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
- **TODO** simplify output for *show()* function (v3.0 or v3.1 ?)

Setup, loading libraries

Install package in development mode

```
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

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
##   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  NaN          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 ?)

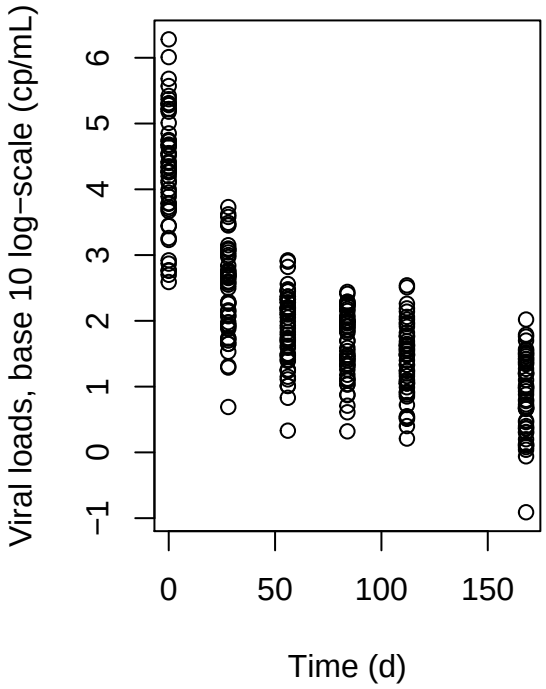
```
## Object of class NpdeObject
## ----------------------------------------
## ----        Component data           ----
## ----------------------------------------
## Object of class NpdeData
##     Structured data: Conc ~ Time | ID
## This object has the following components:
##      data: data
##      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)
```

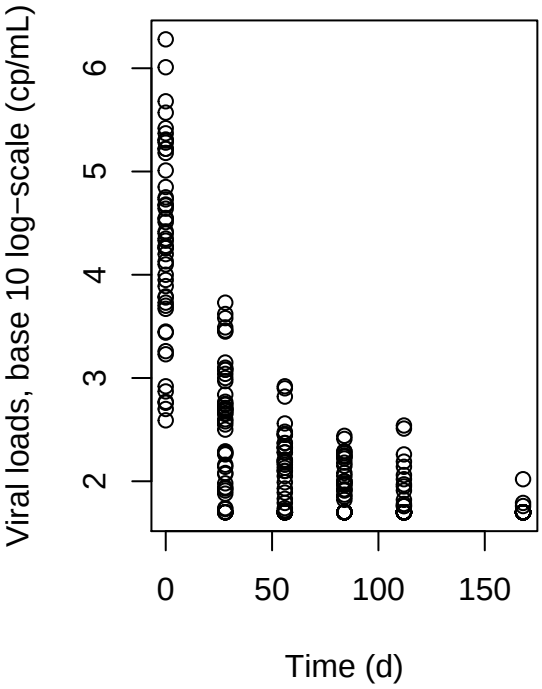
```
##     completed responses (ycomp) for censored data
##     decorrelated responses (ydobs)
##   the dataframe has  120 non-missing observations and 132 lines.
## First 10 lines of results, removing missing observations:
##       ypred ycomp    pd       ydobs      npde     tnpde
## 2  2.923864  2.84 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 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
##   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= -0.0343   (SE= 0.06 )
##        variance= 1.072   (SE= 0.088 )
##        skewness= 0.009137
##        kurtosis= 0.1297
## ------------------------------------------
##
```

```
## Statistical tests
##   t-test                     : 0.566
##   Fisher variance test       : 0.375
##   SW test of normality       : 0.898
## Global adjusted p-value      : 1
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## ------------------------------------------

## ------------------------------------------
## Distribution of npde :
##       nb of obs: 300
##            mean= -0.0518   (SE= 0.052 )
##        variance= 0.8019   (SE= 0.066 )
##        skewness= -0.1503
##        kurtosis= 0.02162
## ------------------------------------------
##
## Statistical tests
##   t-test                     : 0.317
##   Fisher variance test       : 0.00998 **
##   SW test of normality       : 0.377
## Global adjusted p-value      : 0.0299 *
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## ------------------------------------------

## ------------------------------------------
## Distribution of npde :
##       nb of obs: 300
##            mean= -0.02917   (SE= 0.053 )
##        variance= 0.8509   (SE= 0.07 )
##        skewness= -0.1542
##        kurtosis= 0.03102
## ------------------------------------------
##
## Statistical tests
##   t-test                     : 0.584
##   Fisher variance test       : 0.0582 .
##   SW test of normality       : 0.687
## Global adjusted p-value      : 0.175
## ---
## 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
```

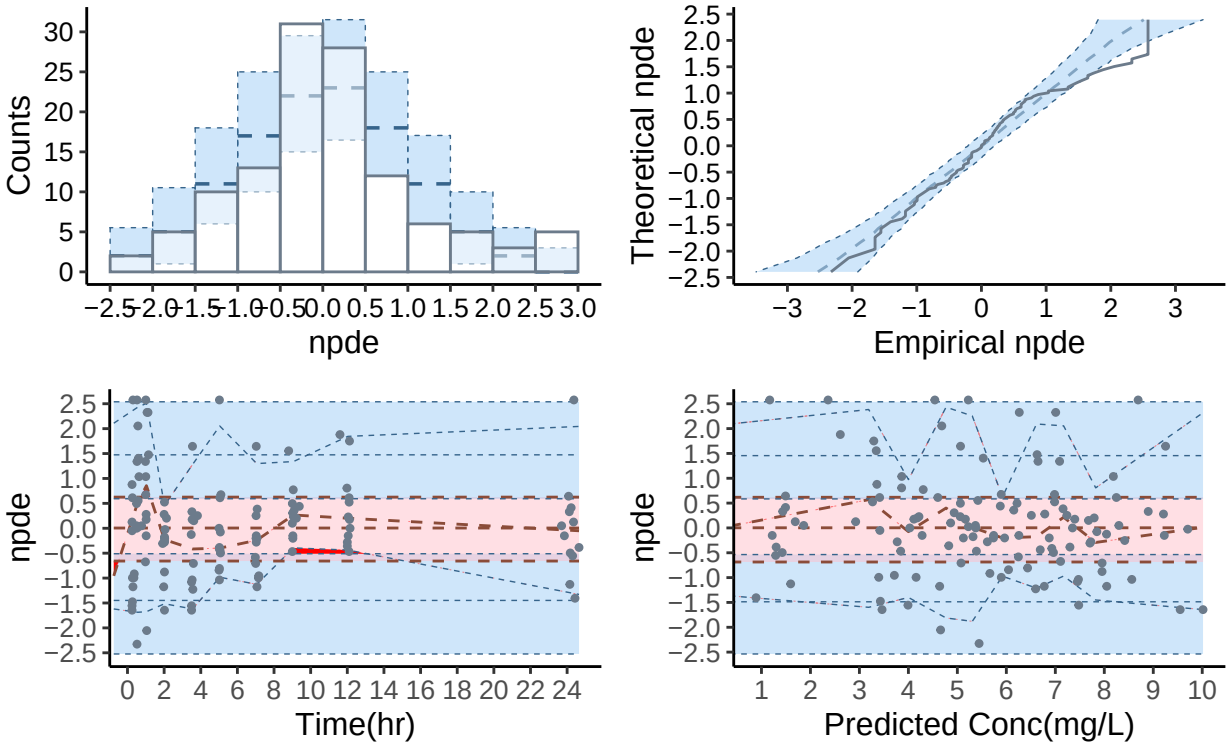
```
##   t-test                     : 0.041 *
##   Fisher variance test       : 0.0822 .
##   SW test of normality       : 0.687
## Global adjusted p-value      : 0.123
## ---
## Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
## ------------------------------------------
## ------------------------------------------
## 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
## ------------------------------------------
```

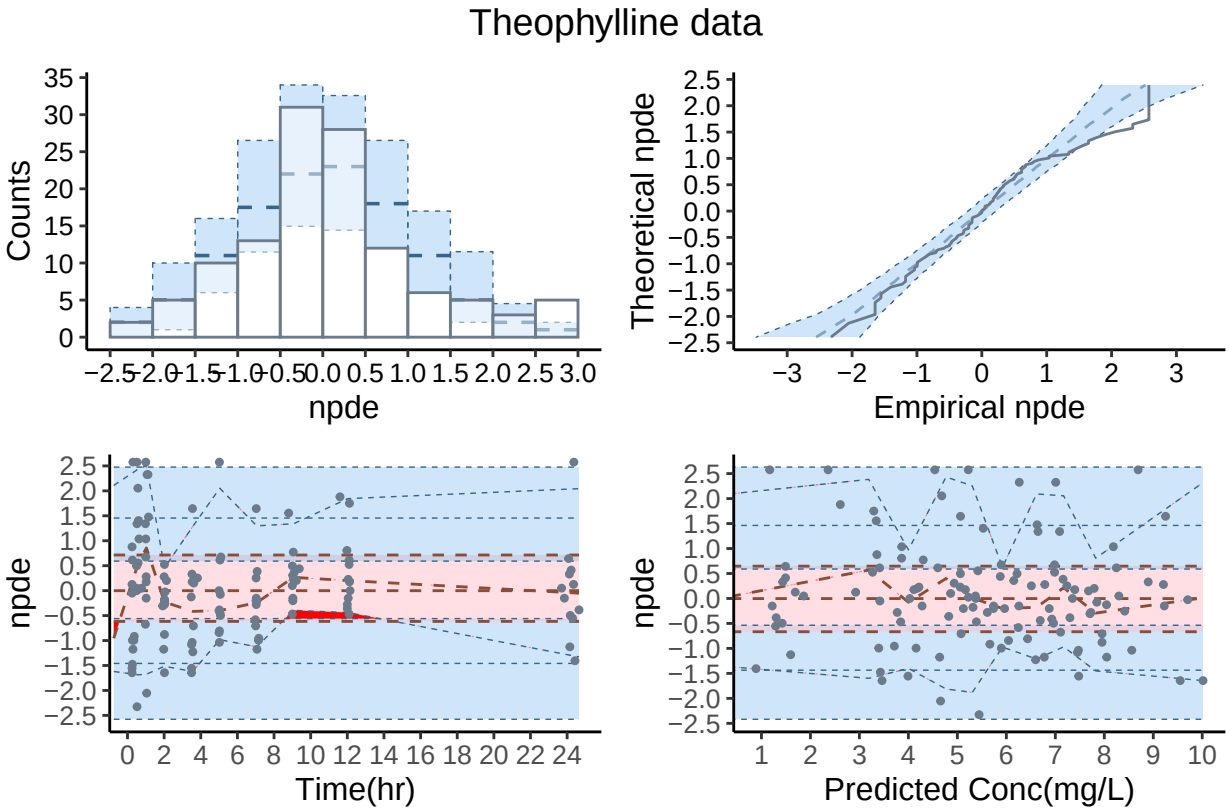
Test plots - default options

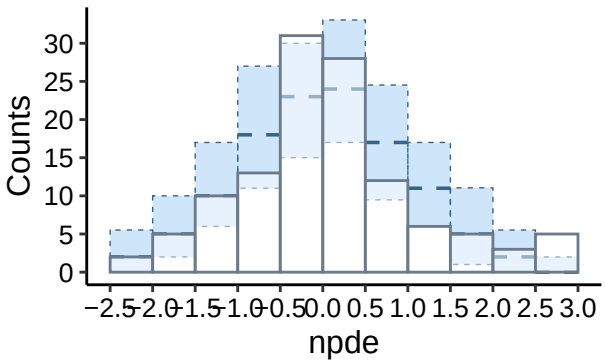
Default plots for theophylline data

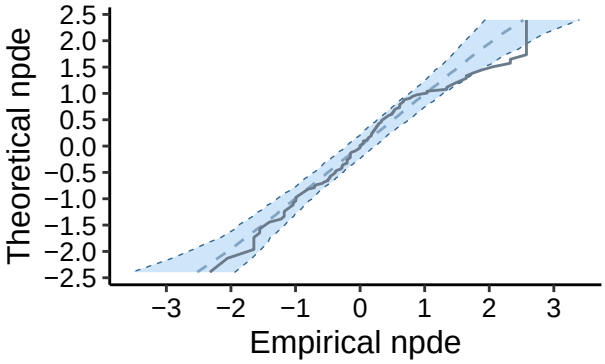
- **BUGS Romain TODOO**
  - **solved** title: in the waffle plot, only one title should appear, above all the 4 graphs (not one title per graph)
  - ajout de la condition plot.opt$plot.default==FALSE) pour l'affichage des titres dans les plot
  - **solved** no axes in the histogram plot => add
  - **solved X-range on PI** extend prediction band to min/max X (it seems to stop at the center of the bin) => option ???

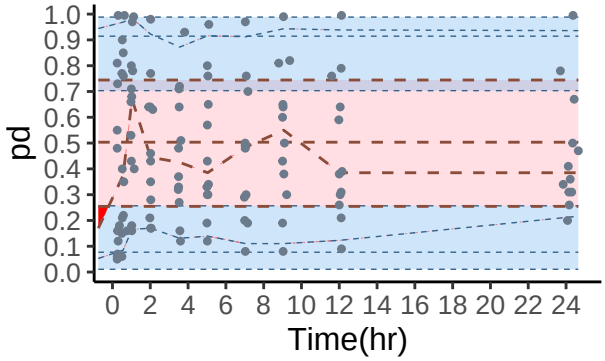
  - dans plotNpde-auxScatterPlot: 1er bin avec valeur minimal de la borne et dernier bin avec valeur maximal de la borne. 1er et dernier le plot de s'arrête pas au milieu.
  - **global options** not passing on to the functions
    * **solved** size
      · **solved** normalement si l'utilisateur spécifie "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 défault (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'histograme
- **WARNING** attention aux effets de bord !!!
  - dans **set.plotoptions.NpdeObject**, xlab et ylab étaient fixés par défaut or on veut pouvoir les fixer automatiquement dans les fonctions graphiques tout en laissant l'utilisateur les passer en argument => on ne peut pas les mettre par défaut (xlab et ylab dépendent ici des variables qu'on veut tracer) => réfléchir avant de faire un truc comme ça (ou bien documenter qu'on puisse voir pourquoi ça plante, sinon on perd du temps à trouver pourqoui ça ne marche pas)
- **Note** : dans la dernière version, la fonction NpdeControl a toujours dans ses argumens xlab="" et ylab"" + which et xaxis Dabs plotNpde-distributionPlot l.156-l.168 il y a différents tests qui restent en conséquence. Pour le moment ça n'empêche pas à l'utilisateur/utilisatrice de changer les labels à sa guise.

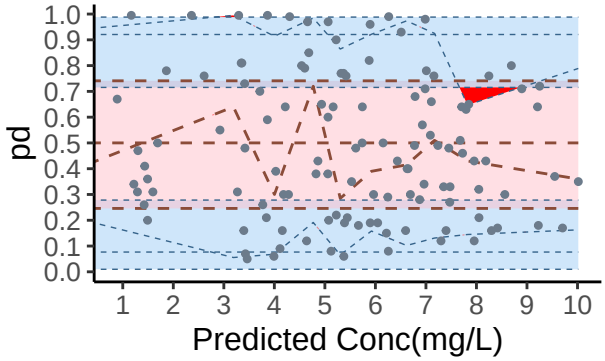


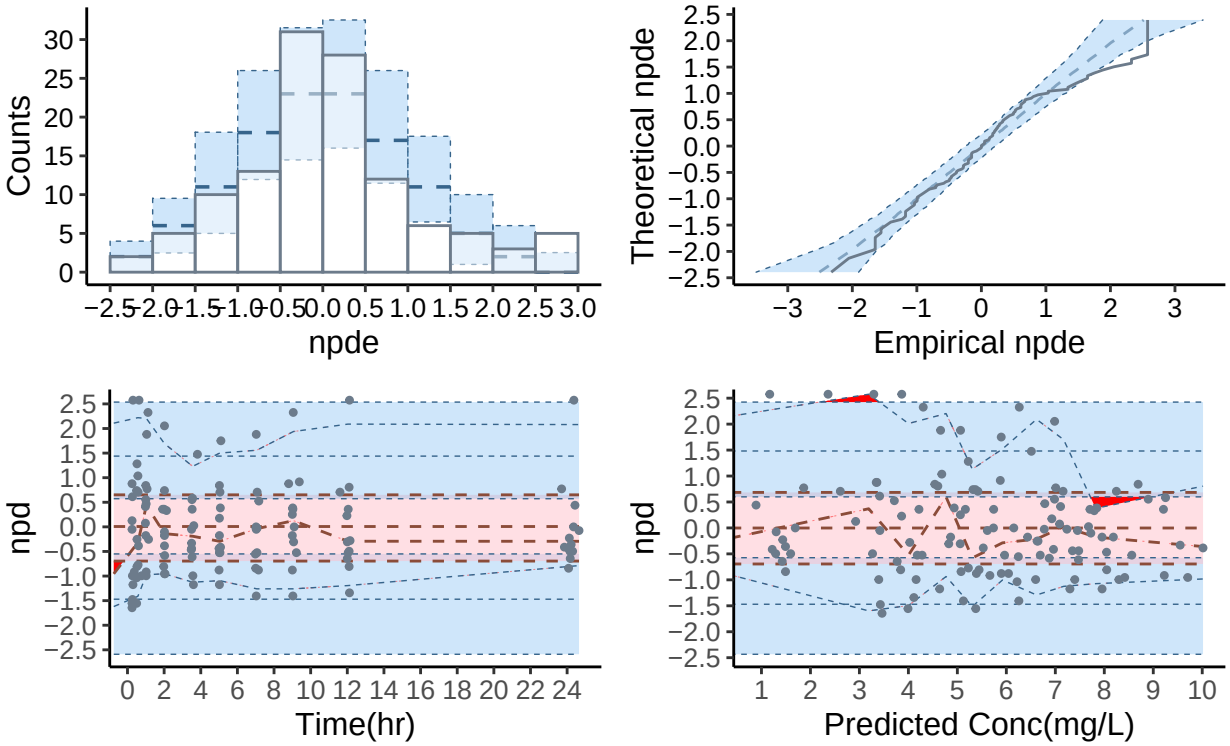

Theophylline data
Counts
npde
Theoretical npde
Empirical npde
npde
Time(hr)
npde
Predicted Conc(mg/L)


Counts
30
25
20
15
10
5
0
−2.5 −2.0 −1.5 −1.0 −0.5 0.0 0.5 1.0 1.5 2.0 2.5 3.0
npde

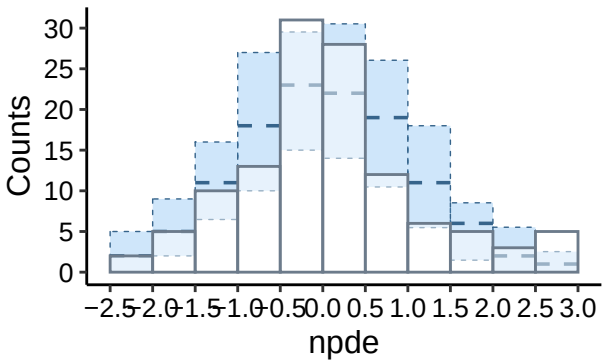

Theoretical npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3 −2 −1 0 1 2 3
Empirical npde

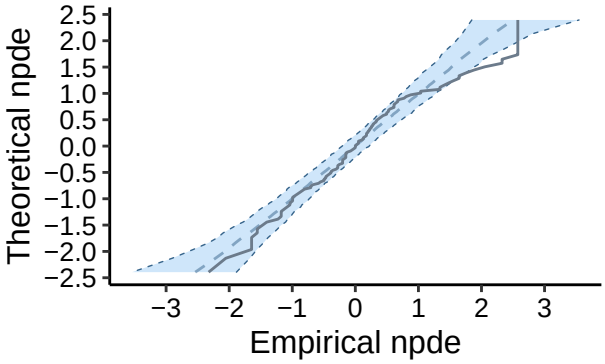

pd
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0 2 4 6 8 10 12 14 16 18 20 22 24
Time(hr)

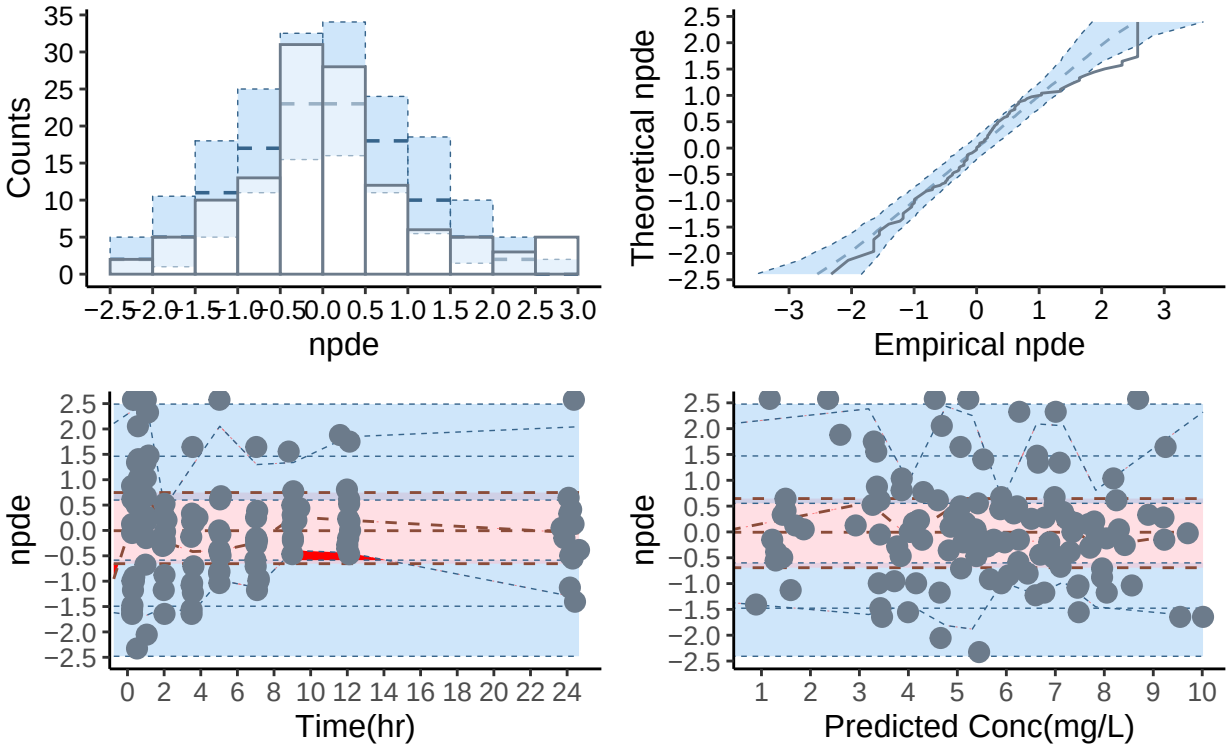

pd
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1 2 3 4 5 6 7 8 9 10
Predicted Conc(mg/L)

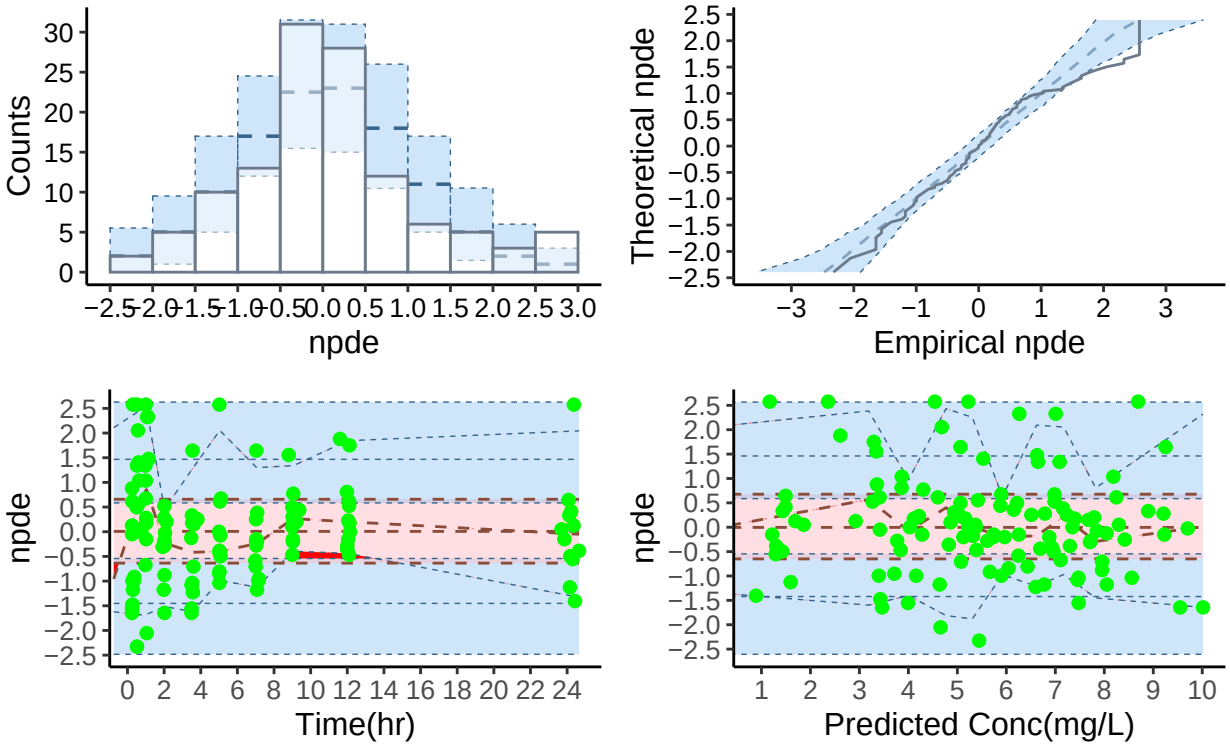


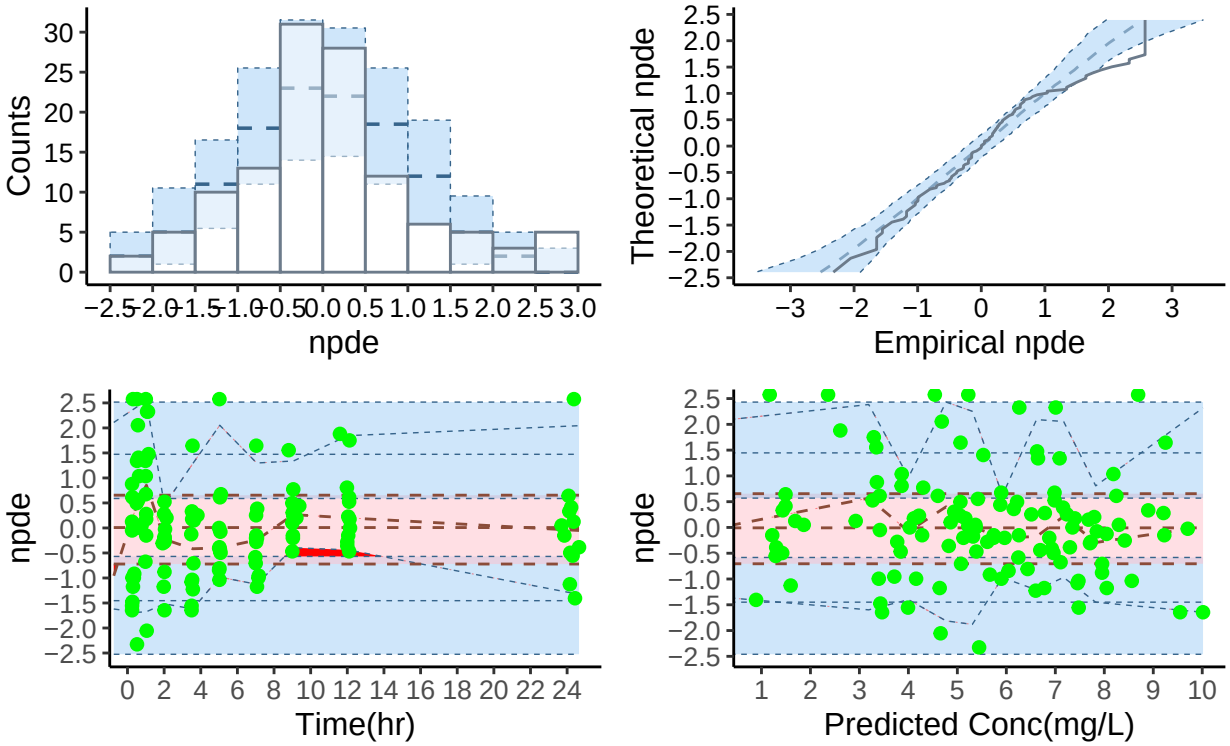

