nlmixr^2: Goodness of Fit plots using nlmixr^2

PSSN Conference 2023 workshop

Matthew Fidler

On behalf of the nlmixr2 development team:

Matt Fidler, Bill Denney, John Harrold, Richard Hooijmaijers, Rik Schoemaker, Max Taubert, Mirjam Trame, Theodoros Papathanasiou, Justin Wilkins, Yuan Xiong

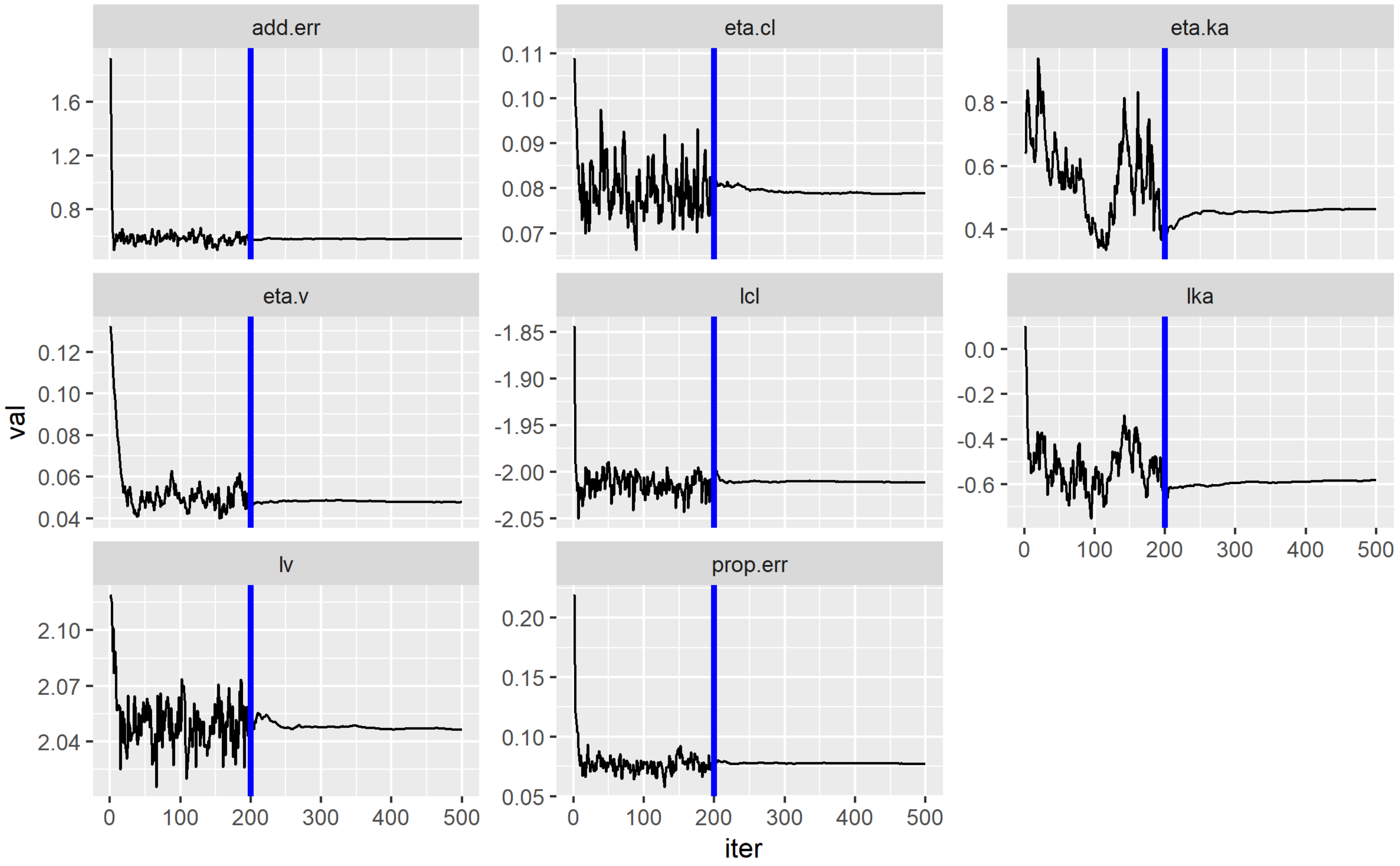
Standard Pharmacometrics goodness of fit plots (NPD or CWRES common, NPD in chart)

Plot	Shrinkage
Plot of normalized prediction distribution errors versus population predictions (NPD *vs.* EPRED)	
Plot of normalized prediction discrepancies versus time (NPD *vs.* TIME)	
Plot of individual weighted residuals versus individual predictions (IWRES *vs.* IPRED)	ε
Plot of individual weighted residuals versus time (IWRES *vs.* TIME)	ε
Visual predictive check (VPC)	
Distribution and quantile-quantile plot of IWRES	ε
Distribution and correlation structure of estimated inter-individual random effects (ETA)	η
Relationships between estimated inter-individual random effects (ETA) and covariates (before and after inclusion of covariates)	η
Plots of observations and model predictions per individual	ε
Plot of observations versus population predictions (DV *vs.* EPRED)	
Plot of observations versus individual predictions (DV *vs.* IPRED)	ε
Plot of absolute individual weighted residuals versus individual predictions (|IWRES| *vs.* IPRED)	ε

Running nlmixr^2 models: save the object, and examine parameter trace plots when using SAEM to check convergence

```r
## results are stored in the nlmixr object and can be viewed:
fitOne.comp.KA.solved_S

## and saved for future use or reference:
save(fitOne.comp.KA.solved_S, file = "fitOne.comp.KA.solved_S.Rdata")

## and for SAEM, convergence can be checked using a parameter trace plot:
traceplot(fitOne.comp.KA.solved_S)
```

