

# Test package with devtools

Emmanuelle

30/11/2020

## Notes before release

- package
  - **TODO** automatically load ggplot libraries when loading *npde* ()
  - changed rbindlist to do.call so we don't need to depend on data.table package
    - \* **Romain**: peut-être réécrire les parties du code générant pimat pour ne pas avoir de liste et faire directement des rbind quand on génère les pimat pour chaque catégorie de covariables
- 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** remove remifentanil example as much too large in size for CRAN => maybe replace with warfarine covariate model (add doc for these as currently raises a warning)
- code **TODO**
  - add an option to remove all output when required and remove warnings if requested
  - modify name of npde.plot.meanprofile to something else like npde.plot.scatter (not that, as we already have one)
- *npde* website (bookdown, Eco)
- **TODO** simplify output for *show()* function (v3.0 or v3.1 ?)

## Setup, loading libraries

### Install package in development mode

```
setwd(workDir)
dev_mode() # development mode

## Dev mode: ON
install.packages(pkgs=file.path(workDir, "npde_3.0.tar.gz"), repos=NULL)

## Installing package into 'C:/Users/Romai/OneDrive/Documents/R-dev'
## (as 'lib' is unspecified)
library(npde)
```

## Running examples from documentation

### Checking help files work

OK

```
## starting httpd help server ... done
```

## Computing npde for theophylline data

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

```
## -----
## 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
##     t-test             : 0.481
##     Fisher variance test : 0.55
##     SW test of normality    : 0.00273 **
##     Global adjusted p-value    : 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)
## -----
## ----- 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      tnpde
## 2  2.923864  2.84 0.550 -0.05124648 0.1256613  5.403817
## 3  4.682299  6.57 0.850  1.96398150 2.0537489 11.302178
## 4  6.264357 10.50 0.990  2.56602650 2.3263479 12.136107
## 5  6.986255  9.66 0.980  0.41616411 0.5244005  6.623631
## 6  6.511039  8.58 0.930  0.28430866 0.2533471  5.794430
## 7  5.895675  8.36 0.960  0.54879386 0.6744898  7.082780
```

```

## 8 5.064736 7.47 0.970 1.79335938 1.6448536 10.051295
## 9 4.302909 6.89 0.990 0.80506269 0.7721932 7.381673
## 10 3.294020 5.94 0.995 1.91537662 1.7506861 10.375055
## 11 1.168743 3.28 0.995 3.25535923 2.5758293 12.899315

## -----
## 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
##   t-test : 0.481
##   Fisher variance test : 0.55
##   SW test of normality : 0.00273 **
##   Global adjusted p-value : 0.00819 **
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## -----
##      ypred ycomp pd      ydobs      npde      tnpde
## 1    NaN  NaN  NA      NA  NA  NA
## 2 2.923864 2.84 0.55 -0.05124648 0.1256613 5.403817
## 3 4.682299 6.57 0.85 1.96398150 2.0537489 11.302178
## 4 6.264357 10.50 0.99 2.56602650 2.3263479 12.136107
## 5 6.986255 9.66 0.98 0.41616411 0.5244005 6.623631
## 6 6.511039 8.58 0.93 0.28430866 0.2533471 5.794430

## 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)
## -----
## ----- 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    tnpde
## 2  2.923864  2.84 0.550 -0.05124648 0.1256613  5.403817
## 3  4.682299  6.57 0.850  1.96398150 2.0537489 11.302178
## 4  6.264357 10.50 0.990  2.56602650 2.3263479 12.136107
## 5  6.986255  9.66 0.980  0.41616411 0.5244005  6.623631
## 6  6.511039  8.58 0.930  0.28430866 0.2533471  5.794430
## 7  5.895675  8.36 0.960  0.54879386 0.6744898  7.082780
## 8  5.064736  7.47 0.970  1.79335938 1.6448536 10.051295
## 9  4.302909  6.89 0.990  0.80506269 0.7721932  7.381673
## 10 3.294020  5.94 0.995  1.91537662 1.7506861 10.375055
## 11 1.168743  3.28 0.995  3.25535923 2.5758293 12.899315

## 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
## First 10 lines of data:
##   index ID  Time  Conc mdv
## 1 1 1 0.00 NA 1
## 2 1 1 0.25 2.84 0
## 3 1 1 0.57 6.57 0
## 4 1 1 1.12 10.50 0
## 5 1 1 2.02 9.66 0
## 6 1 1 3.82 8.58 0
## 7 1 1 5.10 8.36 0
## 8 1 1 7.03 7.47 0
## 9 1 1 9.05 6.89 0
## 10 1 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 -----

```

```

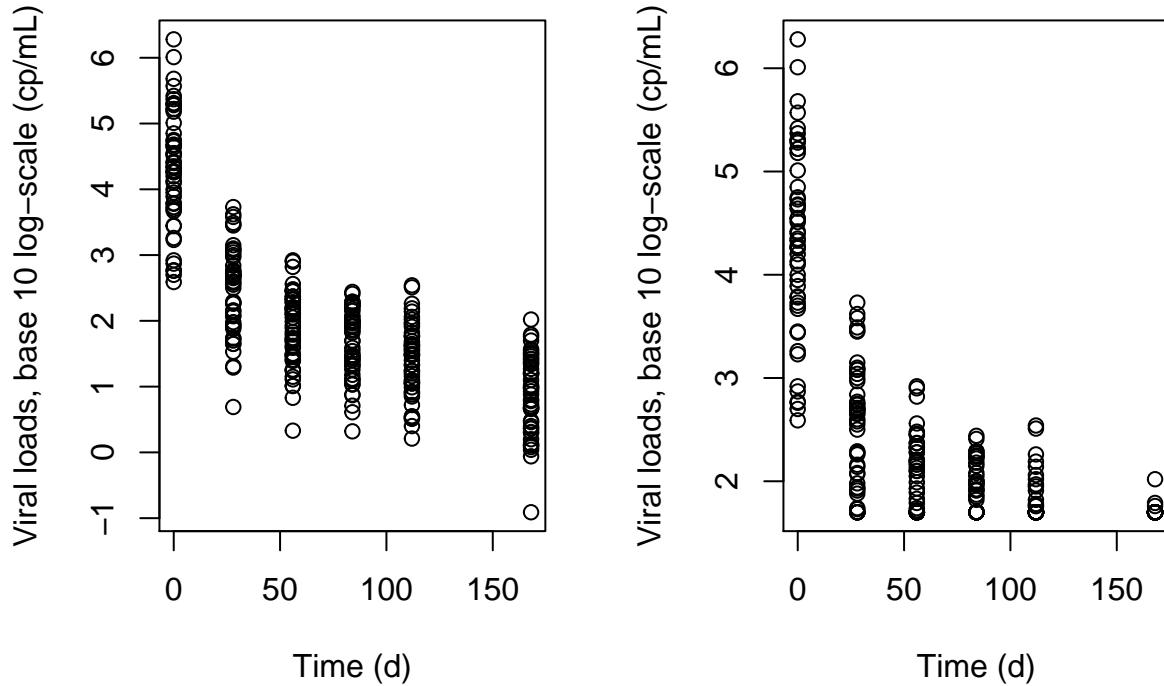
## -----
## Methods
##   compute prediction discrepancies (pd): yes
##   compute normalised prediction distribution errors (npde): yes
##   method for decorrelation: Cholesky decomposition (upper triangular)
##   method to treat censored data: Impute pd* and compute y* as F-1(pd*)
## Input/output
##   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.
## -----
## 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
##   t-test : 0.481
##   Fisher variance test : 0.55
##   SW test of normality : 0.00273 **
##   Global adjusted p-value : 0.00819 **
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## -----

```

### 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 omit and ppred method
  - **TODO** check why 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
##   t-test : 0.469
##   Fisher variance test : 0.032 *
##   SW test of normality : 0.845
##   Global adjusted p-value : 0.0959 .
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## 
## -----
## Distribution of npde :
##   nb of obs: 300
##   mean= -0.01751  (SE= 0.053 )
##   variance= 0.8465  (SE= 0.069 )
##   skewness= 0.009426
##   kurtosis= 0.02397
## 
## 
```

```

## Statistical tests
##   t-test : 0.742
##   Fisher variance test : 0.0506 .
##   SW test of normality : 0.773
## Global adjusted p-value : 0.152
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## -----
## -----
## Distribution of npde :
##   nb of obs: 300
##   mean= -0.05321 (SE= 0.055 )
##   variance= 0.8933 (SE= 0.073 )
##   skewness= -0.3228
##   kurtosis= 0.06921
## -----
## 
## Statistical tests
##   t-test : 0.33
##   Fisher variance test : 0.185
##   SW test of normality : 0.0162 *
## Global adjusted p-value : 0.0485 *
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## -----
## -----
## Distribution of npde :
##   nb of obs: 300
##   mean= -0.09788 (SE= 0.053 )
##   variance= 0.8371 (SE= 0.068 )
##   skewness= -0.08359
##   kurtosis= -0.1088
## -----
## 
## Statistical tests
##   t-test : 0.0649 .
##   Fisher variance test : 0.0372 *
##   SW test of normality : 0.613
## Global adjusted p-value : 0.112
## ---
## 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

```

```

##   t-test                  : 0.041 *
##   Fisher variance test    : 0.0822 .
##   SW test of normality    : 0.687
## Global adjusted p-value   : 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
##   t-test                  : 0.586
##   Fisher variance test    : 0.746
##   SW test of normality    : 0.00121 **
## Global adjusted p-value   : 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
##   t-test                  : 0.624
##   Fisher variance test    : 0.0539 .
##   SW test of normality    : 0.973
## Global adjusted p-value   : 0.162
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## -----

```

### Computing npde for remifentanil dataset with covariates

- **BUGS**

- columns 6 to 12 set as covariates but only 6:11 actually kept
- (à résoudre Romain) est-ce que c'est à cause de la fonction “`avoid_code`” rajoutée dans `setMethod(f="initialize",signature="NpdeData", ?` Cette fonction n'existe pas dans le code original.

- **correction**

- dans `NpdeData.R` : remove `avoid_code` and replace `nchar(string)==0` par `length(string)==0`

