

Development of population PK model using  
R- Case study I

## Table of contents

|  |                  |
|--|------------------|
| <b><u>TABLE OF CONTENTS</u></b> .....  | <b><u>2</u></b>  |
| <b><u>CREATING A DATA FILE</u></b> .....                                     | <b><u>3</u></b>  |
| <i>EXPLANATION OF THE DATA FILE COLUMNS</i> .....                            | <i>4</i>         |
| <b><u>R CODE FOR MODEL FITTING</u></b> .....                                 | <b><u>5</u></b>  |
| <b><u>VIEWING AND INTERPRETING THE OUTPUT FILE</u></b> .....                 | <b><u>6</u></b>  |
| <b><u>DIAGNOSTIC PLOTS FOR STRUCTURAL MODEL AND INTERPRETATION</u></b> ..... | <b><u>7</u></b>  |
| <b><u>EXPLORING COVARIATE RELATIONSHIP</u></b> .....                         | <b><u>12</u></b> |
| <b><u>COVARIATE MODEL</u></b> .....  | <b><u>13</u></b> |
| <b><u>EXPLORING COVARIATE RELATIONSHIP</u></b> .....                         | <b><u>17</u></b> |
| <b><u>COMPILATION OF RESULTS</u></b> .....                                   | <b><u>19</u></b> |

## Creating a data file

The comma delimited data file consists of data obtained from 100 individuals who were randomized to either 100 or 250mg dose. A sample R data file for this study looks the following:

| CID | TIME | CONC   | AMT | DOSE | AGE    | WT     | SCR    | ISM | CLCR   |
|-----|------|--------|-----|------|--------|--------|--------|-----|--------|
| 1   | 0    | 0      | 100 | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 0.25 | 13.026 | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 0.5  | 14.984 | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 0.75 | 14.16  | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 1    | 19.316 | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 1.5  | 13.146 | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 2    | 12.921 | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 2.5  | 8.485  | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 3    | 16.437 | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 4    | 10.724 | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 6    | 8.7352 | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 8    | 7.697  | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 12   | 4.479  | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 16   | 2.4183 | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 20   | 4.7586 | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 1   | 24   | 2.3655 | NA  | 100  | 34.823 | 38.212 | 1.1129 | 0   | 42.635 |
| 2   | 0    | 0      | 100 | 100  | 32.765 | 74.838 | 0.8846 | 1   | 126    |
| 2   | 0.25 | 7.4203 | NA  | 100  | 32.765 | 74.838 | 0.8846 | 1   | 126    |
| 2   | 0.5  | 4.9869 | NA  | 100  | 32.765 | 74.838 | 0.8846 | 1   | 126    |
| 2   | 0.75 | 13.63  | NA  | 100  | 32.765 | 74.838 | 0.8846 | 1   | 126    |

A data file provided for NONMEM use can be used as follows: [Click here for link to the data file.](#) It needs to be imported within R as a dataframe.

To import data from a file or database into a data frame:

```
> cs1iv<- read.table("C:\\..\\CS1_IV1EST_PAR.csv", sep="," , header=TRUE, skip=3)
```

Arguments for importData include:

1. **file:** a character string specifying the name and location of the file to import
2. **sep:** the field separator character
3. **header:** a logical value indicating whether the file contains the names of the variables as its first line
4. **skip:** Number of lines

Note that additional arguments might be needed depending on the complexity of the source file. For the complete details, check the documentation of ?importData

To edit data in order to put it in R compatible format:

The following commands will convert the relevant parts of the string based on the condition provided. For example, the first line will substitute AMT=0 by AMT="NA" because in R "NA" is considered as missing.

```
> cs1iv$AMT[which(cs1iv$AMT==0)]<-as.numeric(c("NA"))
> cs1iv$CONC[which(cs1iv$CONC==0)]<-as.numeric(c("NA"))
```

### *Explanation of the data file columns*

| Column name | Description |
|-------------|-------------|
|-------------|-------------|

#### Required arguments:

|      |   |
|------|---|
| CID  | Subject identification number.  |
| TIME | Time of dosing or blood sampling time.  |
| CONC | Plasma concentration in this case, could be any dependant variable i. e., biomarker concentration or tissue concentration |
| AMT  | Dose administered at dosing time or zero for observation records  |

#### Optional arguments:

|               |   |
|---------------|---|
| DOSE          | Dose randomized for each patient, included for graphical requirements                                       |
| AGE, WT, ISM, | Covariate information for each individual.  |
| CLCR          | AGE = age in yrs, WT= Weight in kg, ISM = is male ( 1), female = 0,<br>CLCR= Creatinine clearance = ml/ min |

## R code for model fitting

The code (with explanations for each argument) for the structural model, which is chosen as a one-compartment i.v. bolus model based on preliminary data analysis, is given below. Load the library to conduct nonlinear mixed effects modeling.

```
>library(nlme)
```

**NLME**- This is a function that is used to fit a nonlinear mixed-effects model.

If not available, it can be obtained from CRAN

```
> local({a <- CRAN.packages() + install.packages  
(select.list(a[,1],TRUE), .libPaths()[1], available=a)})
```

- Click on nlme
- Click ok.

Or

- Click **Packages> Install Package(s) from CRAN.....**

- Click on nlme
- Click ok.

The following code explains the NLME syntax.

```
> csliv.fit<- nlme ( CONC~phenoModel (CID, TIME, AMT, lCl, lV),  
                  fixed = lCl+lV ~1, random = pdDiag(lCl+lV ~1),  
                  data = csliv.grp, start = c(-5, 0),  
                  weight=varConstPower(const=1,fixed=list(power=1)),  
                  na.action=na.include, naPattern = ~ !is.na(CONC) )
```

*Required Arguments:*

Model- For one compartment IV bolus PK data **phenoModel** is a generic function that allows parameterization in terms of clearance and volume.

*Model arguments:* Column names for Concentration, Subject identification Number, Time of dosing or blood sampling time, Dosing information and Fixed effect parameters to be estimated. Note that these names should match with the names in the dataframe.

Data- Data frame containing the variables named in model, fixed, random, etc. For **phenoModel** the dataframe needs to be in the form of groupedData.

```
> csliv.grp<-groupedData (CONC~TIME | CID, data=csliv)
```

Fixed- A linear formula for the model parameters, logarithm of clearance and volume, needs to be provided. A '1' on the right hand side of the formula(s) indicates a single fixed effect for the corresponding parameter(s).

*Optional Arguments:*

Random- A linear formula specifying random effects model needs to be provided.

Start- A numeric vector or list of initial estimates for the fixed effects and random effects.