Traceplot for SAEM parameter estimates using `traceplot` command

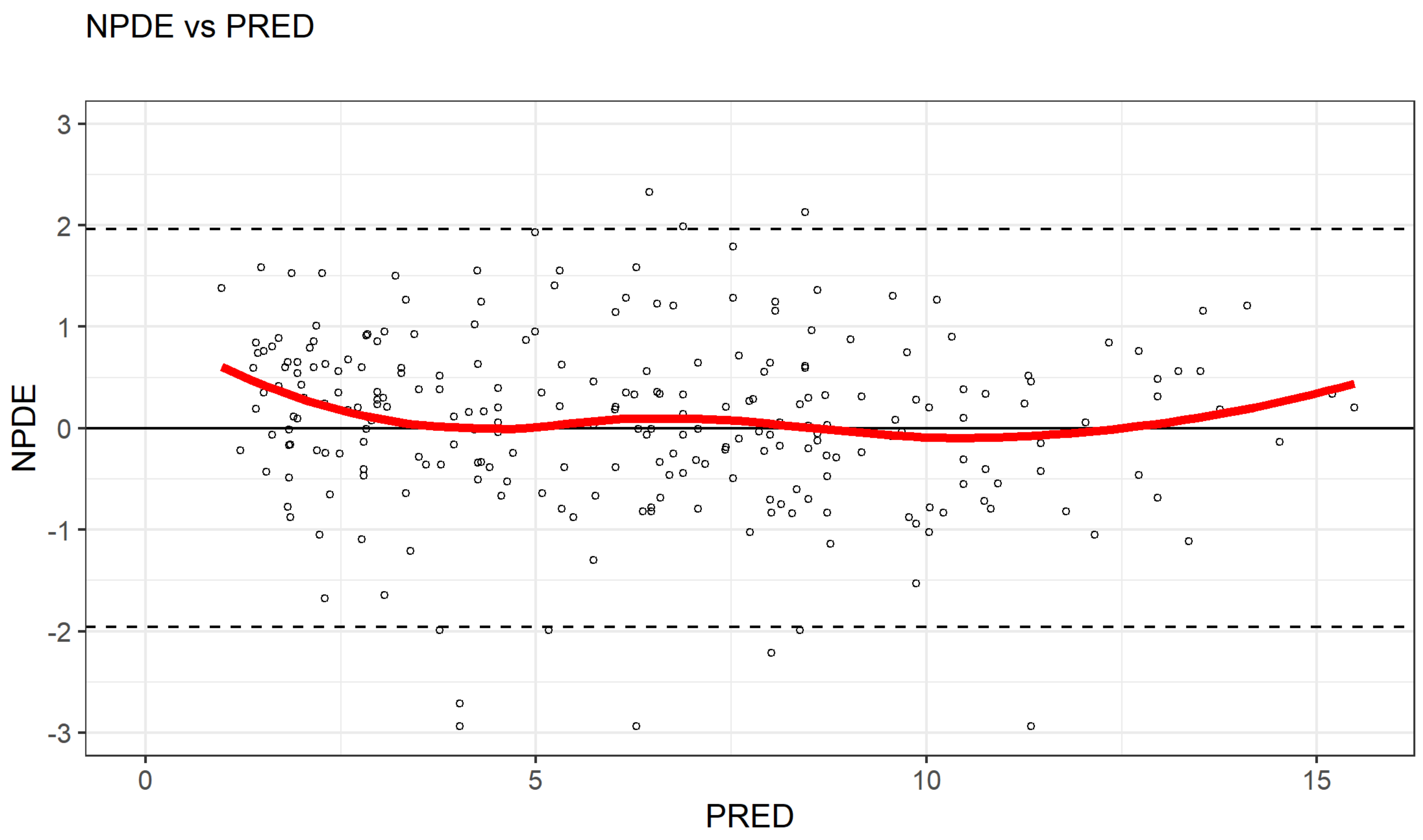


nlmixr^2 is linked to ggPMX

```r
# Model Diagnostics with ggPMX
# The controller is first constructed
ctr <- pmx_nlmixr(fitOne.comp.KA.solved_S,
                  conts = c("WT","AGE"),
                  cats=c("SEX","SPARSE"),
                  vpc=FALSE,
                  settings=pmx_settings(is.draft=FALSE))

# and can then be piped into a specific plot
# syntax for npde vs pred plot
ctr %>% pmx_plot_npde_pred
#alternatively:
pmx_plot_npde_pred(ctr)
```

nlmixr2 is linked to ggPMX

```
# Model Diagnostics with ggPMX
# The controller is first constructed
ctr <- pmx_nlmixr(fitOne.comp.KA.solved_S,
                  conts = c("WT","AGE"),
                  cats=c("SEX","SPARSE"),
                  vpc=FALSE,
       settings=pmx_settings(is.draft=FALSE))

# and can then be piped into a specific plot
# syntax for npde vs pred plot
ctr %>% pmx_plot_npde_pred
#alternatively:
pmx_plot_npde_pred(ctr)
```