```

## -----
## Distribution of npde :
##   nb of obs: 1992

```

```

##          mean= -0.0318   (SE= 0.017 )
## variance= 0.5785   (SE= 0.018 )
## skewness= 0.2633
## kurtosis= 1.346
## -----
## 
## Statistical tests
## t-test                  : 0.0622 .
## Fisher variance test   : 2.38e-56 ***
## SW test of normality    : 2.14e-15 ***
## Global adjusted p-value : 7.13e-56 ***
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## -----

```

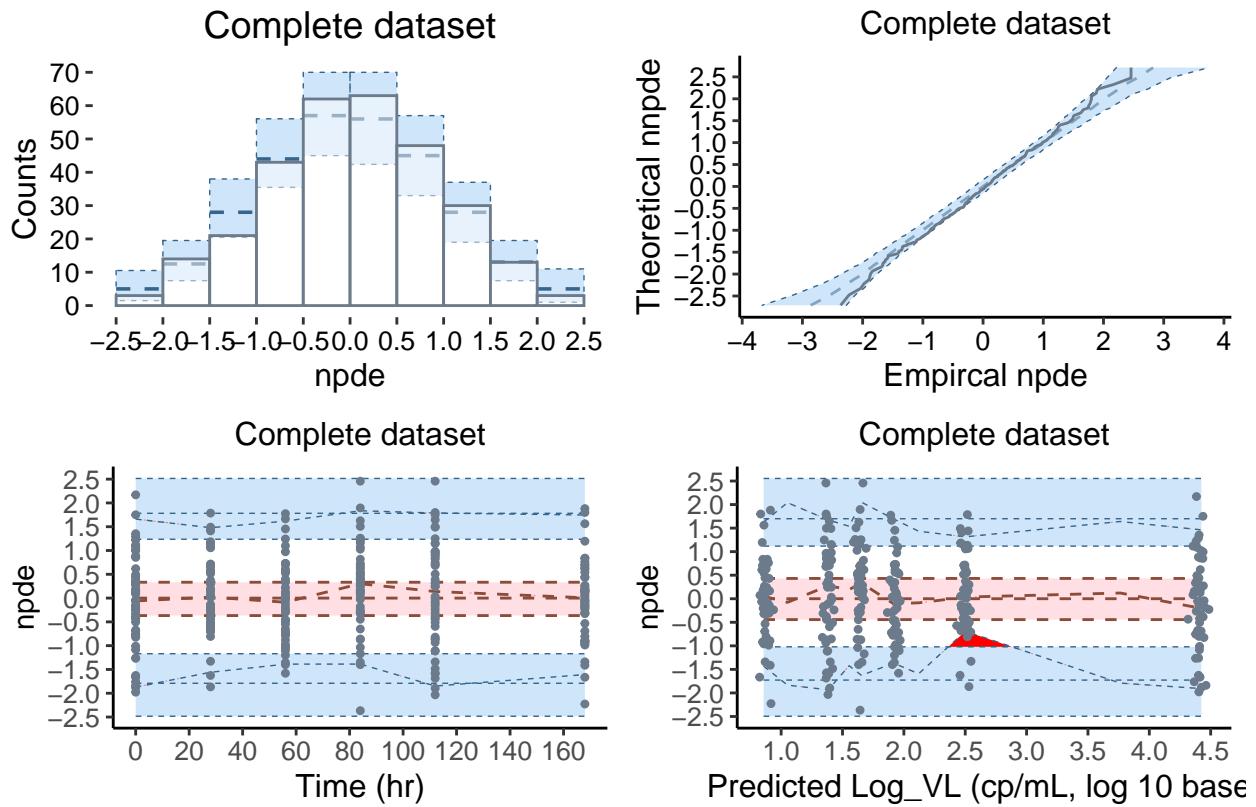
## Test plots - default options

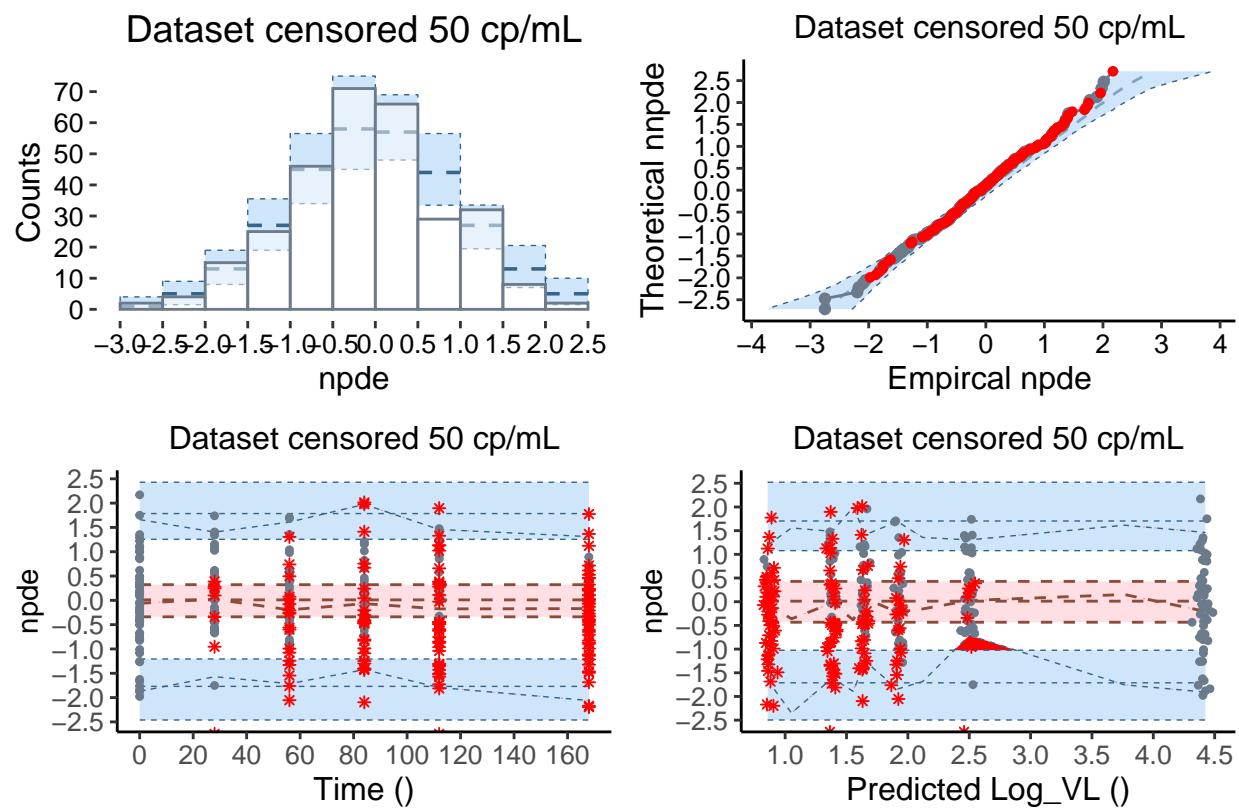
### Default plots for theophylline data

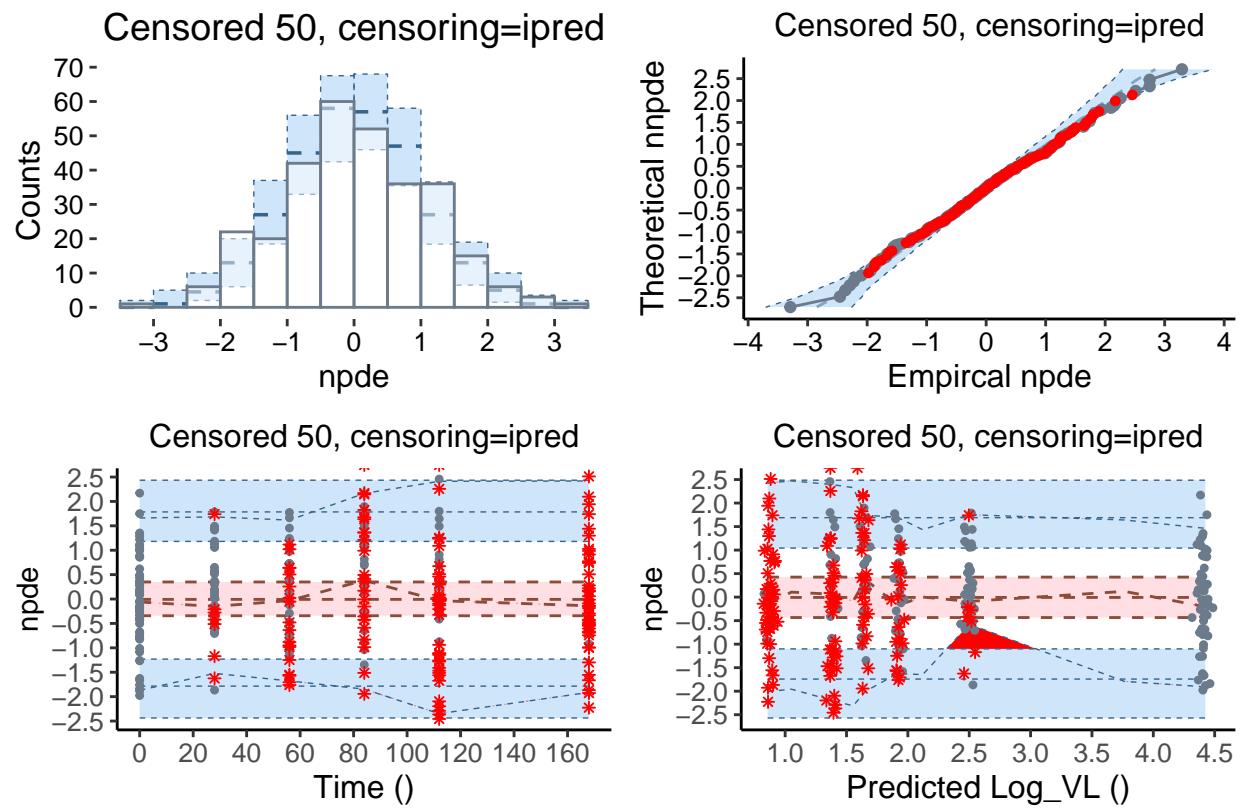
- **Corrections** (changed Romain)
  - option *which* passe (yes !)
  - titre des axes dans les graphes: marchent et s'adaptent au cas pd/npd/npde (cool)
  - corrigé le problème des limites d'axes pour npde vs pred
- **Corrections** (changed Eco)
  - default size for QQ-plot reduced for consistency in the default plot 2x2 (multiplied psizes.pobs and psizes.pcens by 2/3 inside the aux.plot.dist function, after the plot.opt vector is set, and added a note in the documentation) **check**
  - **but** a bit weird, when scatterplots are done individually the size seems very small (should be 1.5) and smaller than the scaling from the default plots appears to be... => plutôt un problème de ggplot2, on n'y peut rien :-/
- **Problems (à résoudre Romain)**
  - sécuriser les erreurs quand on référence un **argument non autorisé**
    - \* ex: *which='pde'*, semble corrigé ? (semble ne plus planter mais continue à donner un message d'erreur lié à un objet introuvable)
    - \* *note*: pas nécessaire d'avoir *pde*, mais nécessaire d'empêcher que ça plante :-) (ie capturer le *which* et tester qu'il correspond à soit *pd*, soit *npd*, soit *npde*, et faire ça pour tous les arguments à choix multiples)
  - **unités**: quand l'élément *units* est vide pour *x* et/ou *y*, on met juste le "nom" et pas "nom ()"
  - **titre** à centrer au-dessus des 4 graphes et pas seulement du premier (mais super qu'il y ait un titre :-) )
    - \* comme ça le graphe histogramme ne sera pas aplati par rapport aux autres
  - **global options** don't pass on to the functions
    - \* *size*
      - 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
      - ni dans le graphe par défaut (ie les 4 graphes), ni individuellement
    - \* *colours*
      - 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*
      - TBD: *col* => *col.pobs*, *col.lobs*, *col.pcens* ?
      - *col.lobs* ne change rien, devrait changer les lignes correspondant aux percentiles et les lignes dans l'histogramme
  - **X-range on PI** est-ce qu'on étend les bandes des PI jusqu'aux bornes du graphe (eg graphe des *npd*) ?

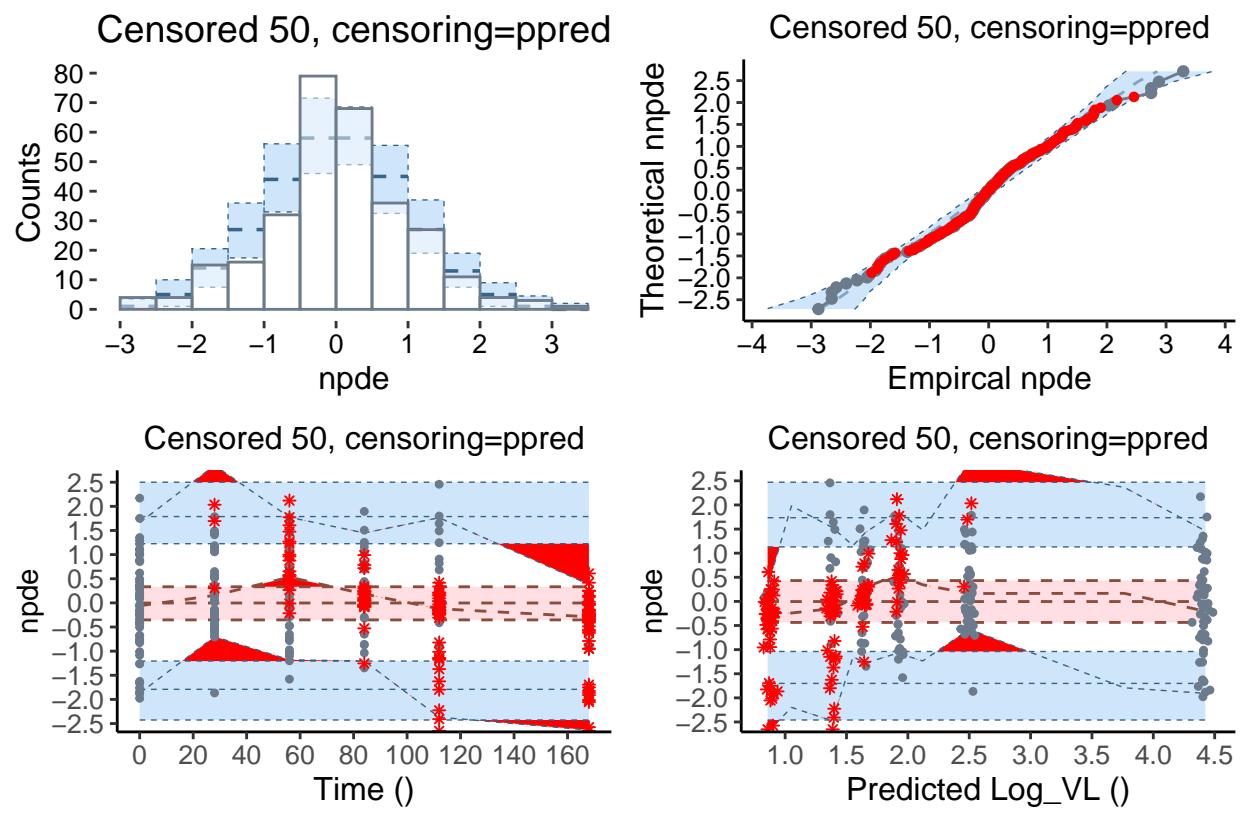
## Default plots for viral load data

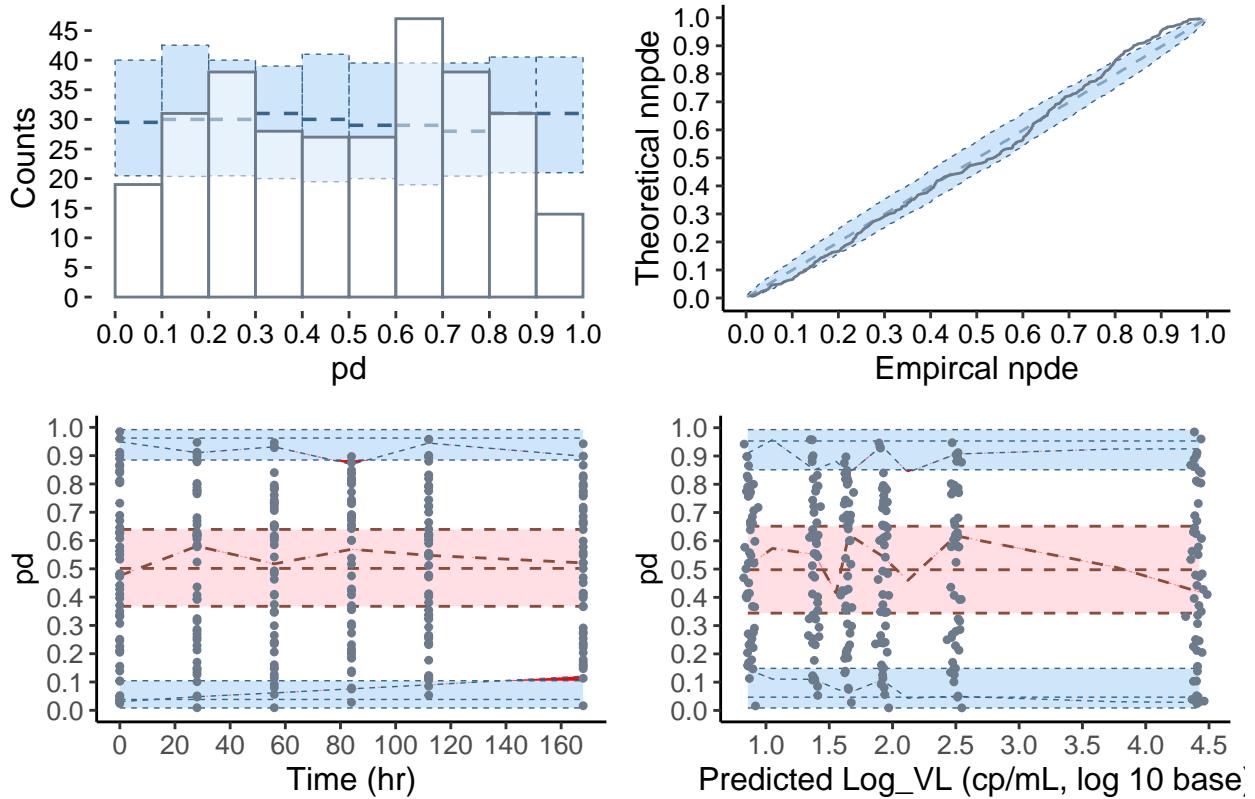
- computation of the PI: seems appropriate in the different cases
- Problems (Eco)**
  - PI for omit aren't consistent with the VPC later on
    - \* **documentation** explain how PI are computed in the omit case
  - don't work \***
- plot(yvir50.omit, main="Censored 50, censoring=omit") # omitting LOQ data







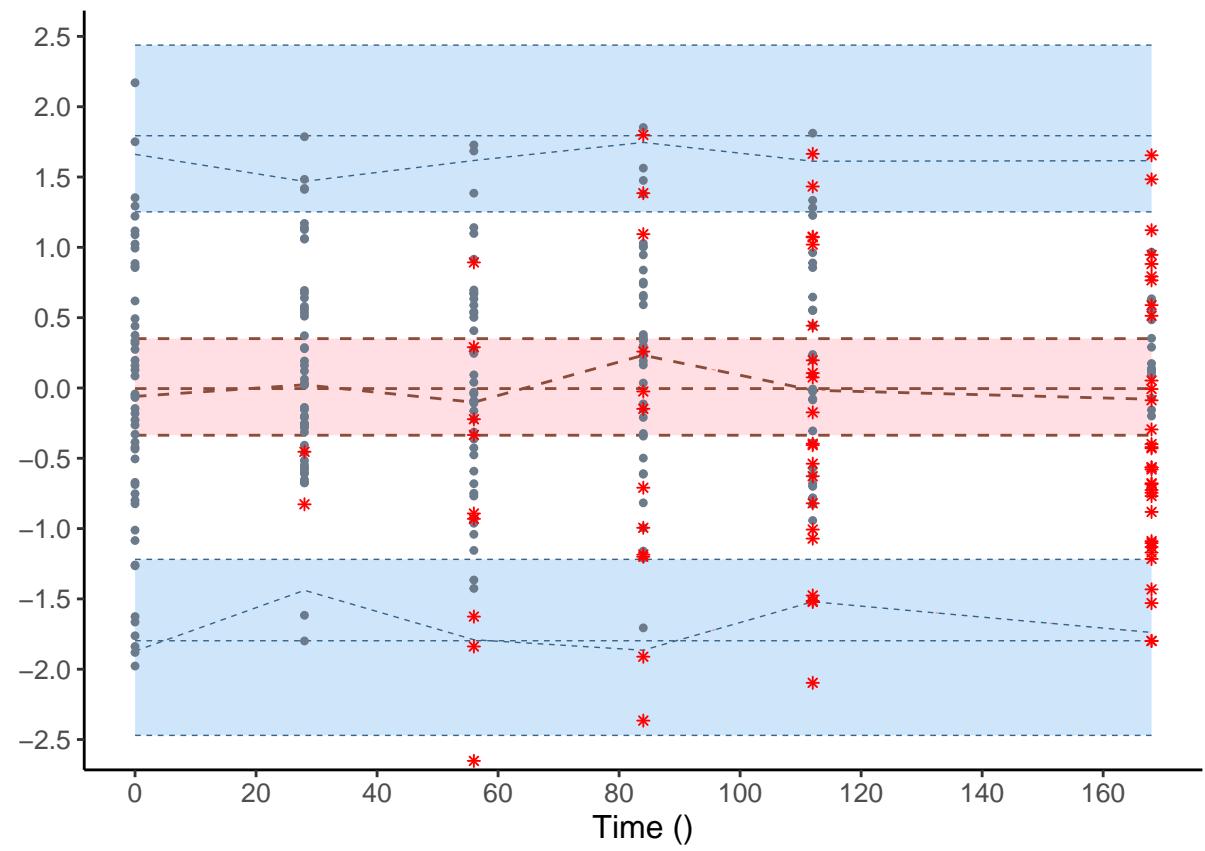


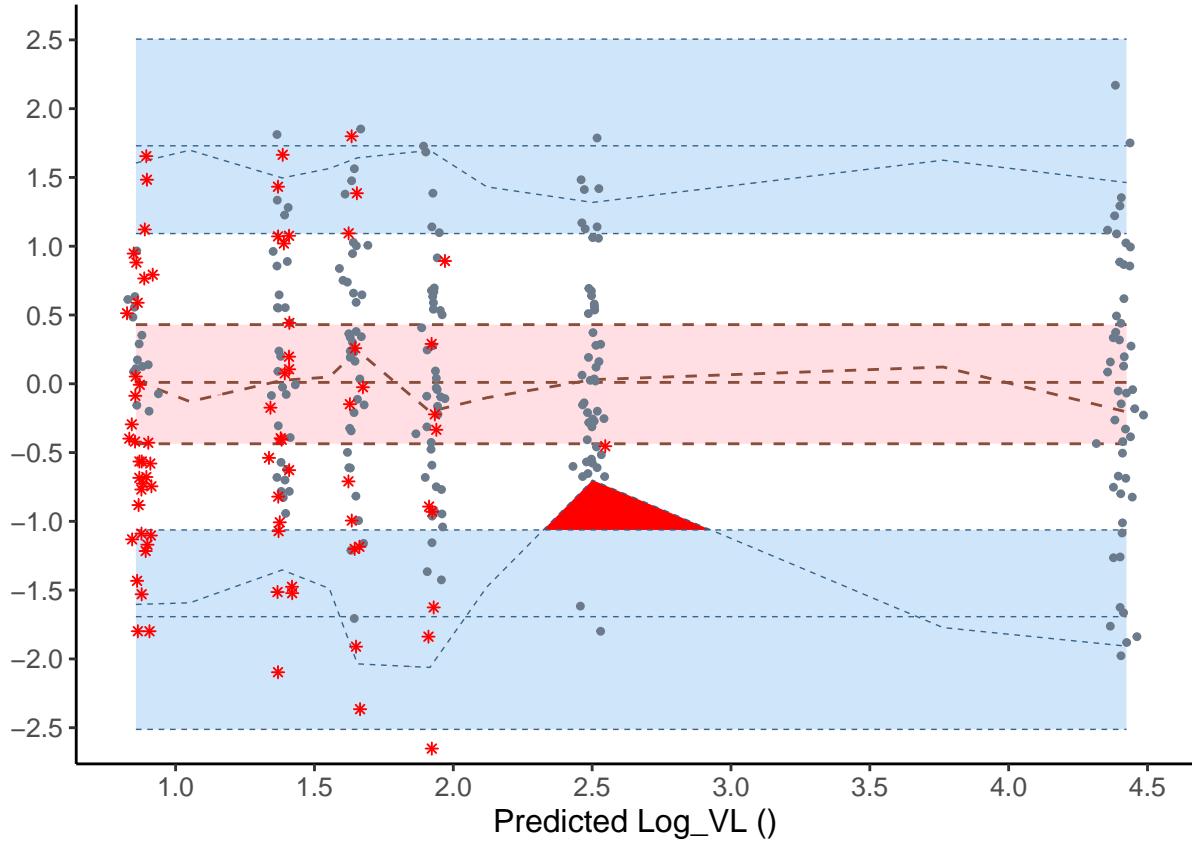


## Plots with default options

### Scatterplots

- **Corrigé (marche)**
  - quand on change size.pcens le symbole des données manquantes change aussi => OK
  - changé la taille par défaut de pcens à 0.5 sinon on a l'impression qu'il y a plus de données censurées qu'il ne devrait
- **Problems (à résoudre Romain)**
  - titles for scatterplots
    - \* (redite): enlever les unités si absentes du jeu de données (ie “Predicted Log\_VL” et pas “Predicted Log\_VL ()”)
  - **size** ne passe pas (devrait changer à la fois size.pobs et size.pcens)
    - \* même pb que dans le graphe par défaut (ie les 4 graphes)

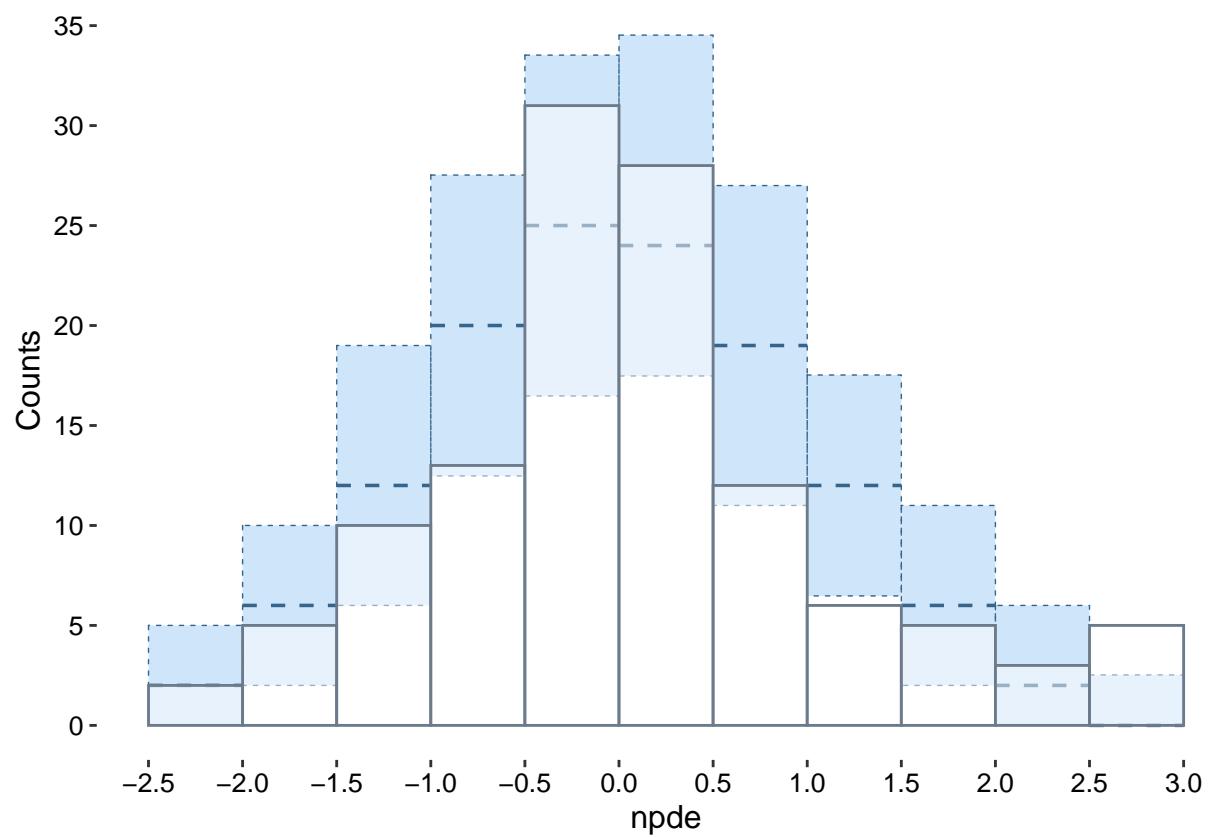


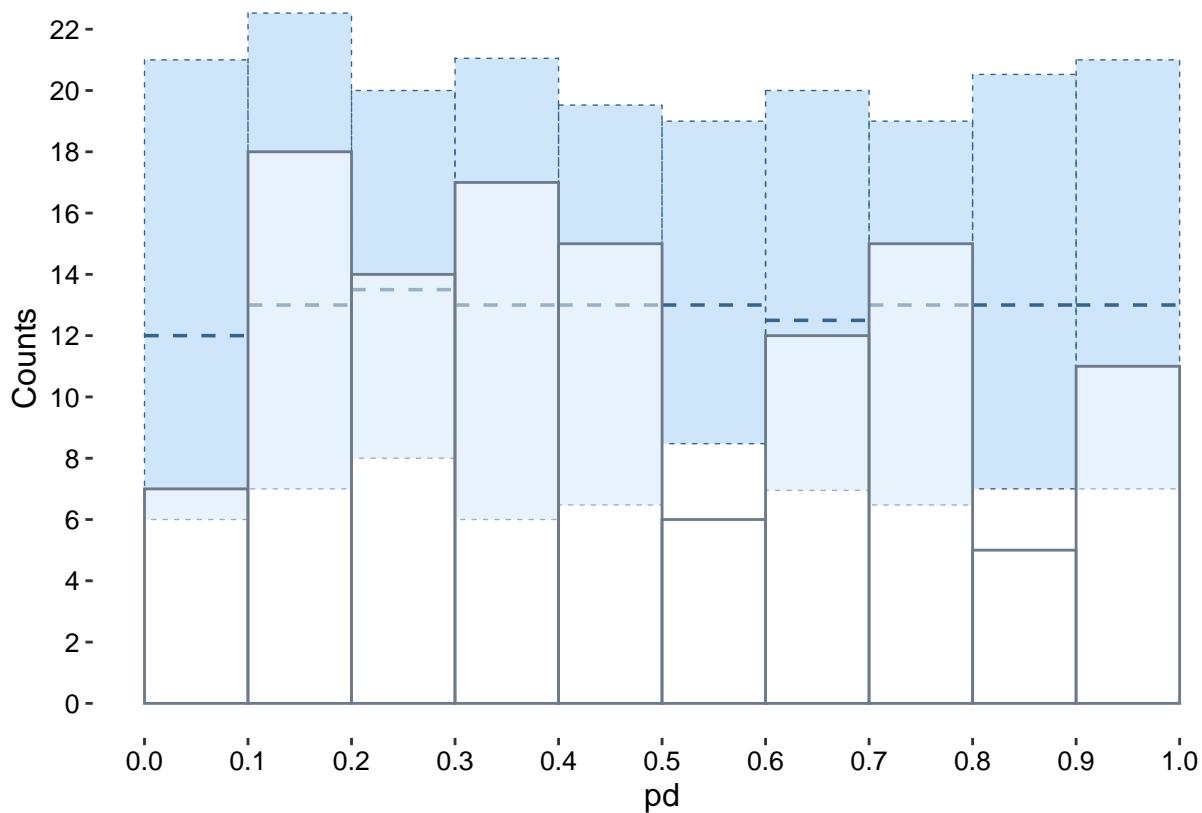