Weight- An option to specify within-group heteroscedasticity structure. The default is 'NULL', corresponding to homoscedastic within-group errors. We have used a modified combined error model (see below) to account for heteroscedasticity. For other models, refer to our theory section.

na.action- a function that indicates what should happen when the data contain NAs. The default action ( na.fail) causes nlme to print an error message and terminate if there are any incomplete observations, “na.include” prevents this from happening.

```
> na.include <- function(x) x
#if function definition is not available
```

naPattern- an expression or formula object, specifying which returned values are to be regarded as missing.

For other details check the documentation of ?nlme

Viewing and interpreting the output file

For each run an output will be created in the assigned R object, “cs1iv.fit” in this case. All the results and plots can be processed from this object.

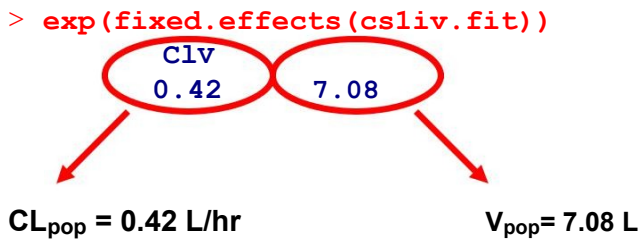
**THE LOGLIKELIHOOD AT CONVERGENCE**

```
> cs1iv.fit$logLik
-4398.72
```

**FINAL PARAMETER ESTIMATE OF BASE MODEL**

```
> fixed.effects(cs1iv.fit)
      lC1lv
-0.87      1.96
```

Parameters are reported as log estimates, to convert these parameters into easily interpretable numbers:



**OMEGA-COV MATRIX FOR RANDOM EFFECTS**

```
> VarCorr(cs1iv.fit)
CID = pdDiag(list(lC1 ~ 1,lV ~ 1))
      Variance StdDev
lC1  0.172      0.414
lV   0.196      0.442
Residual 0.0065 0.081
```

**With-in subject variability (Heteroscedastic error, see below)**

**Inter-individual variability on clearance (CV~41.4%) and volume (CV~44.2%)**

**The with-in group variance model:**

The with-in group variance function provided in this tutorial uses the following variance model

$$Var(\epsilon_{ij}) = \sigma^2 \cdot (\delta_{ij}^2 + \delta_{ij}^2) \quad (1)$$

Where  $\sigma^2$  is the variance of with-group error random variable,  $\delta_1$  and  $\delta_2$  are variance parameters and  $|v_{ij}|$  is the covariate. Thus if we fix the value of  $\delta_2$  to 1, it specifies a model in which standard deviation is a linear function of the covariate, the default is the fitted values in this case.

```
> csliv.fit
.....
Variance function:
Structure: Constant plus power of variance covariate
Formula: ~fitted(.)
Parameter estimates:
constpower
29.311.000
Number of Observations: 1500
Number of Groups: 100
δ1=29.31 δ2=1
```

Therefore,

$\sigma^2$  for the additive component can be approximated to,  $\sigma^2 \cdot \delta_1^2 = 5.64 \mu\text{g/mL}$

$\sigma^2$  for the power component can be approximated to,  $\sigma^2 \cdot |v_{ij}|^2 = 0.006 = 8.1 \% \text{ CV}$

**CAUTION:** It should be noted that these results CANNOT be compared with the results obtained from the NONMEM fits. The variance model implemented in NONMEM is of the following form:

$$y_{ij} = f_{ij} + f_{ij} \cdot ERR(1) + ERR(2)$$

$$Var(\epsilon_j^i) = F^2 \cdot \sigma_1^2 + \sigma_2^2 \tag{2}$$

Simplifying it;

$$\begin{aligned} Var(\epsilon_j^i) &= \sigma_1^2 (F^2 + \sigma_2^2 / \sigma_1^2) \\ &= \sigma_1^2 (F^2 + \delta_1) \\ &= \sigma_1^2 (|v_{ij}|^{\delta_2} + \delta_1) \\ &= \sigma_1^2 (|v_{ij}|^{\delta_2} + \delta_1) \tag{3} \end{aligned}$$

It is evident that equations 1 and 3 have some similarity (they are not identical), however, the equation 1 [`varConstPower(fixed = list(power = 1))`] is the closet implementation of  $y_{ij} = f_{ij} + f_{ij} \cdot ERR(1) + ERR(2)$  model in R. At this time, we are not aware of implementing that model in R.

[Diagnostic plots for structural model and interpretation](#)

The first plot to look at would be a plot of concentration (observed, individual predicted and population predicted) versus time for all the individuals. A representative plot of few individuals



is provided below. These plots need to be generated with additional data processing, a generic function is given below.

Extract individual and population predictions predictions:

```
> ipred<-fitted(csliv.fit,level=1)
> poppred<-fitted(csliv.fit,level=0)
```

Create a dataframe by adding the above information to make plotting convenient. Note that the missing concentrations data are deleted because there are those time points are excluded from predictions.

```
> csliv.pred<-cbind (csliv.grp[!is.na(csliv.grp$CONC),] ,IPRED=ipred,PRED=poppred)
```