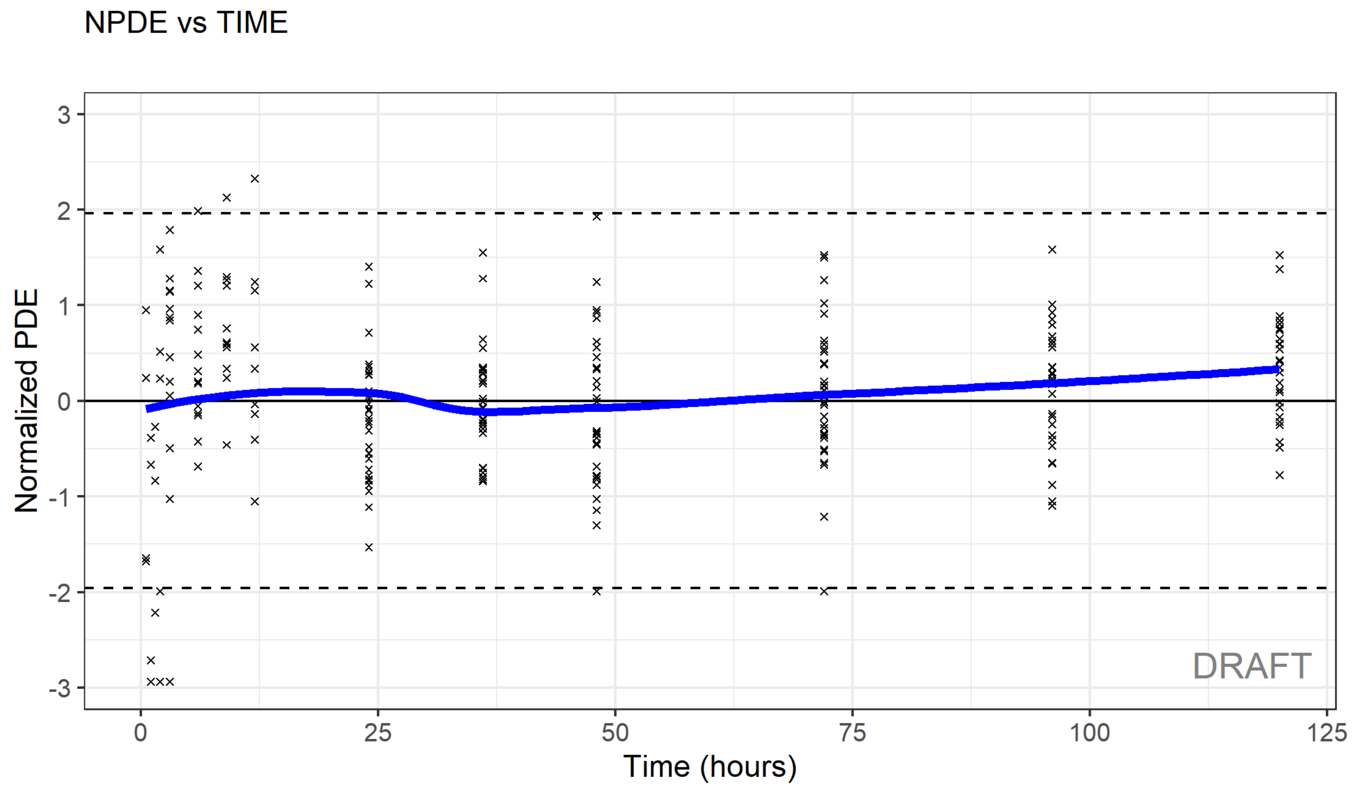


nlmixr^2 is linked to ggPMX

```
## Modify graphical options and add DRAFT label:
ctr %>% pmx_plot_npde_time(smooth = list(color="blue"),
                           point = list(shape=4),
                           s.draft=TRUE,
                           labels = list(x = "Time (hours)",
                                         y = "Normalized PDE"))
```

nlmixr^2 is linked to ggPMX

```r
## Modify graphical options and add DRAFT label:
ctr %>% pmx_plot_npde_time(smooth = list(color="blue"),
                           point = list(shape=4),
                           s.draft=TRUE,
                           labels = list(x = "Time (hours)",
                                         y = "Normalized PDE"))
```



nlmixr2 is linked to ggPMX

```
## DV vs IPRED plot
ctr %>% pmx_plot_dv_ipred(scale_x_log10=TRUE, scale_y_log10=TRUE)

#You can filter to restrict the values of IPRED for instance:
ctr %>% pmx_plot_dv_ipred(scale_x_log10=TRUE, scale_y_log10=TRUE,filter=(IPRED>1))

## DV vs PRED plot
ctr %>% pmx_plot_dv_pred(scale_x_log10=TRUE, scale_y_log10=TRUE)

## Absolute individual weighted residuals to investigate the residual error model
ctr %>% pmx_plot_abs_iwres_ipred
#again, alternatively:
#pmx_plot_abs_iwres_ipred(ctr)


ctr %>% pmx_plot_iwres_dens
ctr %>% pmx_plot_eta_qq
ctr %>% pmx_plot_eta_box
ctr %>% pmx_plot_eta_hist
ctr %>% pmx_plot_eta_matrix
```

nlmixr2 is linked to ggPMX

```
## generate a full report with diagnostics

ctr %>% pmx_report(name="ggPMX_report",
                   save_dir=".",
                   format="report",
                   extension="word")
```