```
## Option which.y= x.scatter not recognised
## Option which.y= pred.scatter not recognised
```

## Distribution plots

- Corrections
  - titles adjust to the metric
- Problems (**à résoudre Romain**)
  - message bizarre sur qqplot:
    - \* `plot(theofit1, plot.type="qqplot",which="npde")` doTryCatch(return(expr), name, parentenv, handler): liste de retraçage des affichages incomplète
    - \* quand je trace plusieurs graphes de suite, celui de qqplot a une tête bizarre (gros points rouges et pas de fill) alors qu'il passe très bien tout seul (affichage d'un truc intermédiaire de ggplot ??? comme pour les patterns l'autre jour ?)
  - pb with **size** (comme plus haut, size ne passe pas, seul size.pobs passe)
  - **axe title** for ecdf plot (should be: ‘npde’ on X-axis (/pd/npd) and ‘Empirical distribution function’ on Y-axis)

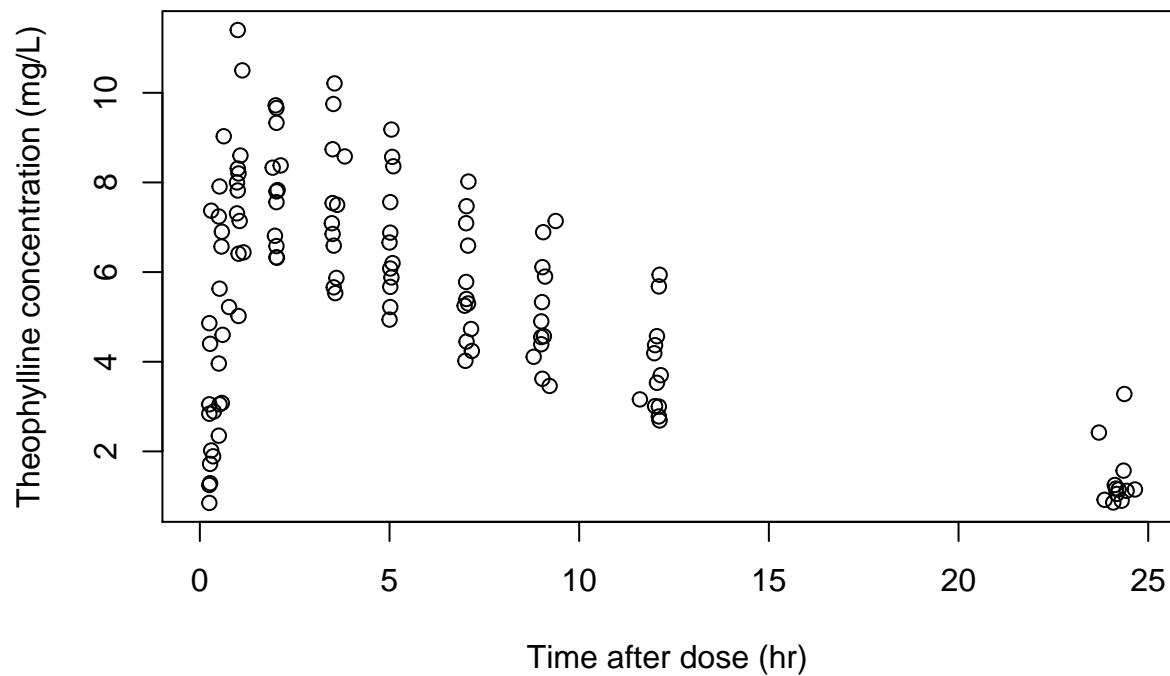




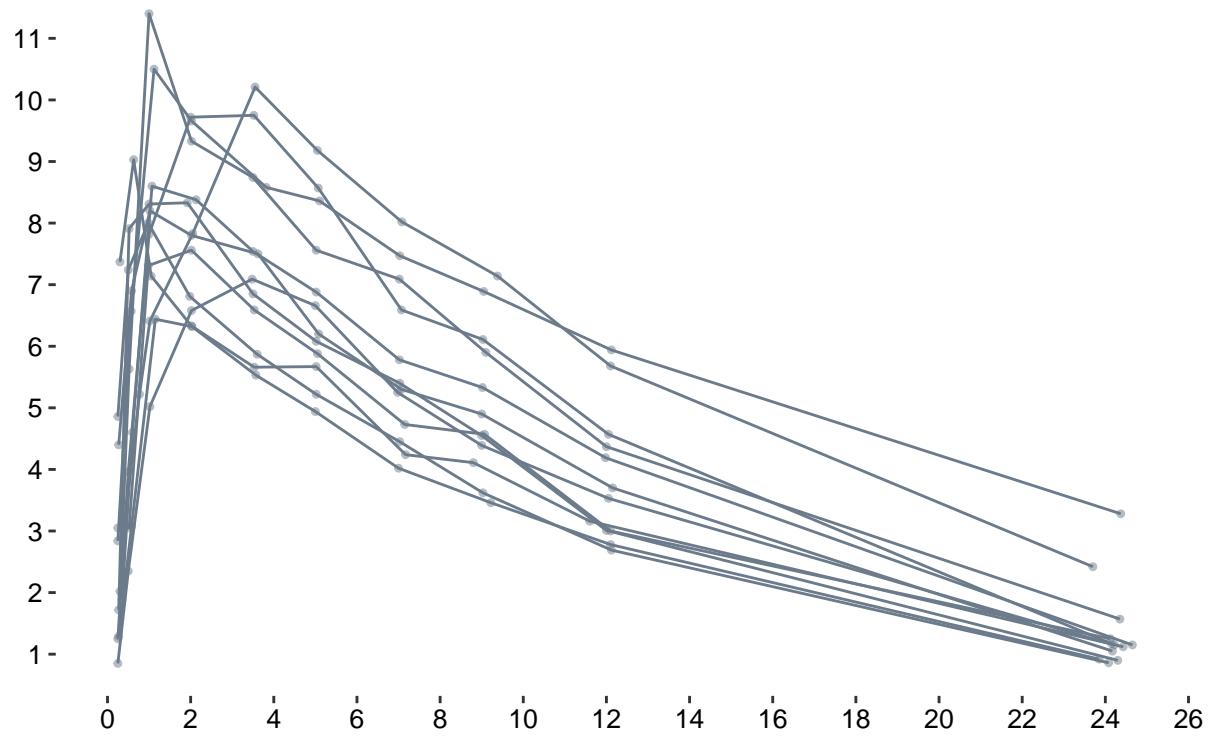
## Data plots

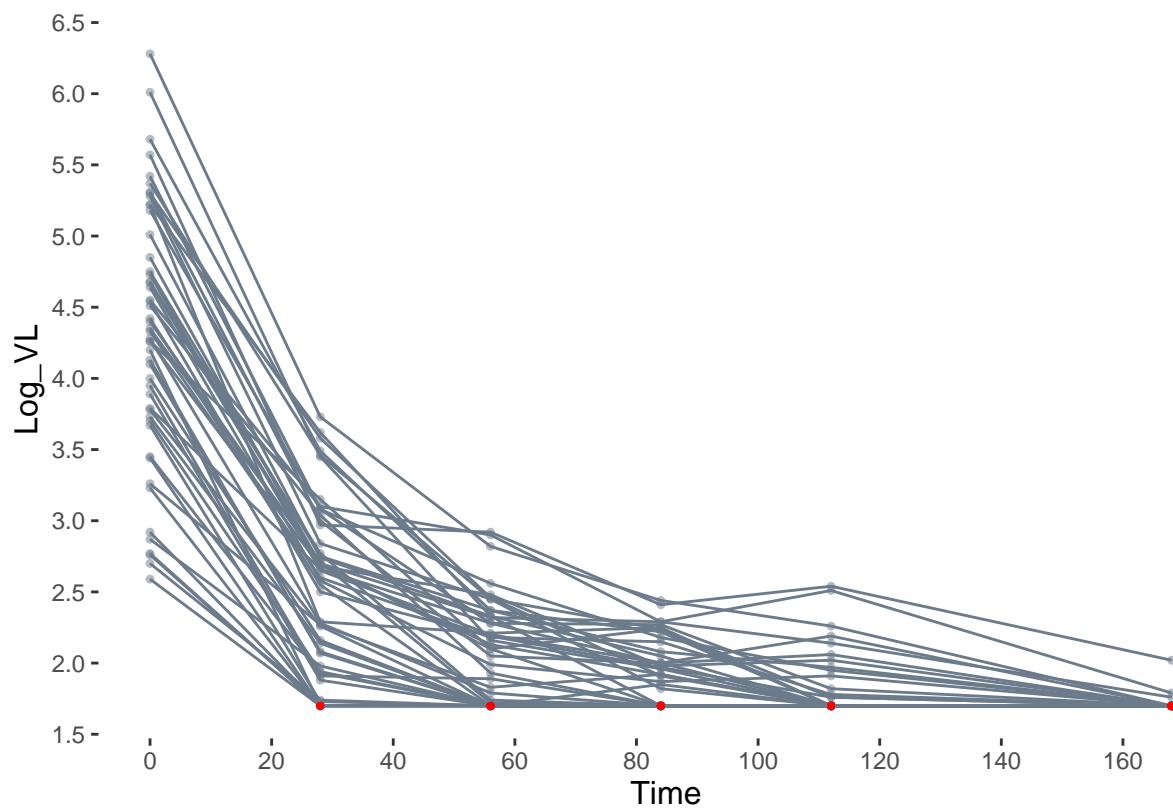
- **Corrections**
  - `plot.loq`: works to remove the LOQ values
- **Problems (à résoudre Romain)**
  - plot d'un **objet npdeData** ne marche pas
    - \* par contre `plot.type="data"` de l'objet `npdeObject` fonctionne => bizarre, vérifier [normalement `plot.type="data"` devrait appeler `plot(objet@data)` mais là ça semble court-circuiter ceci => à rectifier, on veut plutôt l'inverse]
  - with **LOQ data**, the data being plotted is the LOQ
    - \* if applied directly to an objet `npdeData` (eg `plot(yivr50@data)`) => plot LOQ (ie censoring value)
    - \* if applied to the object 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...]
  - **lines for axes** don't appear
  - **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
  - **axis titles**

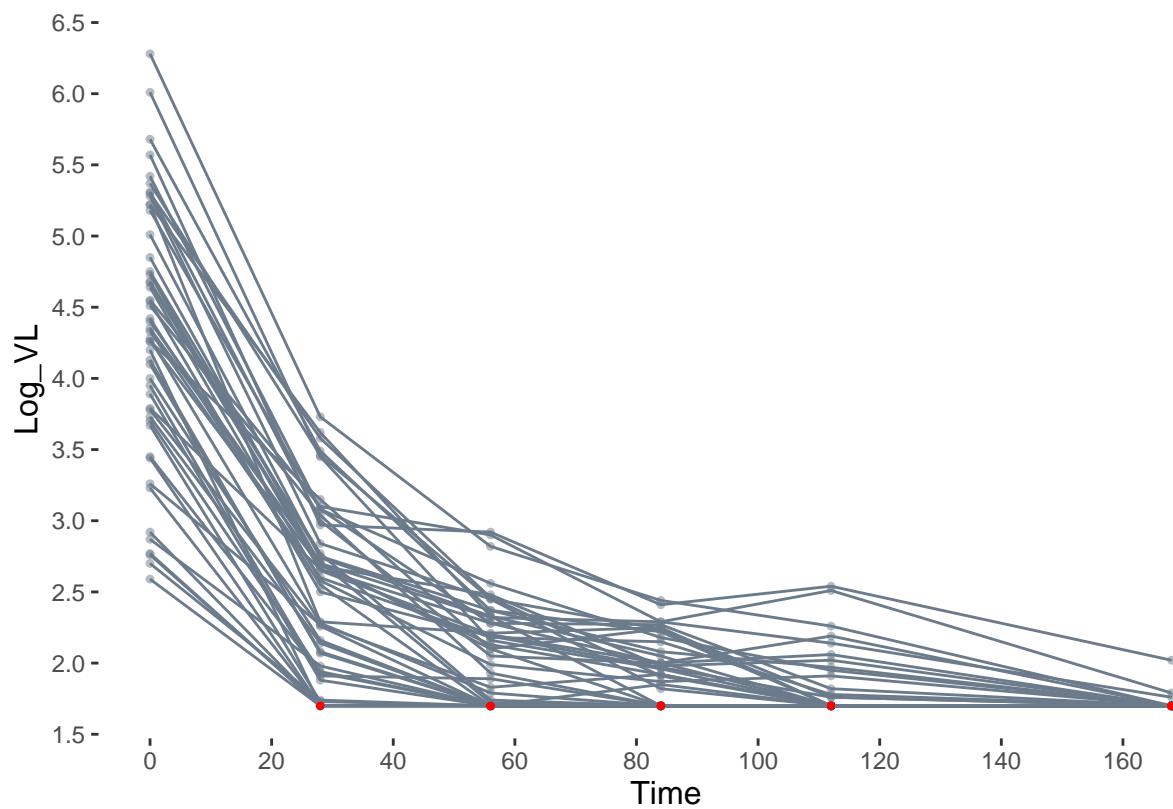
- \* missing for theofit, why ? (present for yvir50, so odd)
- **line.loq** doesn't work (no line appears at the LOQ)

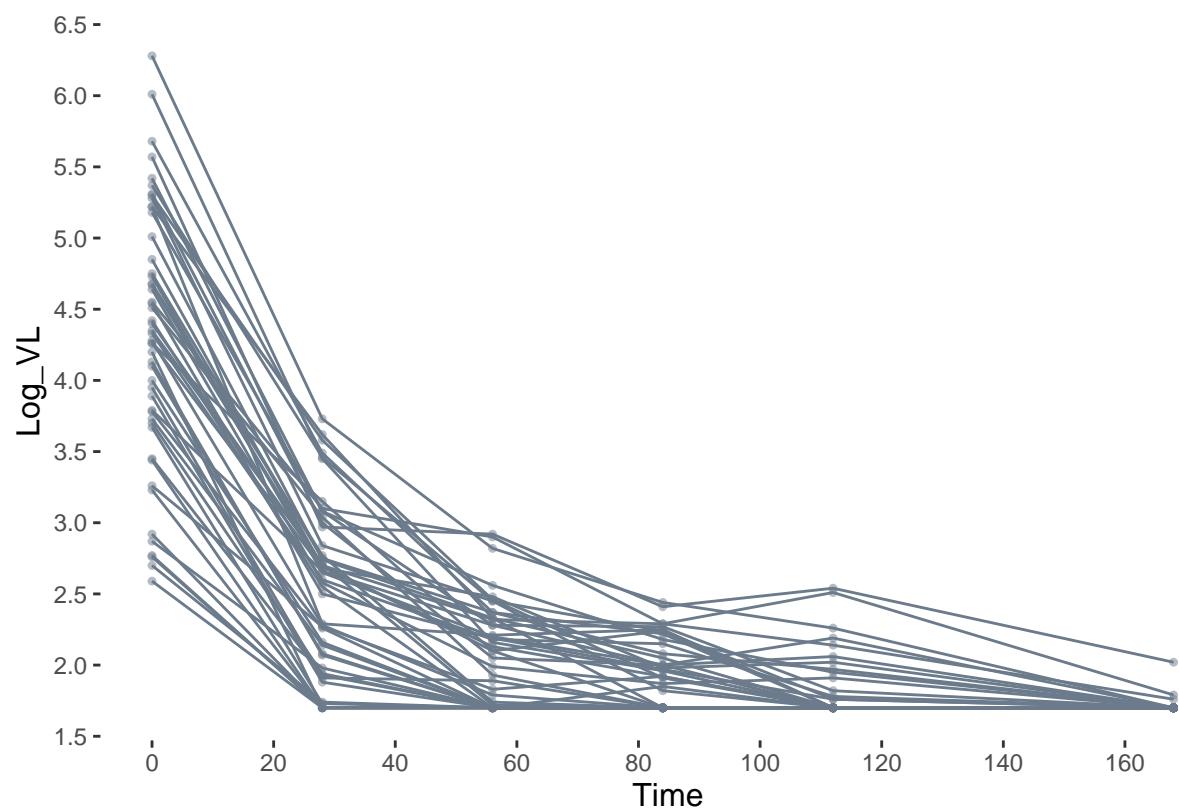


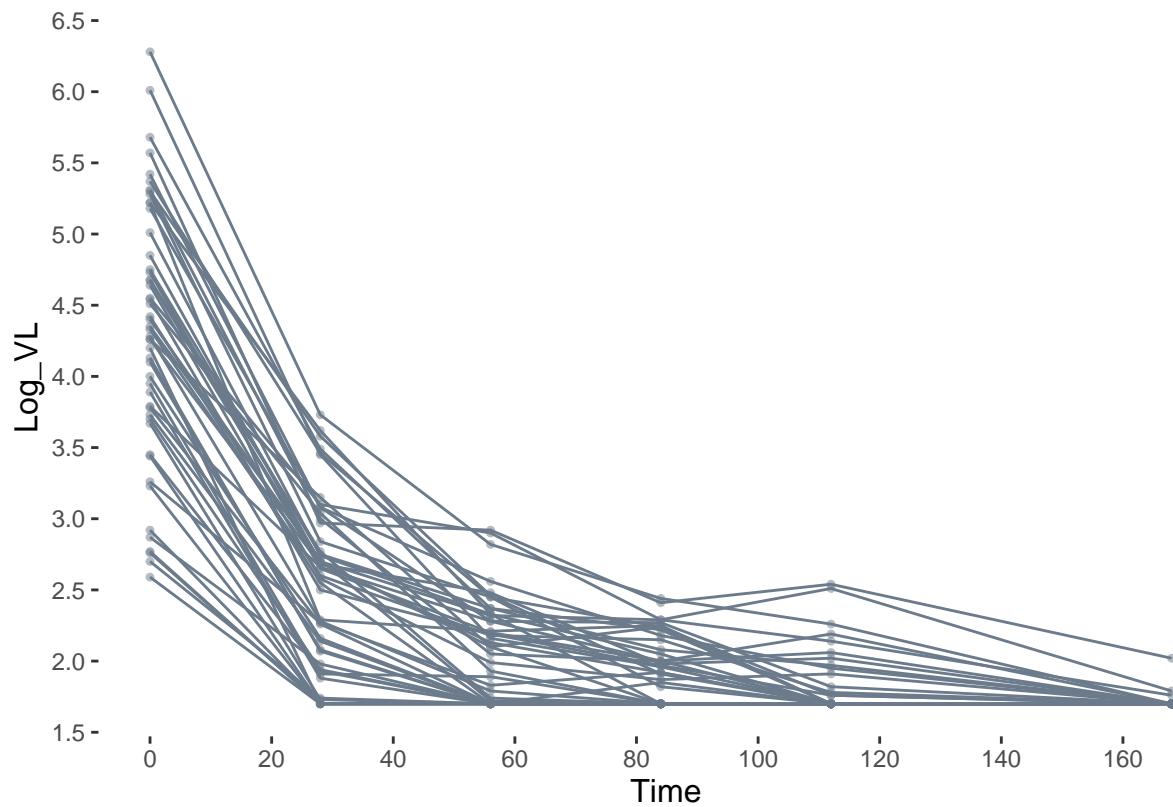
```
## Warning: Removed 12 rows containing missing values (geom_point).
## Warning: Removed 12 row(s) containing missing values (geom_path).
```





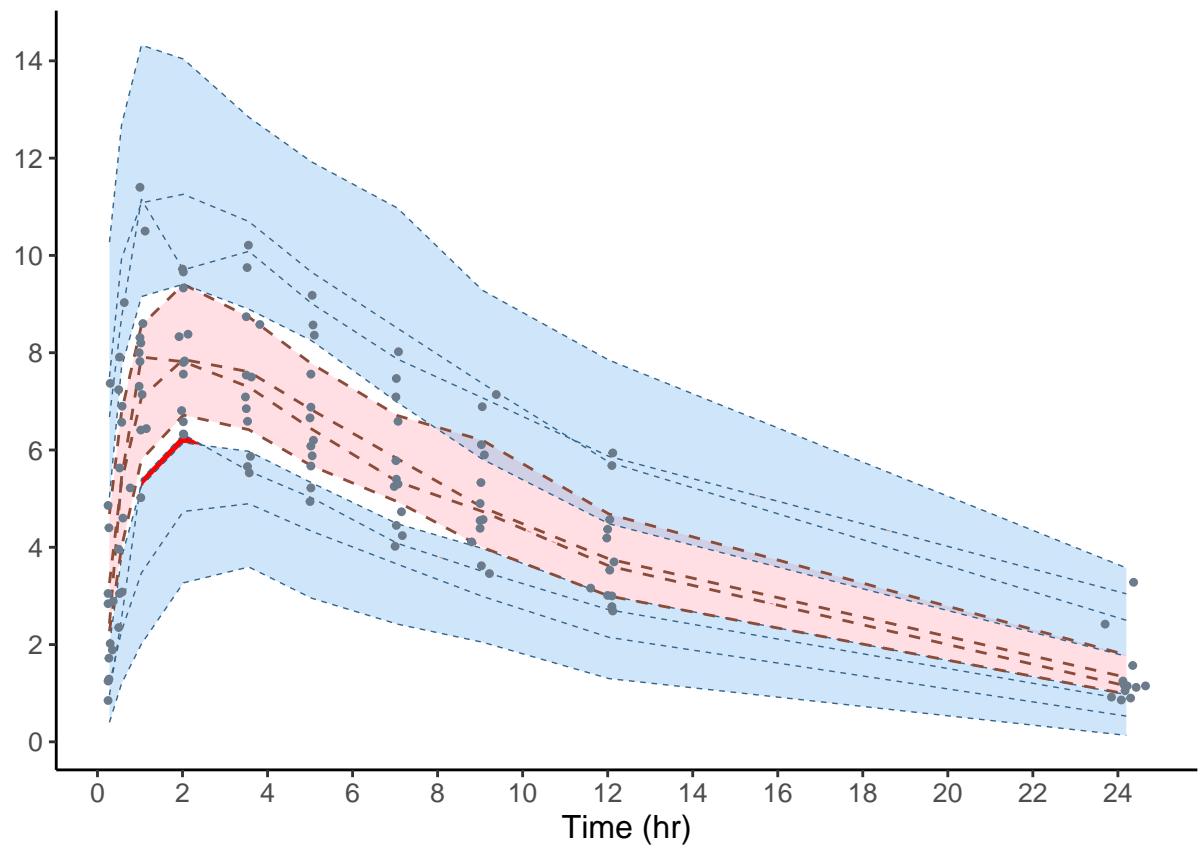


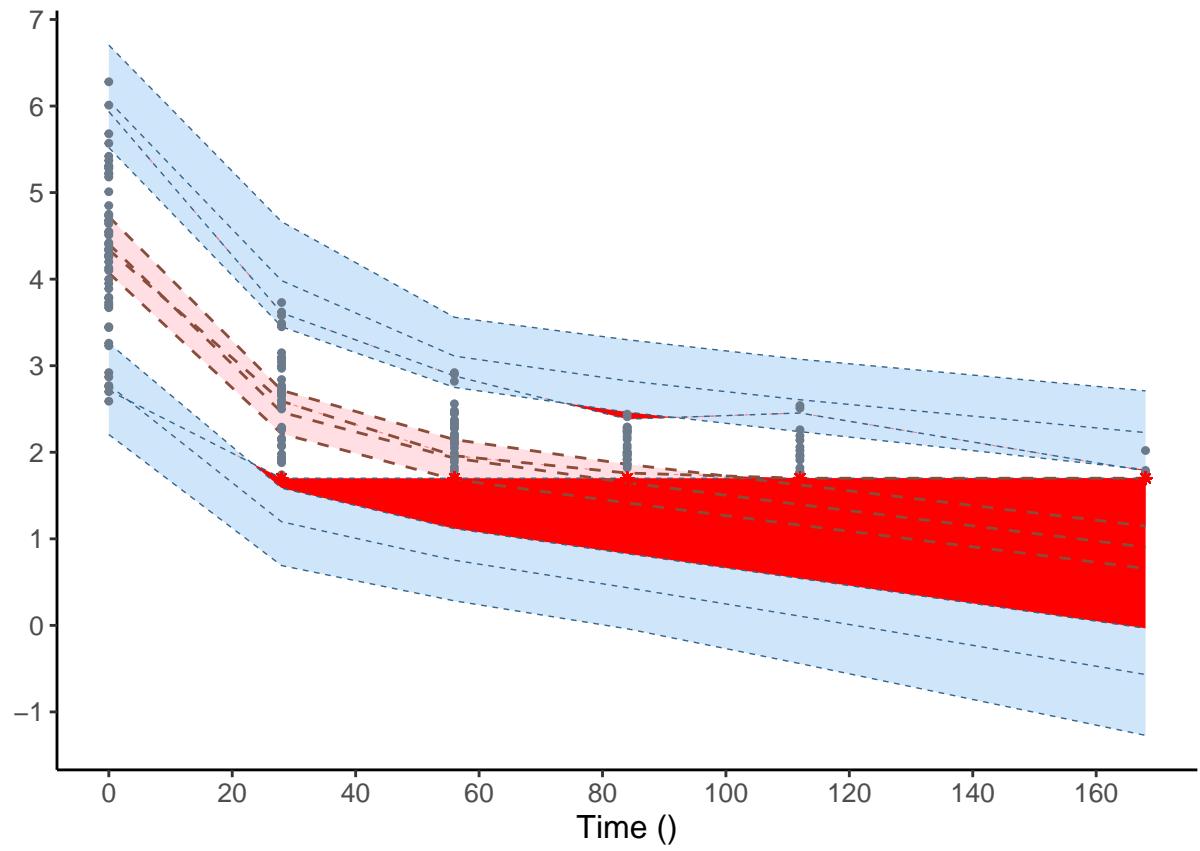


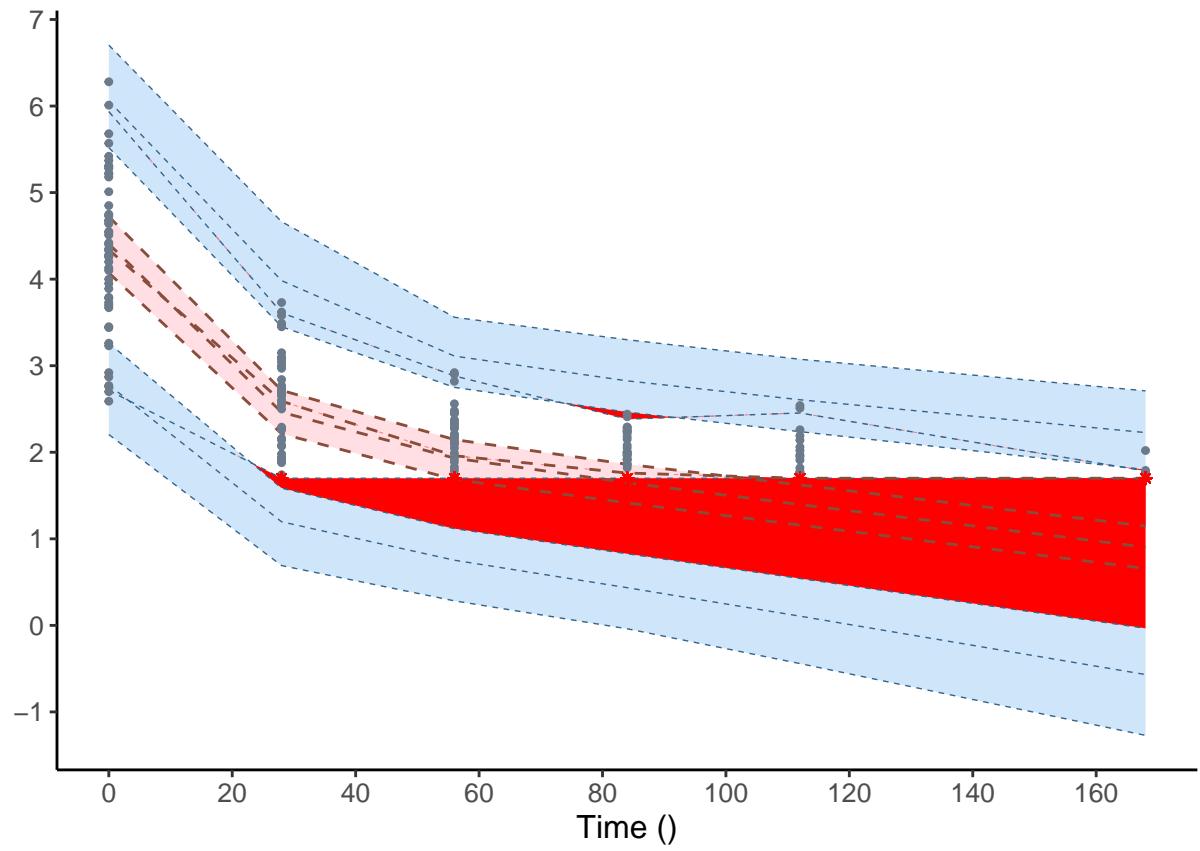


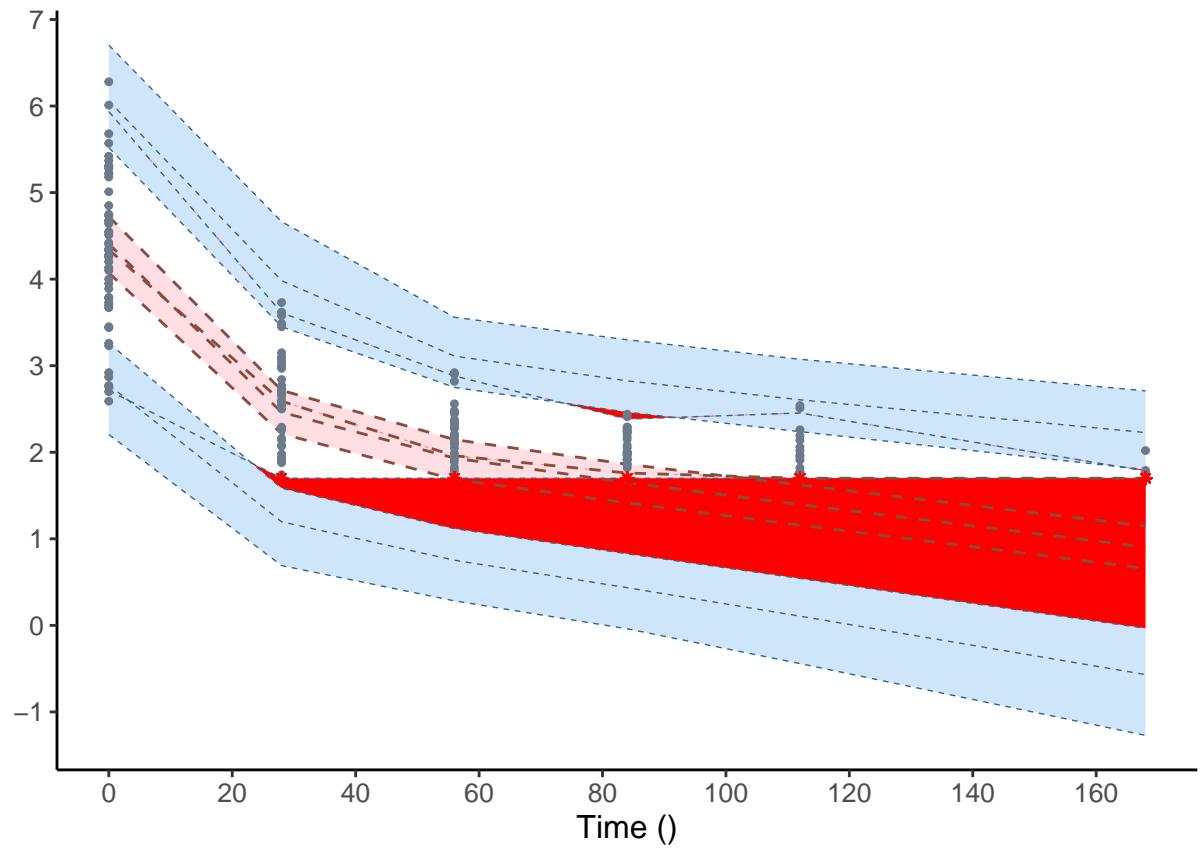
## VPC

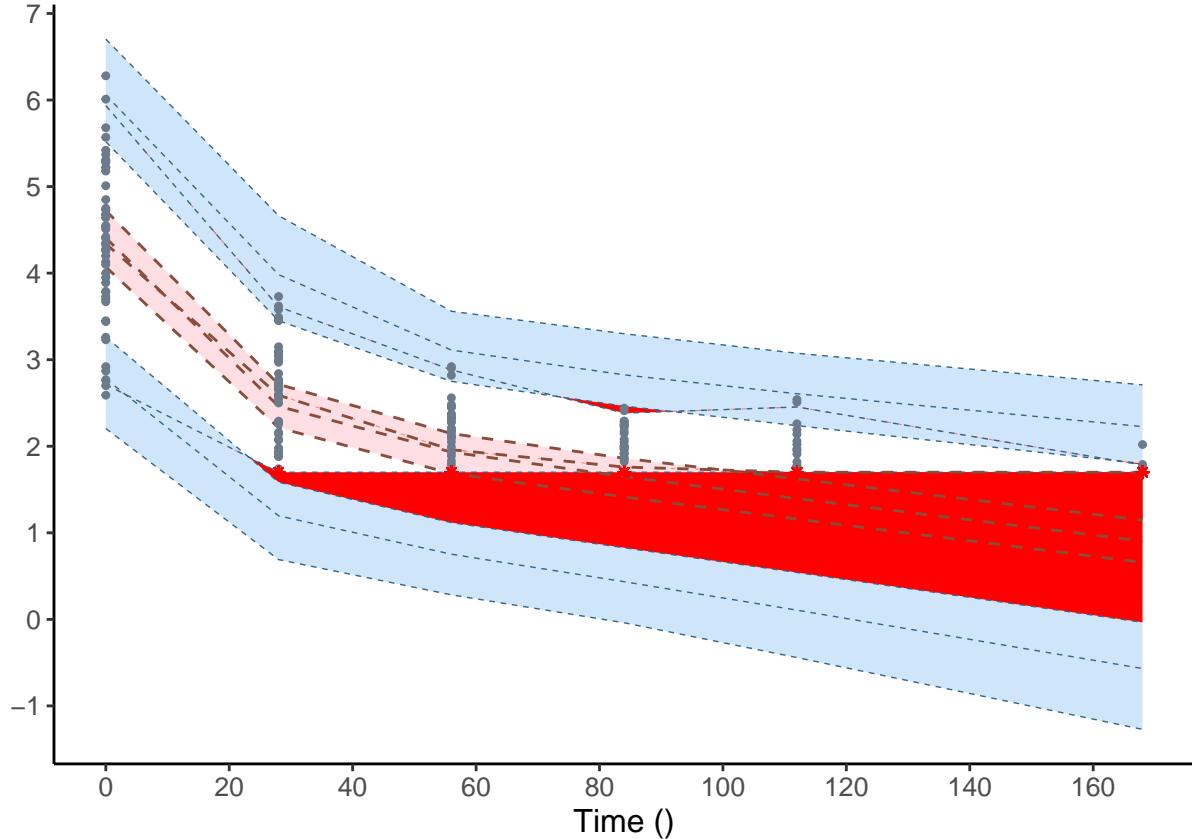
- **Corrections**
  - PI seem ok for cdf, ipred, maybe check ppred ? (outlier band seems a bit large, but possible)
- **Problems (à résoudre Romain)**
  - **Y-axis title "all"** => change to the name (and unit if available), eg "Conc (mg/L)"
  - **units** see above, don't use units when they are ""
  - computation of the PI for **omit** is not 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











