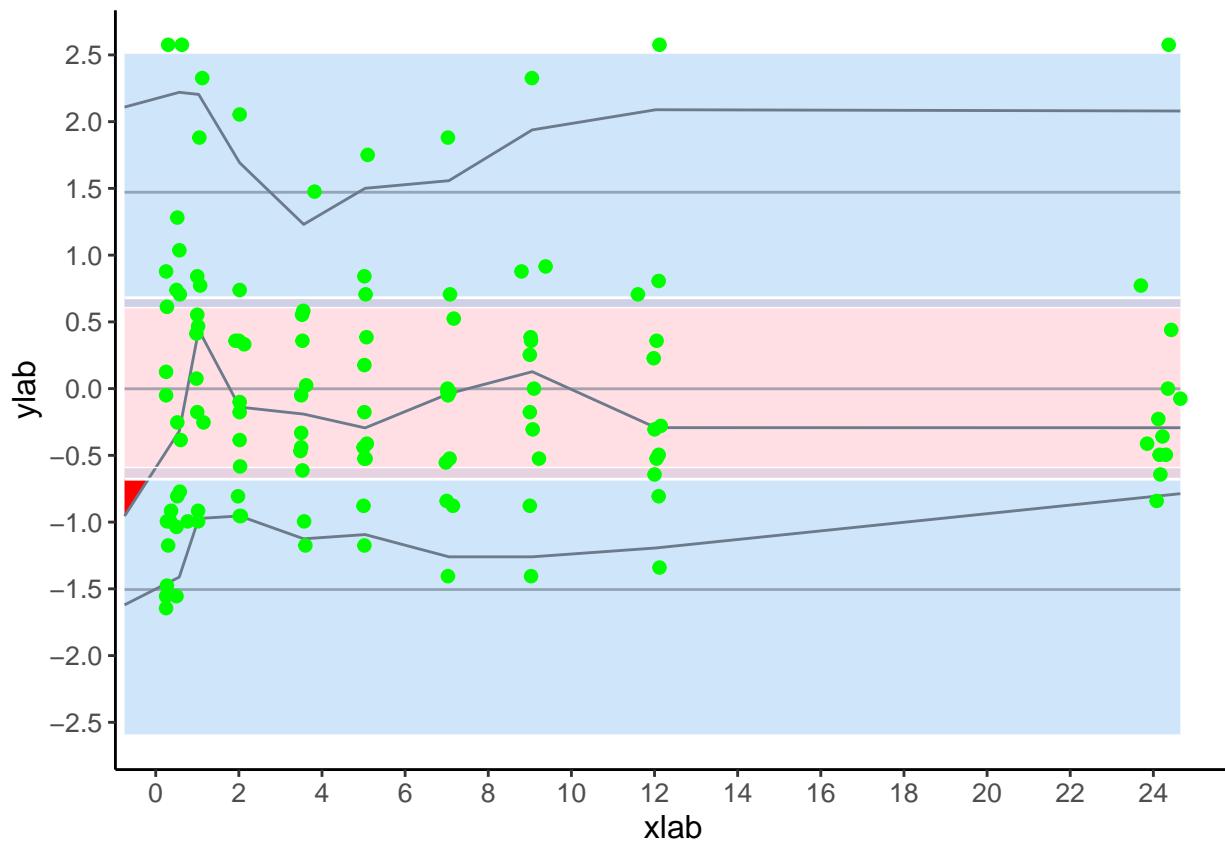
Gof plots for theophylline







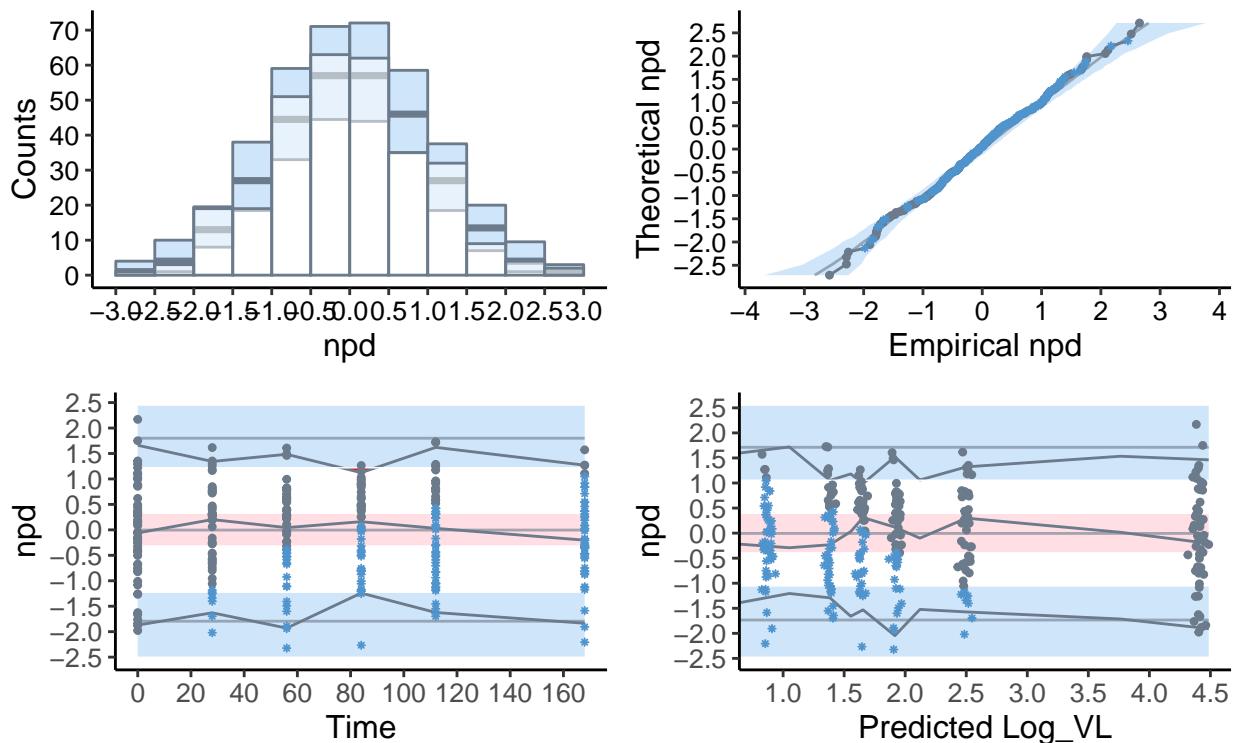
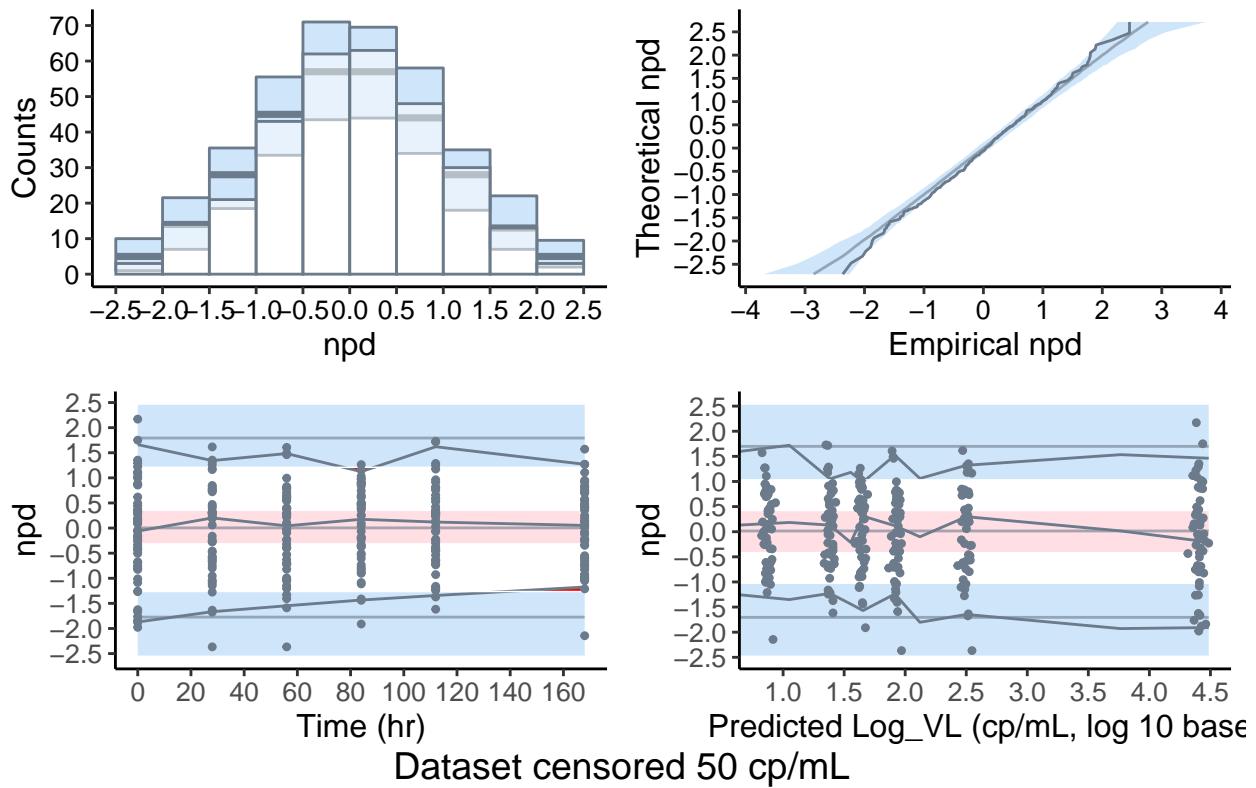




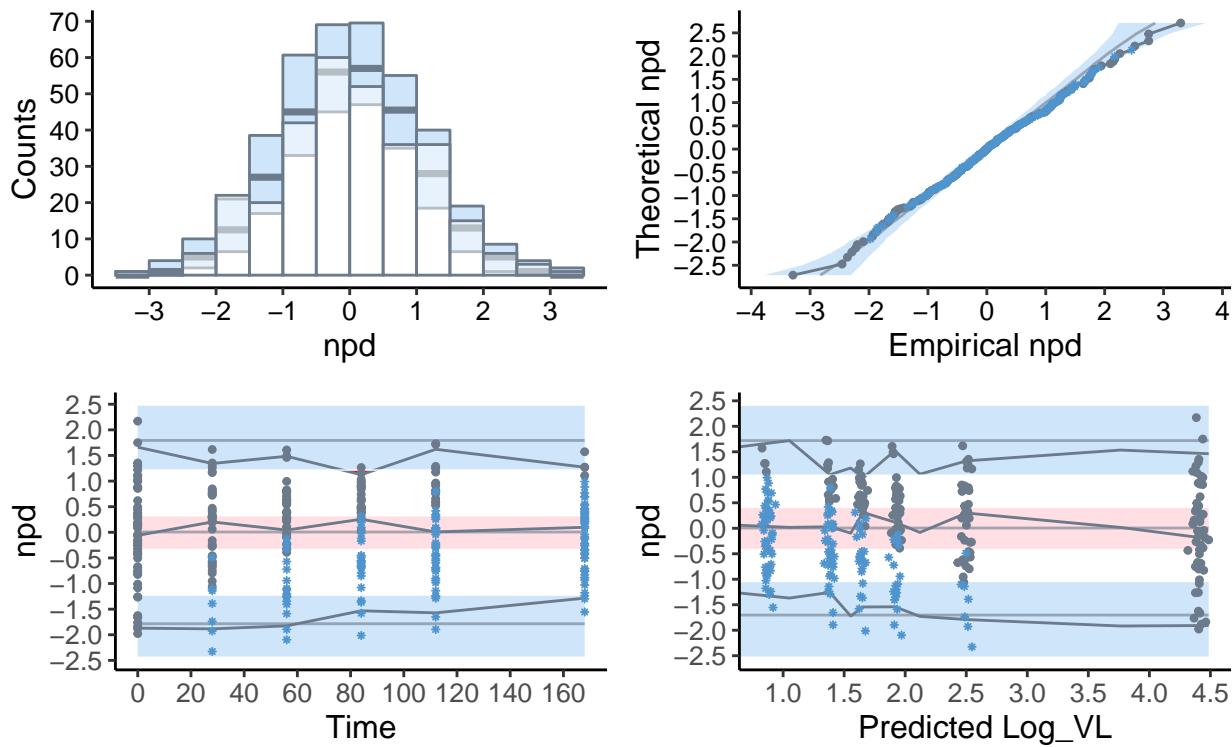
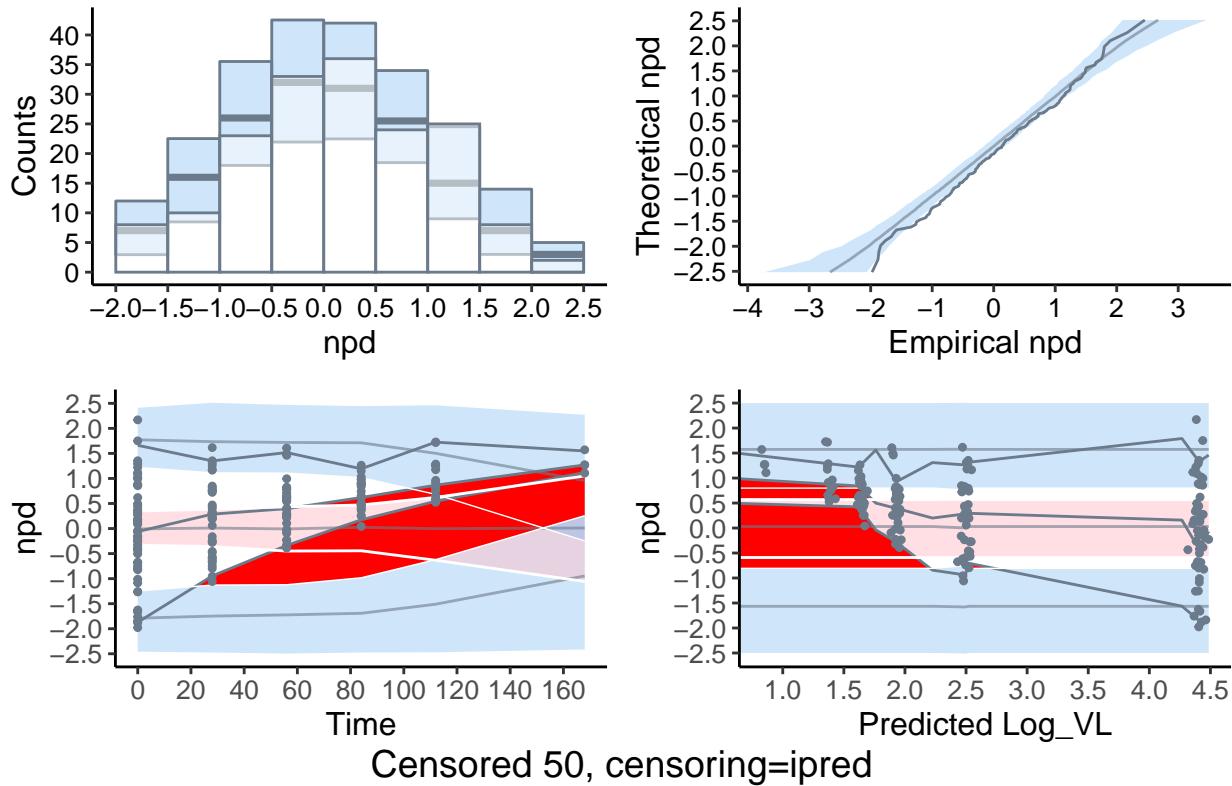
Default plots for viral load data

- computation of the PI: seems appropriate in the different cases
- **Problems (Eco)**
 - PI for omit aren't consistent with the VPC later on
 - * **documentation** explain how PI are computed in the omit case

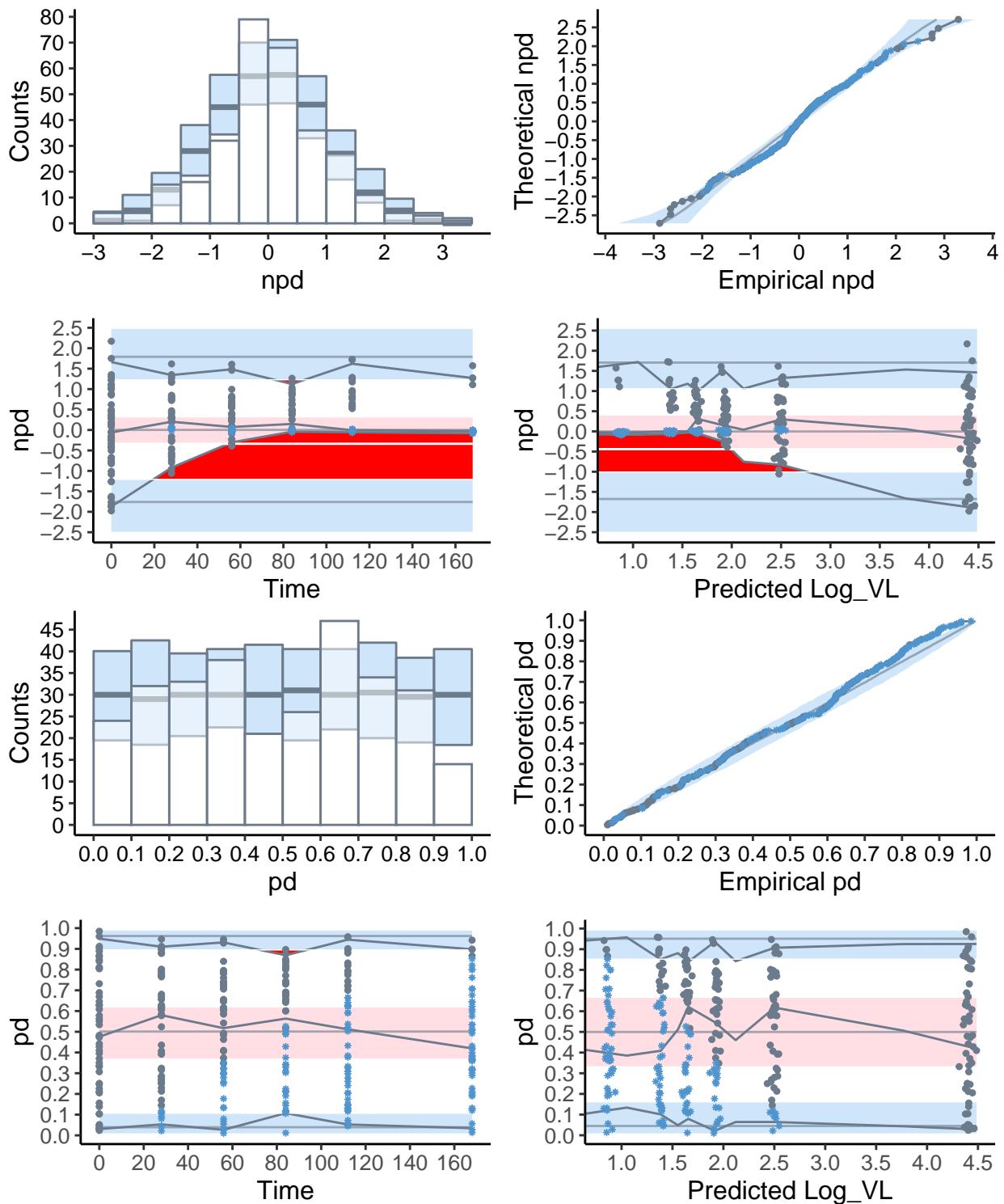
Complete dataset

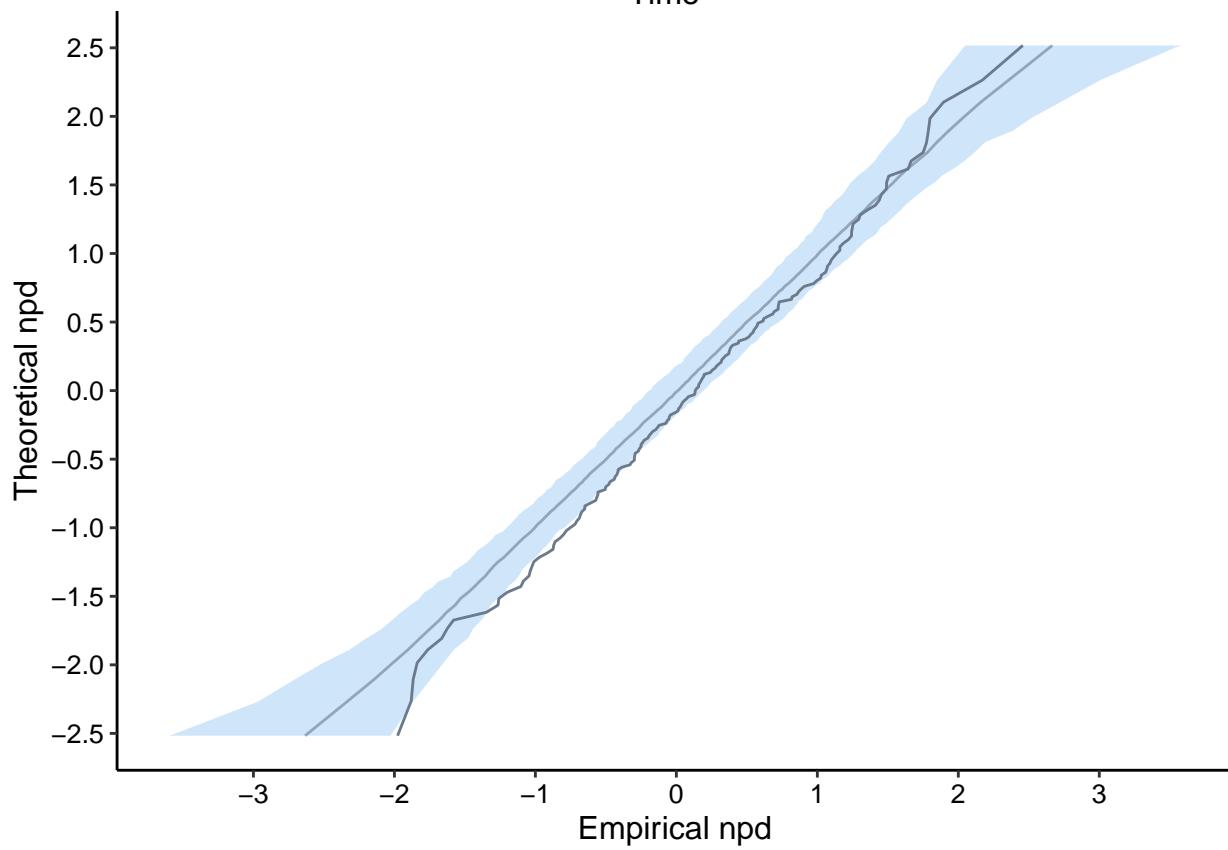
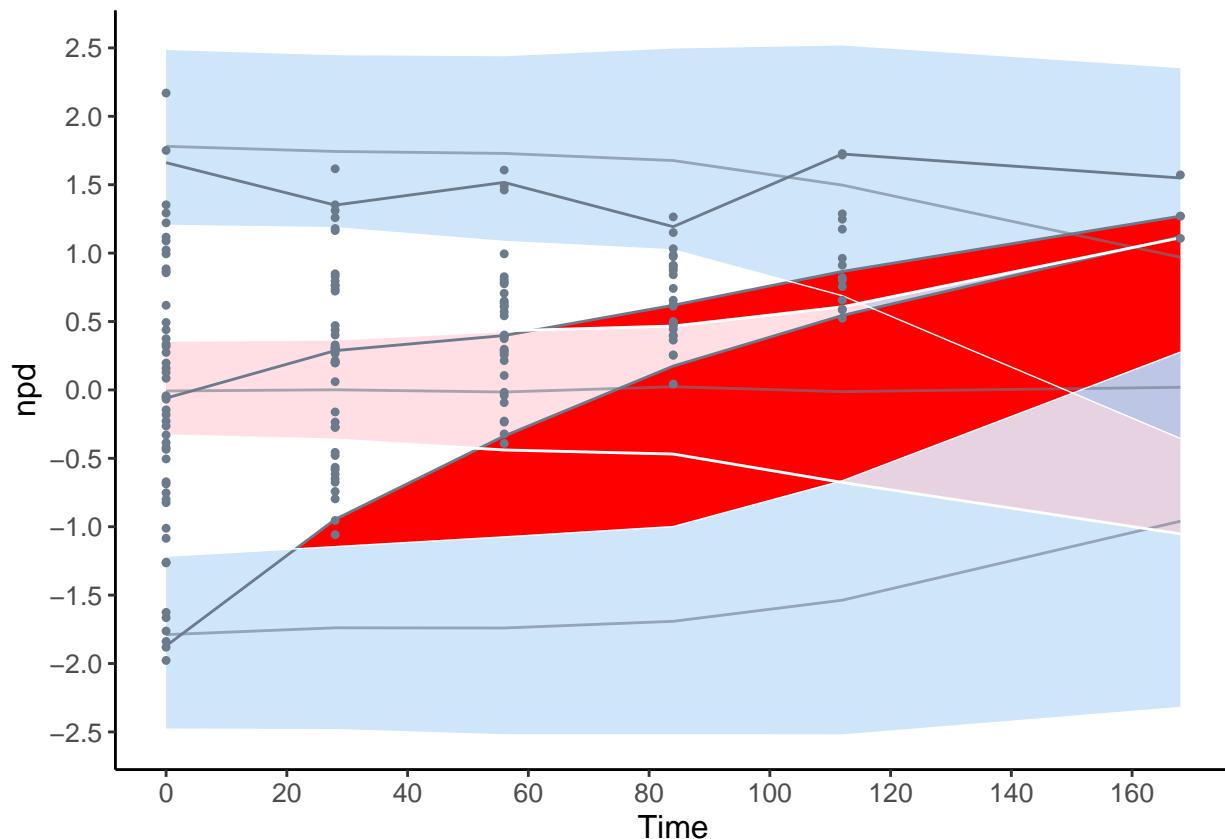