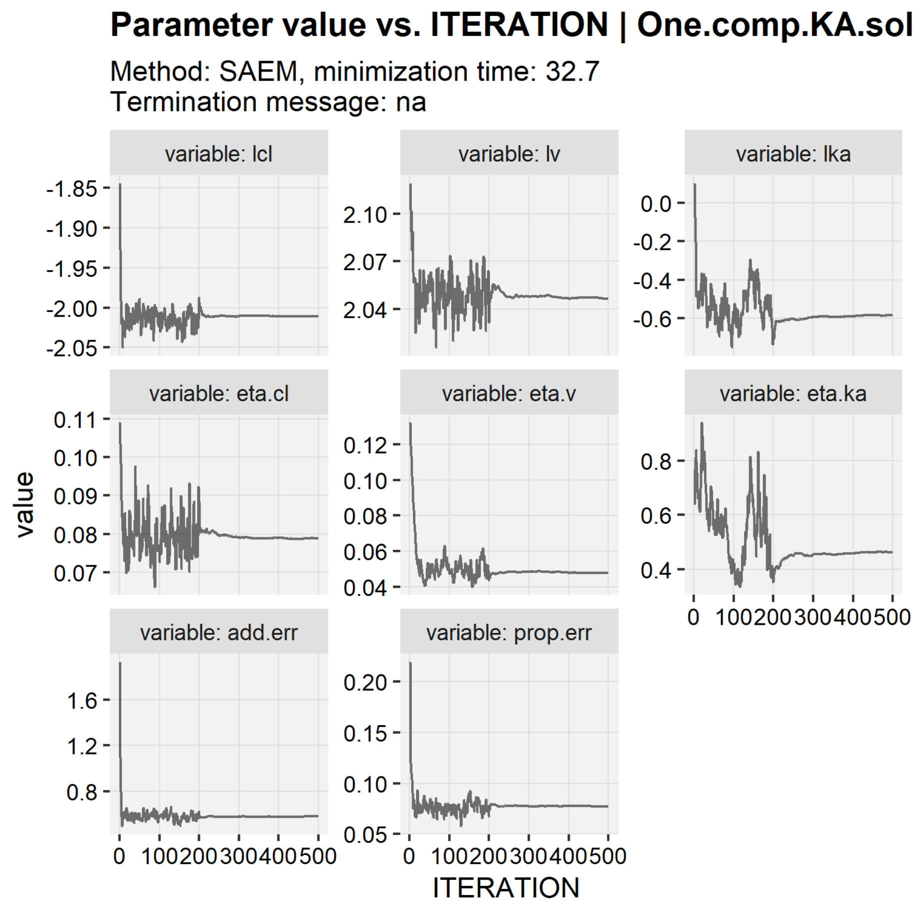
nlmixr^2 is linked to Ben Guiastrennec's xpose* package that uses ggplot2

```r
## the nlmixr object can be transformed into an xpose object to allow diagnostics with the new xpose package
## the link between nlmixr and xpose is provided by the xpose.nlmixr package
## only xpose_data_nlmixr is from xpose.nlmixr
## all further commands (see cheatsheet)  are from the xpose package

xpdb.1s <- xpose_data_nlmixr(fitOne.comp.KA.solved_S)

## this can also be used to generate trace plots (parameters vs iterations:)
prm_vs_iteration(xpdb.1s)
## to remove the path to the script from the plot use:
prm_vs_iteration(xpdb.1s,caption=NULL)
```

*https://uupharmacometrics.github.io/xpose/

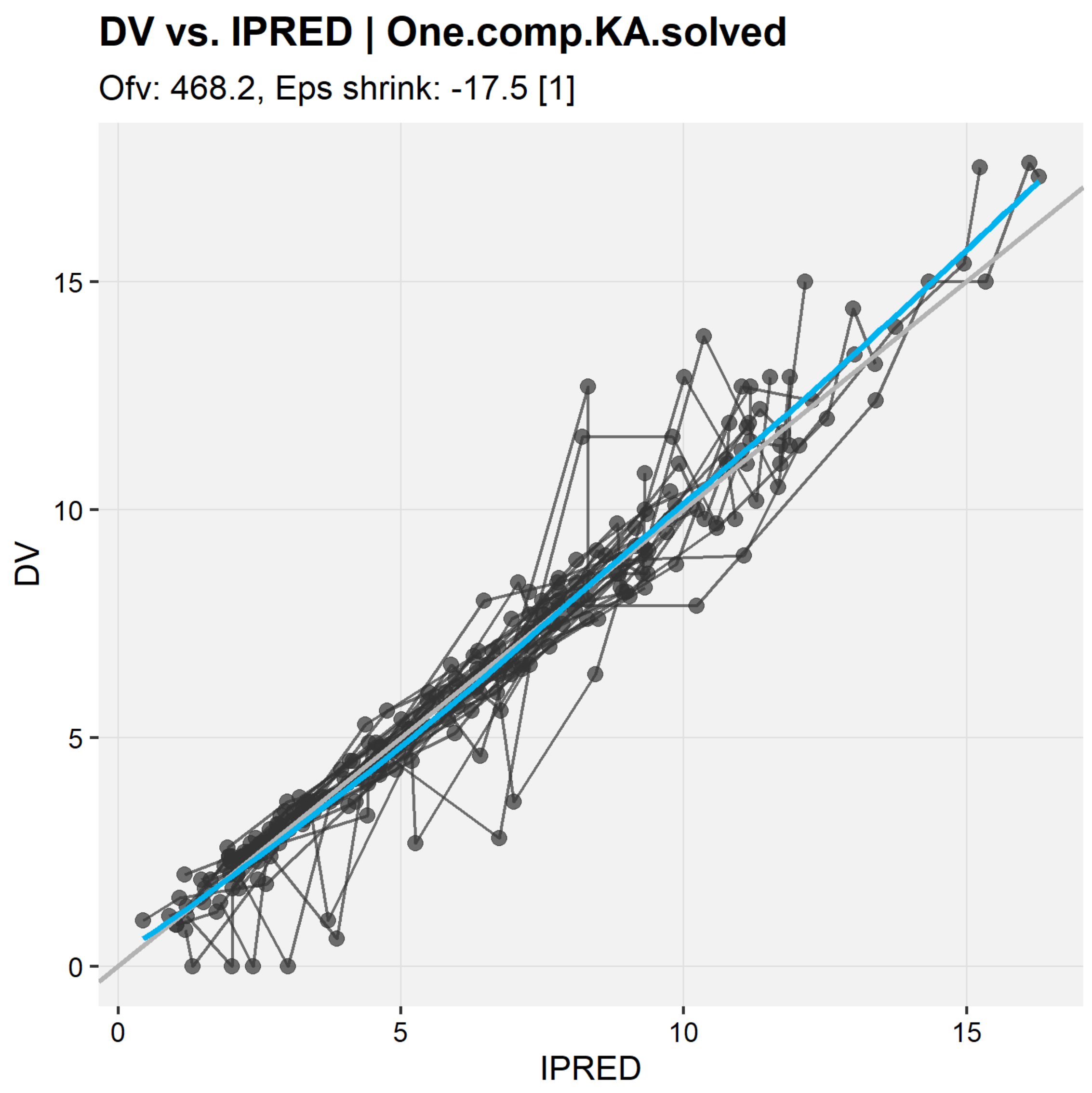
nlmixr2 is linked to Ben Guiastrennec's xpose* package that uses ggplot2

```
## the nlmixr object can be transformed into an xpose object to allow diagnostics with the new xpose package
## the link between nlmixr and xpose is provided by the xpose.nlmixr package
## only xpose_data_nlmixr is from xpose.nlmixr
## all further commands (see cheatsheet)  are from the xpose package

xpdb.1s <- xpose_data_nlmixr(fitOne.comp.KA.solved_S)

## this can also be used to generate trace plots (parameters vs iterations:)
prm_vs_iteration(xpdb.1s)
## to remove the path to the script from the plot use:
prm_vs_iteration(xpdb.1s,caption=NULL)
```