## P(Y<LOQ)

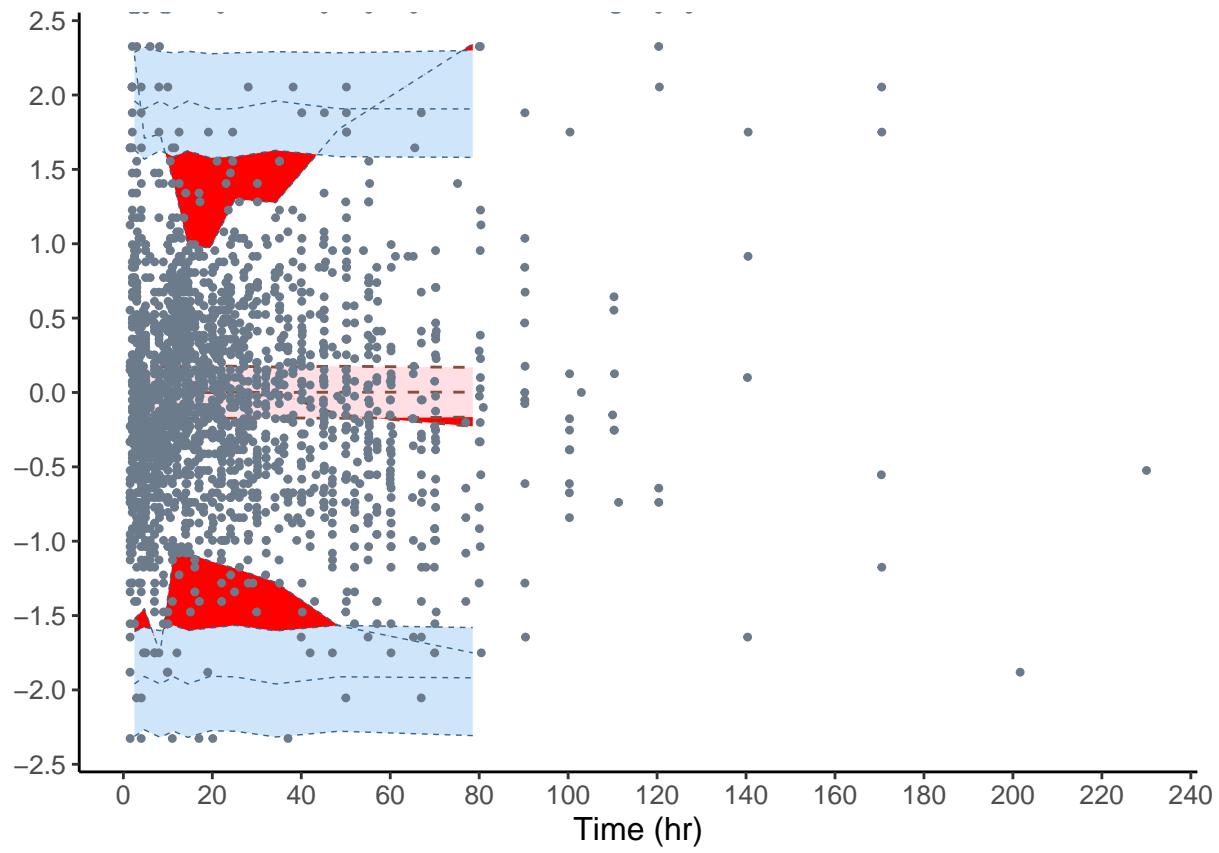
- **Problems (à résoudre Romain)**
  - PI on ne voit pas les bornes extérieures du PI (ça devrait être contrôlé par lty.bands et col.bands aussi)
  - axes missing lines for axes
  - lwd.bands should probably be increased for this plot :-/ (compared to the others), but never mind => show an example in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

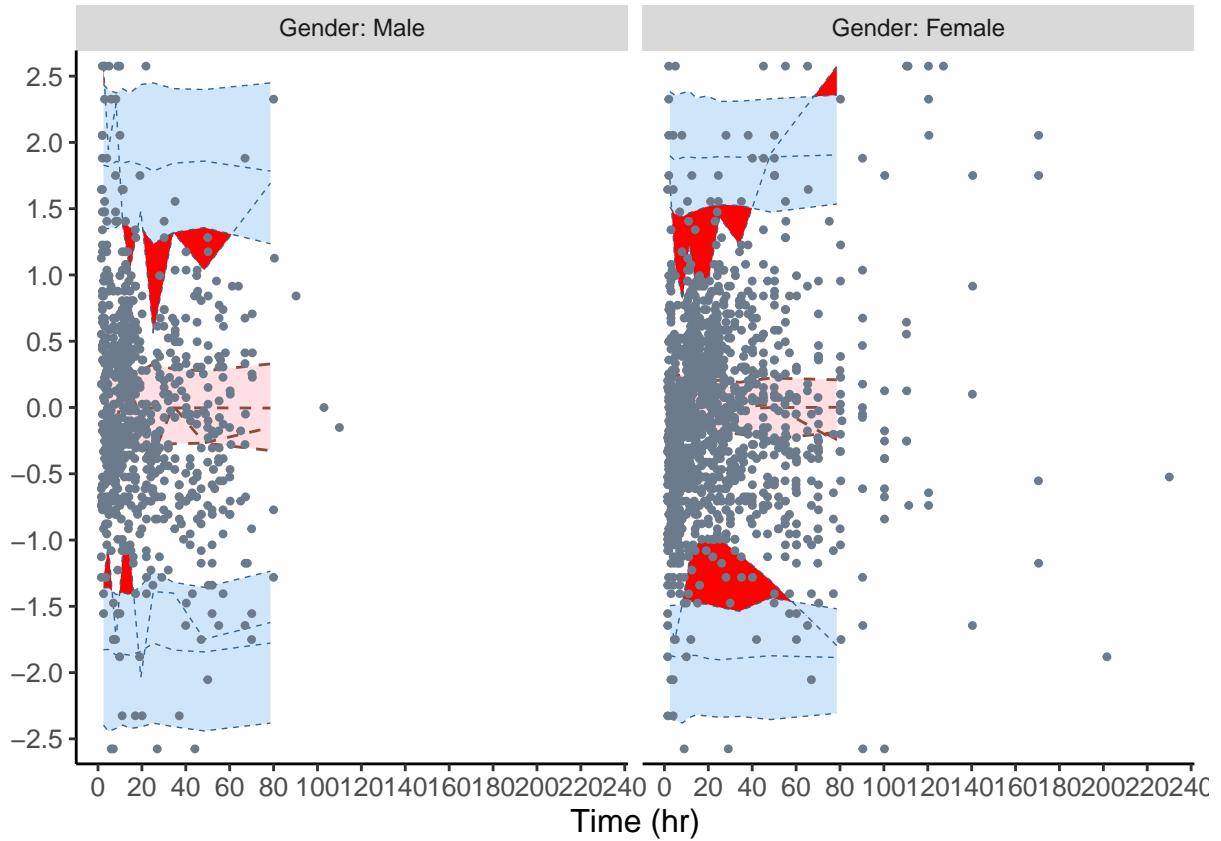
## Covariate plots

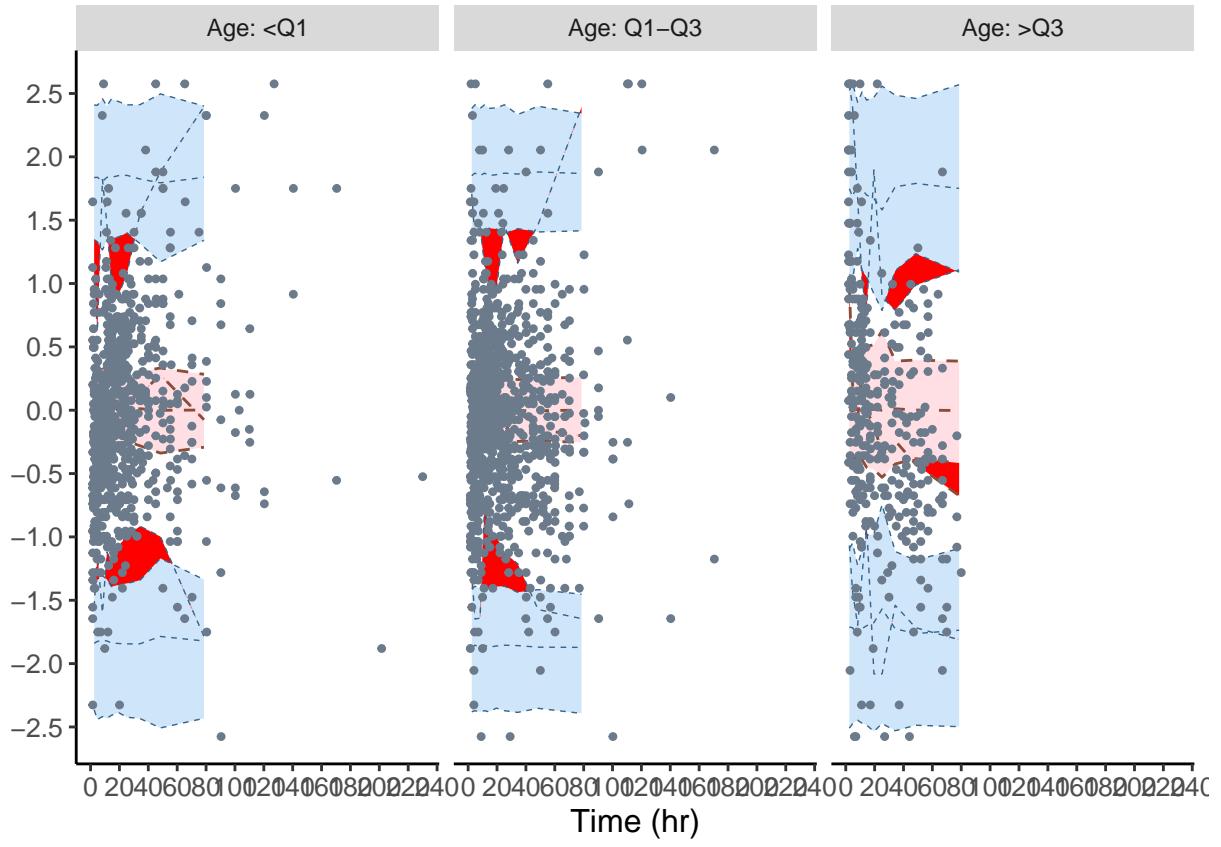
### Covariates

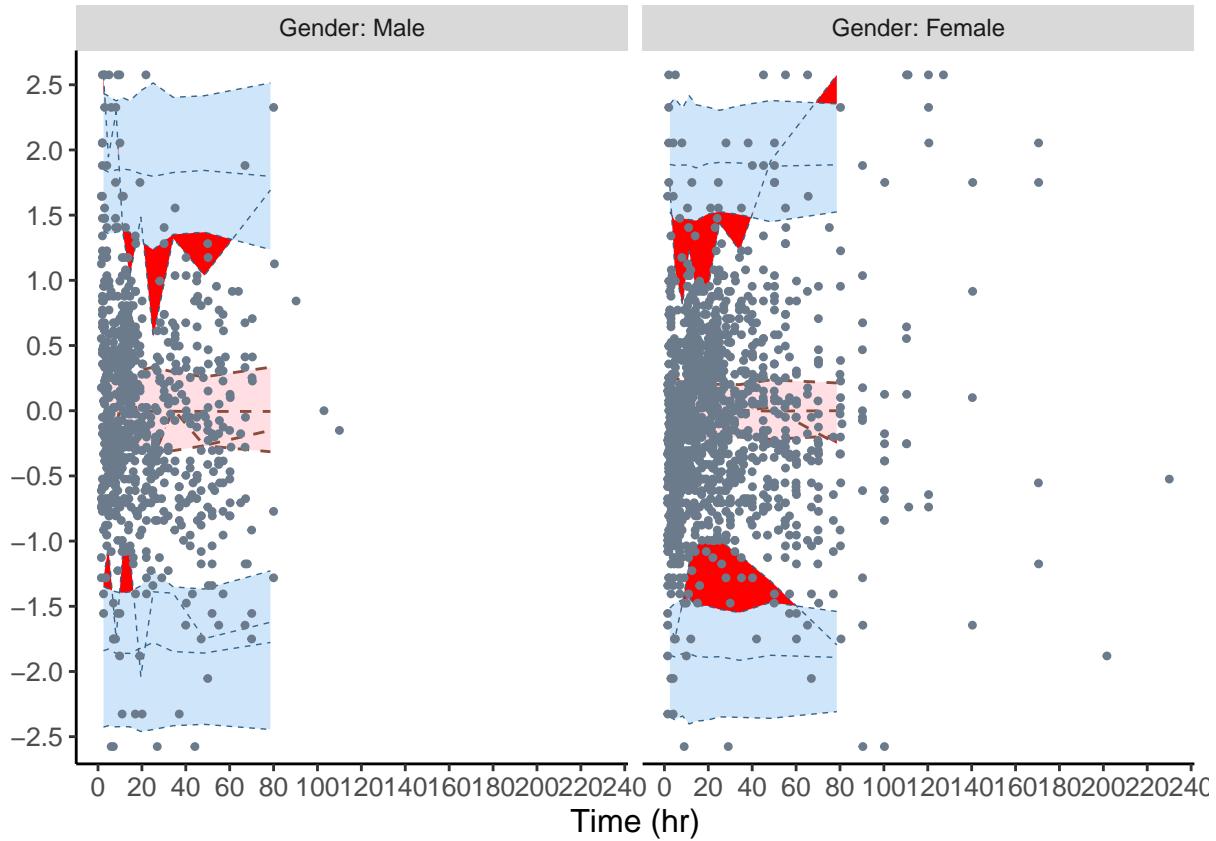
#### Splitting plots according to covariates

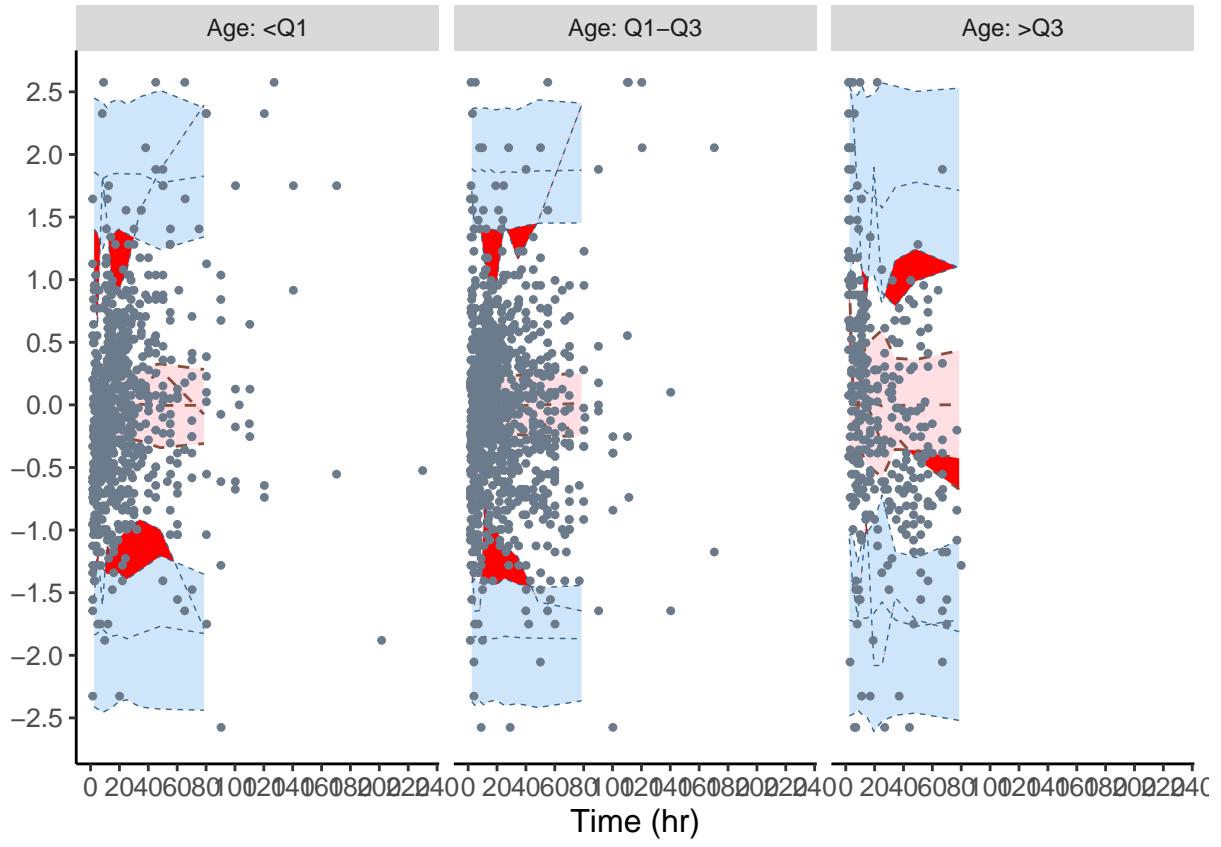
- **Problems (à résoudre Romain)**
  - yes here we really need to extend PI to cover the whole X-range for x.scatter and pred.scatter (not VPC)
    - \* just the PI (not the observed percentiles or outlier bands)
  - **covsplit**
    - \* vpc, x.scatter, pred.scatter:
      - works both with a single covariate, and with a vector of covariates
      - doesn't work with only covsplit => should use all covariates in the model
    - \* doesn't work with the default plots
      - we should get default plots for each category of the covariate
  - **xlim** doesn't work (nor does **ylim**)

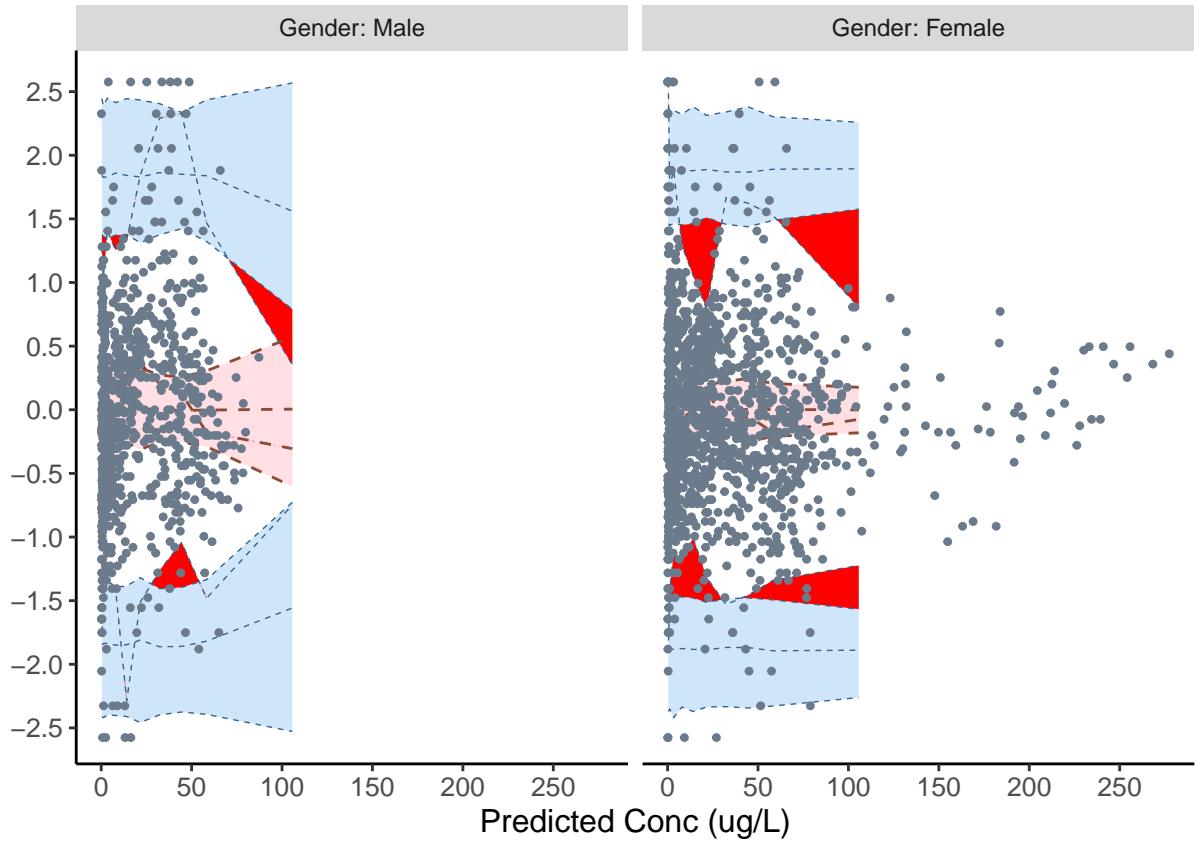


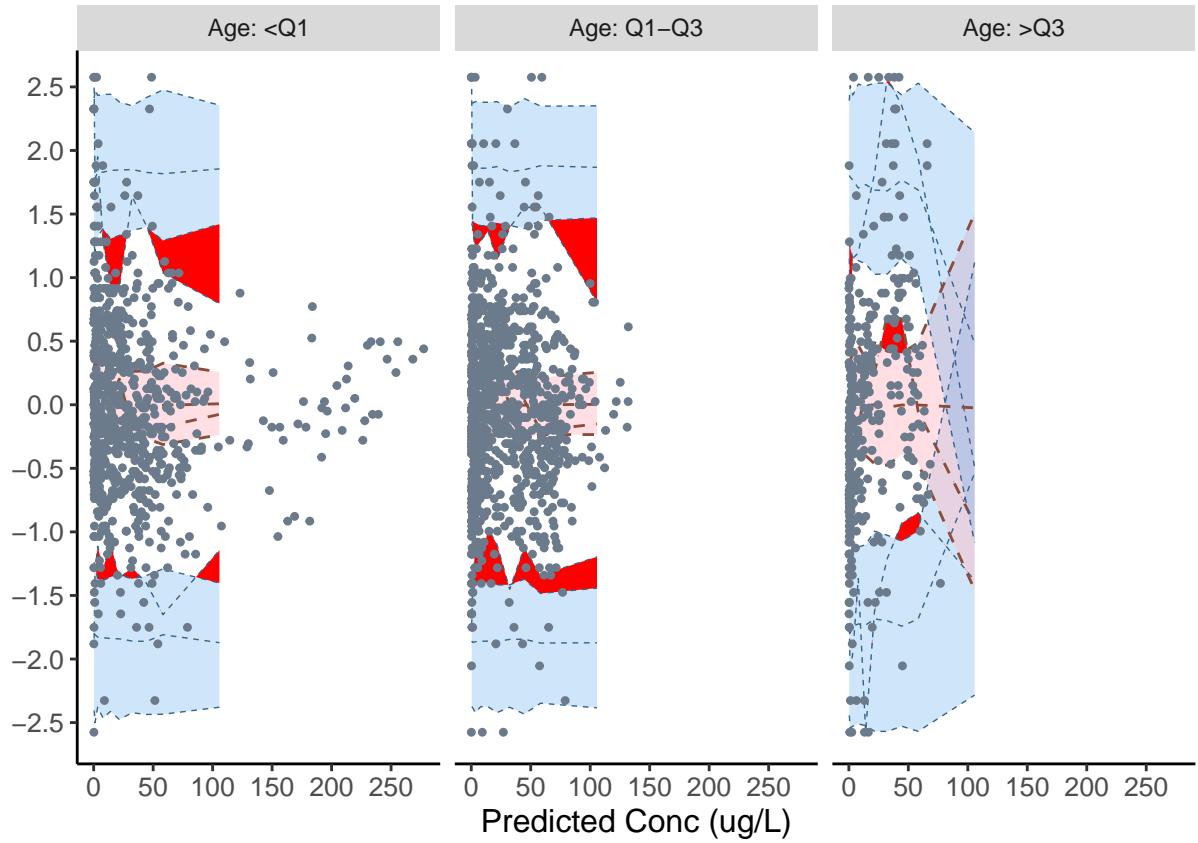


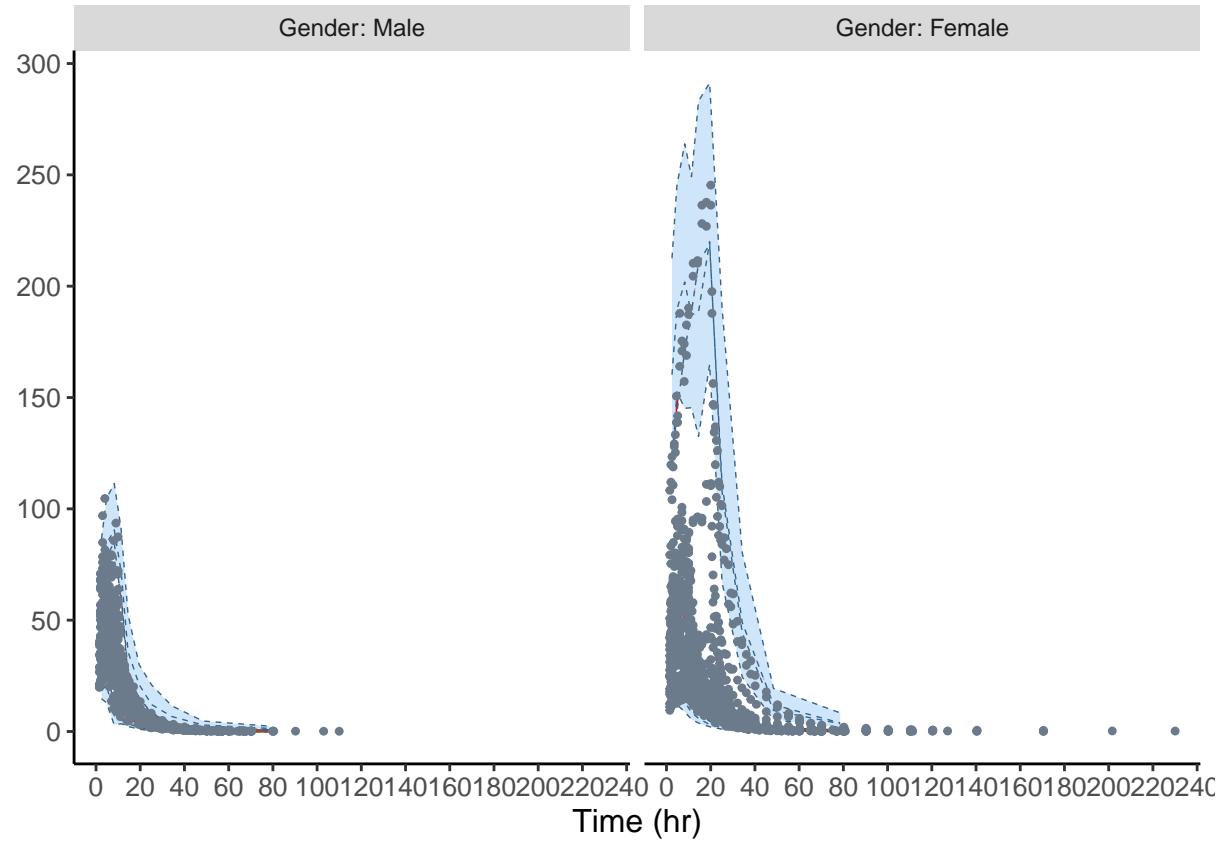


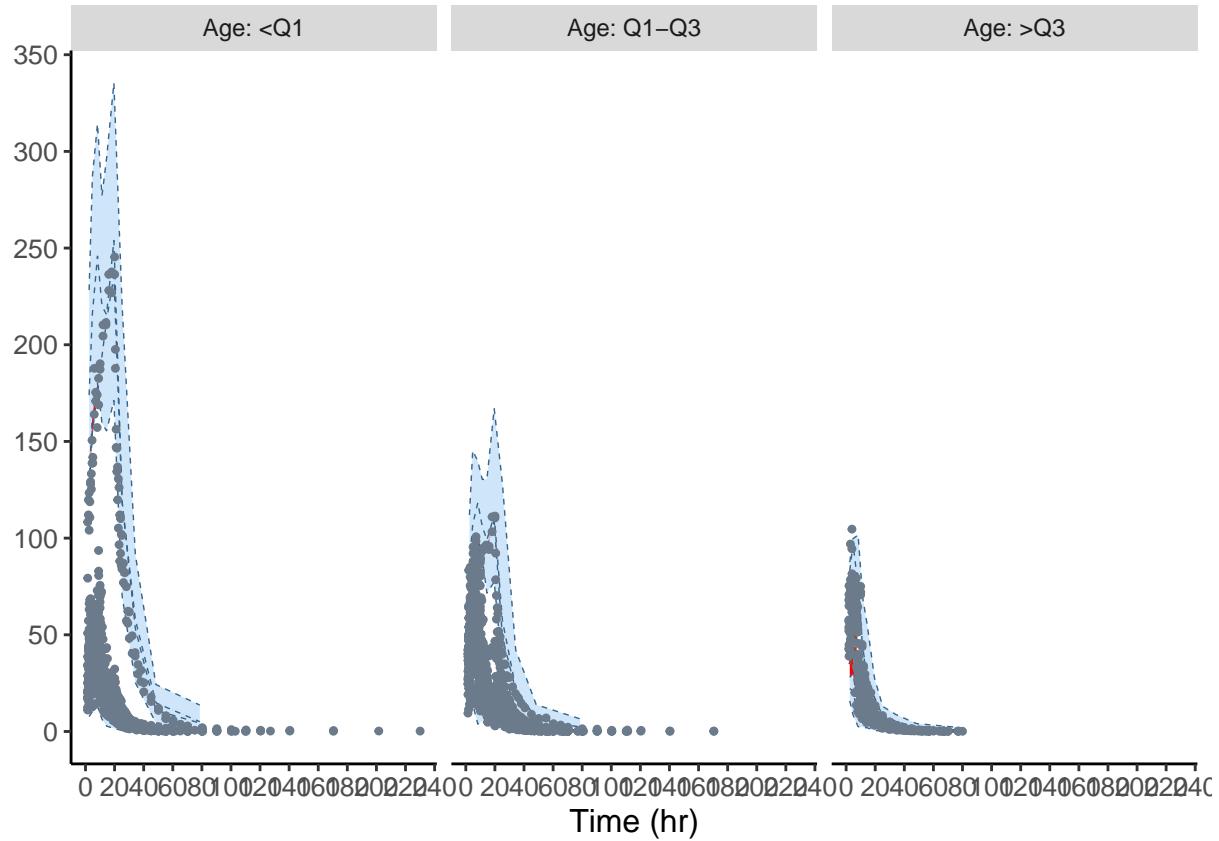


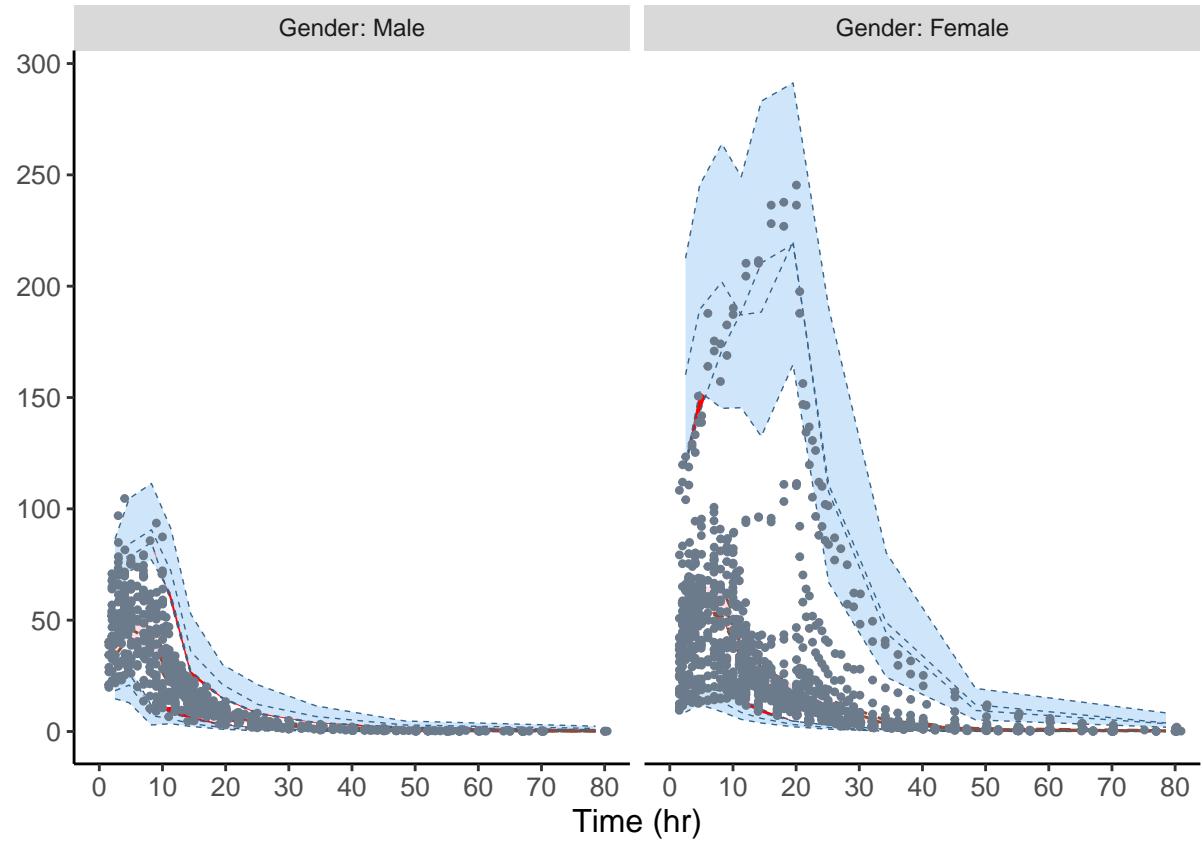


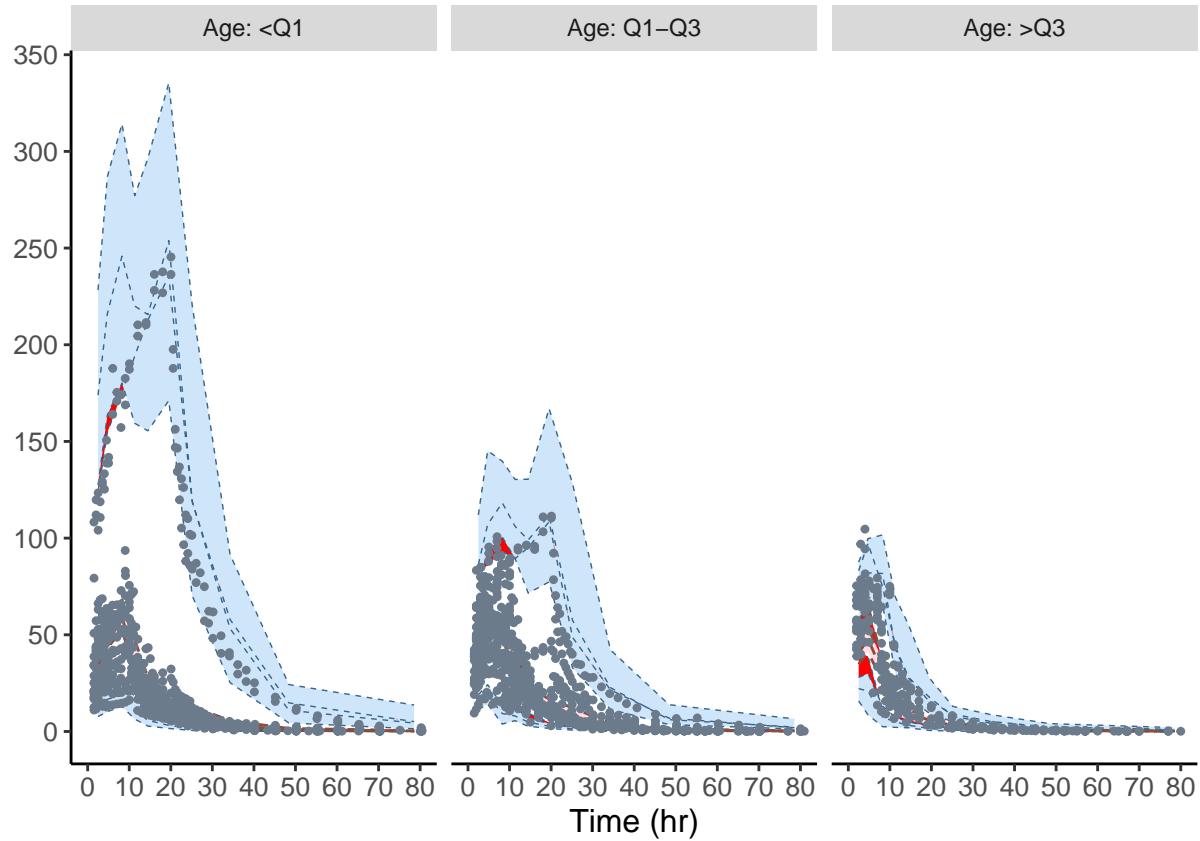


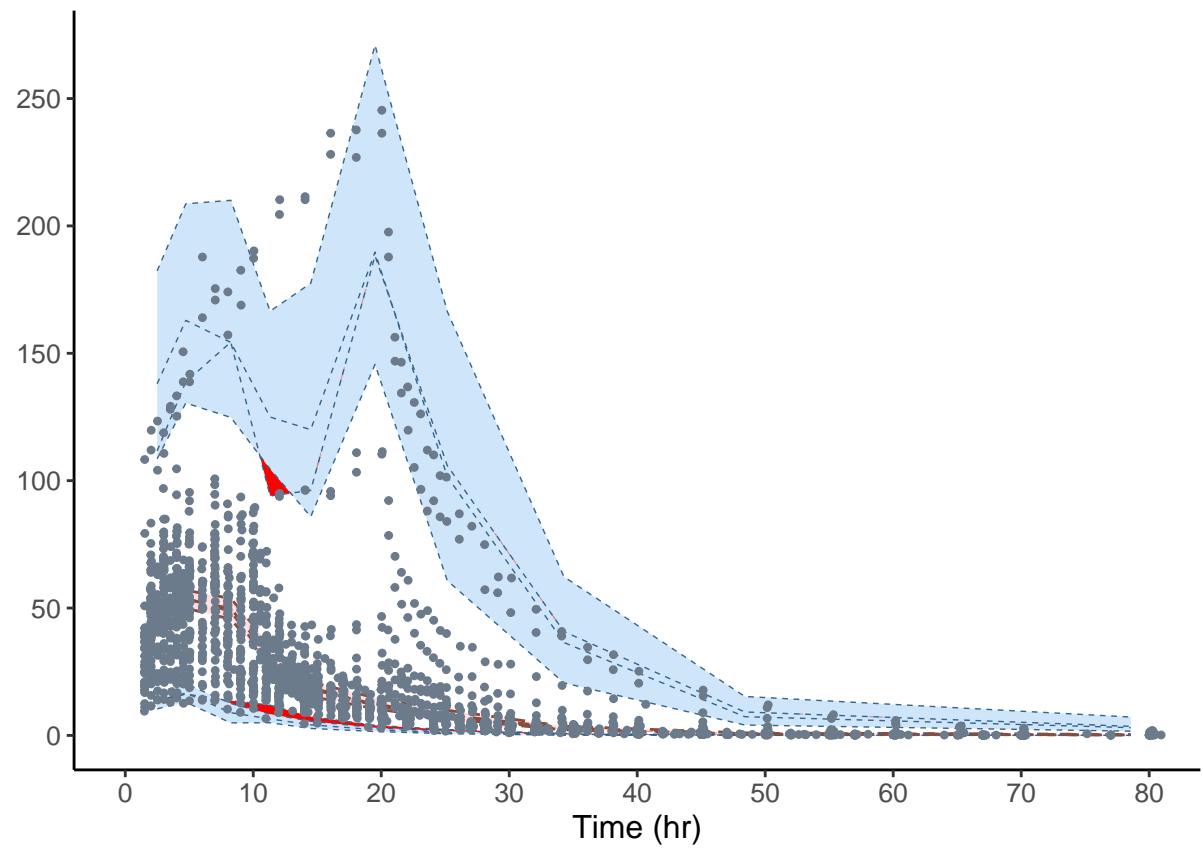


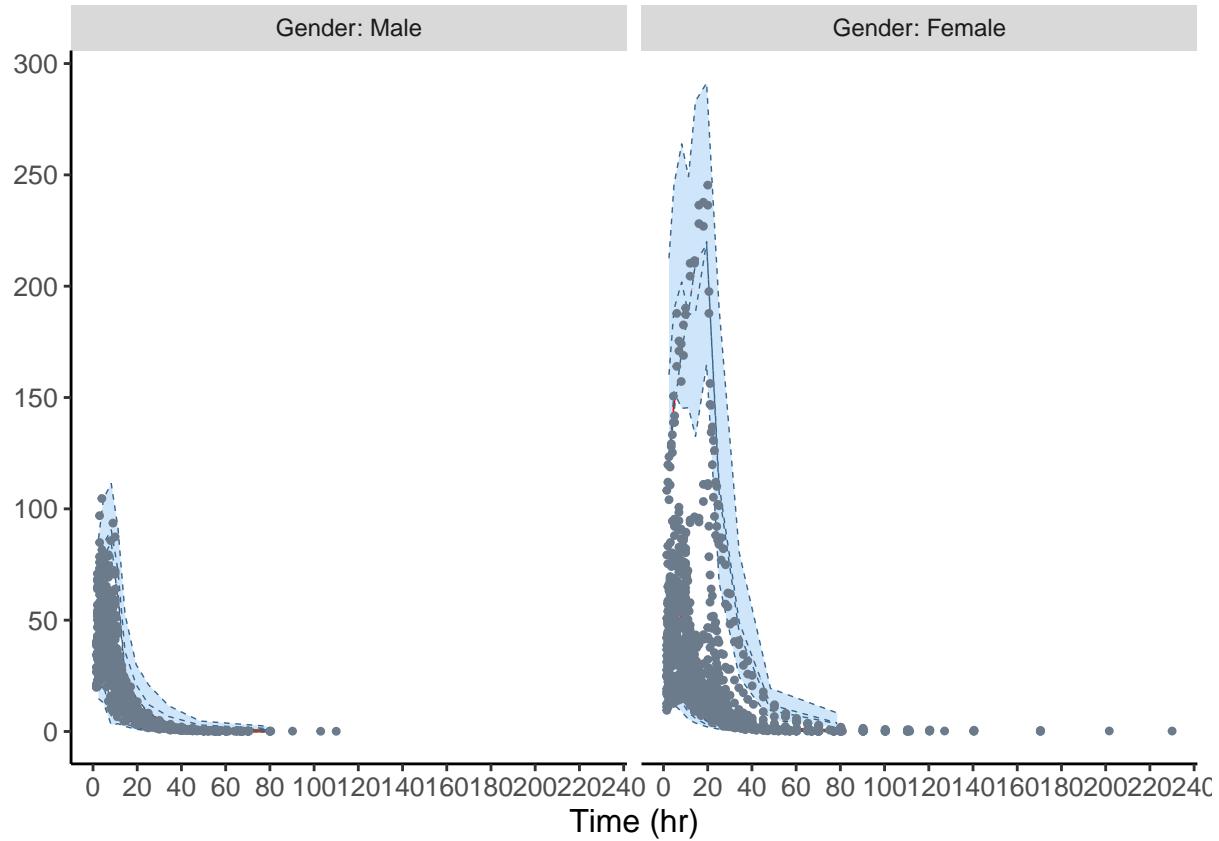


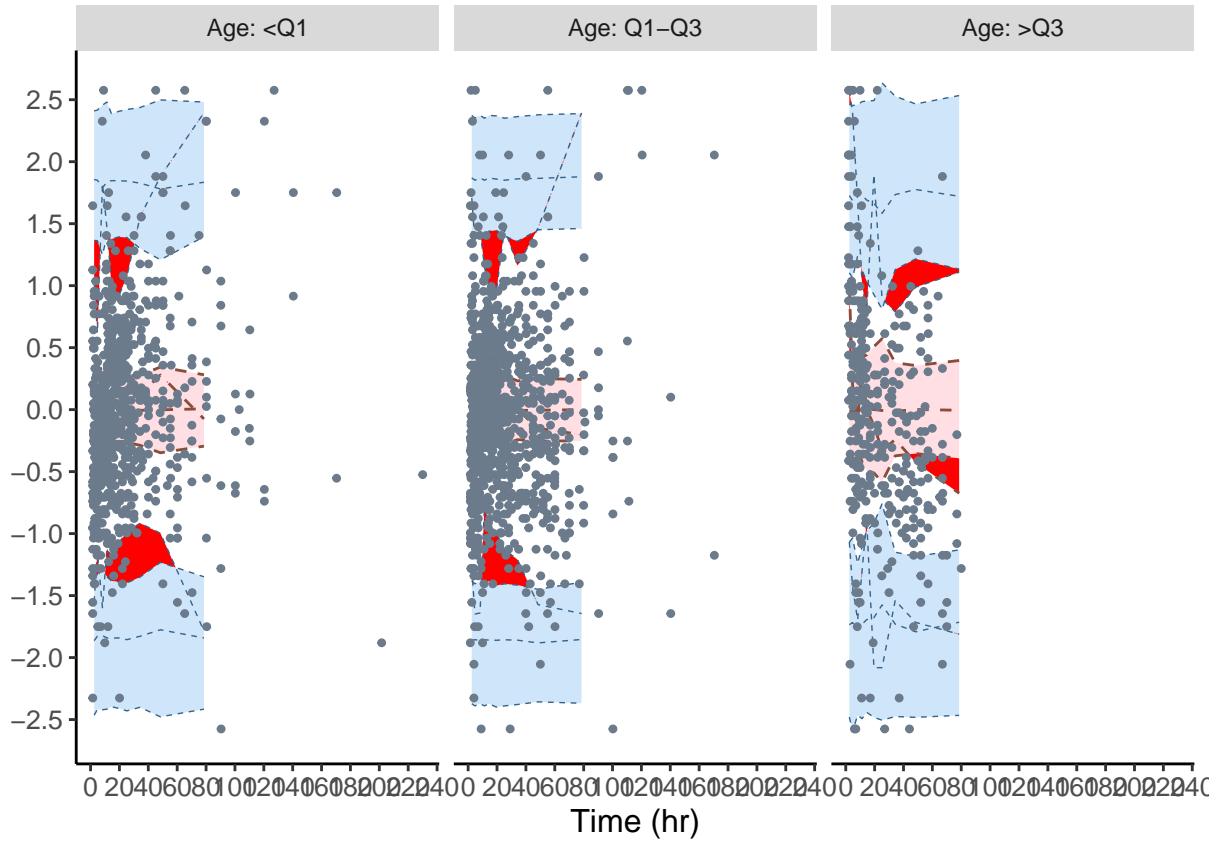


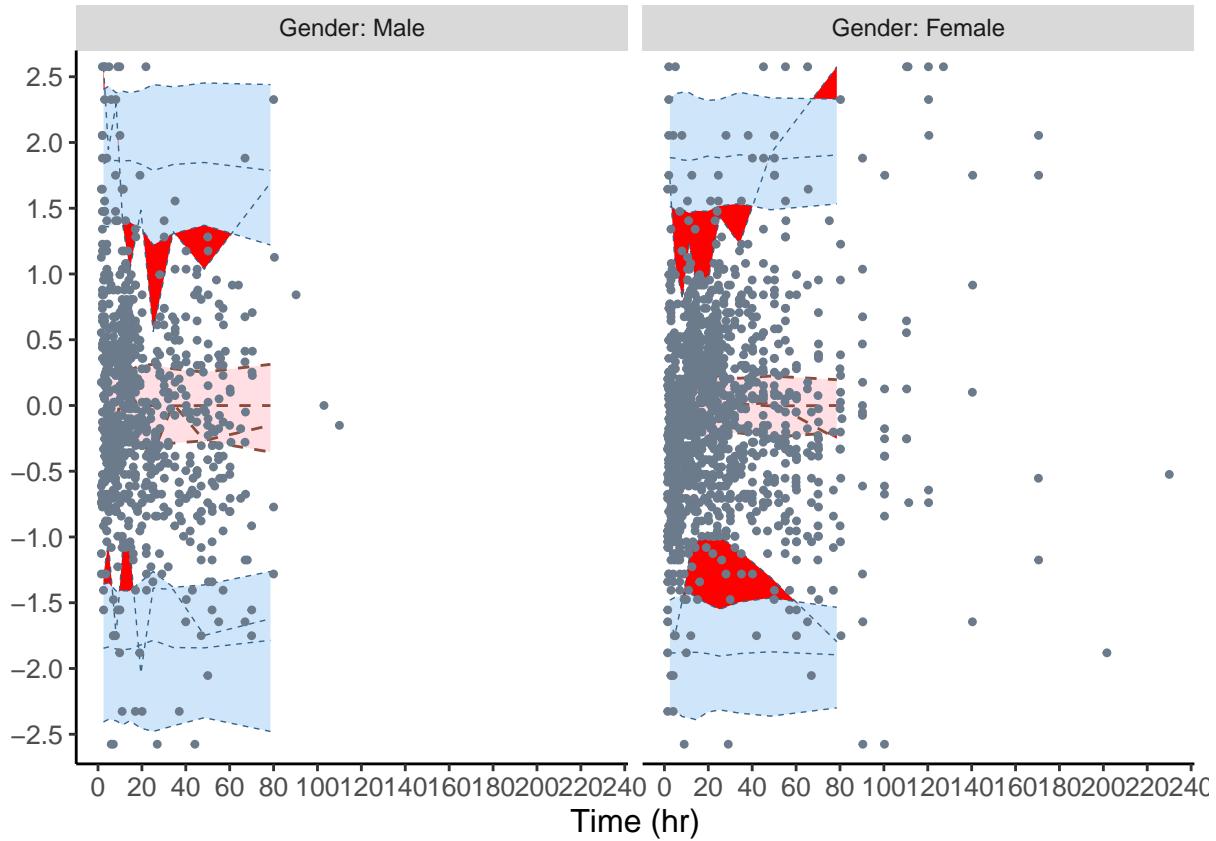


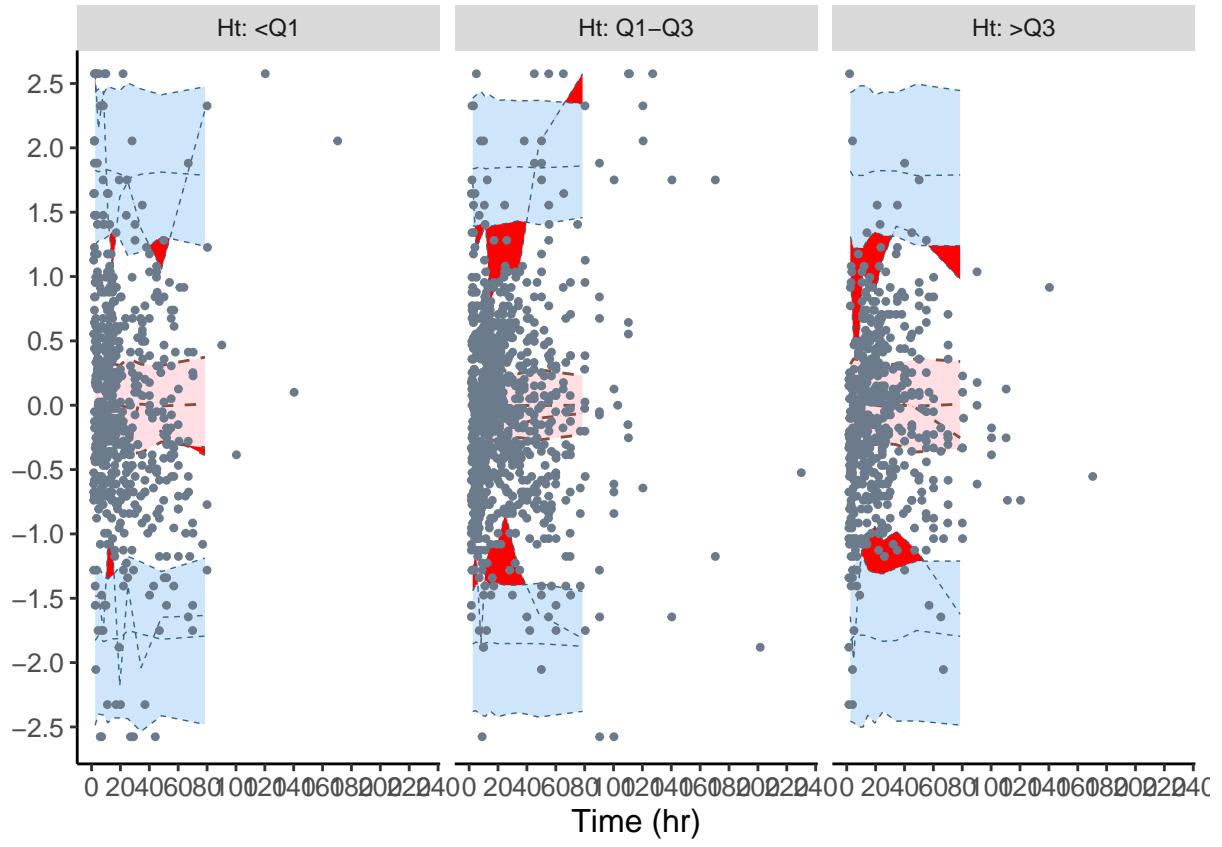


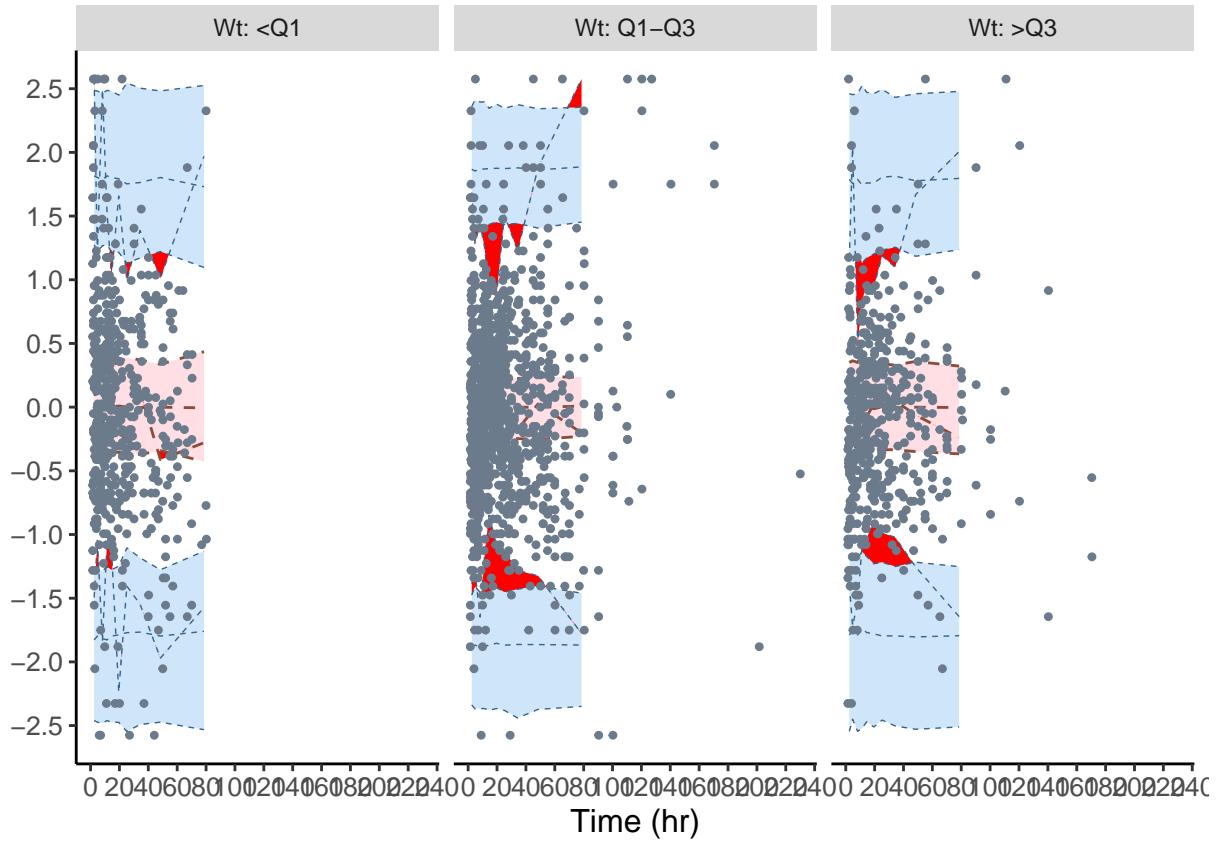


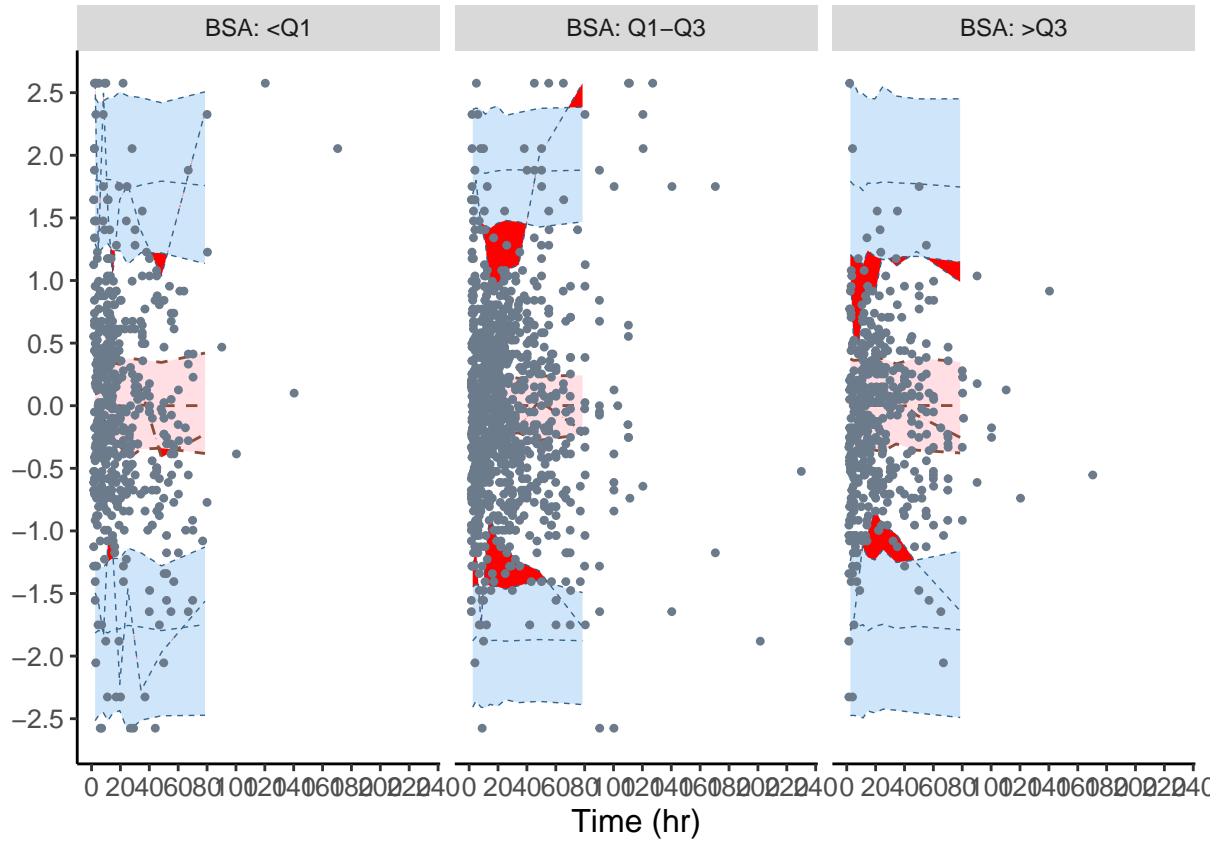


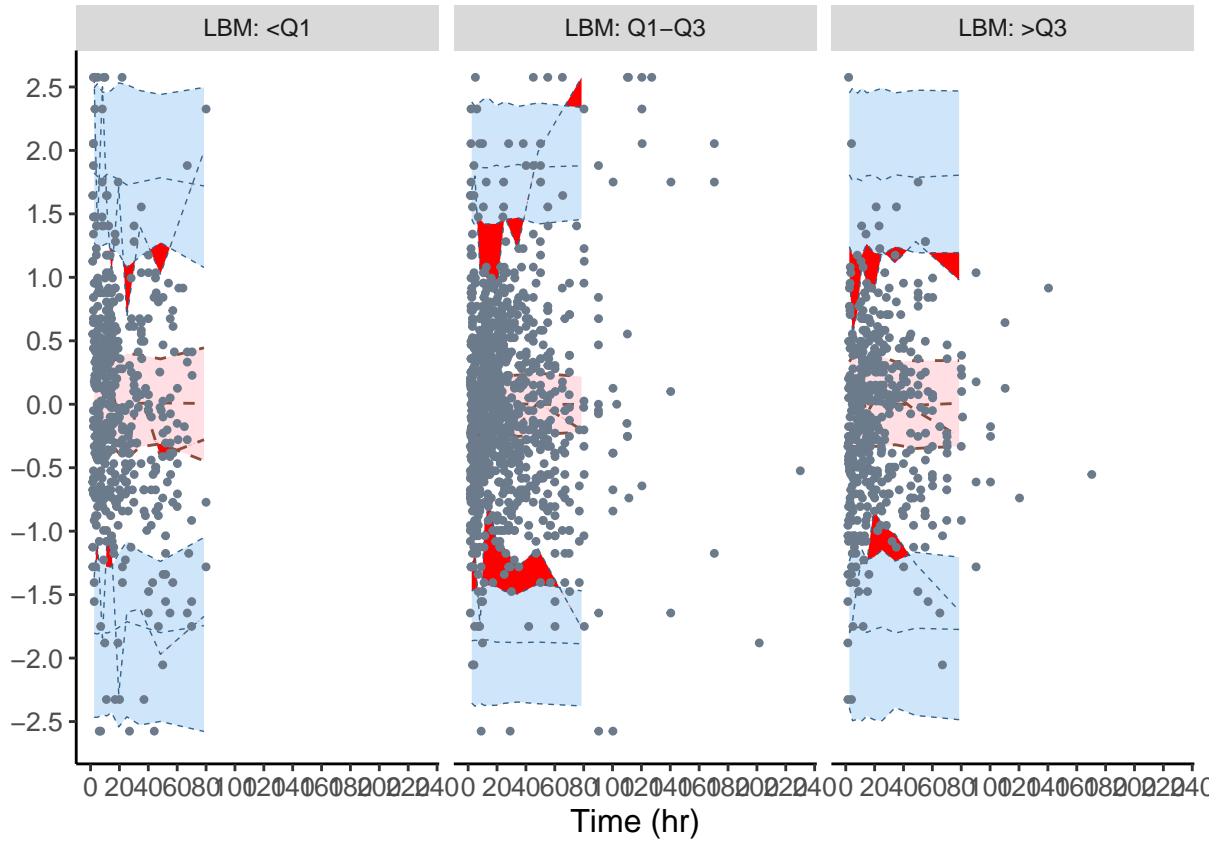


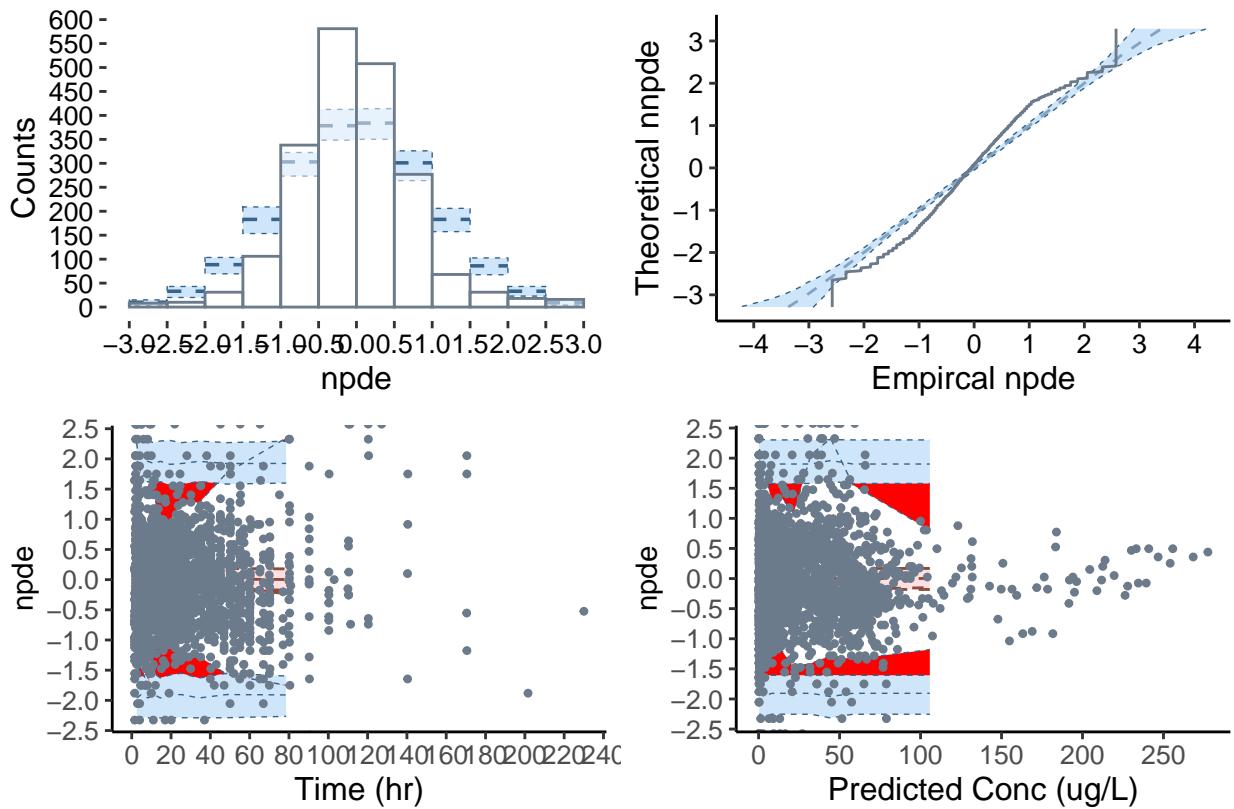








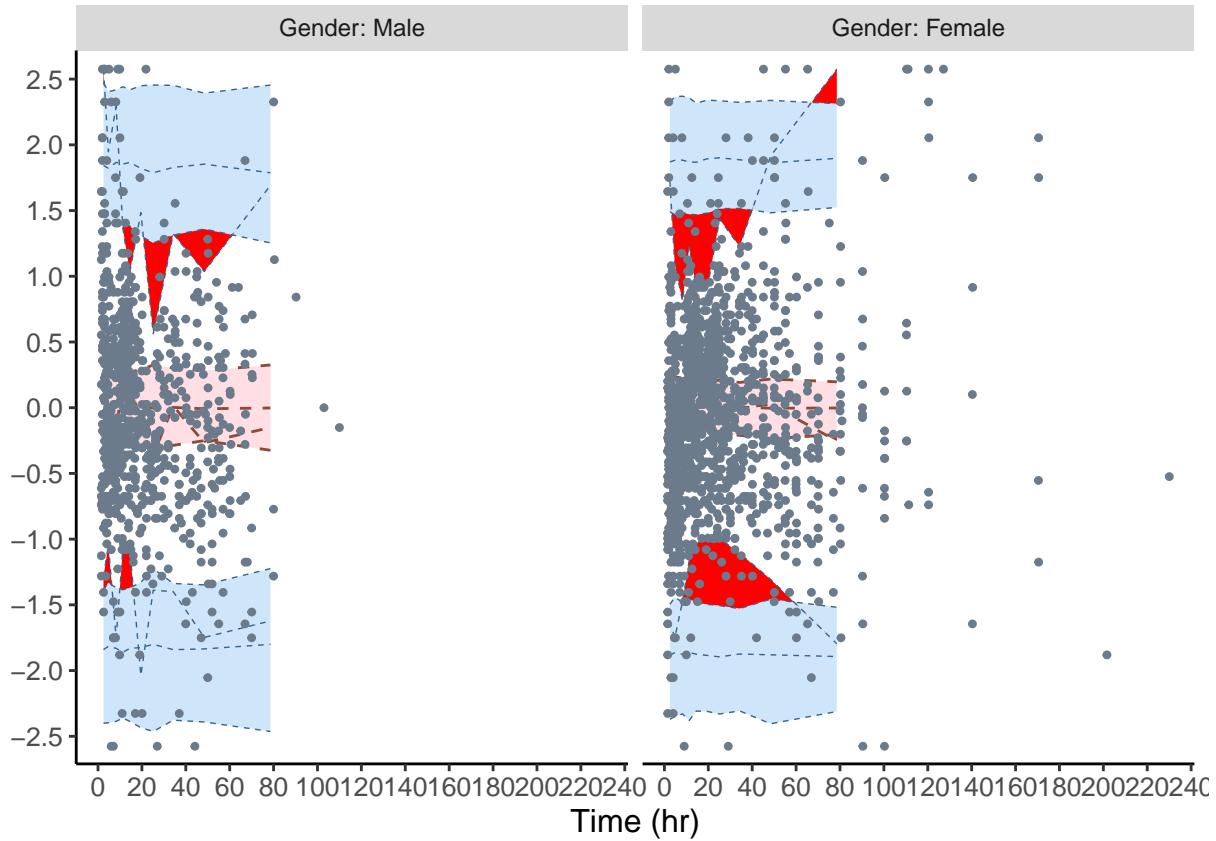


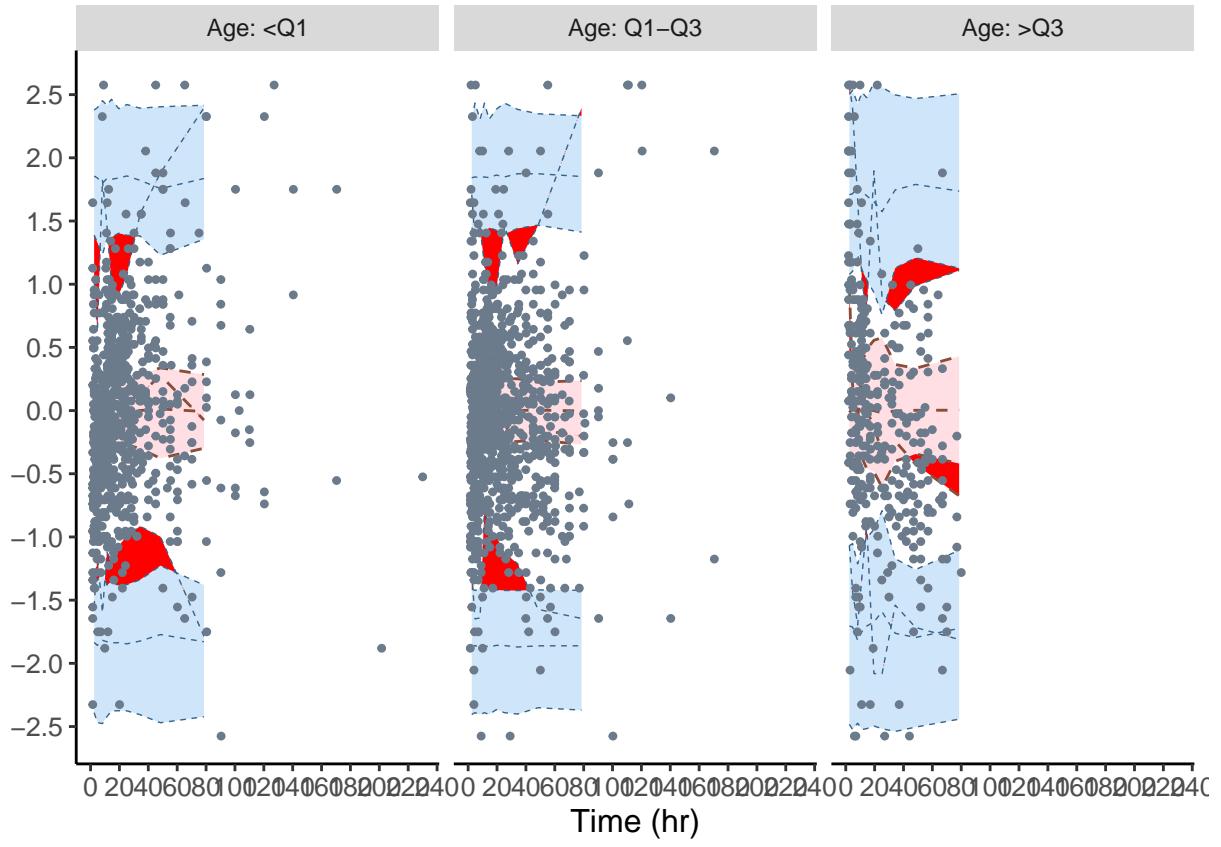


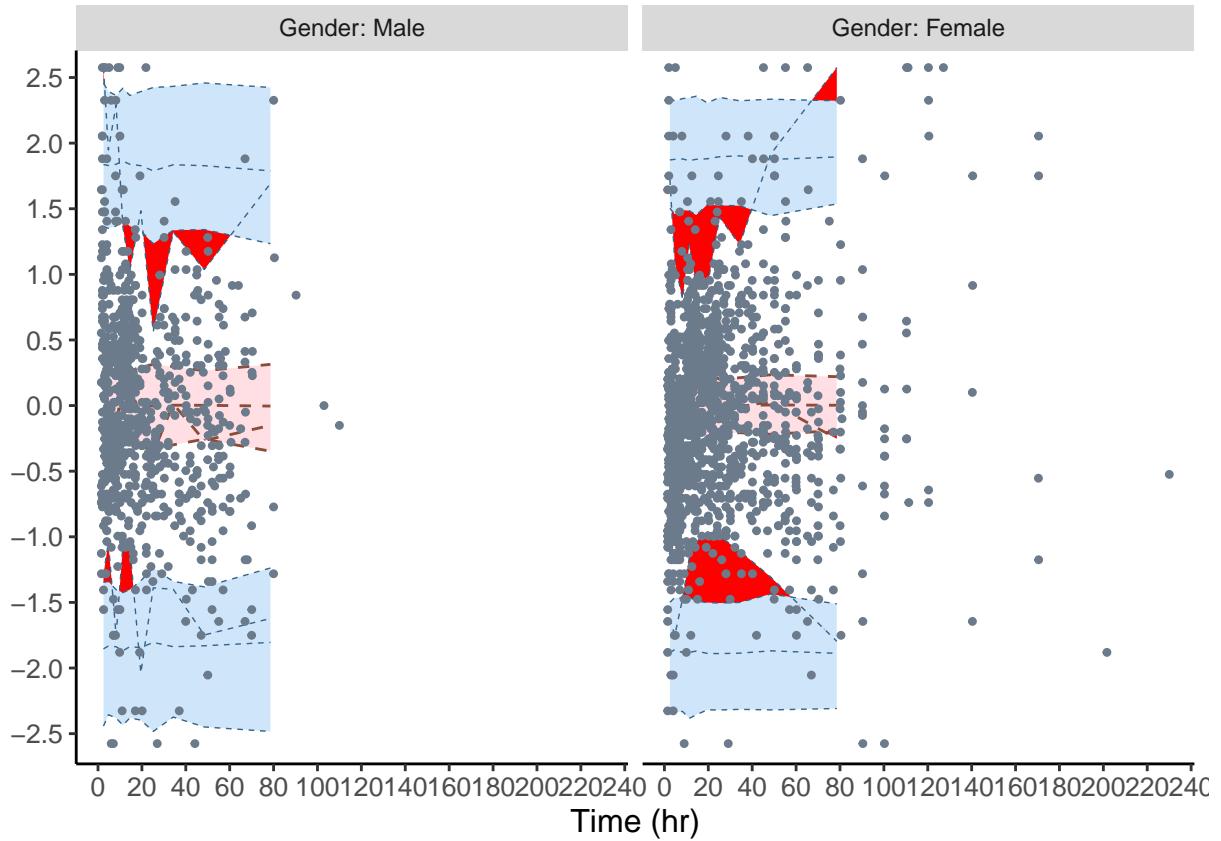
#### Parameters versus covariates

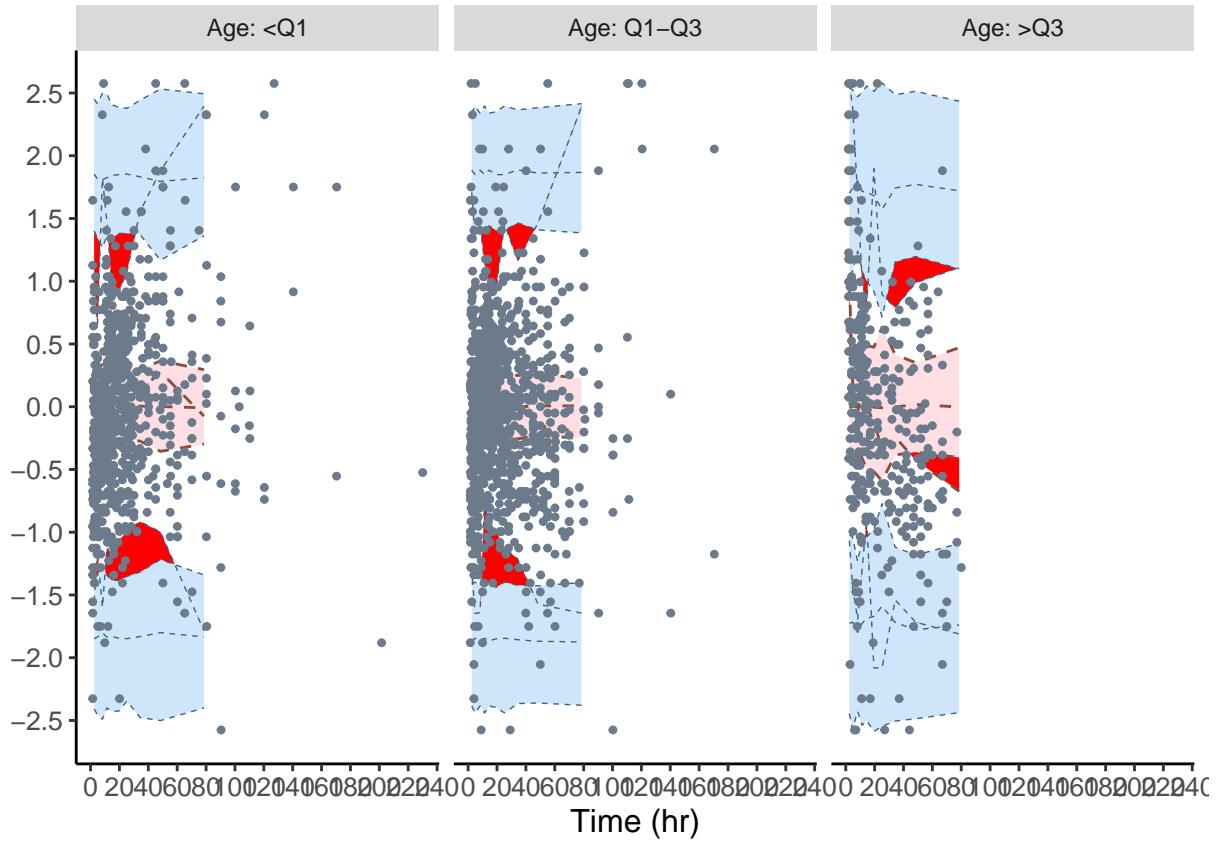
Not implemented yet

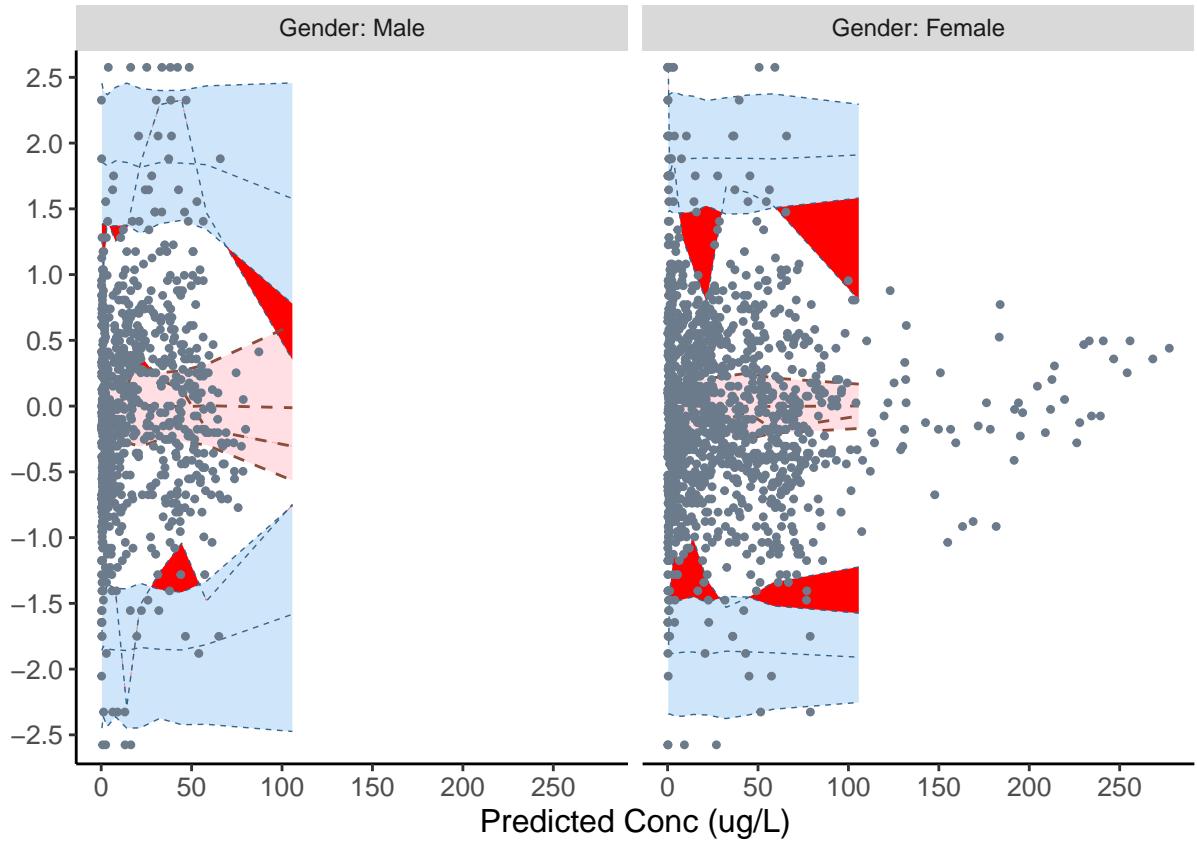
- **Problems (à résoudre Romain)**
  - error when using `plot.type="cov.scatter"`
  - normally we should get a plot of npde versus the covariates

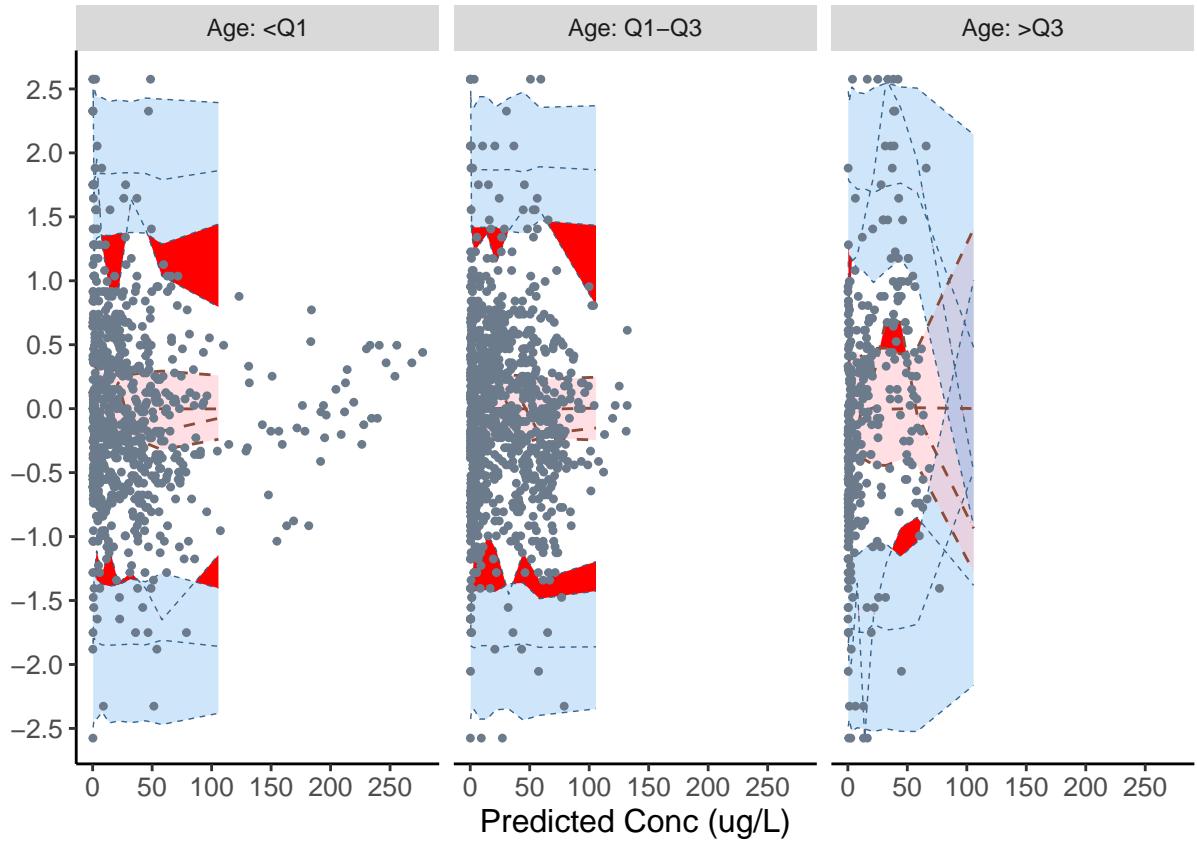


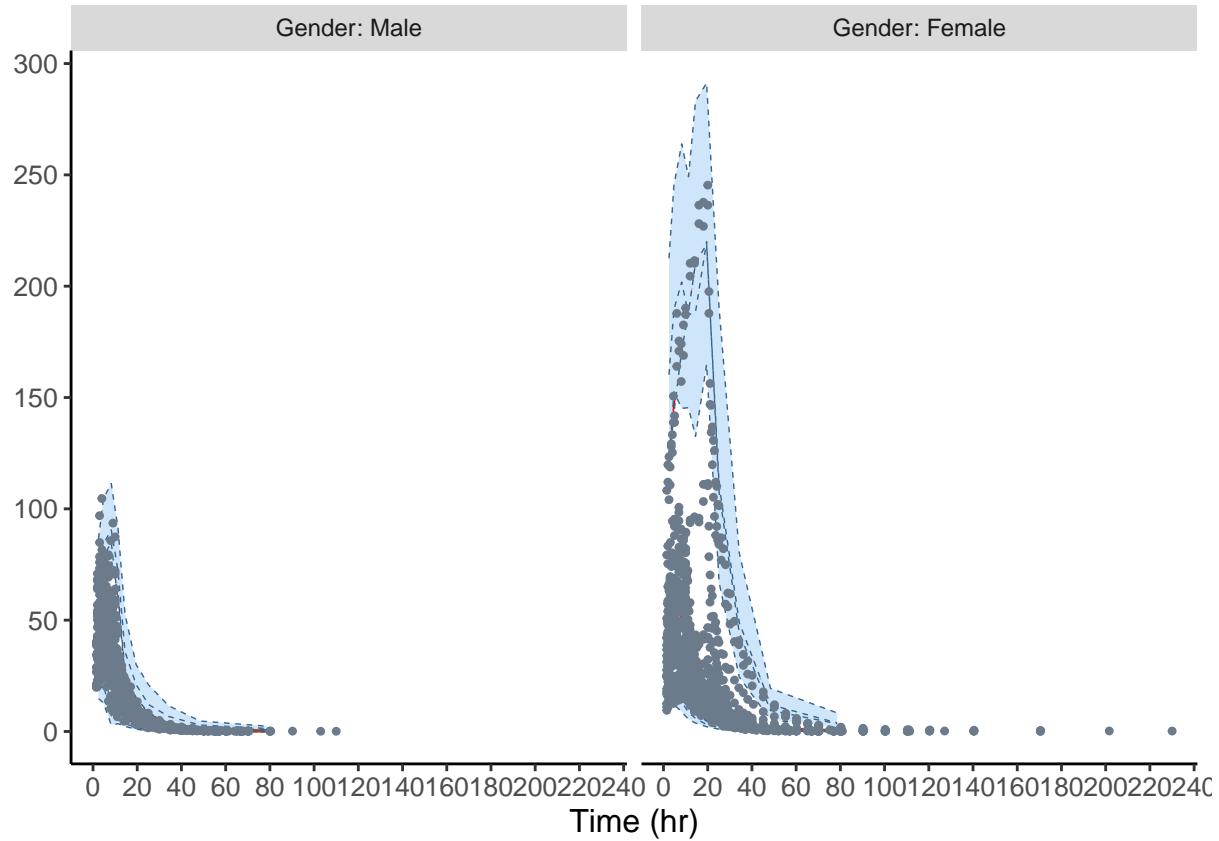


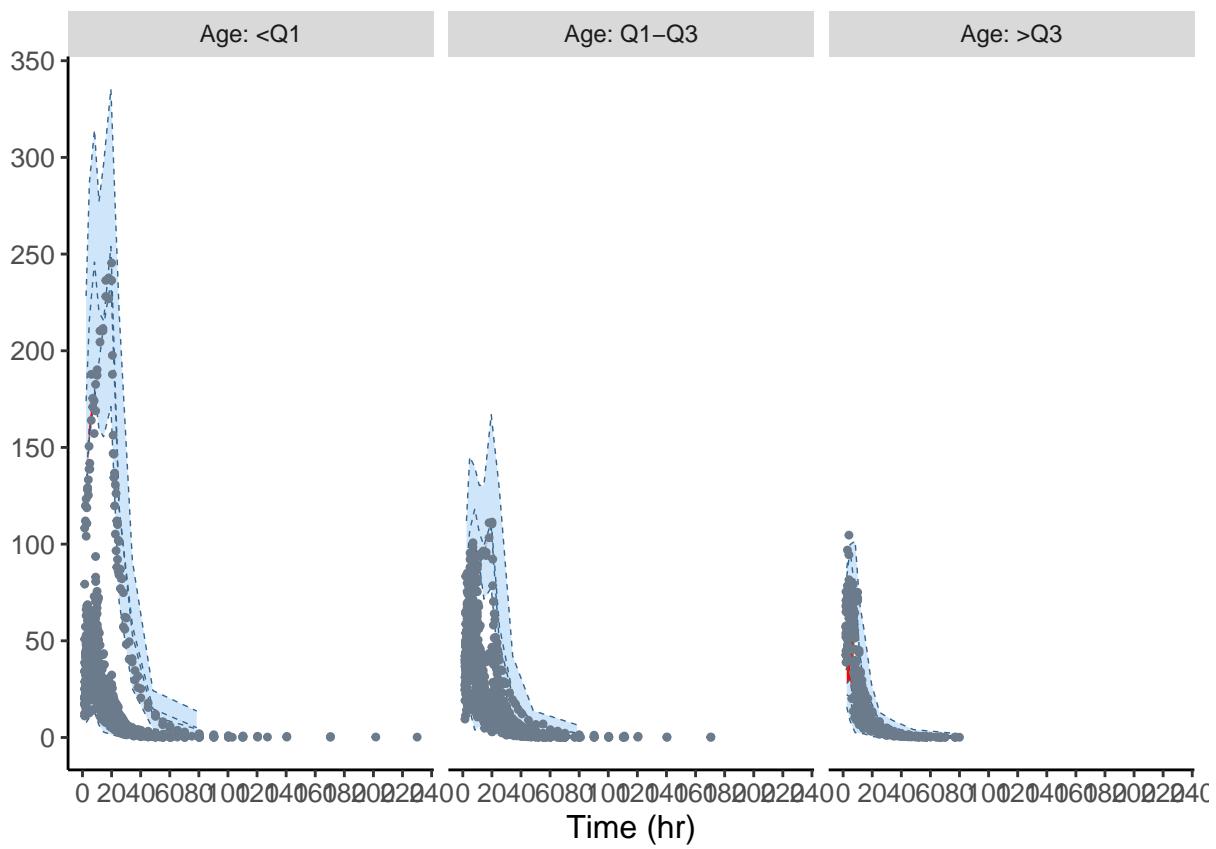


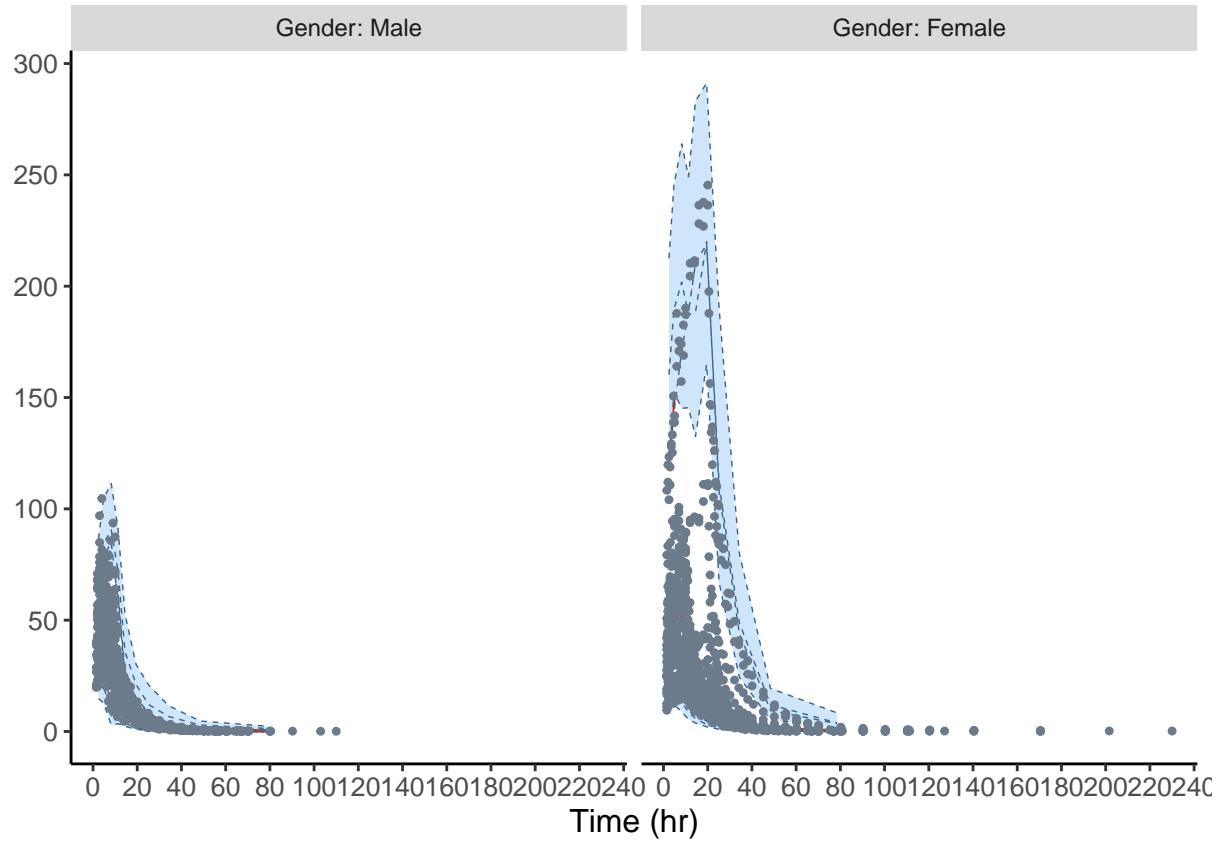


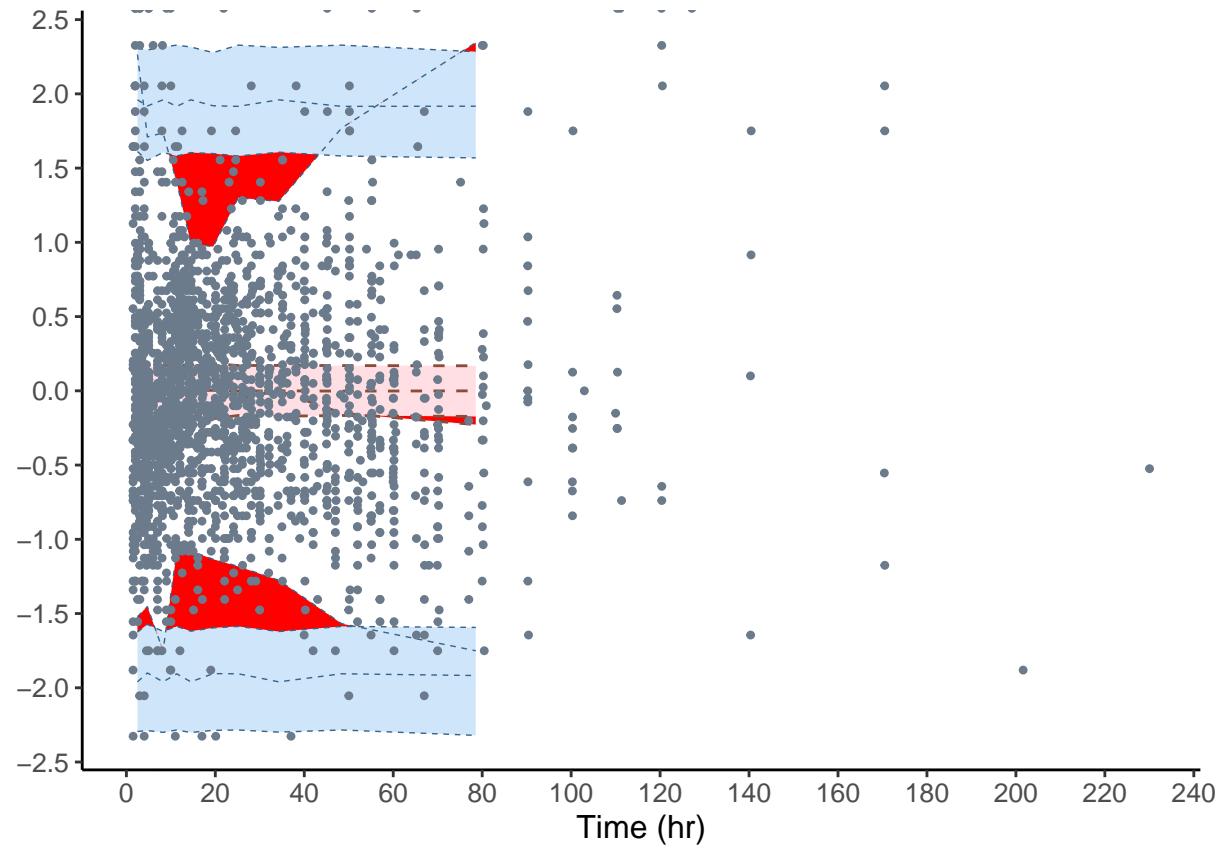


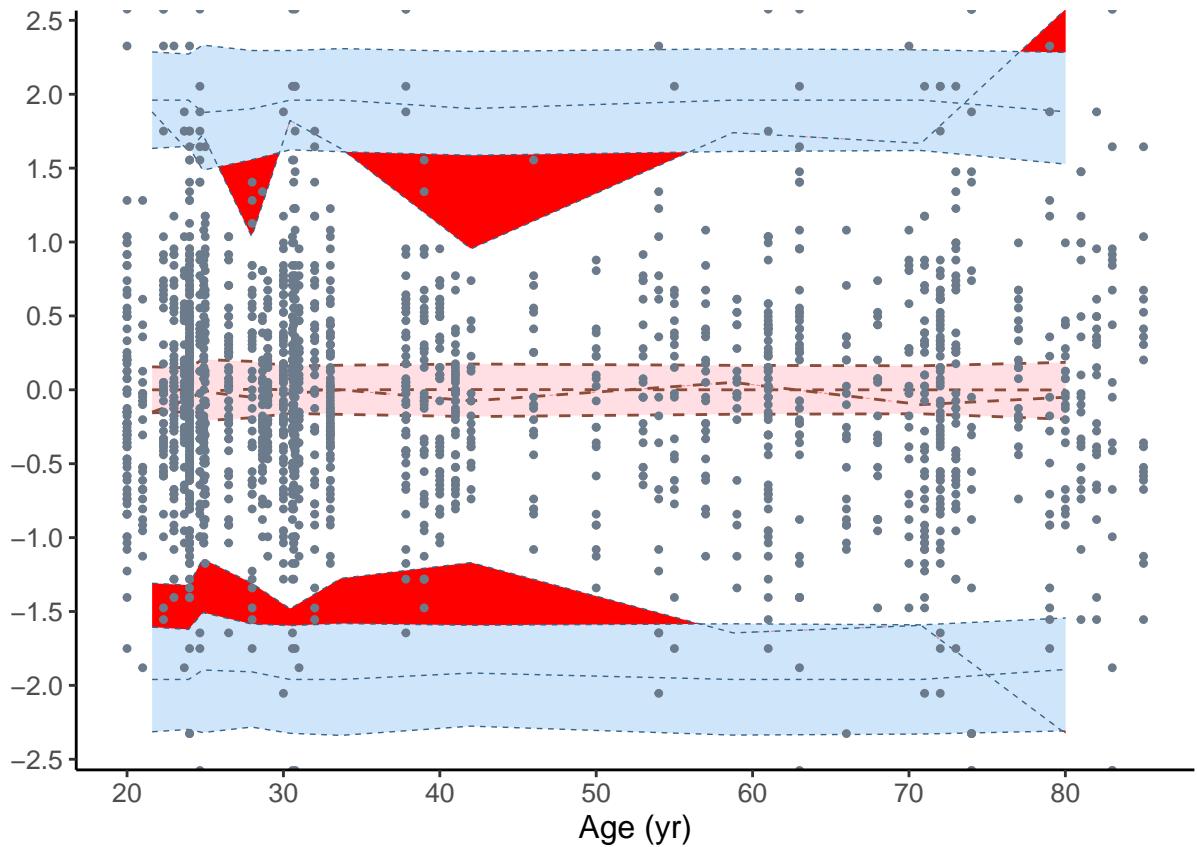


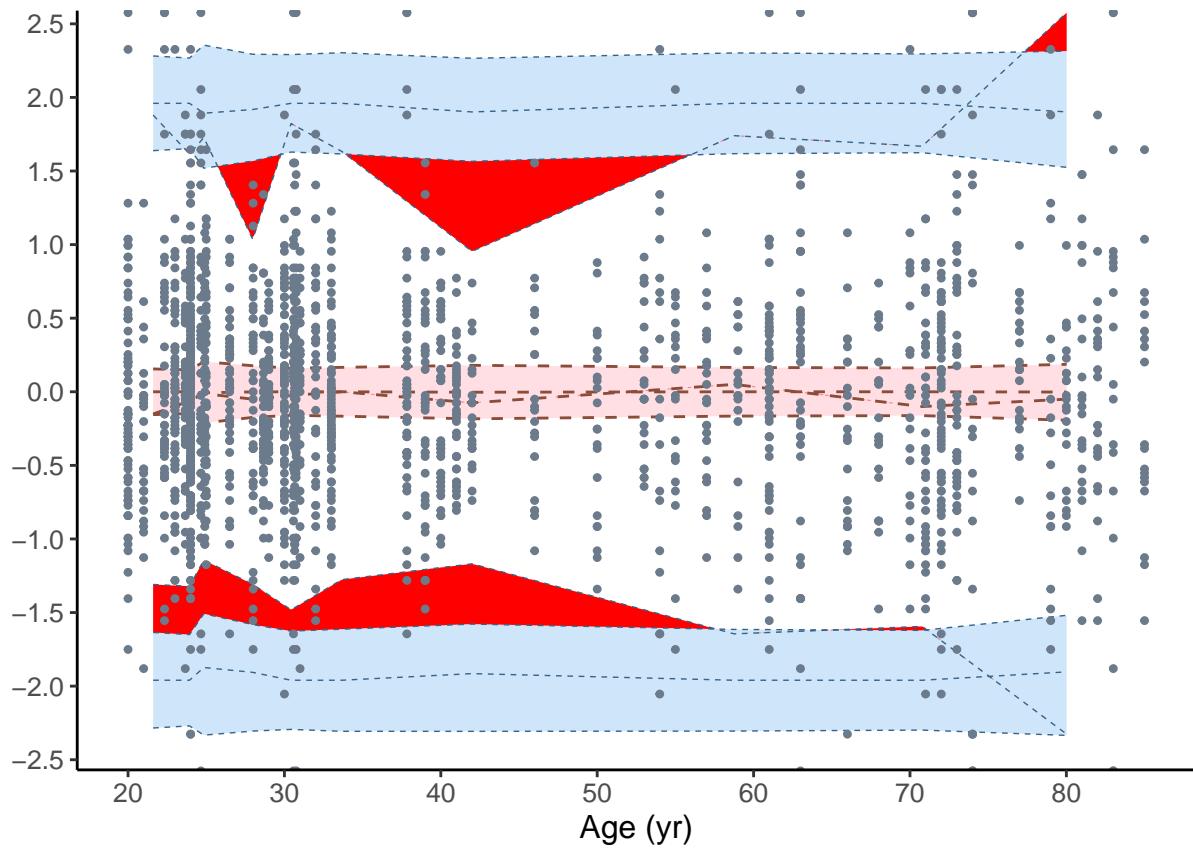












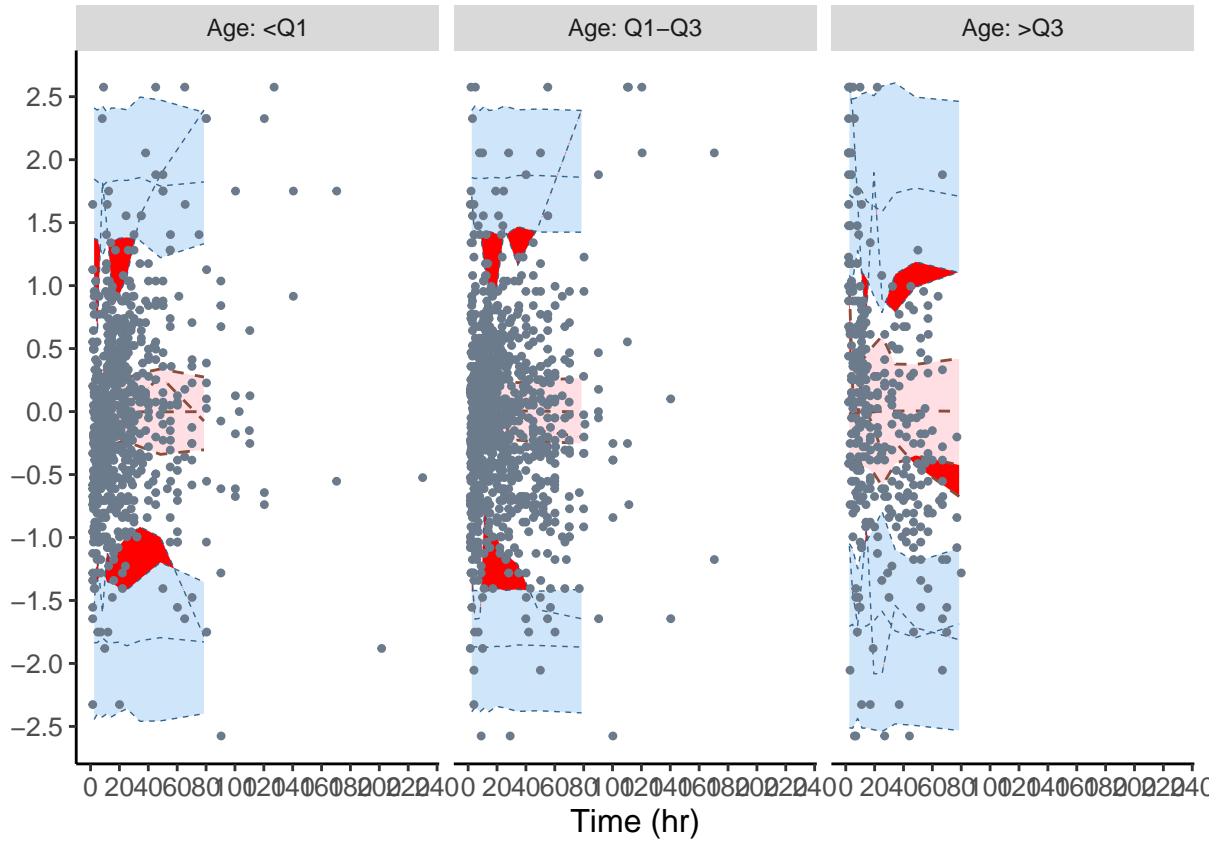
```