```
## Option which.y= faux not recognised
## Option which.y= faux not recognised
```

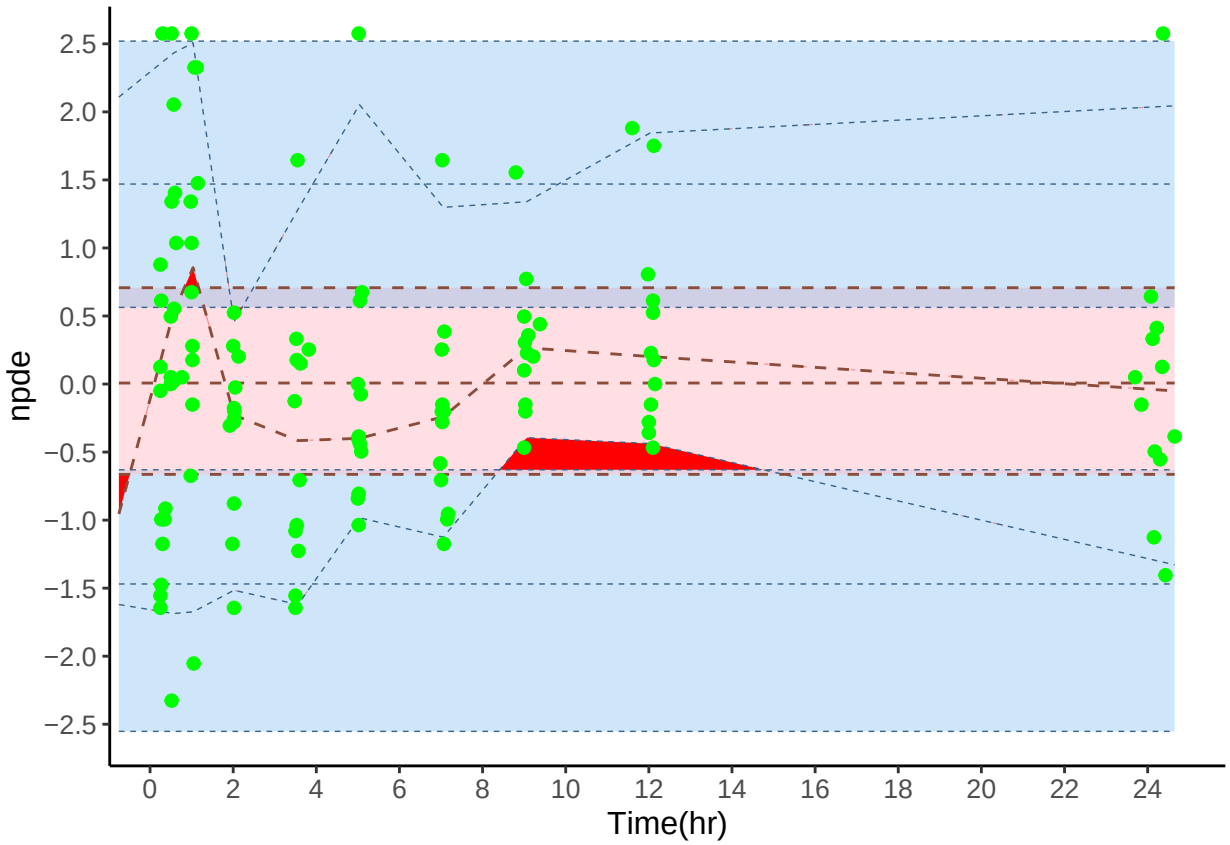

Counts
30
25
20
15
10
5
0
−2.5 −2.0 −1.5 −1.0 −0.5 0.0 0.5 1.0 1.5 2.0 2.5 3.0
npde

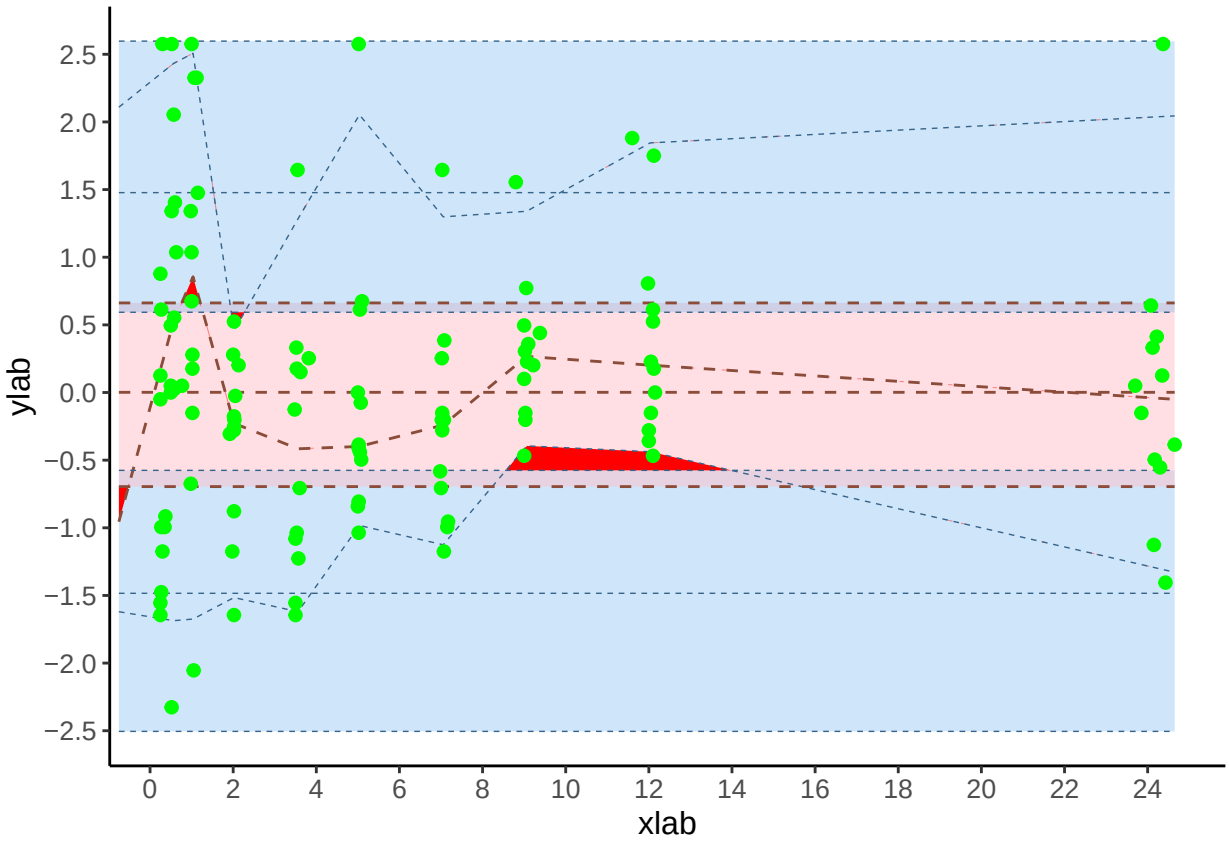

Theoretical npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3 −2 −1 0 1 2 3
Empirical npde


Counts
npde
Theoretical npde
Empirical npde
npde
Time(hr)
npde
Predicted Conc(mg/L)


Counts
npde
Theoretical npde
Empirical npde
npde
Time(hr)
npde
Predicted Conc(mg/L)

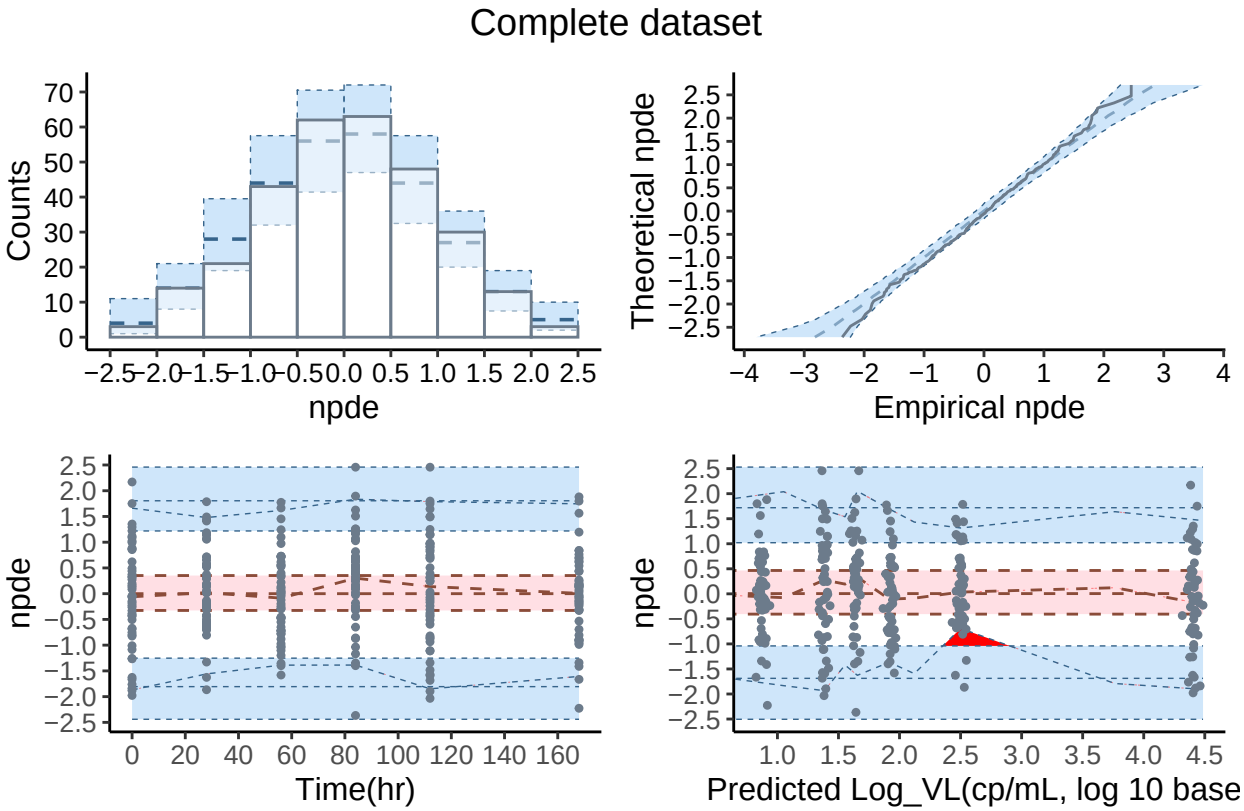

Counts
npde
Theoretical npde
Empirical npde
npde
Time(hr)
npde
Predicted Conc(mg/L)

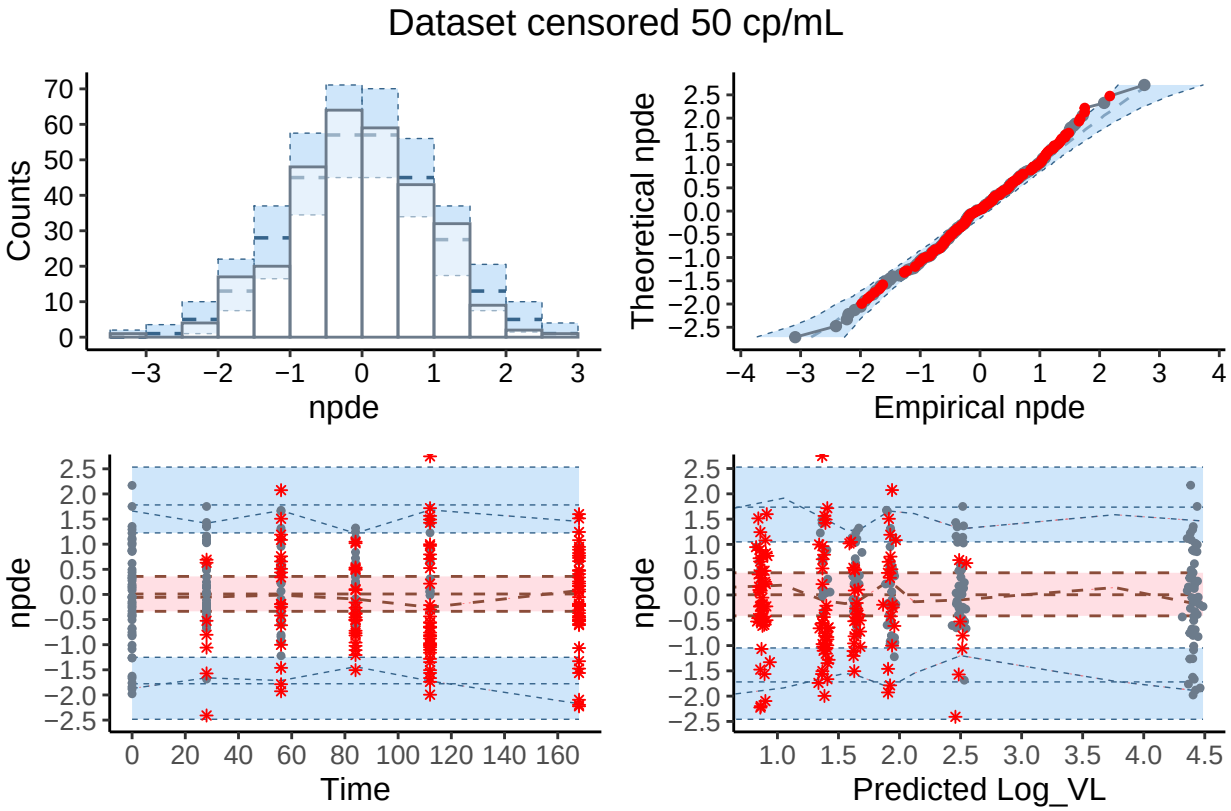

npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0
2
4
6
8
10
12
14
16
18
20
22
24
Time(hr)

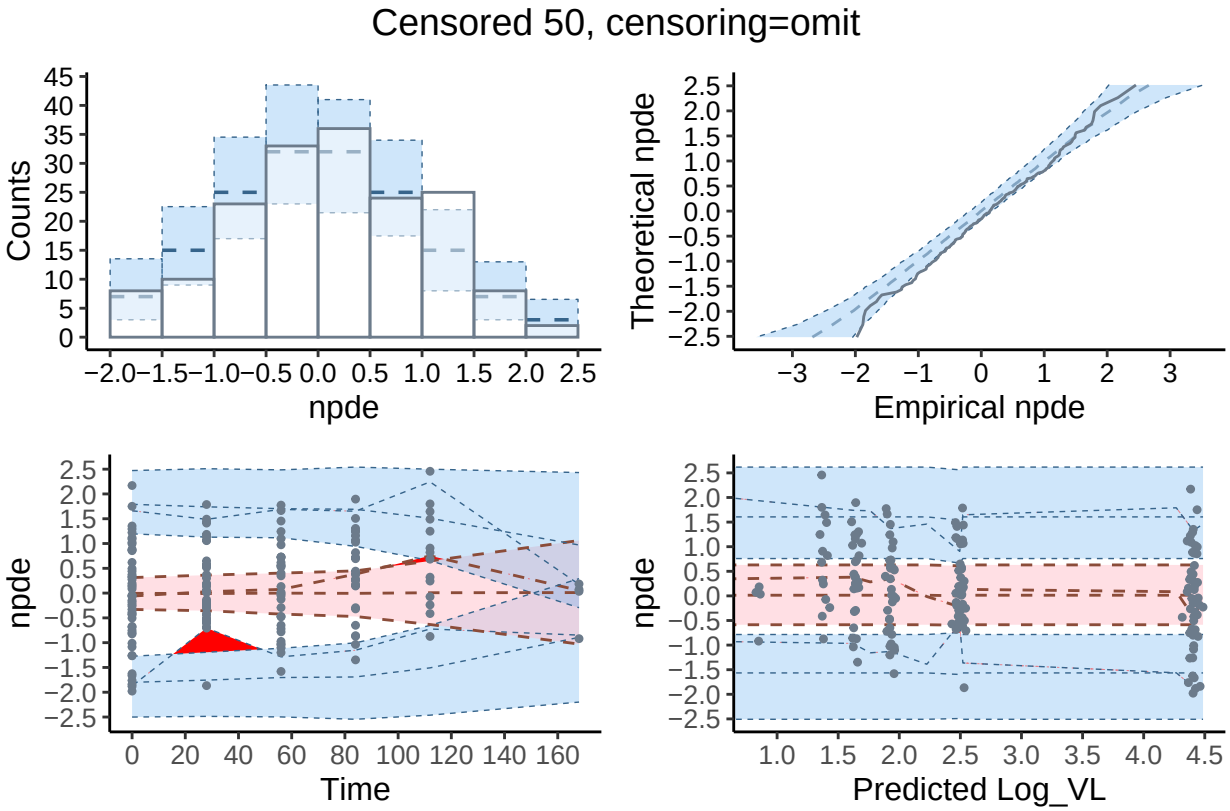


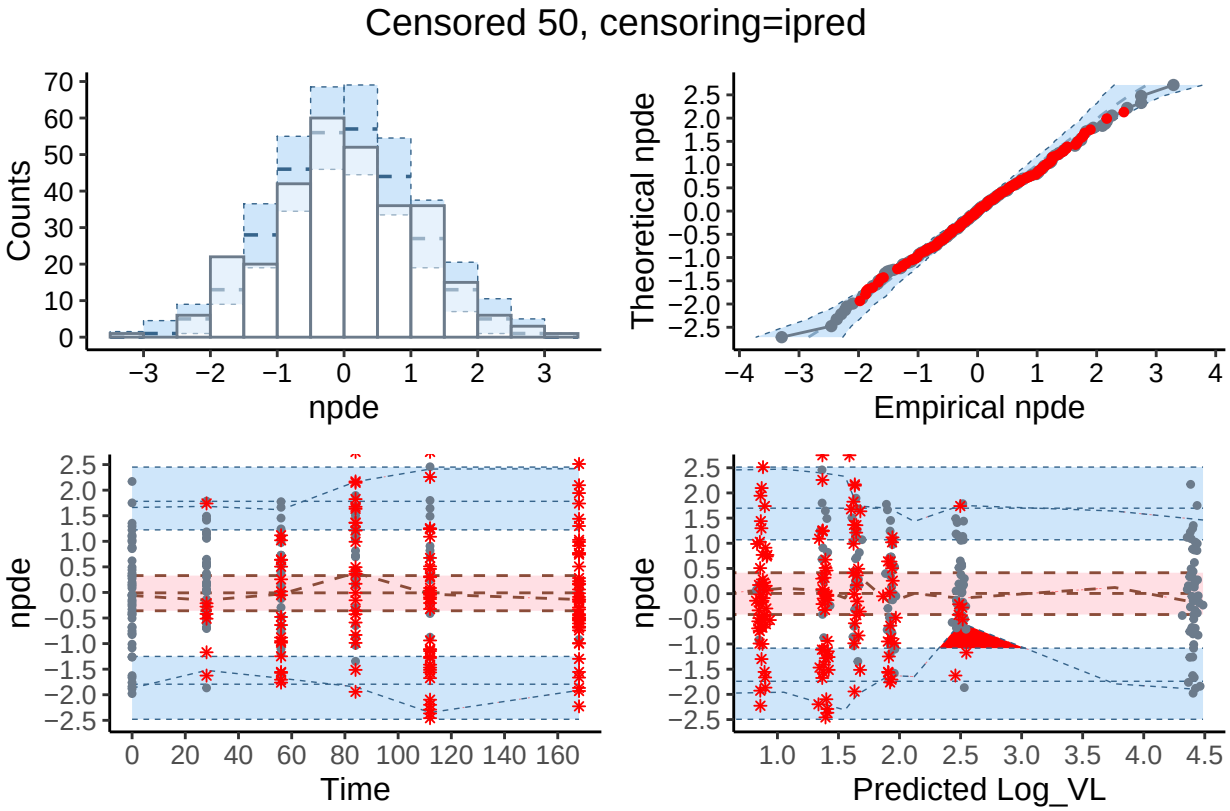
Default plots for viral load data

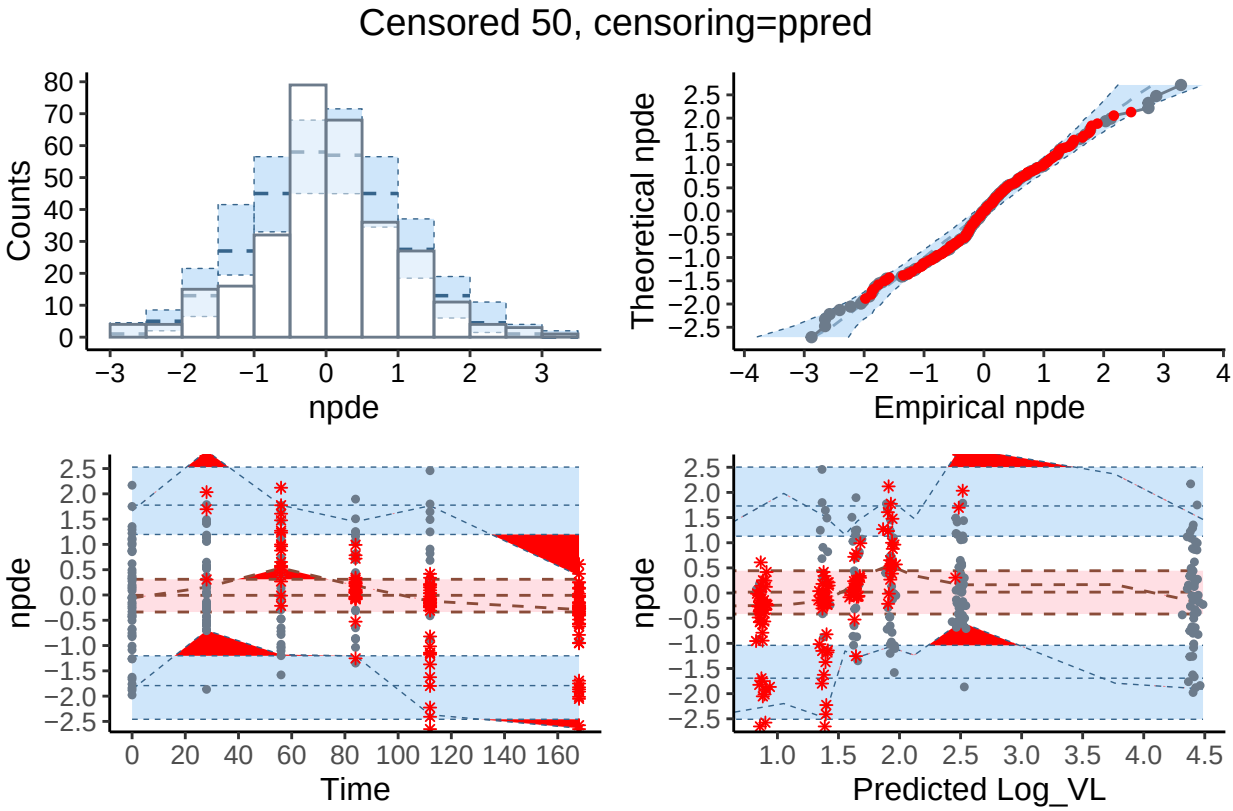
- computation of the PI: seems appropriate in the different cases
- **Problems** (Eco)
  - PI for omit aren't consistent with the VPC later on
    - **documentation** explain how PI are computed in the omit case

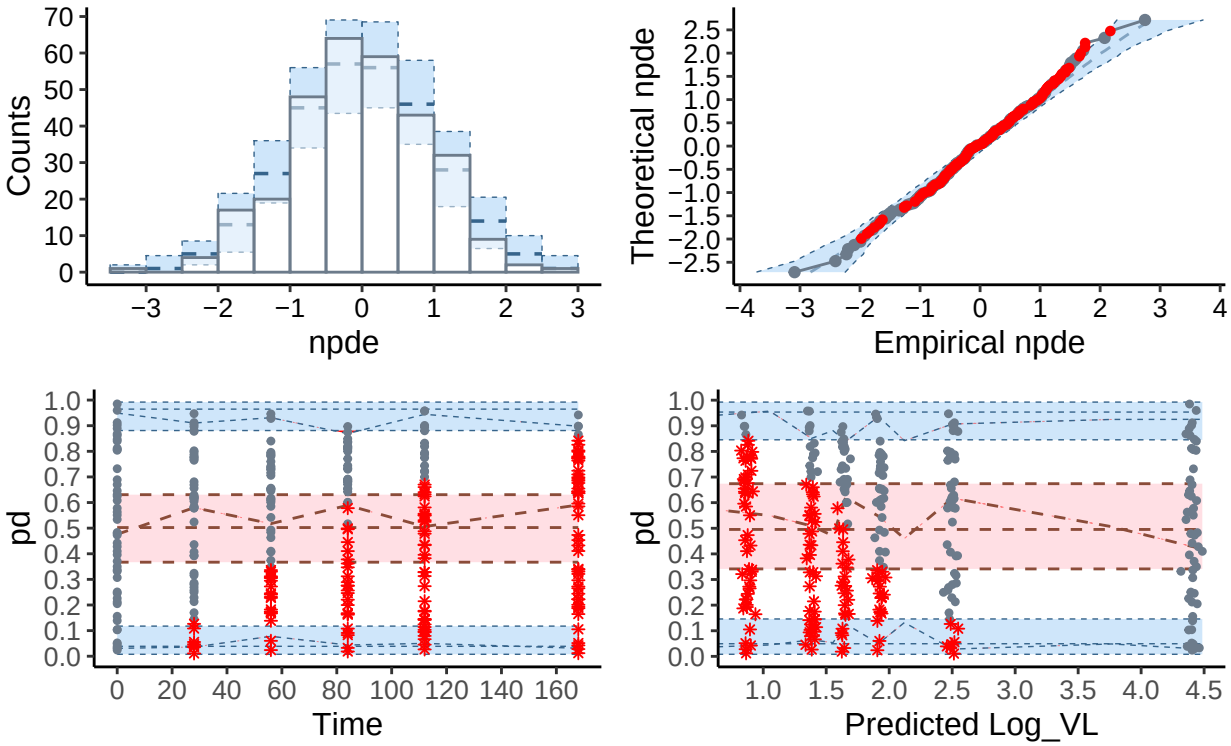

Complete dataset
Counts
npde
Theoretical npde
Empirical npde
npde
Time(hr)
npde
Predicted Log_VL(cp/mL, log 10 base


Dataset censored 50 cp/mL
Counts
npde
Theoretical npde
Empirical npde
npde
Time
npde
Predicted Log_VL


Censored 50, censoring=omit
Counts
npde
Theoretical npde
Empirical npde
npde
Time
npde
Predicted Log_VL


Censored 50, censoring=ipred
Counts
npde
Theoretical npde
Empirical npde
npde
Time
npde
Predicted Log_VL

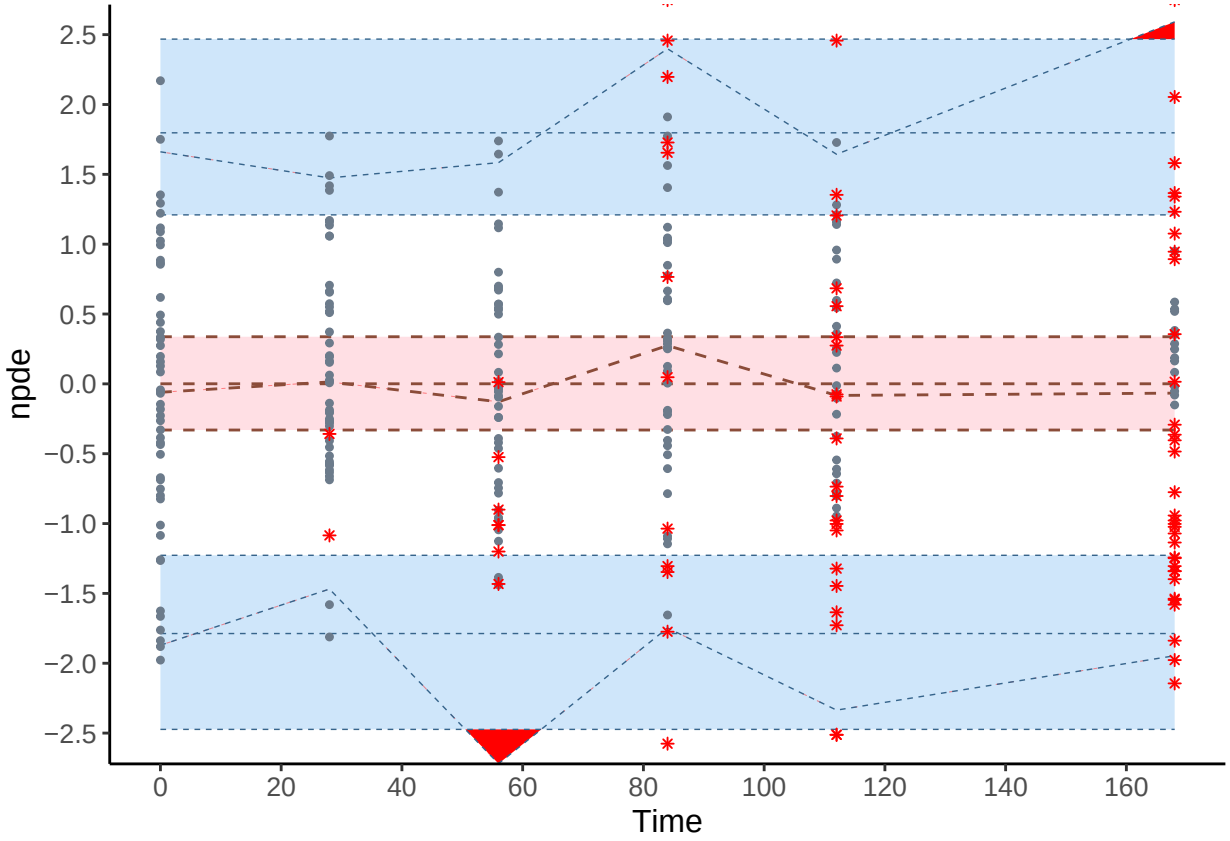

Censored 50, censoring=ppred
Counts
npde
Theoretical npde
Empirical npde
npde
Time
npde
Predicted Log_VL

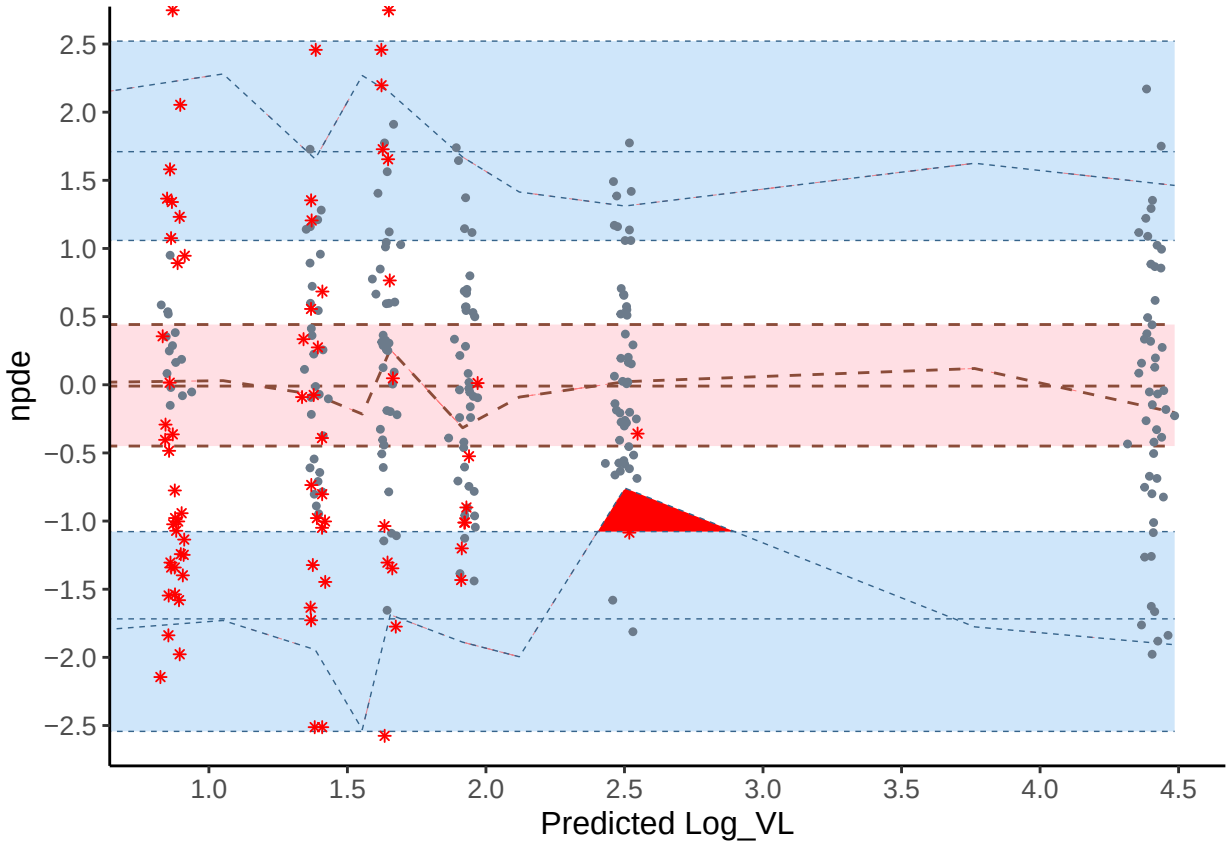


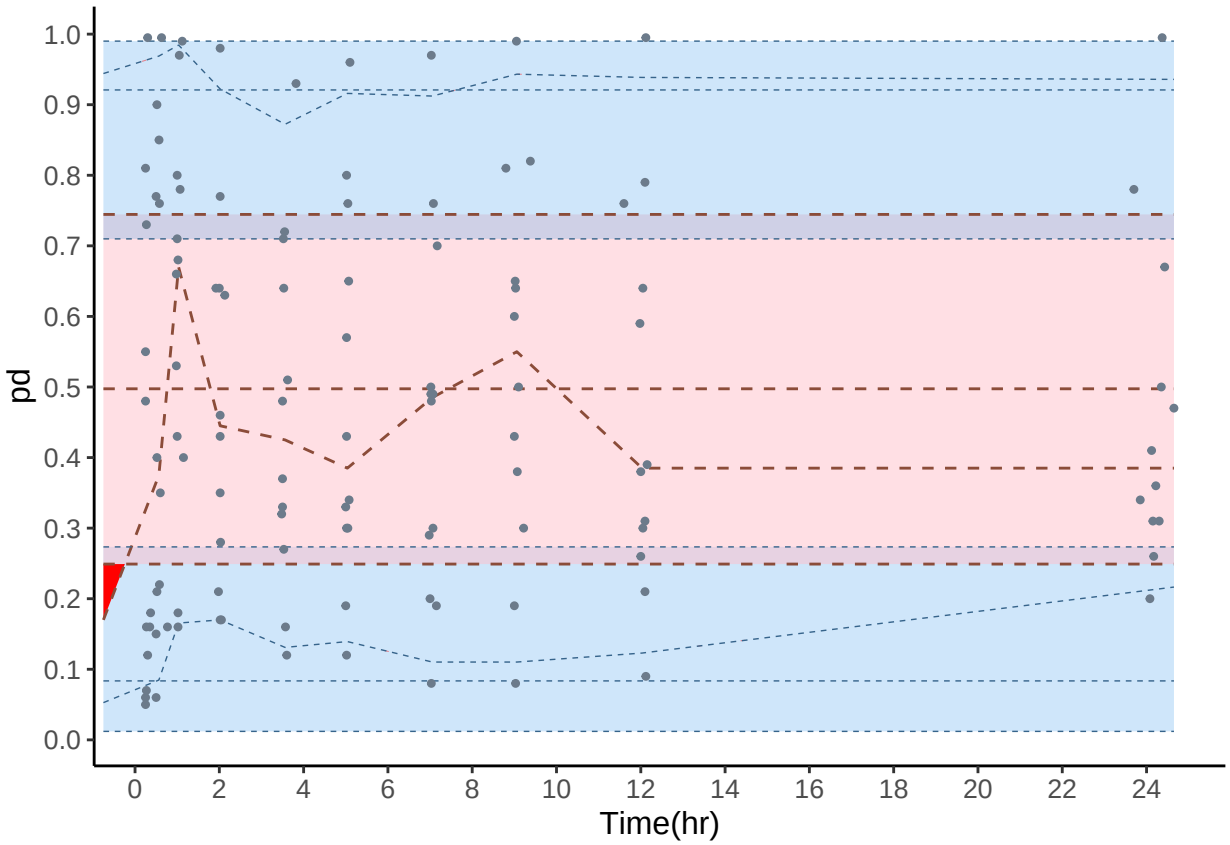
Plots with default options

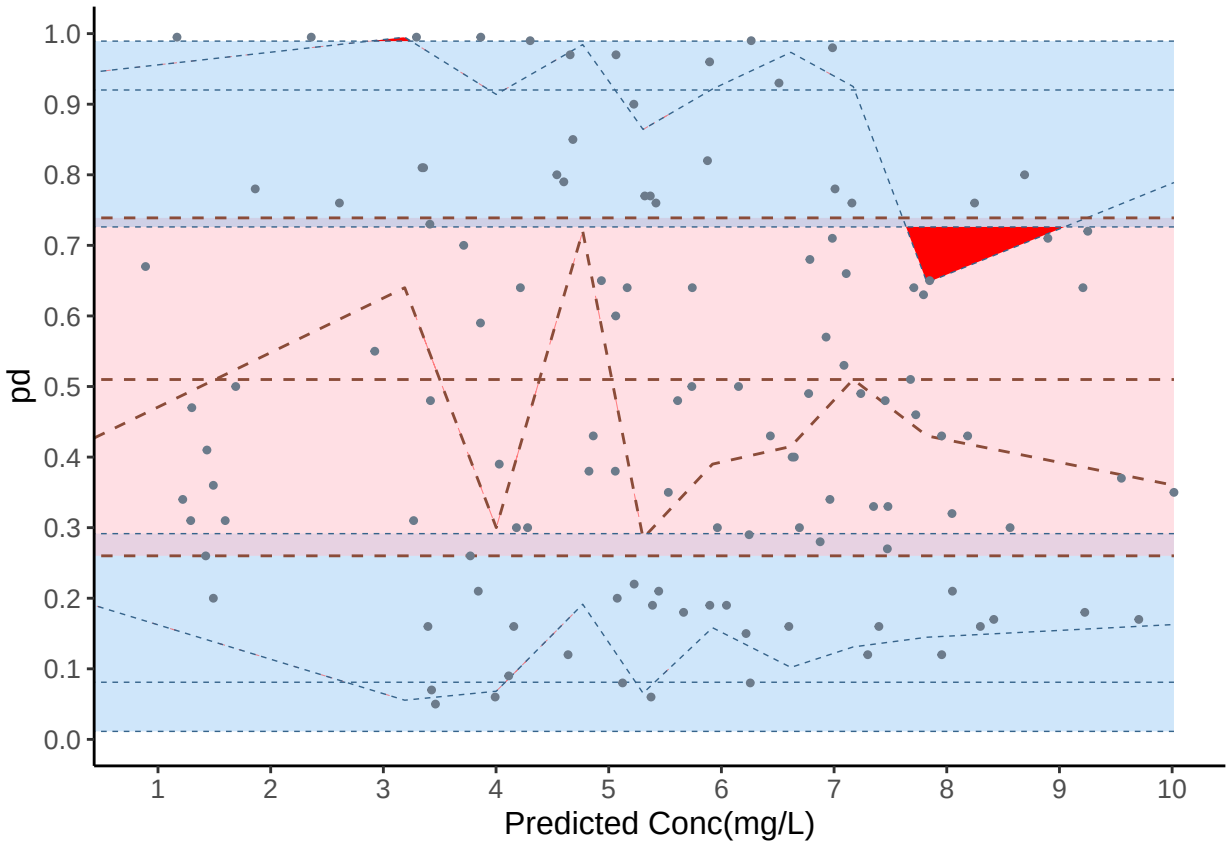
Scatterplots

- **Problems** (à résoudre Romain)
  - **solved** ylab 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 défault (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

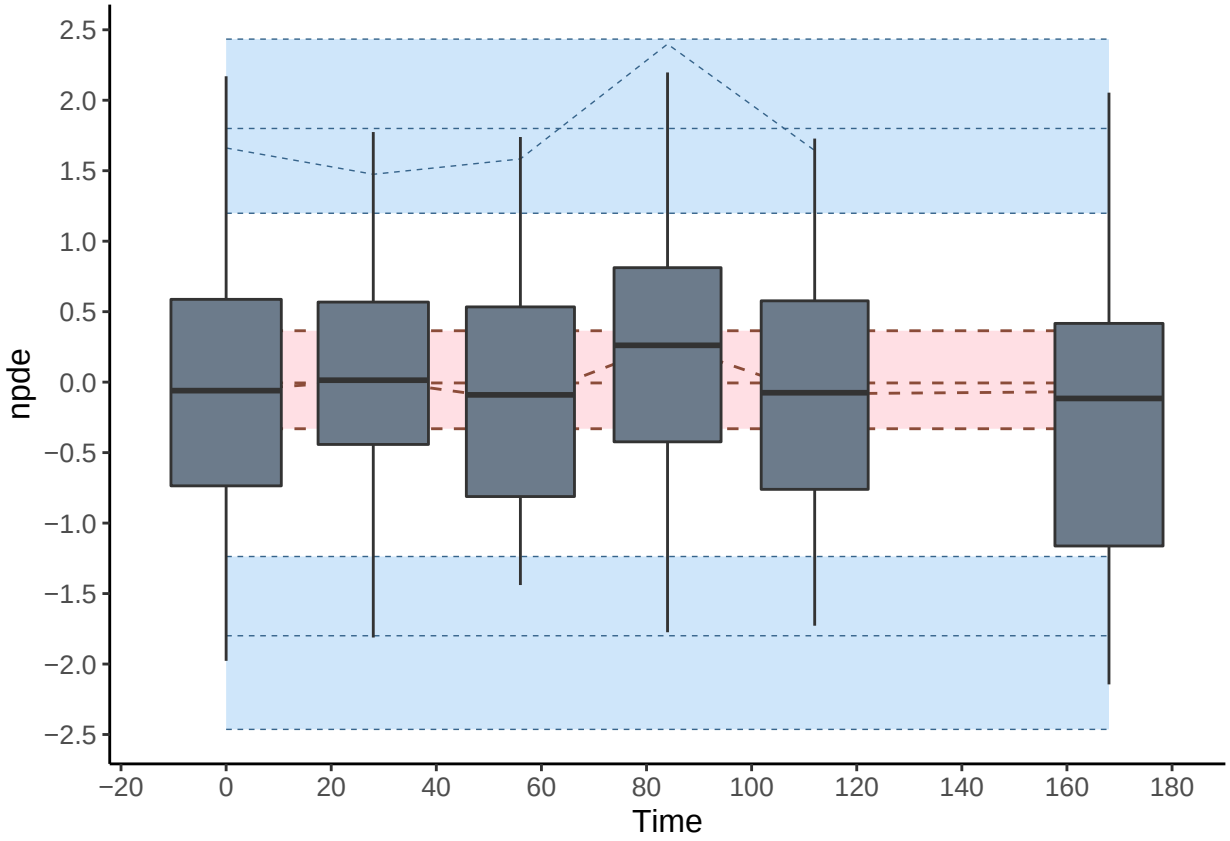

npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0
20
40
60
80
100
120
140
160
Time


npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
Predicted Log_VL


pd
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2
4
6
8
10
12
14
16
18
20
22
24
Time(hr)

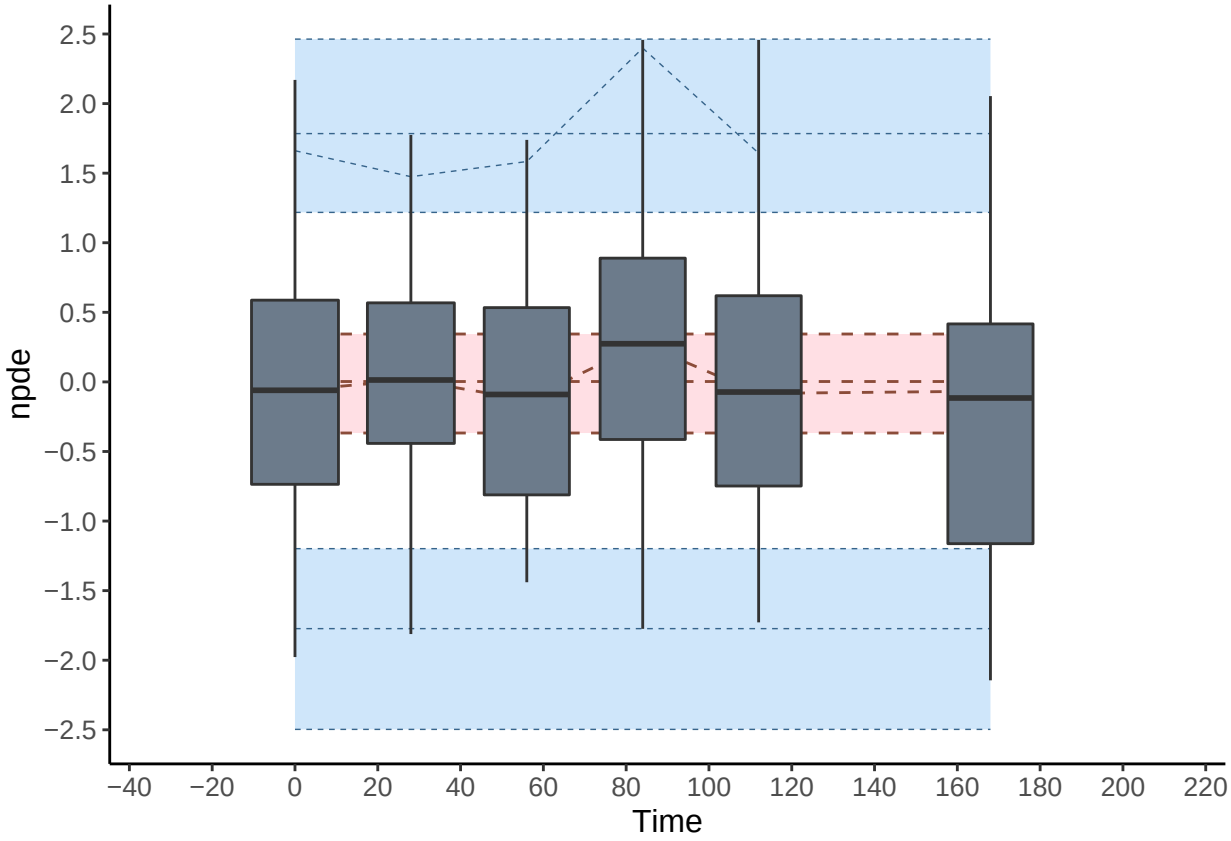


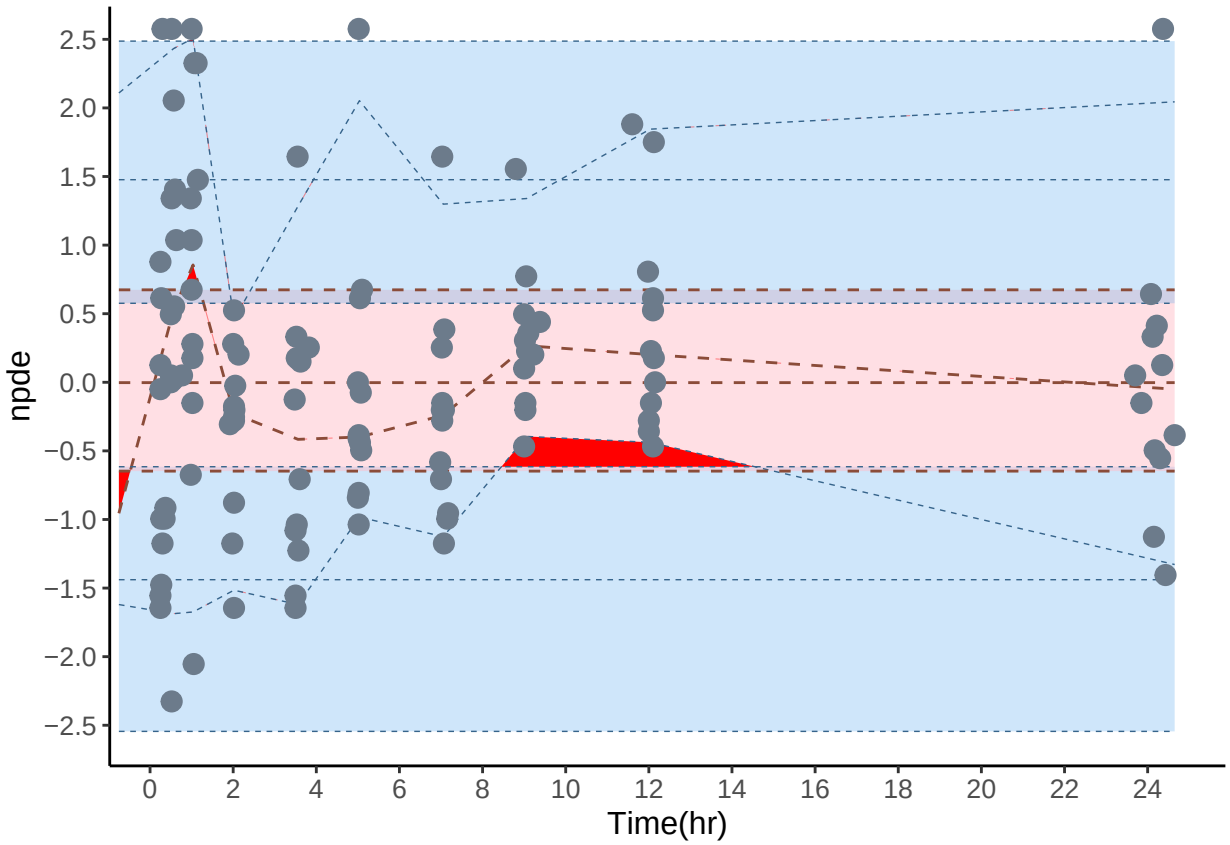

```
## Warning: Removed 10 rows containing non-finite values (stat_boxplot).
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