*https://uupharmacometrics.github.io/xpose/

DV vs IPRED using xpose



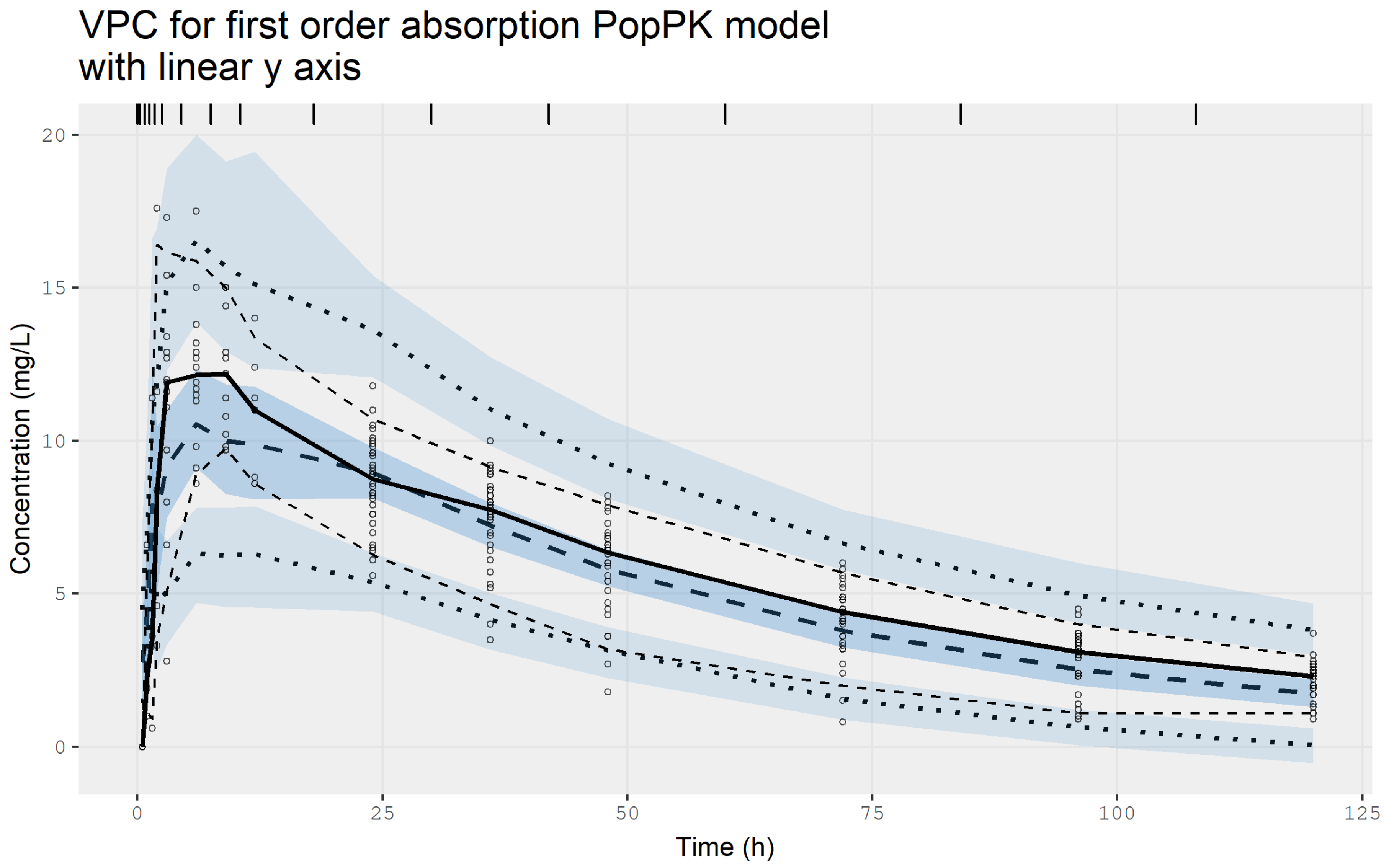
```
xpdb.1s <- xpose_data_nlmixr(fitOne.comp.KA.solved_F)
## dv vs ipred plot:
dv_vs_ipred(xpdb.1s,
            caption = NULL)
```

nlmixr2 is linked to Ron Keizer's vpc* package

```
## nlmixr comes with its own built-in vpc functionality that uses Ron Keizer's vpc package
## see the cheatsheet for further options

## because the data set uses nominal time points, it is nice to have the bins surround these time points
## so that each time point falls in a bin

bin_mids <- sort(unique(PKdata$TIME))
bin_edges <- bin_mids - c(0, diff(bin_mids) / 2)

vpcPlot(
  fitOne.comp.KA.solved_S,        #the nlmixr object
  n = 500,                        #number of trials simulated
  bins = bin_edges,
  show = list(obs_dv = TRUE,      #additional items to show, like the observations
              obs_median = TRUE,
              sim_median = TRUE,
              sim_median_ci = TRUE,
              obs_ci = TRUE,
              pi = TRUE
  ),
  xlab = "Time (h)",              #x-axis label
  ylab = "Concentration (mg/L)",  #y-axis label
  title = "VPC for first order absorption PopPK model"
)
```

