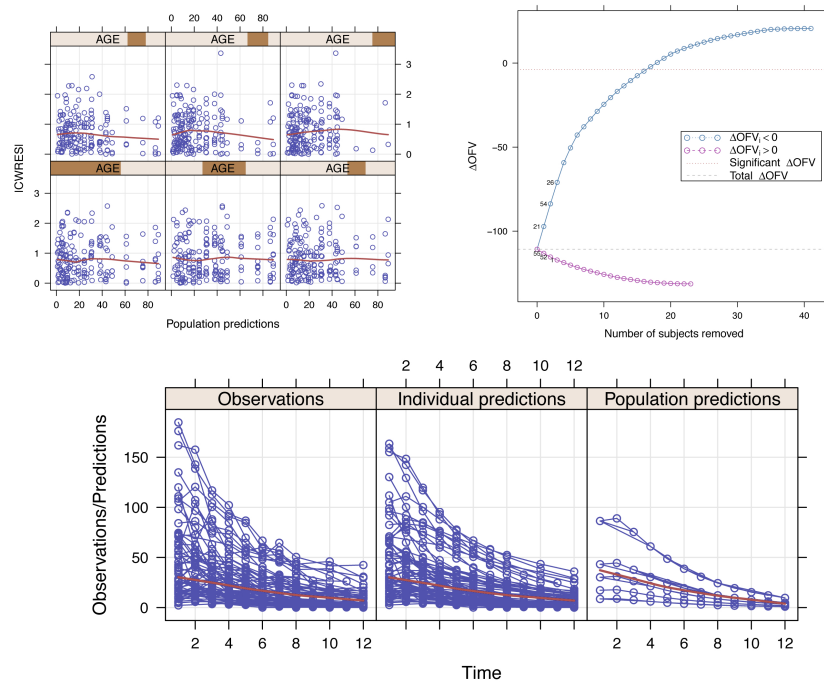


# Xpose 4 Bestiary

Version 1.0

E Niclas Jonsson and Andrew Hooker

Pharmacometrics Research Group  
Uppsala University



*A bestiary, or Bestiarum vocabulum, is a compendium of beasts.*

# Contents

<b>Introduction</b>	<b>6</b>
<b>Data file checkout</b>	<b>8</b>
data.checkout . . . . .	9
<b>Data visualization</b>	<b>10</b>
dv.vs.idv . . . . .	11
pred.vs.idv . . . . .	12
ipred.vs.idv . . . . .	13
cov.hist . . . . .	14
cov.qq . . . . .	15
cov.splom . . . . .	16
cov.summary . . . . .	17
<b>Overall goodness of fit assessment</b>	<b>18</b>
dv.preds.vs.idv . . . . .	19
ind.plots . . . . .	20
basic.gof . . . . .	21
addit.gof . . . . .	22
runsum . . . . .	23
<b>Structural model diagnostics</b>	<b>24</b>
dv.vs.pred . . . . .	25
dv.vs.ipred . . . . .	26
dv.vs.pred.ipred . . . . .	27
dv.vs.pred.by.cov . . . . .	28
dv.vs.ipred.by.cov . . . . .	29
dv.vs.pred.by.idv . . . . .	30
dv.vs.ipred.by.idv . . . . .	31
cwres.vs.idv . . . . .	32
wres.vs.idv . . . . .	33