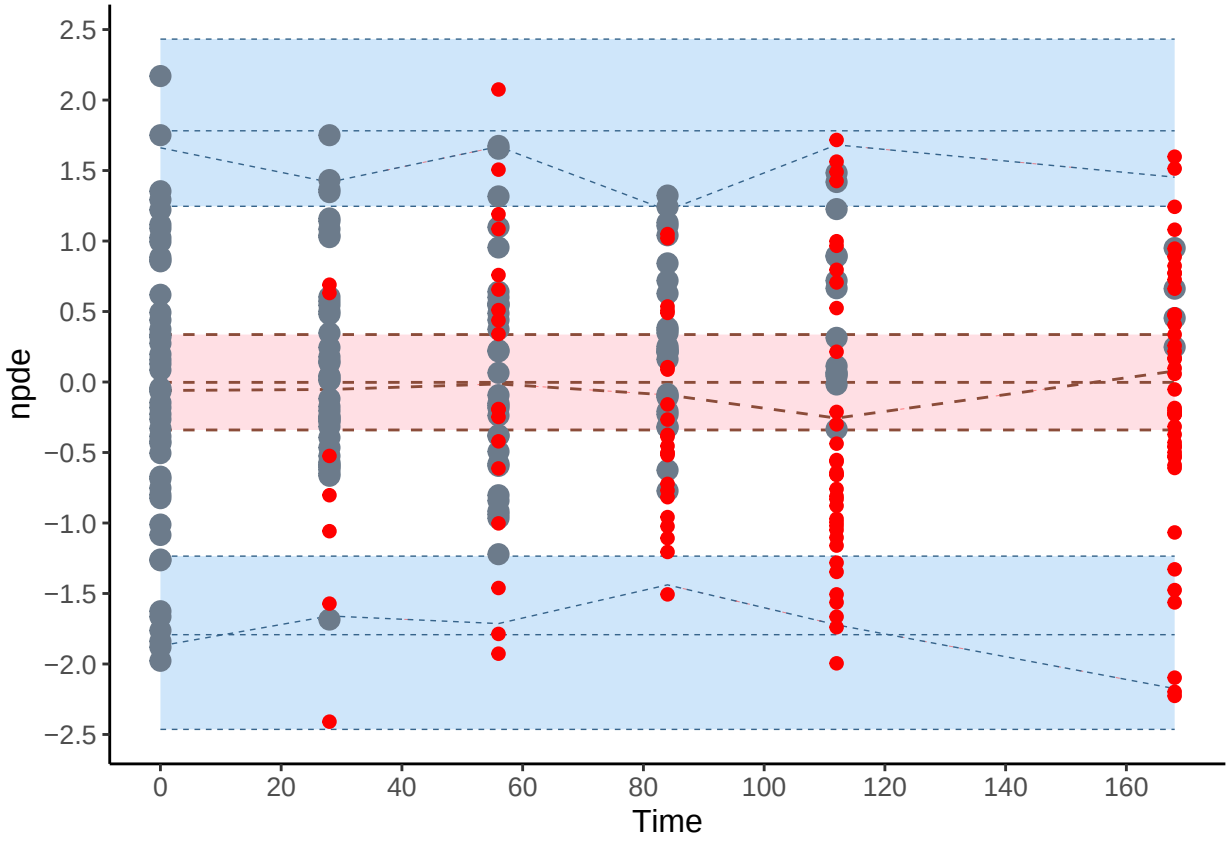



```
## Warning: Removed 8 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

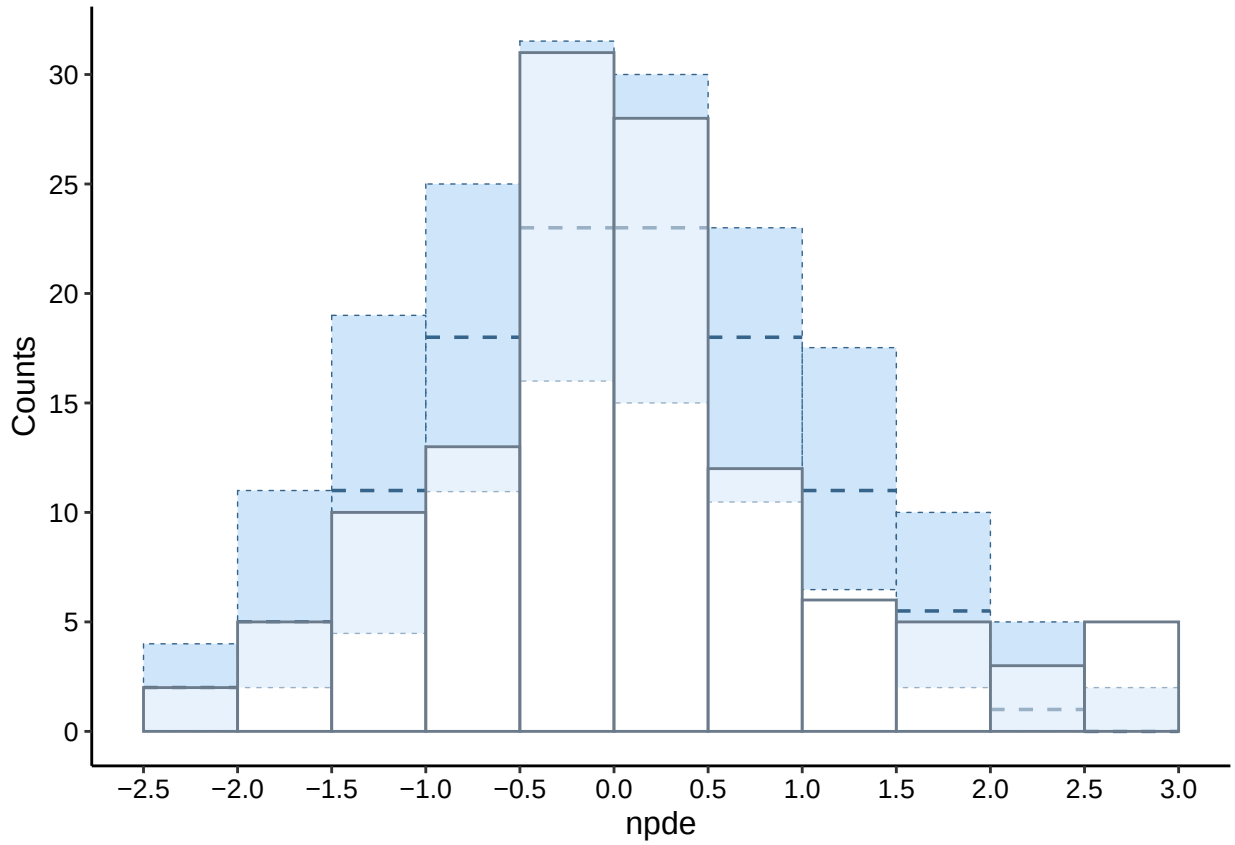

npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−40
−20
0
20
40
60
80
100
120
140
160
180
200
220
Time

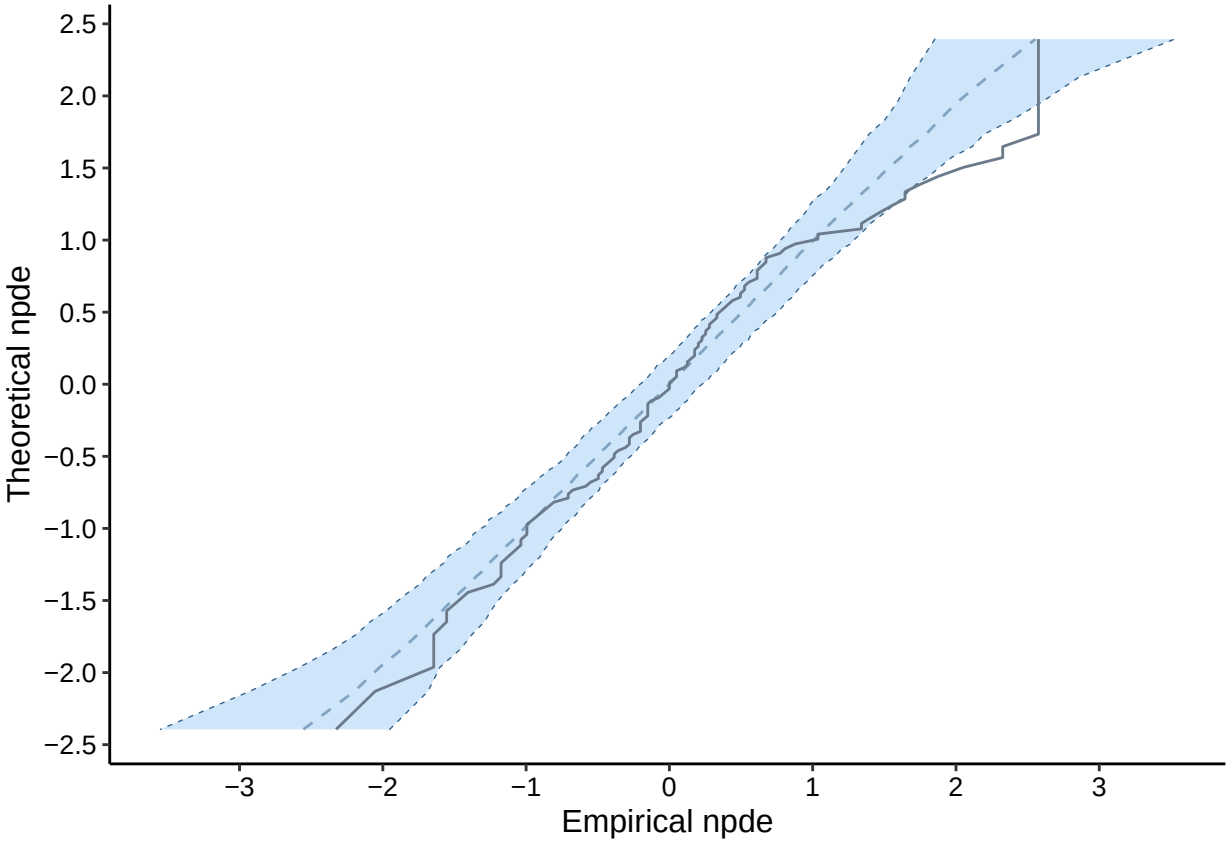

npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0
2
4
6
8
10
12
14
16
18
20
22
24
Time(hr)

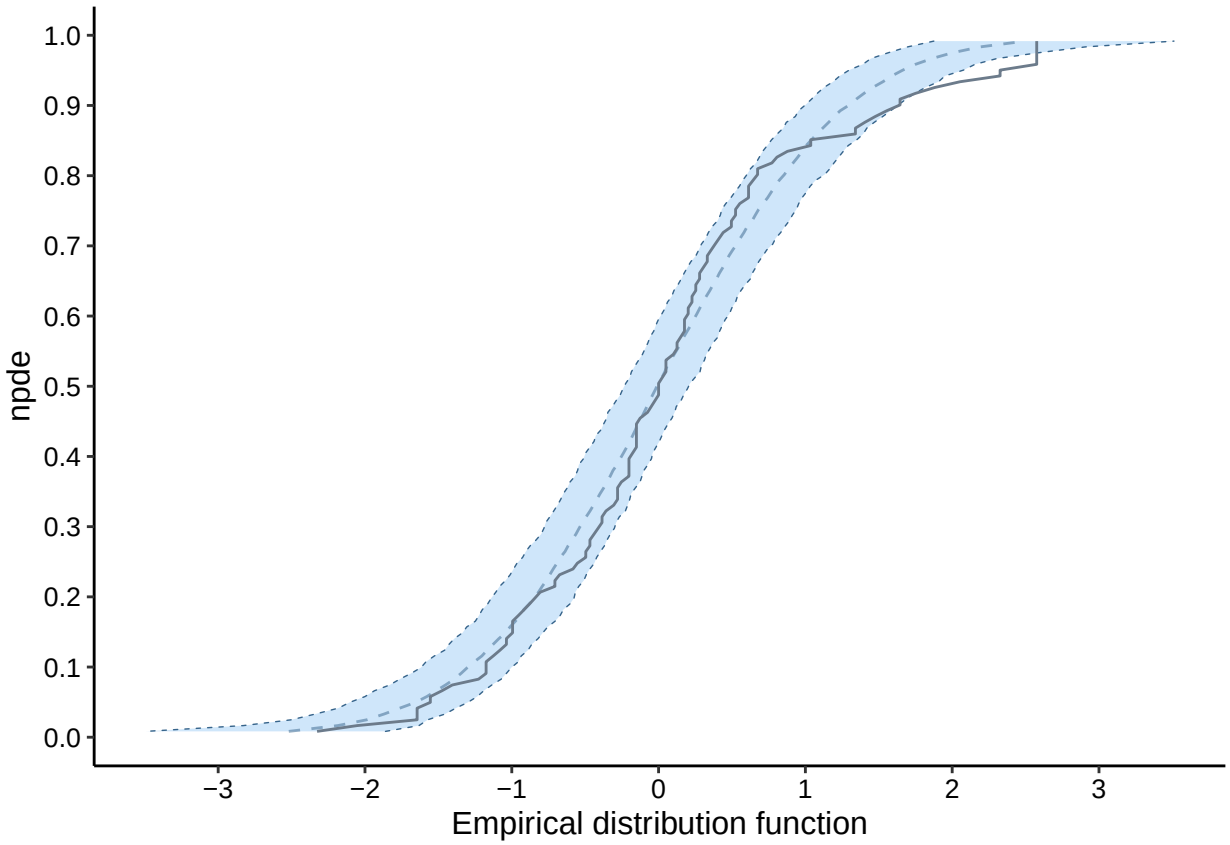


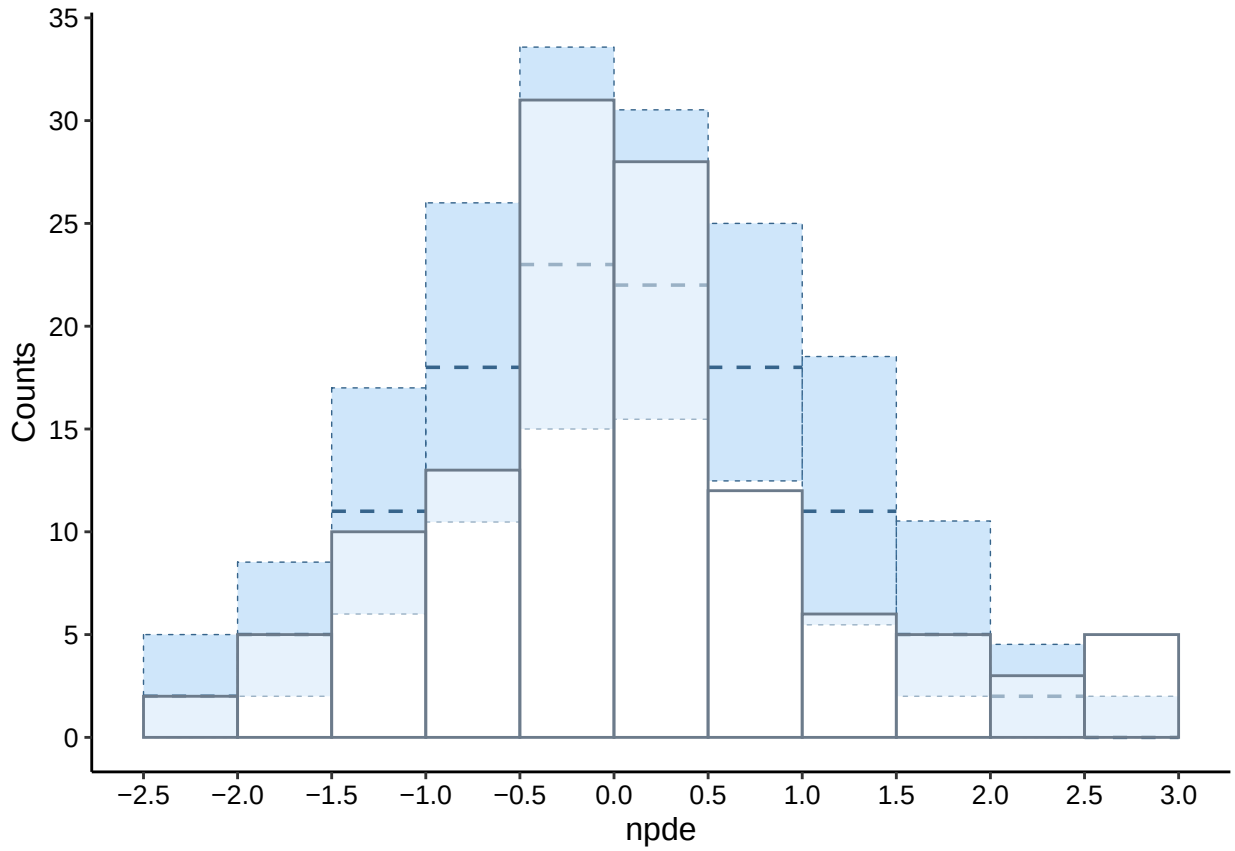
Distribution plots

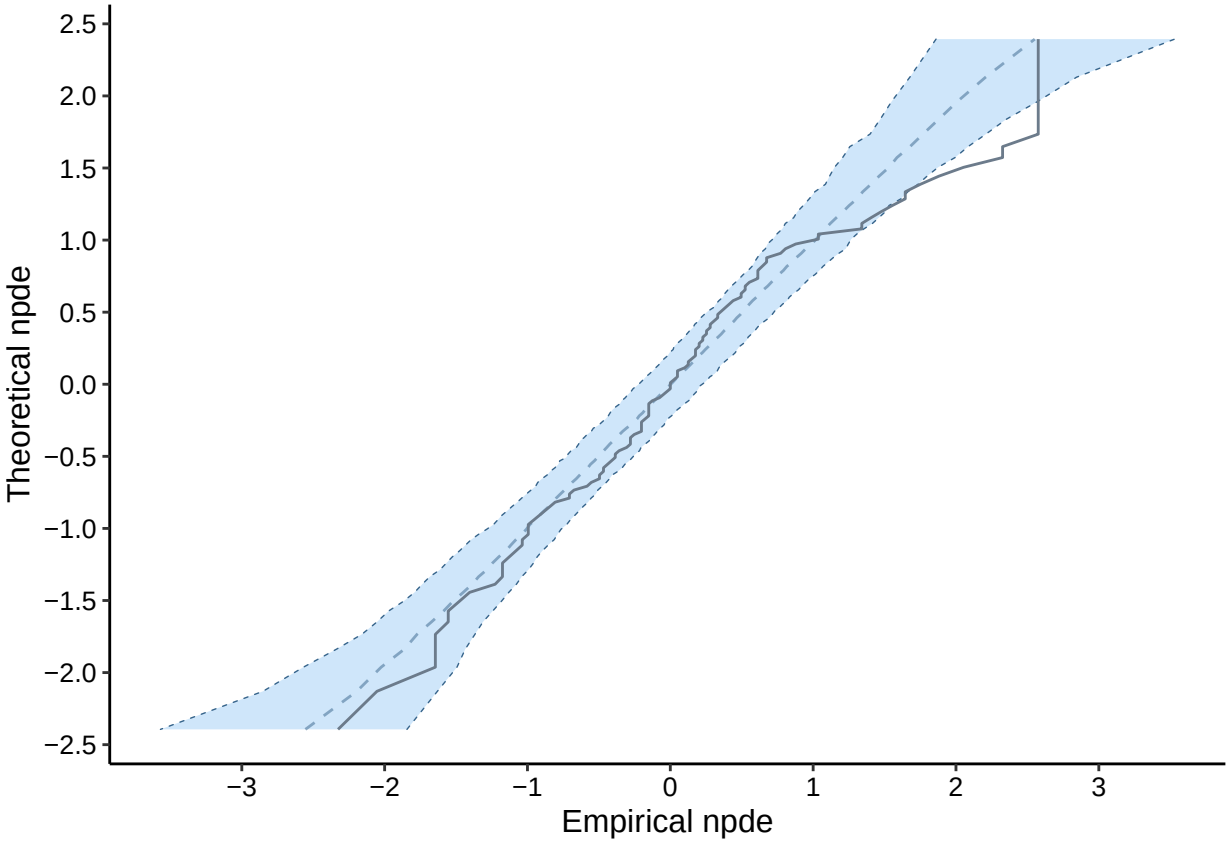
- **Problems** (à résoudre Romain)
  - **solved** pb with **size** (comme plus haut, size ne passe pas, size.pobs/size.pcens passent)
- **Problems** (à résoudre Eco)
  - **axe title** for ecdf plot (should be: 'npde' on X-axis (/pd/npd) and 'Empirical distribution function' on Y-axis) [done, check it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

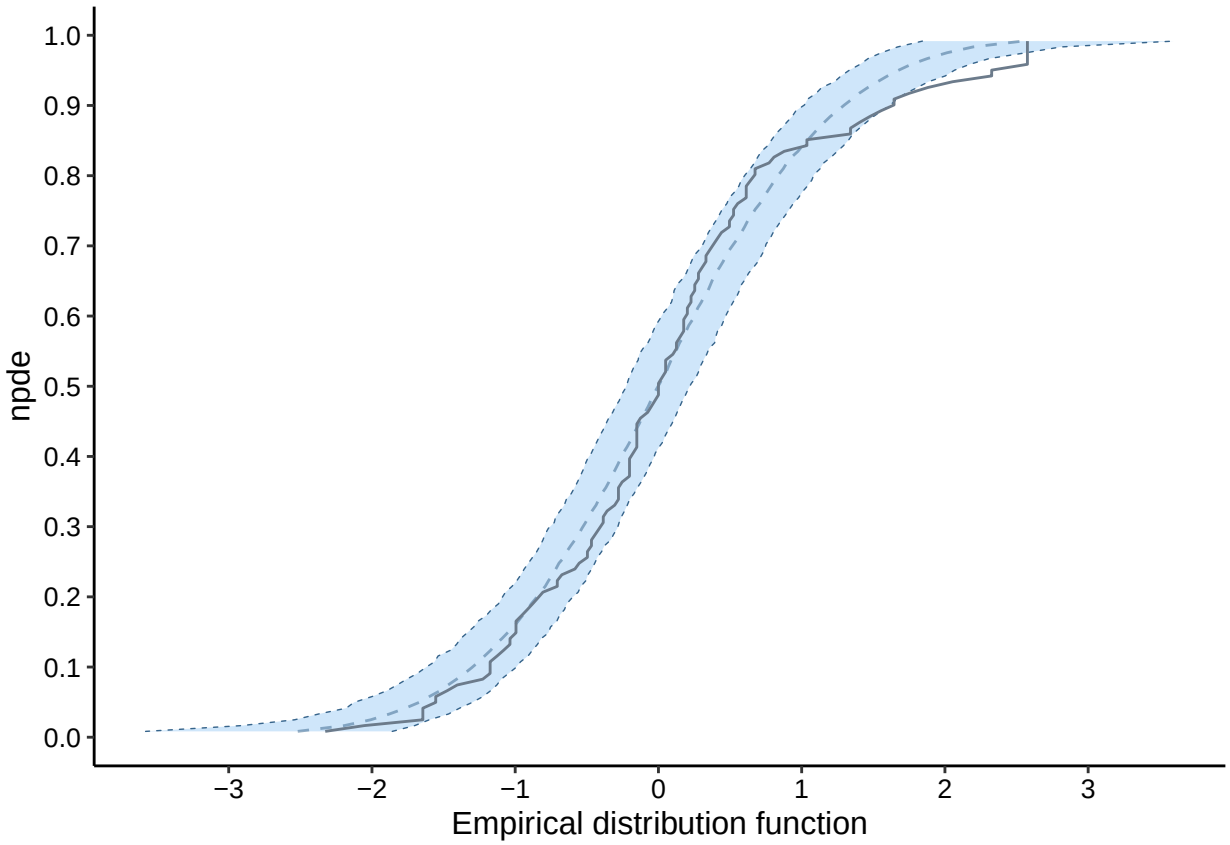

Counts
30
25
20
15
10
5
0
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
npde

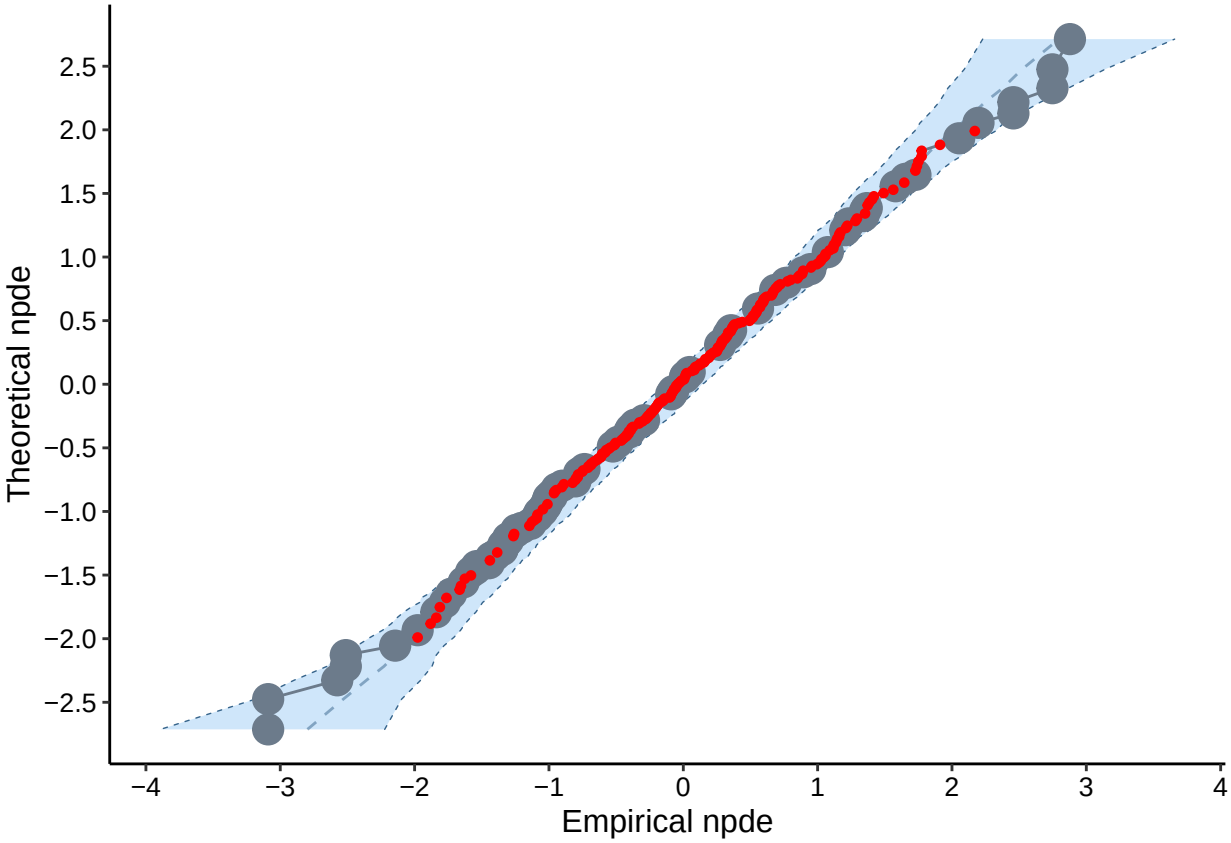

Theoretical npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3
−2
−1
0
1
2
3
Empirical npde

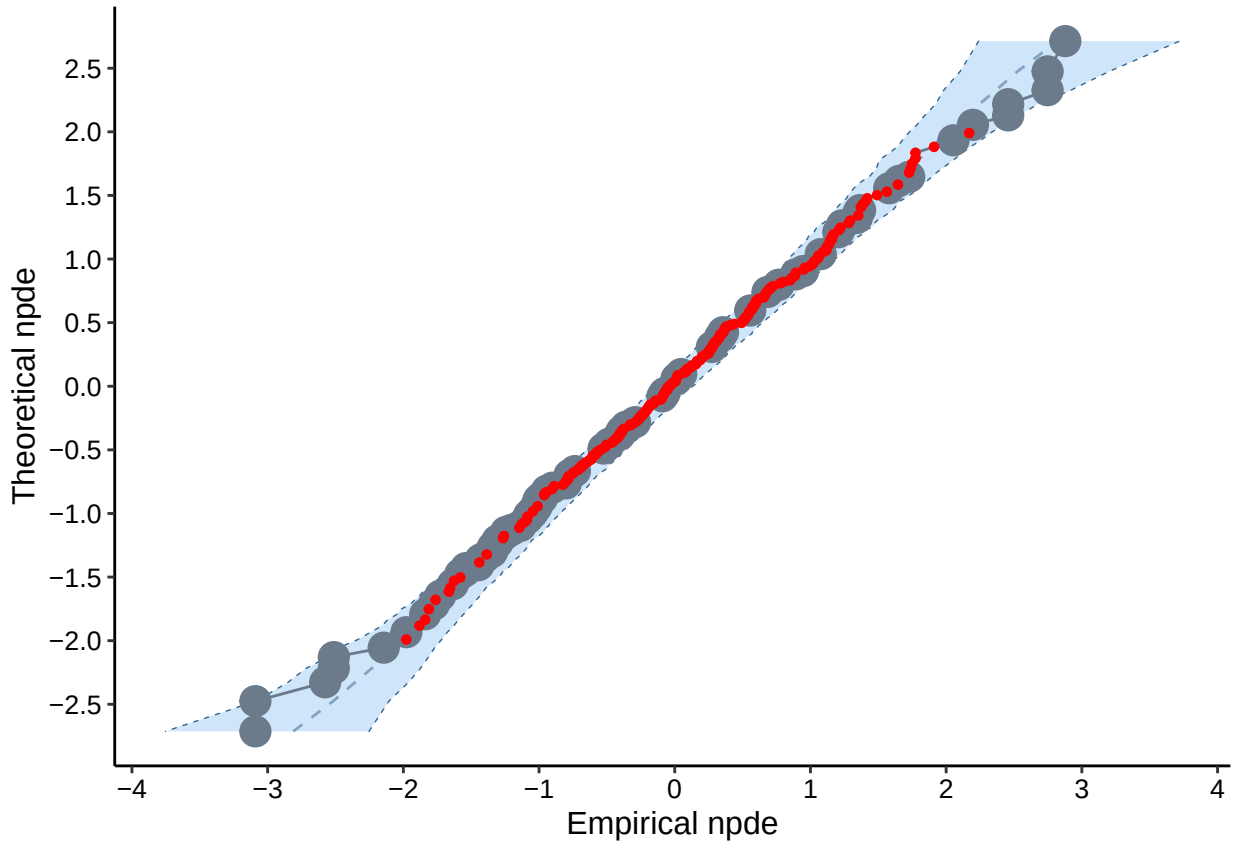

1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
npde
−3
−2
−1
0
1
2
3
Empirical distribution function


Counts
35
30
25
20
15
10
5
0
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
npde


Theoretical npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3
−2
−1
0
1
2
3
Empirical npde


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
npde
−3
−2
−1
0
1
2
3
Empirical distribution function

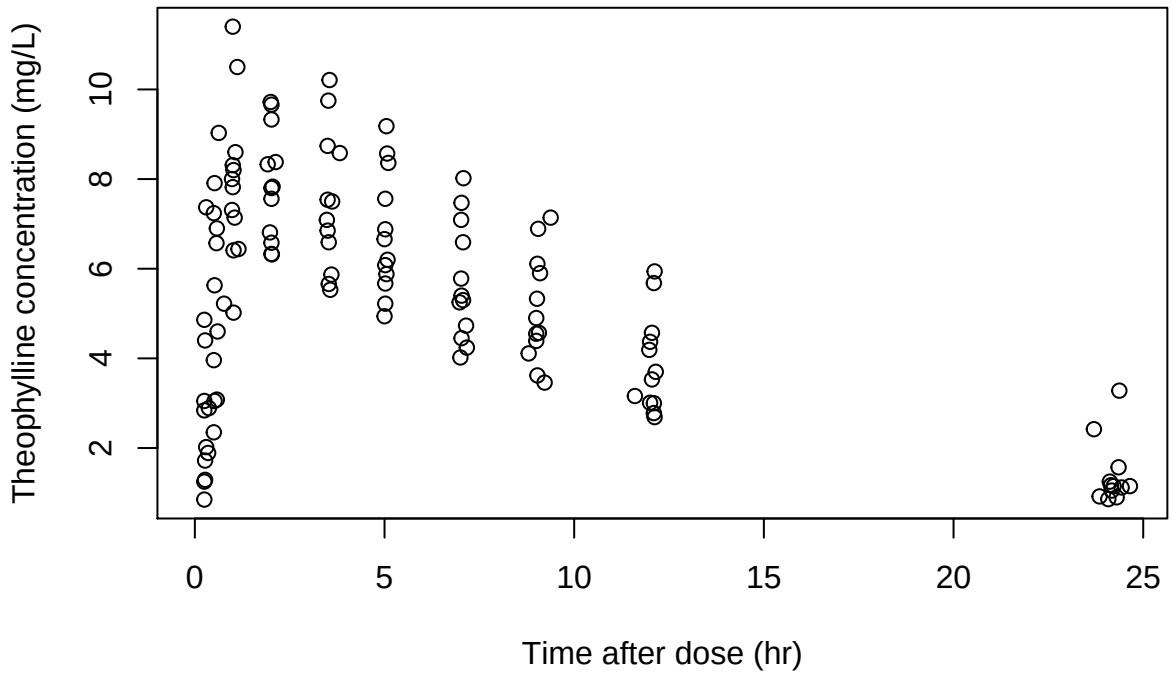

Theoretical npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−4
−3
−2
−1
0
1
2
3
4
Empirical npde



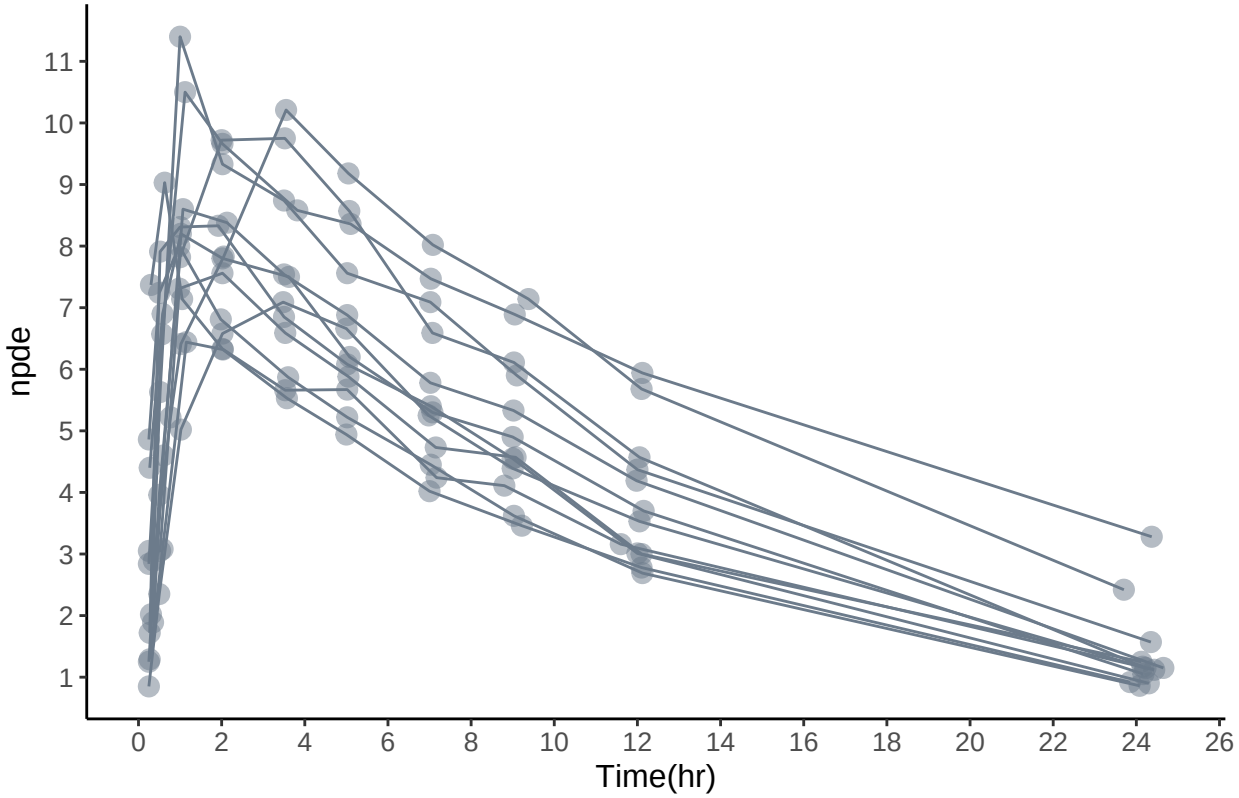
Data plots

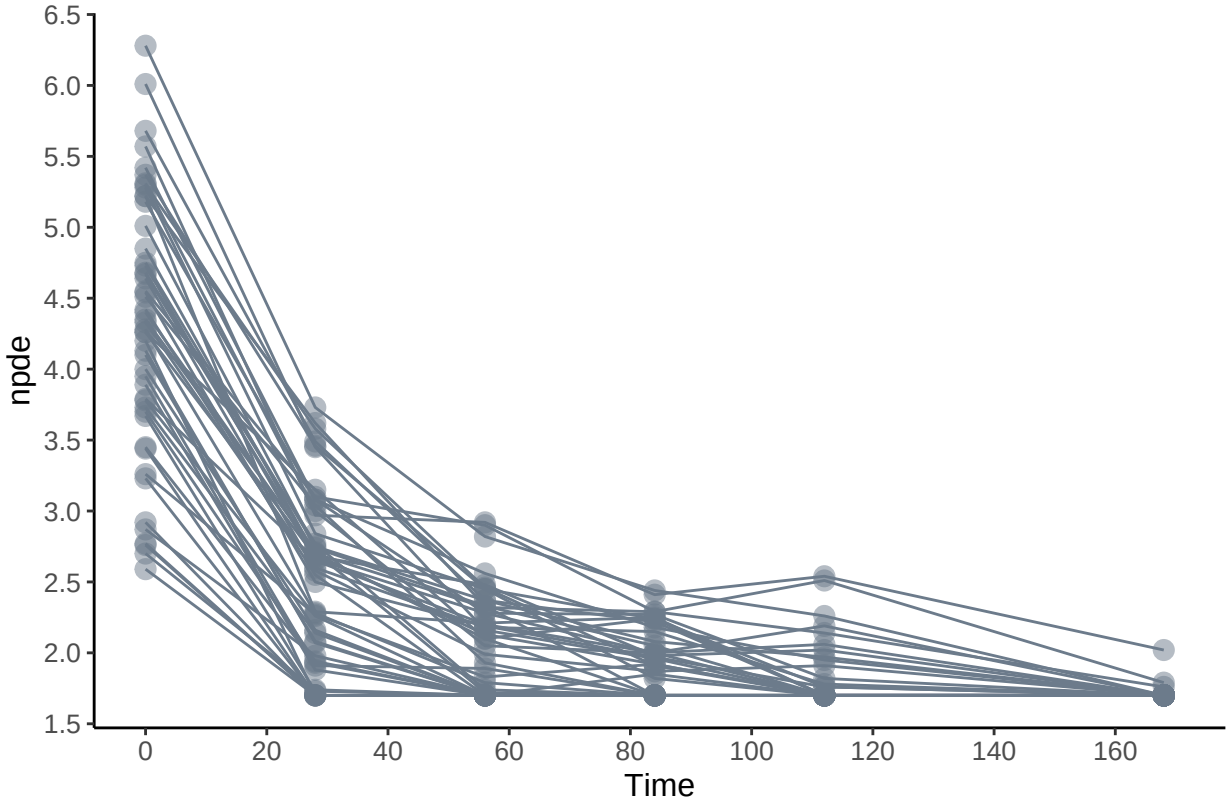
- **Problems** (à résoudre Romain)
- **Note** pour cette version ça fait beaucoup à modifier, à commencer par plotNpde-methods puis npde.plot.data où lee slota data est à modifier. Il faudrait distinguer le plot par défaut et *plot(yivr50 @data)* sans arguments supplémentaires.
  - plot d'un **objet 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]
  - **pour cette version trop de chose à modifier pour ça**
  - 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 ?)

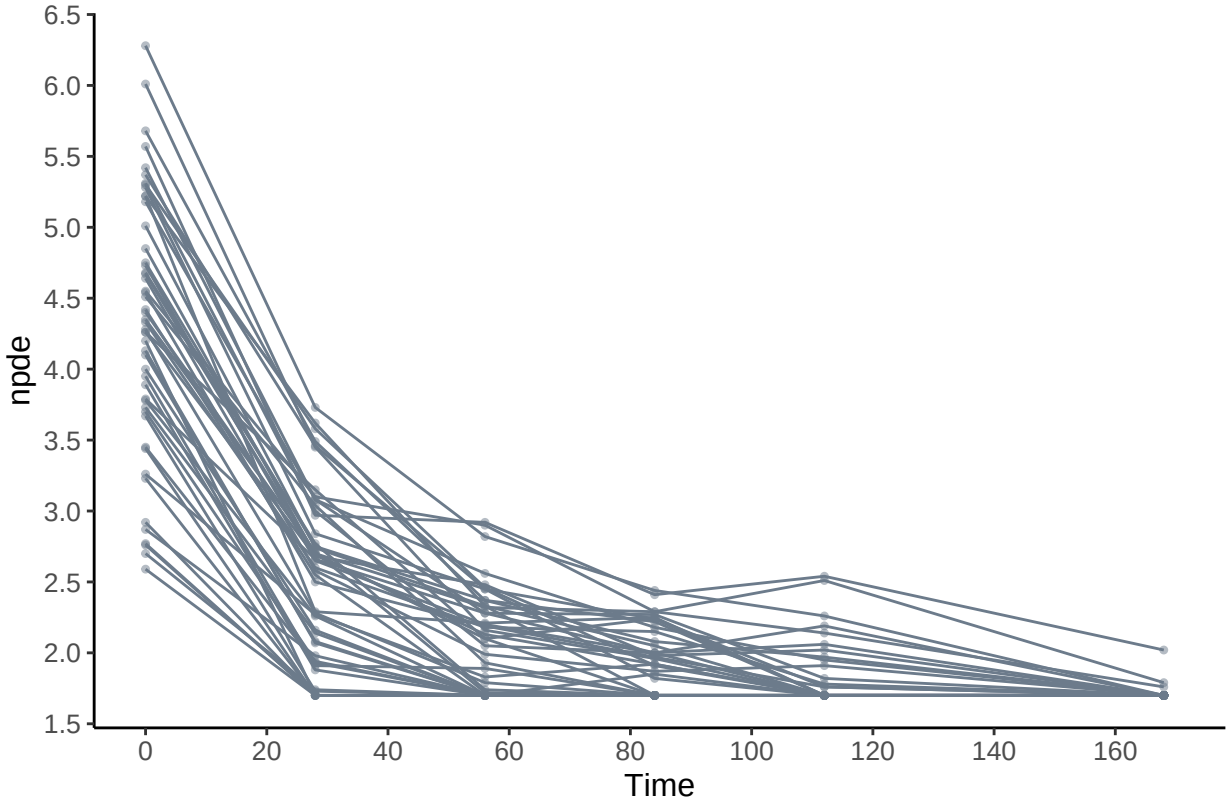
    - size doesn't change size.pobs and size.lobs as it should
    - size.lobs doesn't work
- **solved axis titles**
    - missing for theofit, why ? (present for yvir50, so odd)
- **solved line.loq** doesn't work (no line appears at the LOQ)

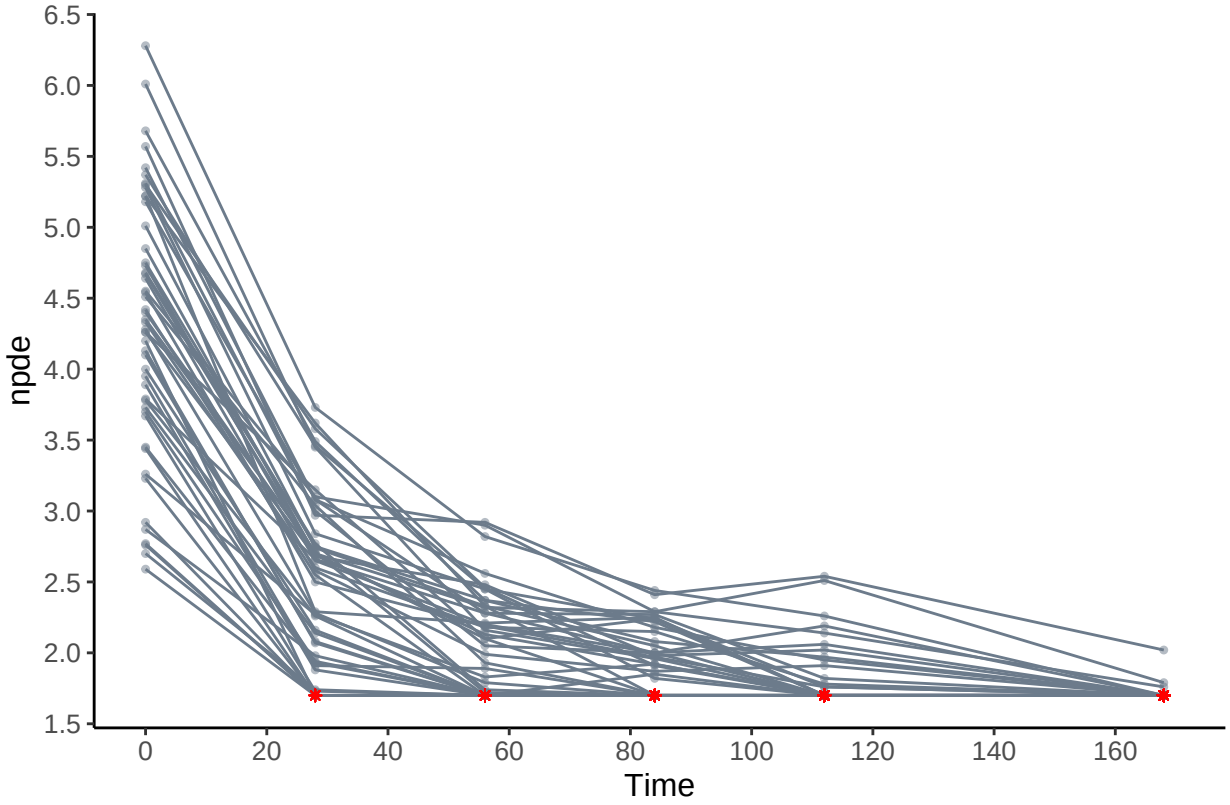


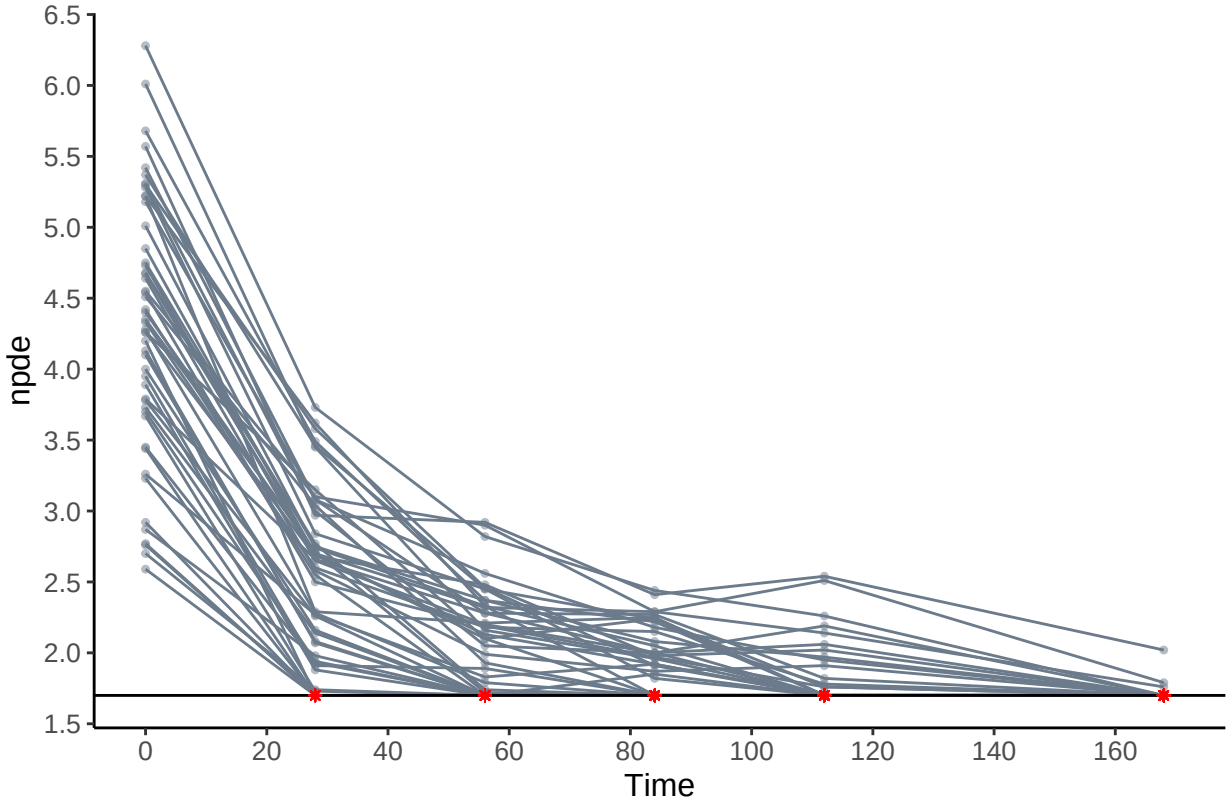