*http://vpc.ronkeizer.com/

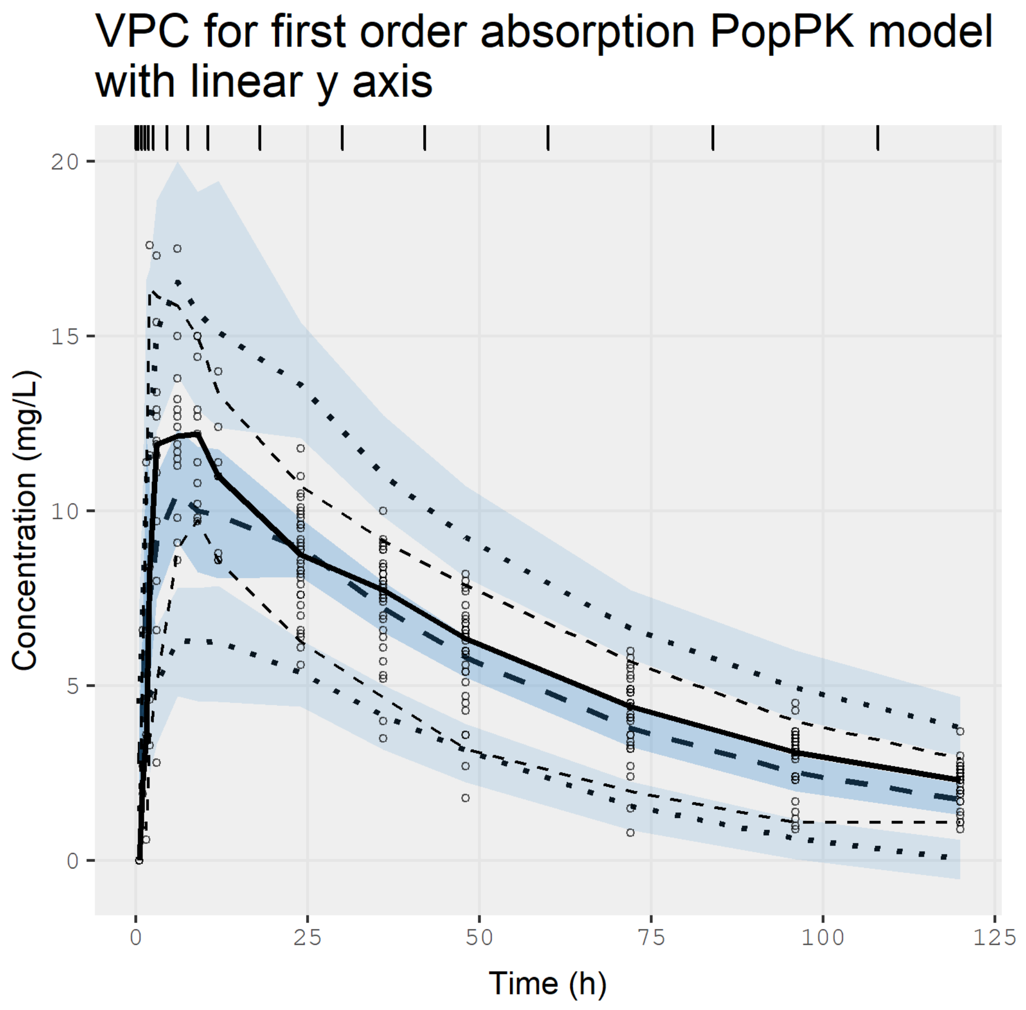
VPC for the base model on linear scale...