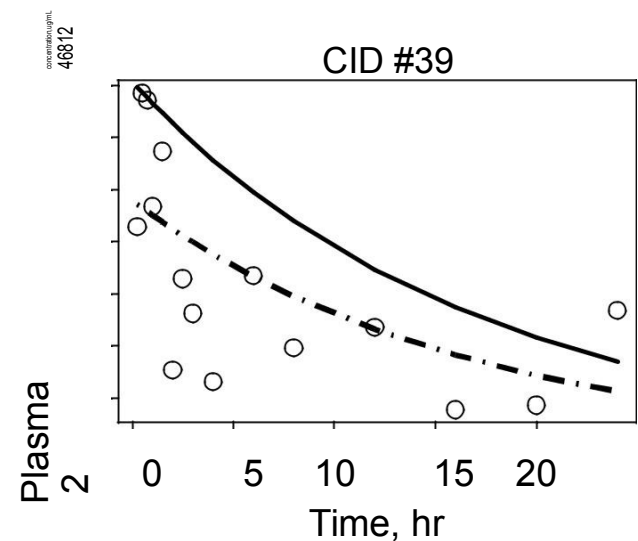
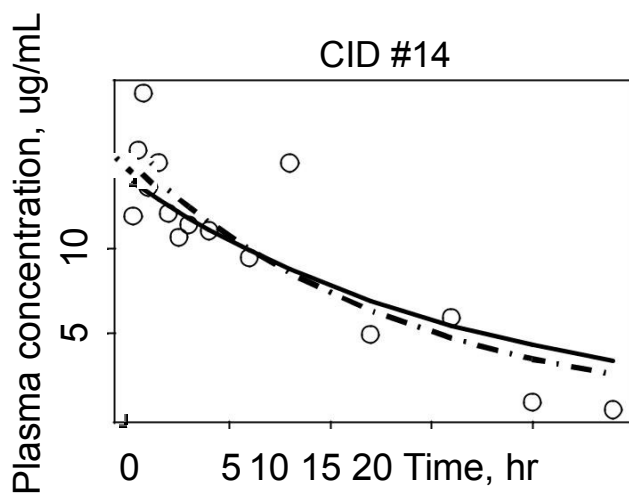
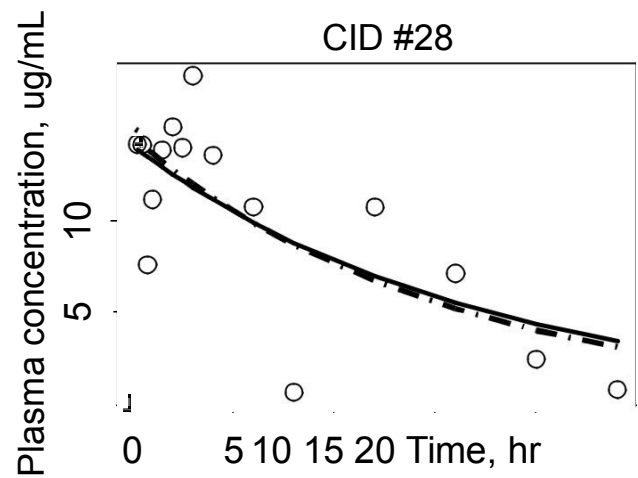
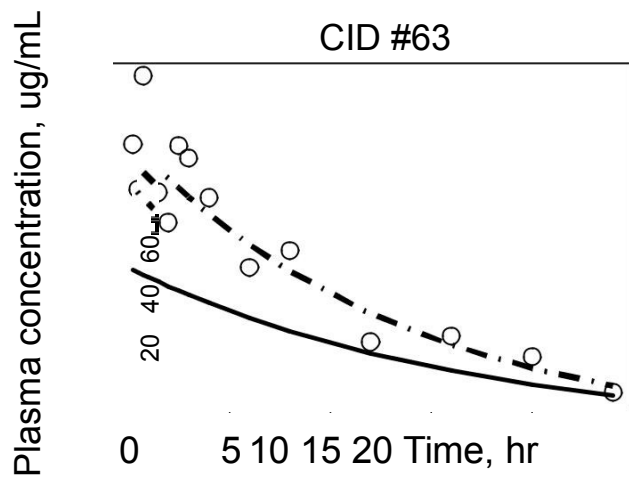
Make diagnostic plots for each individual to view inconsistencies between model and data by using the following code.

```

> attach(csliv.pred)
> par(mfrow=c(2,2),mar=c(8,8,2,2),oma=c(0,0,0,0))
> # plot area specifications
> for(i in unique(CID))
> {
>   j<-CID==i
>   plot(TIME[j],CONC[j],xlab="Time, hr", ylab="Plasma
      concentration, ug/mL",type="p",cex=1.2)
>   lines(TIME[j],IPRED[j],lty =3,lwd=4)
>   lines(TIME[j],PRED[j],lty=1,lwd=3)
>   title(paste("CID #",i,sep=""))
> }
> detach()

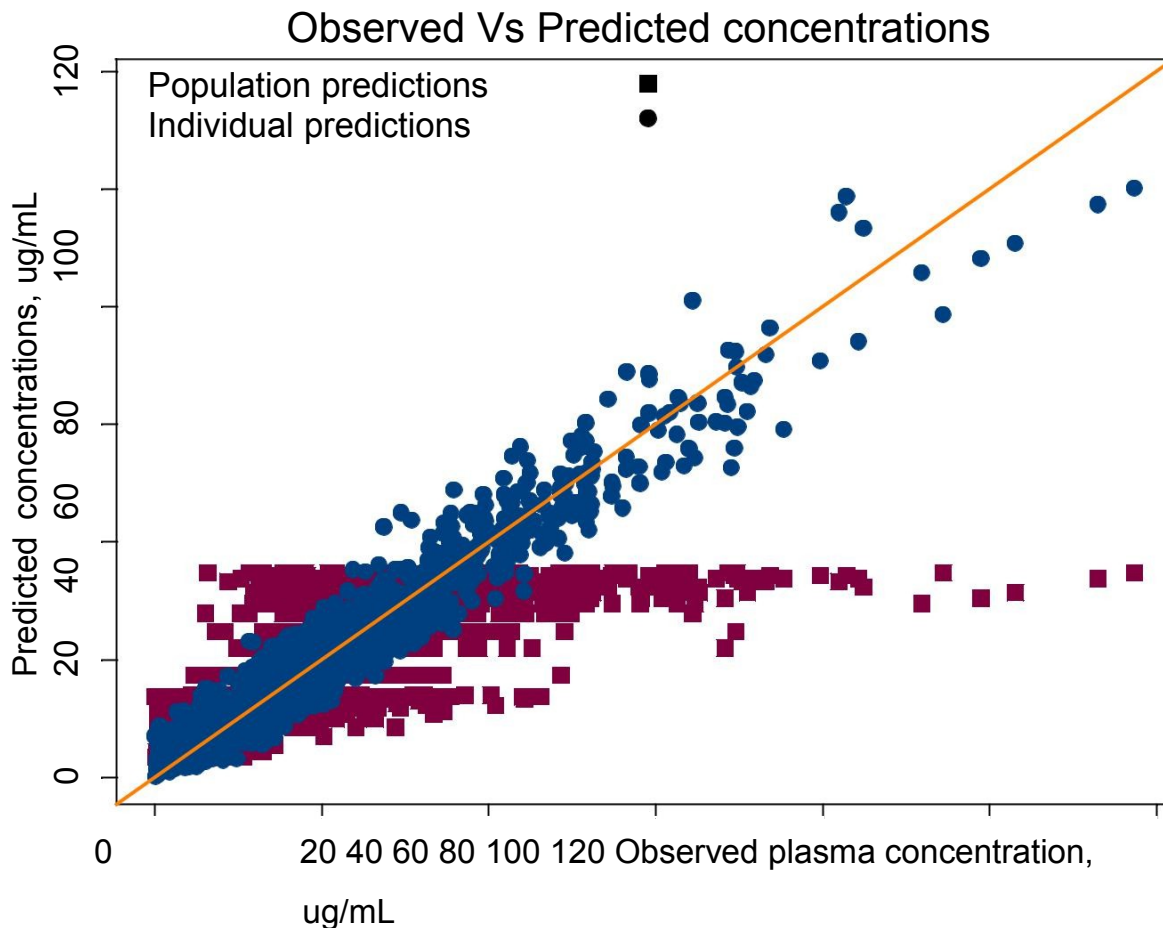
```

The following are representative plots generated using the above code.