Censored 50, censoring=omit



Censored 50, censoring=ppred

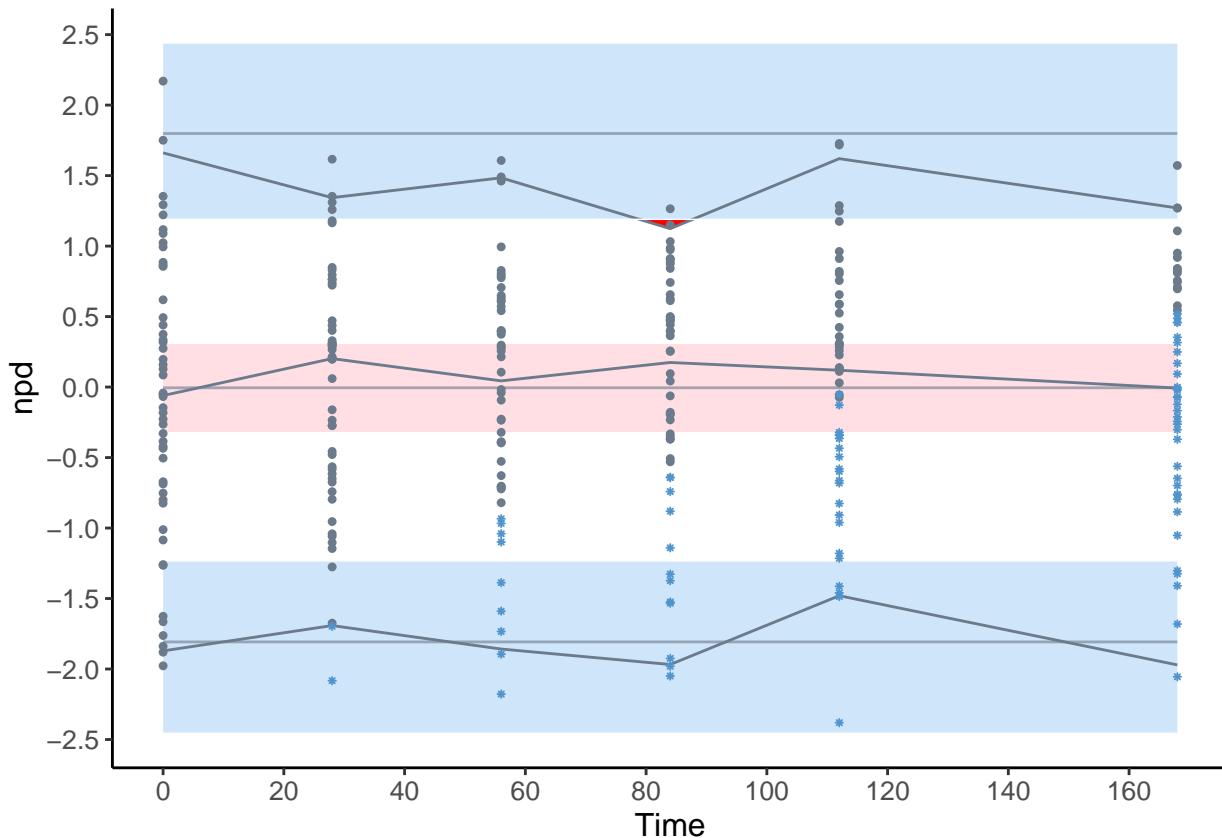


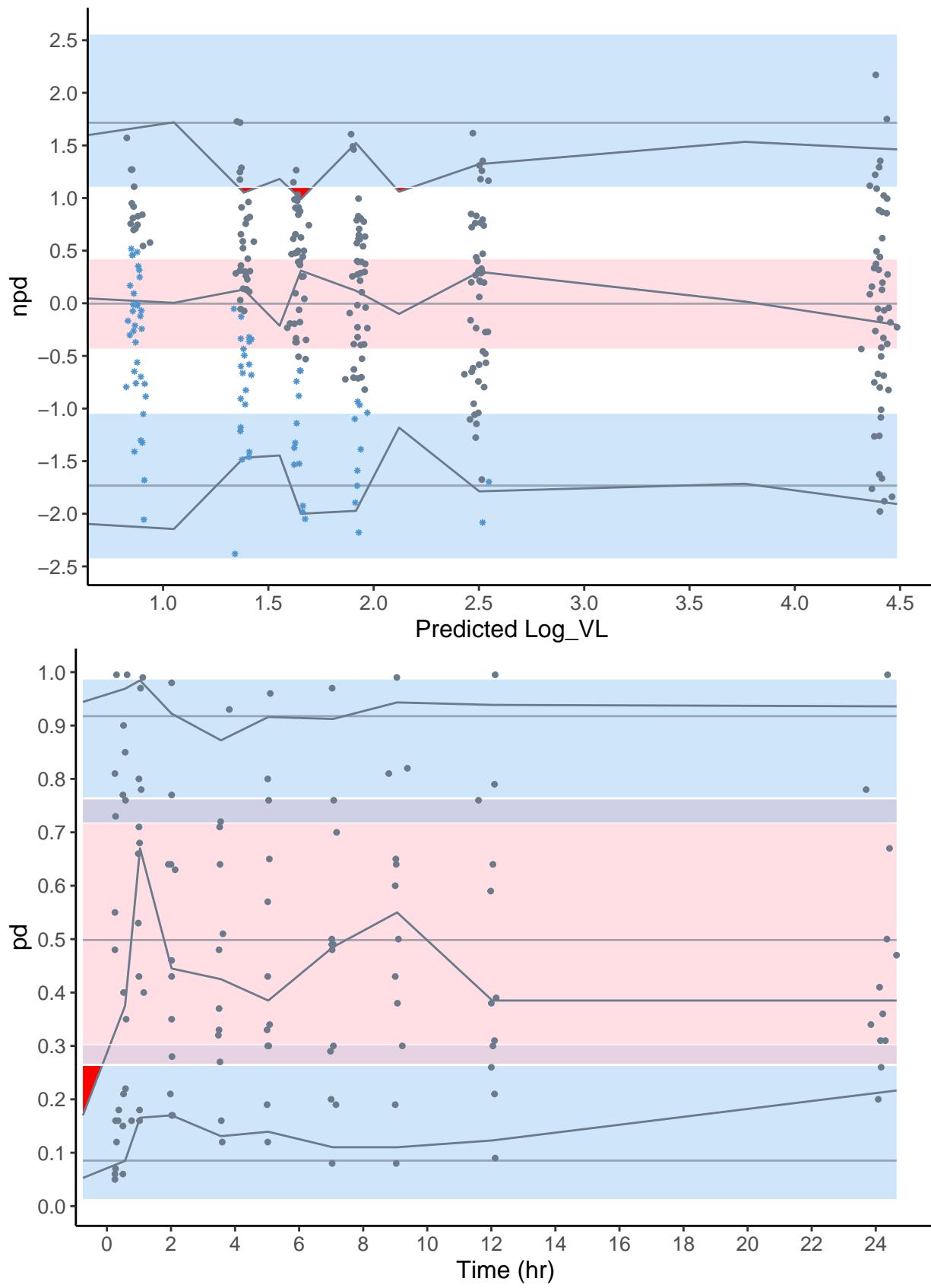


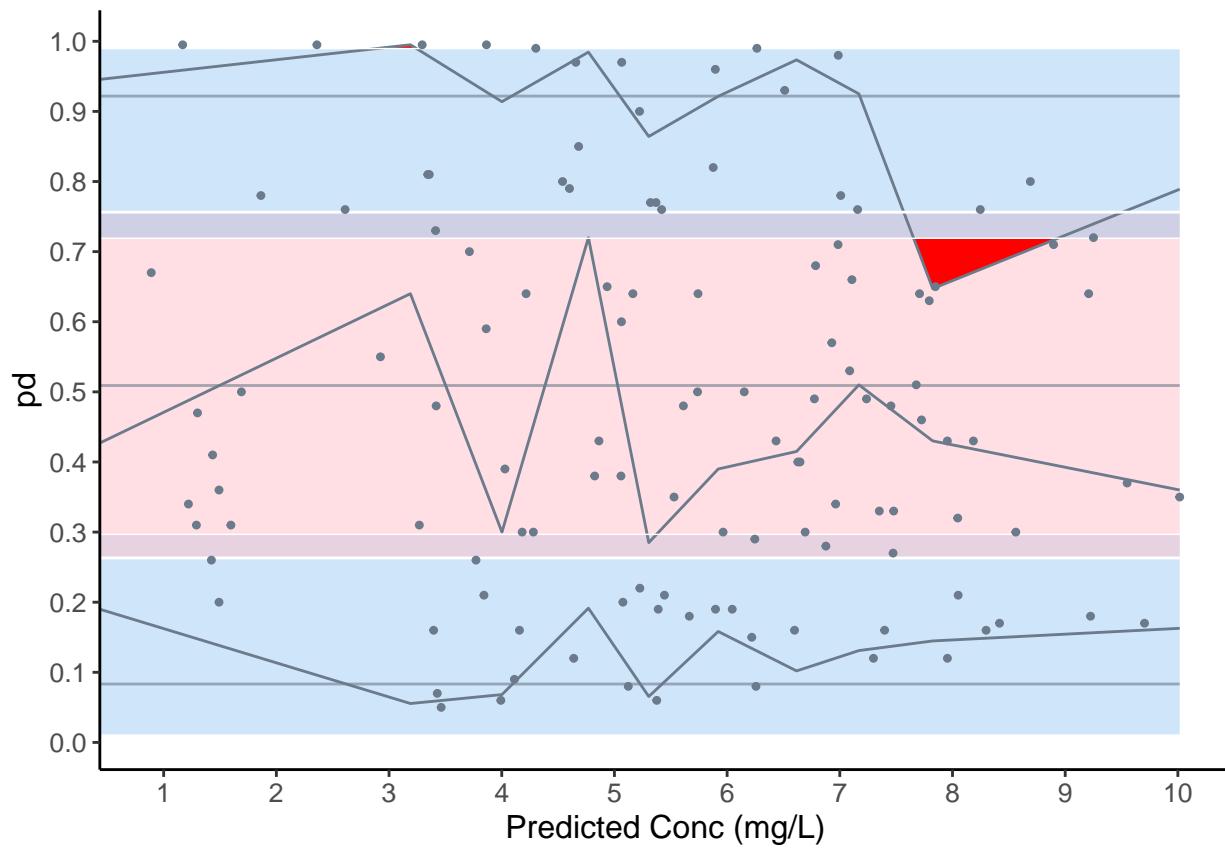
Plots with default options

Scatterplots

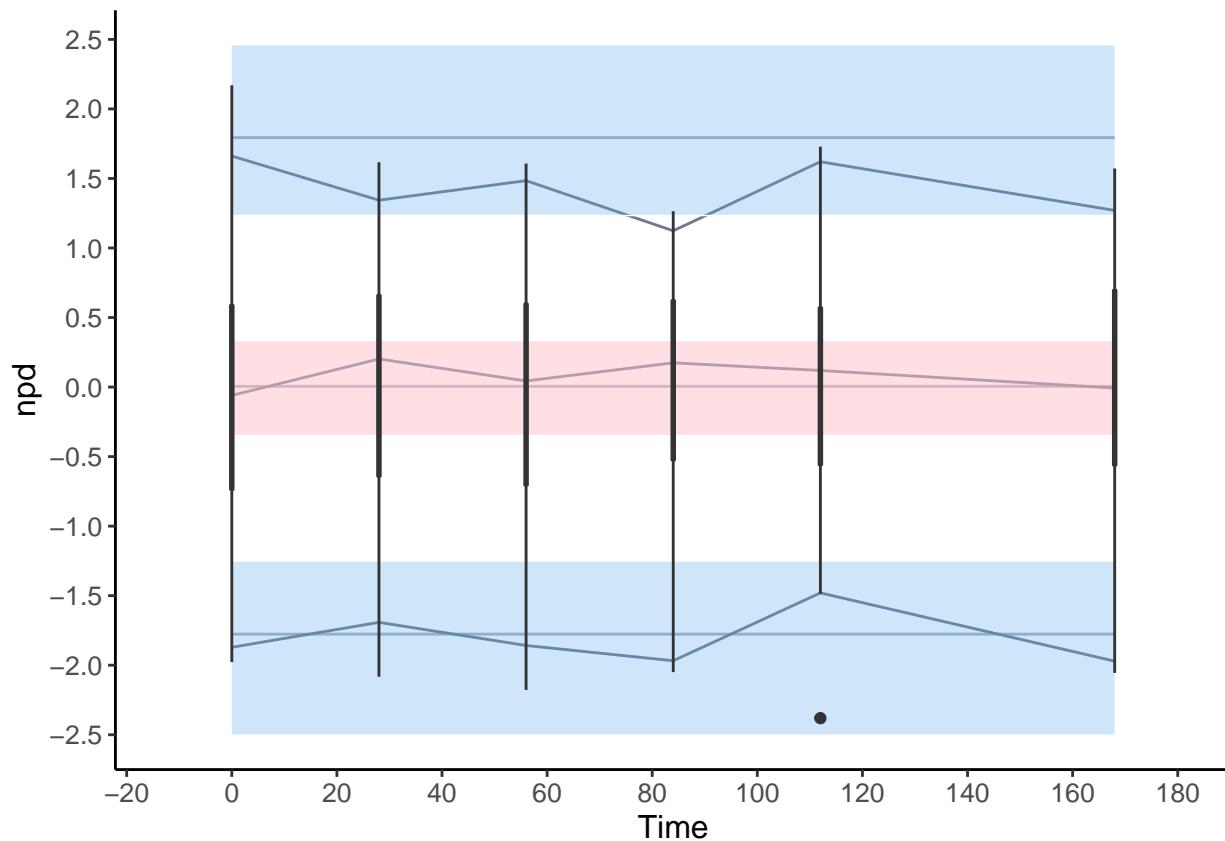
- **Problems (à résoudre Romain)**
 - solved lab for scatterplot doesn't appear (but works in waffle plot, wtf ?)
 - solved size ne passe pas (devrait changer à la fois size.pobs et size.pcens)
 - * solved même pb que dans le graphe par défaut (ie les 4 graphes)
 - solved calling plot with plot.type="x.scatter" and which="pd" (eg) triggers the message "Option which.y= x.scatter not recognised" => debug call to intermediate functions with arguments in **plotNpde-methods.R** (dispatching arguments given in ...)
 - solved plot.box has a weird aspect with seemingly empty boxes for the first 2 bins ?
- **Check**
 - solved size of boxes when plot.box=TRUE



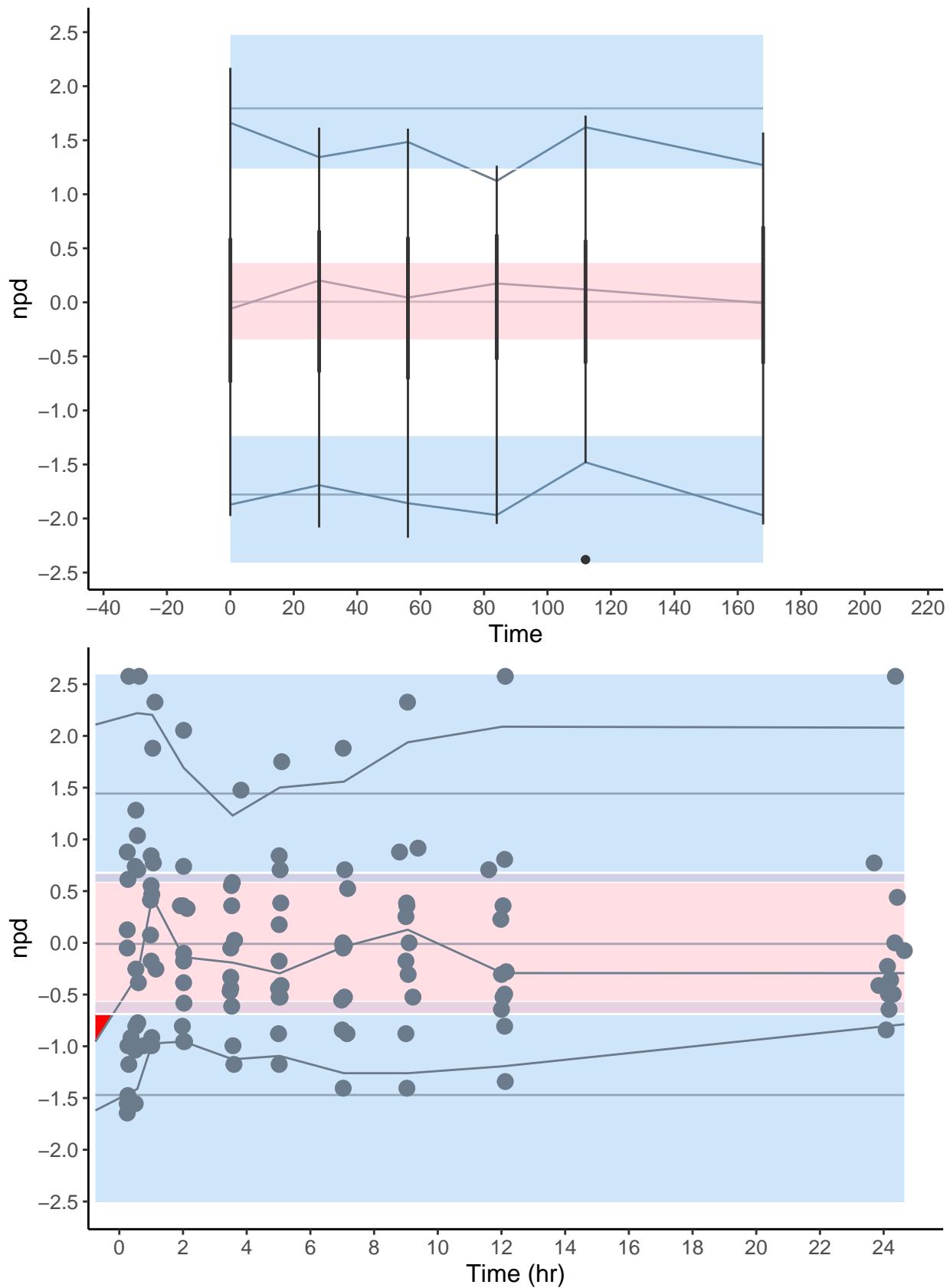


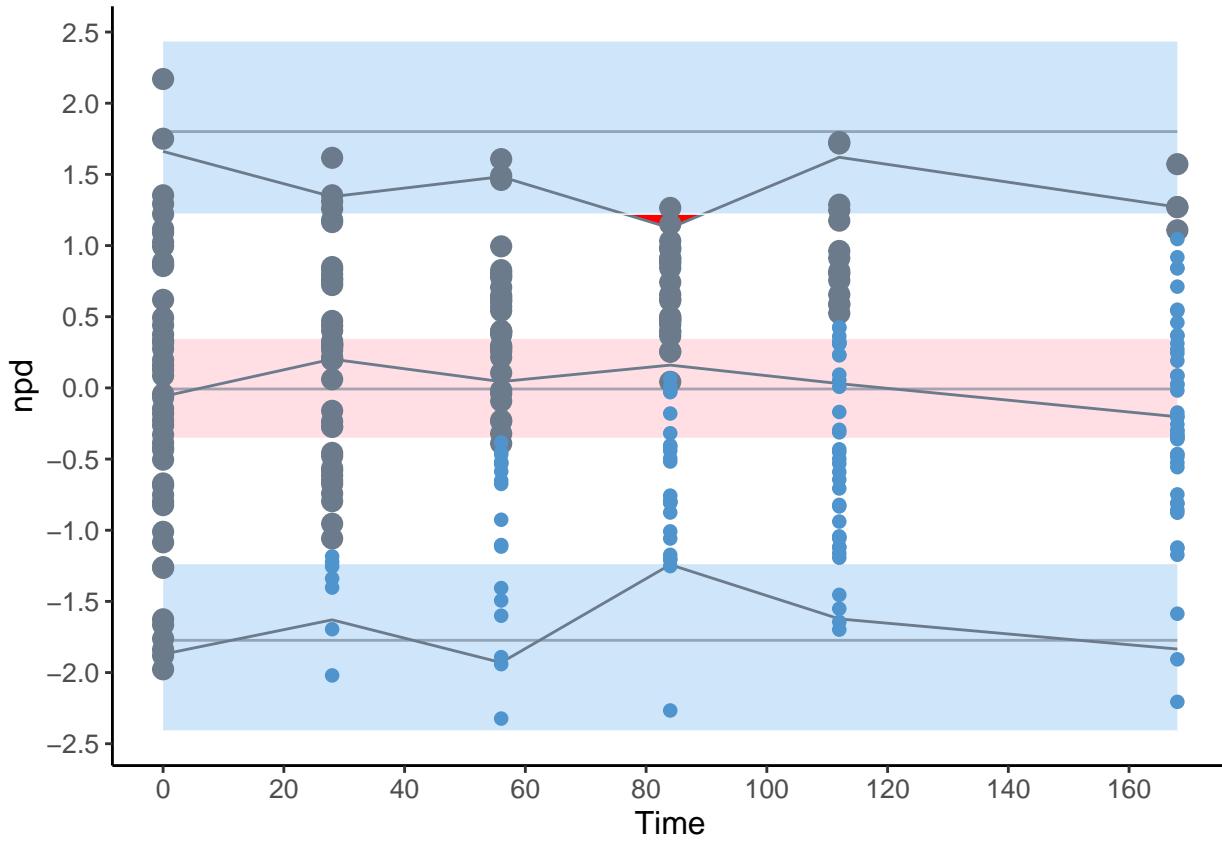


```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```



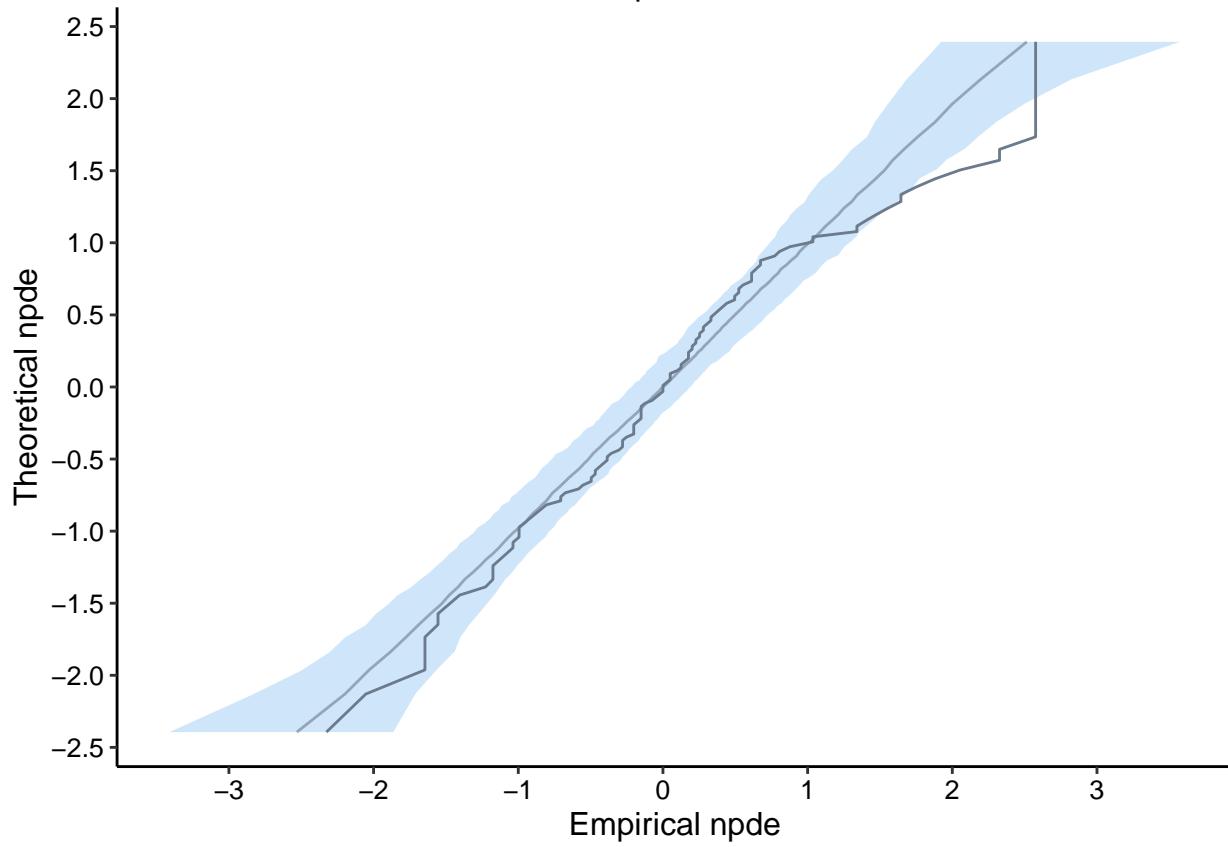
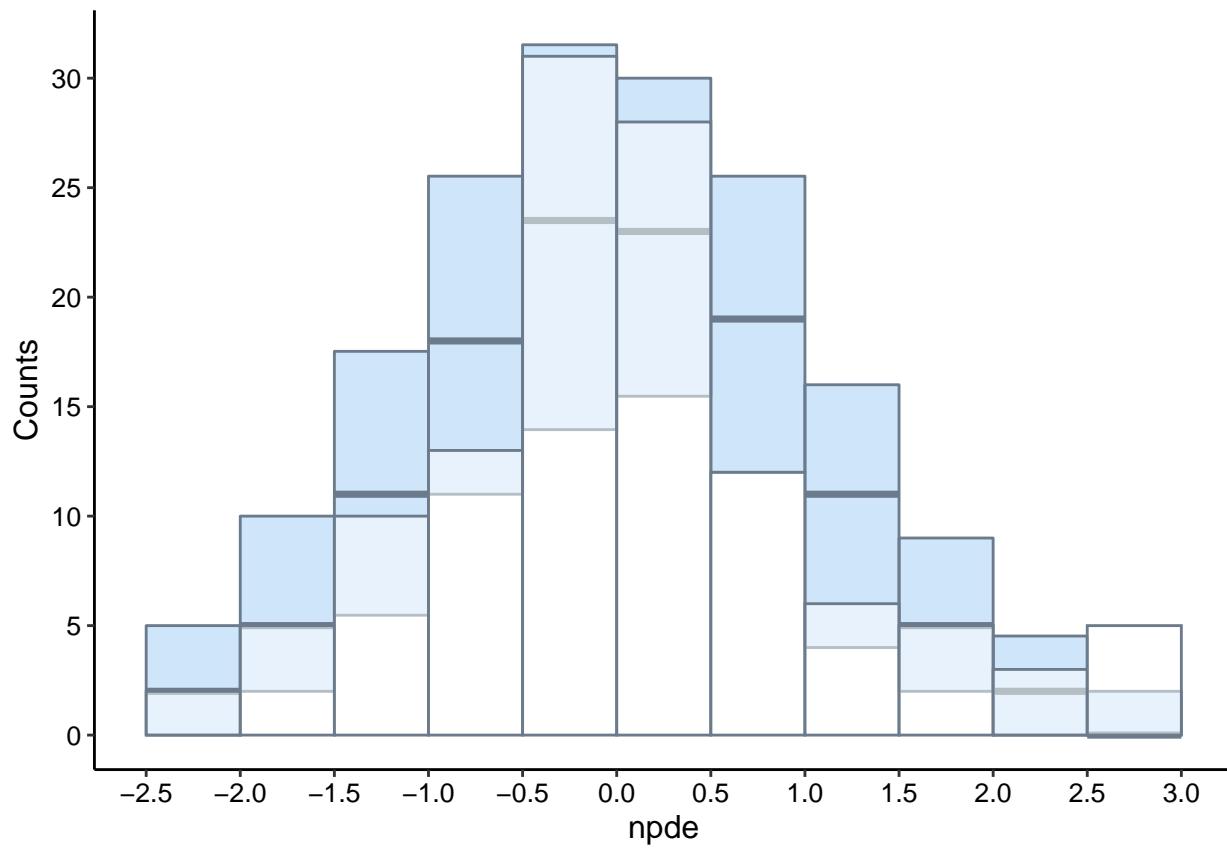
```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```

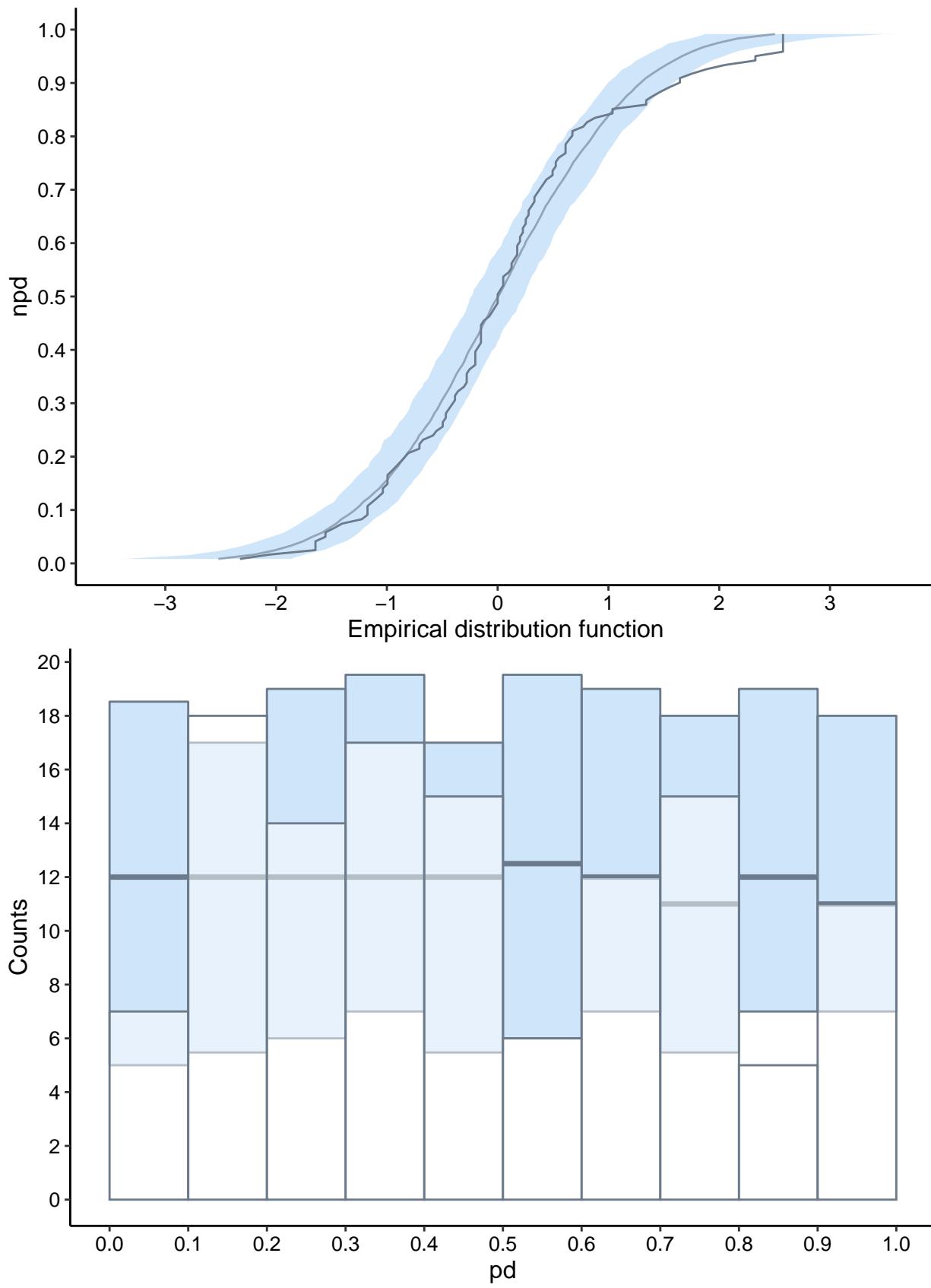


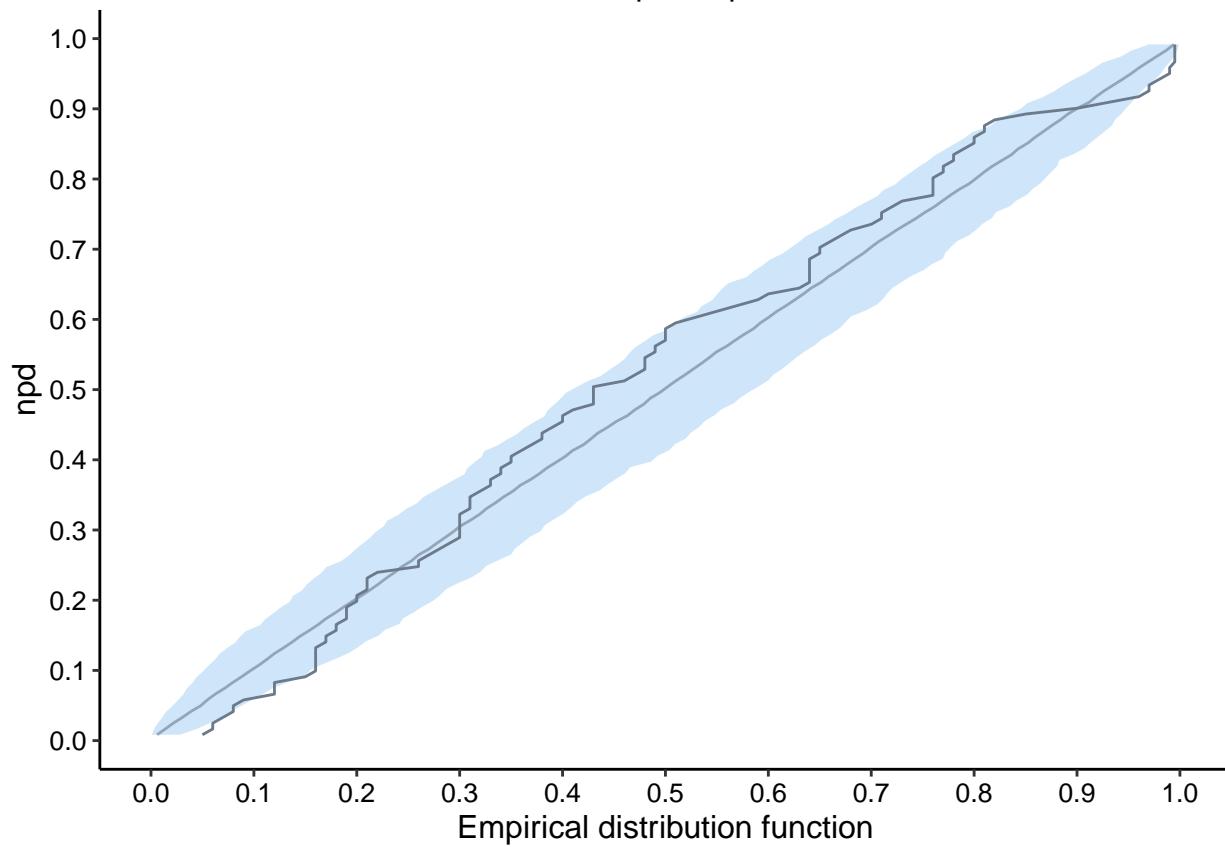
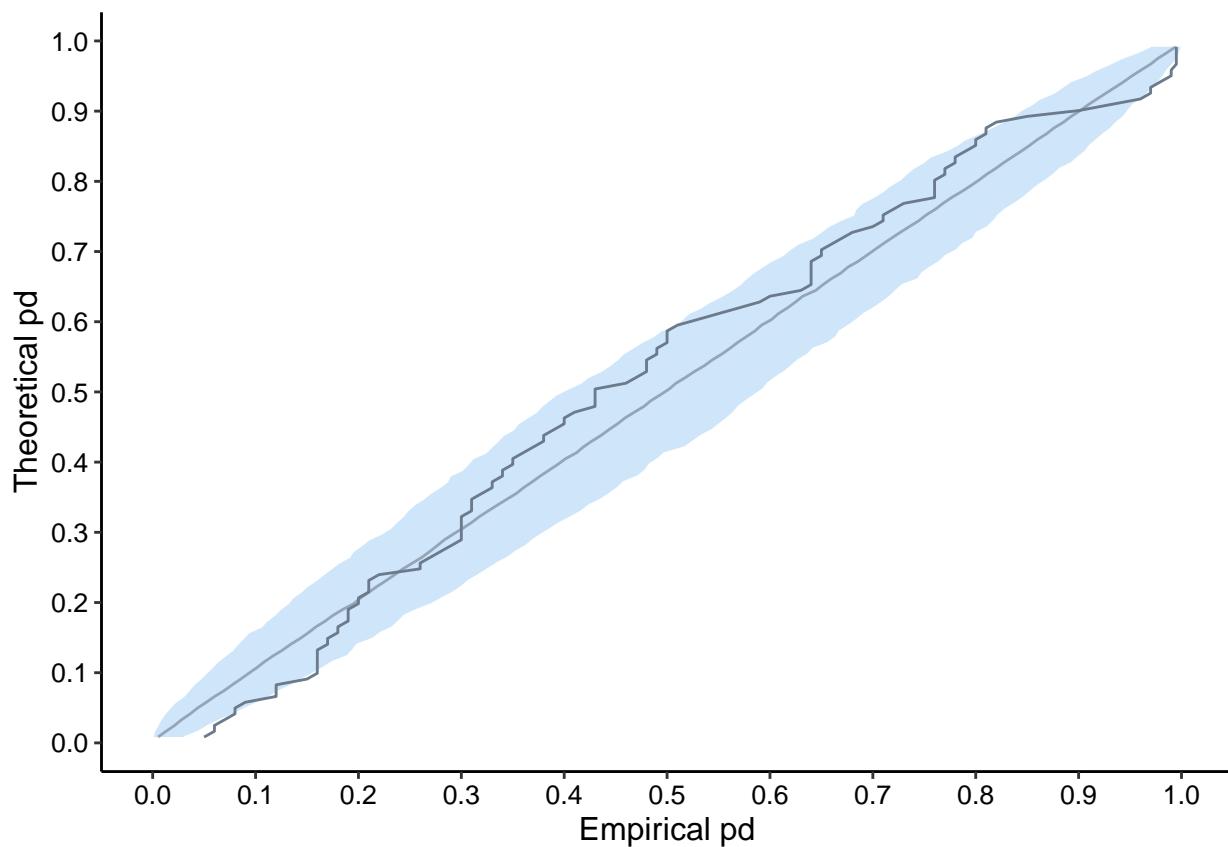