```
## Failed test, no output
## Warning: Removed 12 rows containing missing values (geom_point).
## Warning: Removed 12 row(s) containing missing values (geom_path).
```

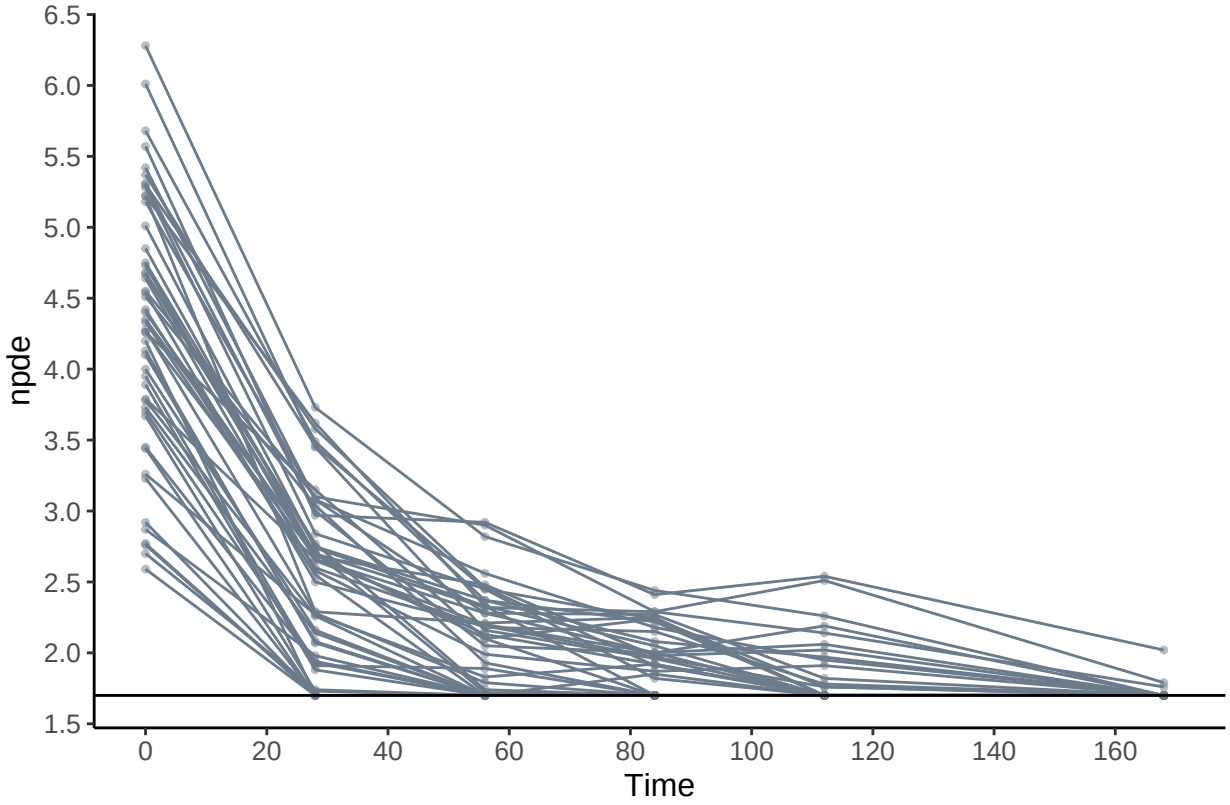

npde
11
10
9
8
7
6
5
4
3
2
1
0
2
4
6
8
10
12
14
16
18
20
22
24
26
Time(hr)

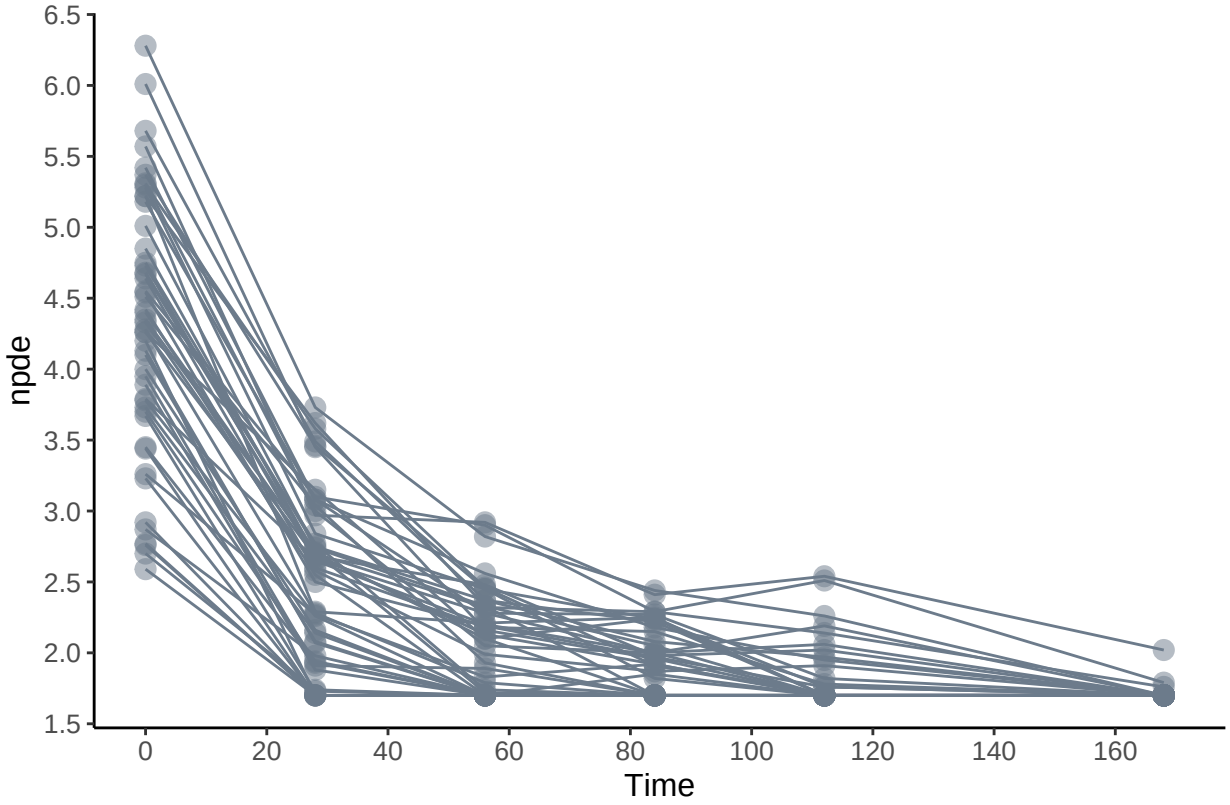

6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
npde
0
20
40
60
80
100
120
140
160
Time

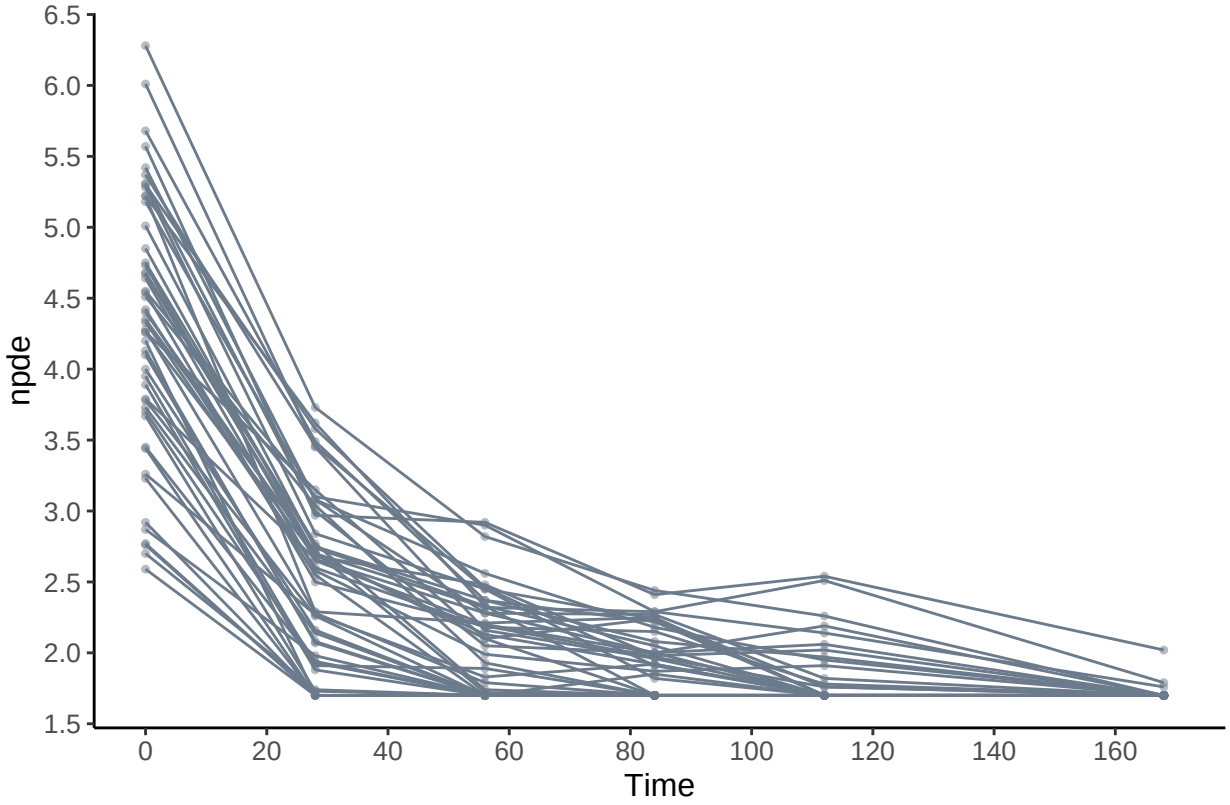

6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
npde
0
20
40
60
80
100
120
140
160
Time

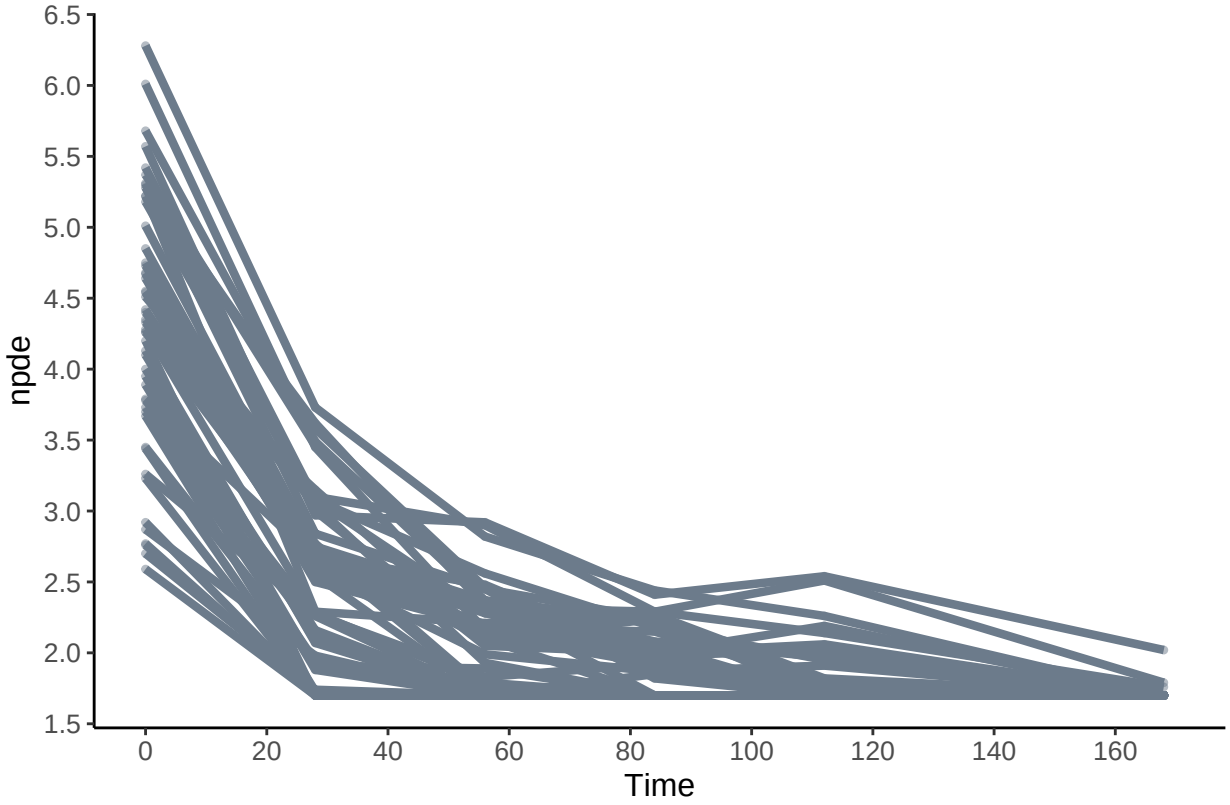

6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
npde
0
20
40
60
80
100
120
140
160
Time


6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
npde
0
20
40
60
80
100
120
140
160
Time


6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
npde
0
20
40
60
80
100
120
140
160
Time

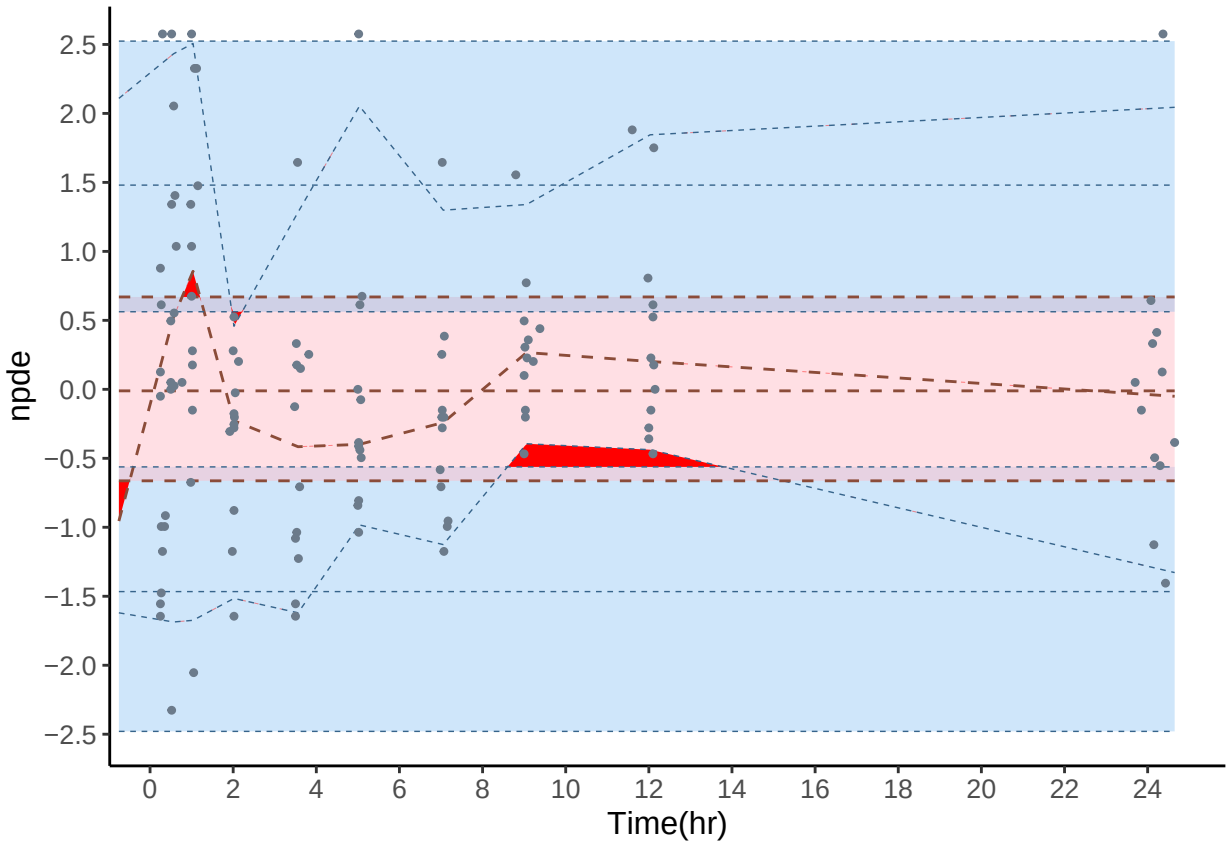

6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
npde
0
20
40
60
80
100
120
140
160
Time

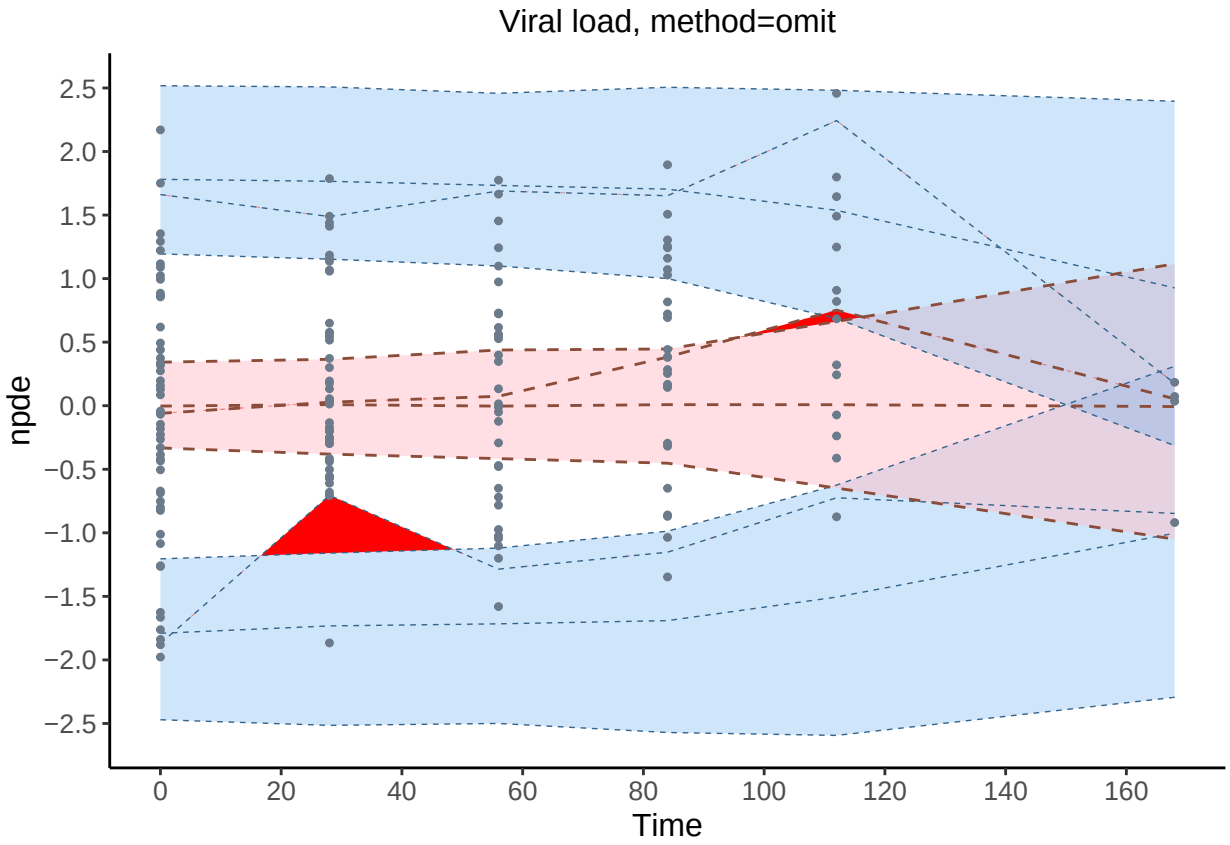

6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
npde
0
20
40
60
80
100
120
140
160
Time

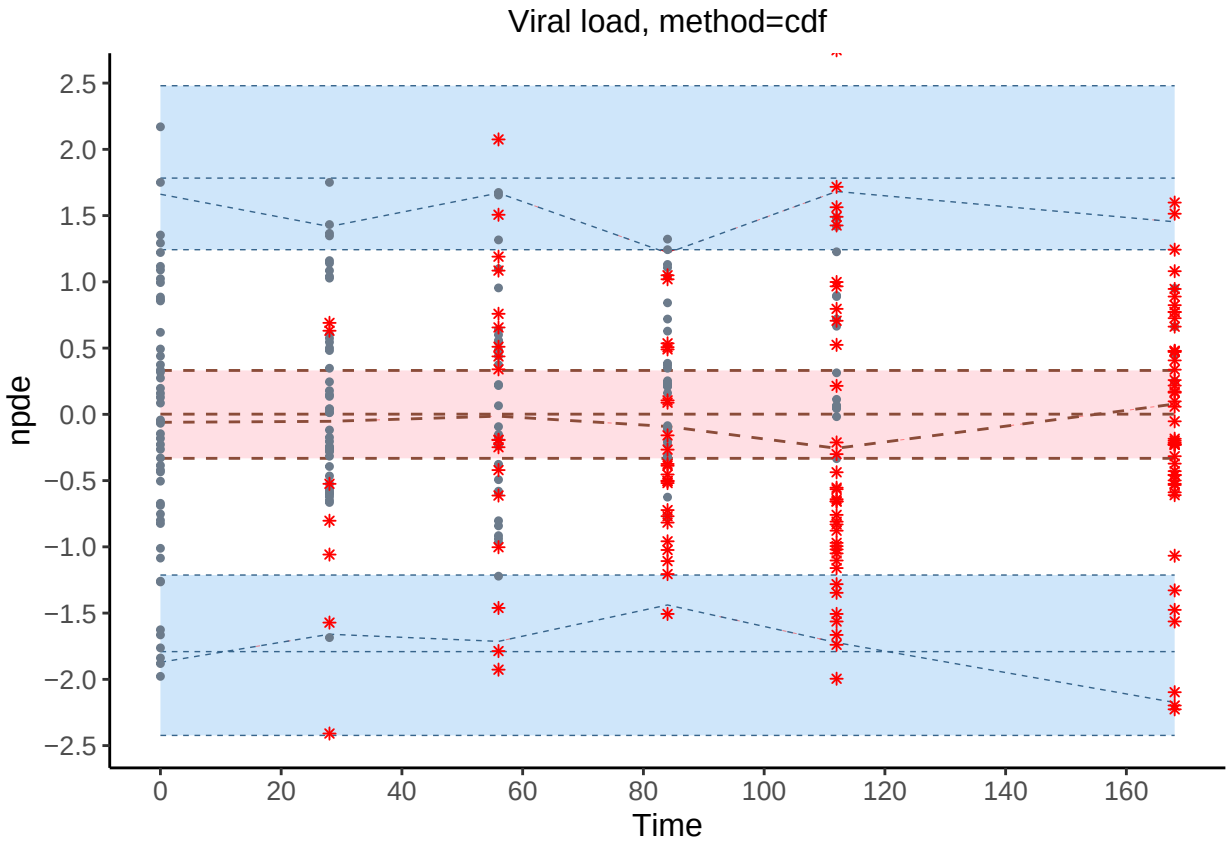


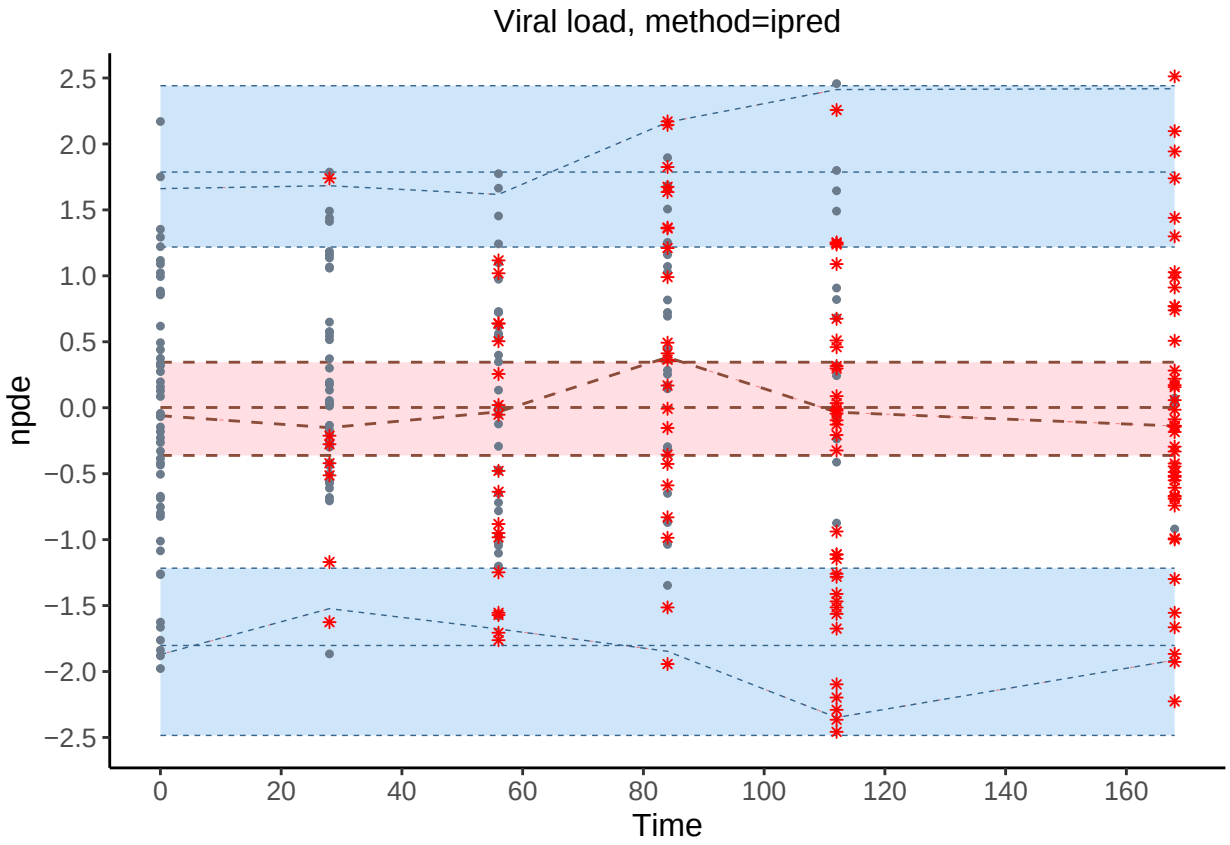
VPC

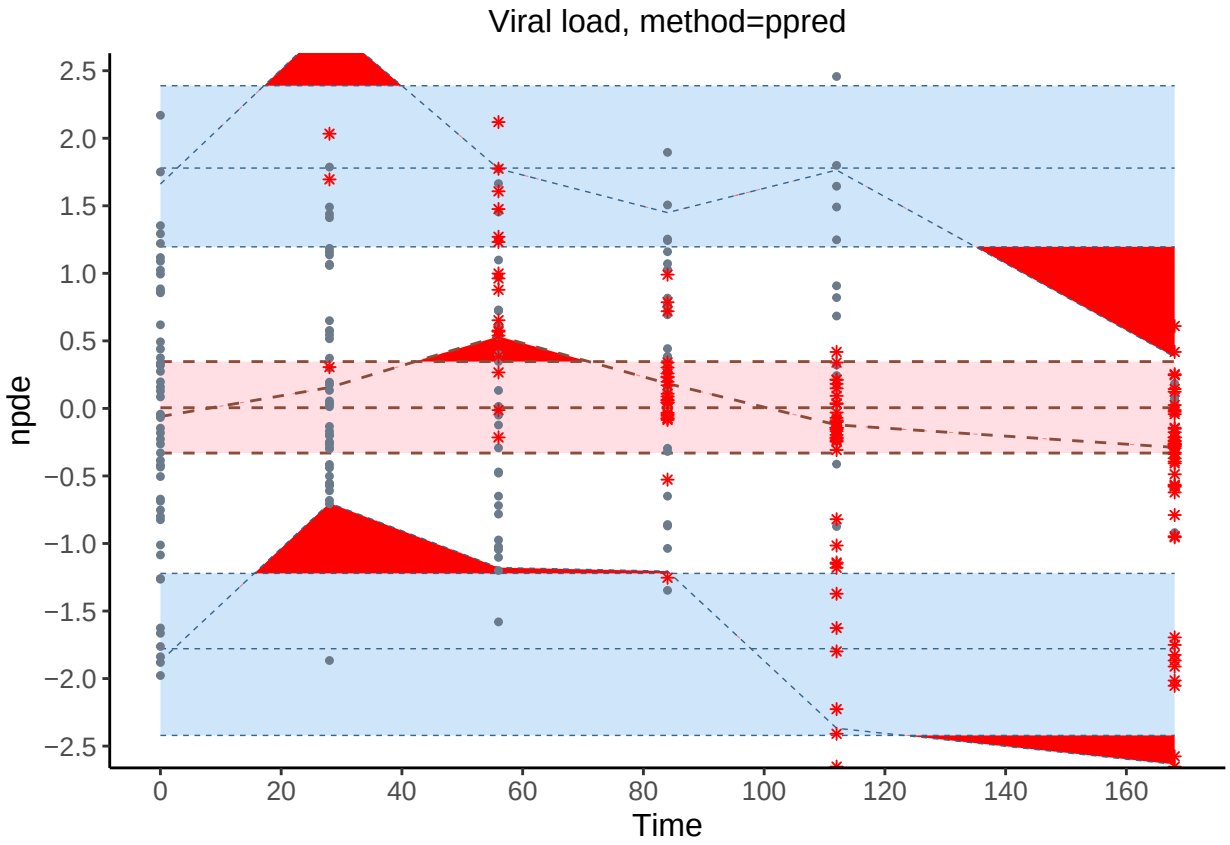
- **Problems** (à résoudre Romain)
  - **Y-axis** missing
  - problem with the PI not being interpolated properly (??? thought that worked :-/)
- **Notes to check Eco**
  - computation of the PI for **omit** consistent with the PI computed for npde ?
    - compute the PI in this case by omitting the 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)


npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0
2
4
6
8
10
12
14
16
18
20
22
24
Time(hr)


Viral load, method=omit
npde
Time
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0
20
40
60
80
100
120
140
160

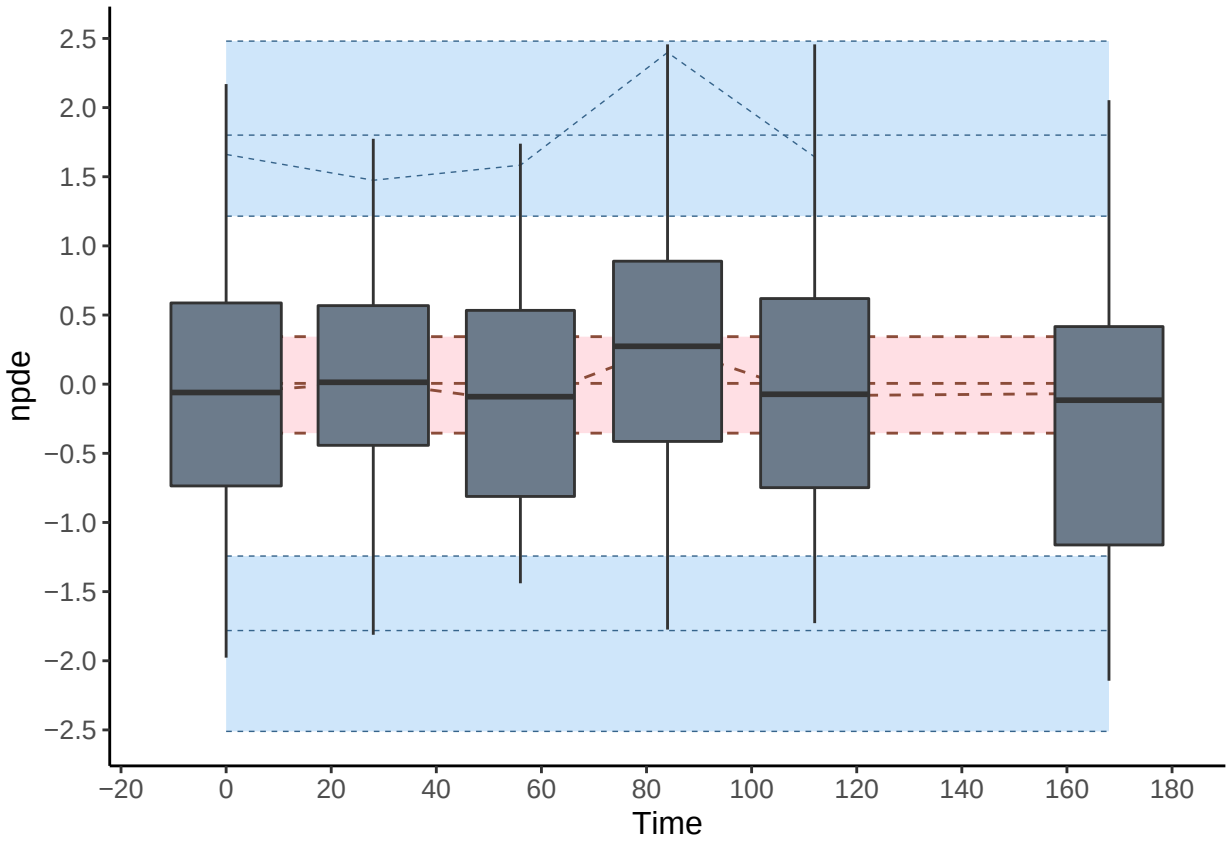

Viral load, method=cdf
npde
Time
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0
20
40
60
80
100
120
140
160


Viral load, method=ipred
npde
Time
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0
20
40
60
80
100
120
140
160



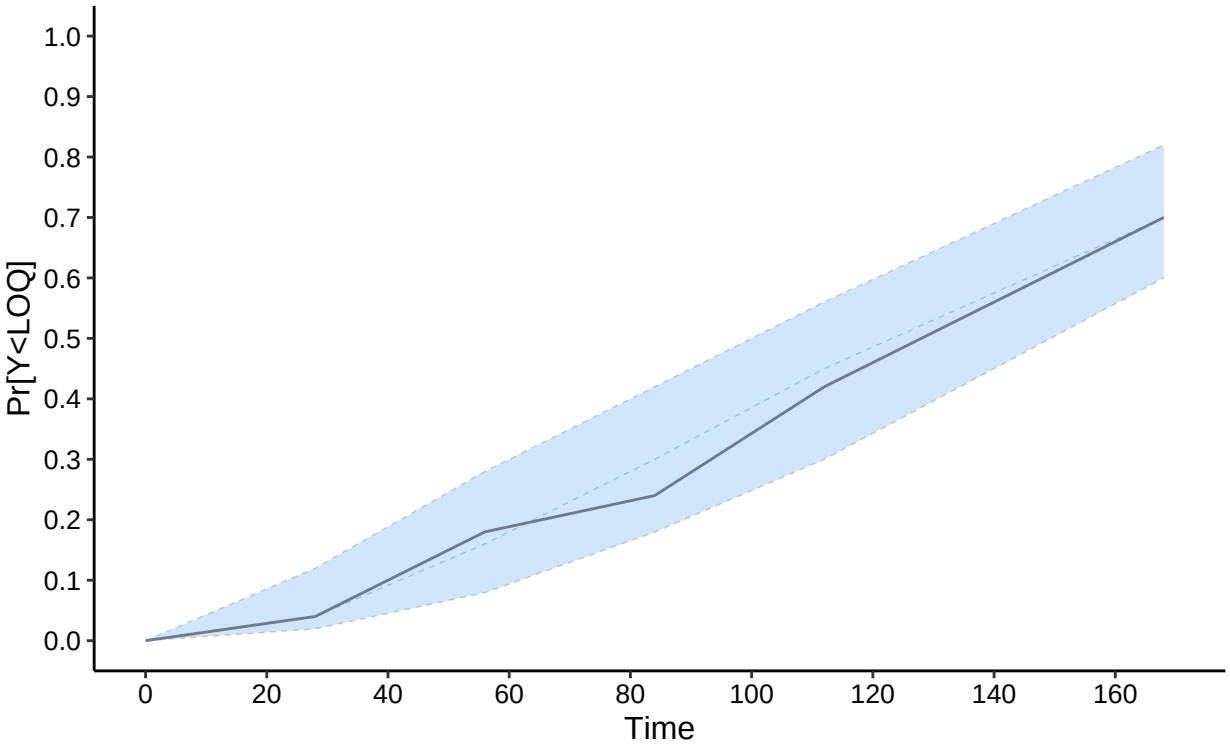