Some other useful diagnostic plots (Observed vs individual predicted concentrations and observed vs population predicted concentrations). These graphs could be looked for any bias in the predictions. Ideally the points should be uniformly randomly distributed along the line of identity. Here we could see that population predictions are biased at higher observed concentrations. The reason for it needs to be explored. The following code can be used to generate these plots

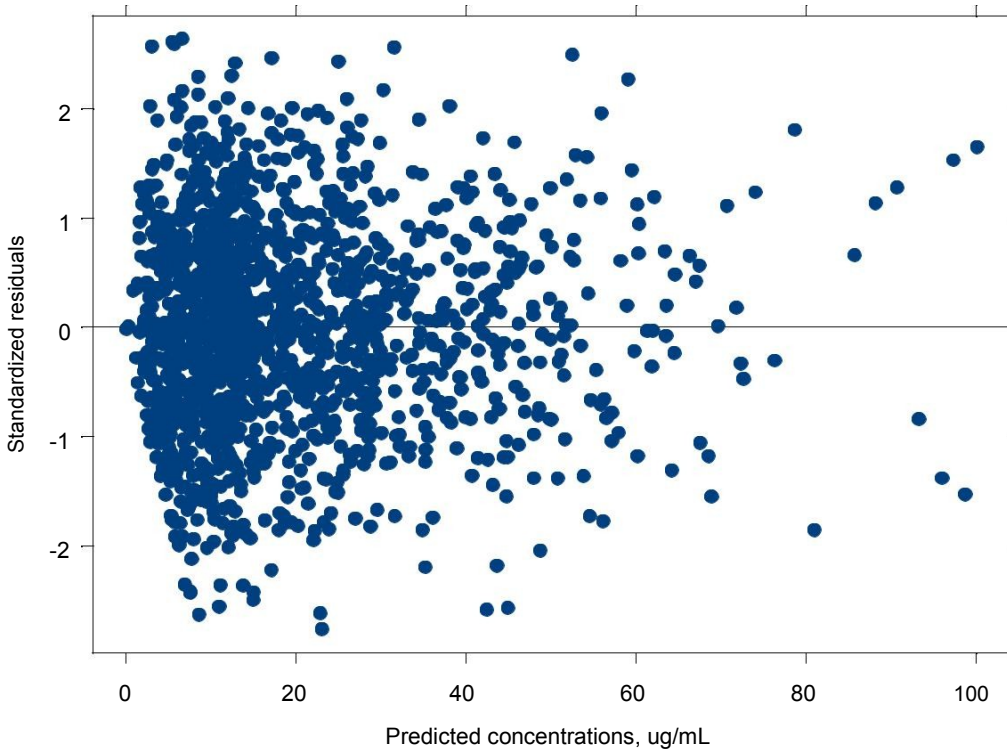
```
> attach(csliv.pred)
> limx <- c(min(c(CONC, PRED, IPRED)), max(c(CONC, PRED, IPRED)))
  #obtain limits fore the plots to be generated
> par(mar = c(6, 6, 2, 2), cex = 1.2) ##plot area specifications
> plot(x = CONC, y = PRED, xlab = "Observed plasma concentration,
  ug/mL", ylab = "Predicted concentrations, ug/mL", type = "n",
  xlim = limx, ylim = limx, cex = 1.2)
> points(x = CONC, y = PRED, pch = 15, cex = 1.2, col = 3)
> points(x = CONC, y = IPRED, pch = 16, cex = 1.2, col = 2)
> abline(0, 1, lwd = 3, col = 5) ##Line of identity
> title("Observed Vs Predicted concentrations")
> legend(0,limx[2],c("Population predictions", "Individual
  predictions"), pch = c(15, 16),cex=0.75, cex = 1)
> detach()
```



Residual plots will also be useful to monitor any heteroscedasticity that is still unaccounted for.

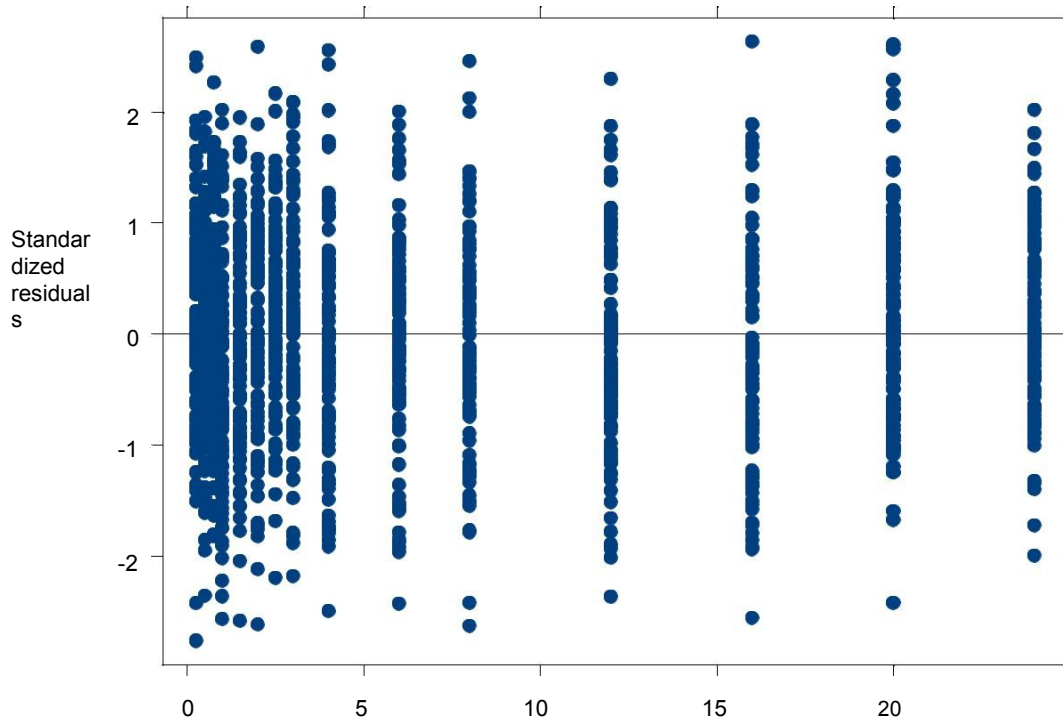
```
> plot(csliv.fit, resid(., type="p")~fitted(.), abline=0)
```

Standardized residuals plot



```
> plot(csliv.fit, resid(., type="p")~TIME, abline=0)
```