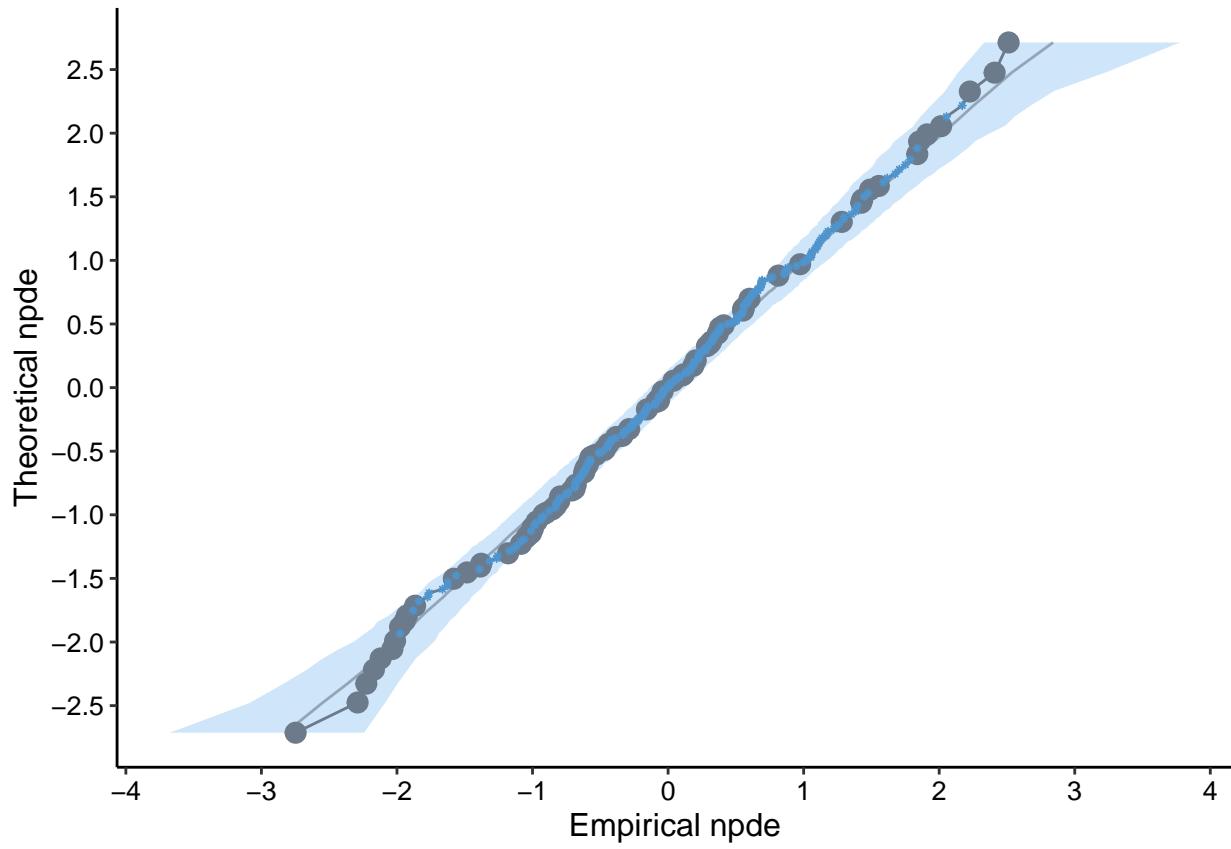
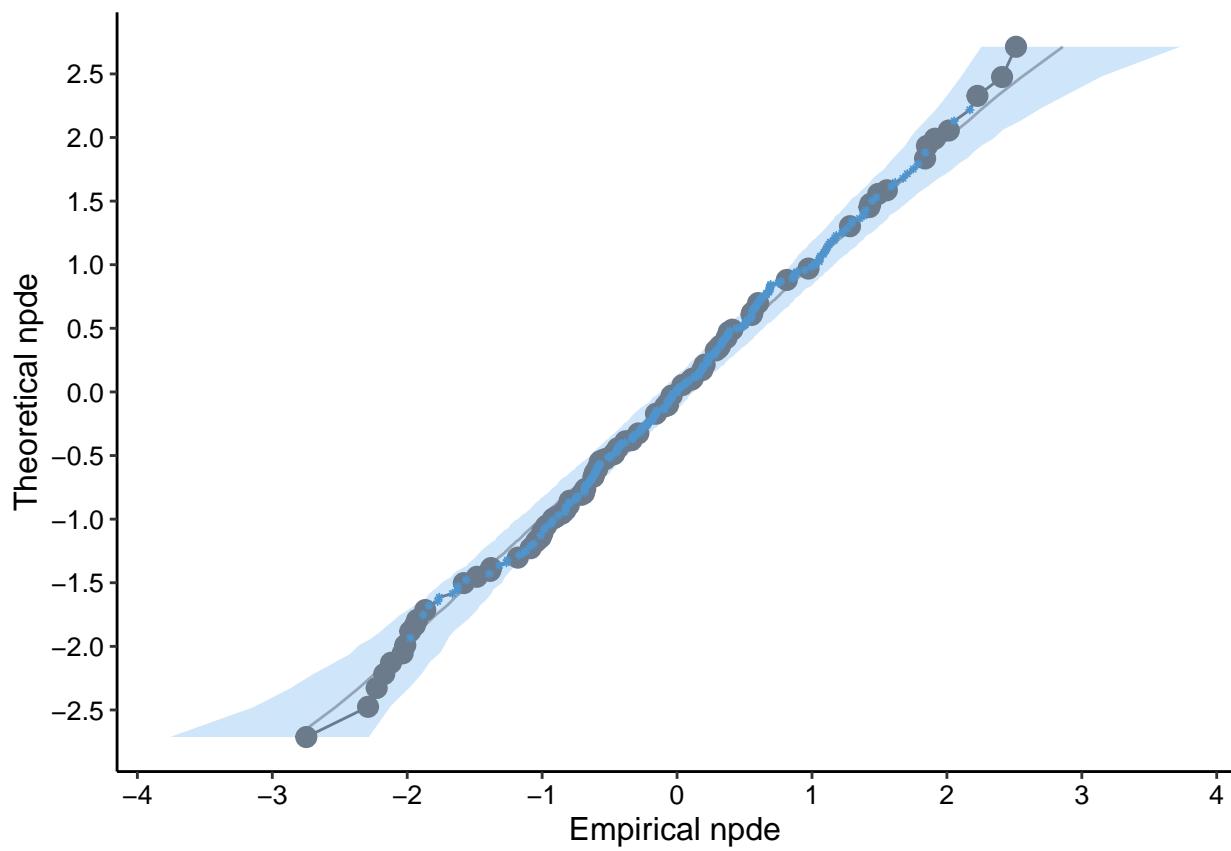
Distribution plots

- **Problems (à résoudre Romain)**
 - solved pb with **size** (comme plus haut, size ne passe pas, size.pobs/size.pcens passent)
- **Problems (à résoudre Eco)**
 - **axe title** for ecdf plot (should be: ‘npde’ on X-axis (/pd/npd) and ‘Empirical distribution function’ on Y-axis) [done, check it works]



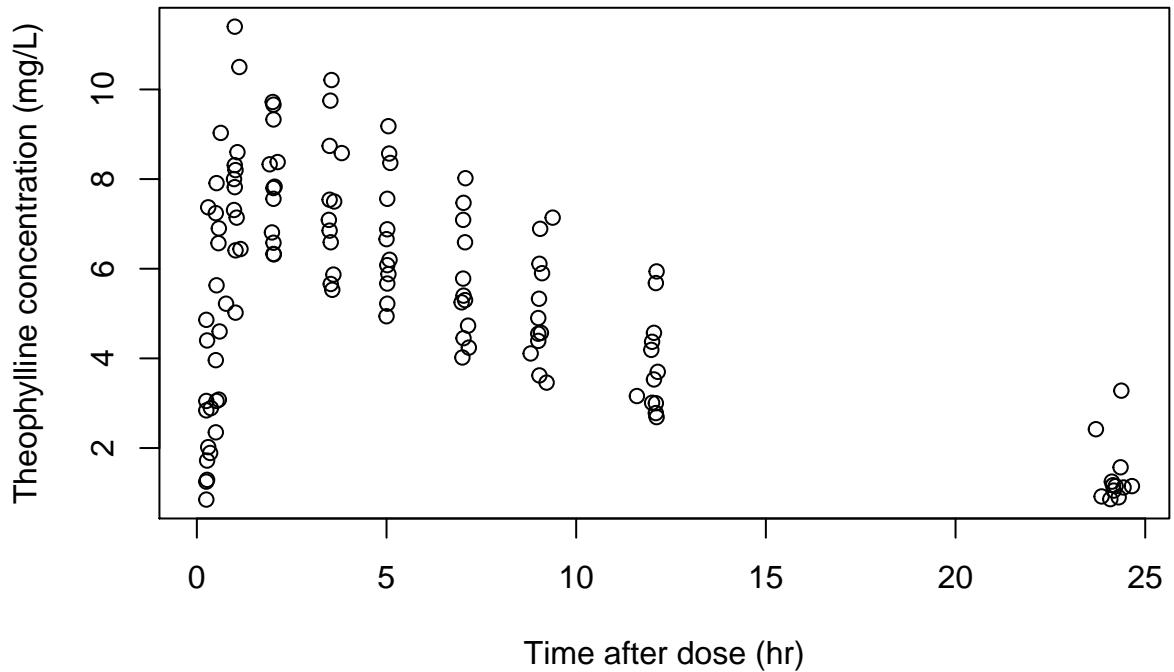




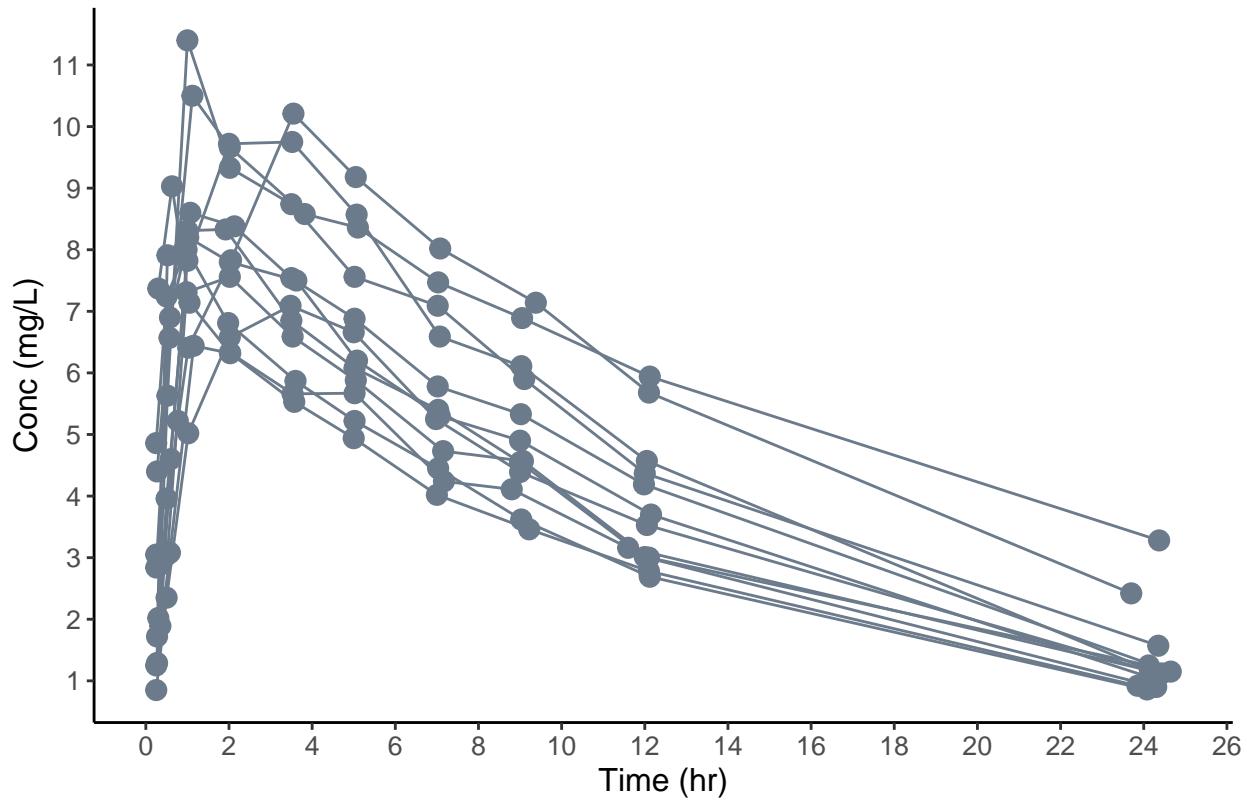


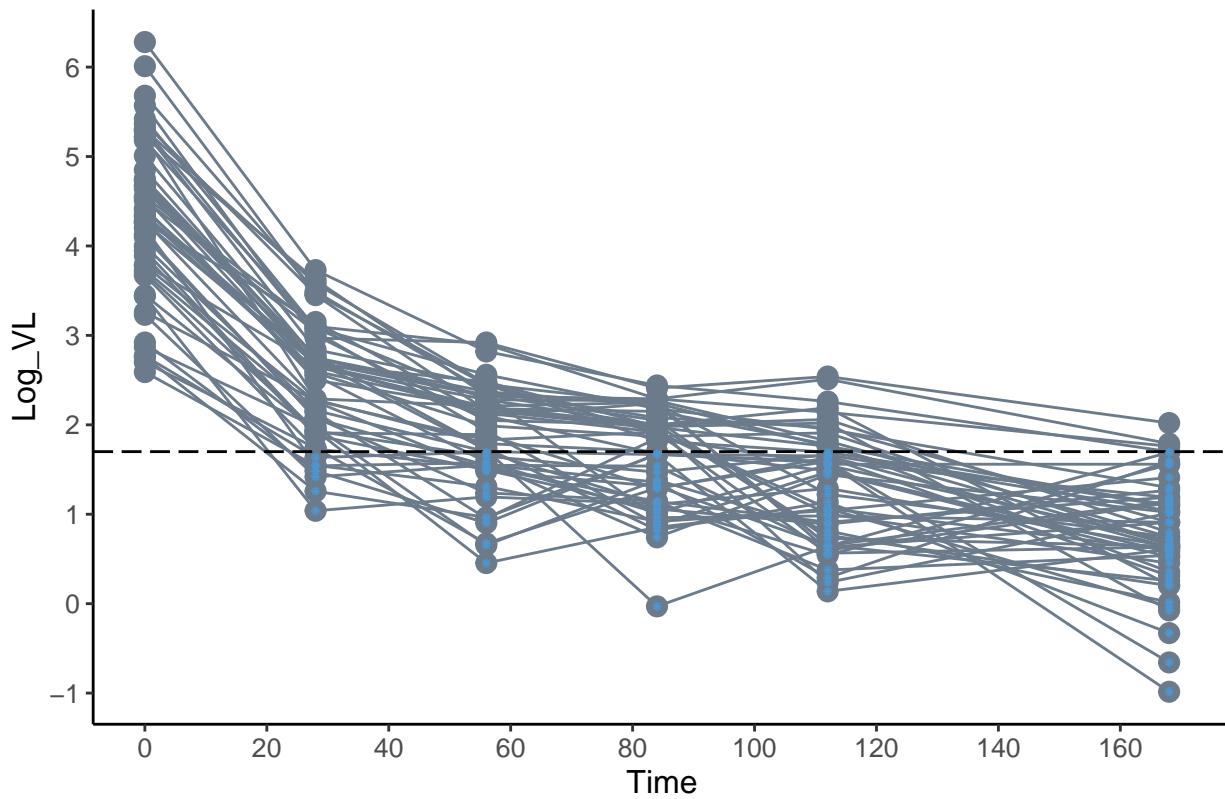
Data plots

- **Problems** (à résoudre Romain)
 - **Note** pour cette version ça fait beaucoup à modifier, à commencer par `plotNpde-methods` puis `npde.plot.data` où le slot `data` est à prendre ne compte dans le reste du code.
 - plot d'un **objet npdeData** ne marche pas
 - * par contre `plot.type="data"` de l'objet `npdeObject` fonctionne => bizarre, vérifier [normalement `plot.type="data"` devrait appeler `plot(objet@data)` mais là ça semble court-circuiter ceci => à rectifier, on veut plutôt l'inverse]
 - * problem with the data in the presence of **LOQ data**, the data being plotted is the LOQ
 - * if applied directly to an objet `npdeData` (eg `plot(yivr50@data)`) => should plot `y` or the LOQ (ie censoring value)
 - * if applied to the **npdeObject** resulting from a run => data plotted should depend on the censoring method (same as VPC)
 - for `cdf`, plot imputed `y`
 - for `omit`, don't plot anything
 - for `ipred`, plot `ipred`
 - for `ppred`, plot `ppred`
 - * all other options should be the same [axis titles, axes, grid, default colours, etc...]
 - **solved lines for axes** don't appear
 - **solved size**
 - * size should be controlled by `size.pobs` and `size.pcens`, currently too small (not the same defaults as for the other plots ?)
 - * size doesn't change `size.pobs` and `size.lobs` as it should
 - * `size.lobs` doesn't work
 - **Note**
 - entre `l.obs`, `p.obs`, etc. qui modifie qui au final ? L'utilisation de templates ggplot serait-elle plus appropriée ? On utilise directement la syntaxe ggplot + on modifie les paramètres directement dans les listes `theme()` et `layers()`. Il y aurait ainsi les templates par défauts + les templates utilisateurs.
 - **solved axis titles**
 - * missing for `theofit`, why ? (present for `yvir50`, so odd)
 - **solved line.loq** doesn't work (no line appears at the LOQ)

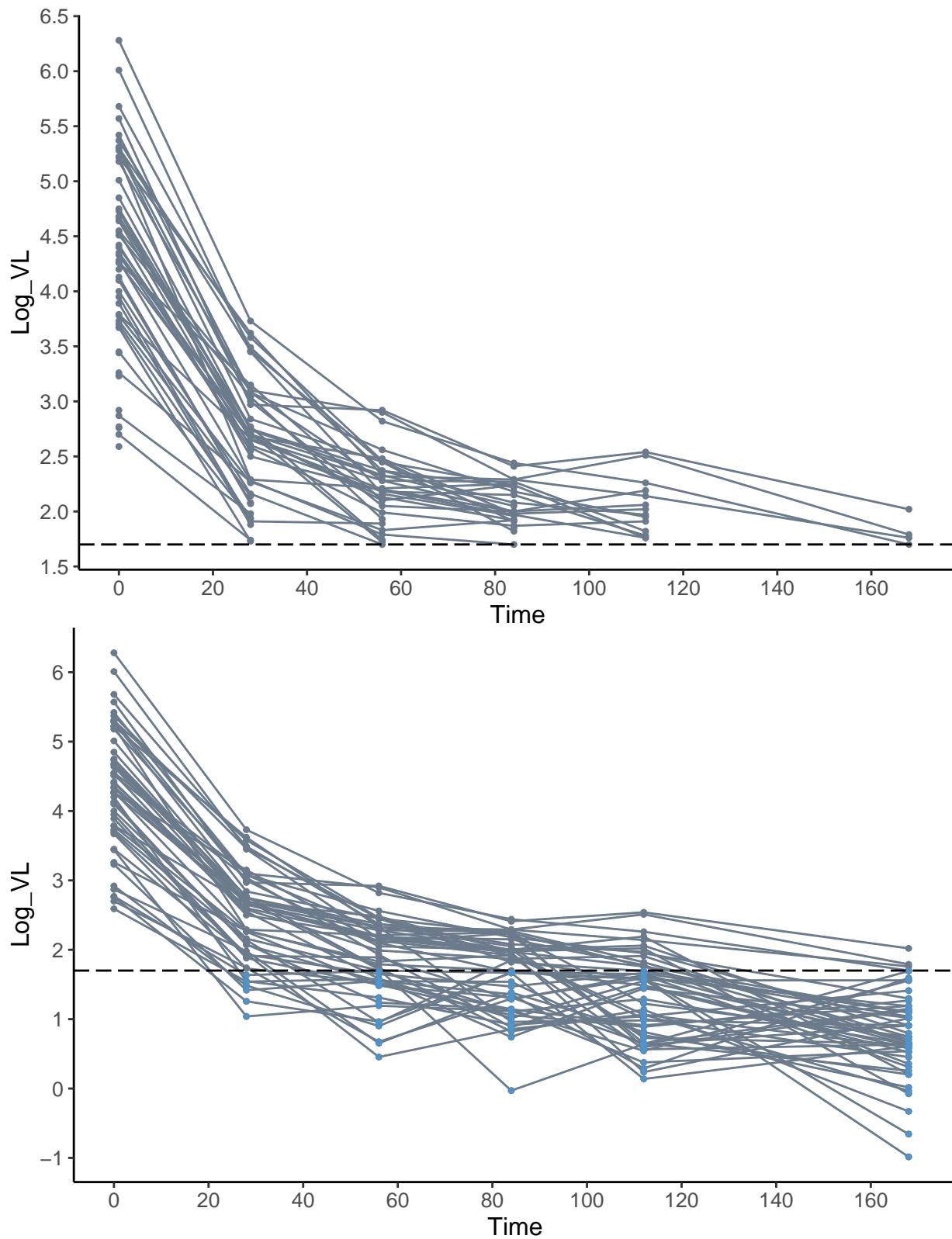


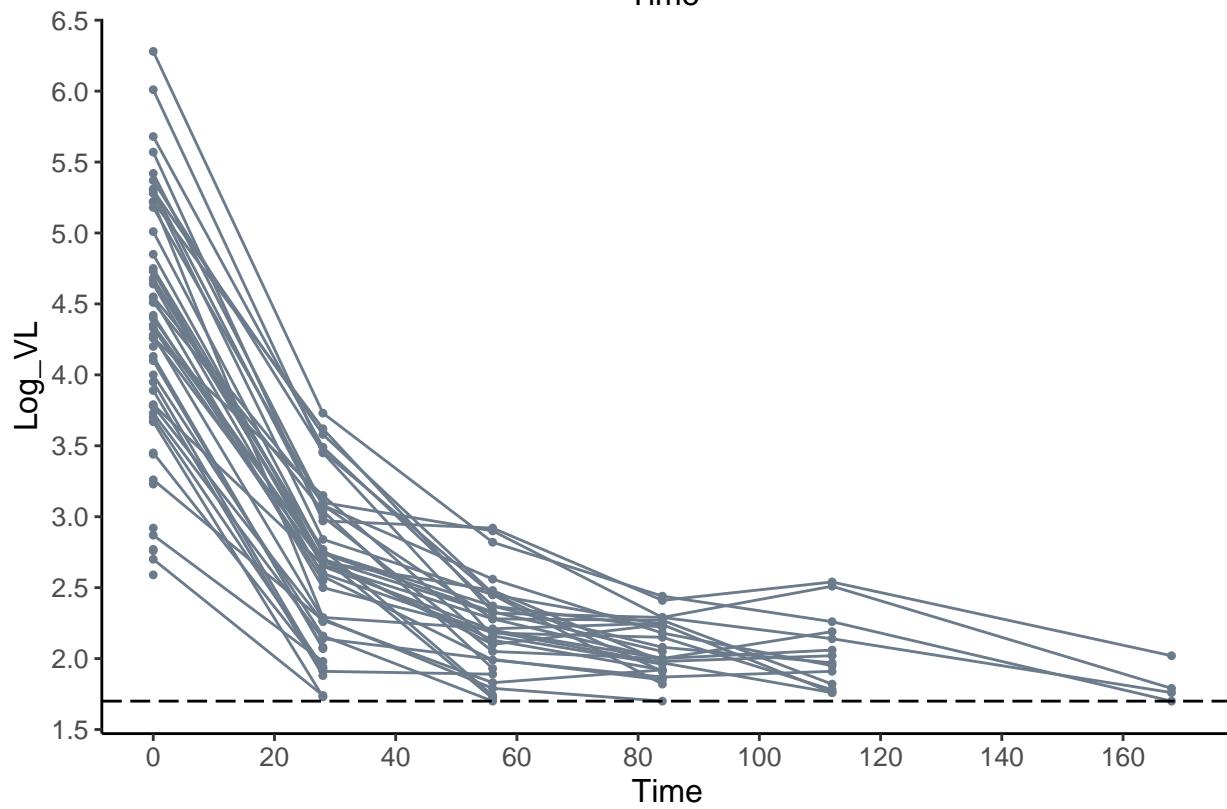
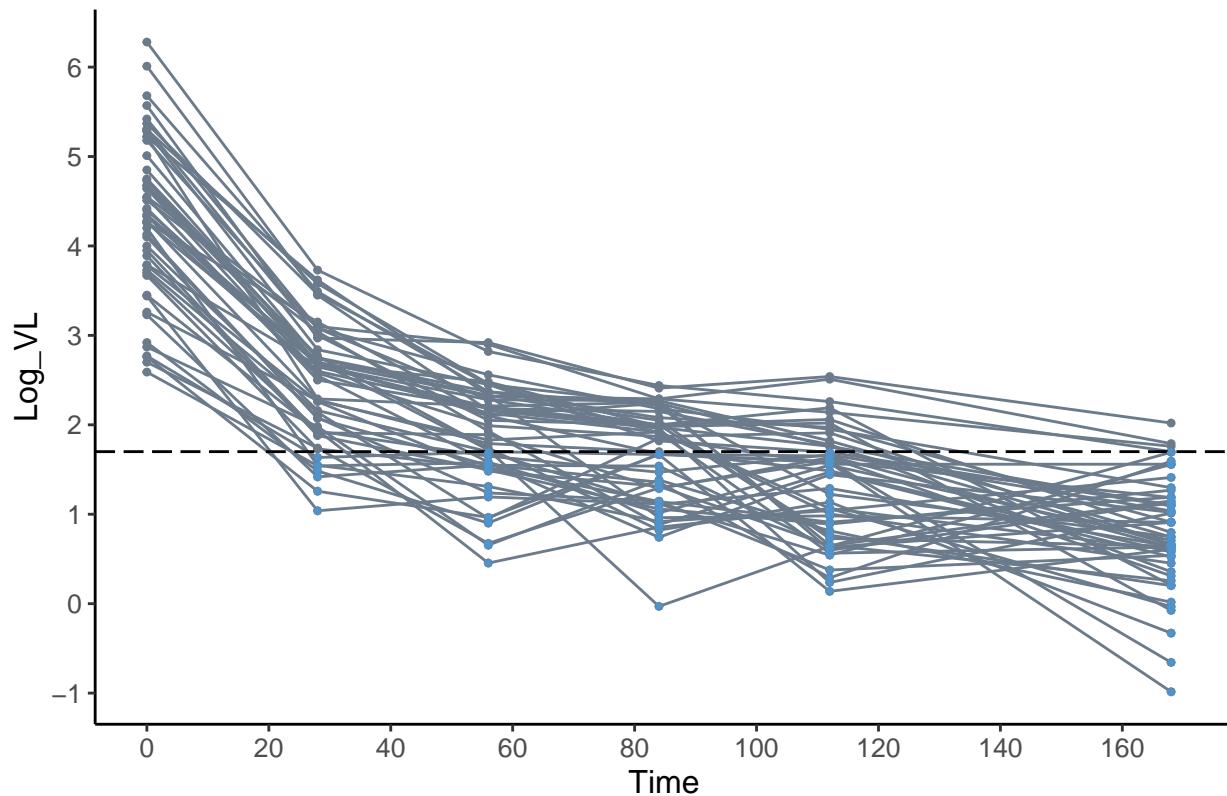
```
## Failed test, no output
## Warning: Removed 12 rows containing missing values (geom_point).
## Warning: Removed 12 row(s) containing missing values (geom_path).
```

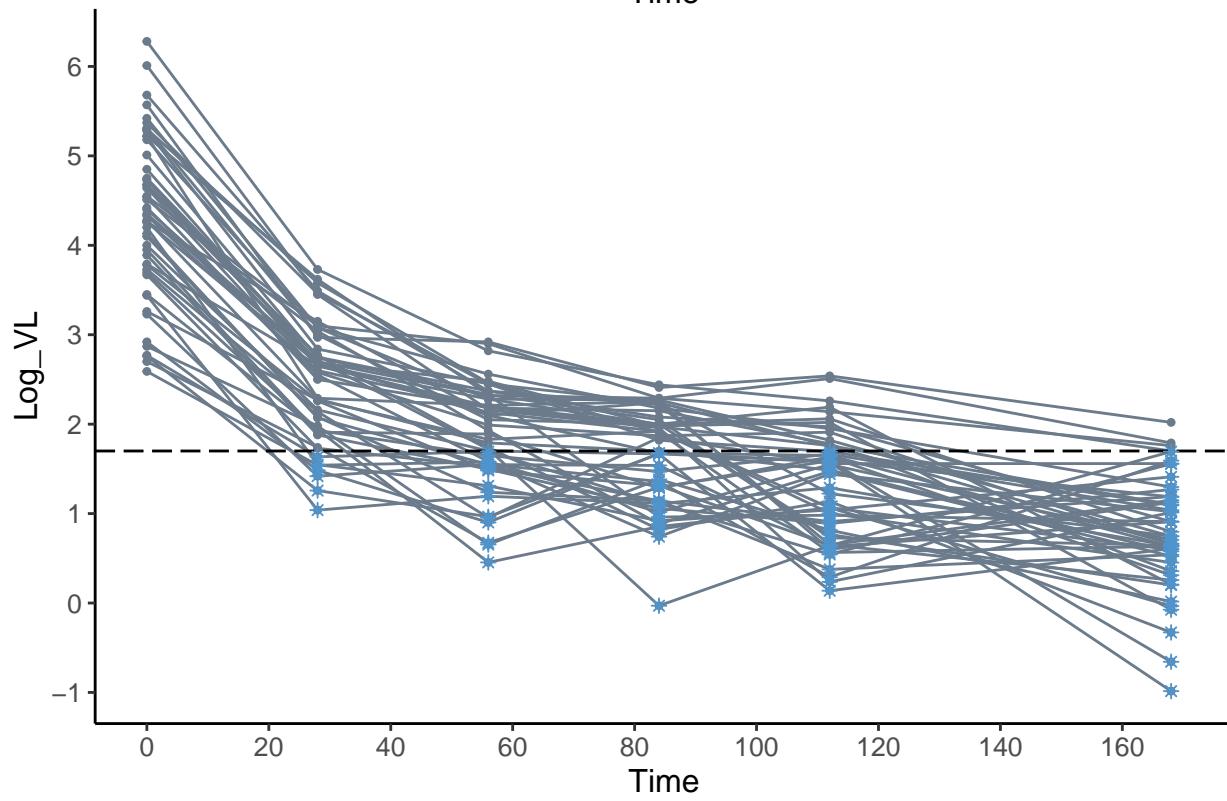
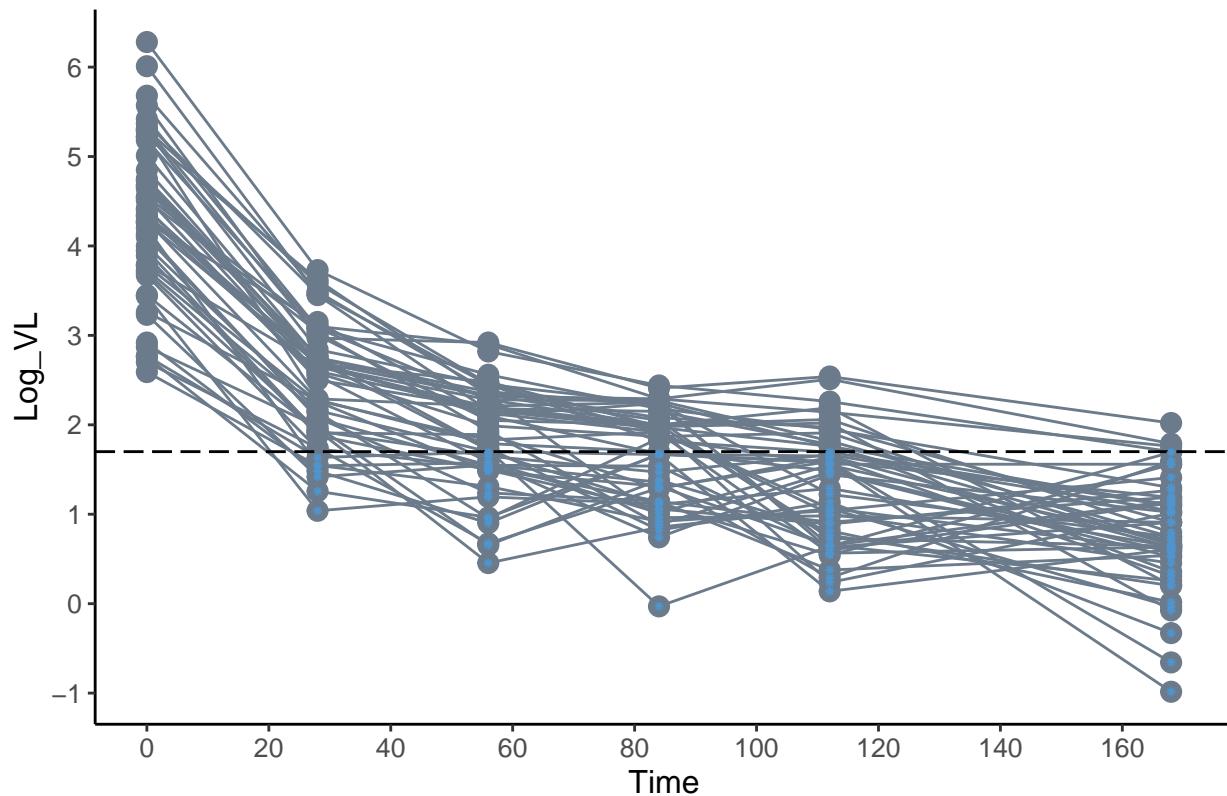