```
## Warning: Removed 8 rows containing non-finite values (stat_boxplot).
```

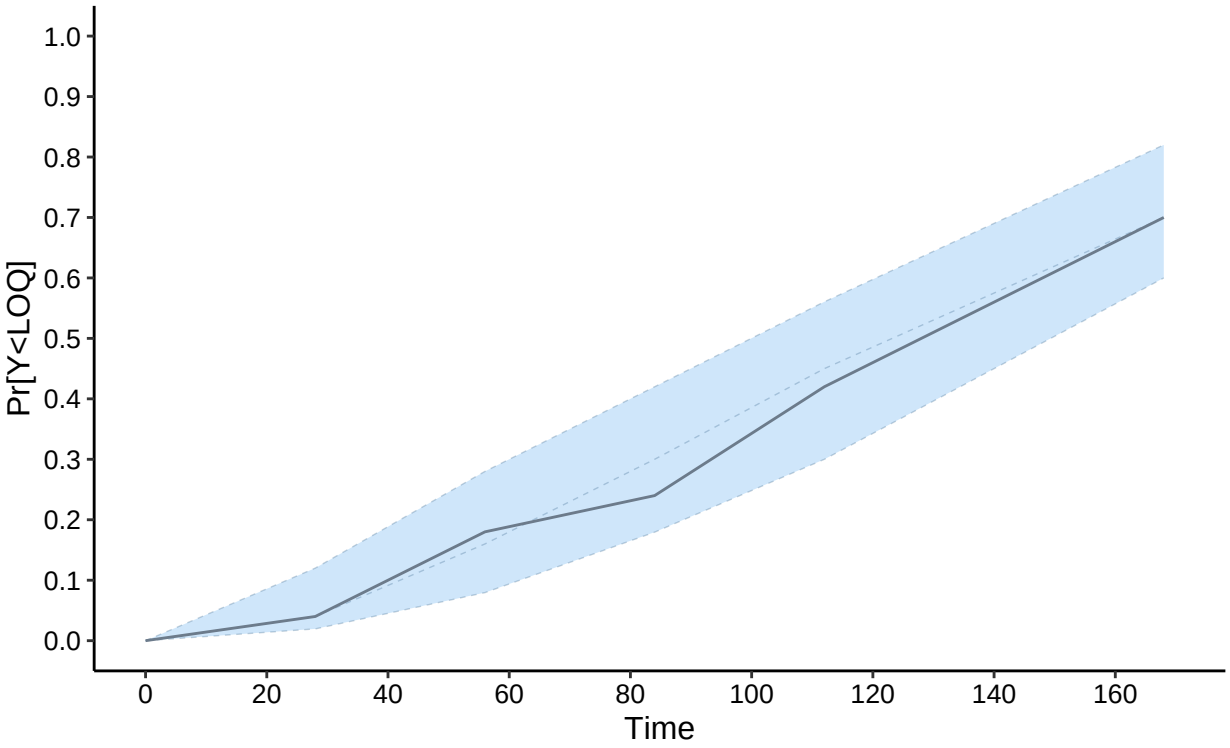
```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

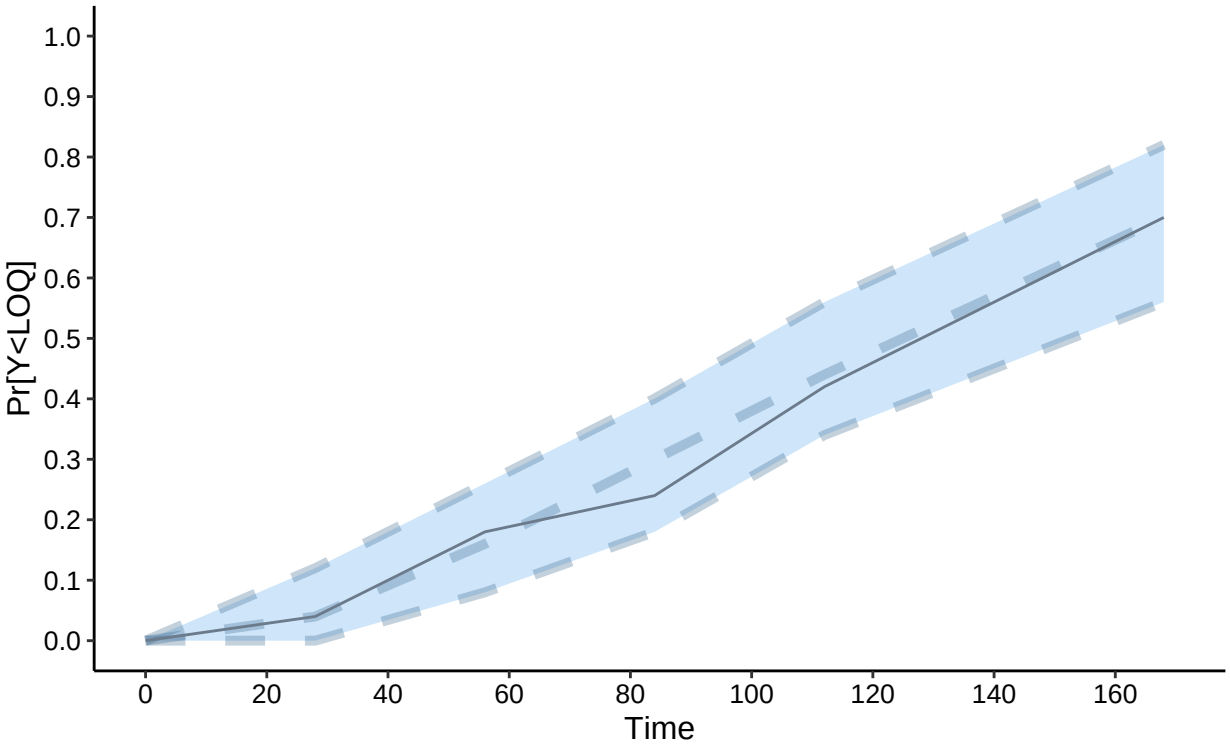


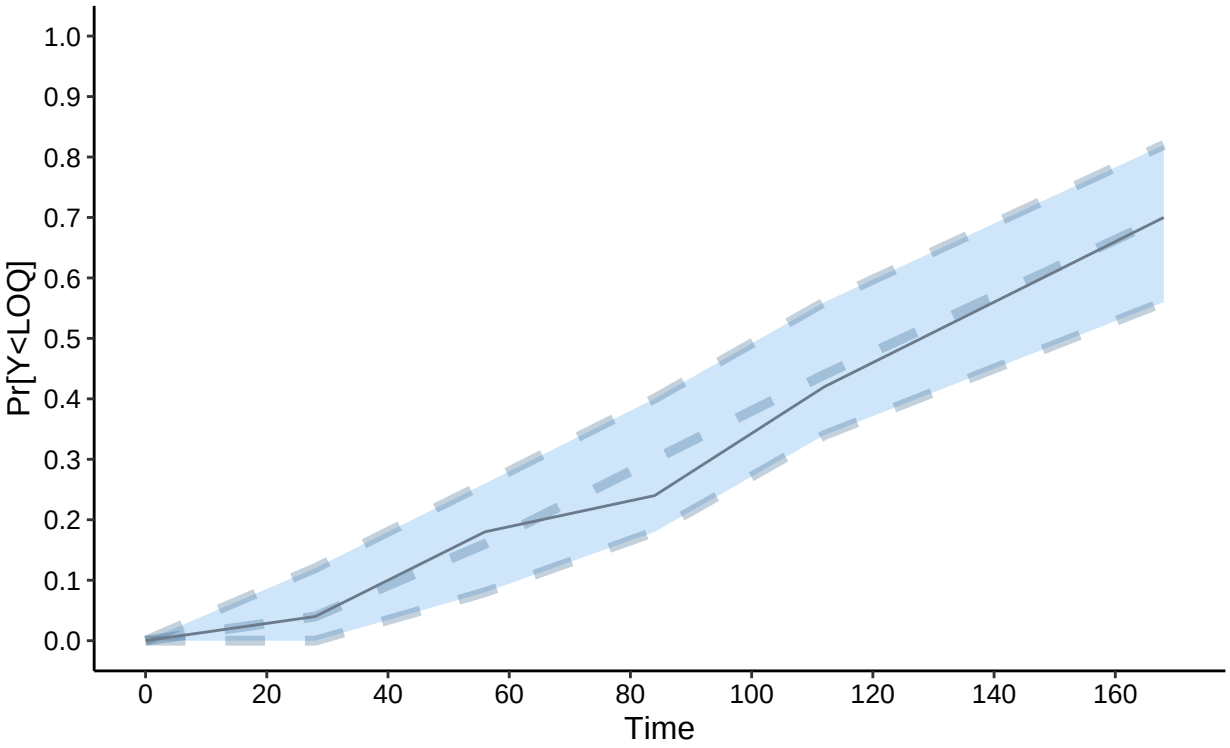
P(Y<LOQ)

- **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


Pr[Y<LOQ]
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
20
40
60
80
100
120
140
160
Time


Pr[Y<LOQ]
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
20
40
60
80
100
120
140
160
Time


Pr[Y<LOQ]
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
20
40
60
80
100
120
140
160
Time

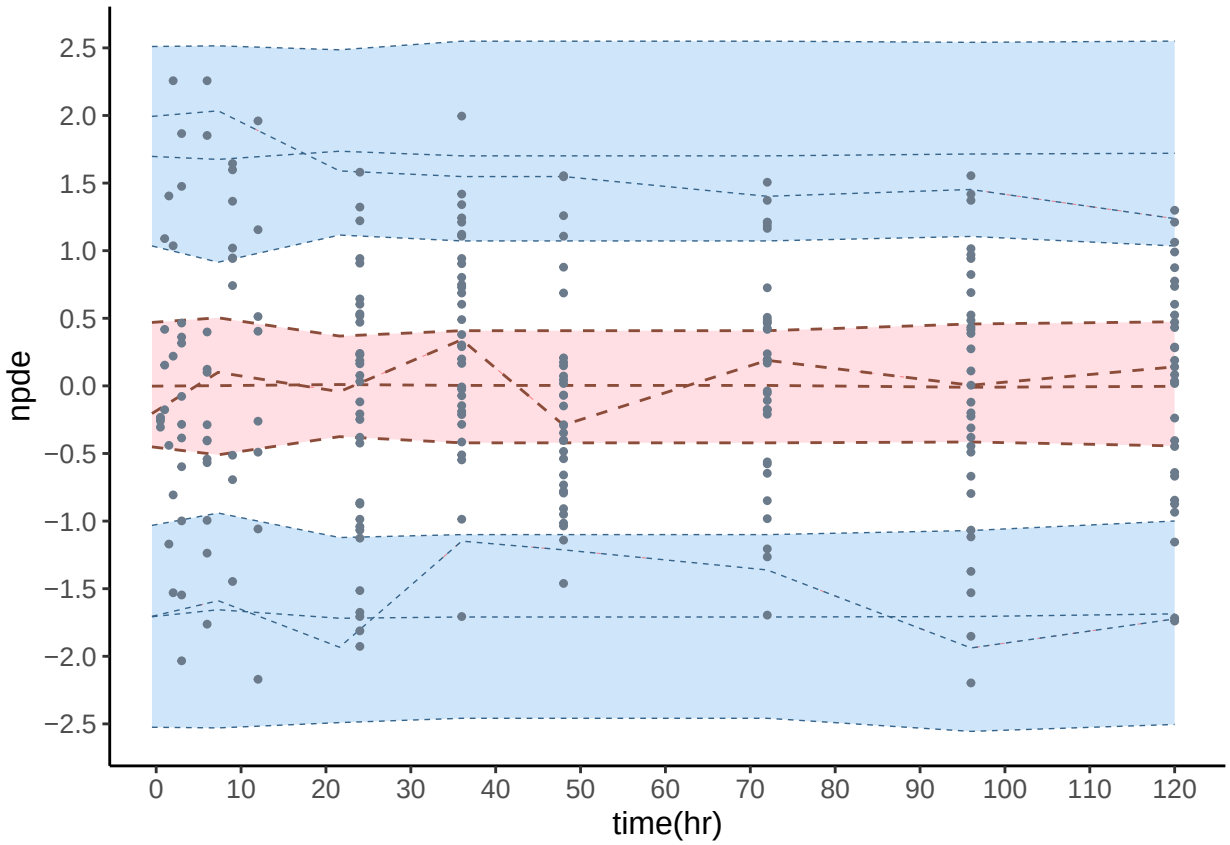


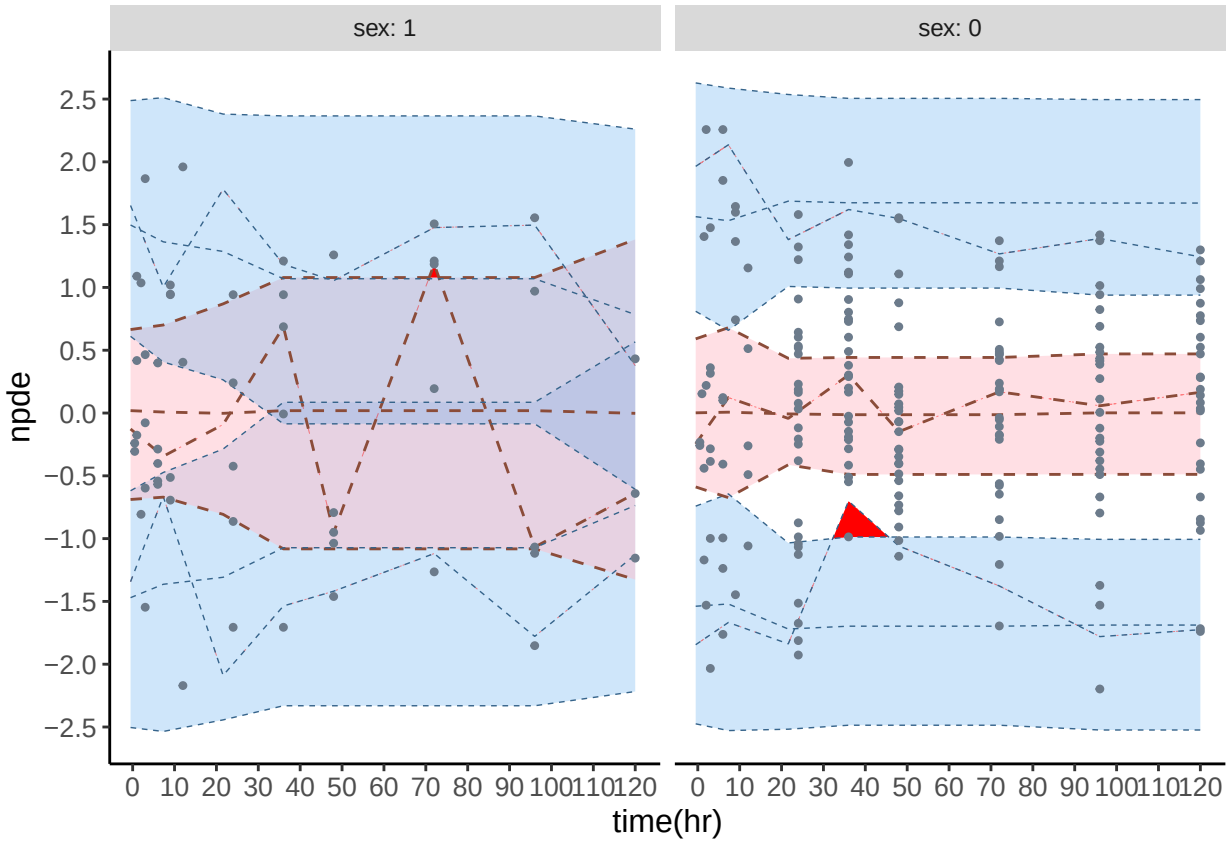
Covariate plots

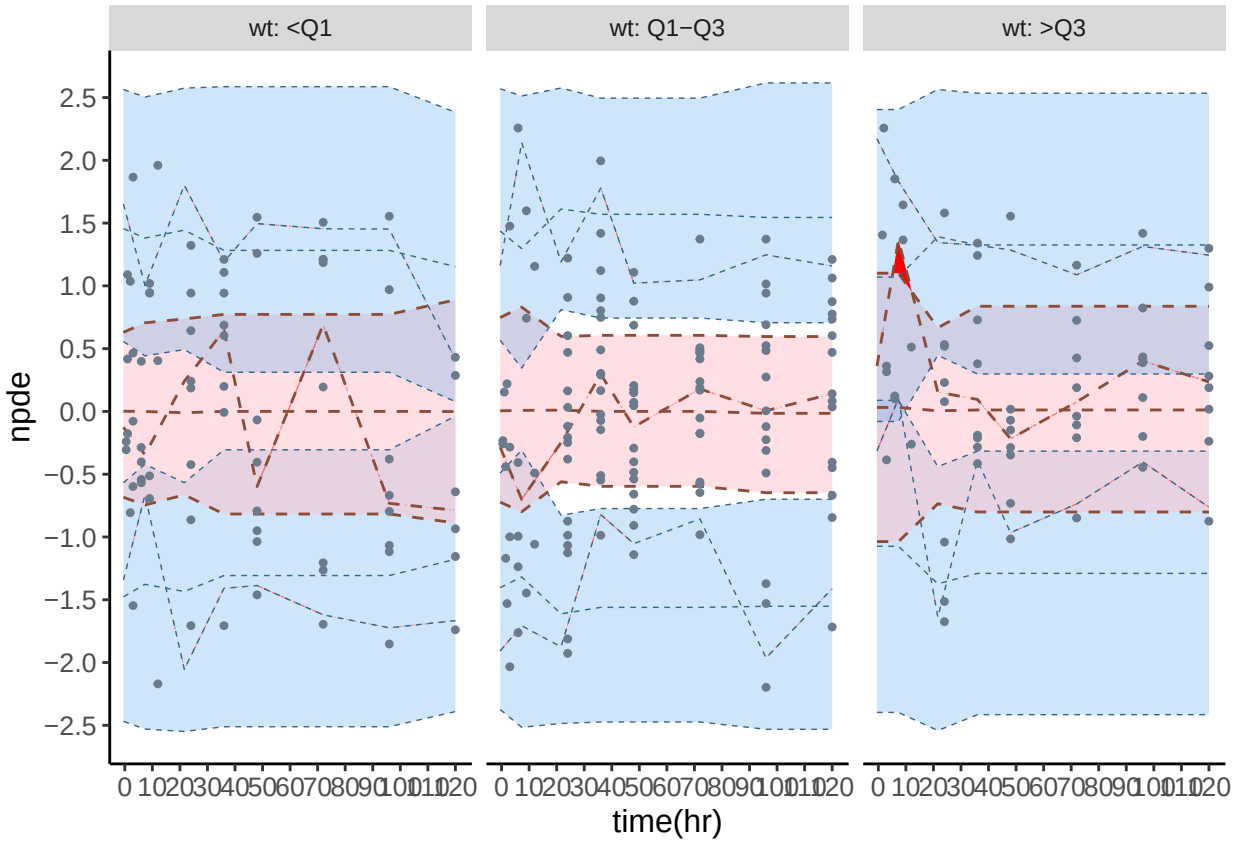
Covariates

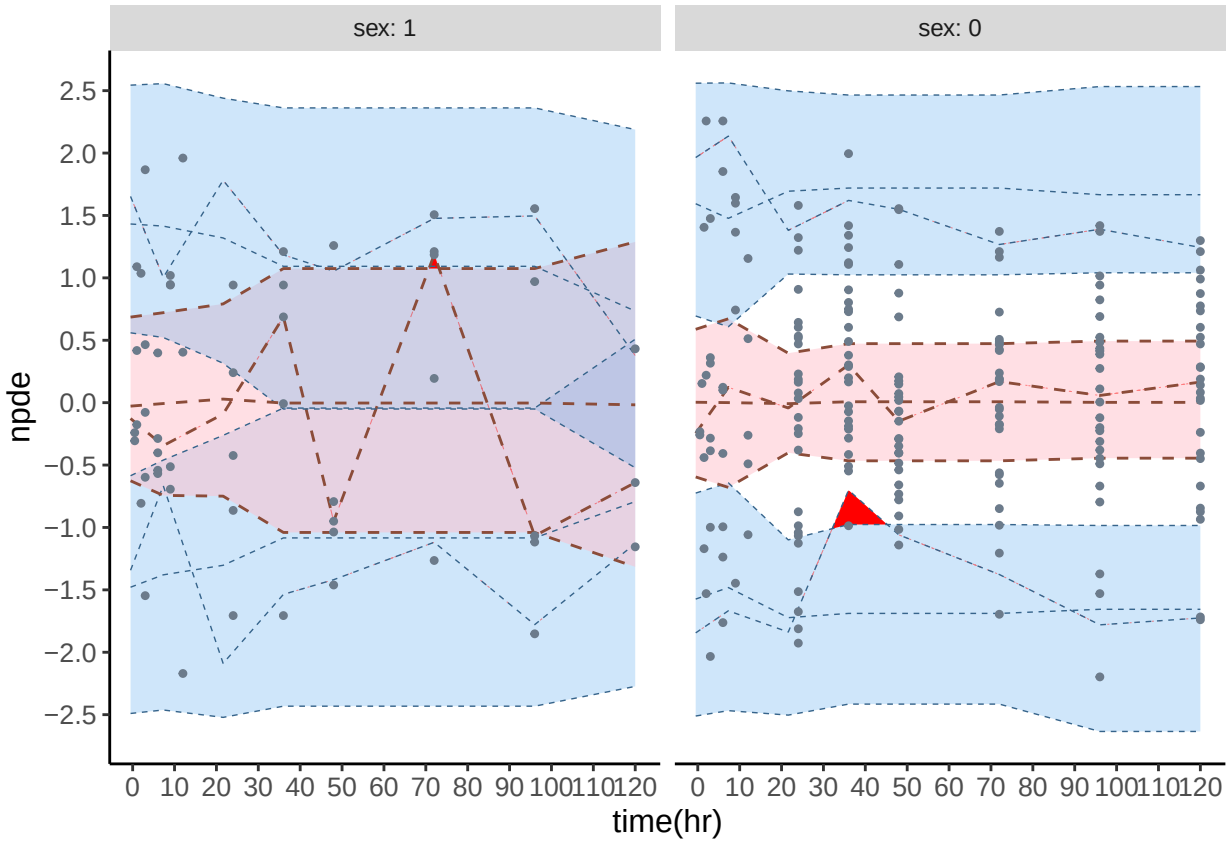
Splitting plots according to covariates

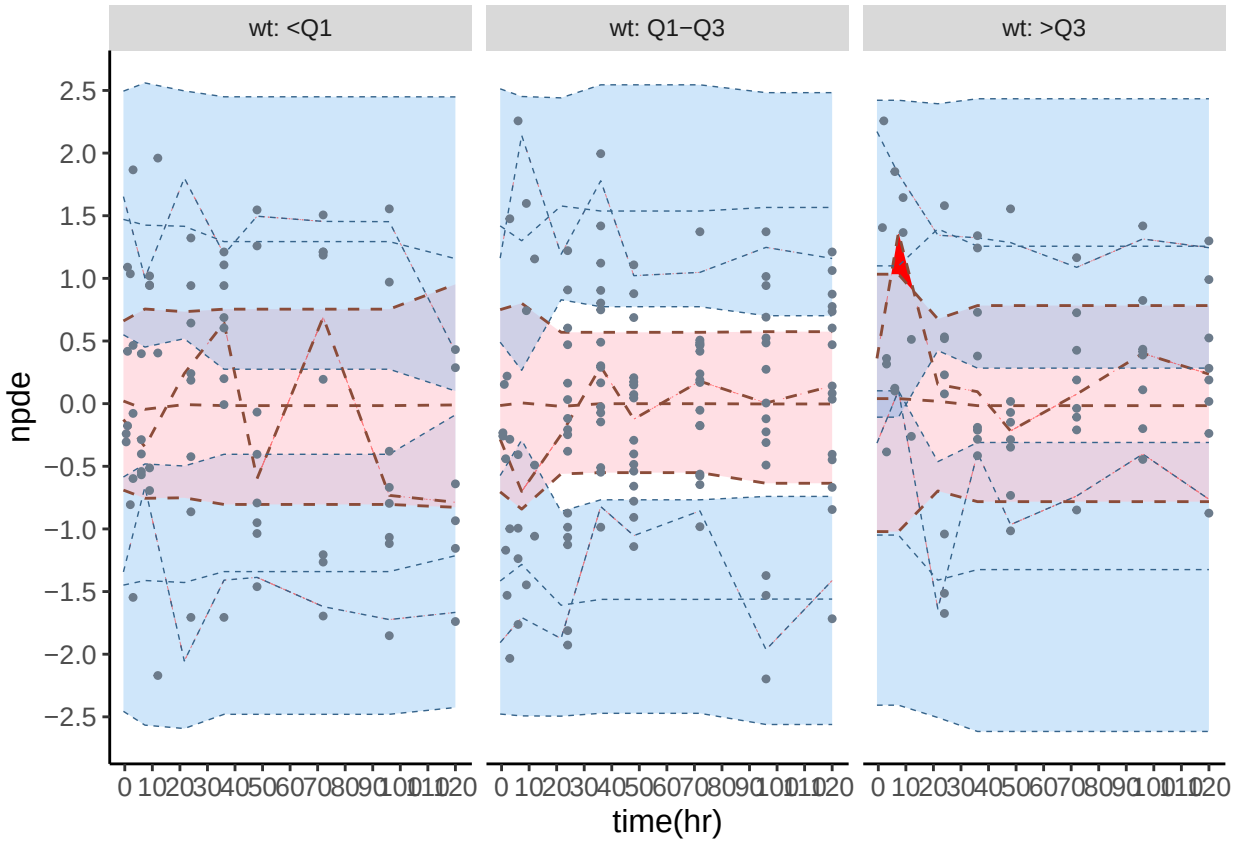
- **Problems** (à résoudre Romain)
  - **solved** yes here we really need to extend PI to cover the whole X-range for x.scatter and pred.scatter (not VPC)
    - just the PI (not the observed percentiles or outlier bands)
  - **covsplit**
    - **solved** doesn't work with the default plots => behaviour 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

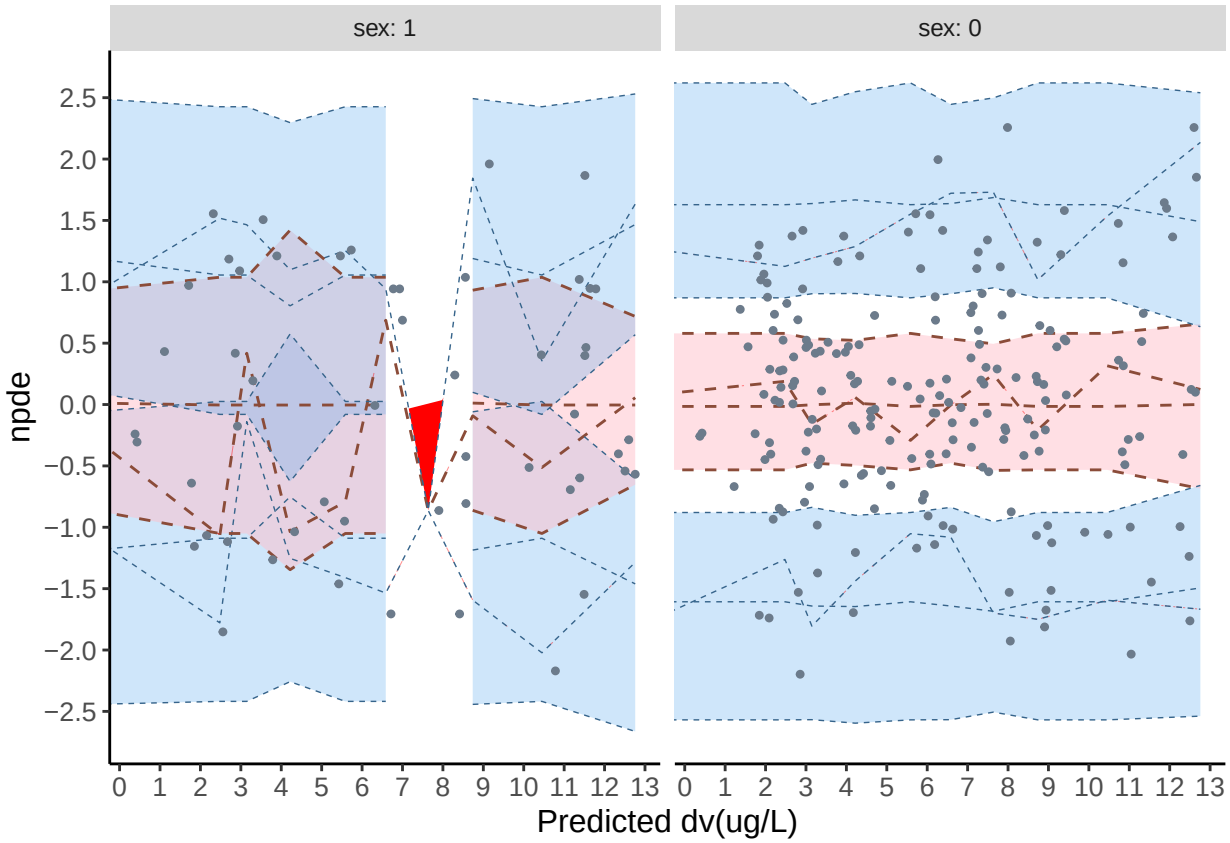

npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0
10
20
30
40
50
60
70
80
90
100
110
120
time(hr)

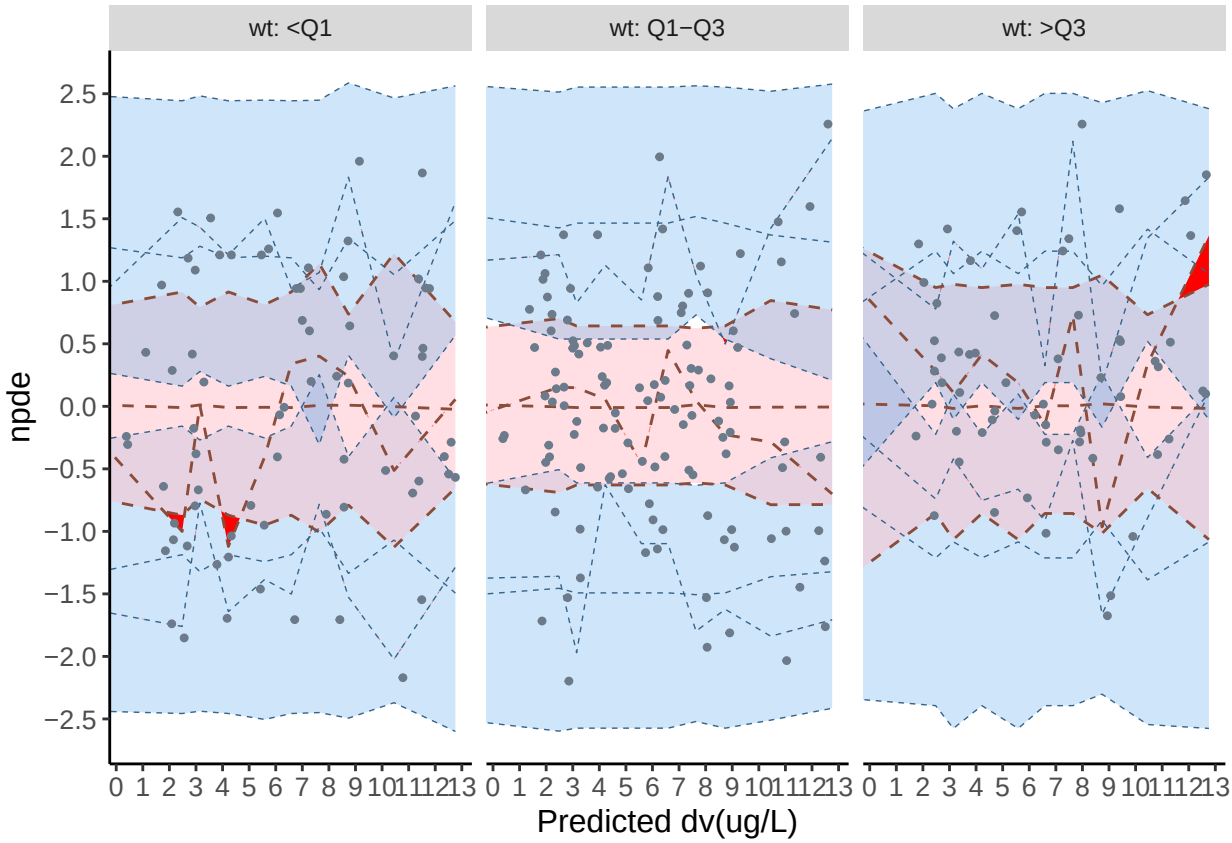

sex: 1
sex: 0
npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0 10 20 30 40 50 60 70 80 90 100 110 120
time(hr)