iwres.vs.idv . . . . .	34
cwres.wres.vs.idv . . . . .	35
cwres.vs.idv.bw . . . . .	36
wres.vs.idv.bw . . . . .	37
cwres.vs.pred . . . . .	38
wres.vs.pred . . . . .	39
cwres.wres.vs.pred . . . . .	40
cwres.vs.pred.bw . . . . .	41
wres.vs.pred.bw . . . . .	42
cwres.vs.cov . . . . .	43
wres.vs.cov . . . . .	44
<b>Residual model diagnostics</b>	<b>45</b>
absval.cwres.vs.pred . . . . .	46
absval.wres.vs.pred . . . . .	47
absval.iwres.vs.pred . . . . .	48
absval.iwres.vs.ipred . . . . .	49
absval.wres.vs.idv . . . . .	50
absval.iwres.vs.idv . . . . .	51
absval.iwres.cwres.vs.ipred.pred . . . . .	52
absval.iwres.wres.vs.ipred.pred . . . . .	53
cwres.dist.hist . . . . .	54
wres.dist.hist . . . . .	55
iwres.dist.hist . . . . .	56
cwres.dist.qq . . . . .	57
wres.dist.qq . . . . .	58
iwres.dist.qq . . . . .	59
ind.plots.cwres.hist . . . . .	60
ind.plots.wres.hist . . . . .	61
ind.plots.cwres.qq . . . . .	62
ind.plots.wres.qq . . . . .	63

absval.cwres.vs.pred.by.cov . . . . .	64
absval.wres.vs.pred.by.cov . . . . .	65
absval.iwres.vs.ipred.by.cov . . . . .	66
absval.cwres.vs.cov.bw . . . . .	67
absval.wres.vs.cov.bw . . . . .	68
absval.iwres.vs.cov.bw . . . . .	69
autocorr.cwres . . . . .	70
autocorr.wres . . . . .	71
<b>Parameter distribution diagnostics</b>	<b>72</b>
parm.hist . . . . .	73
ranpar.hist . . . . .	74
parm.qq . . . . .	75
ranpar.qq . . . . .	76
parm.splom . . . . .	77
ranpar.splom . . . . .	78
parm.vs.parm . . . . .	79
tabulate.parameters . . . . .	80
parm.summary . . . . .	81
<b>Model comparison</b>	<b>82</b>
basic.model.comp . . . . .	83
add.model.comp . . . . .	84
absval.dpred.vs.cov.model.comp . . . . .	86
absval.dipred.vs.cov.model.comp . . . . .	87
absval.dcwres.vs.cov.model.comp . . . . .	88
absval.dwres.vs.cov.model.comp . . . . .	89
absval.diwres.vs.cov.model.comp . . . . .	90
dOFV.vs.id . . . . .	91
dOFV.vs.cov . . . . .	92
dOFV1.vs.dOFV2 . . . . .	93

<b>Model development</b>	<b>94</b>
parm.vs.cov . . . . .	95
ranpar.vs.cov . . . . .	96
xpose.gam . . . . .	97
xp.plot . . . . .	99
xp.ind.inf.fit . . . . .	100
xp.ind.inf.terms . . . . .	101
xp.ind.stud.res . . . . .	102
xp.akaike.plot . . . . .	103
<b>Getting more help</b>	<b>104</b>
<b>What is not included?</b>	<b>105</b>
<b>How to make NONMEM generate input to Xpose</b>	<b>106</b>
How Xpose identifies NONMEM runs . . . . .	106
Xpose data variables . . . . .	106
Default variable definitions . . . . .	106
Changing the default variable definitions . . . . .	106
<b>How Xpose identifies data items</b>	<b>109</b>
<b>How to extend and modify the behavior of Xpose specific functions</b>	<b>110</b>

# Introduction

Xpose is an R library for post-processing of NONMEM output. It takes one or more standard NONMEM table files as input and generates graphs or other analyses. It is assumed that each NONMEM run can be uniquely identified by a run number (see Section “How to make NONMEM generate input to Xpose” on page 106) for how to generate the appropriate input to Xpose). Xpose is implemented using the `lattice` graphics library in R.

The Xpose library has five sub-libraries:

**xposedata** Contains functions for managing the input data and manipulating the Xpose database.

**xposegeneric** Contains generic wrapper functions around the `lattice` functions. These functions can be invoked by the user but require quite detailed instructions to generate the desired output.