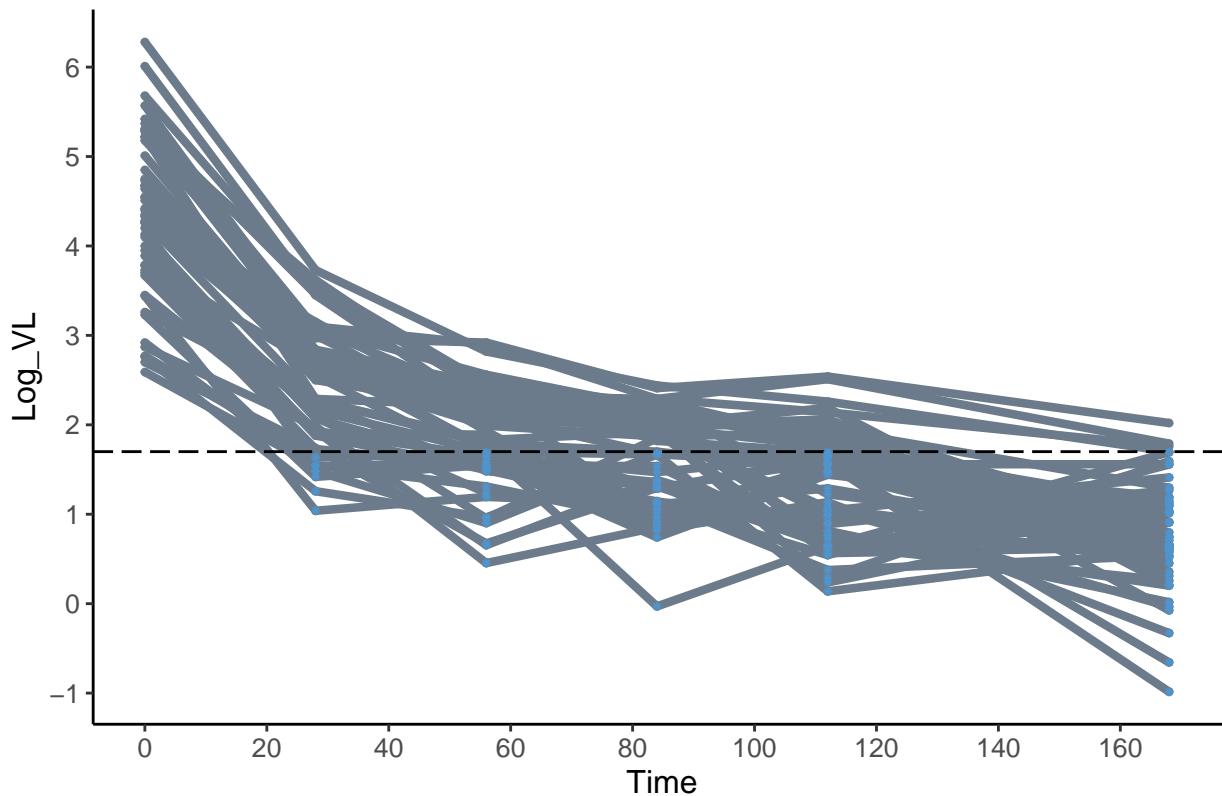


```
## Warning: Removed 131 rows containing missing values (geom_point).
## Warning: Removed 130 row(s) containing missing values (geom_path).
## Warning: Removed 131 rows containing missing values (geom_point).
```



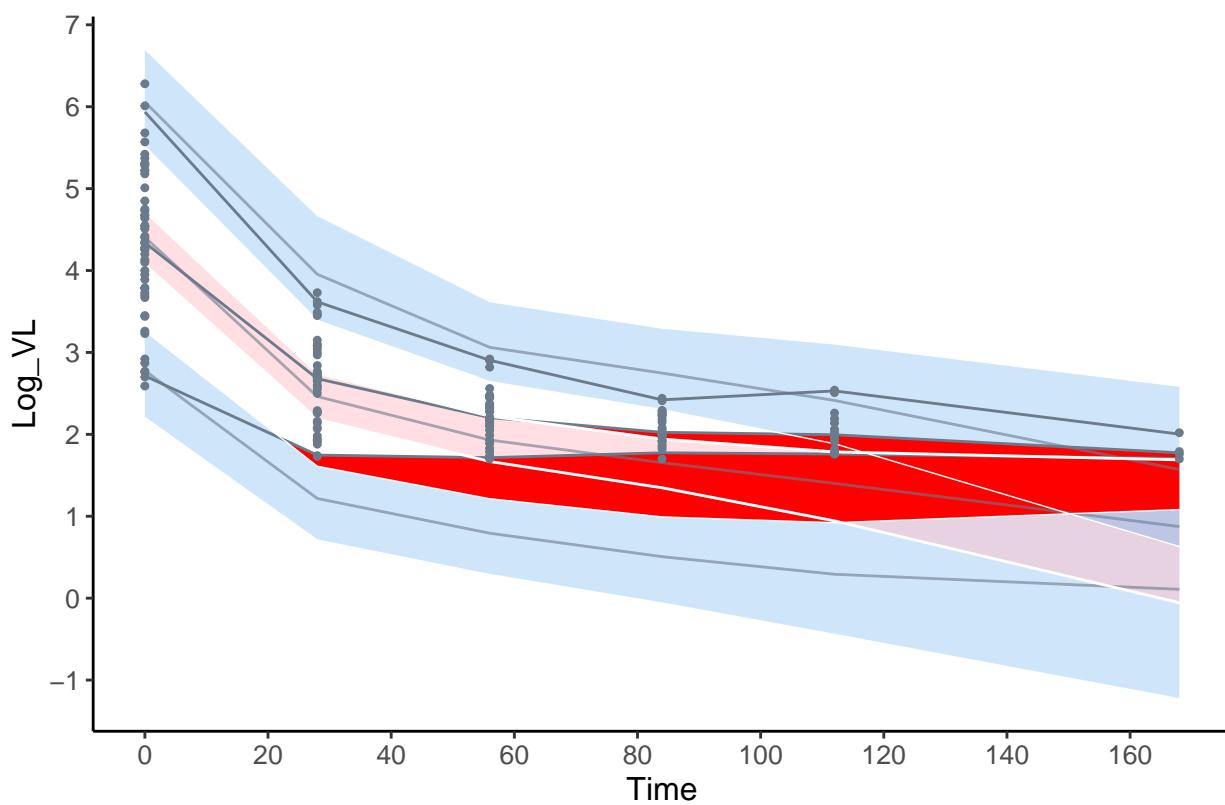
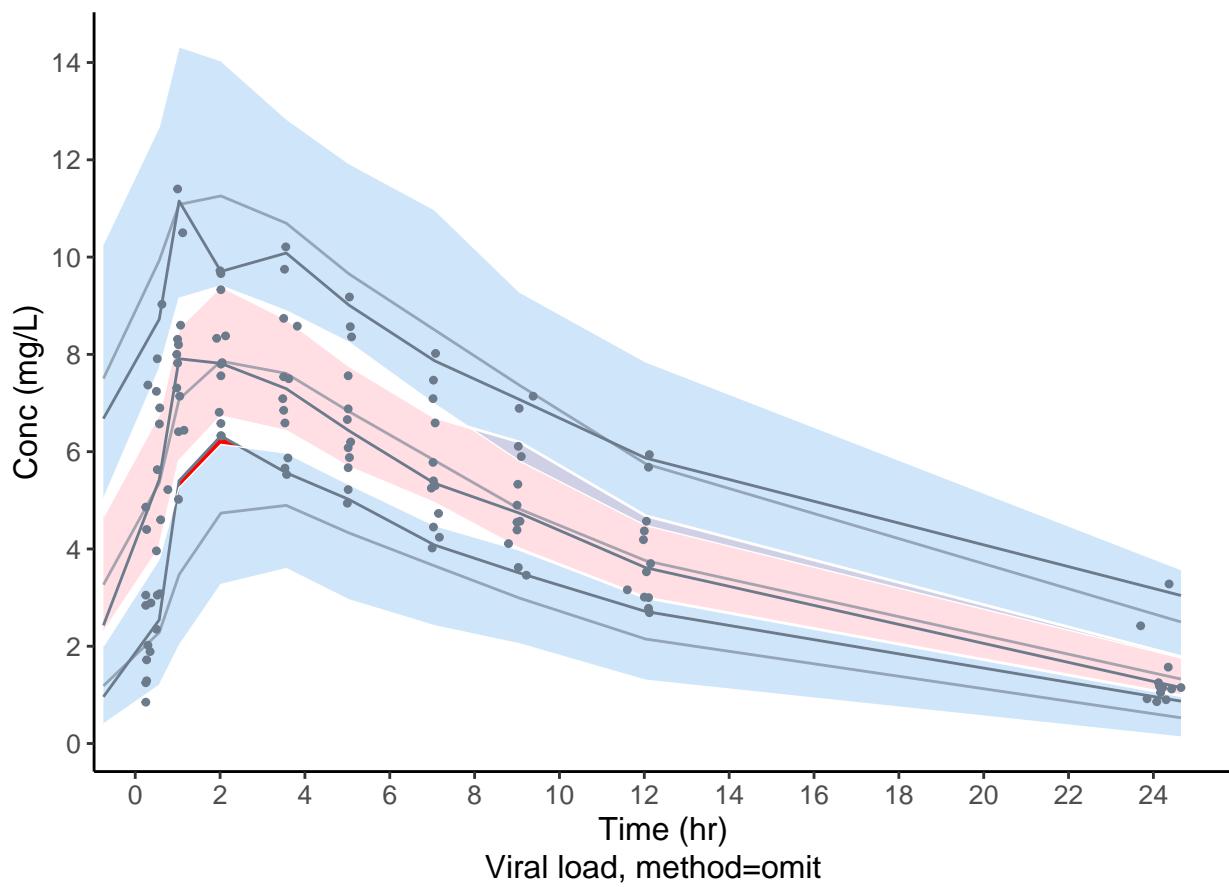




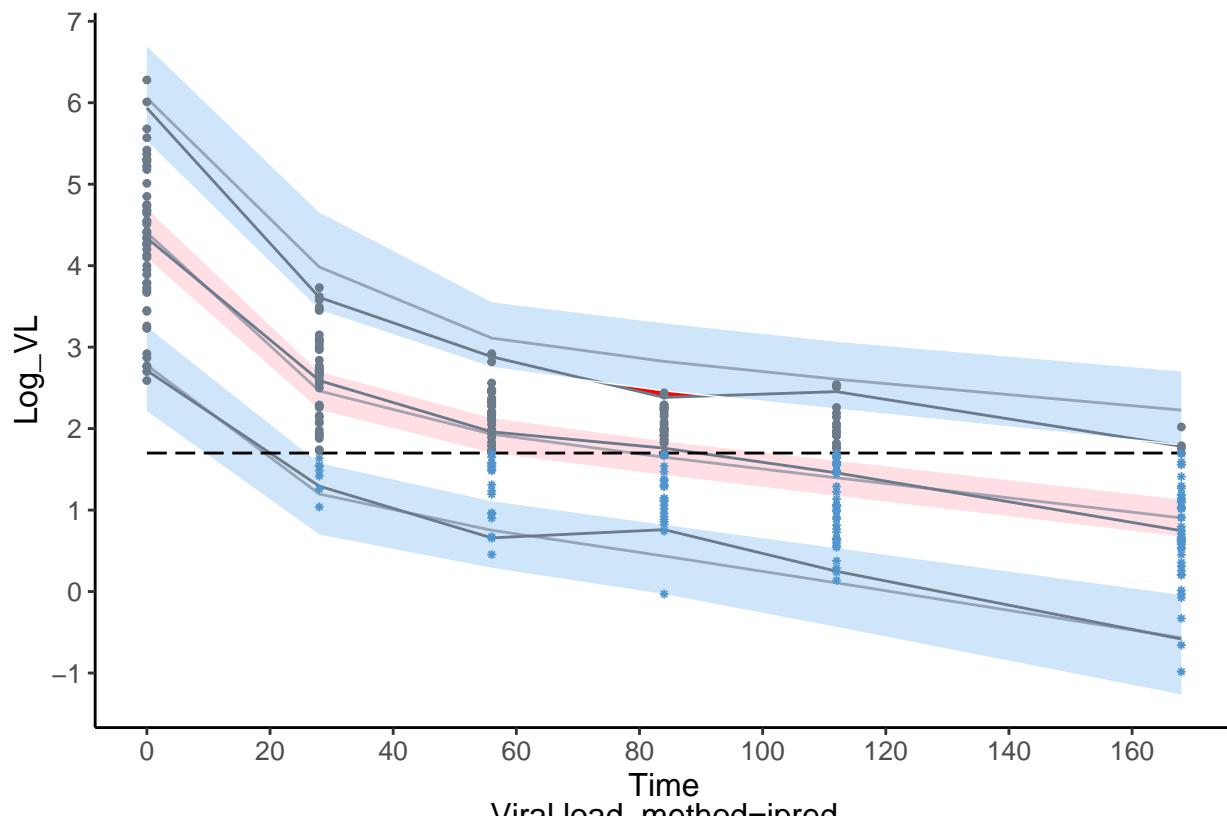


VPC

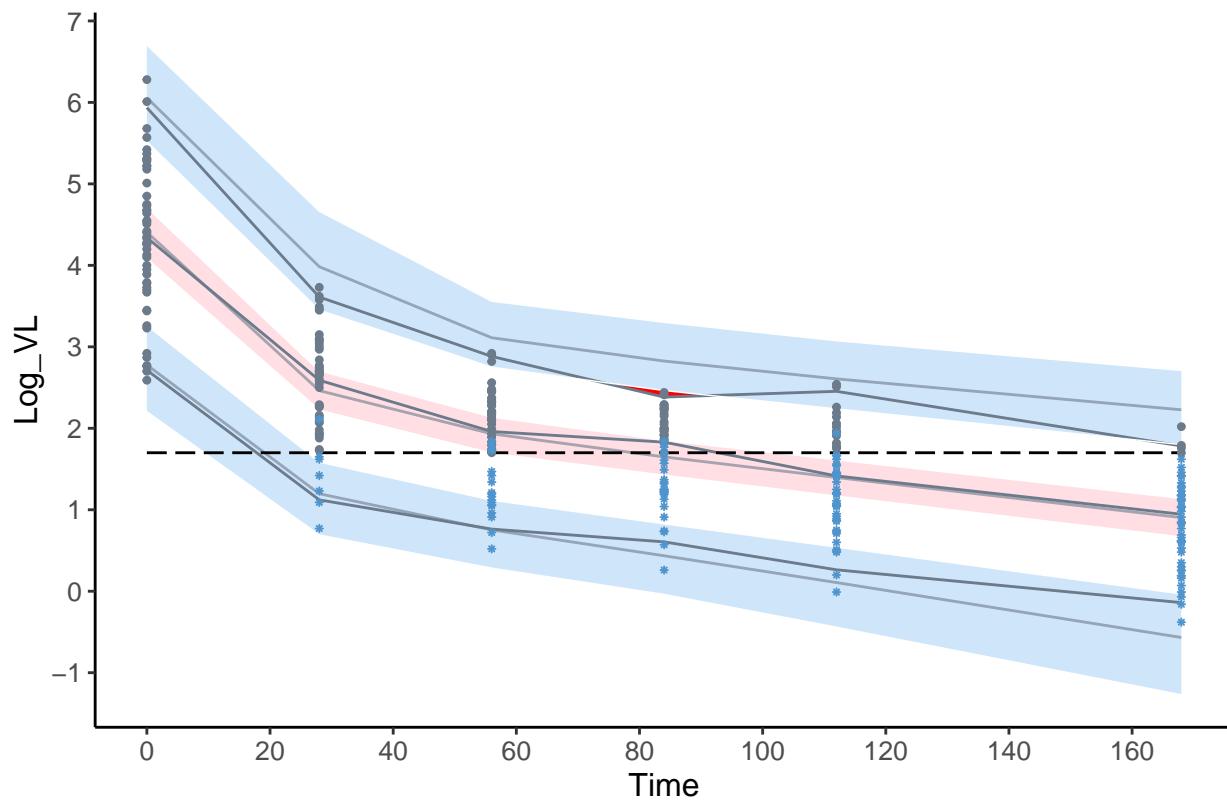
- **Problems (à résoudre Romain)**
 - Y-axis missing
 - problem with the PI not being interpolated properly (??? thought that worked :-/)
- **Notes to check Eco**
 - computation of the PI for **omit** consistent with the PI computed for **npde** ?
 - * compute the PI in this case by omitting the data > LOQ for the simulated data as well, to be consistent with the way the PI are computed for **npde**
 - * check also that the PI are computed consistently for **ppred**
 - * **ipred** ??? in this case can't be consistent (we have **isim** in the dataset but not **ipred**) so keep the simulations as is
 - **but** VPC with LOQ data done as here => **Eco TODO:** expliquer dans la doc les méthodes de censoring et la différence entre les graphes **npde** et **VPC** dans le cas de la méthode **omit** (message=the discrepancy seen in **VPC** comes from not treating observed and simulated data in the same way, “corrected” in **npde** but at the price of a loss of power)