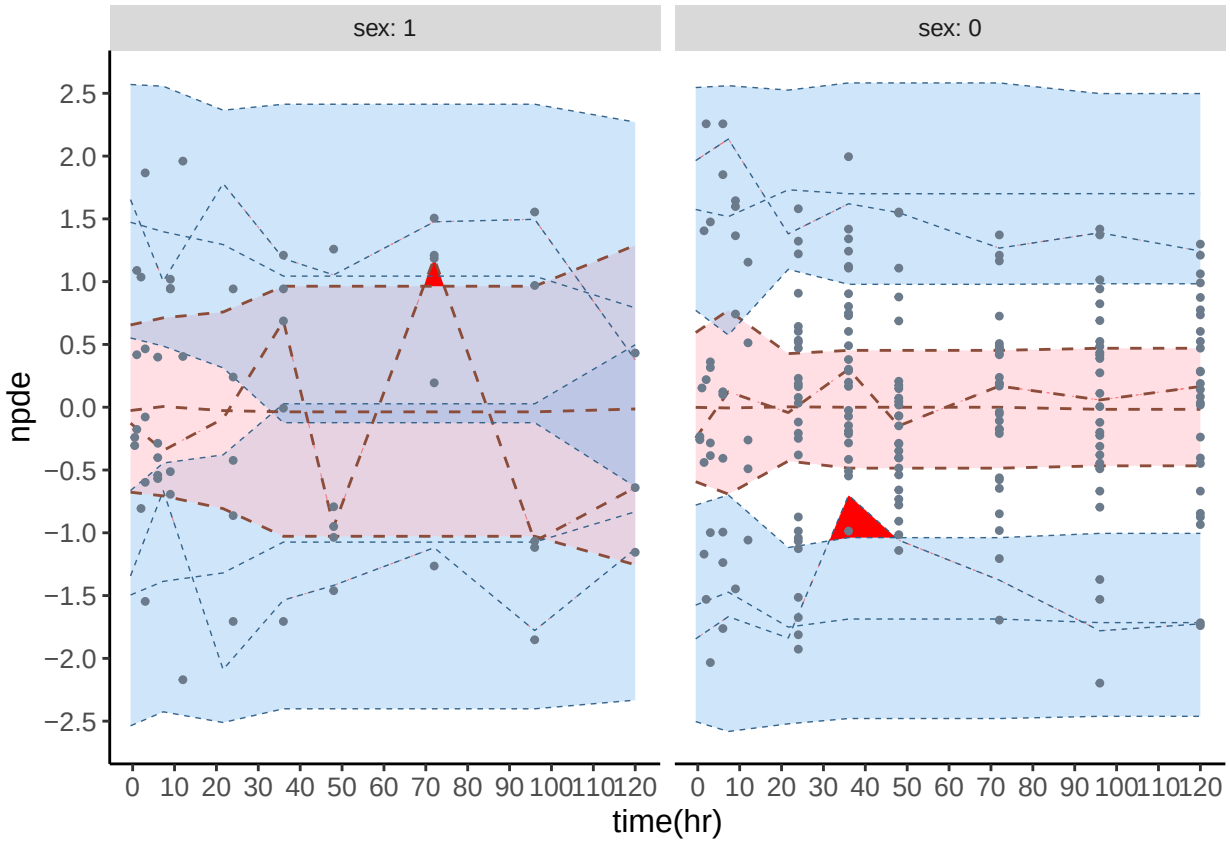

wt: <Q1
wt: Q1−Q3
wt: >Q3
npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0 10 20 30 40 50 60 70 80 90 100 110 120
time(hr)

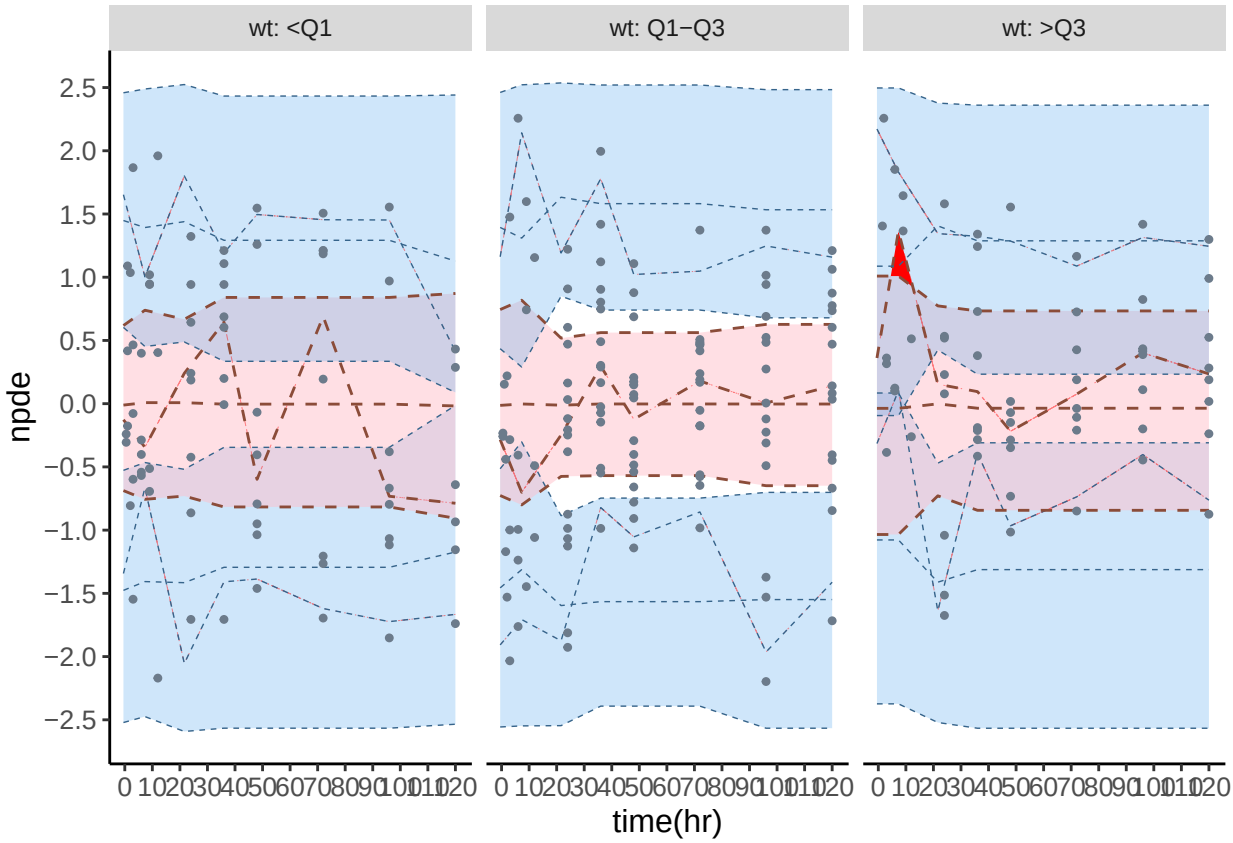

sex: 1
sex: 0
npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0 10 20 30 40 50 60 70 80 90 100 110 120
time(hr)

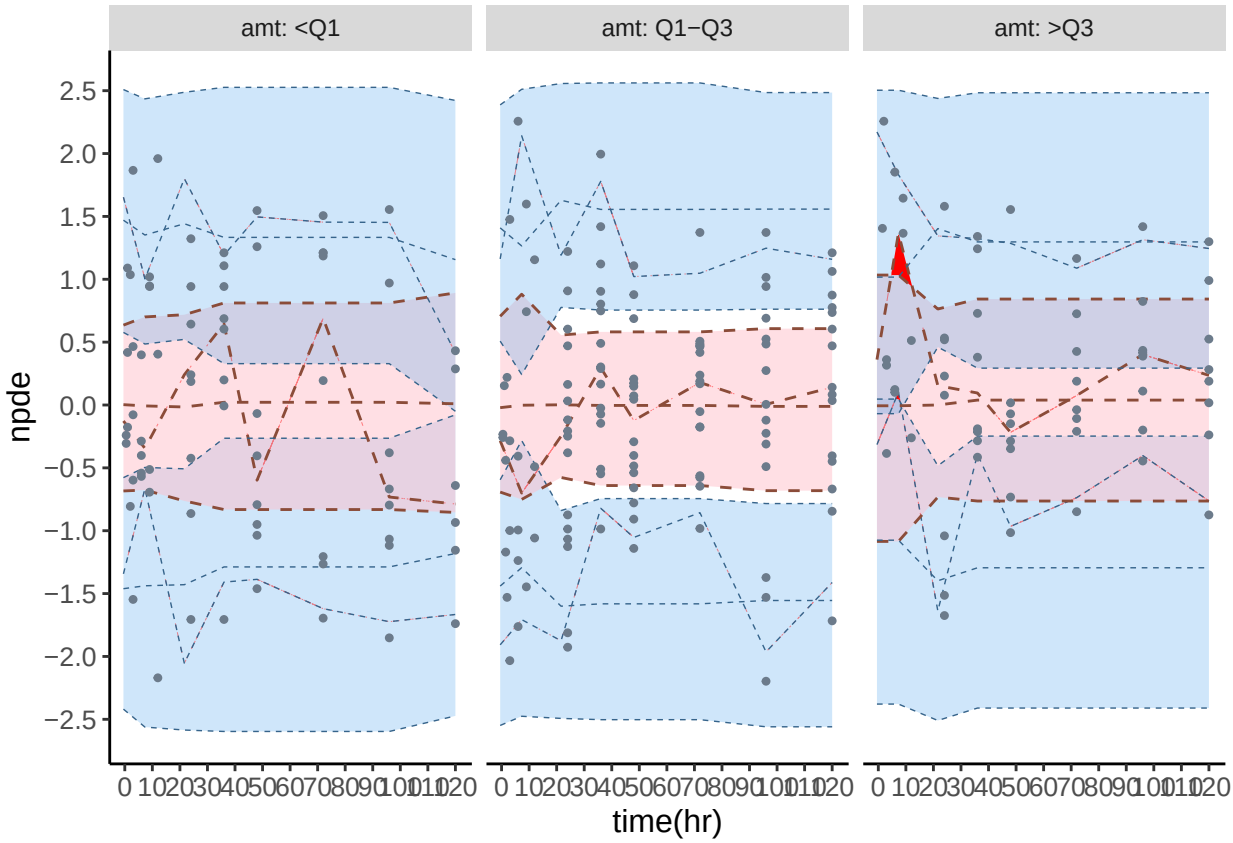

wt: <Q1
wt: Q1−Q3
wt: >Q3
npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0 10 20 30 40 50 60 70 80 90 100 110 120
time(hr)

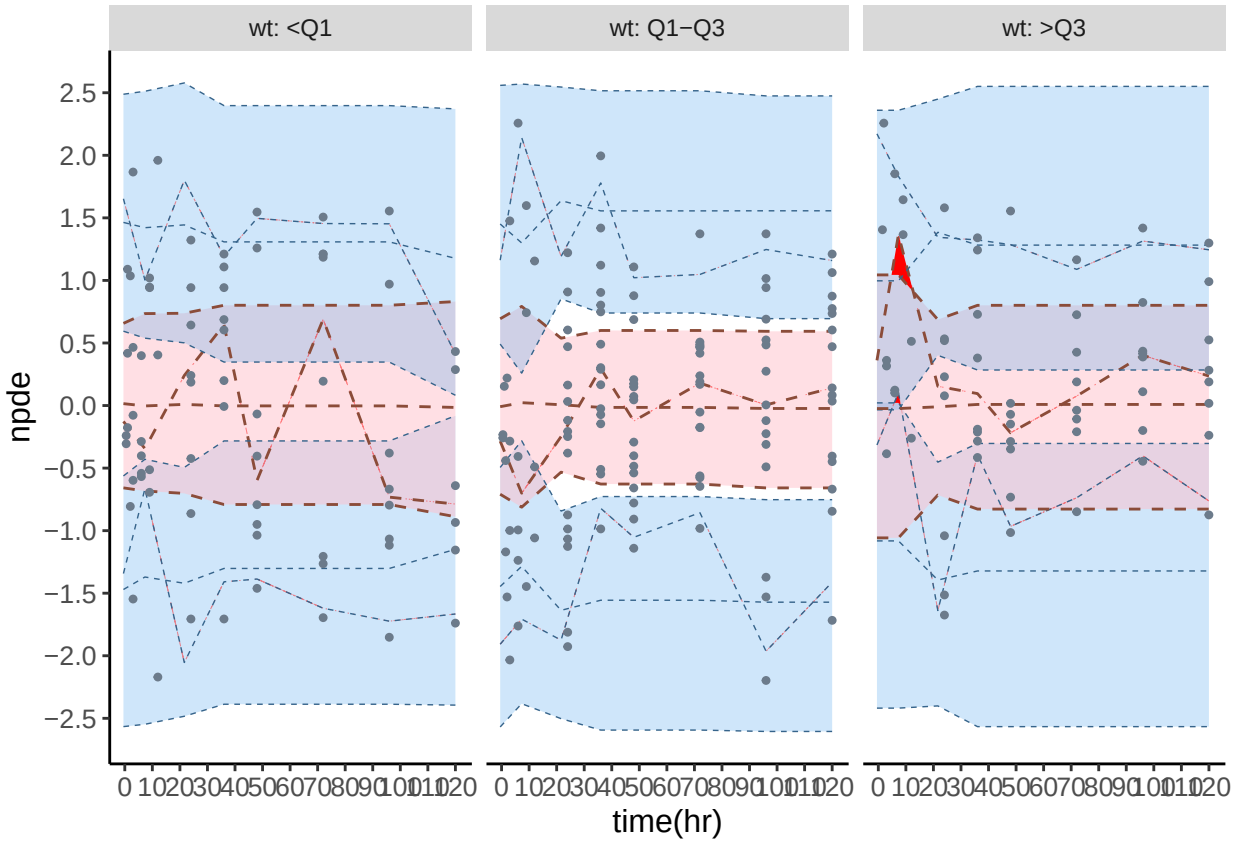

sex: 1
sex: 0
npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0 1 2 3 4 5 6 7 8 9 10 11 12 13
Predicted dv(ug/L)

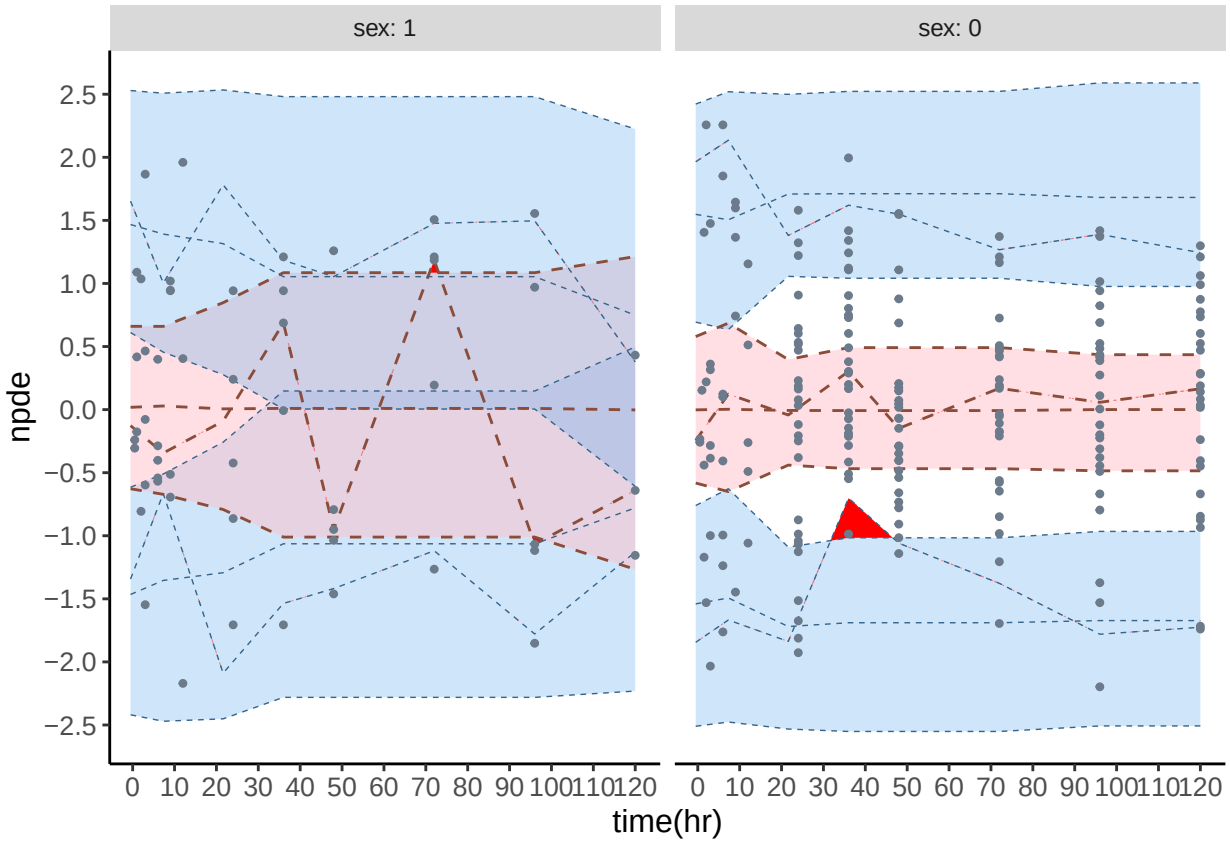

wt: <Q1
wt: Q1−Q3
wt: >Q3
npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0 1 2 3 4 5 6 7 8 9 10 11 12 13
Predicted dv(ug/L)

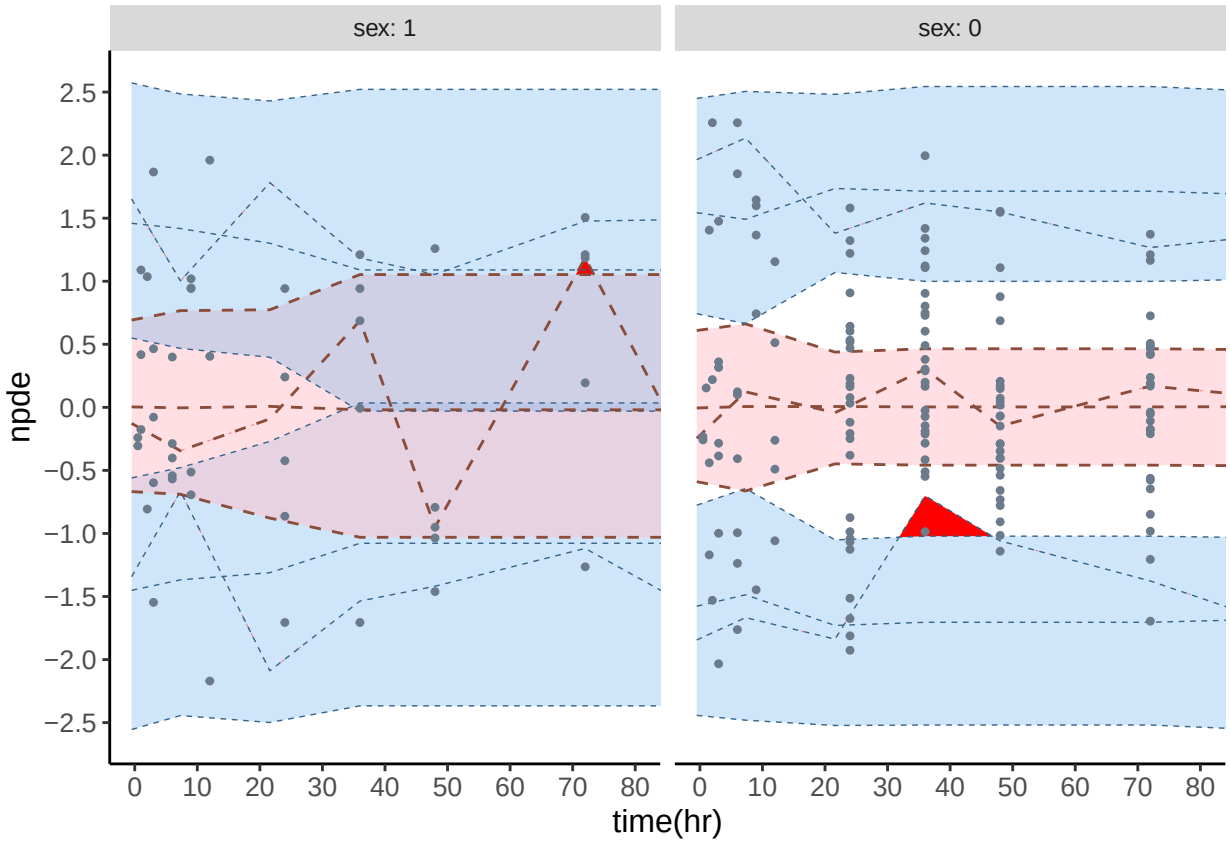

sex: 1
sex: 0
npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0 10 20 30 40 50 60 70 80 90 100 110 120
time(hr)

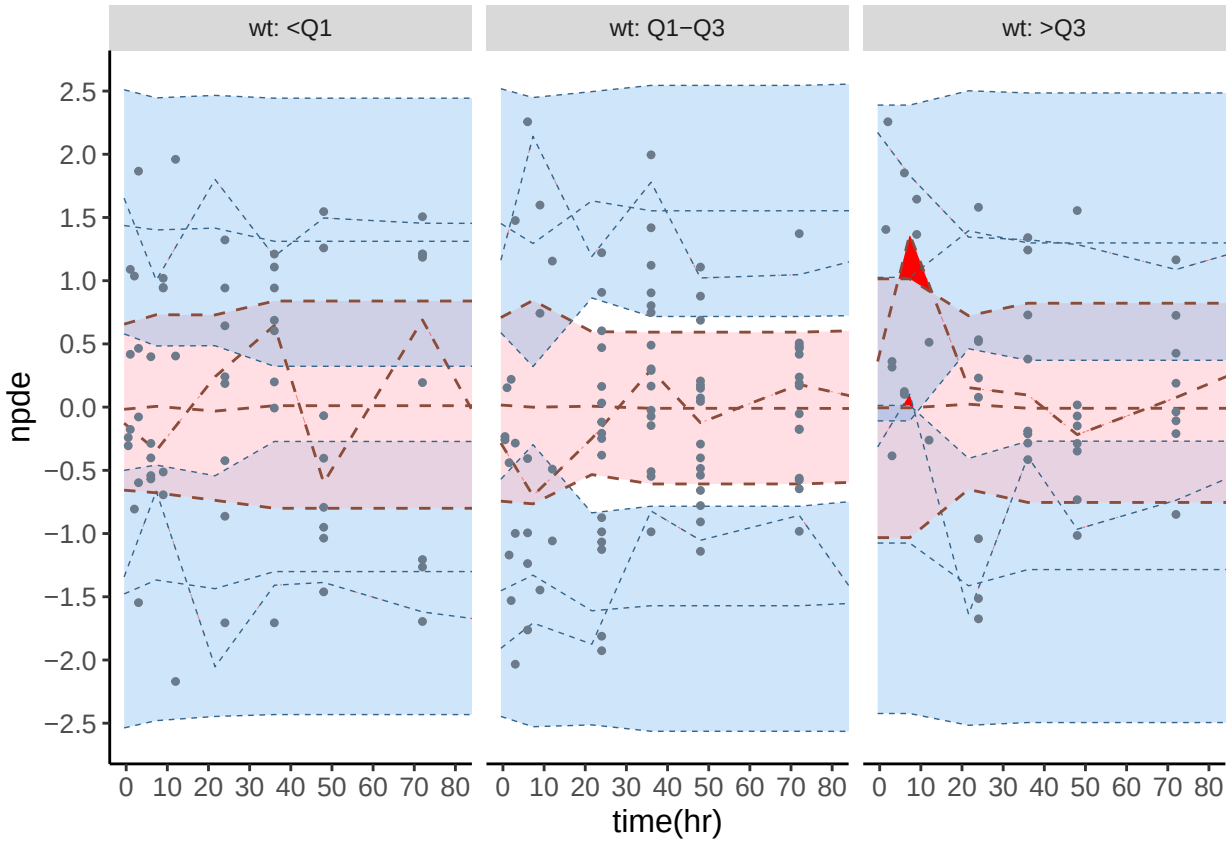

wt: <Q1
wt: Q1−Q3
wt: >Q3
npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0 10 20 30 40 50 60 70 80 90 100 110 120
time(hr)

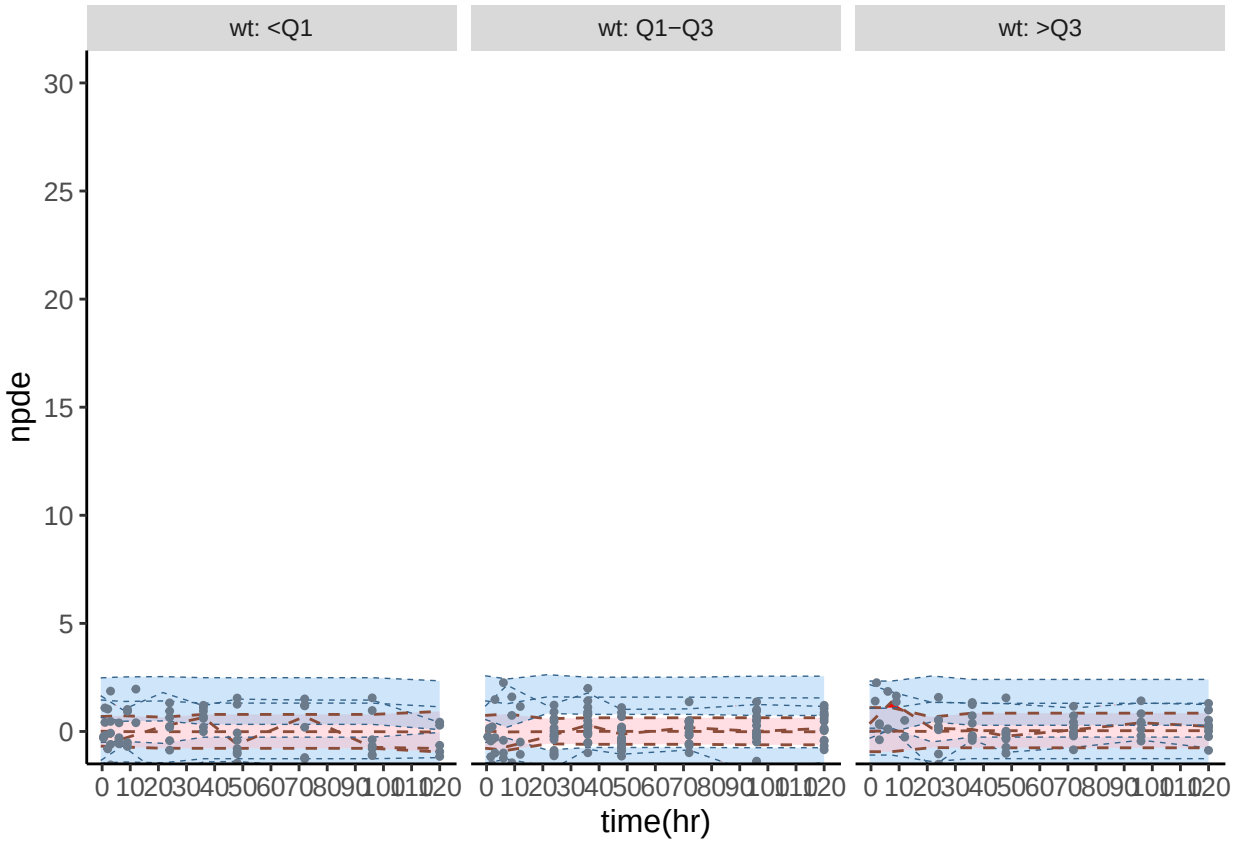

amt: <Q1
amt: Q1−Q3
amt: >Q3
npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0 10 20 30 40 50 60 70 80 90 100 110 120
time(hr)

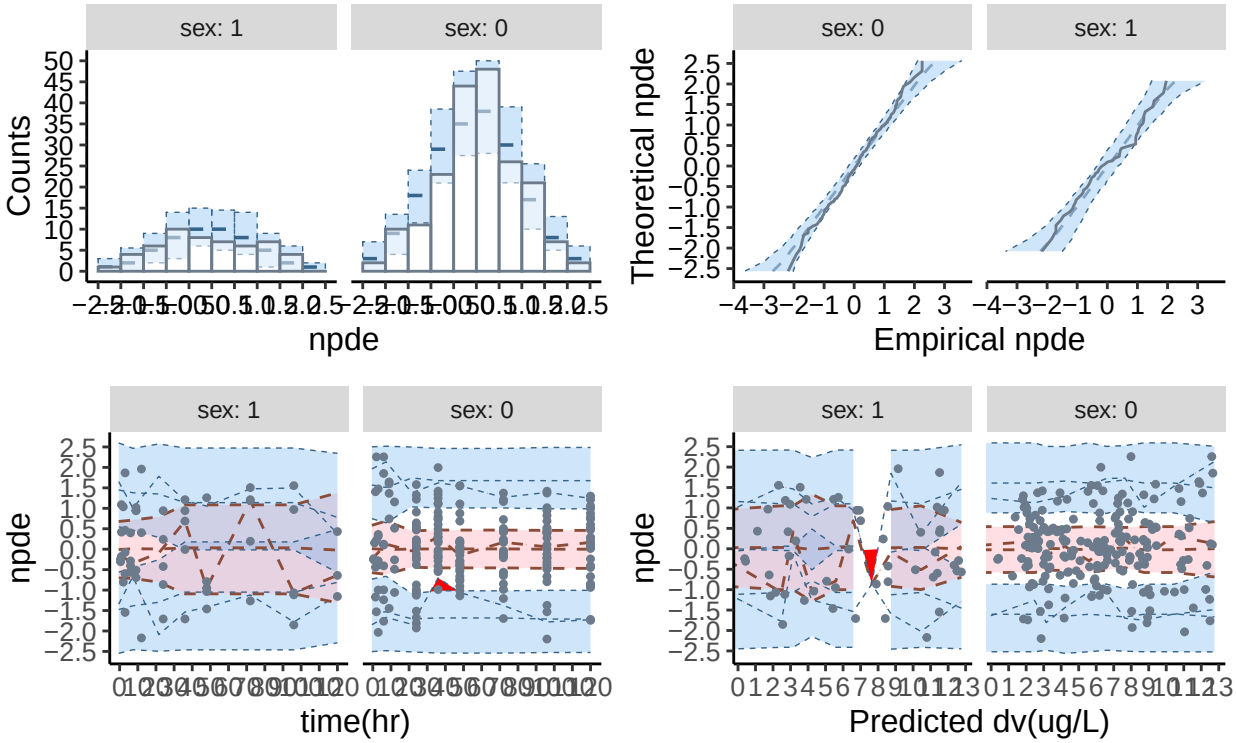

wt: <Q1
wt: Q1−Q3
wt: >Q3
npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0 10 20 30 40 50 60 70 80 90 100 110 120
time(hr)

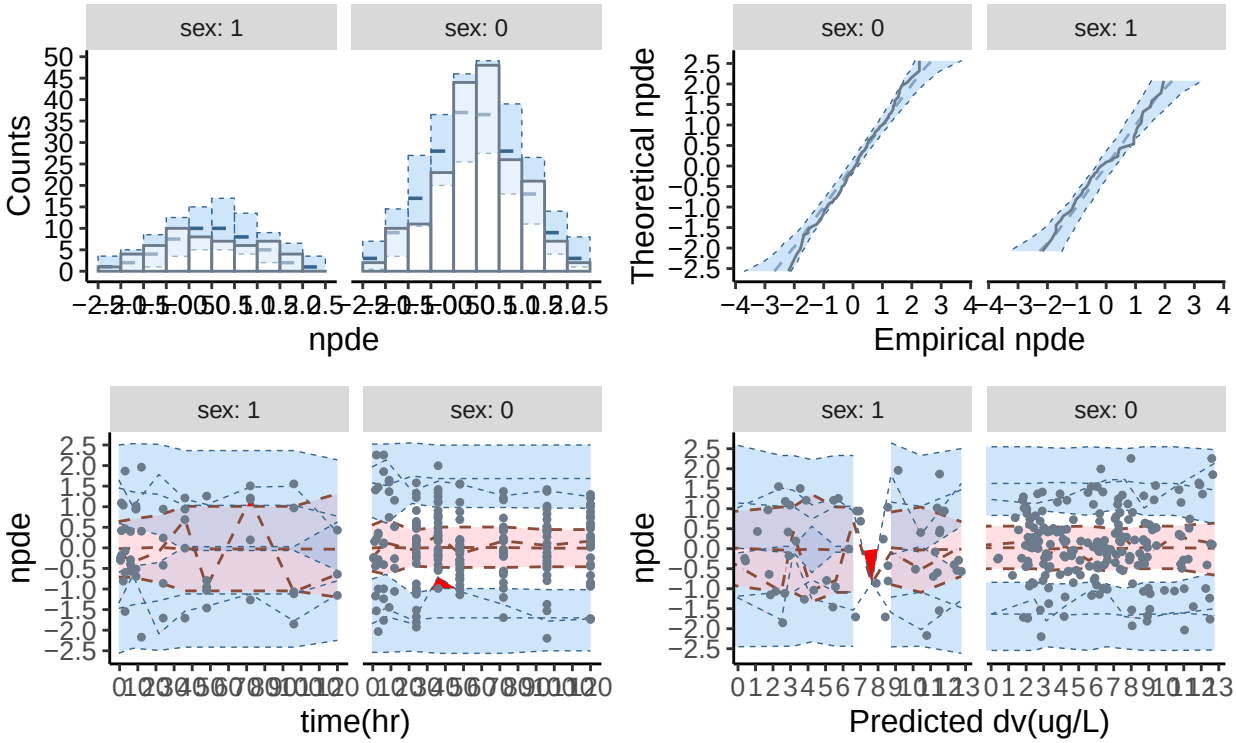

sex: 1
sex: 0
npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0 10 20 30 40 50 60 70 80 90 100 110 120
time(hr)

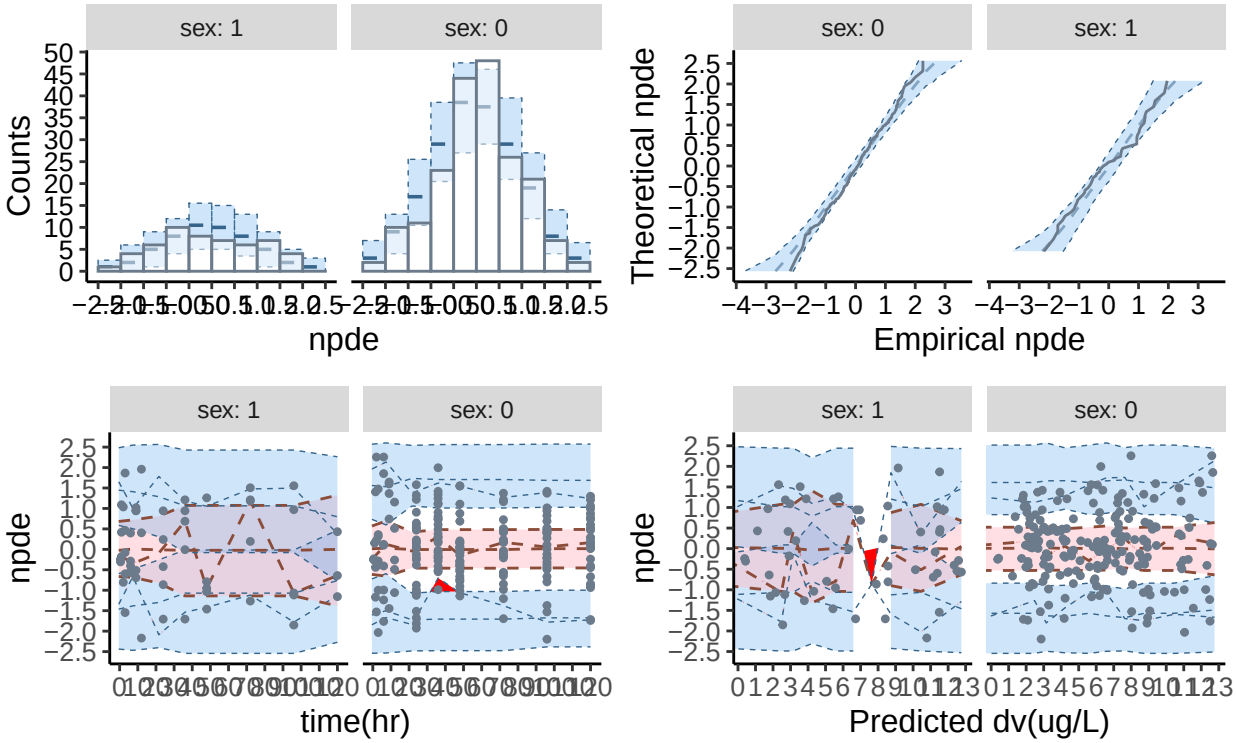

sex: 1
sex: 0
npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0
10
20
30
40
50
60
70
80
time(hr)

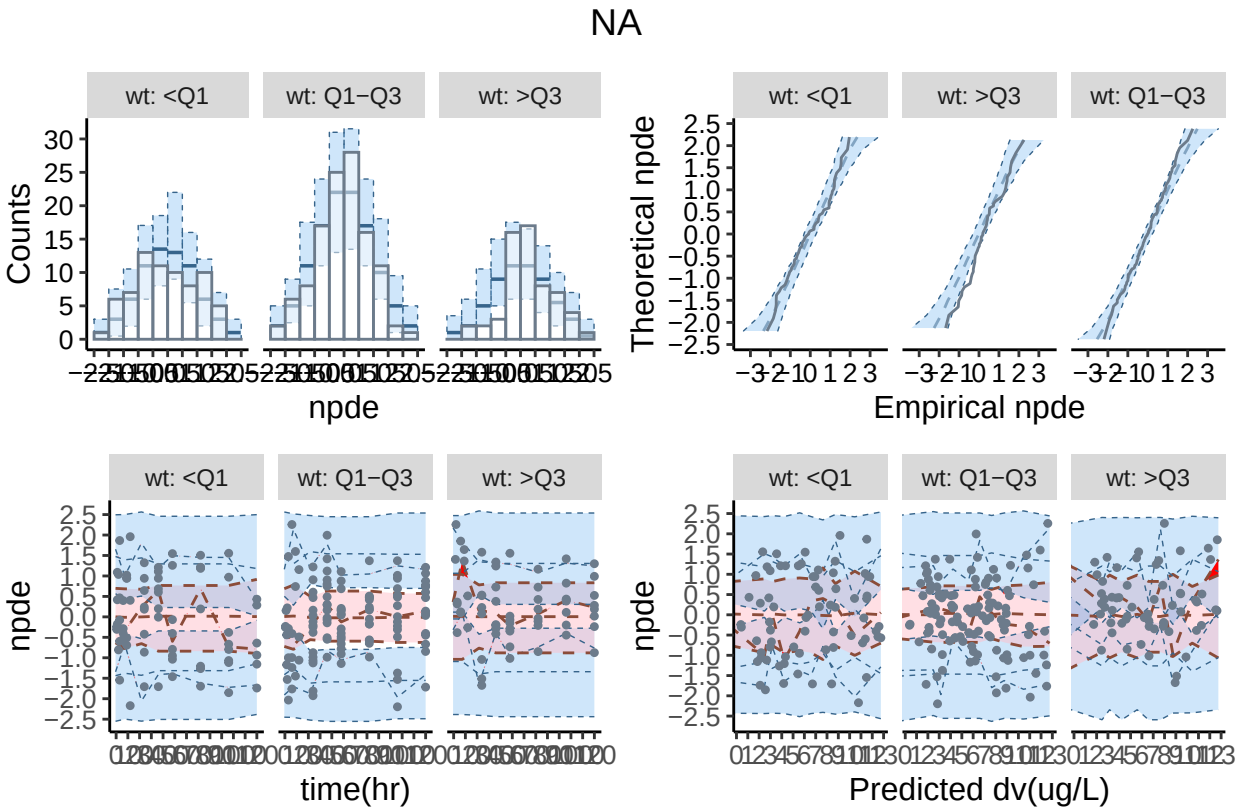

wt: <Q1
wt: Q1−Q3
wt: >Q3
npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0 10 20 30 40 50 60 70 80
time(hr)