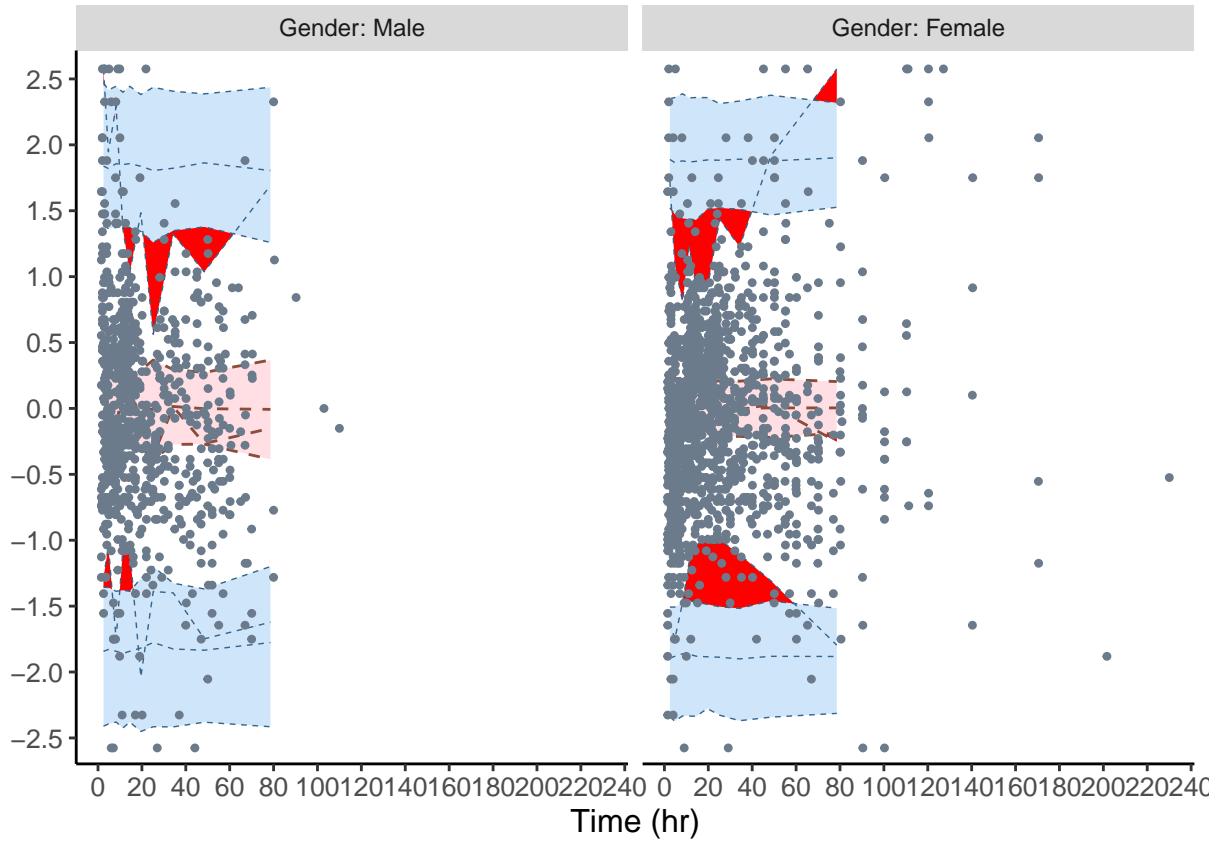
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