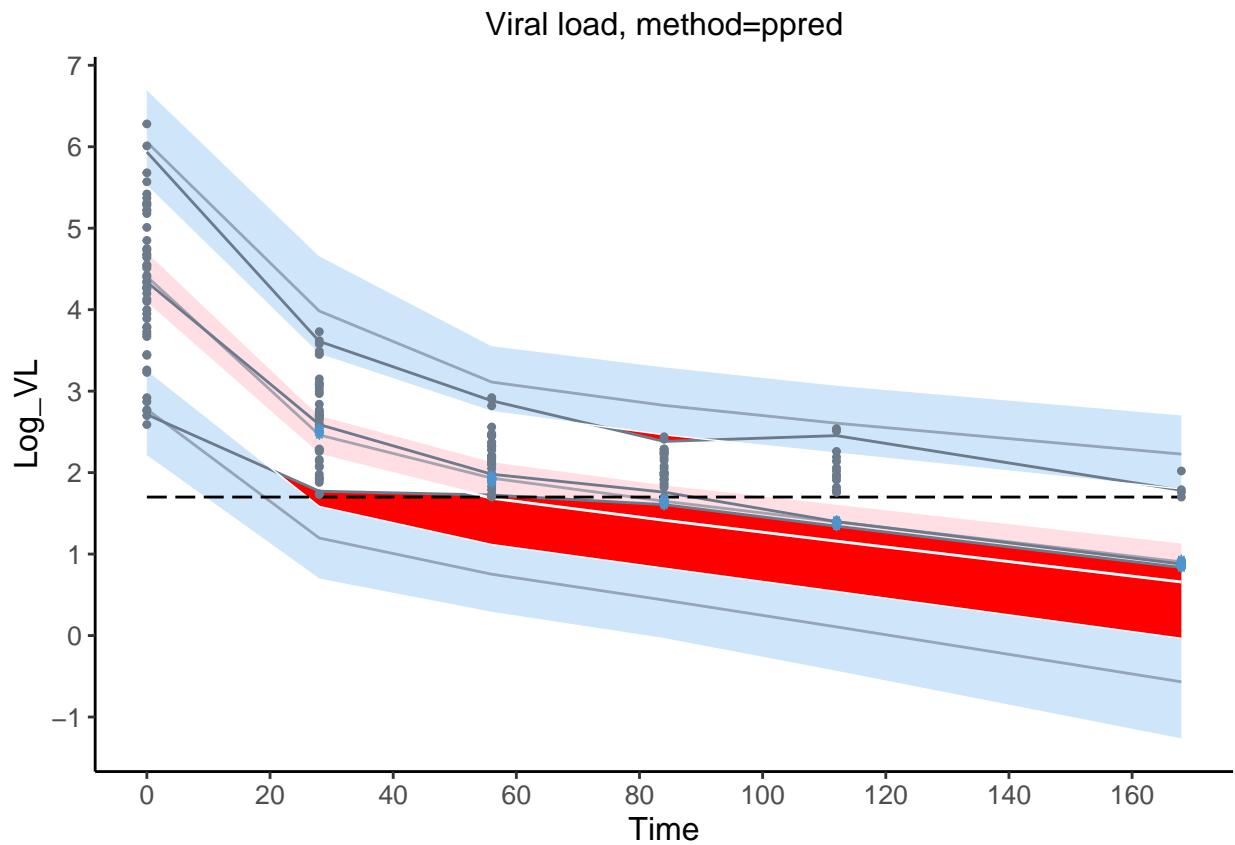


Viral load, method=cdf

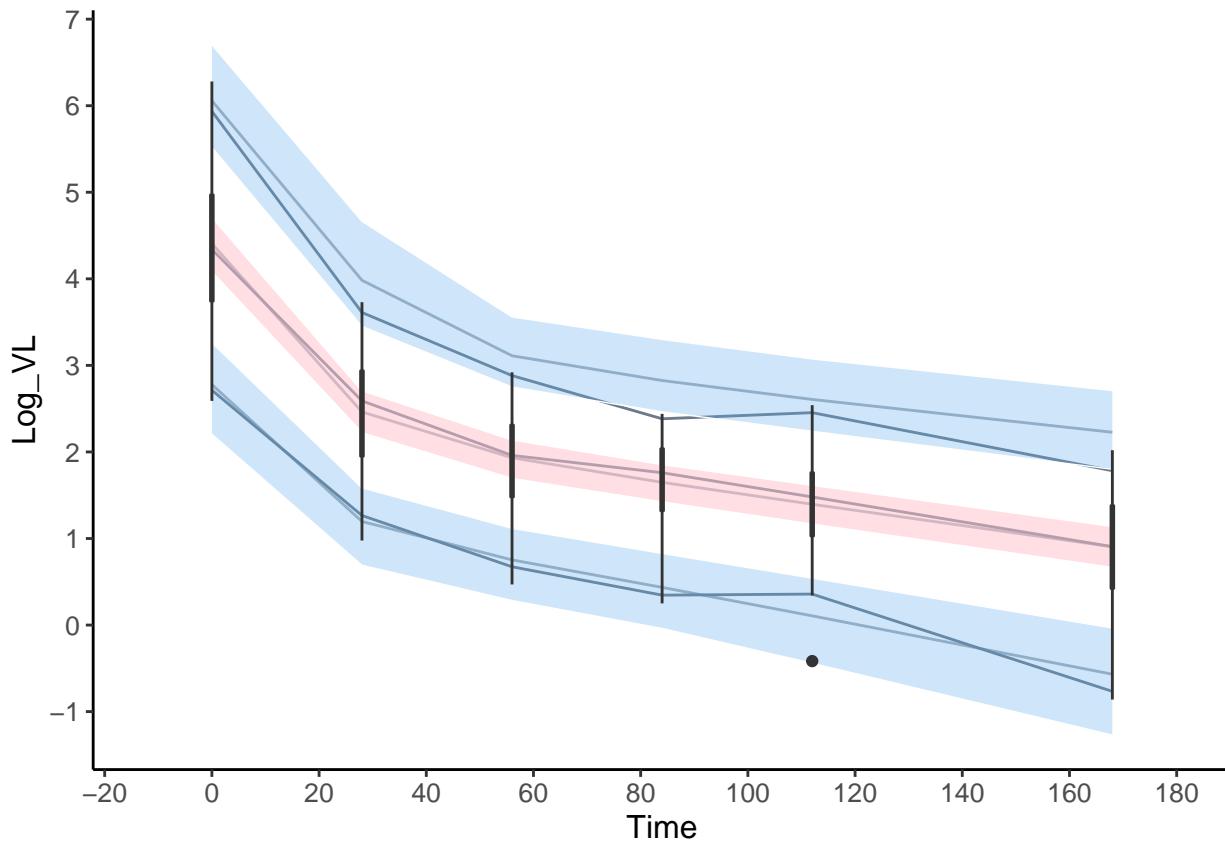


Viral load, method=ipred



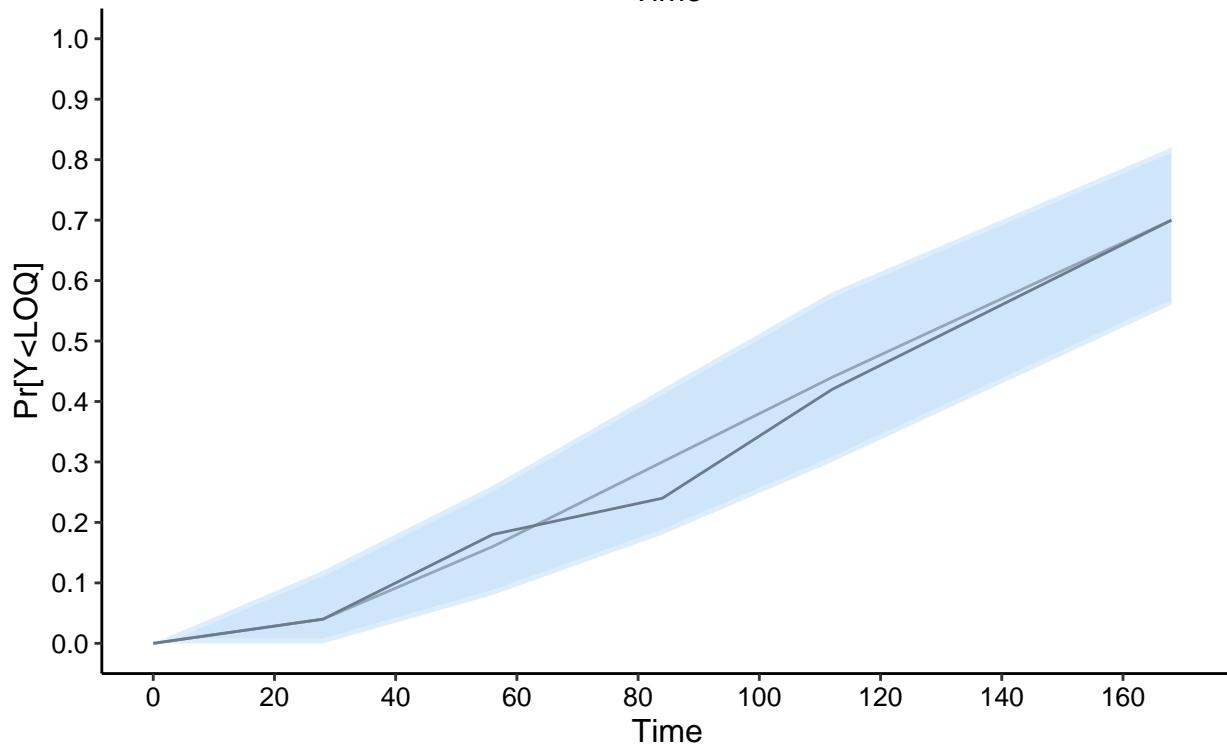
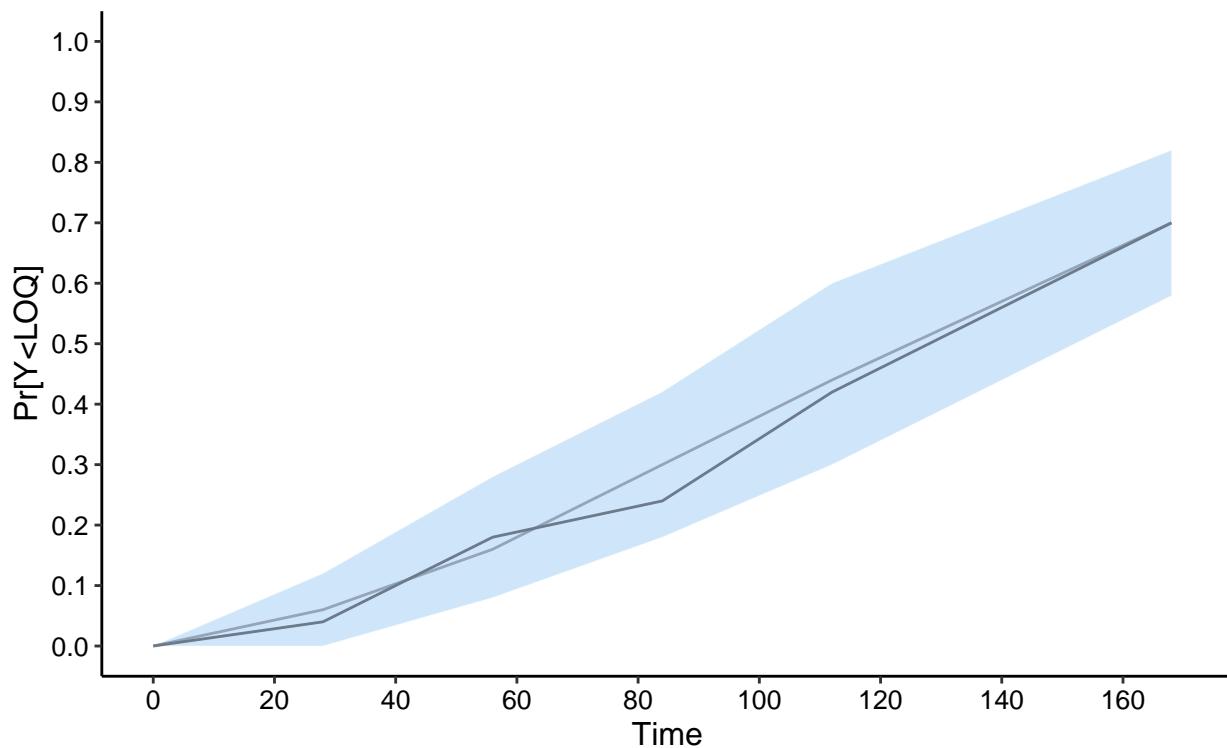


```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```



P(Y<LOQ)

- **Problems** (à résoudre Romain)
- **solved** fails
- avant, vérifier: - **solved PI** on ne voit pas les bornes extérieures du PI (ça devrait être contrôlé par lty.bands et col.bands aussi) - **solved axes** missing lines for axes - **solved lwd.bands** should probably be increased for this plot :-/ (compared to the others), but never mind => show an example in Beautiful graphs

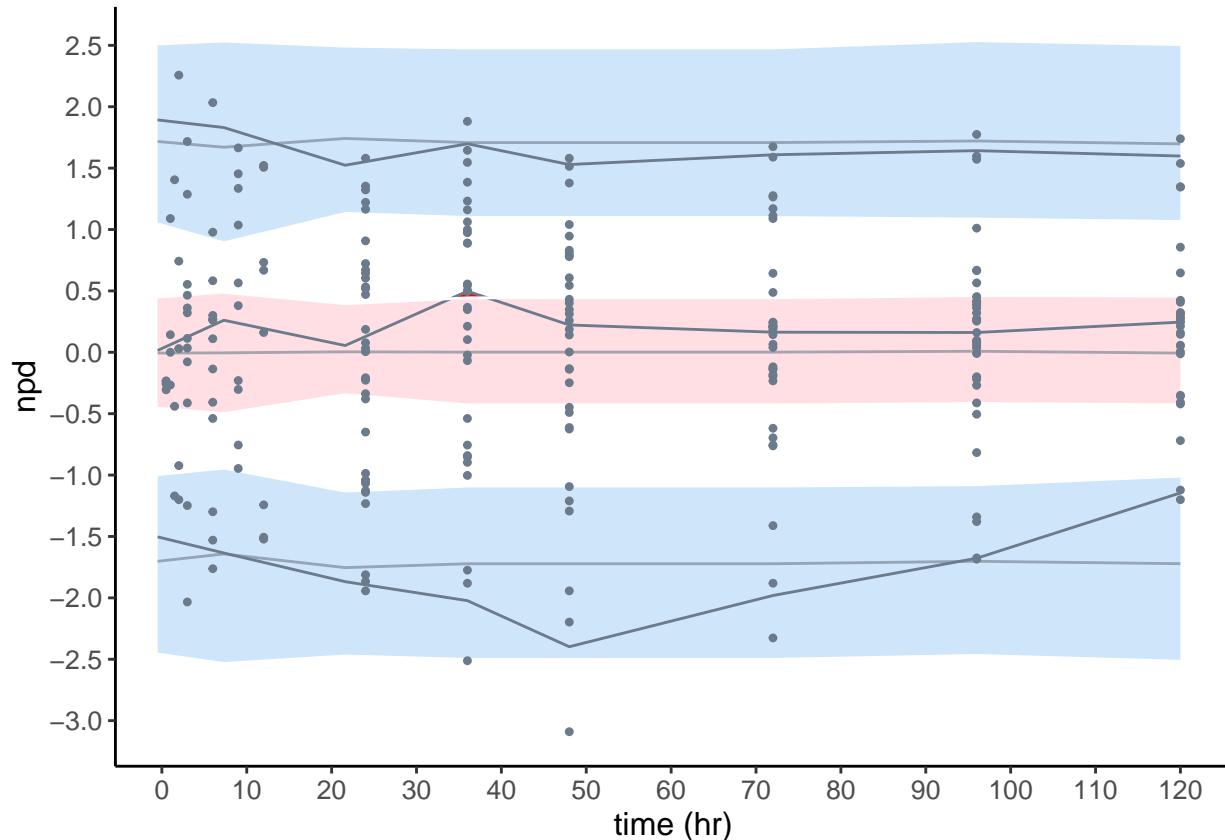


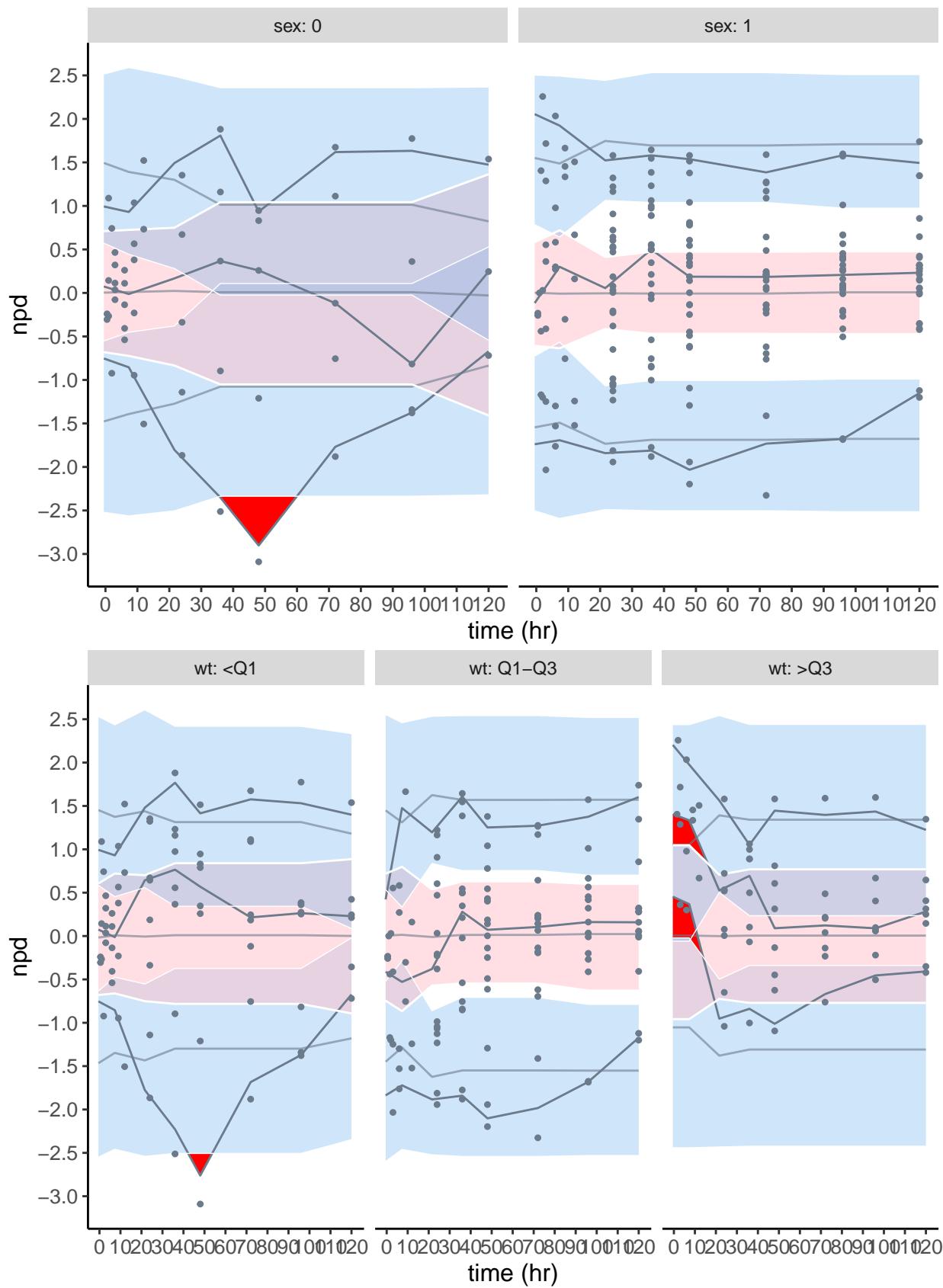
Covariate plots

Covariates

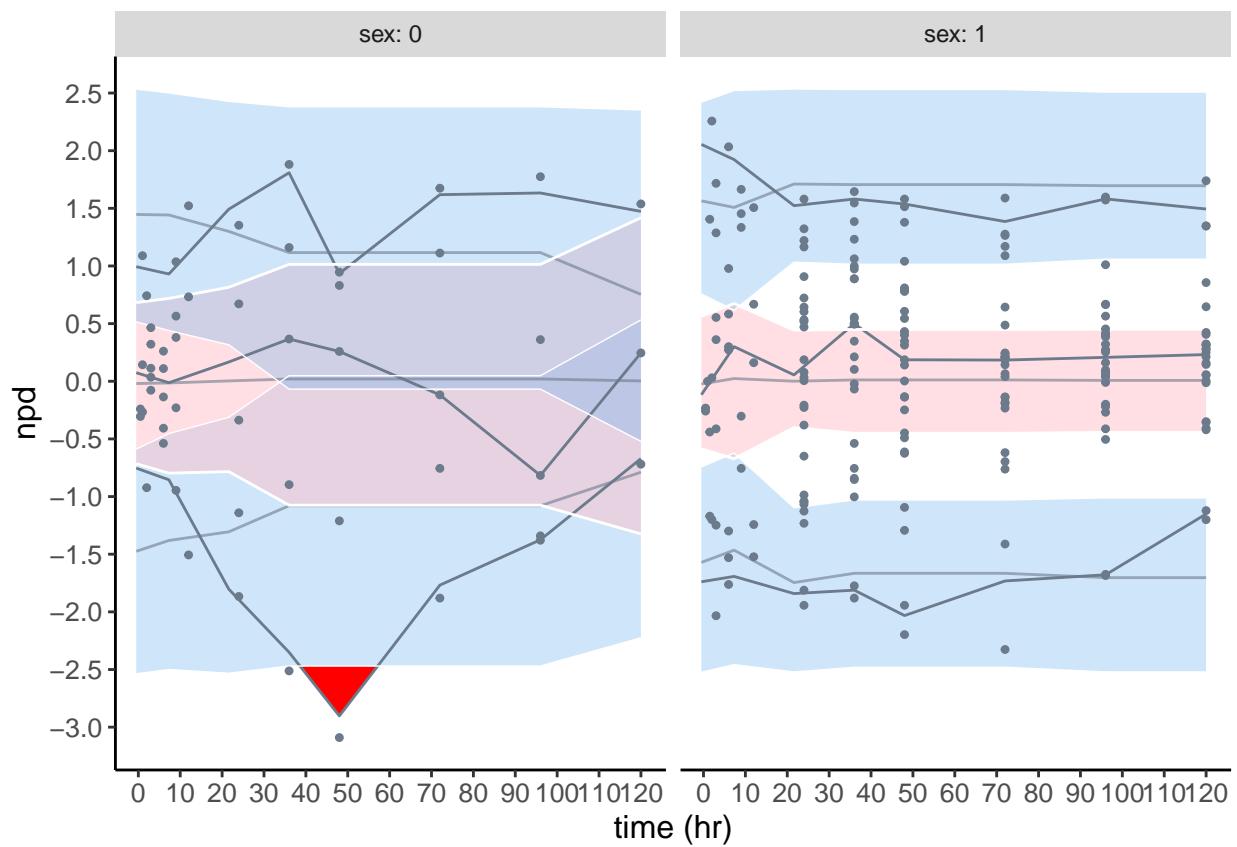
Splitting plots according to covariates

- **Problems (à résoudre Romain)**
 - **solved** yes here we really need to extend PI to cover the whole X-range for x.scatter and pred.scatter (not VPC)
 - * just the PI (not the observed percentiles or outlier bands)
 - **covsplit**
 - * **solved** doesn't work with the default plots => behaviour TBD
 - with one covariate, gives default plot (not split by the covariate)
 - with several covariates, message *arrangeGrob(...)* ...: *nrow ncol >= n* is not TRUE
 - we should get default plots for each category of the covariate but maybe only for one covariate ??? (too crowded otherwise)
- **Problems (à résoudre Eco)**
 - ordering categorical variable (?)
 - why is there a blank space in the pred.scatter plot for sex=1 ?
- **other checks to do**
 - check if PI extend on the X-range when less homogenous data



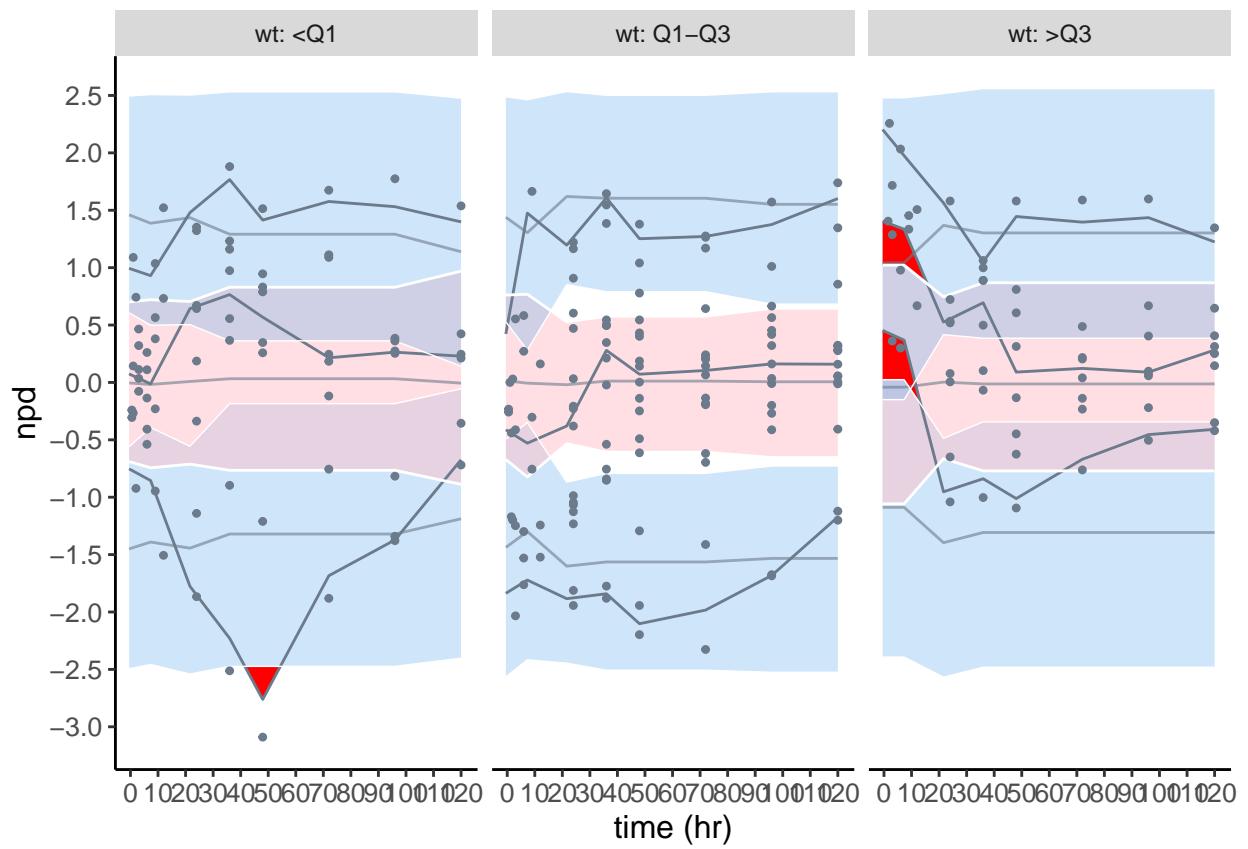