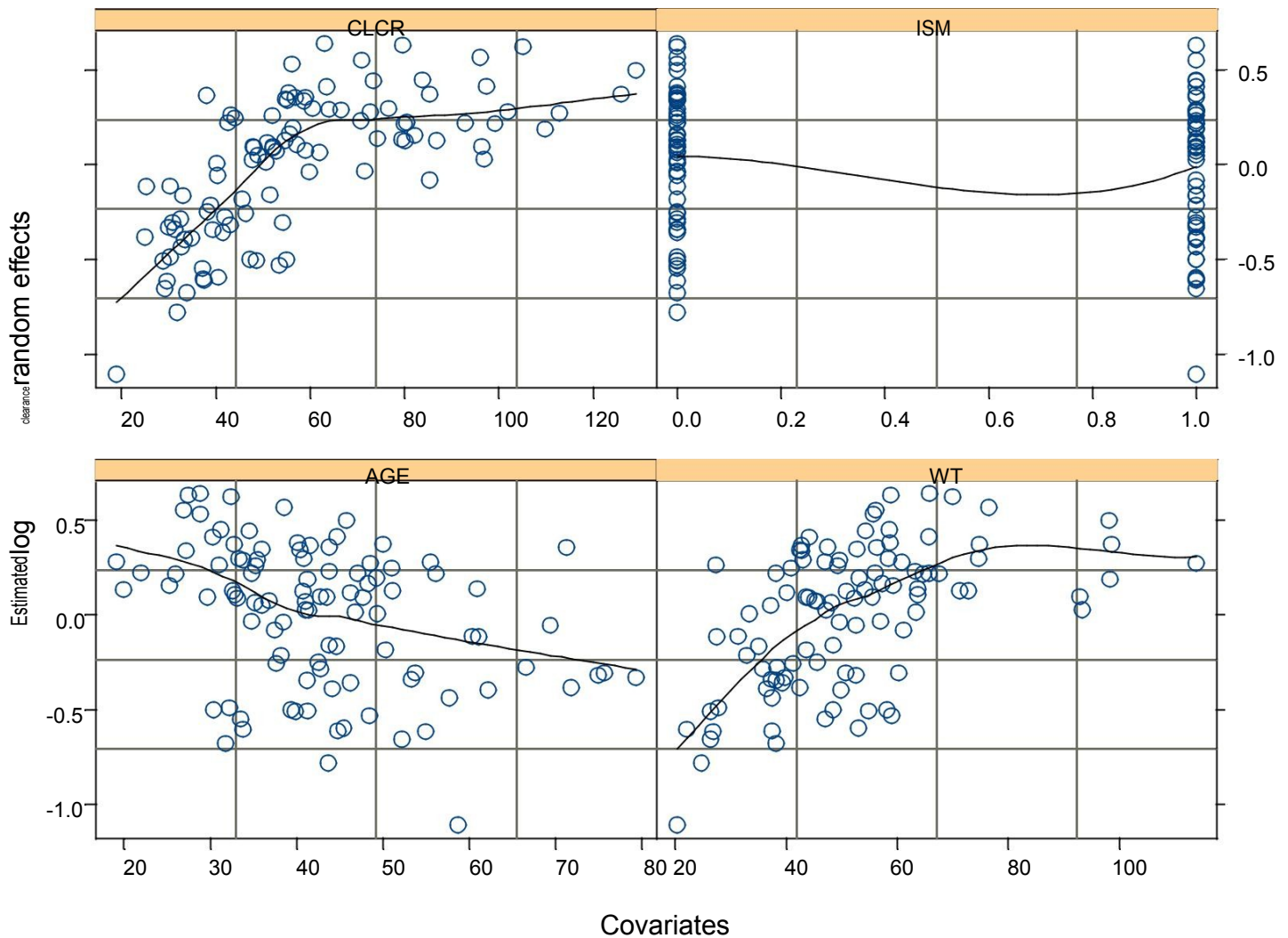
Standardized residuals plot



## Exploring covariate relationship

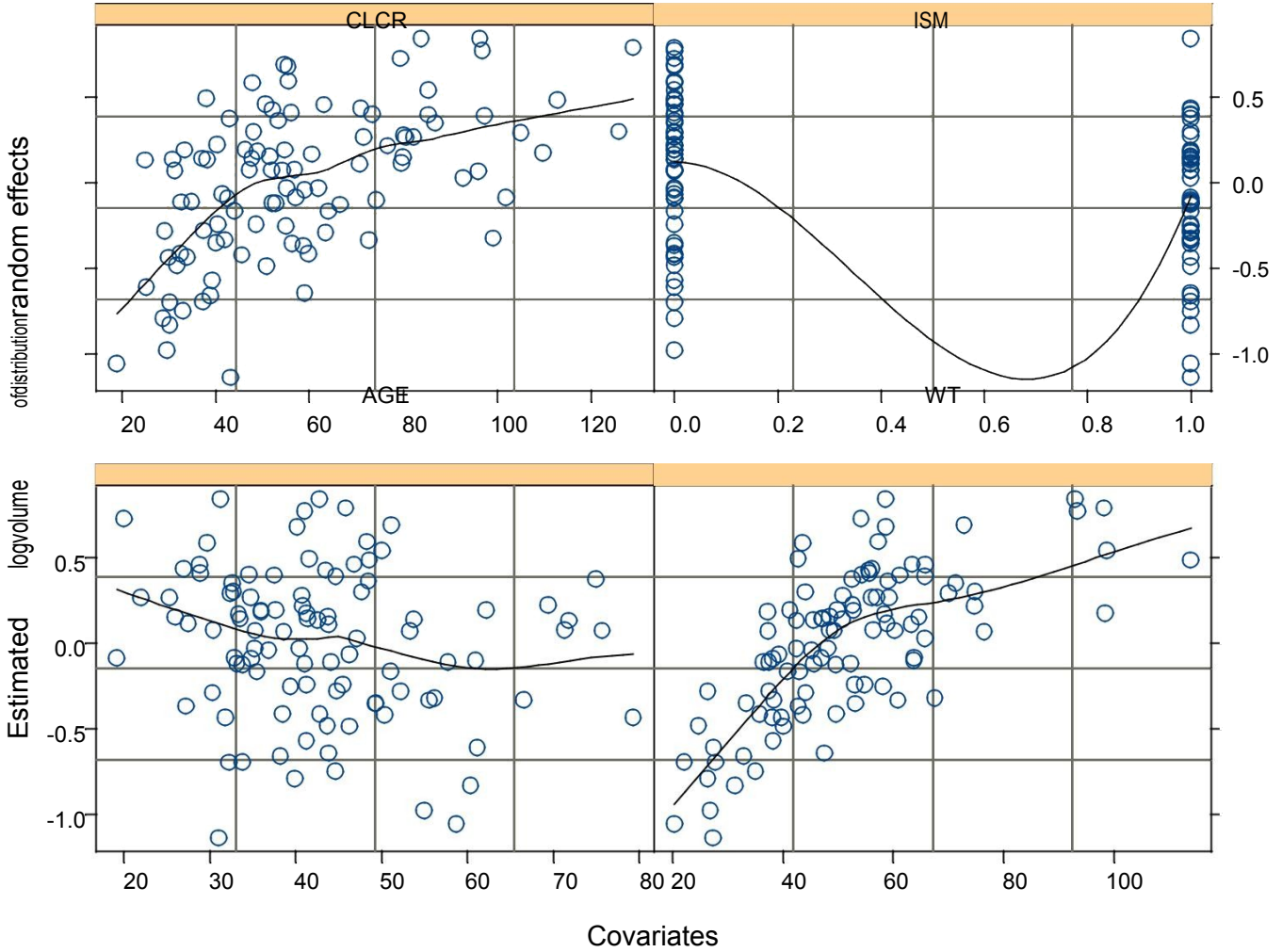
As the Observed concentrations vs population predictions plot looks biased, it would be necessary to explore covariate parameter relationship. According to background information of the drug, the drug is said to be renally eliminated and a general range of CLCR values from the data is from 20 to 130 ml/ min. To make plots data need to be extracted from the fitted object. First we will explore covariate effects on clearance followed by volume of distribution.

```
> csliv.ranef<-ranef(csliv.fit,augFrame=T)
> par(mar=c(8,8,2,2), cex = 1.2,oma=c(0,0,0,0))
> plot(csliv.ranef,form=lCl~AGE+WT+CLCR+ISM,cex=1.25)
```