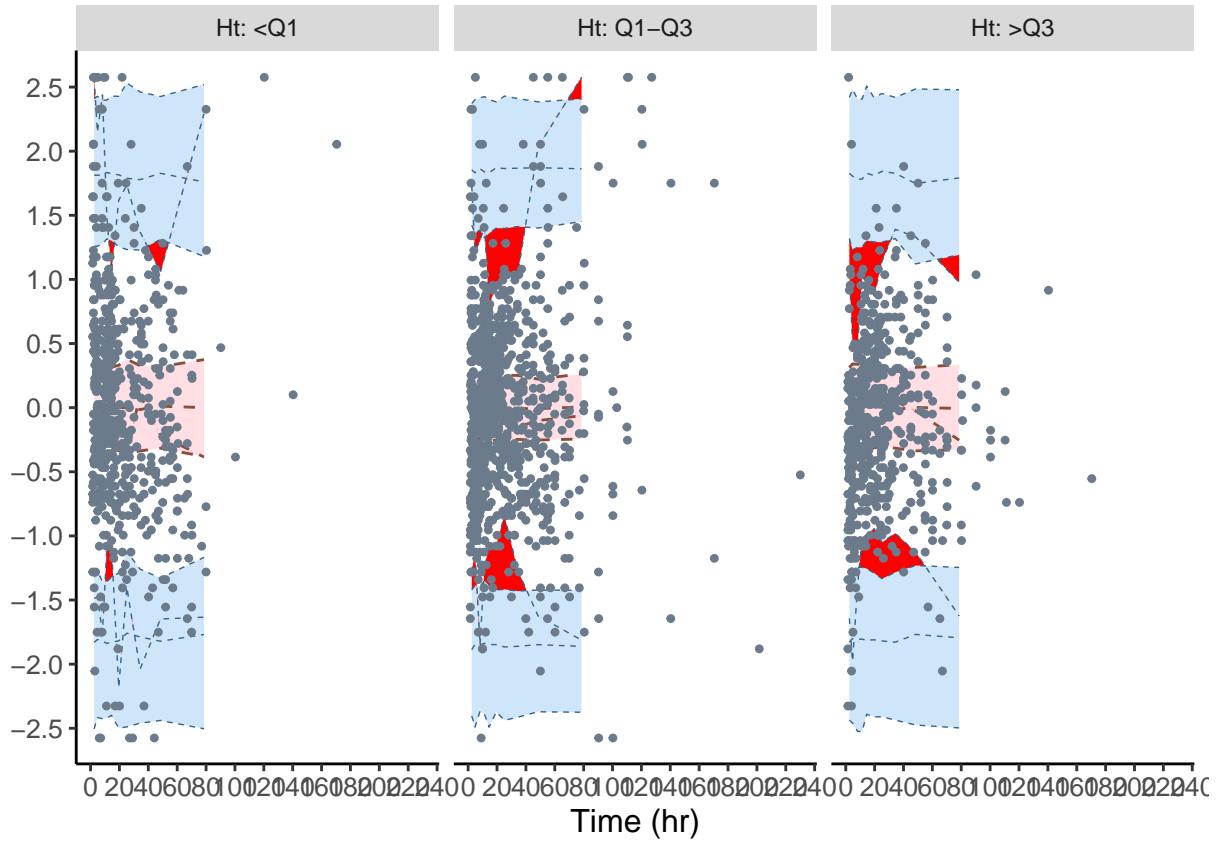
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