```
## [[1]]
```

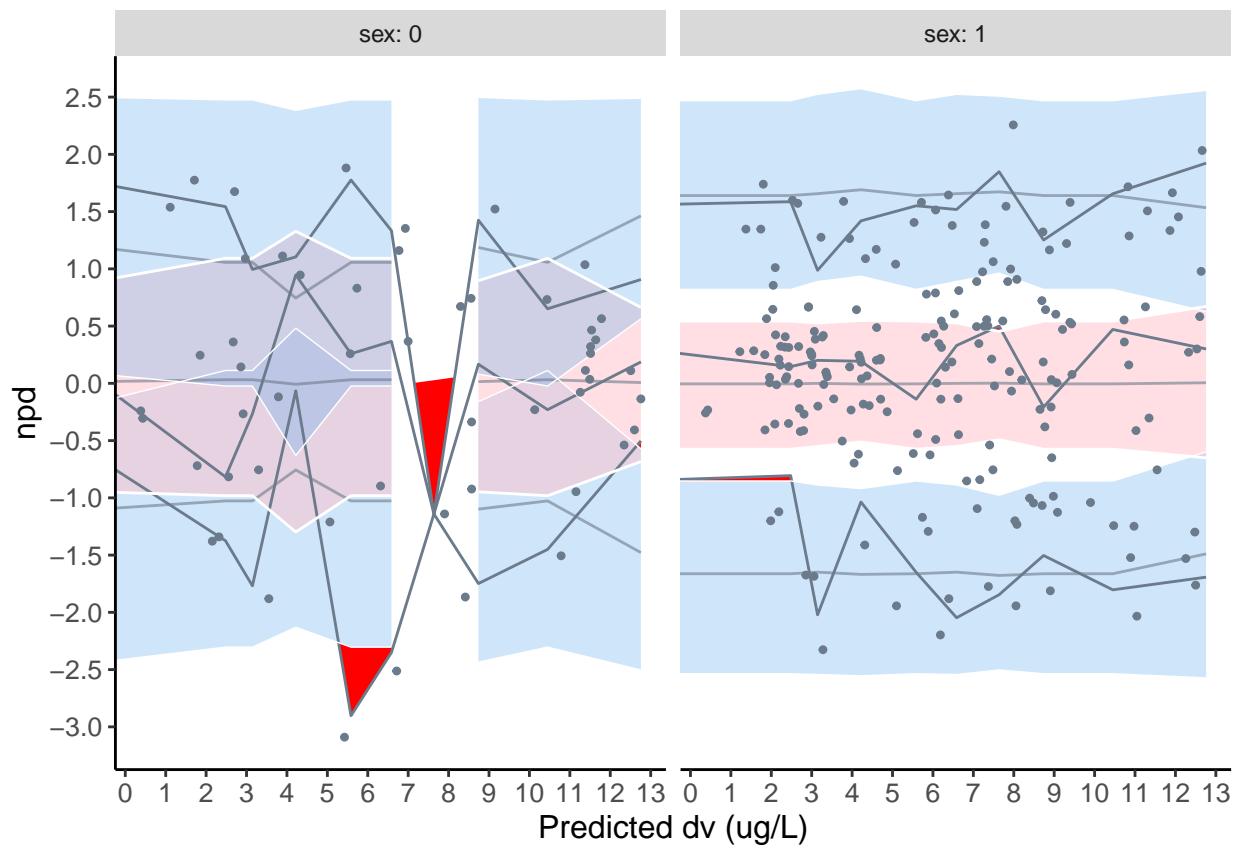


```
##
```

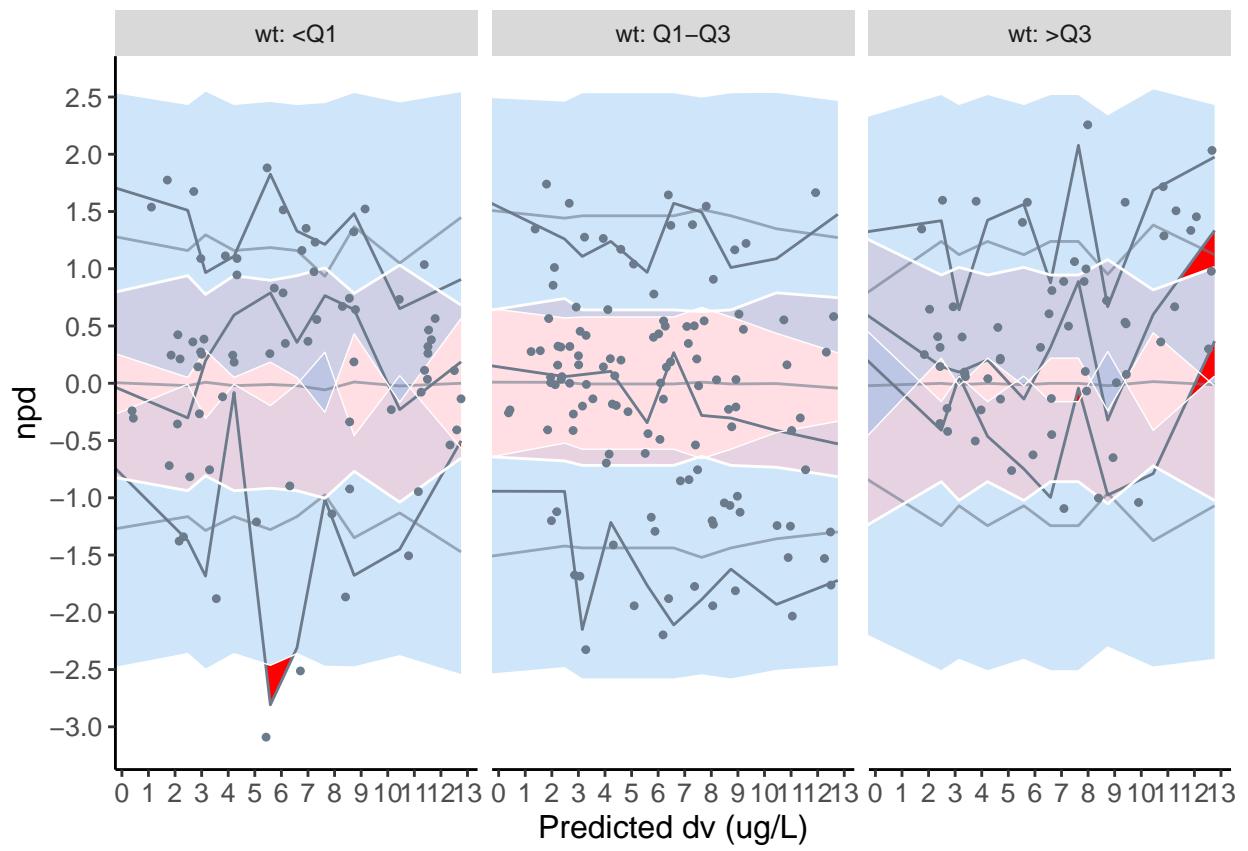
```
## [[2]]
```



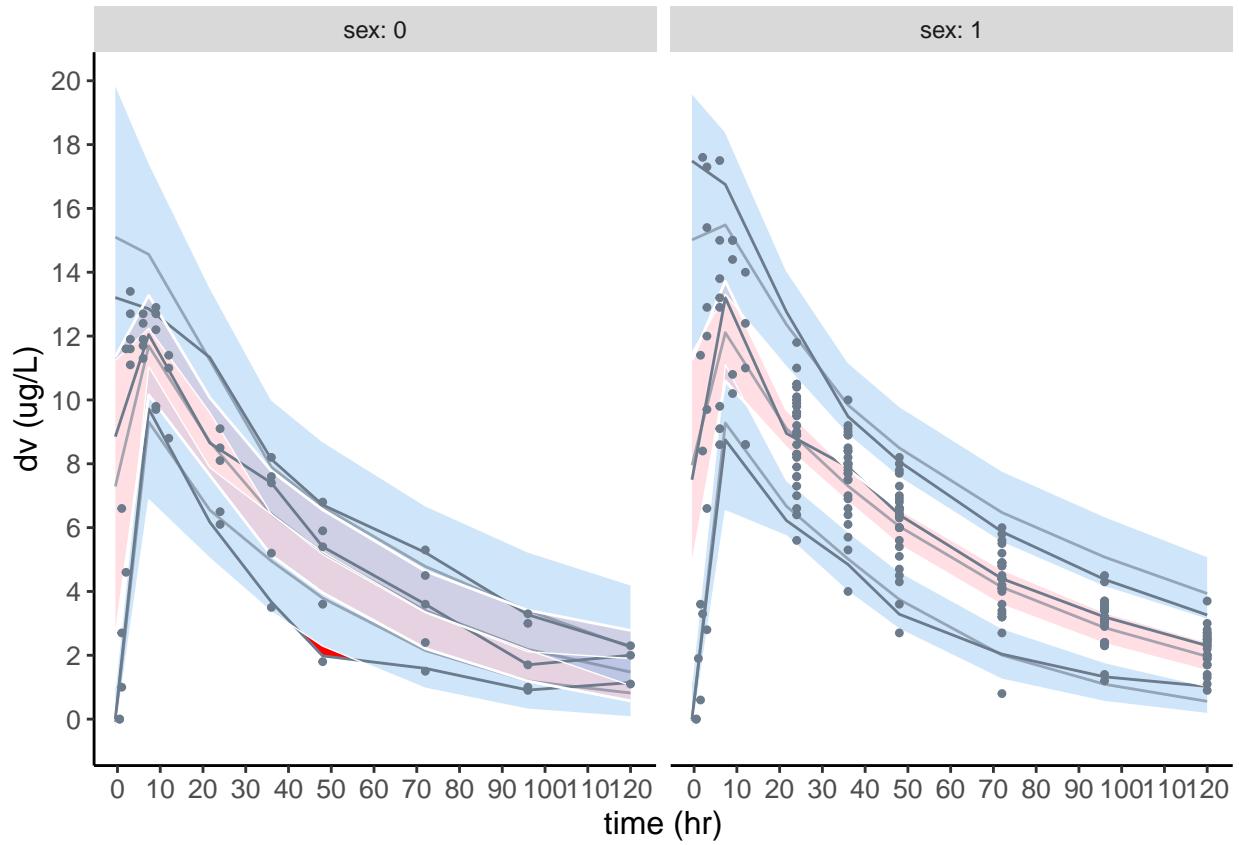
```
## [[1]]
```



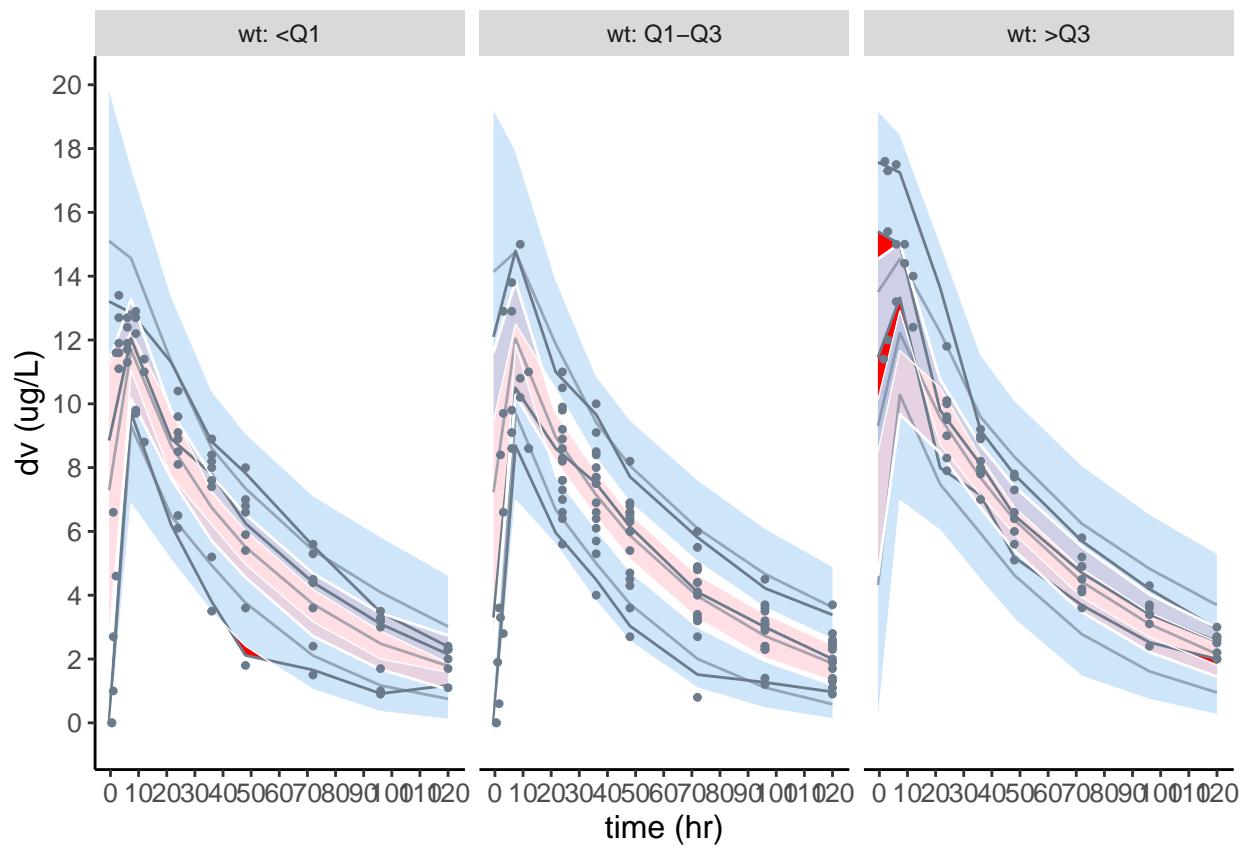
```
##  
## [[2]]
```



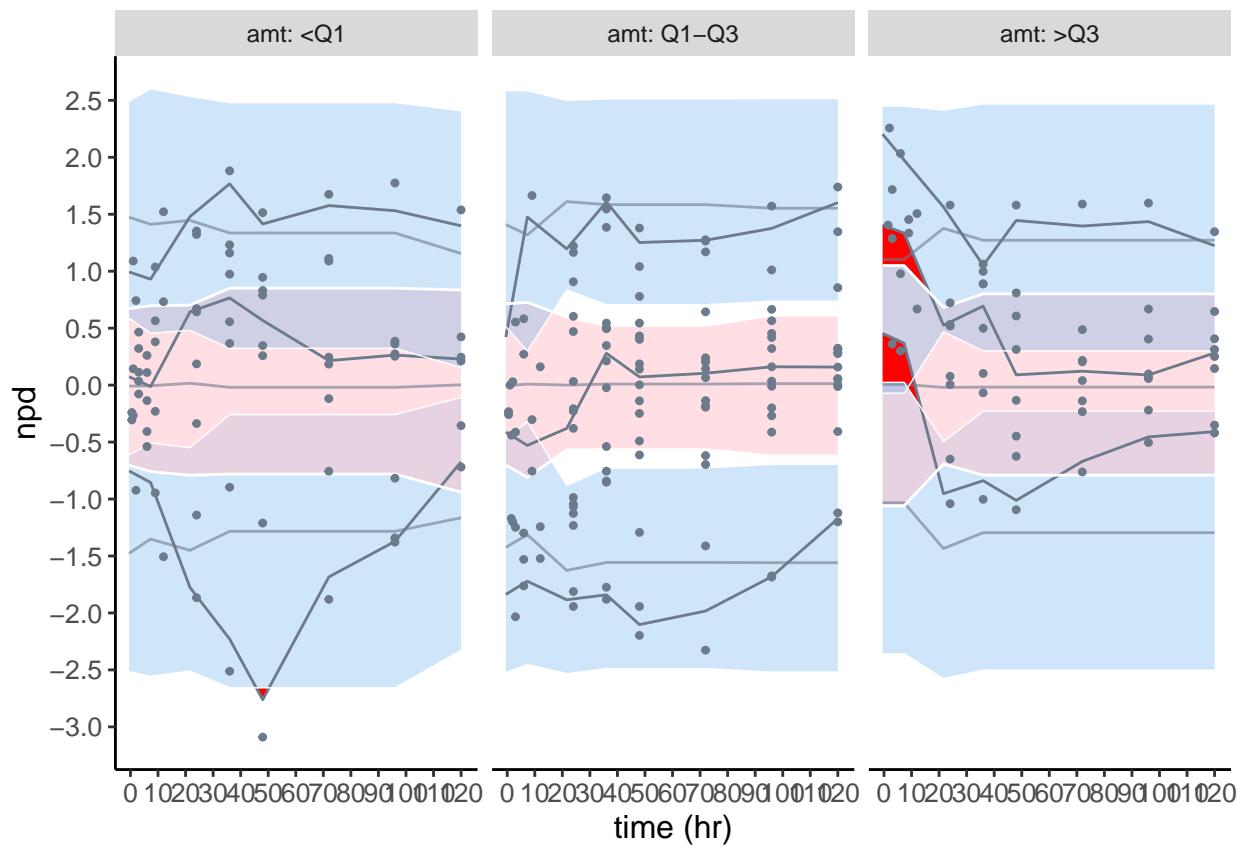
```
## [[1]]
```



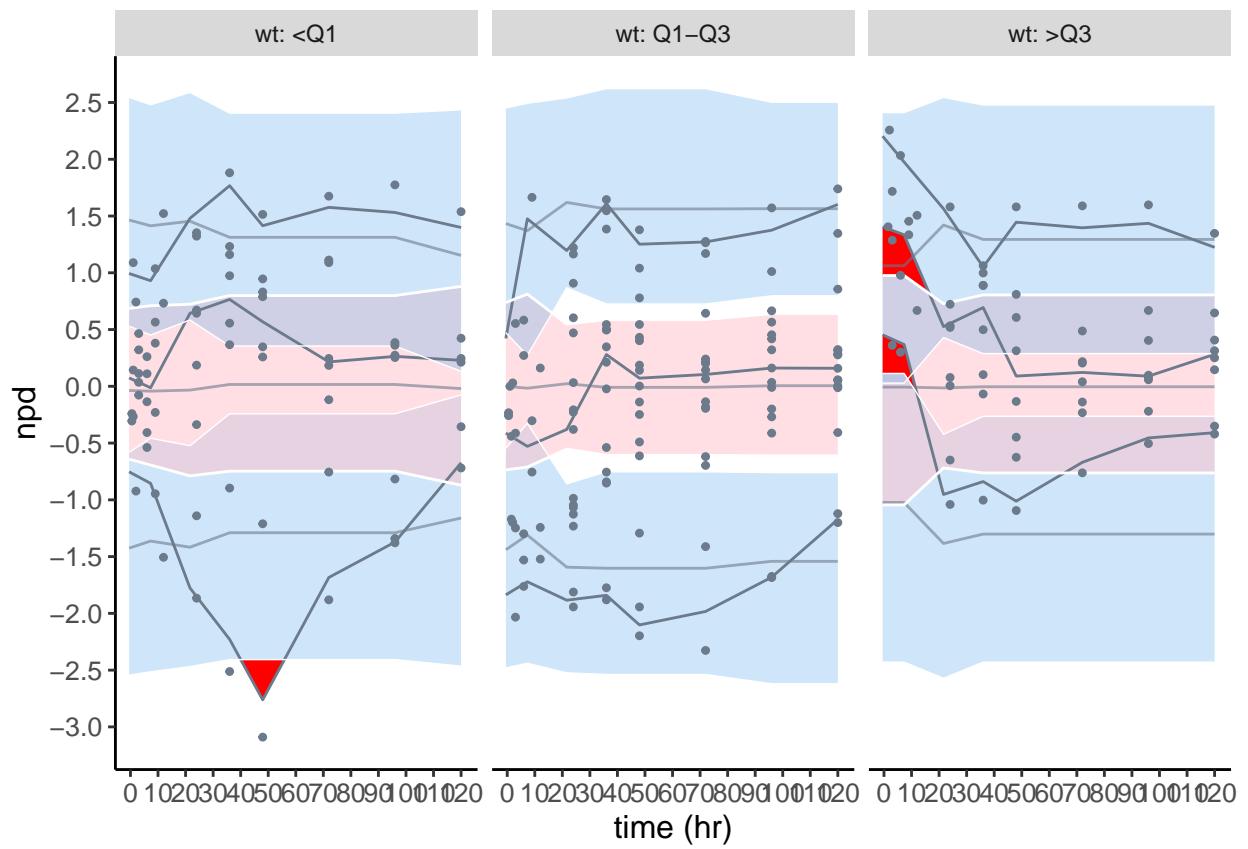
```
##  
## [[2]]
```



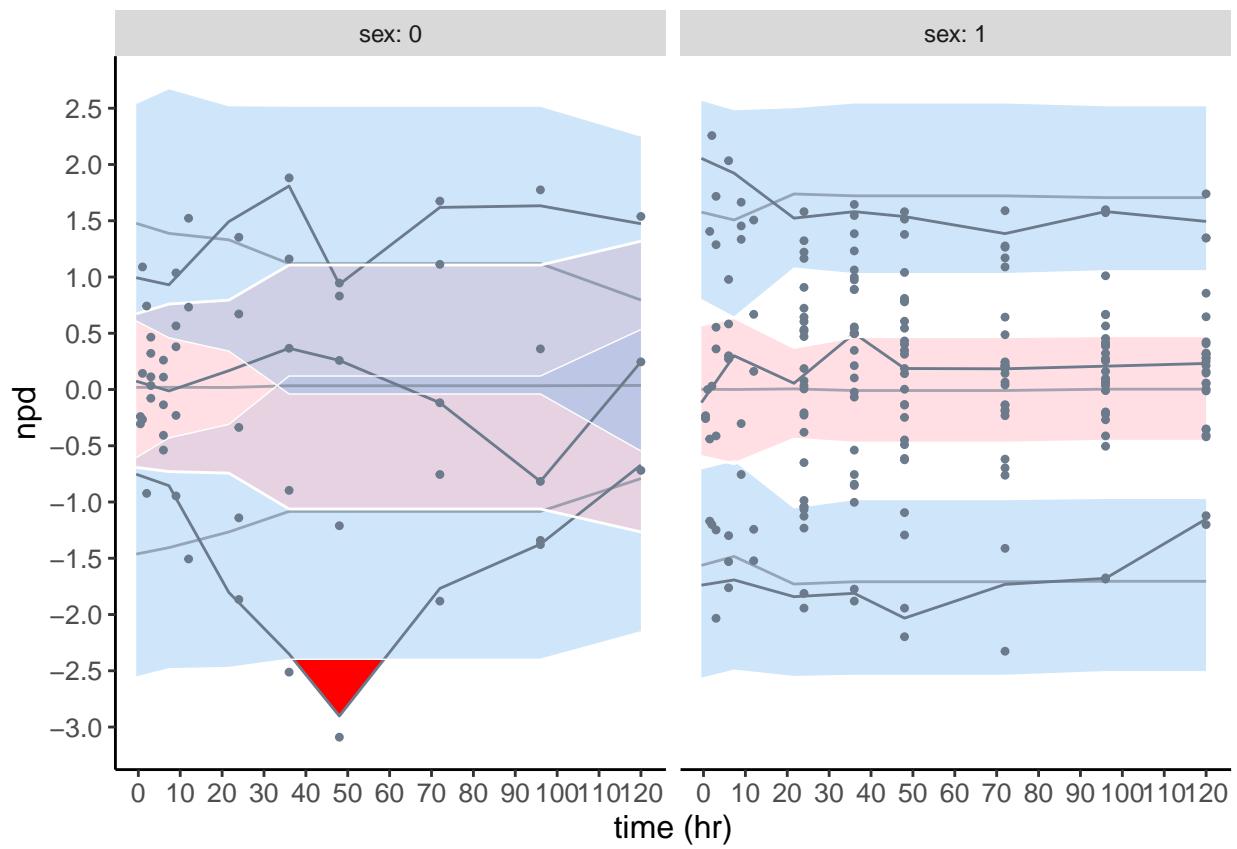
```
## [[1]]
```



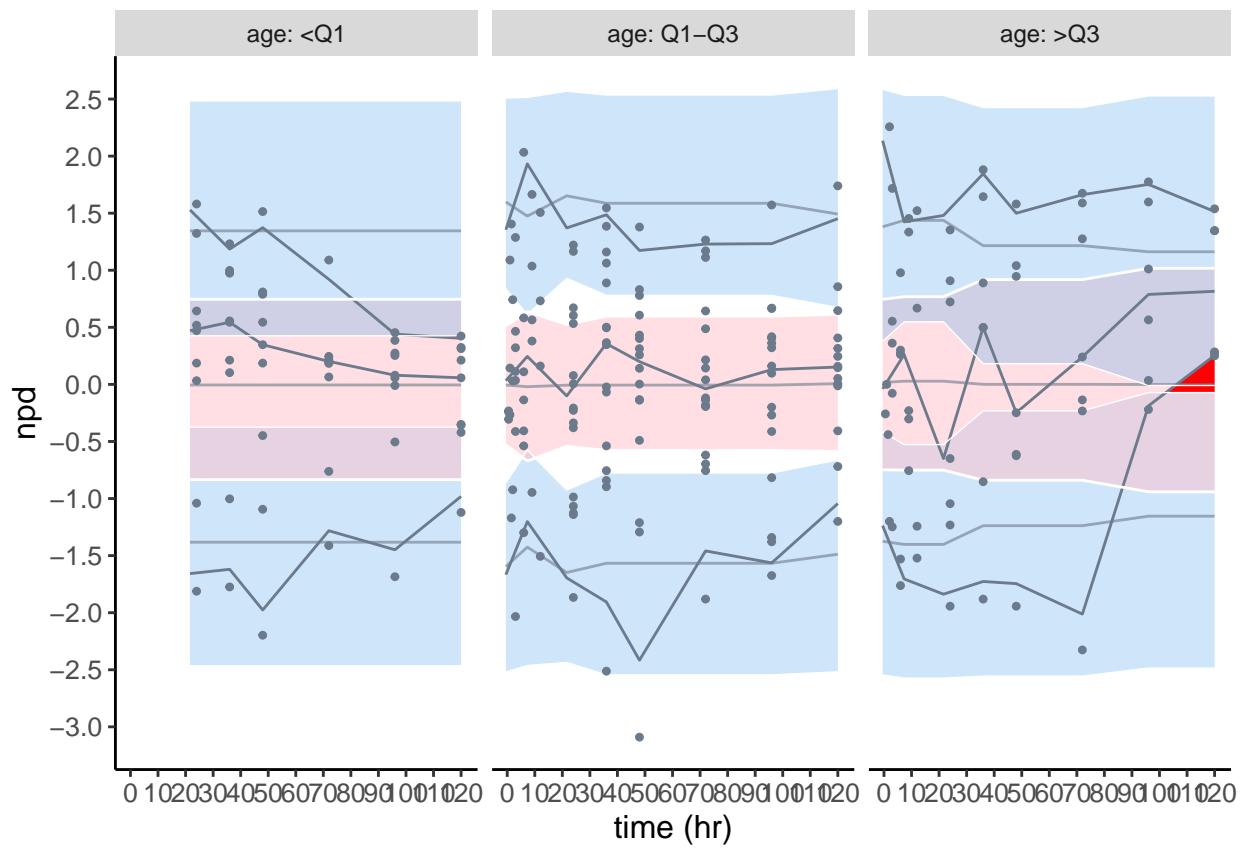
```
##  
## [[2]]
```



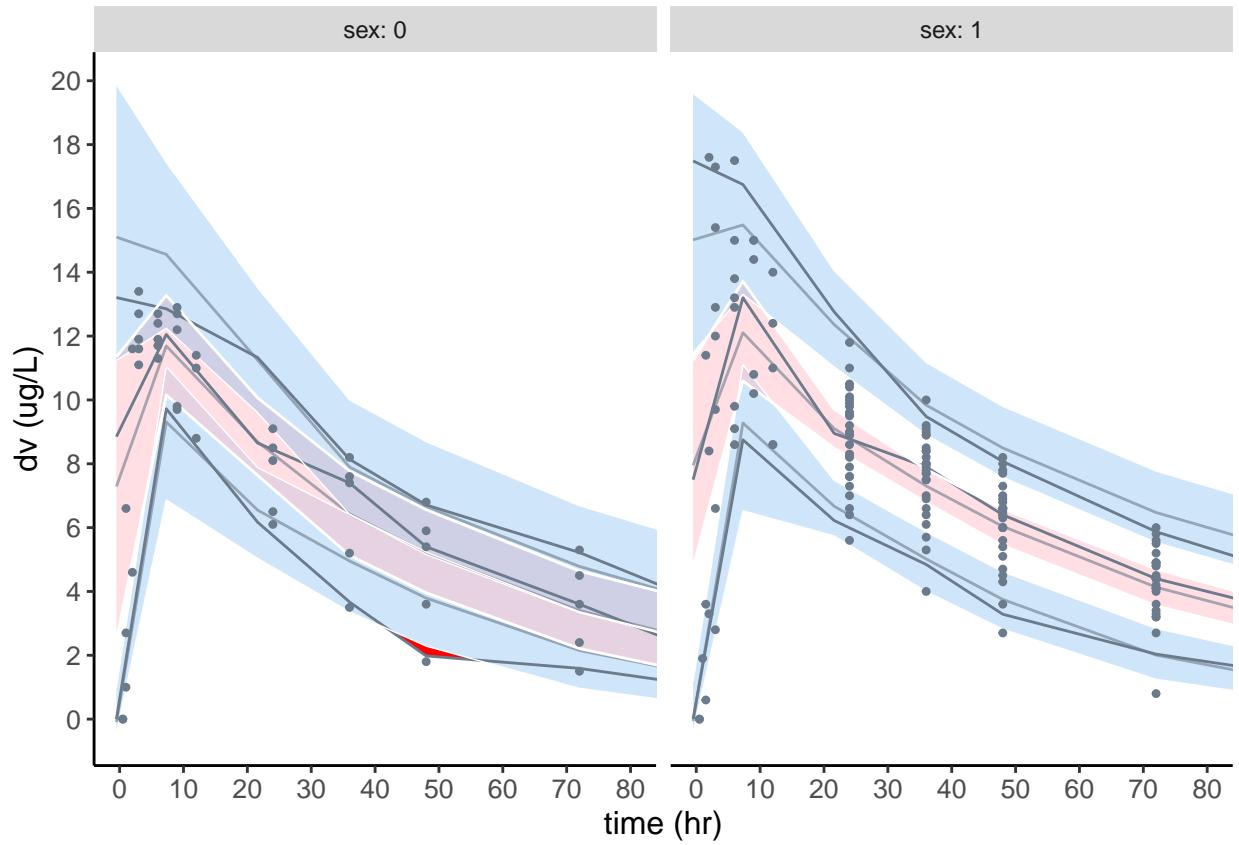
```
##  
## [[3]]
```



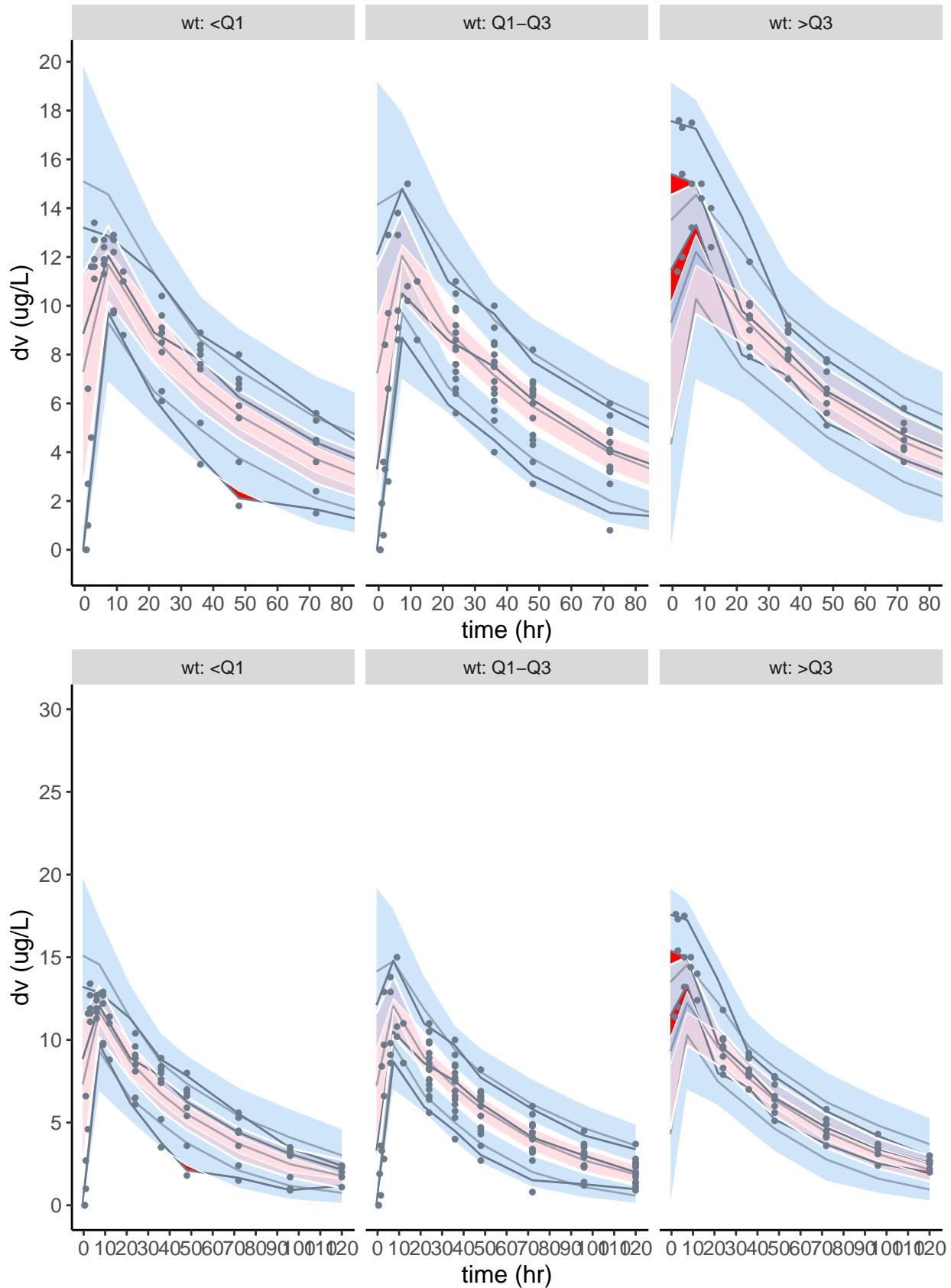
```
##  
## [[4]]
```

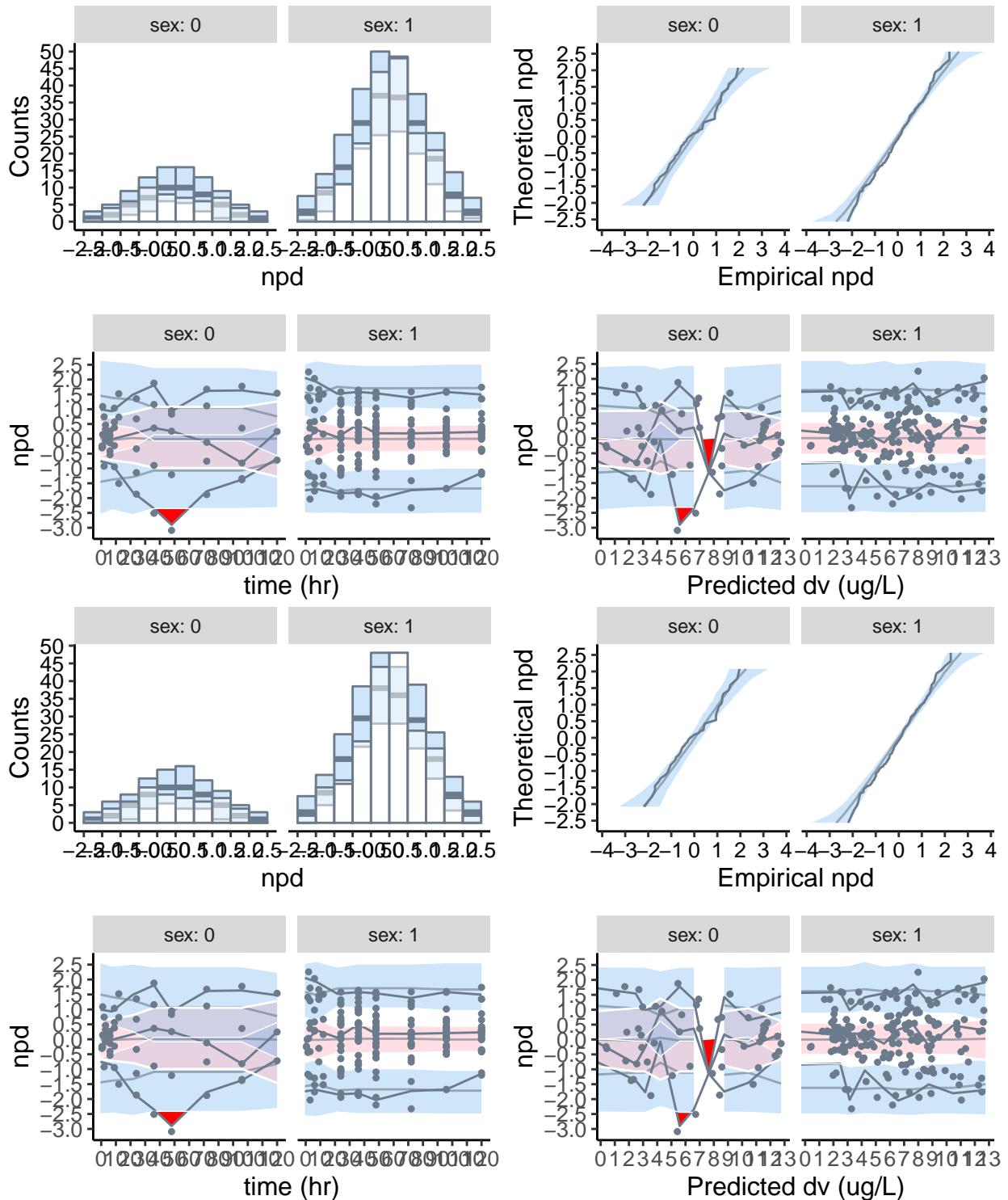


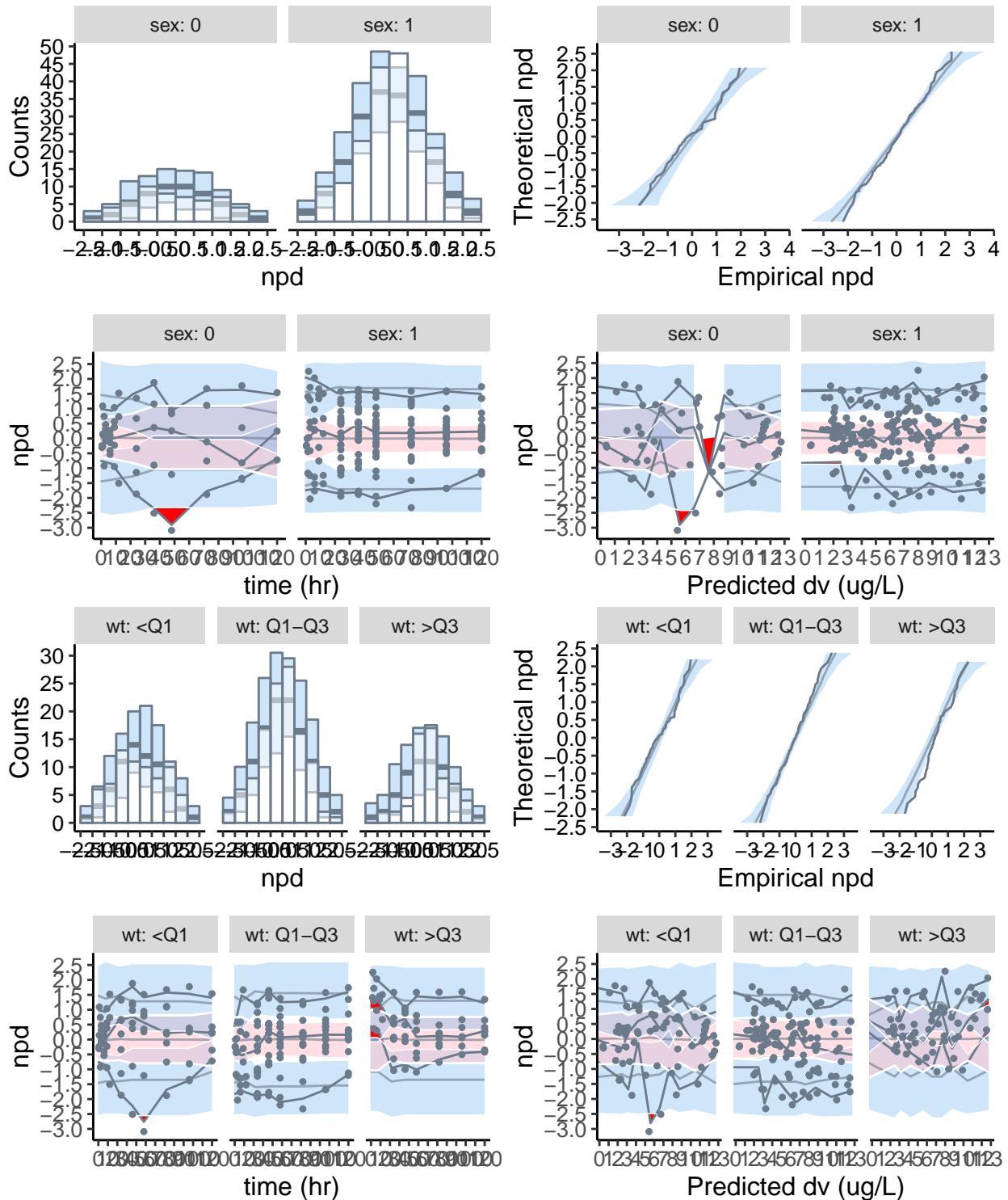
```
## [[1]]
```

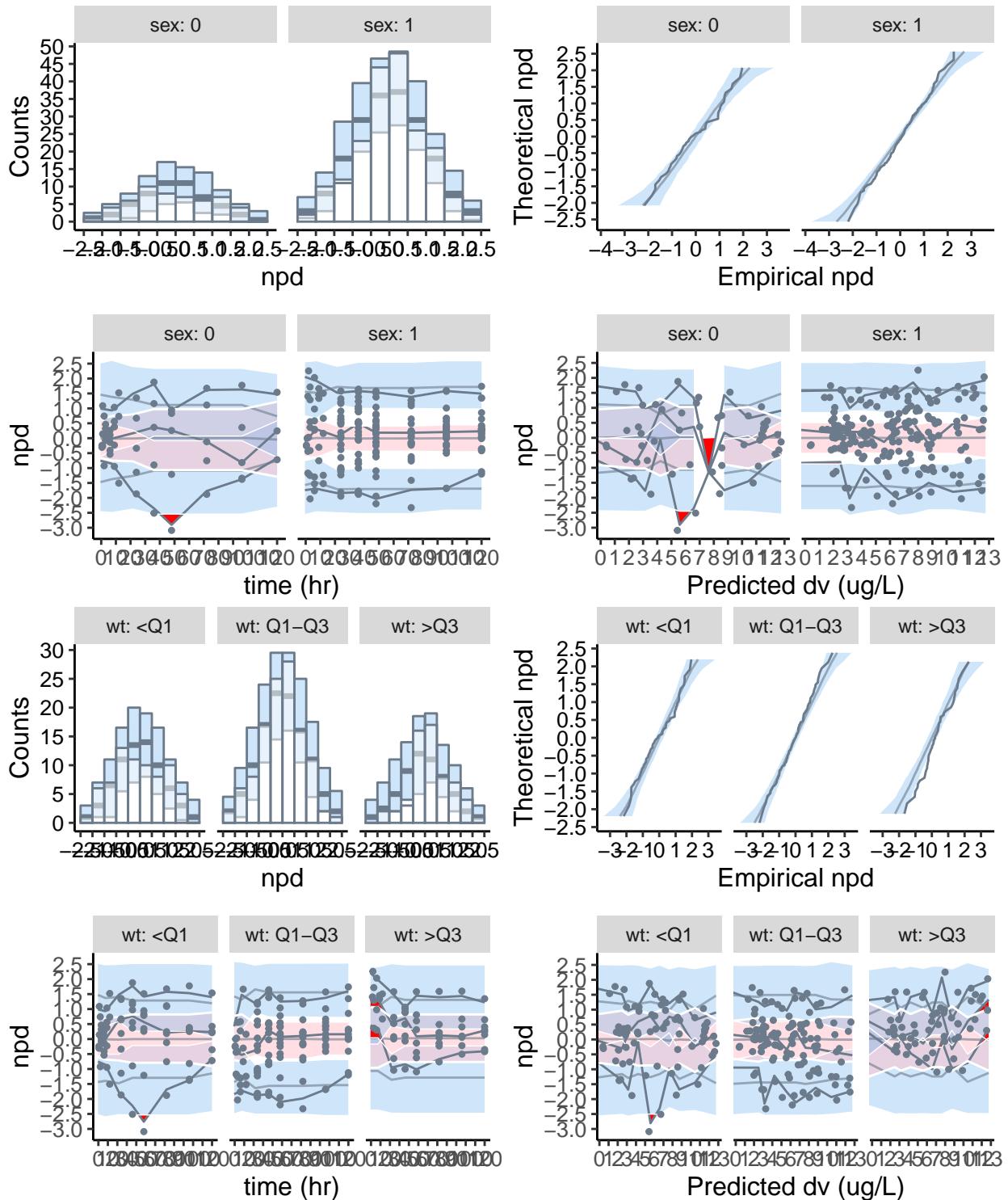


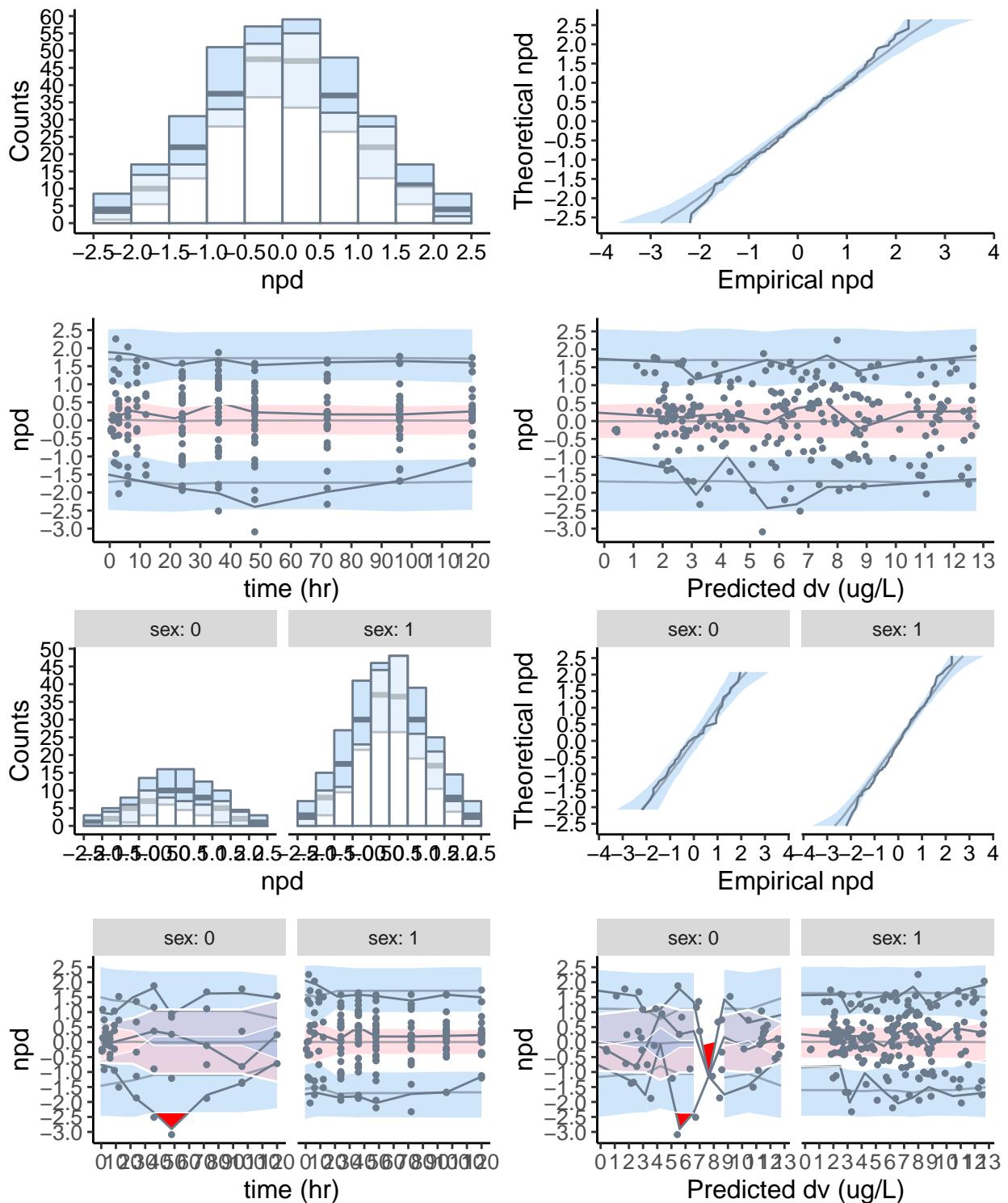
```
##  
## [[2]]
```

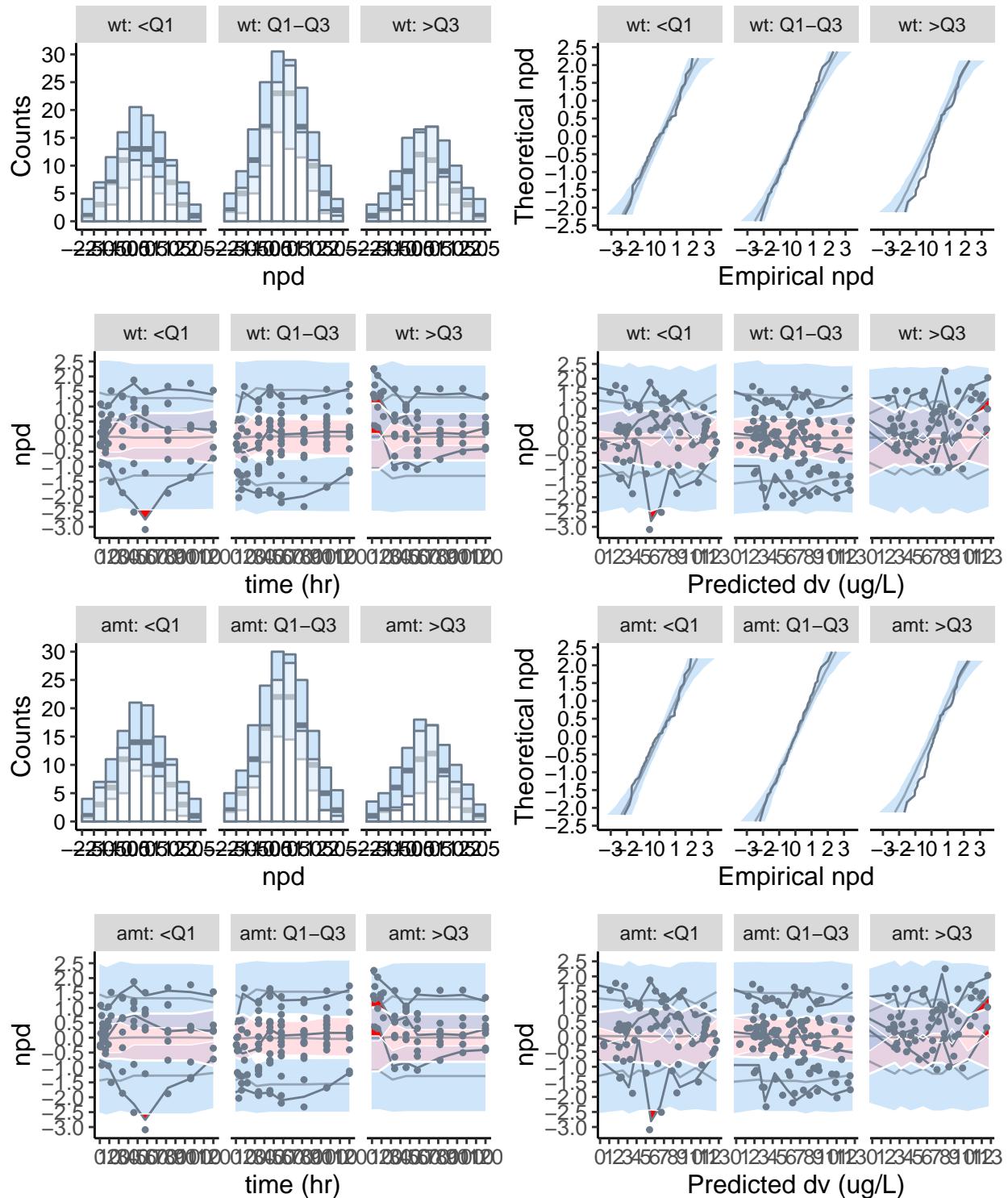


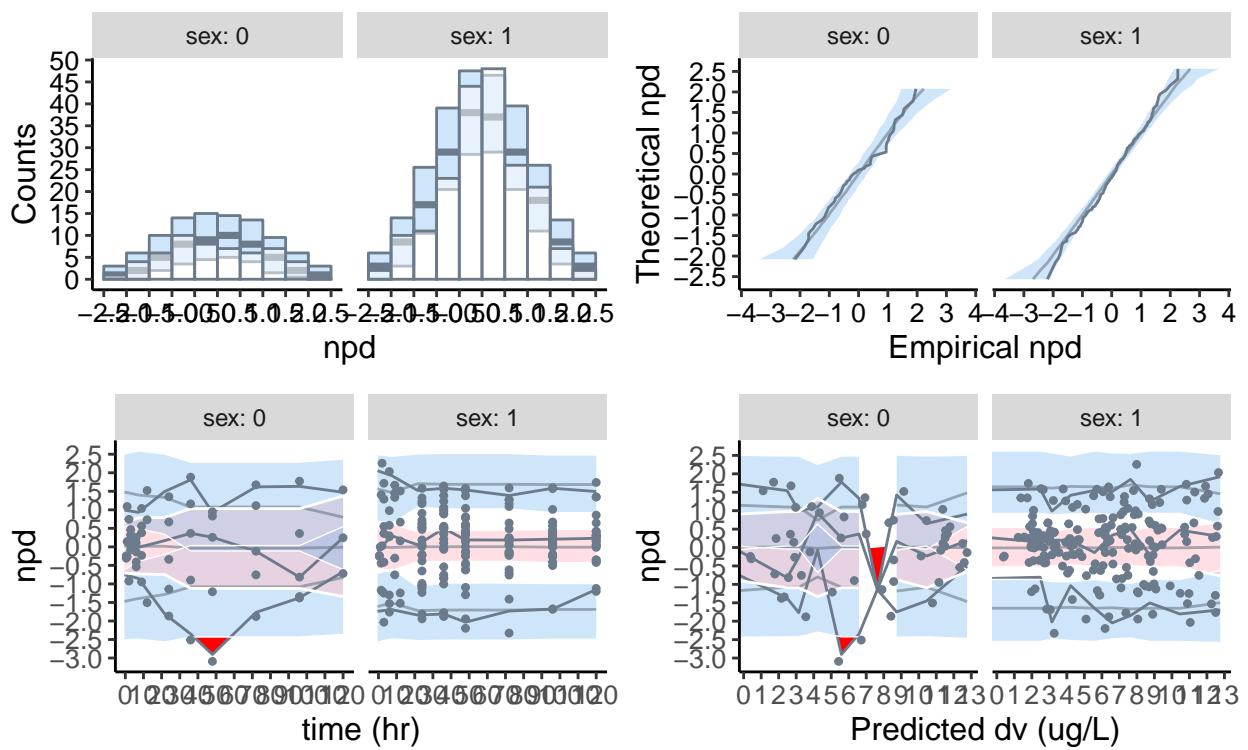
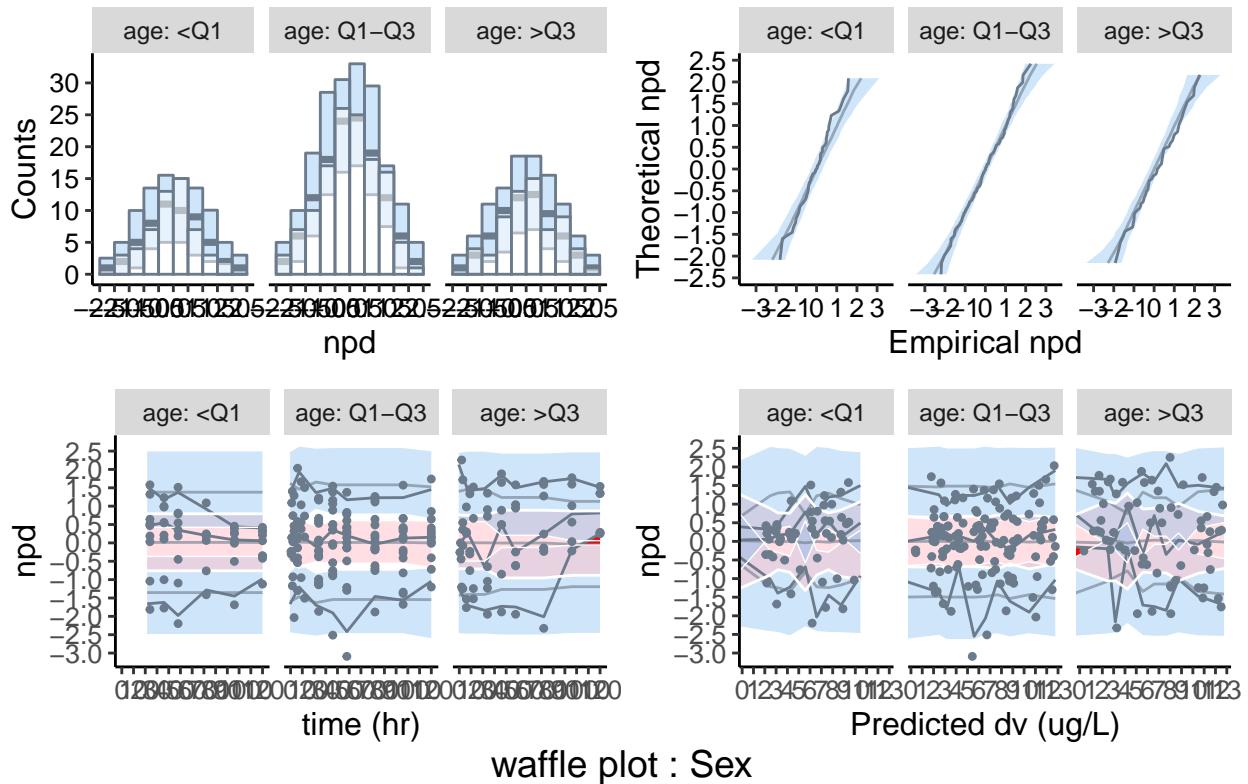