**xposespecific** These functions are single purpose functions that generate specific output given only the Xpose database as input. The behavior can to some extent be influenced by the user.

**xposeclassic** Historically, Xpose has had a text based menu interface to make it simple for the user to invoke the Xpose specific functions. This interface is called Xpose Classic. Given the limitations a text based interface imposes, Xpose Classic is not very flexible but may still be useful for quick assessment of a model.

**xposesspecial** These functions are the interface between Xpose and PsN, i.e. they do not post-process NONMEM output but rather PsN output.

This document is intended to be a showcase for the graphical displays that can be produced using the functions in the Xpose specific library. Many of the graphs can be generated using the classical Xpose text based interface (albeit in a less flexible way) but not all. On the other hand, to use the Xpose specific functions directly, it is necessary to type the functions on the command line and, in some cases, do some simple data management before the functions are used.

In the following, each of the Xpose specific functions will be exemplified. Each display will have a legend that hopefully will be a good starting point for reports and papers. Together with each example will be the code used to generate the graph starting from NONMEM table files.

Each example assumes that the Xpose library has been attached:

```
> library(xpose4)
```

and that appropriate table files for Xpose have been produced. When preparing this document, the table files were located in the directory above from where R was running, hence the `directory=".."` argument to `xpose.data` in all of the examples.

There are quite a few specific functions in the Xpose library and for the sake of presentation, they have been categorized into sections:

**Data visualization** Graphs intended to plot the raw data, for example concentrations versus time or histograms of the covariates.

**Overall goodness of fit assessment** Xpose has a number of composite displays (multiple graphs of different kinds on the same page) that provide diagnostics for multiple aspects of the fit. These plots go into this section.

**Structural model diagnostics** These are graphs that are more geared towards diagnosing the appropriateness of the structural model.

**Residual model diagnostics** Plots for diagnosing the statistical portions of the model (for example, residual error and between subject variability).

**Parameter distribution diagnostics** Plots in this section focus on the inter-individual variability model.

**Model comparison** Graphs for comparing various aspects of two models fit to the same data.

**Model development** Graphs and functions that are supposed to help the modeler decide what to do next with the model. Most of these functions focus on aspects of covariate inclusion.

Please note that these sections are “fuzzy”, in the sense that many plots could be categorized into several different categories.