Note: Please note that this argument includes a loess smoother to aid in visualizing trends but it does not make sense when covariates are categorical or discrete variables (for example: Gender).

```
> plot(csliv.ranef,form=lV~AGE+WT+CLCR+ISM,cex=1.25)
```



It could be seen from the graph that estimated log clearance random effects and CLCR have a correlation, which is expected in our case and needs to be explored in the covariate model. Correlation between age and CLCR and Weight and CLCR is expected, but since one of the correlational covariates ( CLCR) based on pre- existing clinical reasons will be explored, there is no need to explore weight and age as covariates for clearance. In the case of volume, based on existing clinical reasoning, weight could be an influential covariate and needs to be explored in the covariate model. Although in the scatter plot, estimated log volume of distribution random effects and CLCR looks to have some relation, it does not carry any physiological meaning.

## Covariate model

Based on graphical analysis, in our covariate model we will include CLCR as a covariate on CL and weight as a covariate on V. To explore these effects, we will center these data on the typical values of the covariates.

```
> csliv.cov<- update( csliv.fit,
                      CONC~phenoModel(CID,TIME, AMT, lCl, lV+log(WT/70)),
                      fixed = list(lCl ~log(CLCR/120),lV~1),
                      start = c(-5, 0,0)
                      ) #Fit covariate model
```

The above model includes covariates on clearance and volume based on mechanistic reasoning. The following models are implemented:

$$CL = CL_{pop} \cdot (CLCR / 120)^{\alpha} \cdot \exp(\eta_{CL})$$

$$V = V_{pop} \cdot (WT / 70)^{\beta} \cdot \exp(\eta_V) \text{ (based on allometry)}$$

In the code, lCl~log(CLCR/120) ensures that  $\alpha$  is estimated for CLCR centered on the typical value of 120. Note that the log parameters are modeled here so the coefficient of log(CLCR/120) would be an estimate of  $\alpha$

On the other hand, it well established that the coefficient of log(WT/70), that is  $\beta$  is equal to 1. So the fixed effect parameter was influenced directly by the covariate information.

## THE LOGLIKELIHOOD AT CONVERGENCE

```
> csliv.cov$logLik
-4319.439
```

Objective function value decreased from the base model by 79.0 points. A log-likelihood ratio test is used to compare between base and covariate model (nested models). Here the inclusion of CLCR as a covariate on CL and weight as a covariate on V are significant ( $p < 0.001$ ).

## FINAL PARAMETER ESTIMATE OF MODEL

```
> fixed.effects(csliv.cov)
      lCl      lCl.log(CLCR/120)      lV
-0.14         0.85                2.31
```

Parameters are reported as log estimates, to convert these parameters into easily interpretable numbers:

```
> exp(c(fixed.effects(csliv.cov)[1], fixed.effects(csliv.cov)[3]))
```

CL<sub>pop</sub> = 0.87 L/hr      V<sub>pop</sub> = 10.09 L

## OMEGA-COV MATRIX FOR RANDOM EFFECTS

```
> VarCorr(csliv.cov)
      CID = pdDiag(list(lCl ~ 1,lV ~ 1))
      Variance      StdDev
lCl.(Intercept) 0.057      0.239
lV.(Intercept)  0.086      0.293
Residual        0.0065     0.081
```



**With-in subject variability  
(Heteroscedastic within group error )**

**Inter-individual variability on clearance  
(23.9%) and volume (29.3%)**

Inclusion of significant covariates namely CLCR and Weight has decreased the unexplained variability in the data.

**STANDARD ERROR ESTIMATE FOR PARAMETERS**

The function call "summary" yields standard error of parameter estimates.

```
>summary(csliv.cov)
```

```
.....
```

|                   | Value | Std.Error | DF   | t-value | p-value |
|-------------------|-------|-----------|------|---------|---------|
| lCl. (Intercept)  | -0.14 | 0.074     | 1397 | -1.88   | 0.0608  |
| lCl.log(CLCR/120) | 0.86  | 0.078     | 1397 | 11.02   | 0.0000  |
| lV. (Intercept)   | 2.31  | 0.046     | 1397 | 50.41   | 0.0000  |
| lV.log(WT/70)     | 0.99  | 0.093     | 1397 | 10.63   | 0.0000  |

```
.....
```

Diagnostic plots for structural model and interpretation

We will make plot of concentration (observed, individual predicted and population predicted) versus time for all the individuals. A representative plot of few individuals is provided below. These plots need to be generated with additional data processing, a generic function is given above. This is of the same form as given previously.