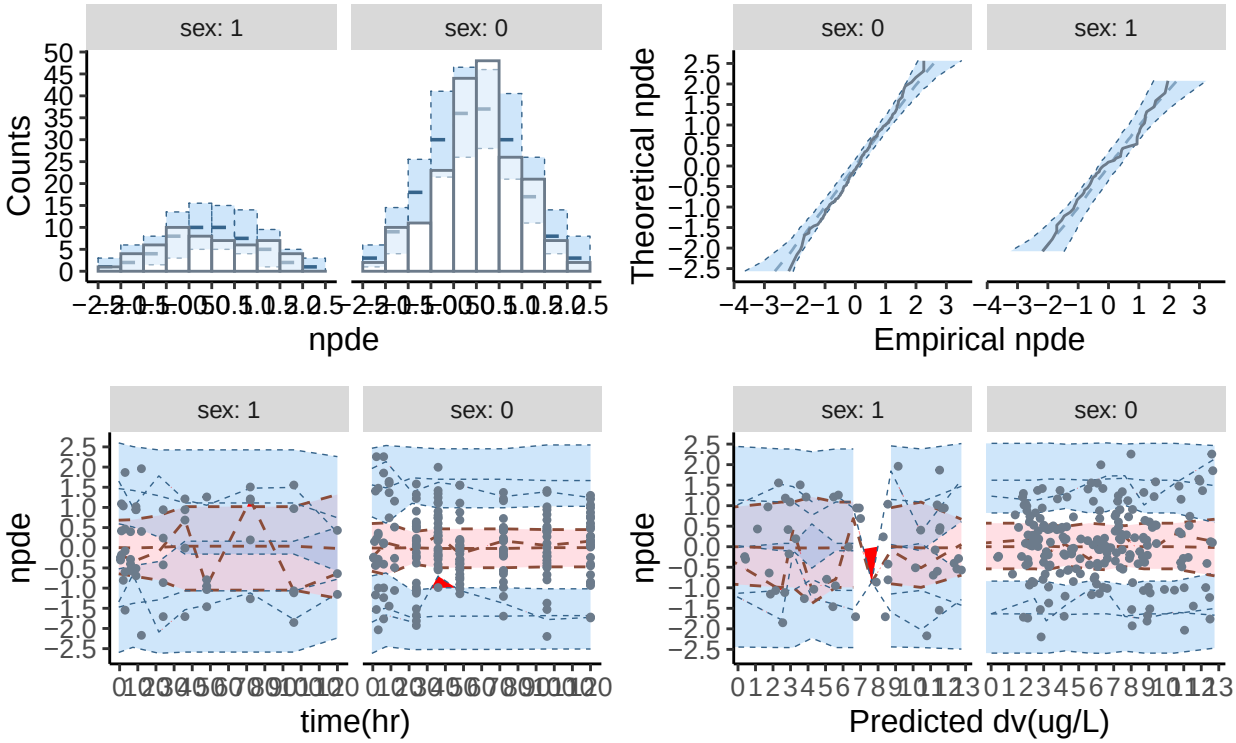

wt: <Q1
wt: Q1−Q3
wt: >Q3
npde
30
25
20
15
10
5
0
0 10 20 30 40 50 60 70 80 90 100 110 120
time(hr)

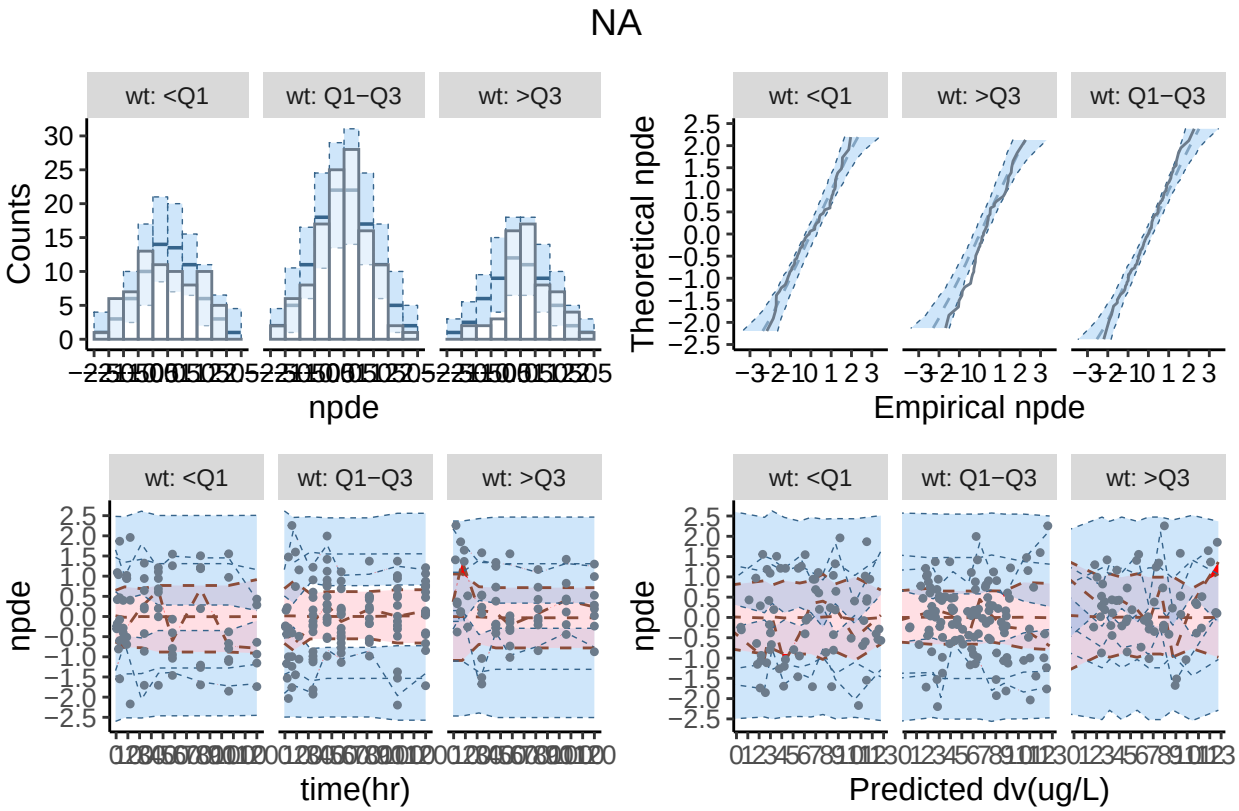

sex: 1
sex: 0
Counts
npde
sex: 0
sex: 1
Theoretical npde
Empirical npde
sex: 1
sex: 0
npde
time(hr)
sex: 1
sex: 0
npde
Predicted dv(ug/L)

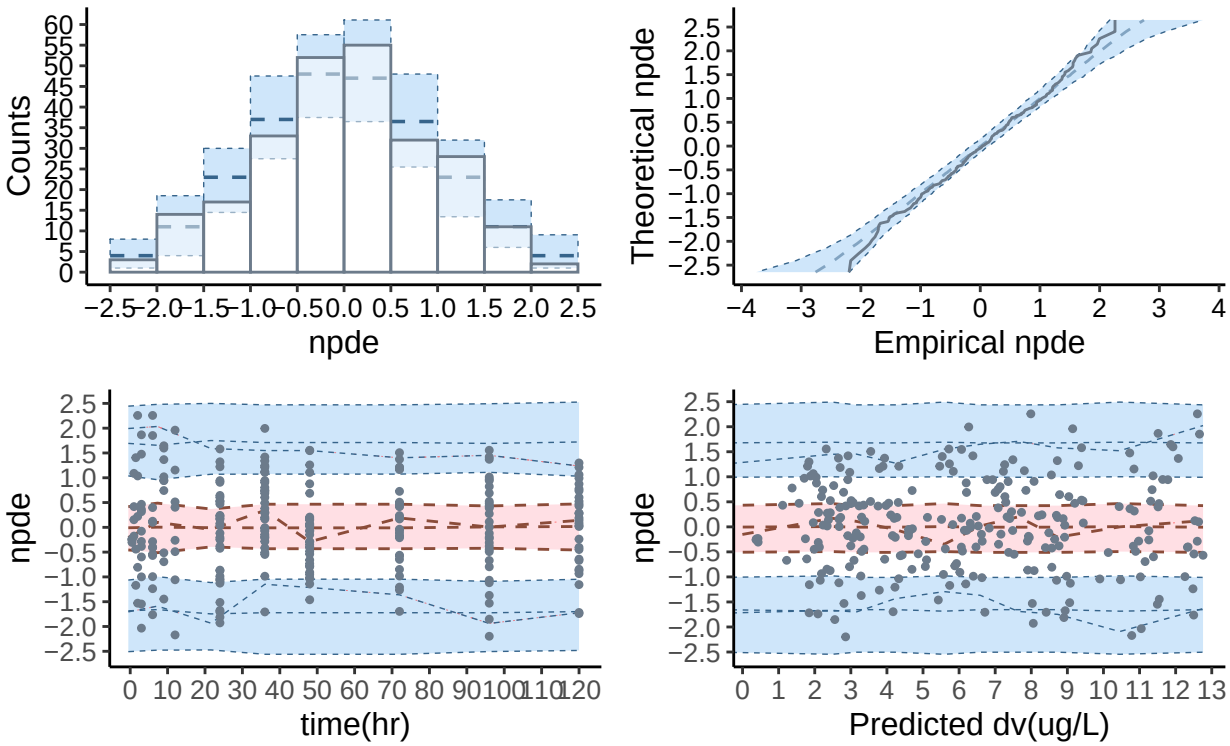

sex: 1
sex: 0
Counts
npde
sex: 0
sex: 1
Theoretical npde
Empirical npde
sex: 1
sex: 0
npde
time(hr)
sex: 1
sex: 0
npde
Predicted dv(ug/L)

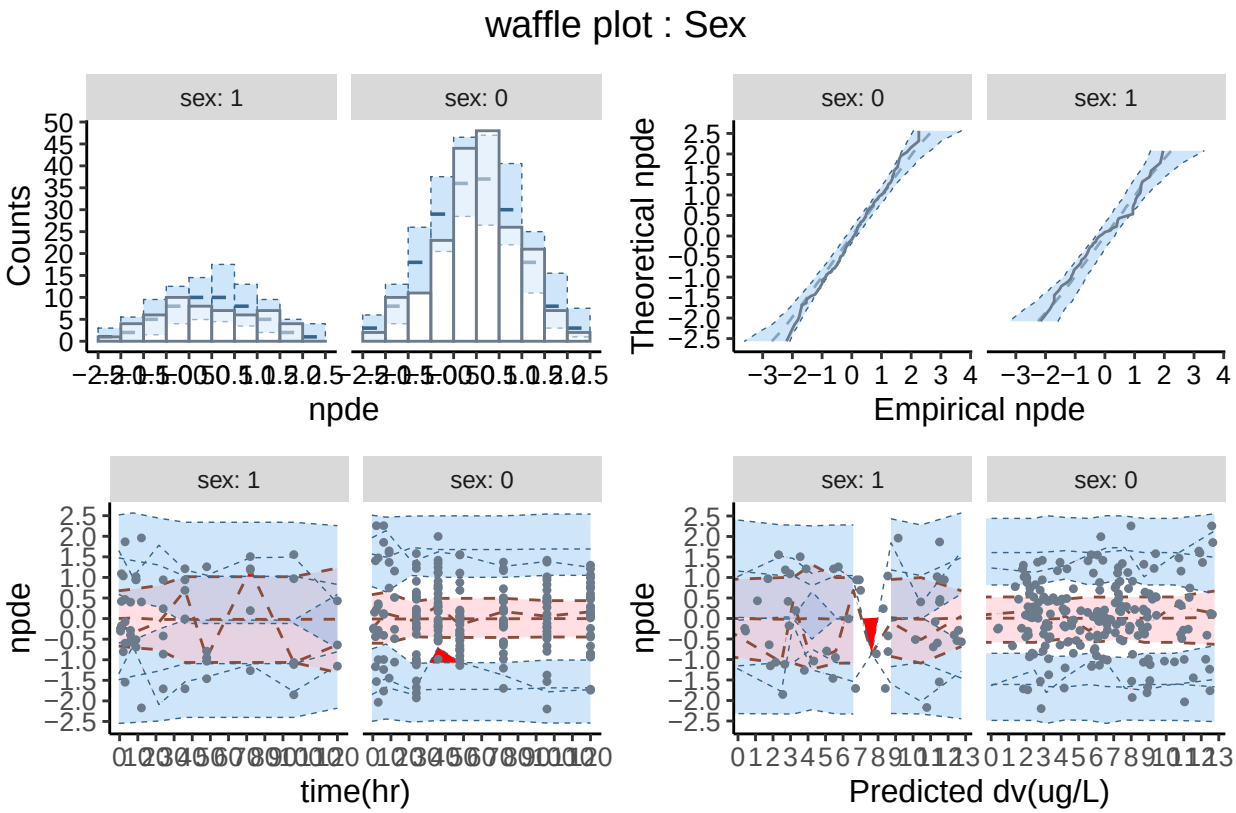

sex: 1
sex: 0
Counts
npde
sex: 0
sex: 1
Theoretical npde
Empirical npde
sex: 1
sex: 0
npde
time(hr)
sex: 1
sex: 0
npde
Predicted dv(ug/L)

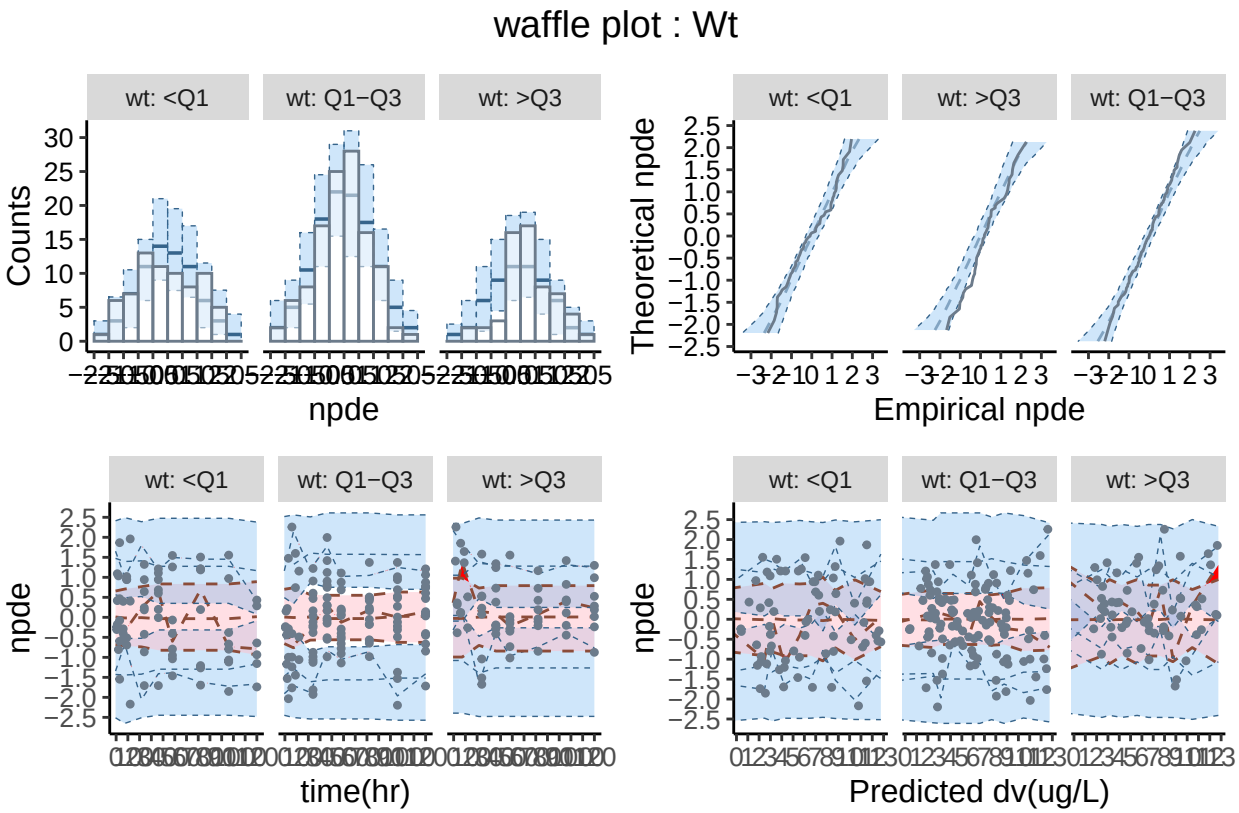

NA
wt: <Q1
wt: Q1−Q3
wt: >Q3
Counts
npde
Theoretical npde
Empirical npde
time(hr)
Predicted dv(ug/L)


sex: 1
sex: 0
Counts
npde
sex: 0
sex: 1
Theoretical npde
Empirical npde
sex: 1
sex: 0
npde
time(hr)
sex: 1
sex: 0
npde
Predicted dv(ug/L)

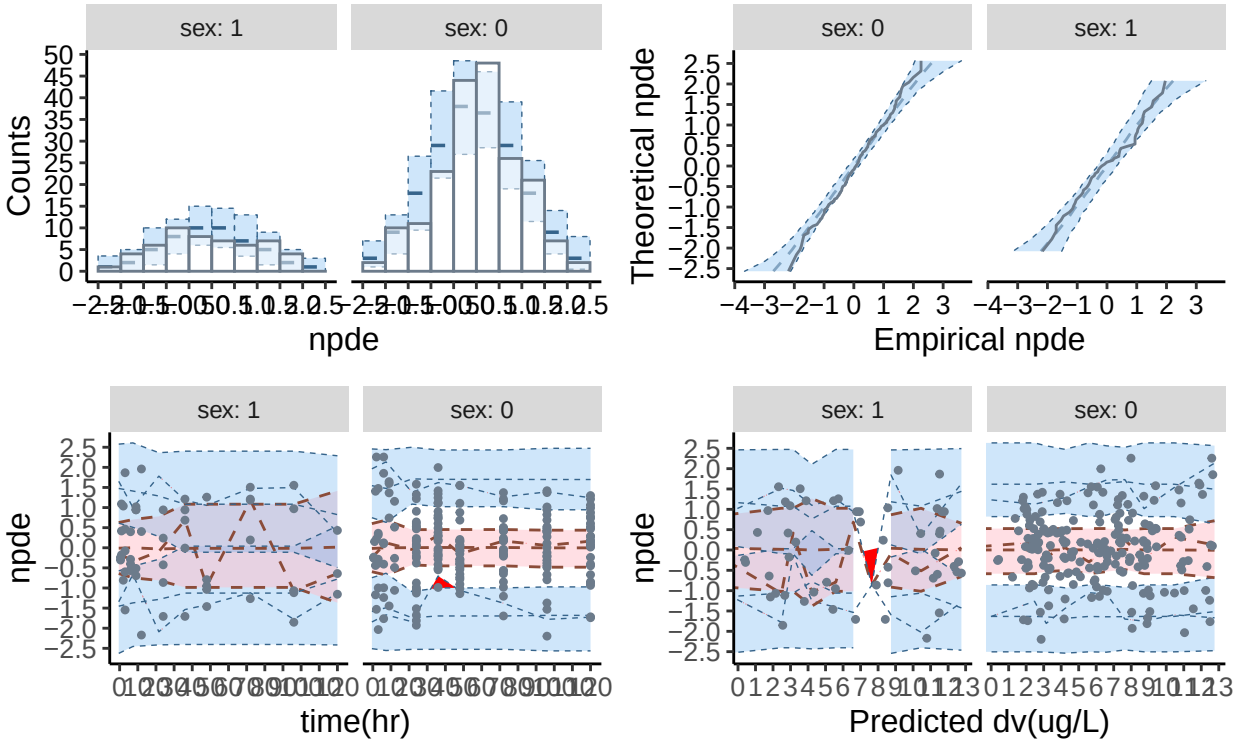

NA
wt: <Q1
wt: Q1−Q3
wt: >Q3
Counts
npde
Theoretical npde
Empirical npde
time(hr)
Predicted dv(ug/L)

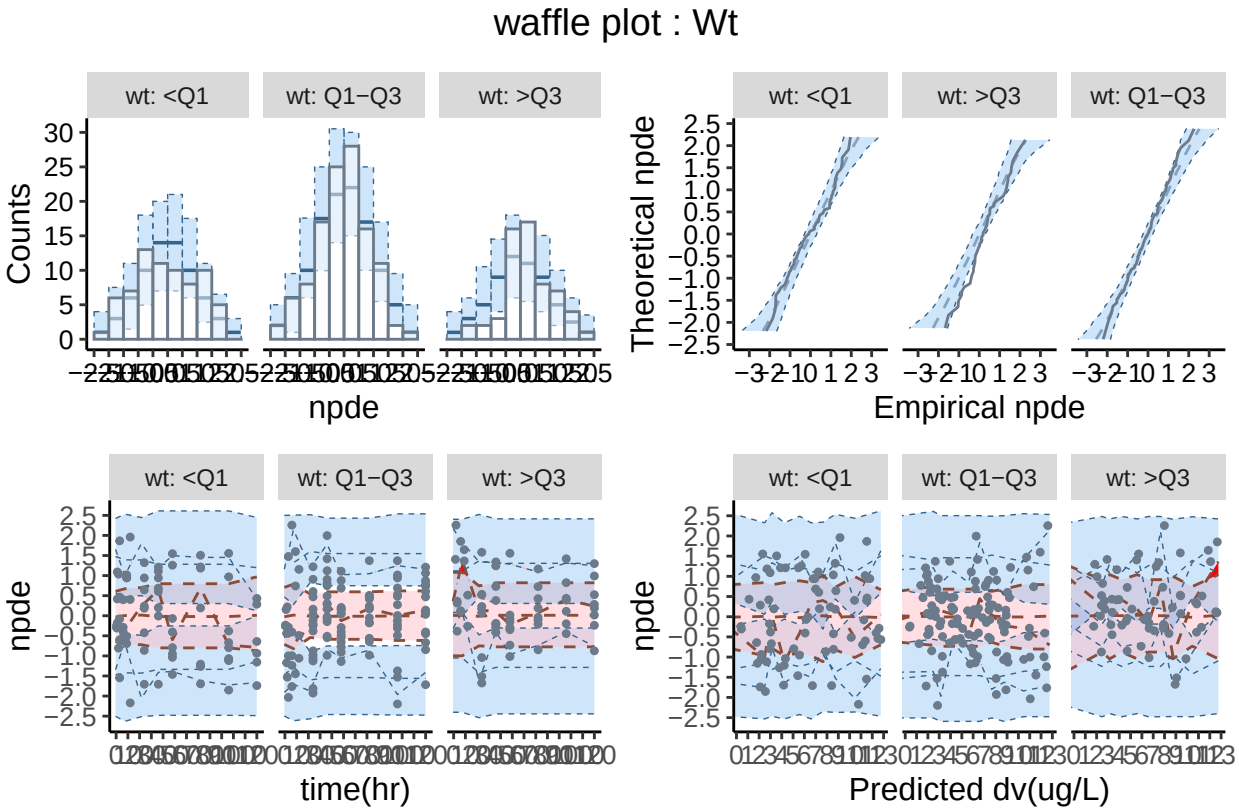

Counts
npde
Theoretical npde
Empirical npde
npde
time(hr)
npde
Predicted dv(ug/L)

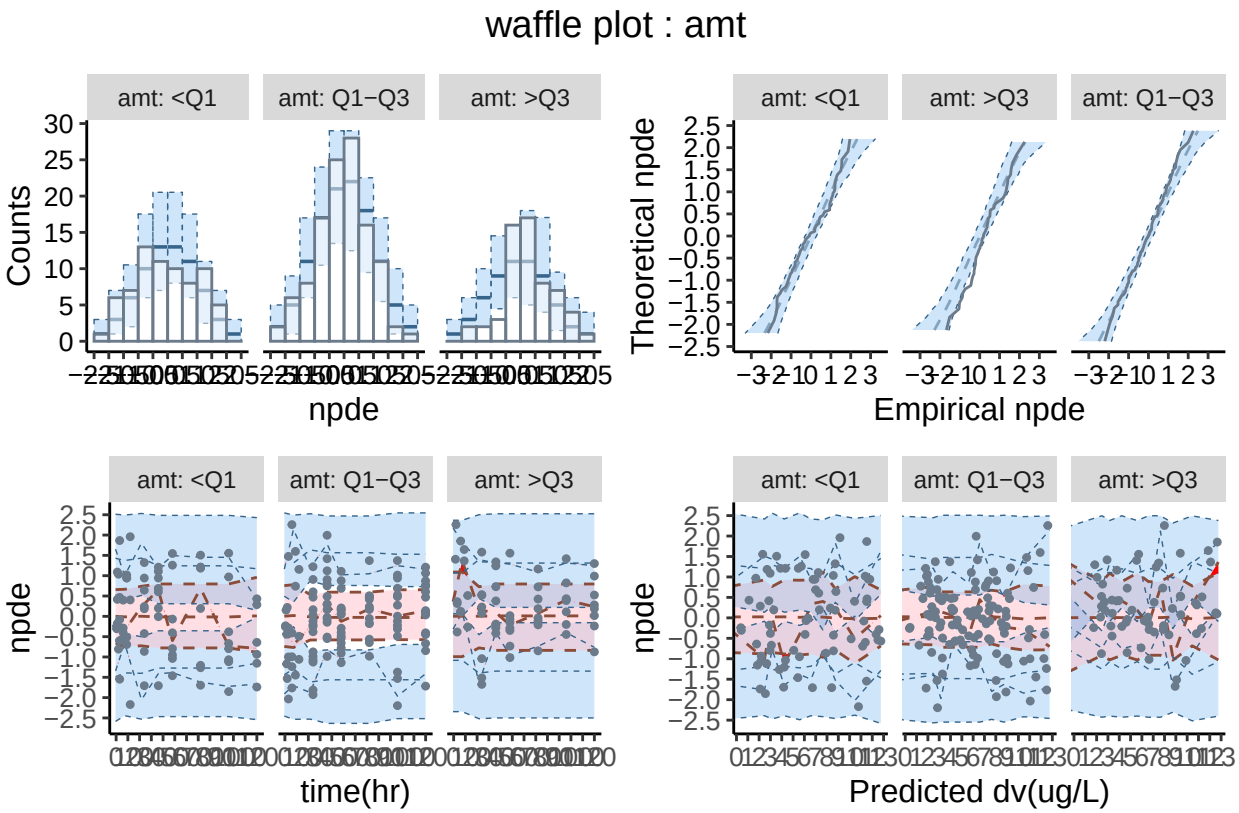

waffle plot : Sex
sex: 1
sex: 0
Counts
npde
Theoretical npde
Empirical npde
time(hr)
Predicted dv(ug/L)

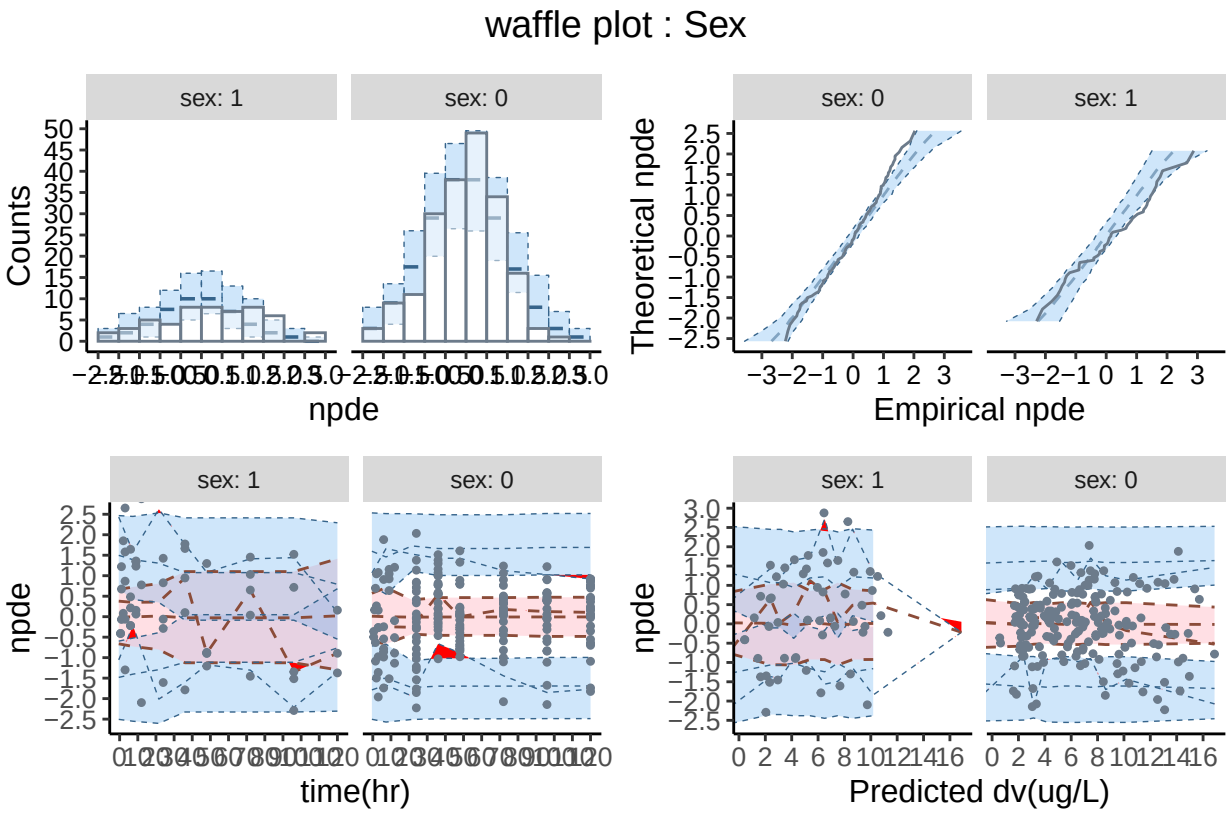

waffle plot : Wt
wt: <Q1
wt: Q1−Q3
wt: >Q3
Counts
npde
Theoretical npde
Empirical npde
time(hr)
Predicted dv(ug/L)

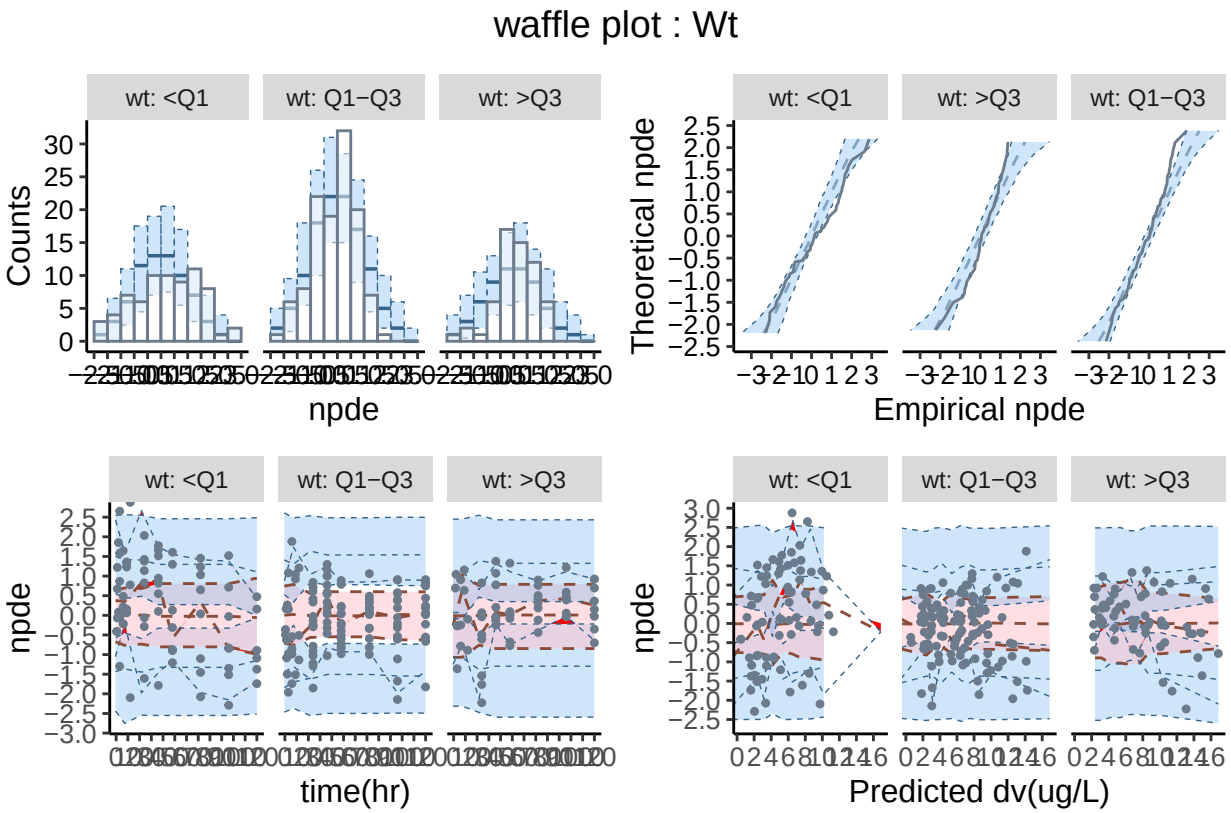
waffle plot : Sex

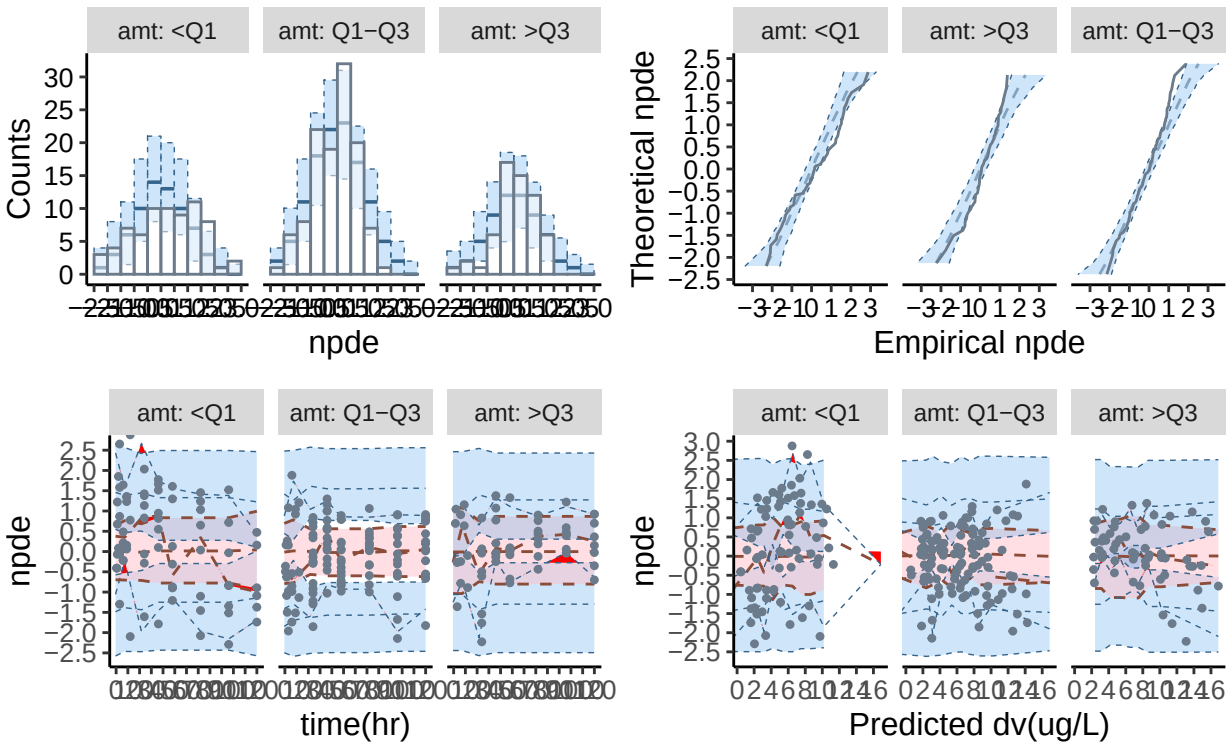


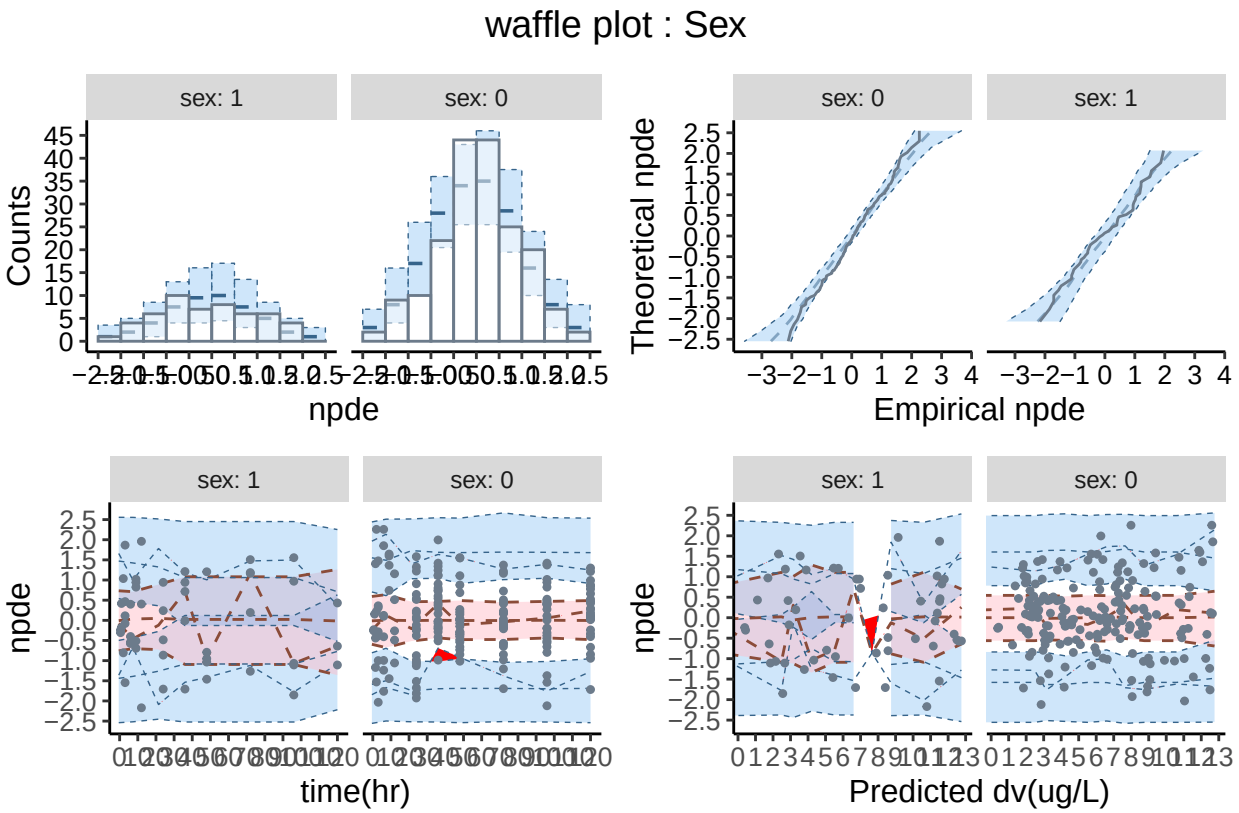

waffle plot : Wt
wt: <Q1
wt: Q1−Q3
wt: >Q3
Counts
npde
Theoretical npde
Empirical npde
time(hr)
Predicted dv(ug/L)