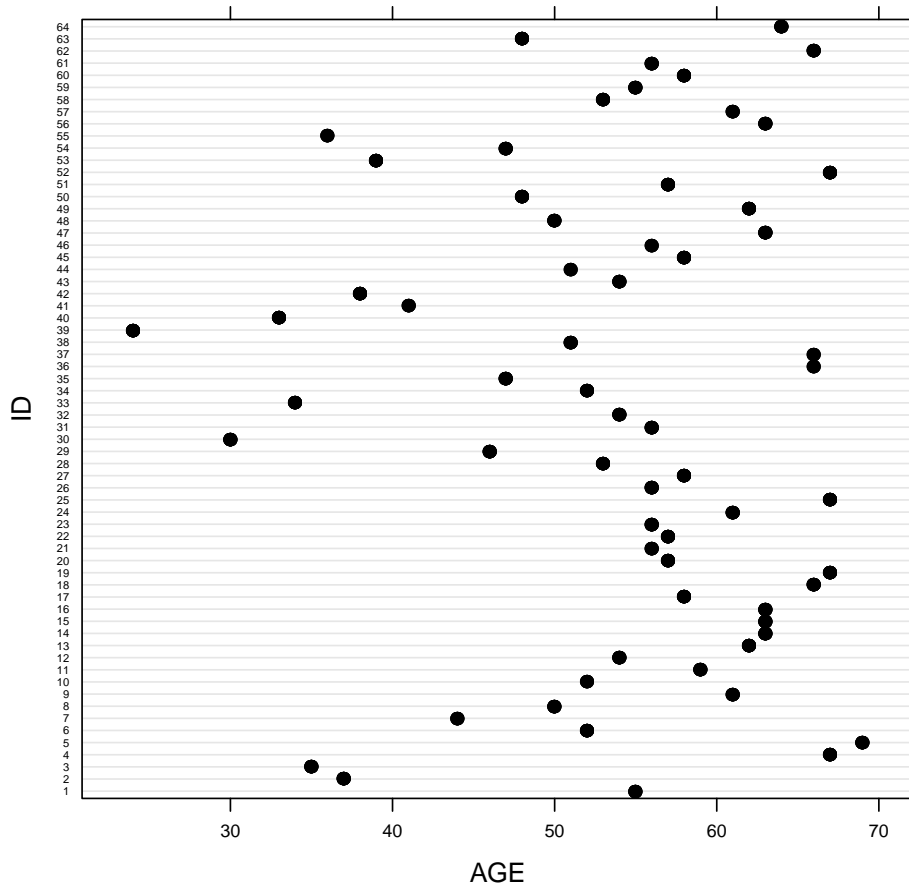
# Data file checkout

Exploratory graphical analysis of the raw data is important and Xpose has numerous functions to support this activity. However, most of this functionality assumes that the data is available in NONMEM table file format. This is usually fine but one needs to be aware that Xpose excludes all lines with `WRES=0` and only the first line for each unique ID number when plotting covariates and/or parameters (this is the default behavior and can be changed by invoking appropriate arguments when executing the specific functions).

One challenge that is related to the exploratory analysis but usually overlooked is checking if the data file is correctly coded, for example making sure that all individuals have the dose records they are supposed to have. Typically such checking is done by “reading” the data file, a tedious and error prone activity. Xpose offers some support with the `data.checkout` function.



## data.checkout



**Figure 1:** *The ID-number versus the values in the AGE column.*

This graph can be used to check the contents of the data file columns. The ID number is plotted versus the values of each row in the column. In this particular case only one panel=column is shown but the function generates one panel per column (excluding the ID column) in the data set. `hlin` specifies the row number in the data file that contains the column headers. Check the help page for further