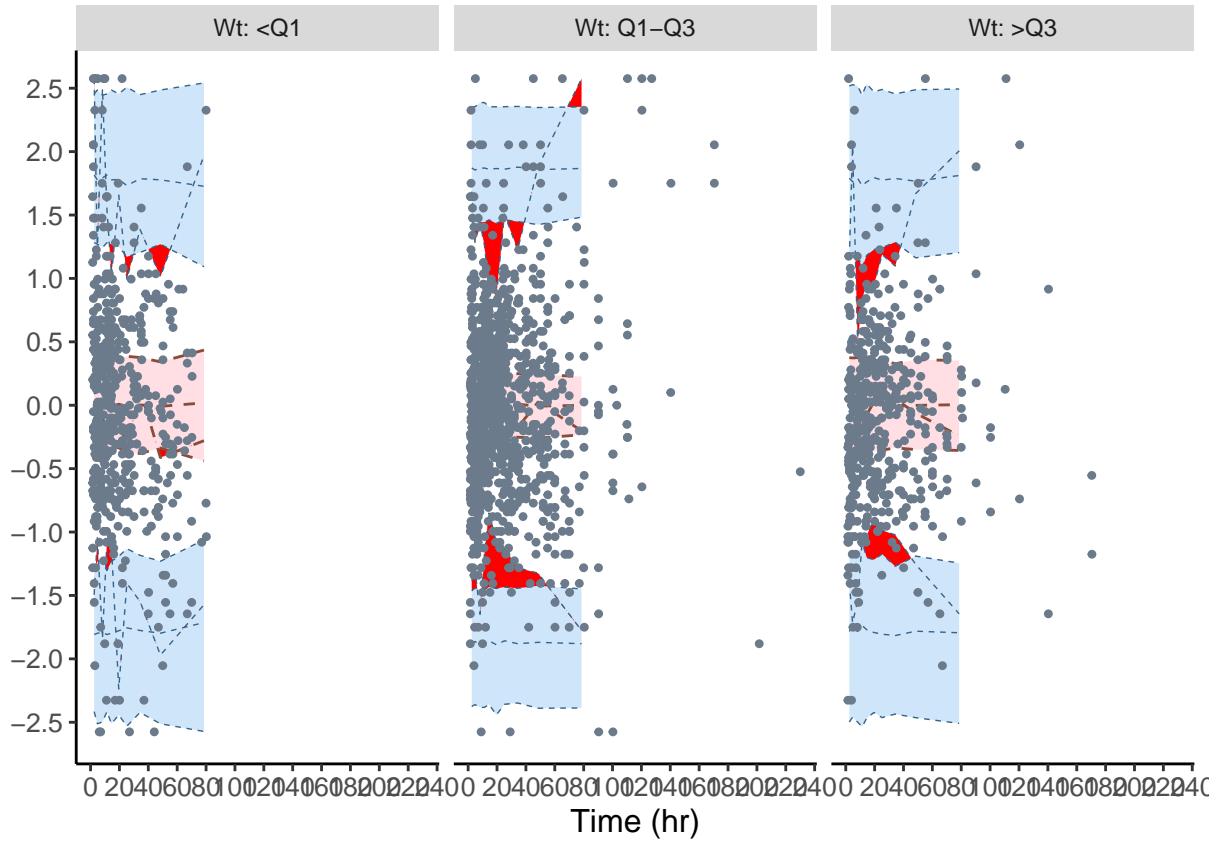
## Error in Summary.factor(structure(c(2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L,
##   'min' not meaningful for factors