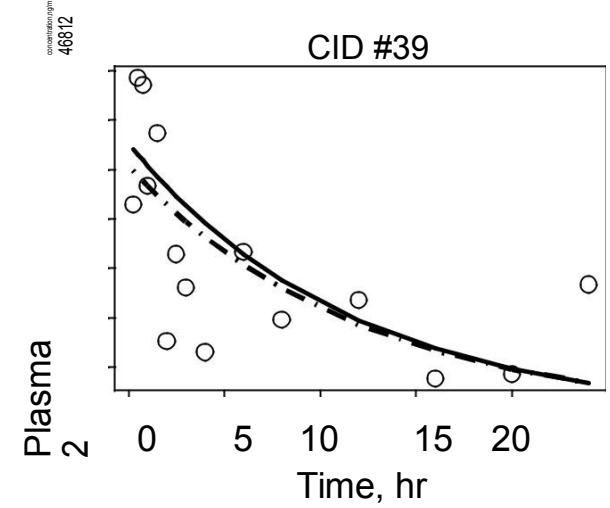
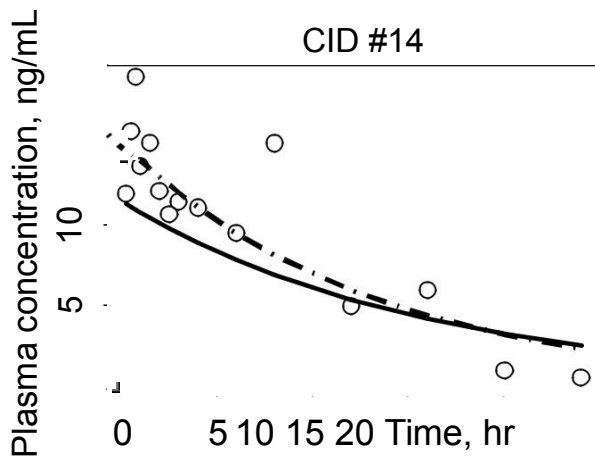
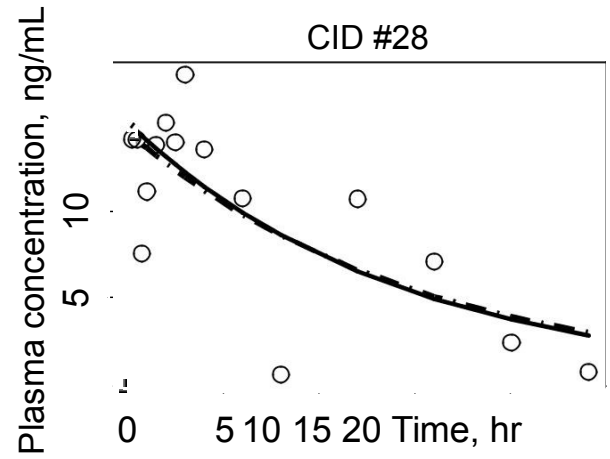
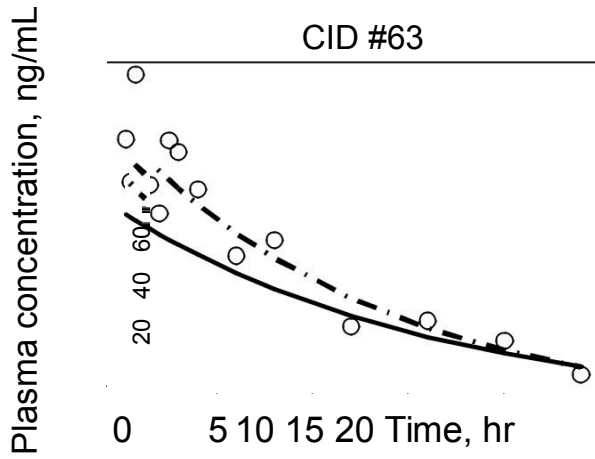
```
> covipred<-fitted(csliv.cov,level=1)
> covpoppred<-fitted(csliv.fit,level=0)
```

Create a dataframe by adding the above information to make plotting convenient. Note that the missing concentrations data are deleted because there are those time points are excluded from predictions.

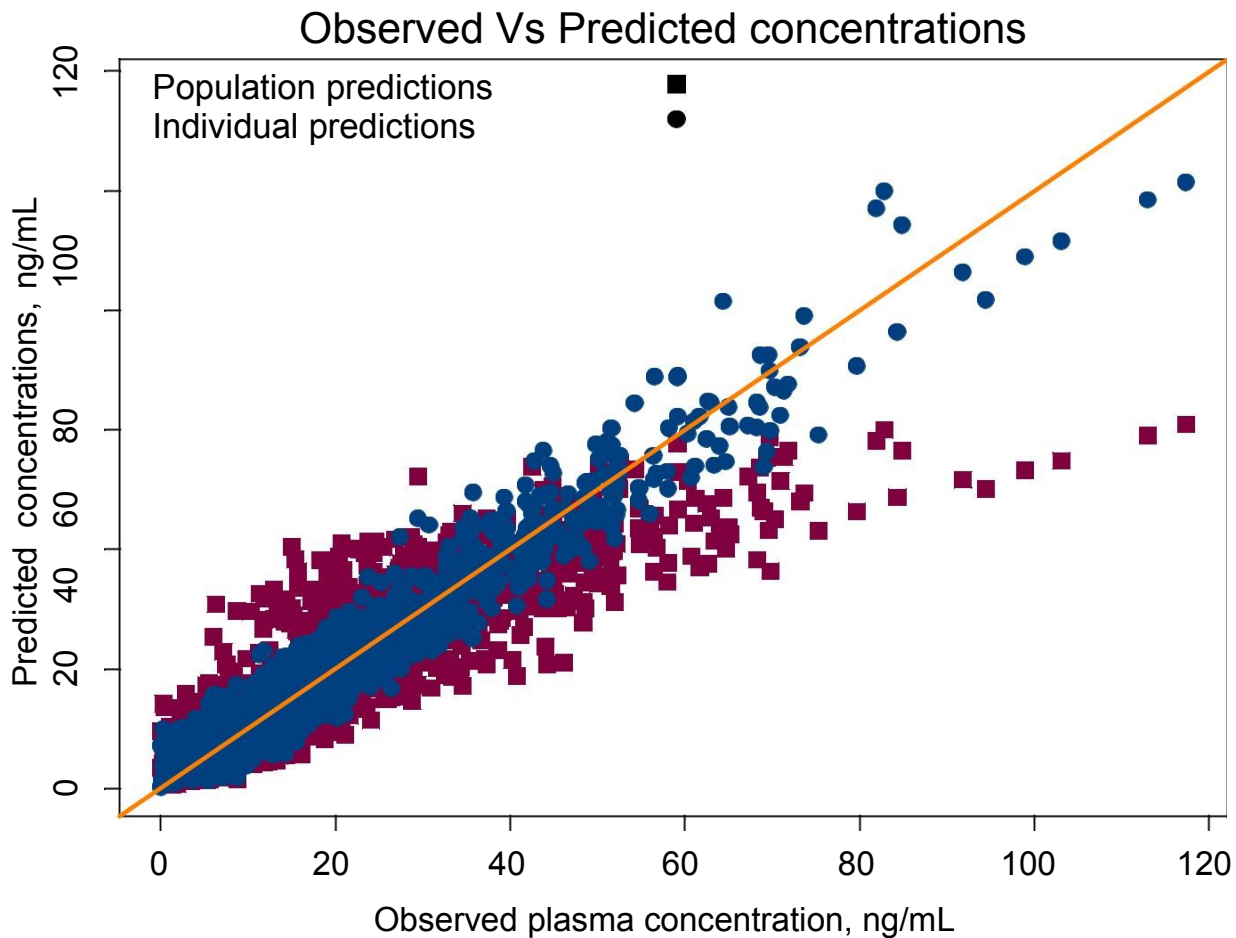
```
> csliv.covpred<-cbind (csliv.grp[!is.na(csliv.grp$CONC),] ,IPRED=covipred,PRED=covpoppred)
```

Run the exact same codes as previously to generate concentration versus time plots. The following are representative plots generated using the above code.