details.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- data.checkout(obj = xpdb,
+   datafile = "../newdata2.dta",
+   csv = F, hlin = 1, max.plots.per.page = 1)
> print(xplot@plotList[[2]])
```

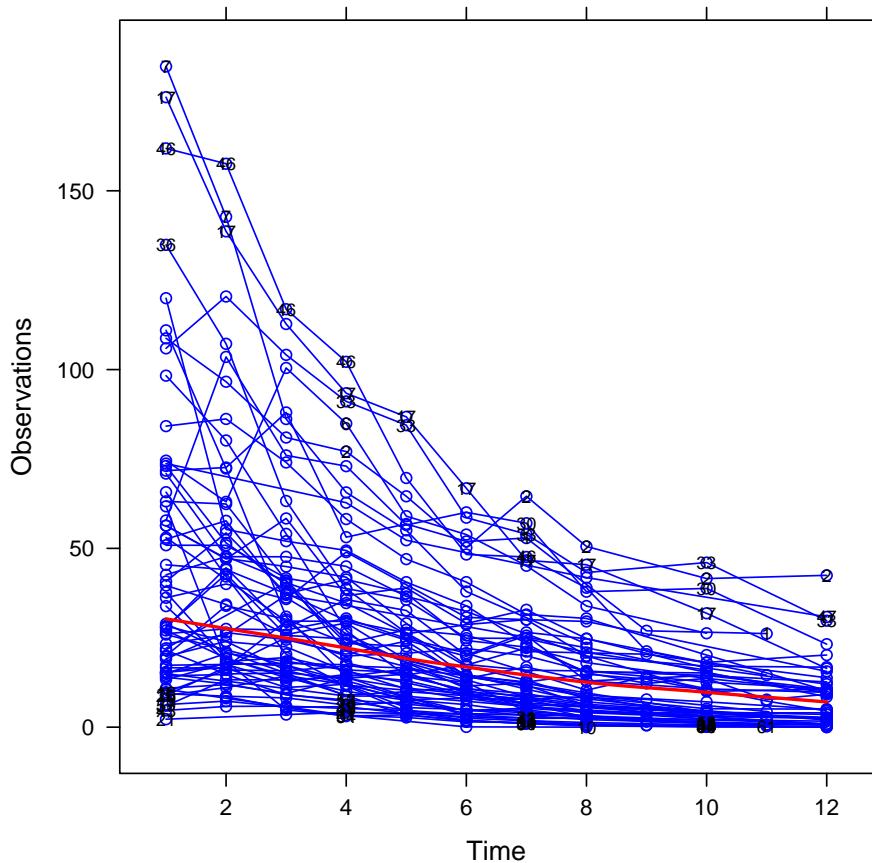
# Data visualization

Some people claim that exploratory analysis of the observed data is a sufficient data analysis methodology and that models offer only marginal additional benefit. There are of course people that argue for the opposite. Regardless, exploratory graphical analysis is a powerful tool which may reveal a huge amount of information. It is sometimes also necessary to demonstrate that conclusions drawn from a model based analysis is supported by the observed data itself. On the other hand, one may argue that the purpose of a model based analysis is to reveal things that are not obvious from the raw data alone. With complicated underlying data structures and dependencies, as in PKPD data, this is even more true.

The graphical displays in this section depend only on the raw data and aim at supporting exploratory graphical analysis to establish candidate models and to establish distributional representations of the underlying study population.

## dv.vs.idv

### Observations vs. Time (Run 1)



**Figure 2:** Observed data versus time. The data points from each individual are connected with a line and each data point is indicated by a circle. Extreme data points (outermost 5%) are labeled with the ID-number of the corresponding individual. The red line is a smooth.

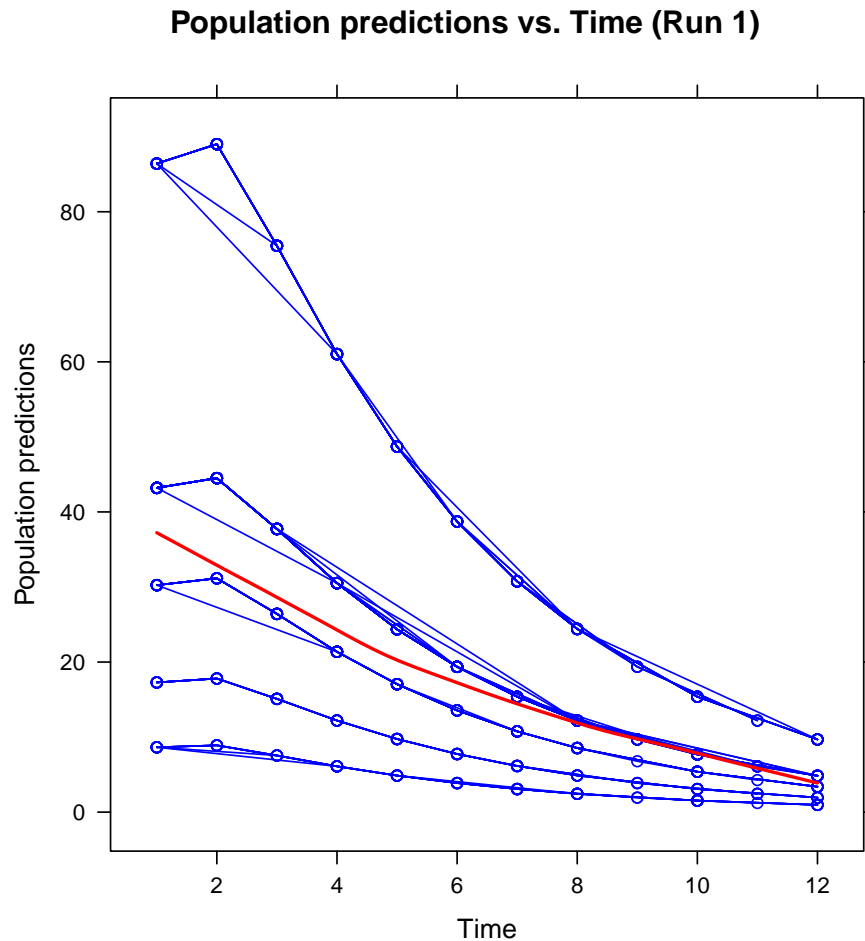
This graph is used for an overall visualization of the data over the main predictor (`idv` or **i**ndependent **v**ariable). To use this graph for initial PK model selection it is useful to use *time after last dose* on the x-axis. The Xpose function `add.tad` can be used to compute this. A useful variation of this graph is to use a semi-

logarithmic scale (`logy=TRUE`), no id-numbers (`ids=FALSE`) and no smooth (`smooth=NULL`):

**Code used to generate the graph:**

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- dv.vs.idv(xpdb, ids = TRUE)
> print(xplot)
```