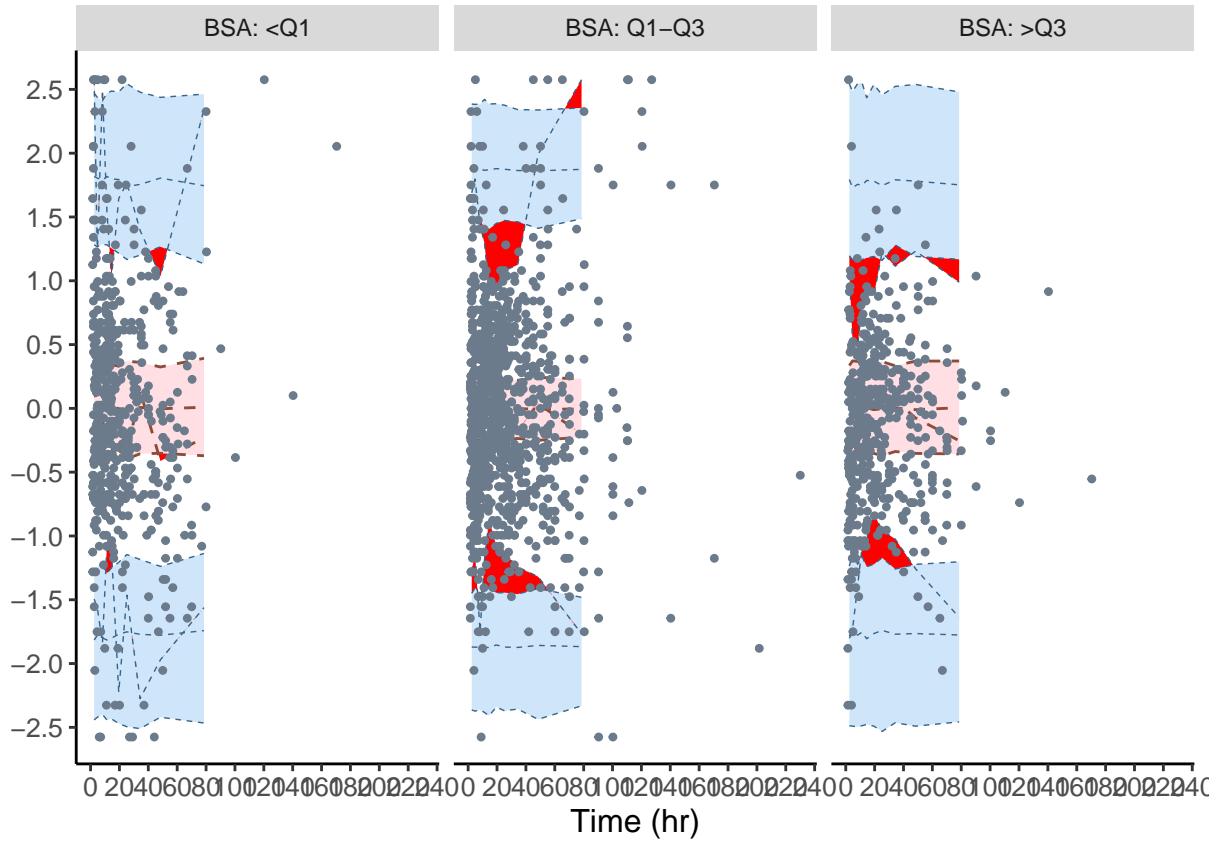
```

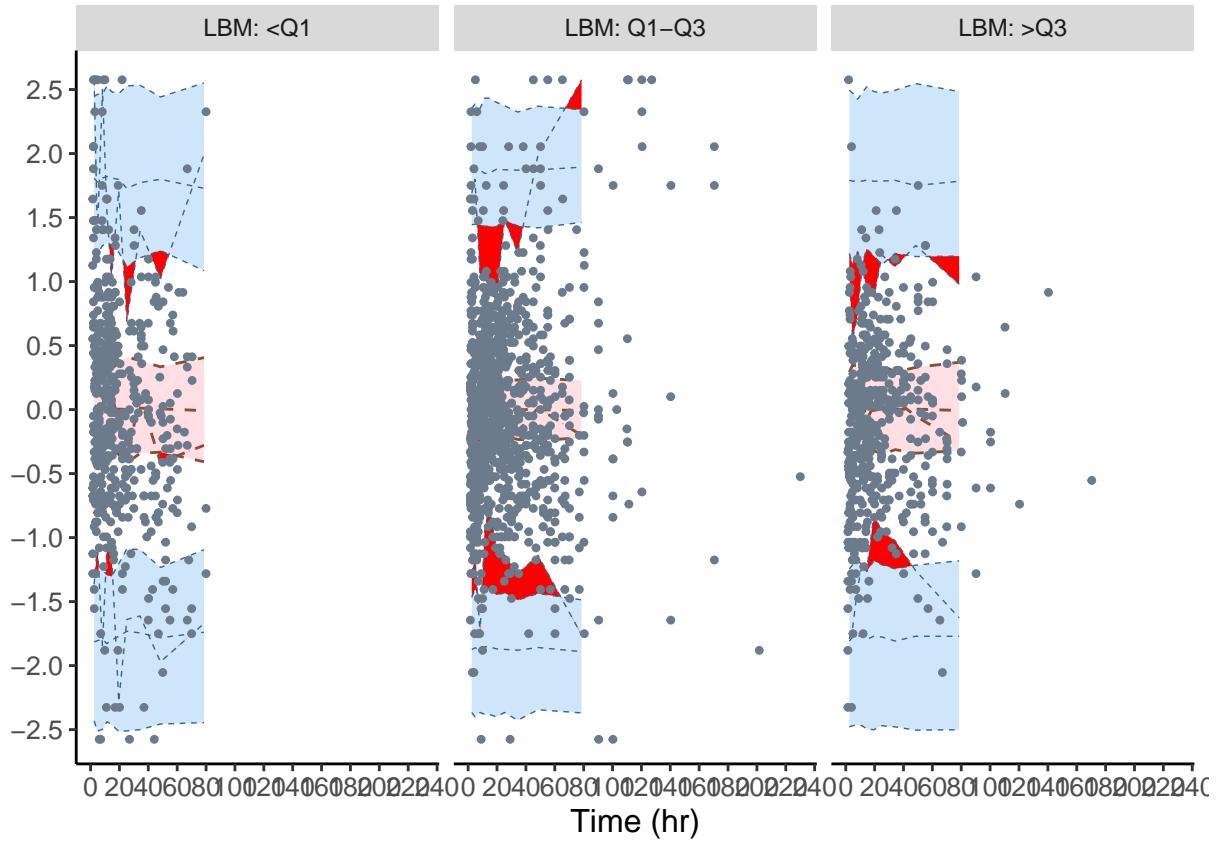












```
## [1] "ok"
```

## Creating the plots for the user guide

### Theophylline

- Default plots and VPC: **OK pour doc**
- **Problem** with plot of data (see above)

### Viral load data

- **TODO**
  - 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) ?
    - \* 2 graphes de données (voire 3)
    - \* 2 graphes de VPC côte à côté (=> donner code correspondant dans la doc)
    - \* **note** si c'est l'option *plot.default* il faudrait en changer le nom (pas du tout explicite), en un truc comme *return* ou *return.plot...*
  - corriger le graphe de VPC

## End of file, deactivating development mode

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