The second set of diagnostic plots would be the Observed vs individual predicted concentrations and observed vs population predicted concentrations. Again, the exact same code can be used to generate these plots.



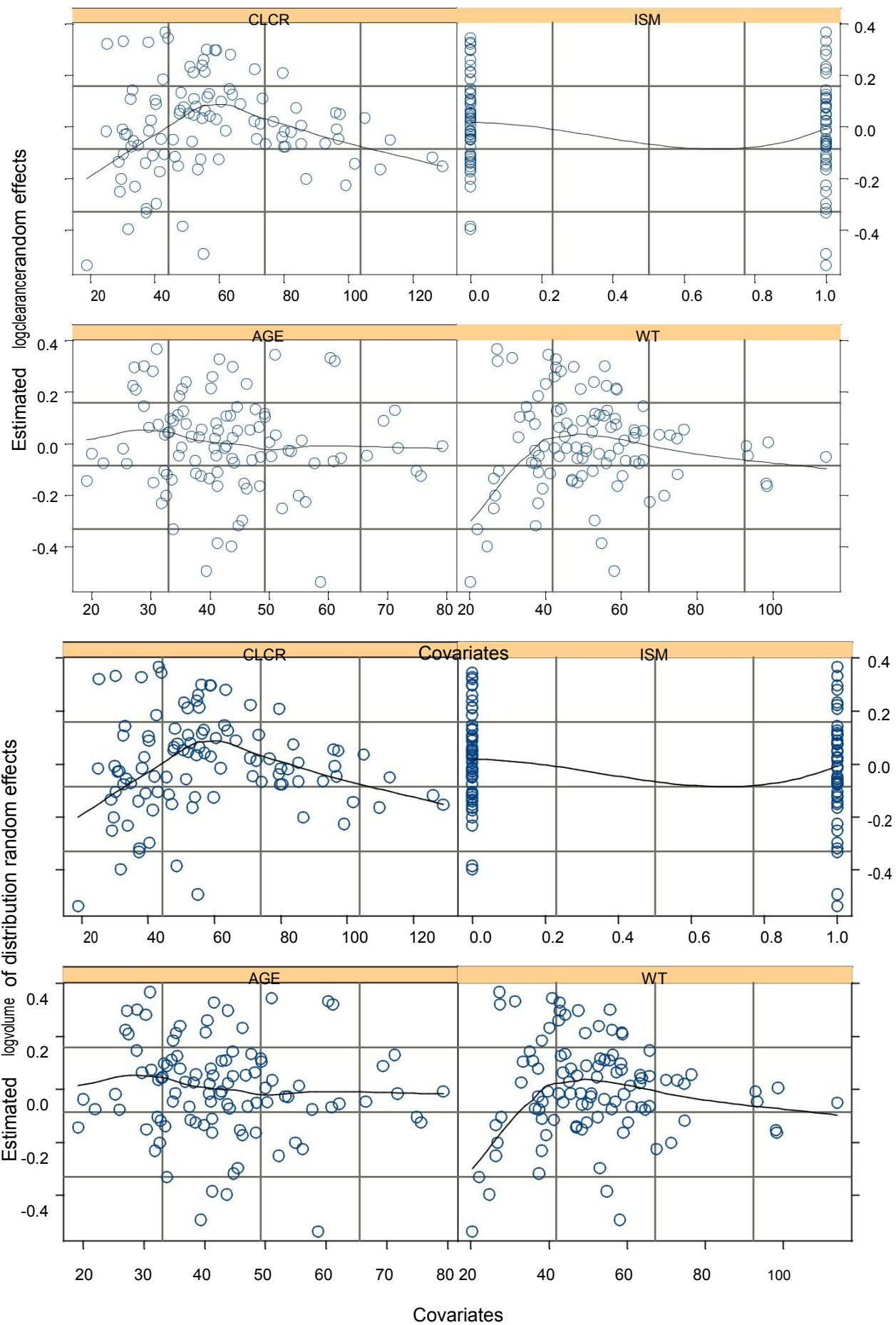
It could be seen that DV vs PRED shows uniformly randomly distributed points on either sides of lines of identity. The bias seen before is not seen here.

#### Exploring covariate relationship

We will look at the covariates the same way.

```
> csliv.cov.ranef<-ranef(csliv.cov,augFrame=T)
> plot(csliv..cov.ranef,form= lC1.
      (Intercept)~AGE+WT+CLCR+ISM, ylab="Estimated log clearance
      random effects", xlab="Covariates",cex=1.25)
> plot(csliv..cov.ranef,form= lV.(Intercept)~AGE+WT+CLCR+ISM,
      ylab="Estimated log clearance random effects",
      xlab="Covariates",cex=1.25)
```

Also note that estimated log clearance random effects versus CLCR and estimated log volume of distribution random effects versus weight does not have any trend.



### Compilation of results

The results obtained could be compiled and documented according to the user's documentation procedure. Following is a way of compiling the results.

| Model   | Log likelihood   | Population estimate (%SE) | Interindividual variability (%SE) |
|---|------------------|---------------------------|-----------------------------------|
| <b>Base model</b><br>i.v. 1 compartment model<br>$CL = CL_{pop} \cdot \eta CL$<br>$V = V_{pop} \cdot \eta V$  | -4398.72         |                           |                                   |
| <b>Final model</b><br>Iv 1 cmt with covariates<br><br>$CL = CL_{pop} \cdot (CLCR / 120)^{\alpha} \cdot \exp(\eta CL)$<br><br>$V = V_{pop} \cdot (WT / 70) \cdot \exp(\eta V)$ | -4319.439        |                           |                                   |
|   | <b>CL (L/hr)</b> | 0.87 (9.3)                | 23.9 %                            |
|   | <b>V (L)</b>     | 10.09 (10.7)              | 29.3 %                            |
| <b>Residual variability</b>   |                  |                           |                                   |
| Additive error  | 5.64 µg/mL       |                           |                                   |
| Proportional error  | 8.01%            |                           |                                   |