## pred.vs.idv



**Figure 3:** *Population predictions versus time. The predictions from each individual are connected with a line and each data point is indicated by a circle. The red line is a smooth.*

This graph is used for visualization of the population predictions over the main predictor (*idv*). Its is more useful as method of communicating model properties than goodness-of-fit or diagnostics since there is no reference to which it should be compared. See also

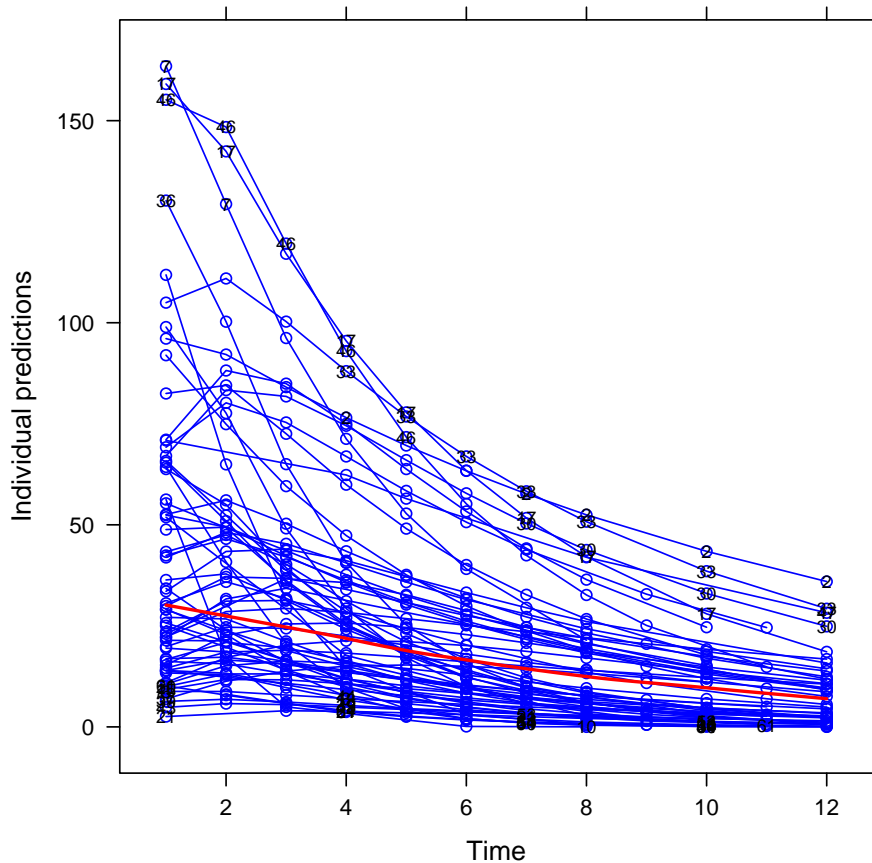
“*dv.preds.vs.idv*” on page 19.