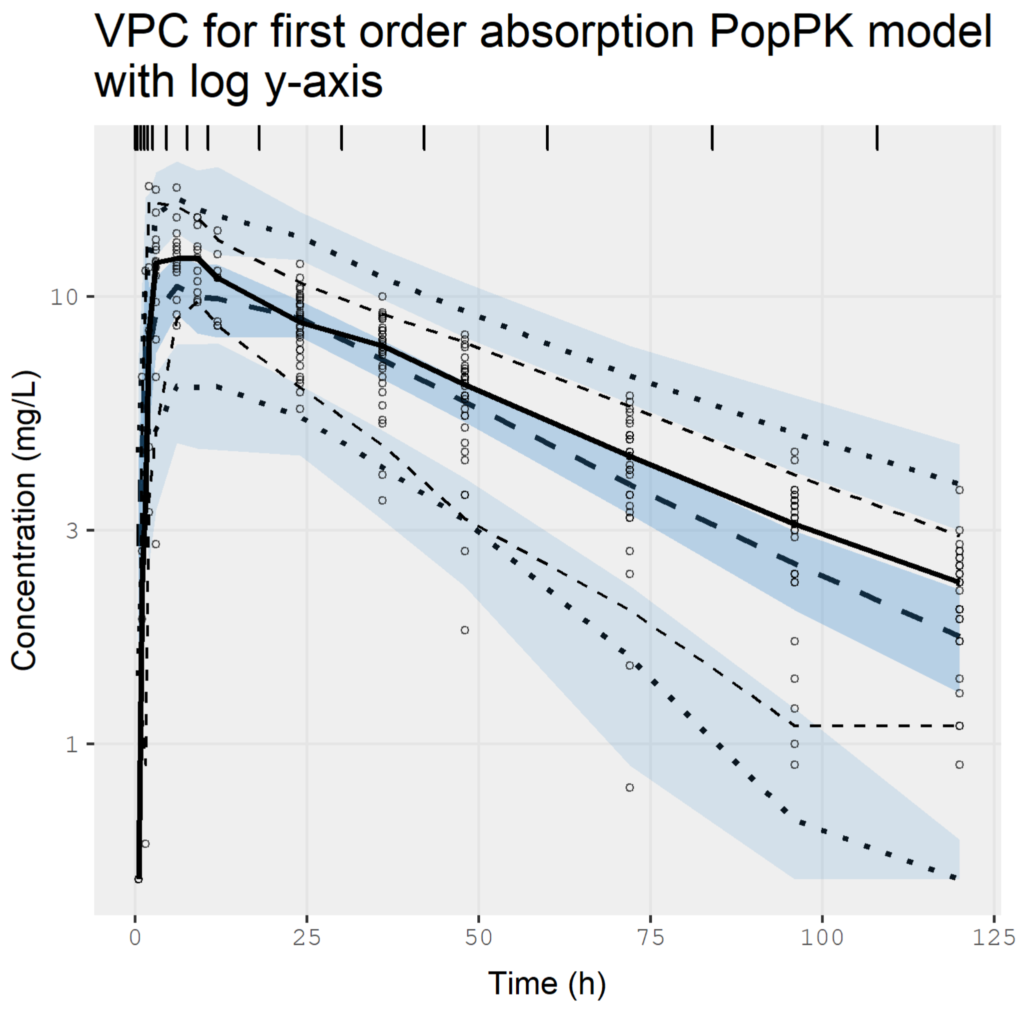


...and on log scale

```r
## or with a log y-axis starting at 0.5
vpcPlot(
  fitOne.comp.KA.solved_S,        #the nlmixr object
  n = 500,                        #number of trials simulated
  bins = bin_edges,
  show = list(obs_dv = TRUE,      #additional items to show, like the observations
              obs_median = TRUE,
              sim_median = TRUE,
              sim_median_ci = TRUE,
              obs_ci = TRUE,
              pi = TRUE
  ),
  xlab = "Time (h)",              #x-axis label
  ylab = "Concentration (mg/L)",  #y-axis label
  title = "VPC for first order absorption PopPK model"
  log_y = TRUE,            #to request a log y-axis
  log_y_min = 0.5          #starting at 0.5
)
```

...and on log scale. It's super fast ☺



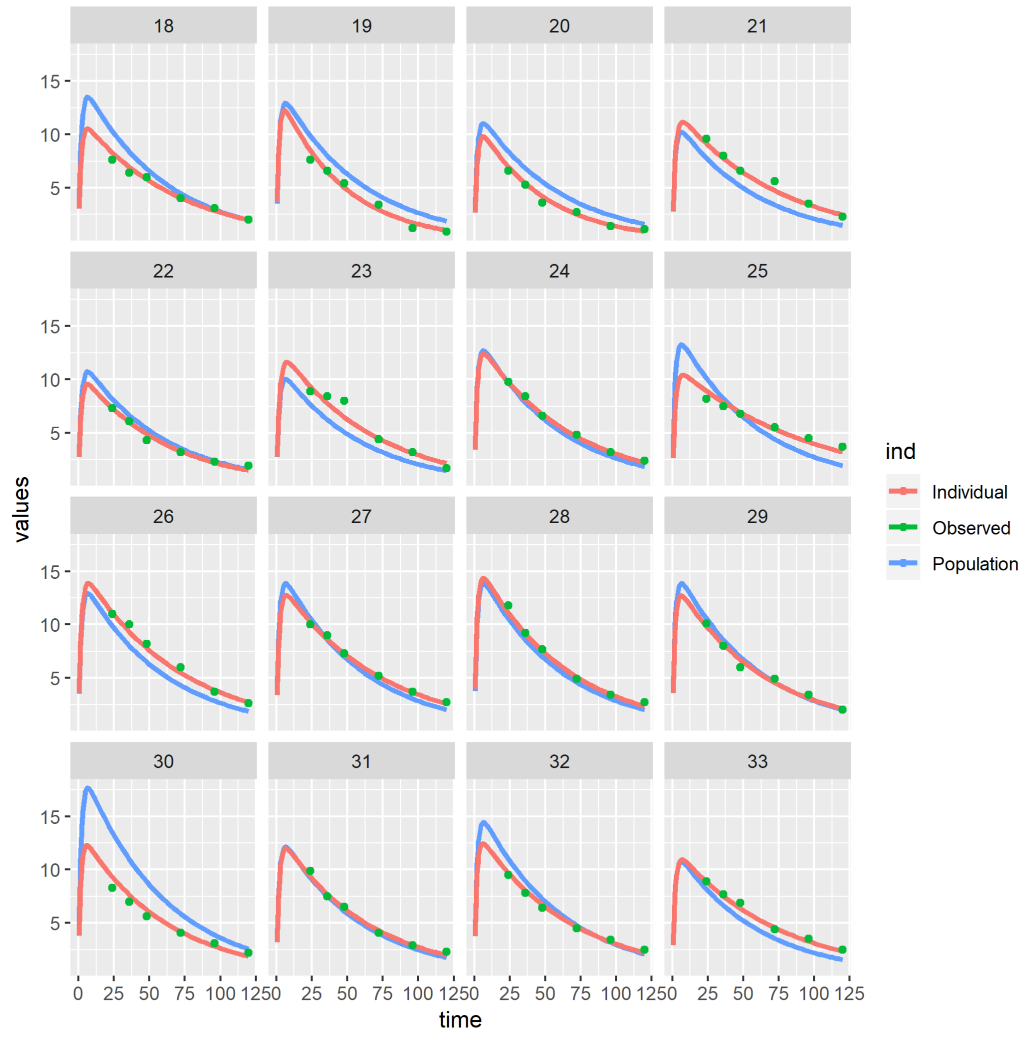


nlmixr^2 can generate individual graphs using `augPred`

```
## Individual fits can be generated using augPred (augmented predictions)
## that provides smooth profiles by interpolating the predictions between observations:
plot(augPred(fitOne.comp.KA.solved_S))
## ...use the arrows in the plot window to examine the earlier curves
```