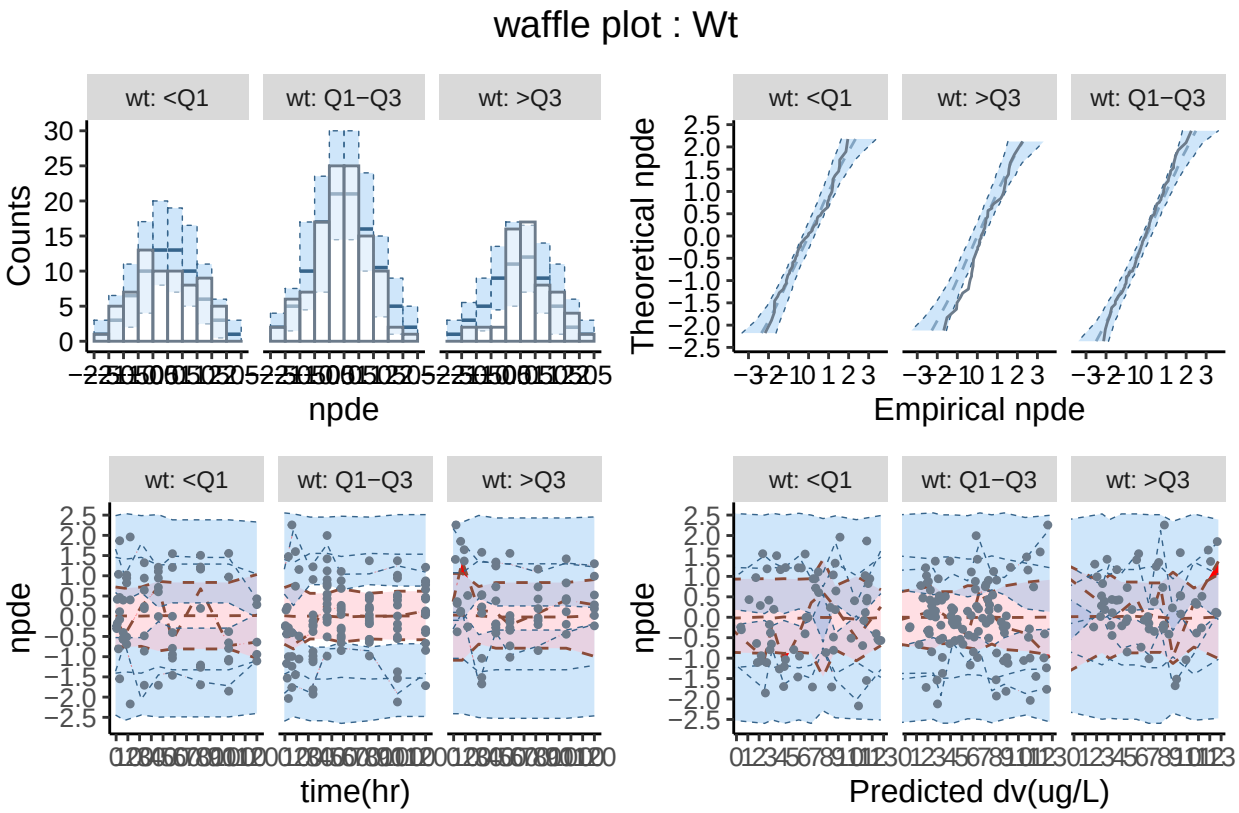

waffle plot : amt
amt: <Q1
amt: Q1−Q3
amt: >Q3
Counts
npde
Theoretical npde
Empirical npde
time(hr)
Predicted dv(ug/L)

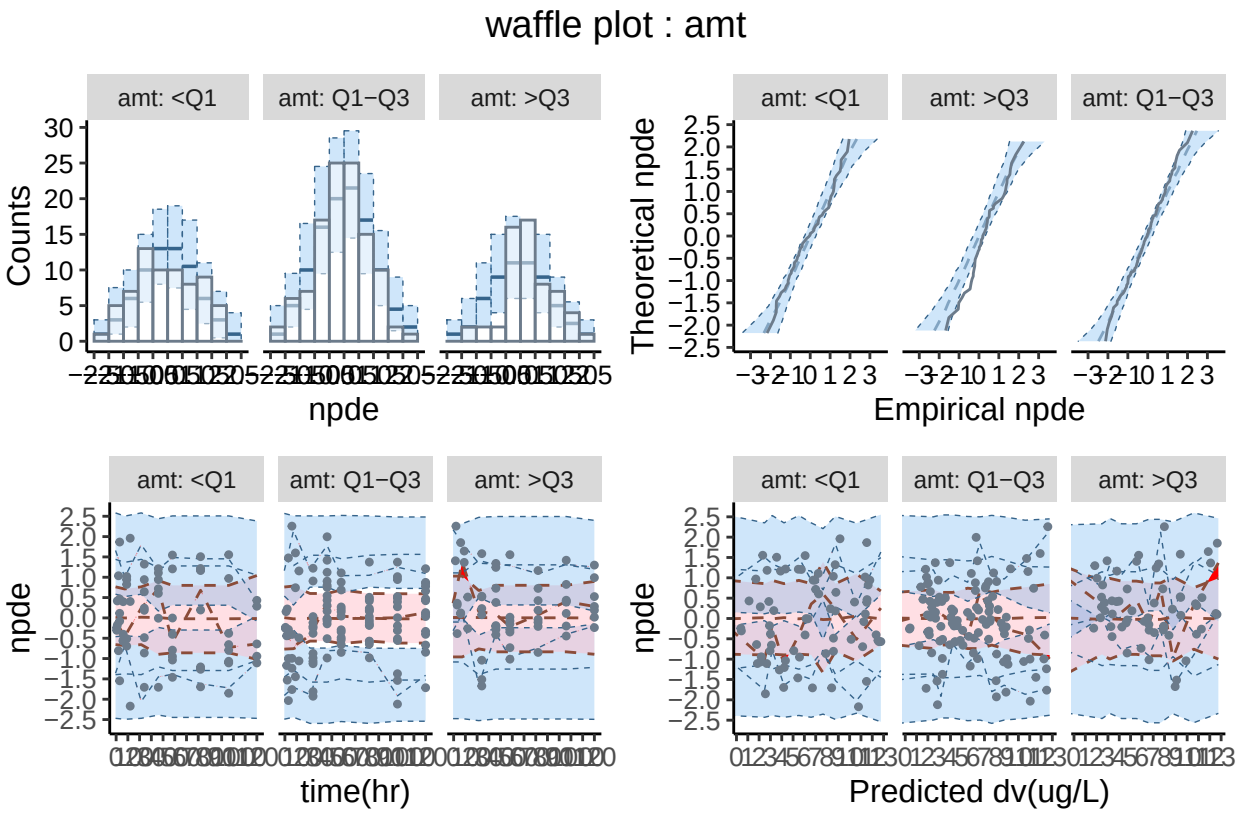

waffle plot : Sex
sex: 1
sex: 0
Counts
npde
Theoretical npde
Empirical npde
time(hr)
Predicted dv(ug/L)


waffle plot : Wt
wt: <Q1
wt: Q1−Q3
wt: >Q3
Counts
npde
Theoretical npde
Empirical npde
time(hr)
Predicted dv(ug/L)


waffle plot : amt
amt: <Q1
amt: Q1−Q3
amt: >Q3
Counts
npde
Theoretical npde
Empirical npde
time(hr)
Predicted dv(ug/L)

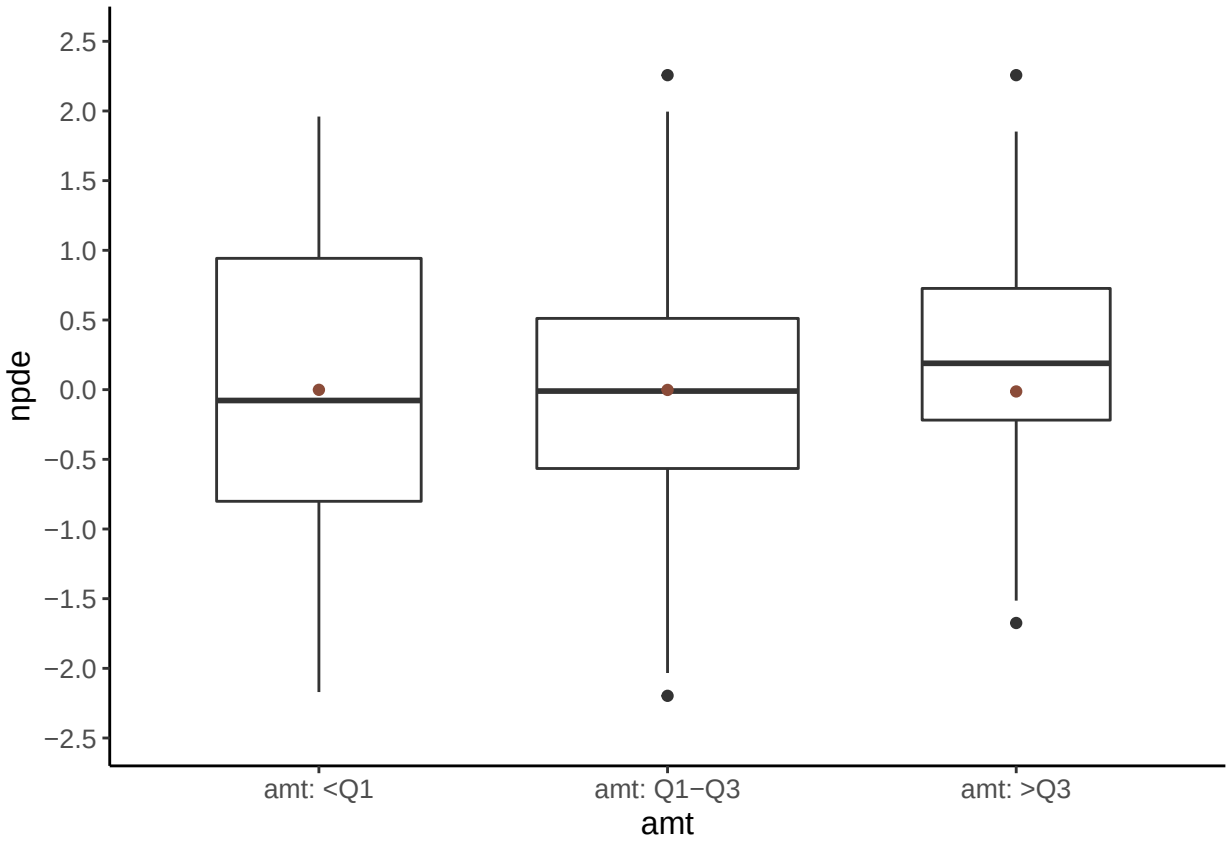

waffle plot : Sex
sex: 1
sex: 0
Counts
npde
Theoretical npde
Empirical npde
time(hr)
Predicted dv(ug/L)

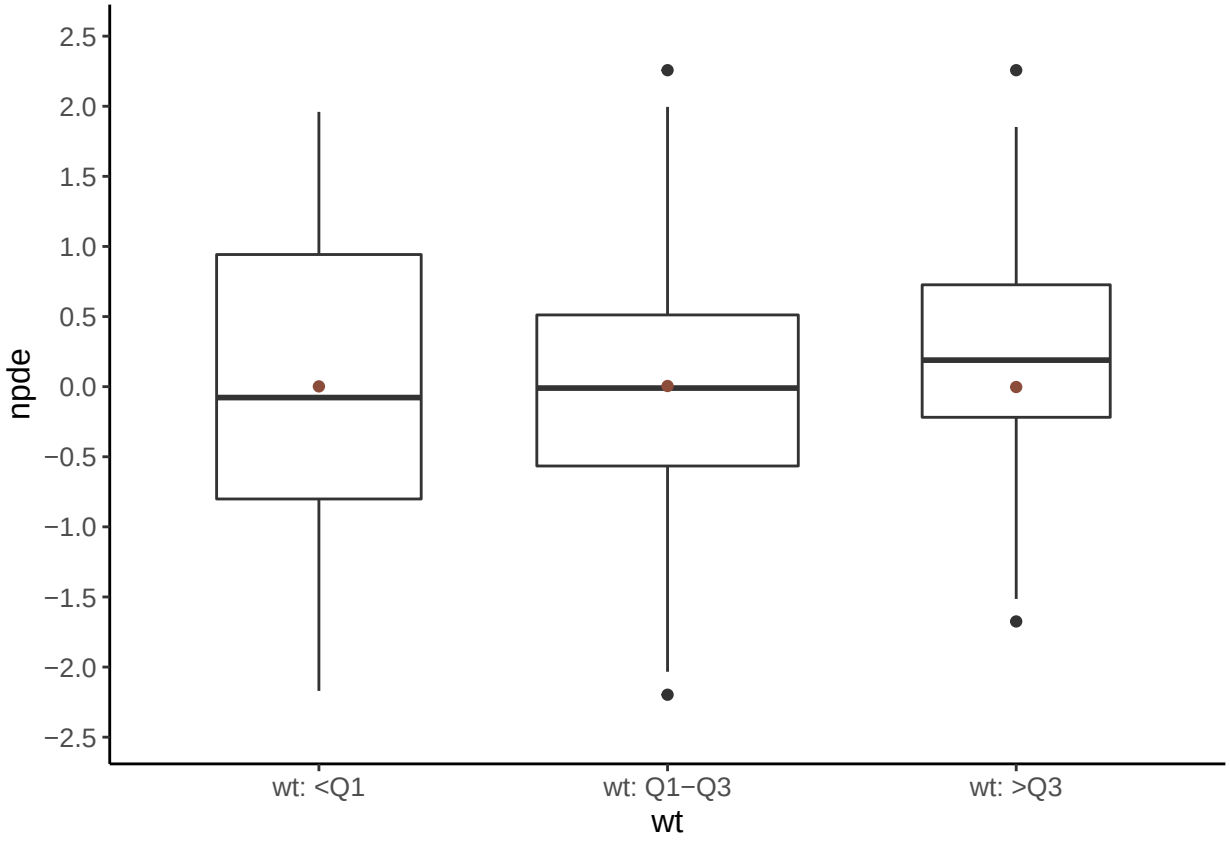

waffle plot : Wt
wt: <Q1
wt: Q1−Q3
wt: >Q3
Counts
npde
Theoretical npde
Empirical npde
time(hr)
Predicted dv(ug/L)

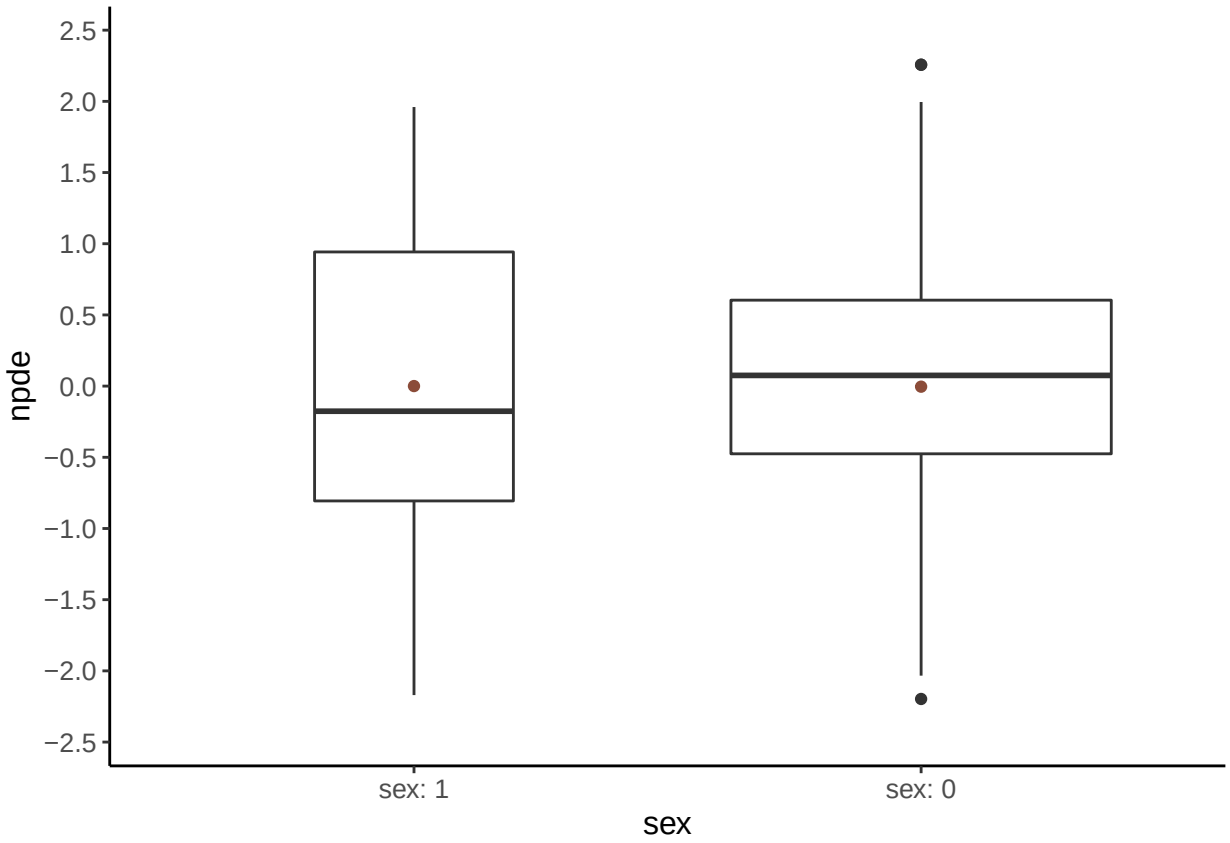


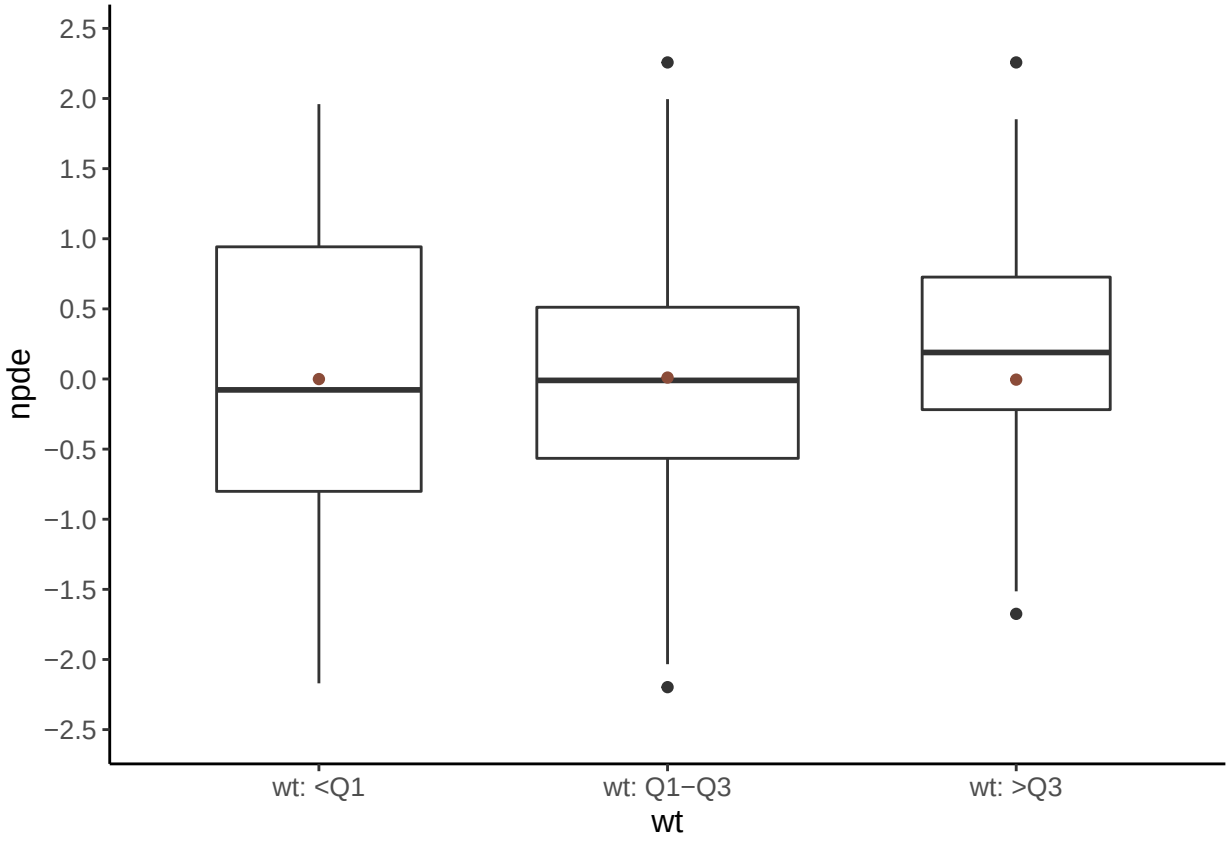
Parameters versus covariates

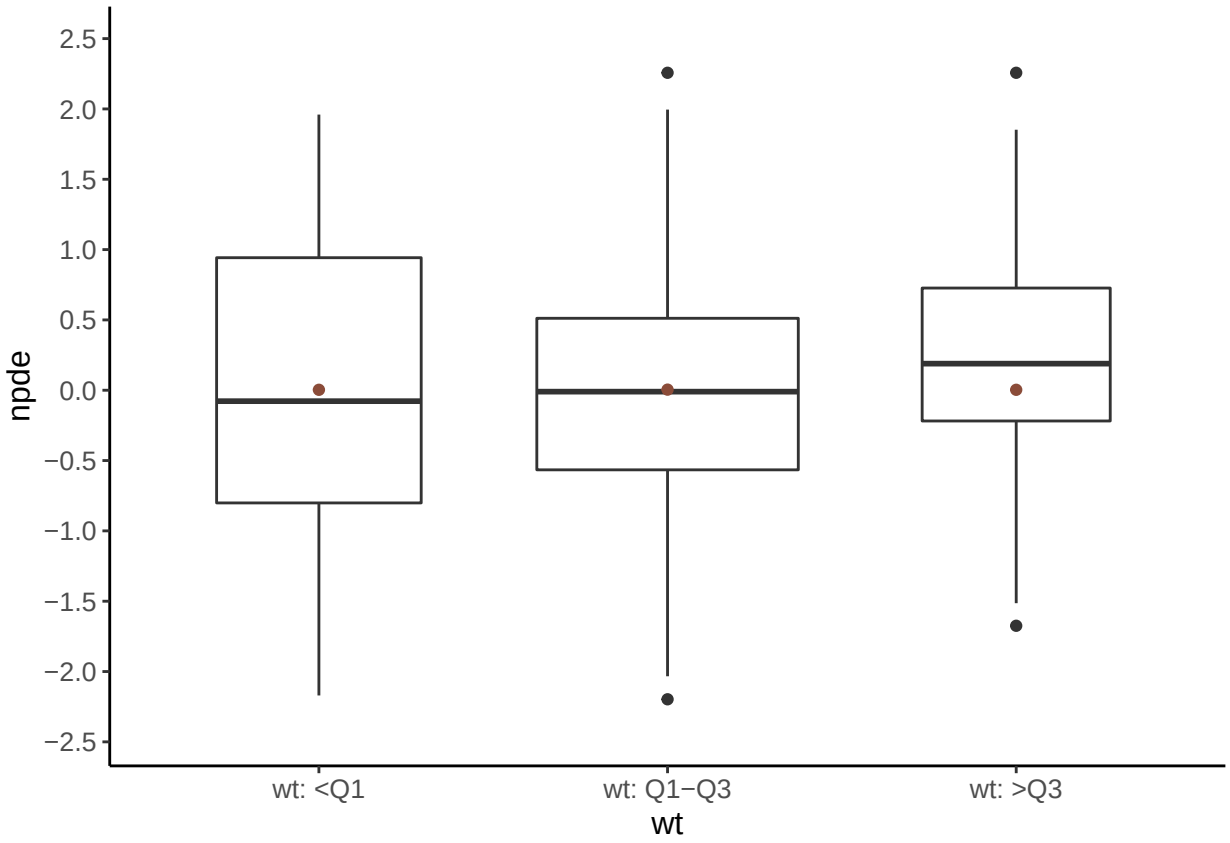
- **Problems** (à résoudre Romain)
  - **solved** yes here we really need to extend PI to cover the whole X-range for x.scatter and pred.scatter (not VPC)
    - **solved** just the PI (not the observed percentiles or outlier bands)
  - **solved** cov.scatter (eg npde versus covariates)
    - boxplot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


2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
npde
amt: <Q1
amt: Q1−Q3
amt: >Q3
amt

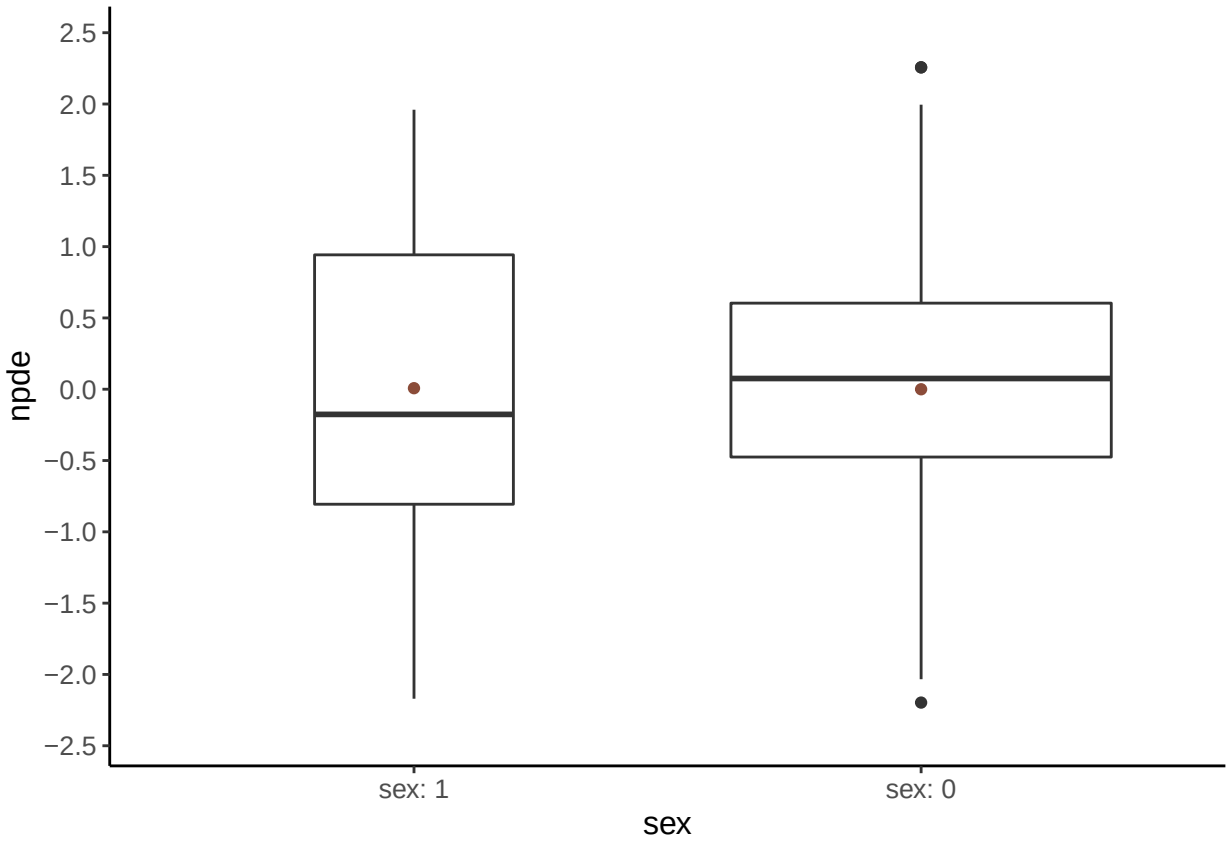

2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
npde
wt: <Q1
wt: Q1−Q3
wt: >Q3
wt

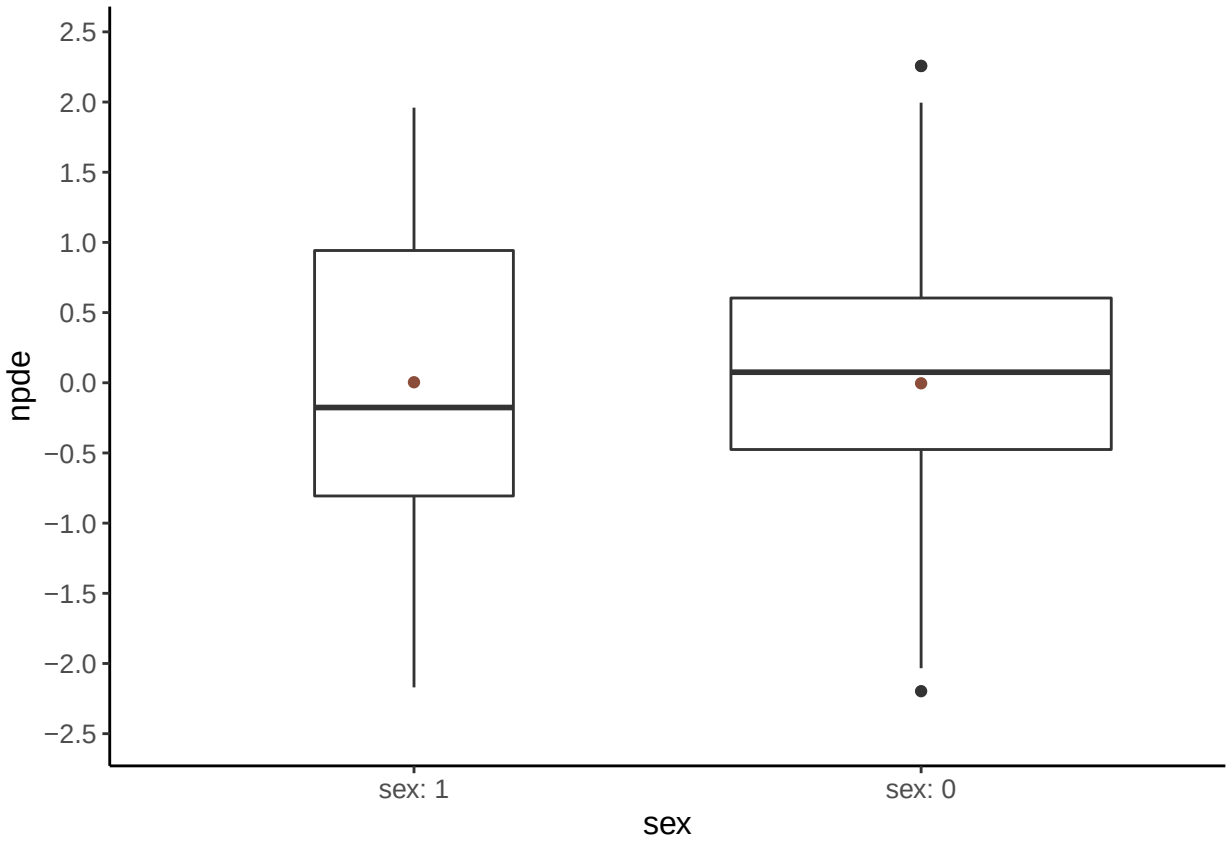

2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
npde
sex: 1
sex: 0
sex


2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
npde
wt: <Q1
wt: Q1−Q3
wt: >Q3
wt




```
## Test failed, not the expected behaviour
```


2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
npde
sex: 1
sex: 0
sex



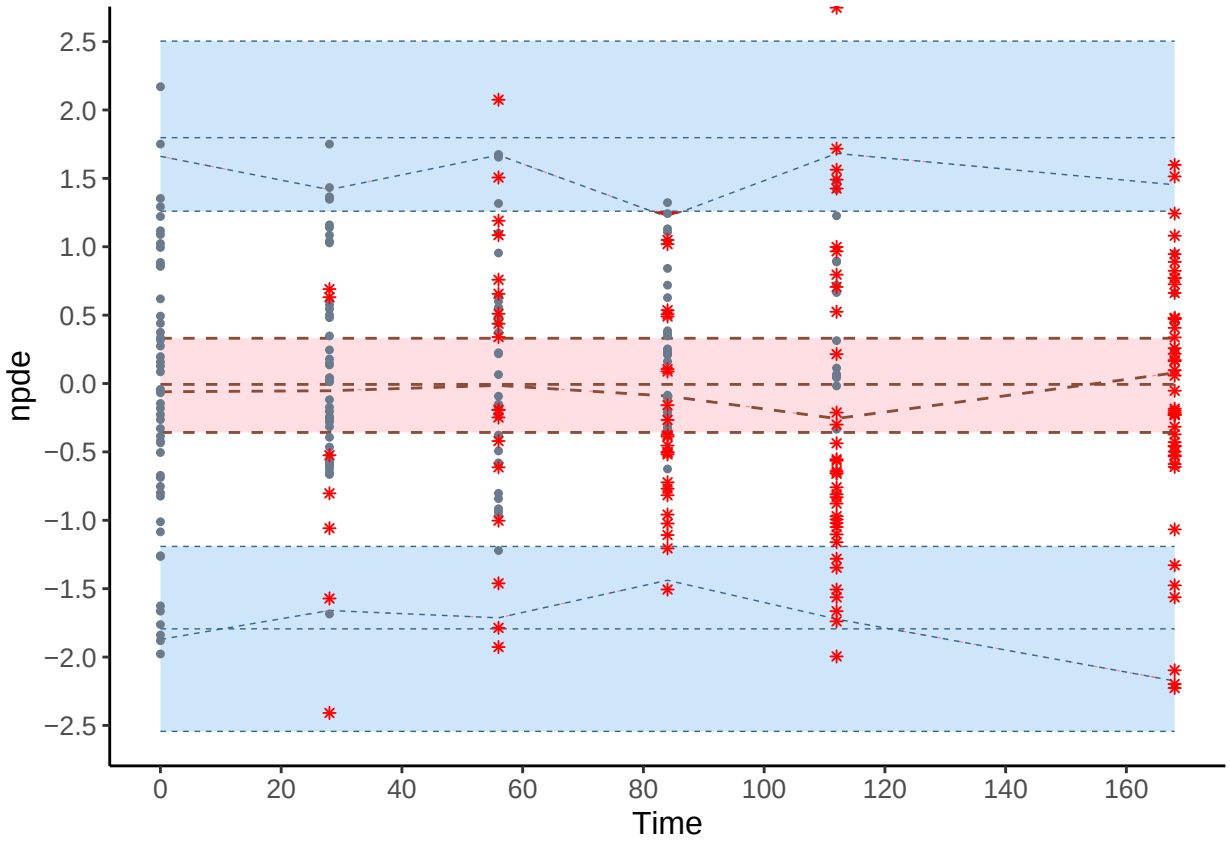
Creating the plots for the user guide

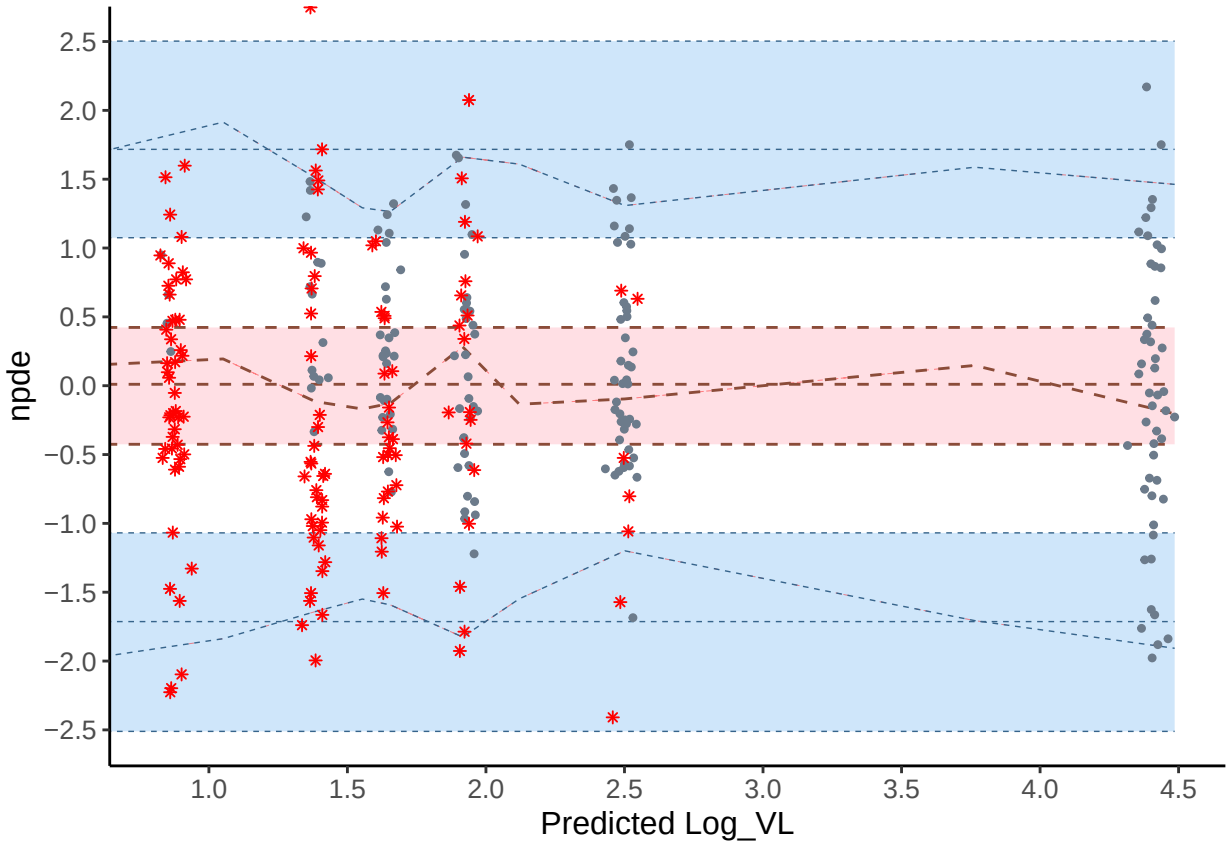
Theophylline

- 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) ?
  - **note**
  - Maintenant les foncions graphiques renvoient objet=NULL pour les plots, avant non. Peut-être faire remonter le return des objets ggplots jusque dans la fonction "plot" générale?
    * 2 graphes de données (voire 3)
    * 2 graphes de VPC côte à côte (=> 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


npde
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
0
20
40
60
80
100
120
140
160
Time




```
## Error in mapply(child_vp, vp_name = vpname(x$layout), t = .subset2(x$layout,  :
##   des entrées de longueur nulle ne peuvent être mélangés avec celles de longueur non nulle
```

End of file, deactivating development mode

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