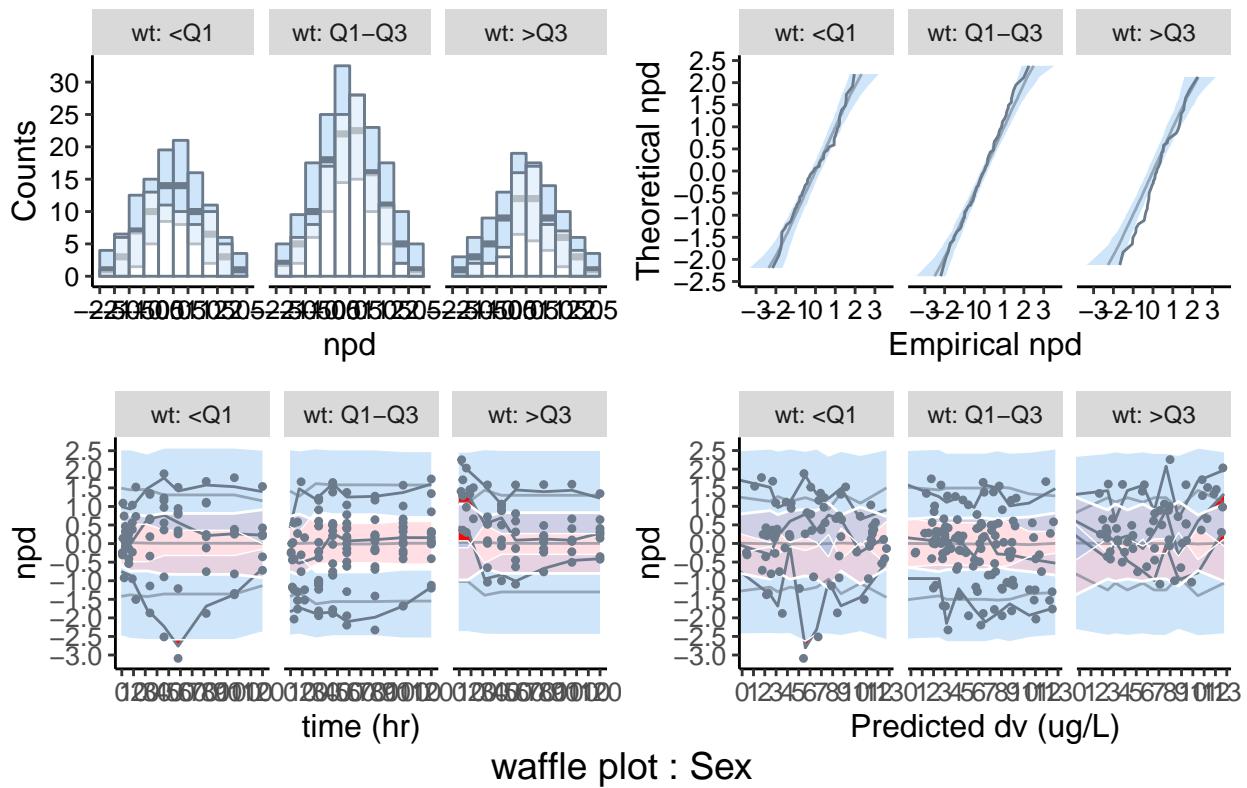




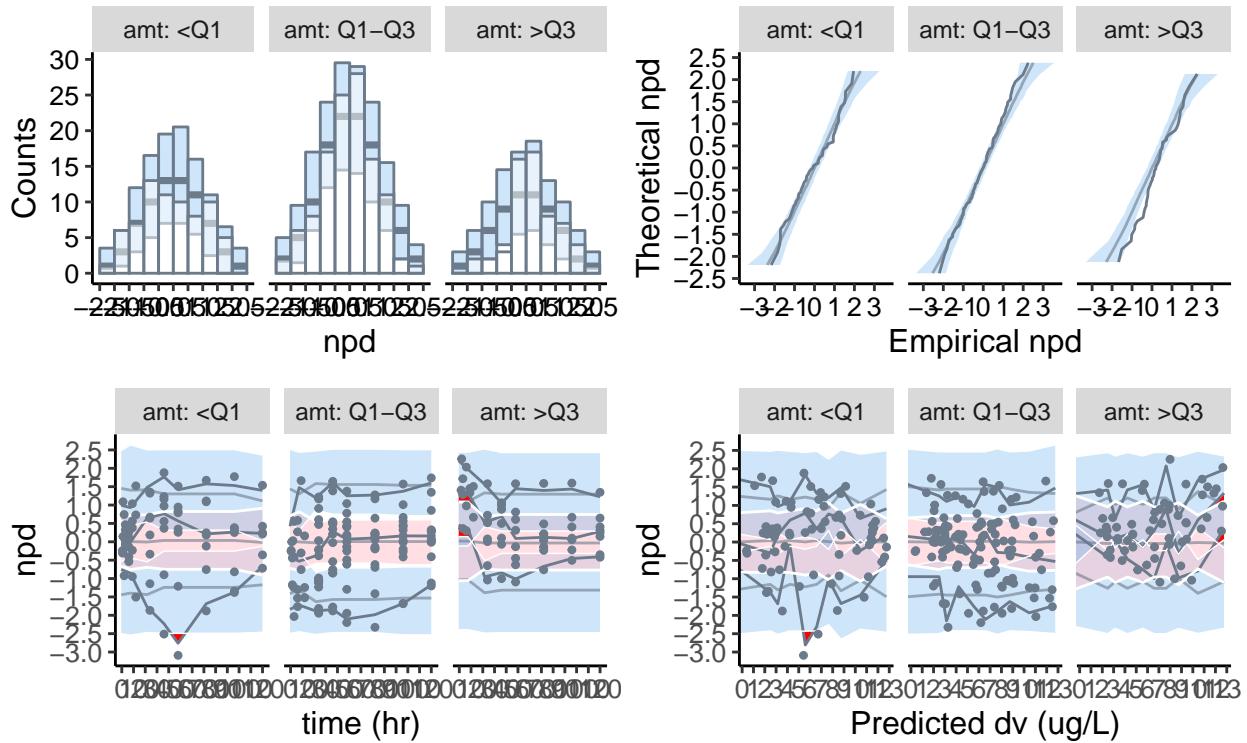




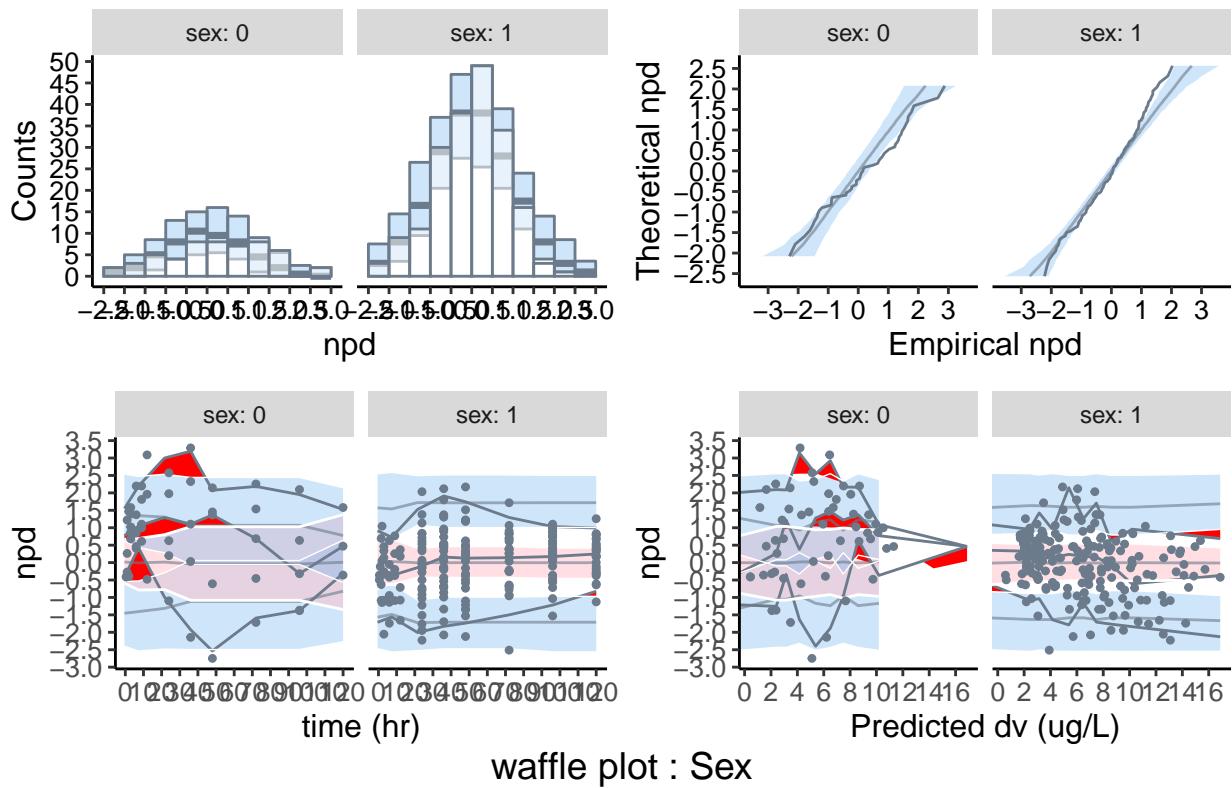
waffle plot : Sex



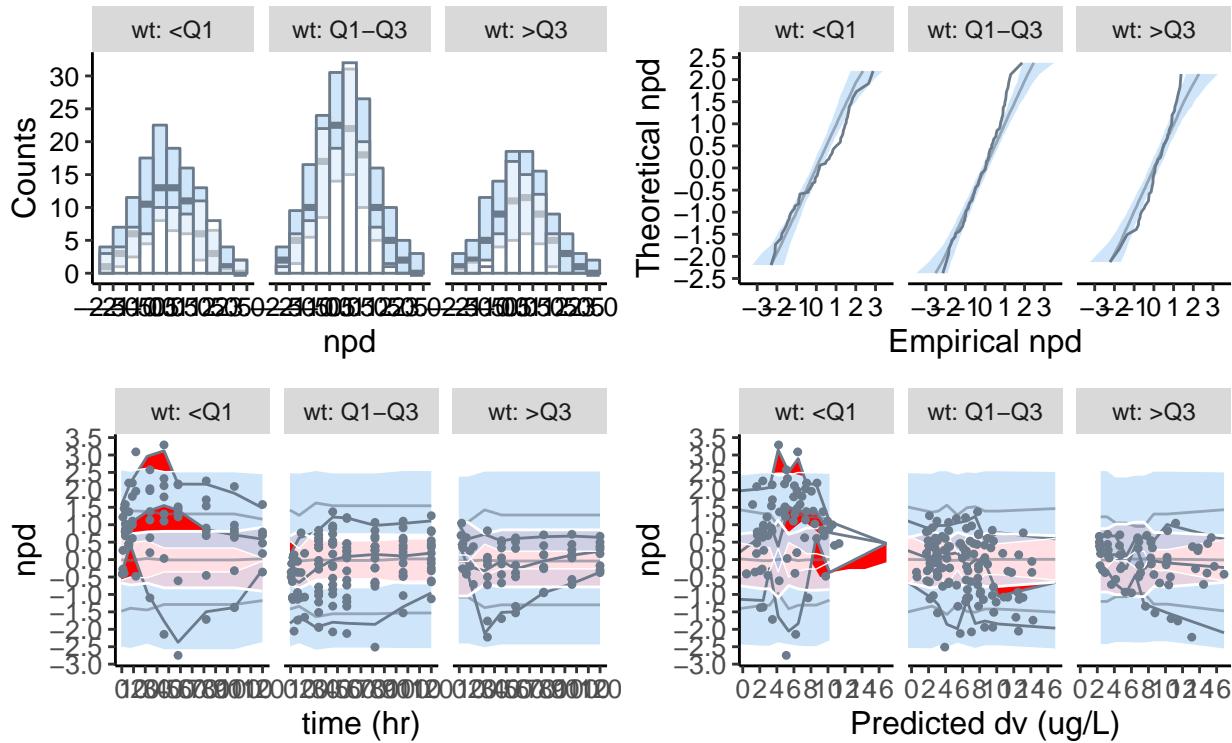
waffle plot : Sex



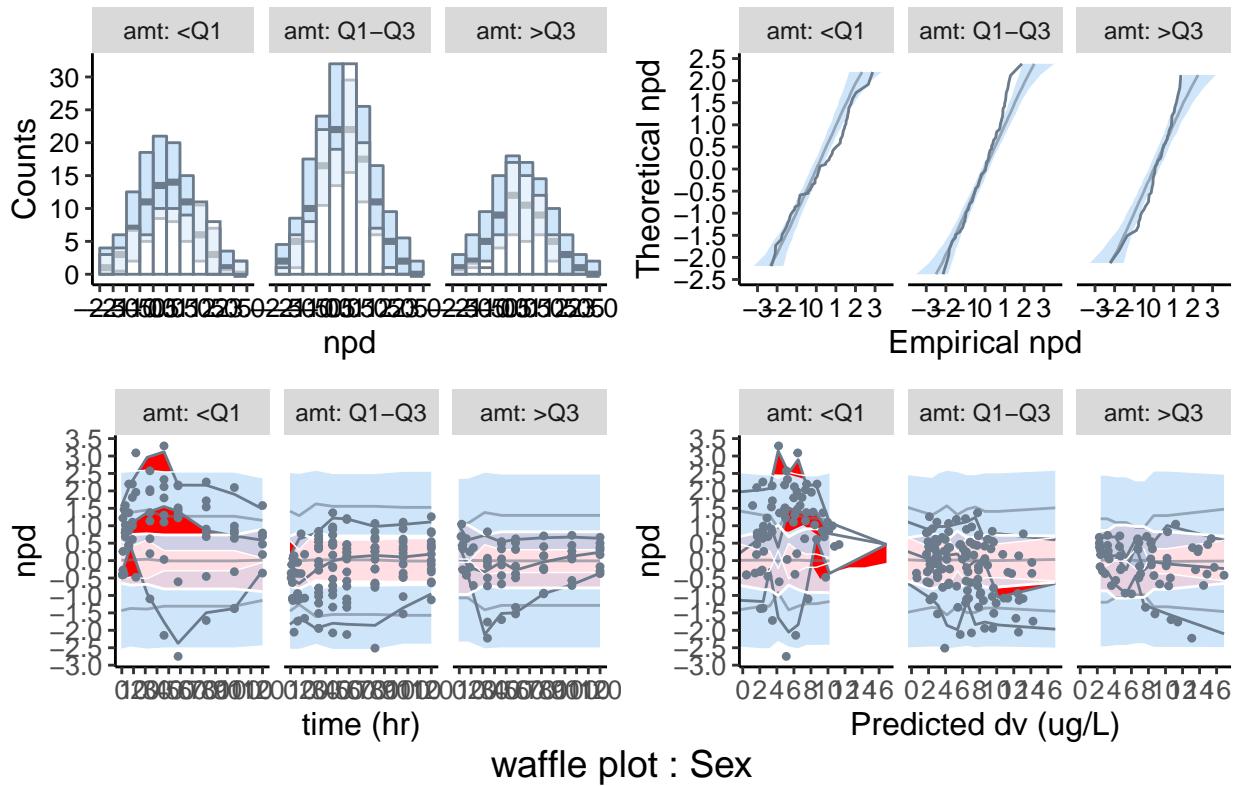
waffle plot : Sex



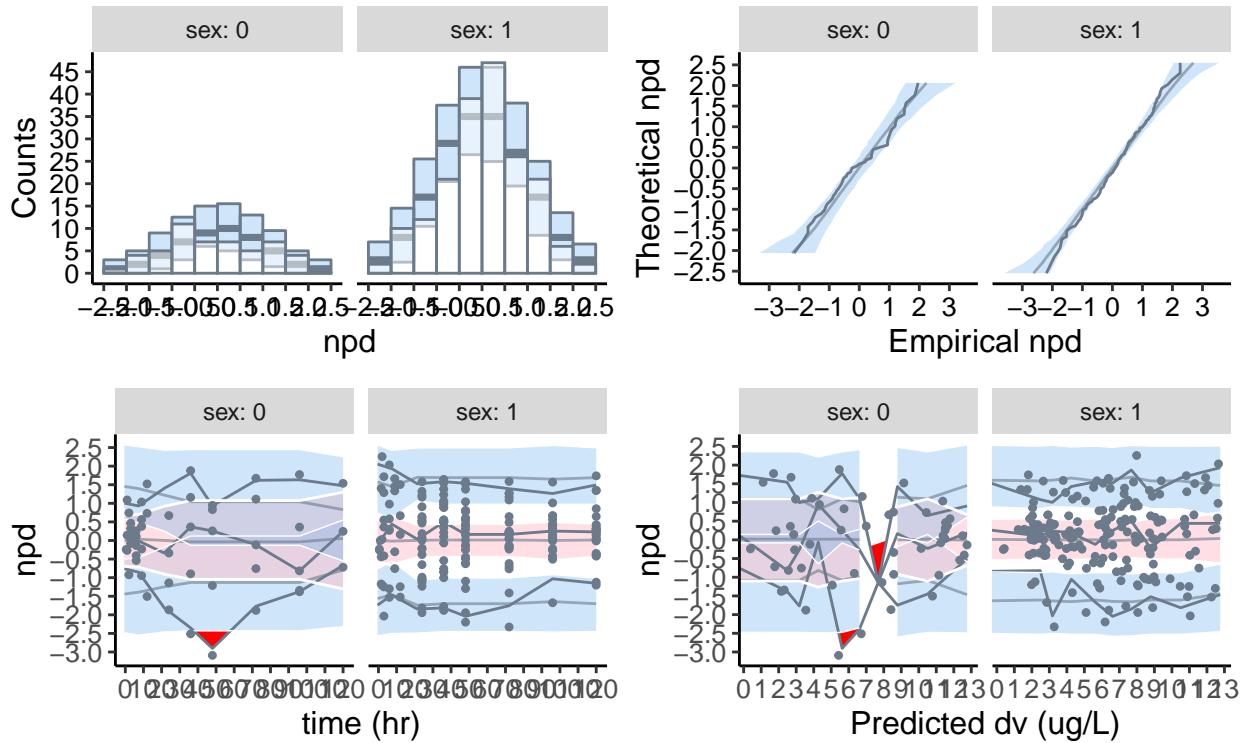
waffle plot : Sex



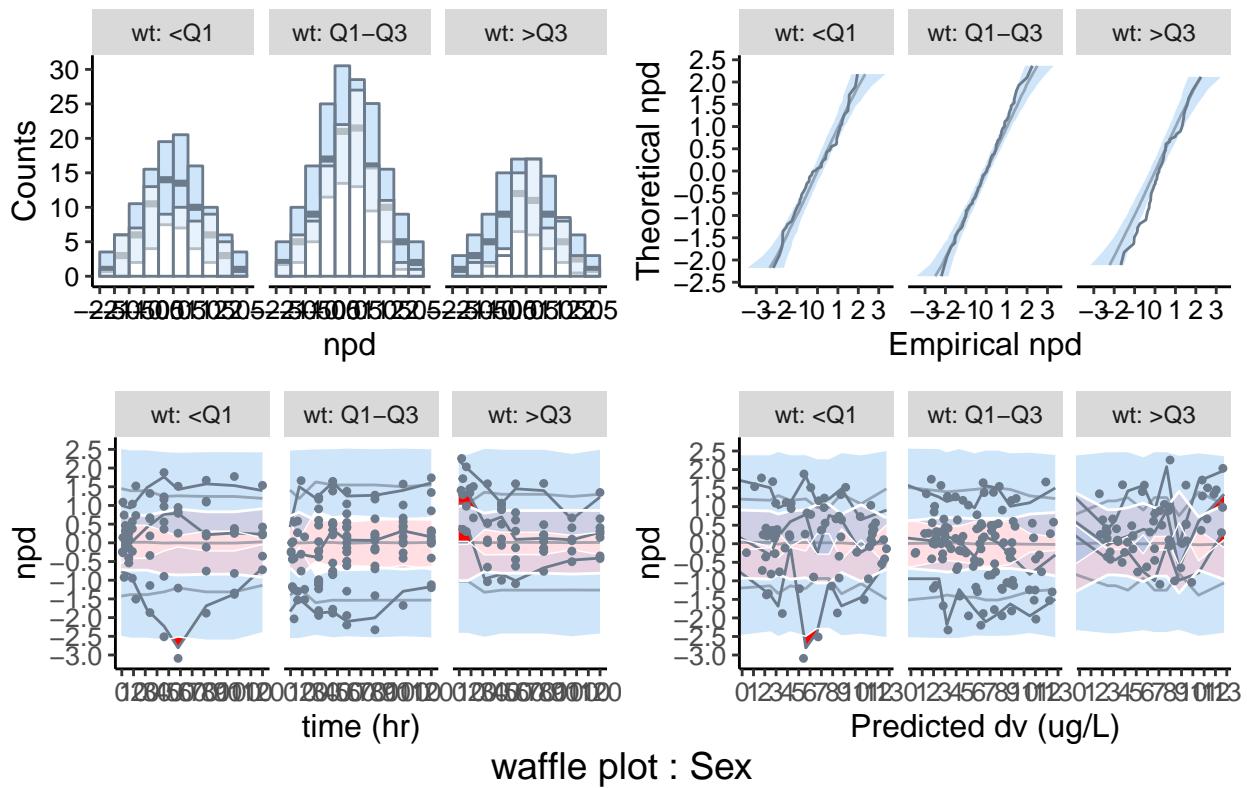
waffle plot : Sex



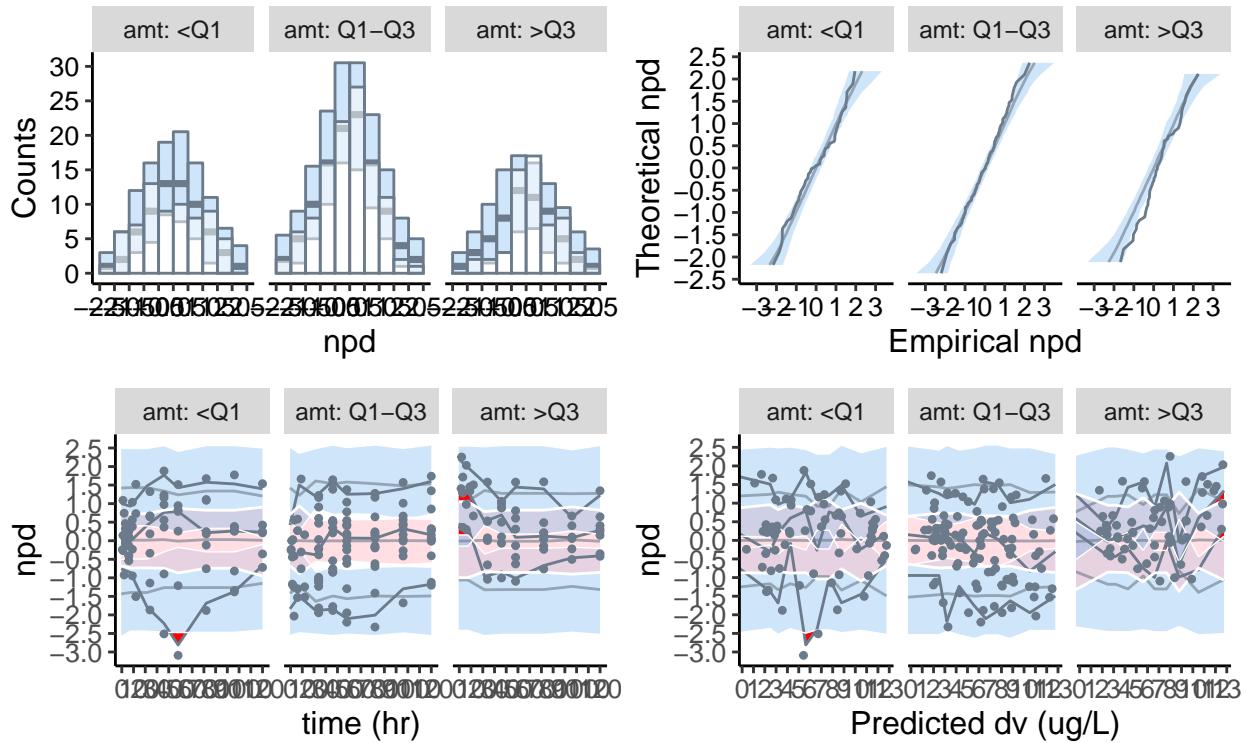
waffle plot : Sex



waffle plot : Sex



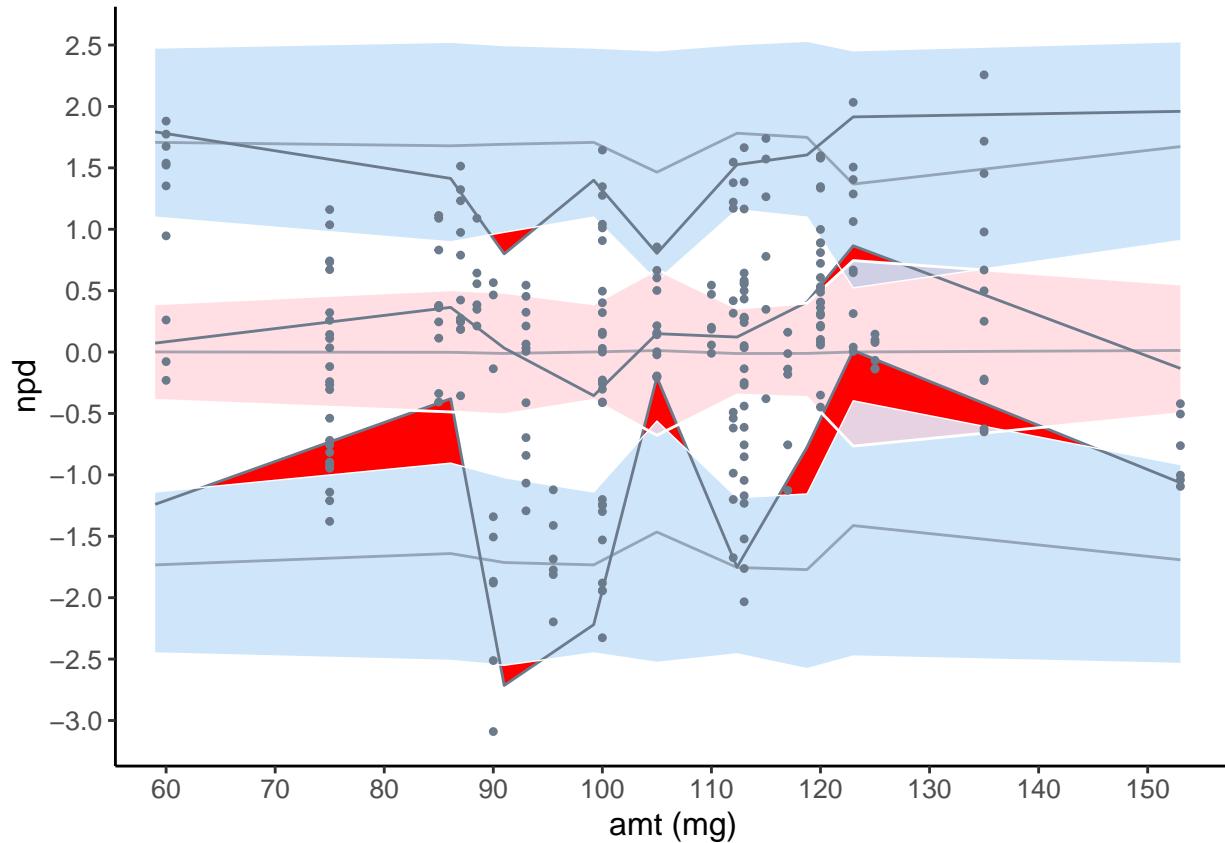
waffle plot : Sex



Parameters versus covariates

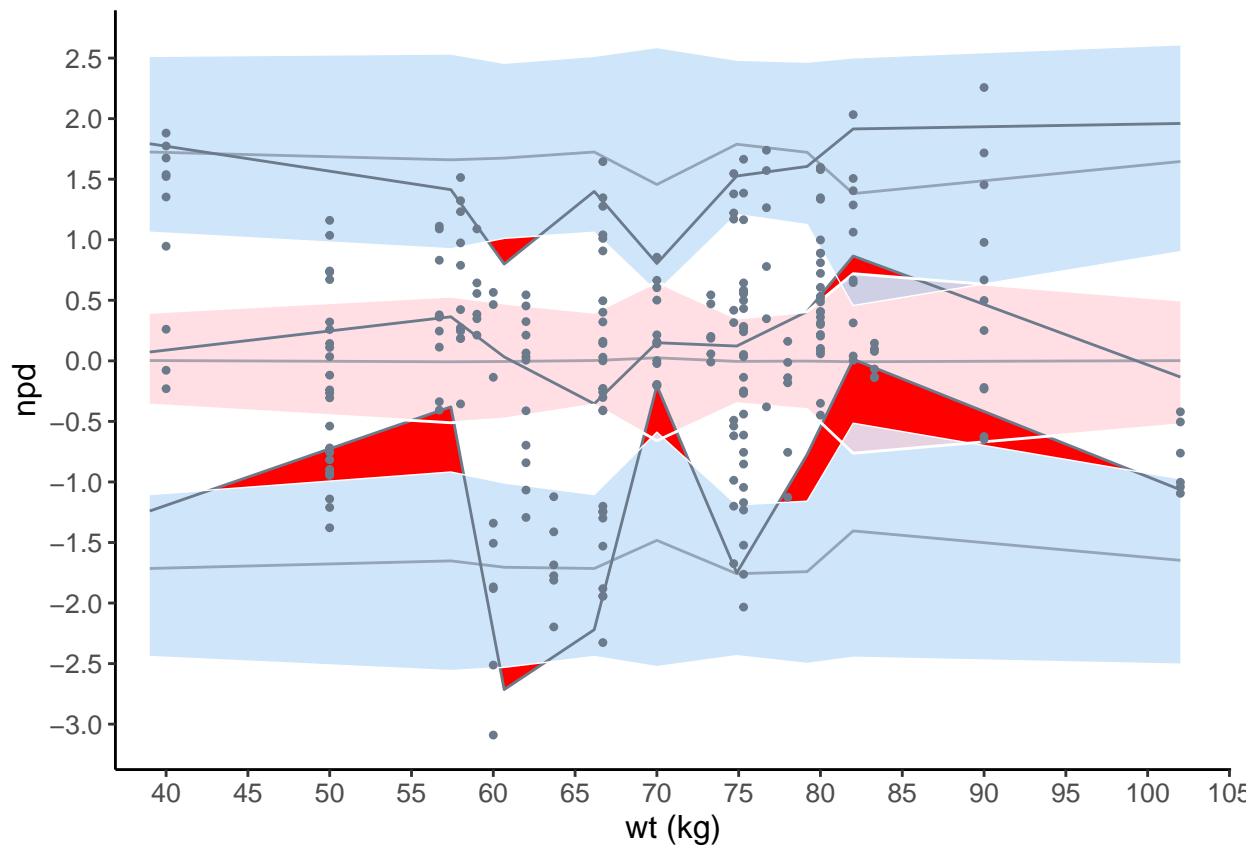
- **Problems (à résoudre Romain)**
 - solved yes here we really need to extend PI to cover the whole X-range for x.scatter and pred.scatter (not VPC)
 - * solved just the PI (not the observed percentiles or outlier bands)
 - solved cov.scatter (eg npde versus covariates)
 - * boxplot for categorical covariates (overlaid on PI)
 - * here two different behaviours ?