nlmixr^2 can generate individual graphs using `augPred`

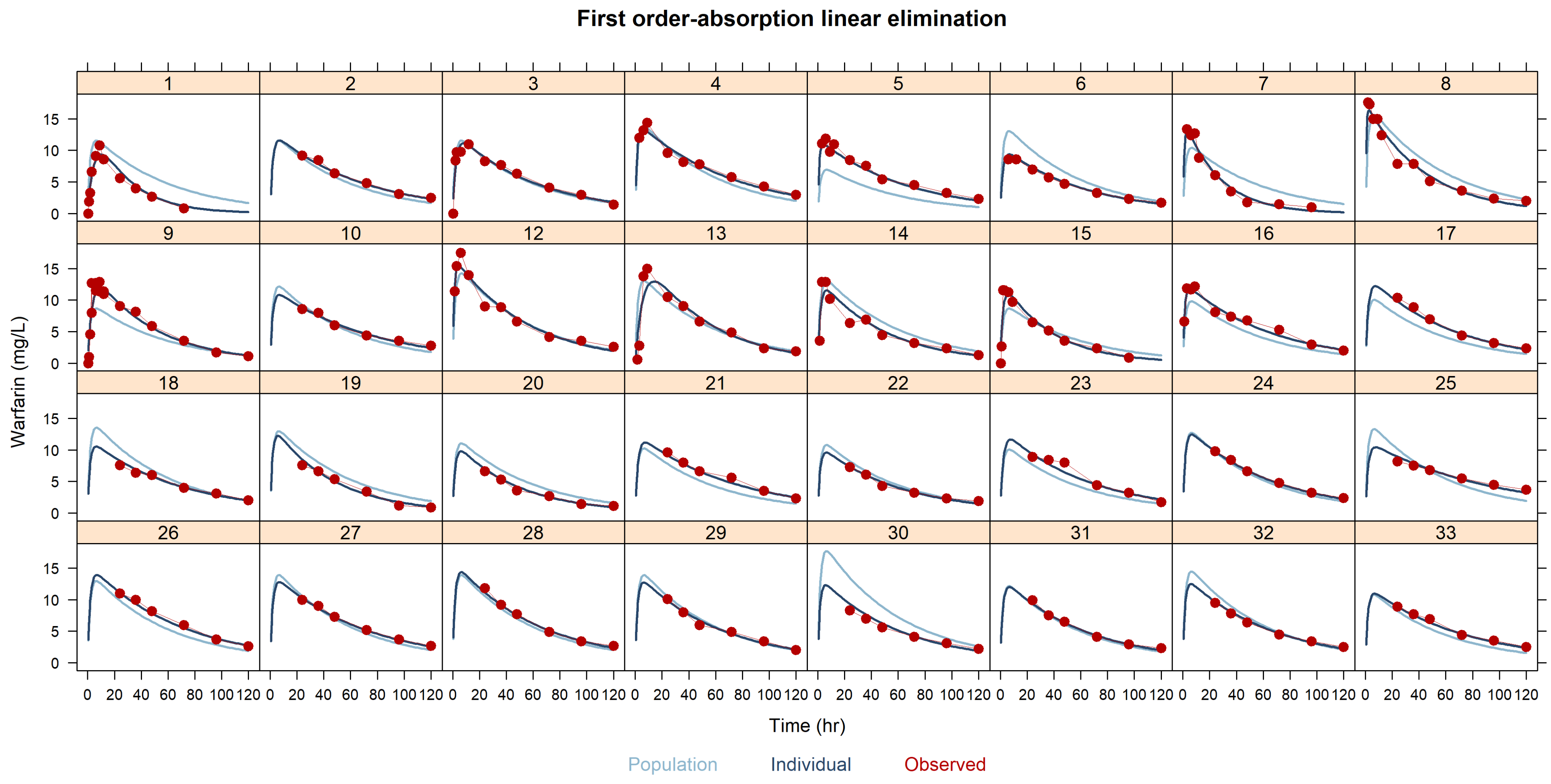
```
## Individual fits can be generated using augPred (augmented predictions)
## that provides smooth profiles by interpolating the predictions between observations:
plot(augPred(fitOne.comp.KA.solved_S))
## ...use the arrows in the plot window to examine the earlier curves
```



use augPred output to plot using your favourite package...

```
#or the augPred output can be plotted to your liking, for instance using ggplot2 or the lattice function xyplot:
indivpk<-augPred(fitOne.comp.KA.solved_S)
nlmixCOLS <- c("#28466A","#8DB6CD","#B40000") ## specify array of colours for curves

xyplot(
  values~time|id,          ## plot the variable values by time and make a separate panel for each id
  data=indivpk,            ## data source with smooth interpolated predictions and observations
  groups=ind,              ## make separate curves by ind that separates Observed data,
                           ## Individual predictions and Population predictions
  layout=c(8,4),           ## arrange as 8 columns and 4 rows
  type=c("l","l","p"),     ## represent these three by a line, a line and only markers (l=line, p=points)
  col=nlmixCOLS[c(2,1,3)], ## colours for each curve
  cex=c(0.1,0.1,1),        ## character size for the markers
  lwd=c(2,2,0.1),          ## line width of the lines
  pch=19,                  ## use closed circles as marker
  xlab="Time (hr)\n",      ## x-axis label
  ylab="Warfarin (mg/L)",  ## y-axis label
  as.table=TRUE,           ## have the first plot at the top left (otherwise plot 1 starts at the lower left corner)
  scales=list(alternating=1),  ## have axis labels at left and bottom (and not alternating)
  main="First order-absorption linear elimination", ## title for plot
  auto.key=list(adj=1,col=nlmixCOLS[c(2,1,3)],columns=3,space="bottom",rectangles=FALSE,points=FALSE) ## key for curves
)
```

..like lattice



nlmixr²