```
## [[1]]
```

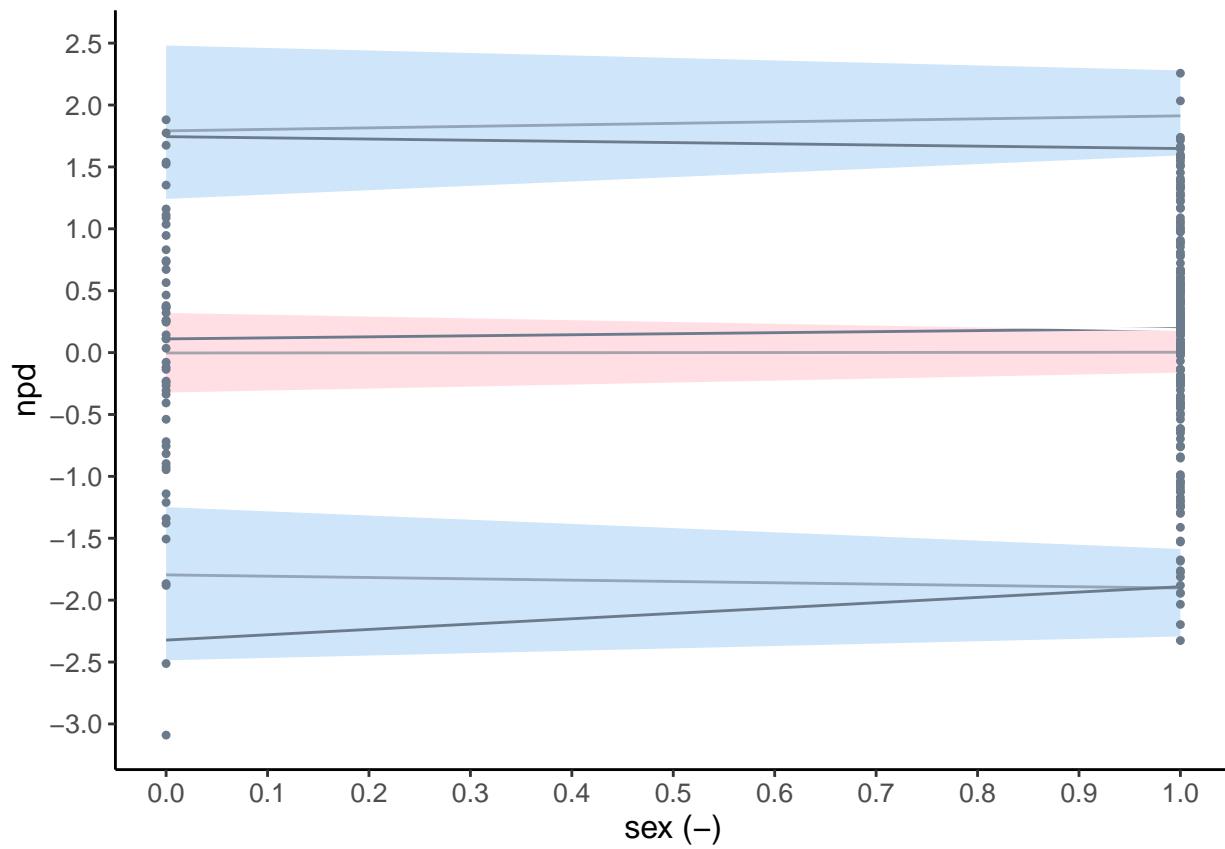


```
##
```

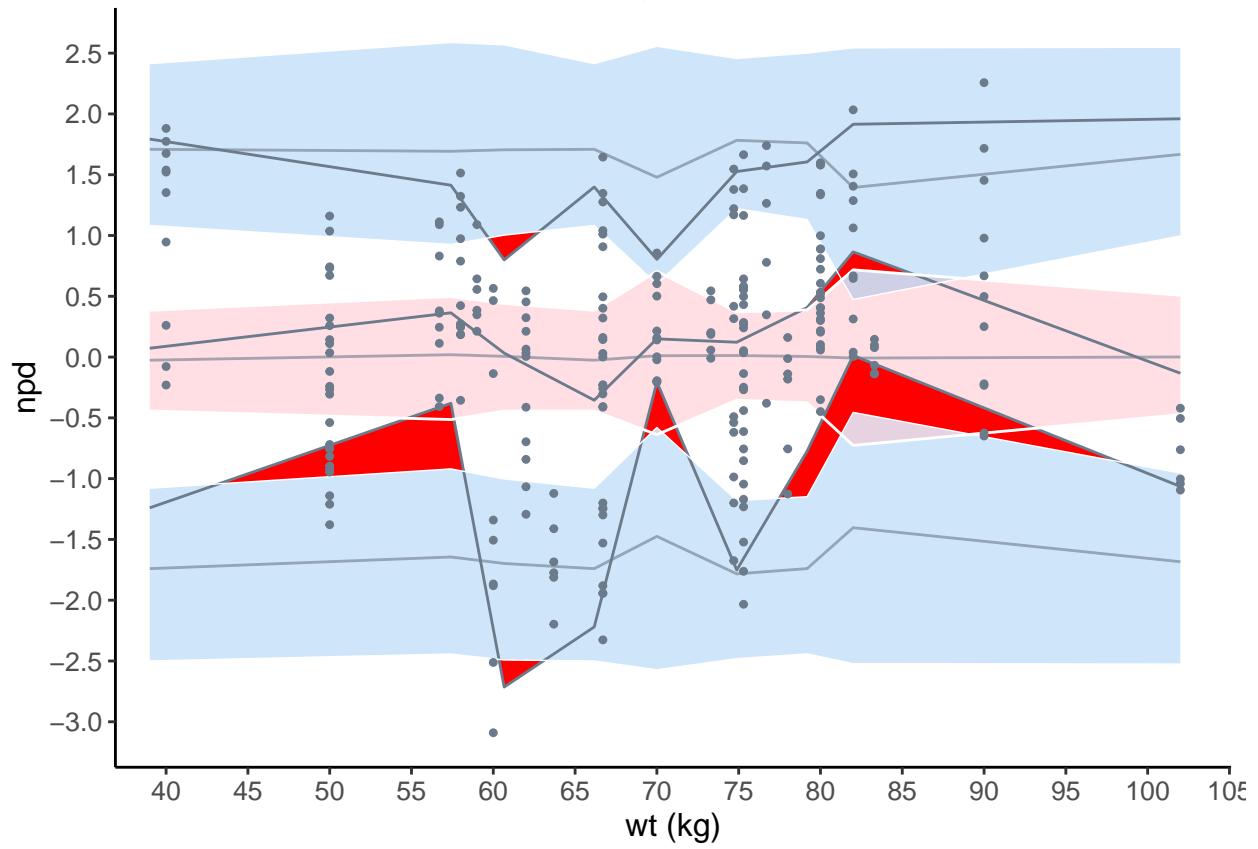
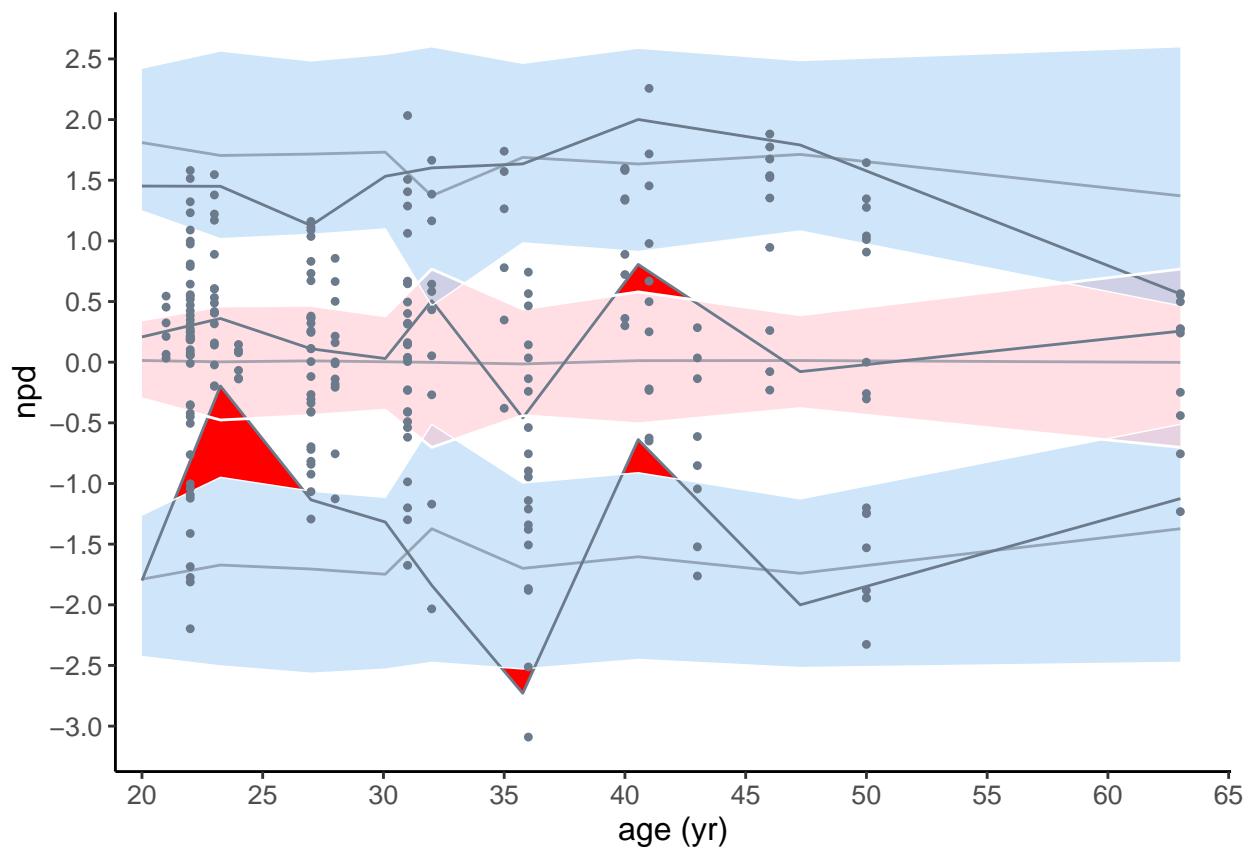
```
## [[2]]
```

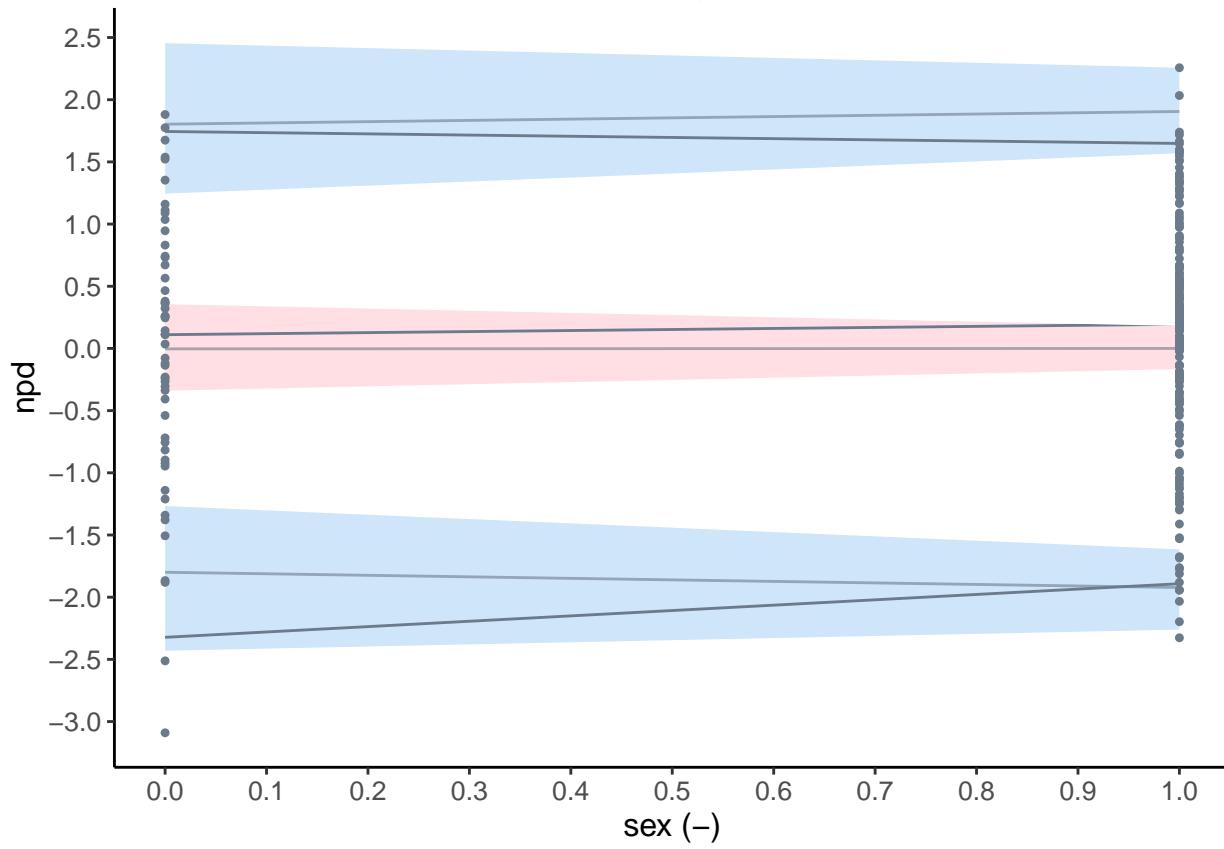
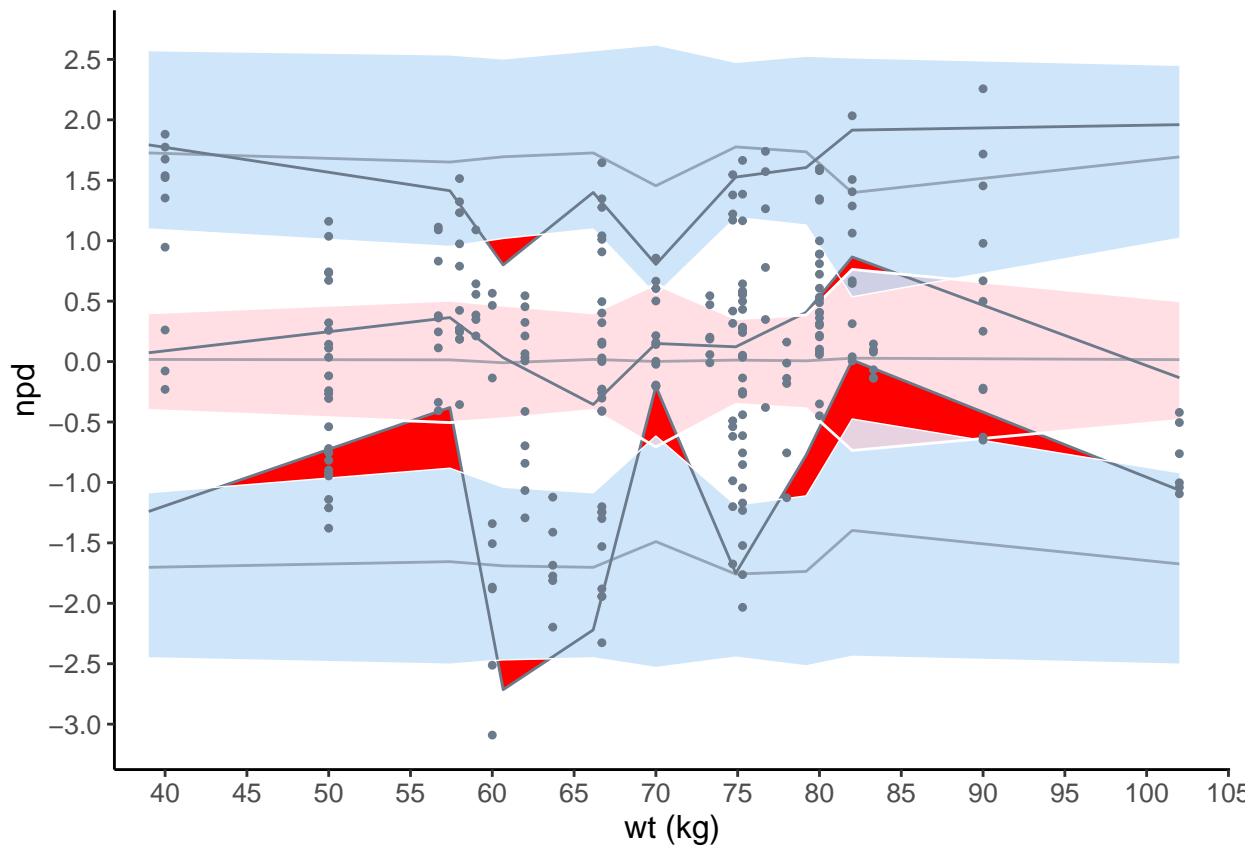


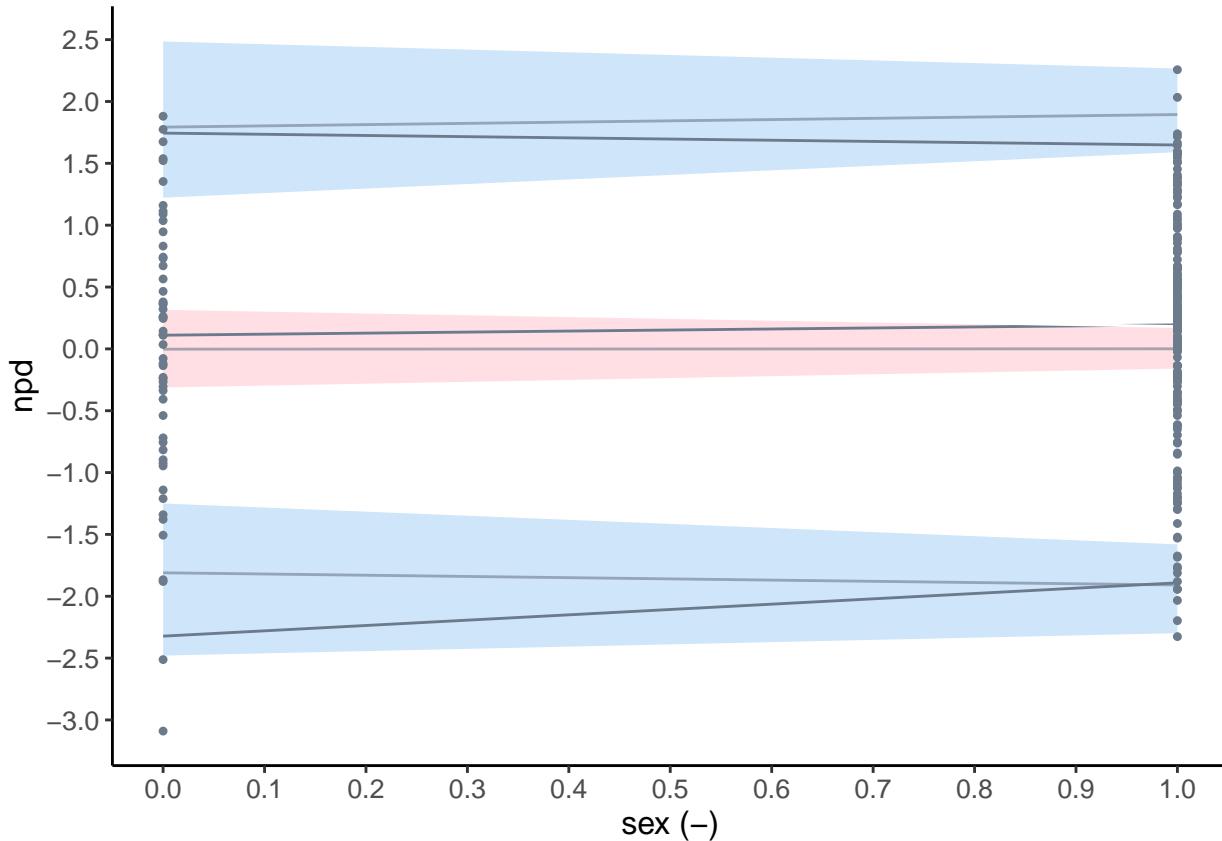
```
##  
## [[3]]
```



```
##  
## [[4]]
```







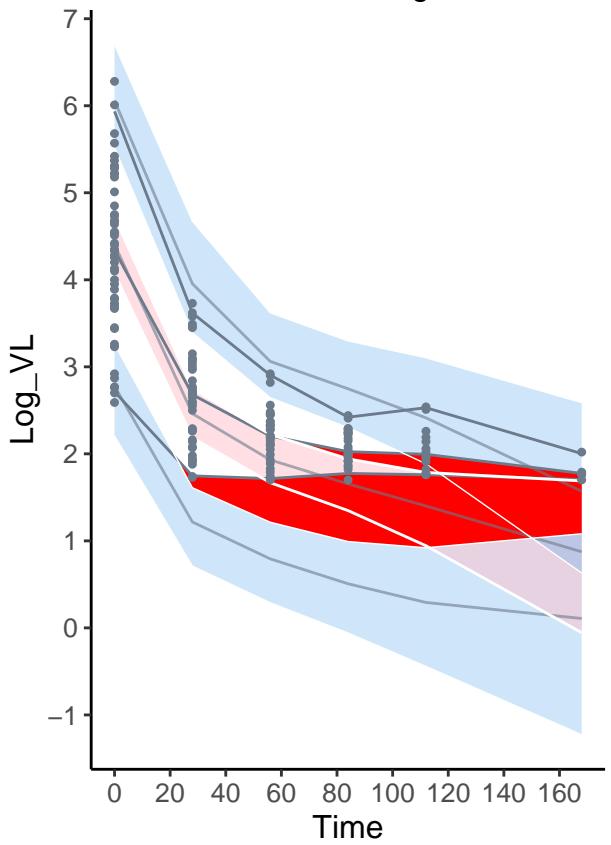
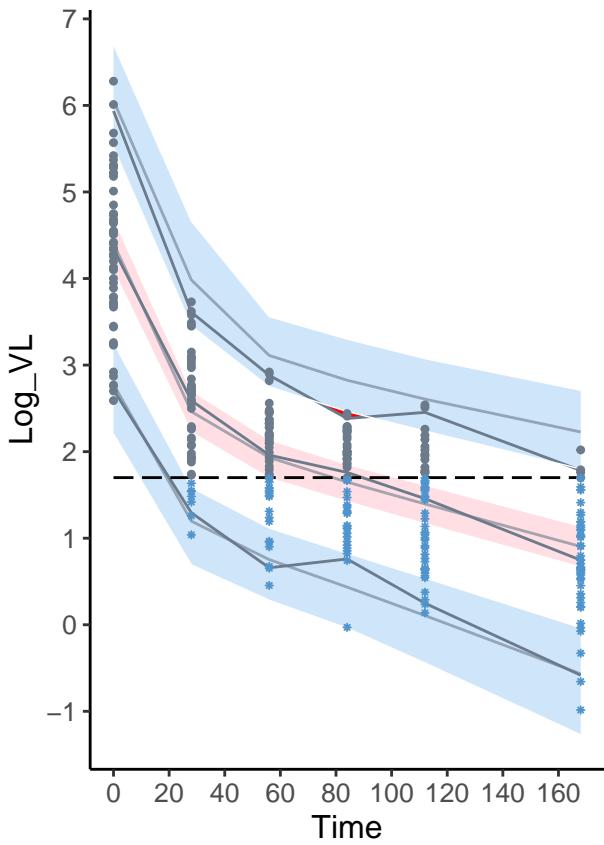
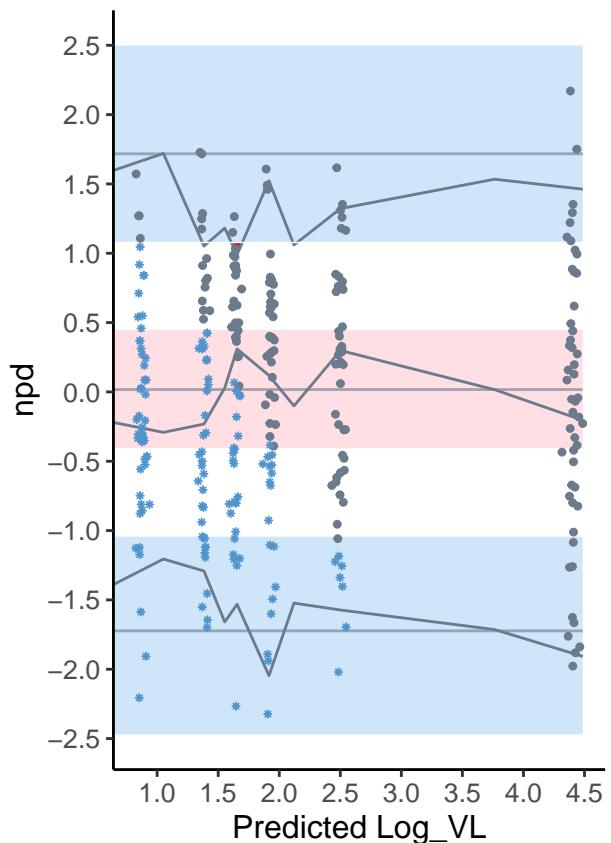
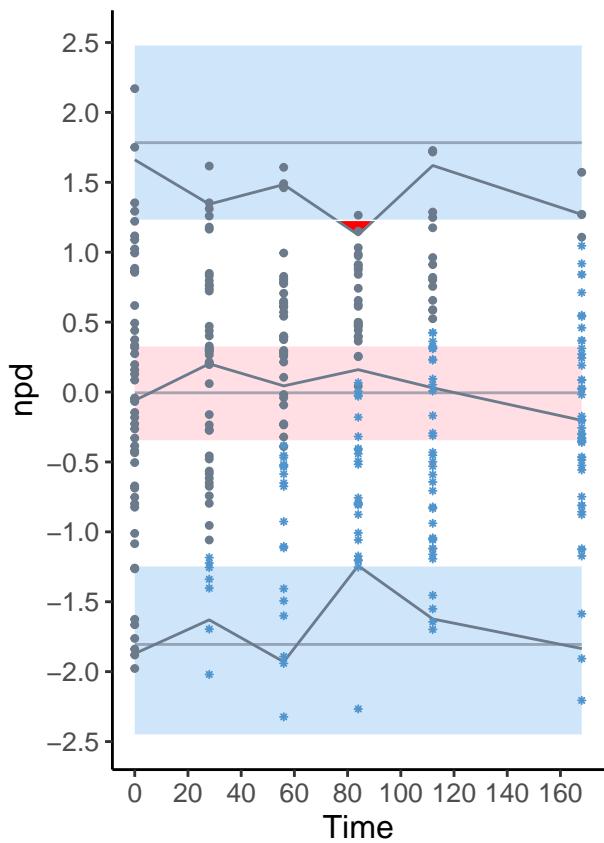
Creating the plots for the user guide

Theophylline

- Default plots and VPC: **OK pour doc**
- **Not solved Problem** with plot of data (see above)

Viral load data

- **TODO**
 - **solved** plot data
 - est-ce que les fonctions graphiques ‘de base’ pourraient renvoyer les graphes comme objet pour pouvoir ensuite en faire un *grid.arrange()* (on avait parlé de cette option pour pouvoir faire ensuite ses propres graphes comme avant) ?
- **Note** - ok mais pour le moment que pour : hist, qqplot, x.scatter, pred.scatter, vpc, ecdf, data
 - le pipeline est le suivant : fonction auxiliaire plot -> fonction plot -> plot dans plotNpde-methods
 - * **solved** 2 graphes de données (voire 3)
 - * **solved** 2 graphes de VPC côté à côté (=> donner code correspondant dans la doc)
 - * **Eco** modifié car le code plantait, toujours ce mélange de [[1]]



Note - Pour modifier les graphes directement avec un template ggplot. Le theme d'un plot ggplot est

accessible via le slot theme et layers, exemple : *plot.vpc.yvir50theme* et *plot.vpc.yvir50layers*. On peut ainsi définir un template ggplot par défaut pour chaque plot et créer une fonction qui va directement le modifier en fonctions des options utilisateurs. Comme *plot.vpc.yvir50theme* * et * *plot.vpc.yvir50layers* sont des listes, avec modifyList les slots sont facilement accessibles et modifiables. Dans les fonctions plots pourraient être ainsi appelées un theme tels que *theme_plot_vpc()*, *theme_plot_scatter()*, etc.

End of file, deactivating development mode

```
## v Dev mode: OFF
```