**Code used to generate the graph:**

```
> xpdb <- xpose.data(1, directory = "../")  
> xplot <- pred.vs.idv(xpdb)  
> print(xplot)
```

## ipred.vs.idv

### Individual predictions vs. Time (Run 1)



**Figure 4:** Individual (posterior Bayes) predictions versus time. The predictions from each individual are connected with a line and each data point is indicated by a circle. Extreme data points (outermost 5%) are labeled with the ID-number of the corresponding individual. The red line is a smooth.

This graph is used for visualization of the individual predictions over the main predictor (`idv`). Its usefulness lies more in model communication rather than in goodness-of-fit or diagnostics since there is no reference to which it should be compared. See also

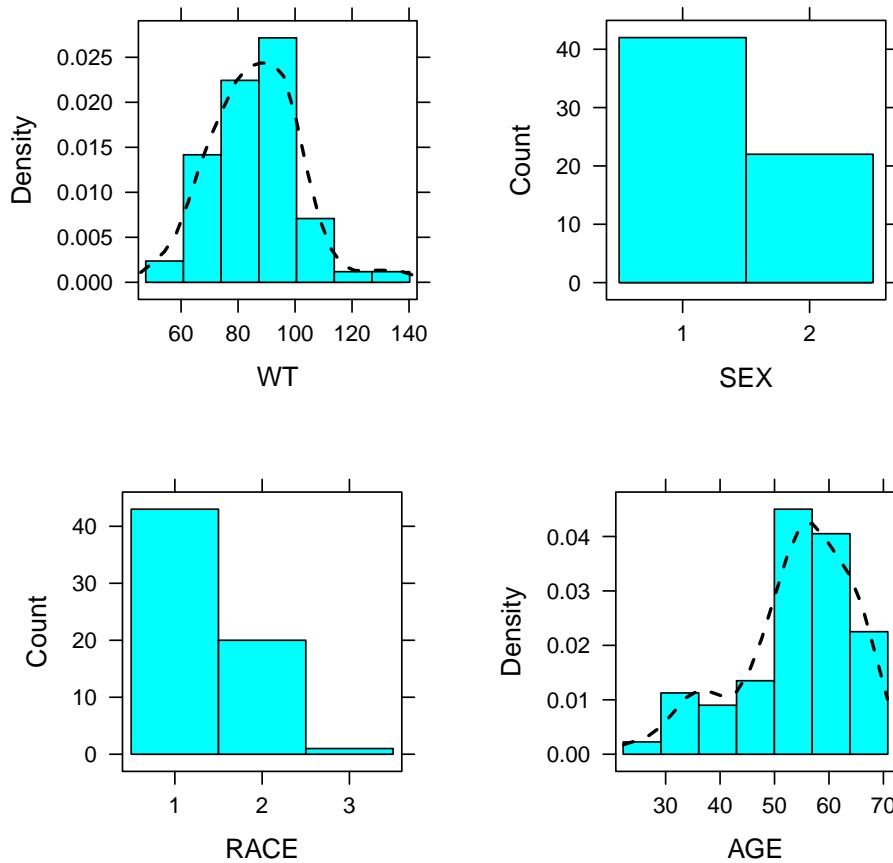
“dv.preds.vs.idv” on page 19.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- ipred.vs.idv(xpdb, ids = TRUE)
> print(xplot)
```

# cov.hist

## Distribution of covariates (Run 1)



**Figure 5:** Histograms of WT, SEX, RACE and AGE. For the continuous covariates the y-axis indicates the probability density and the line is a smoothed representation of this density. For the categorical covariates, the y-axis shows the counts of individuals in the respective covariate category.

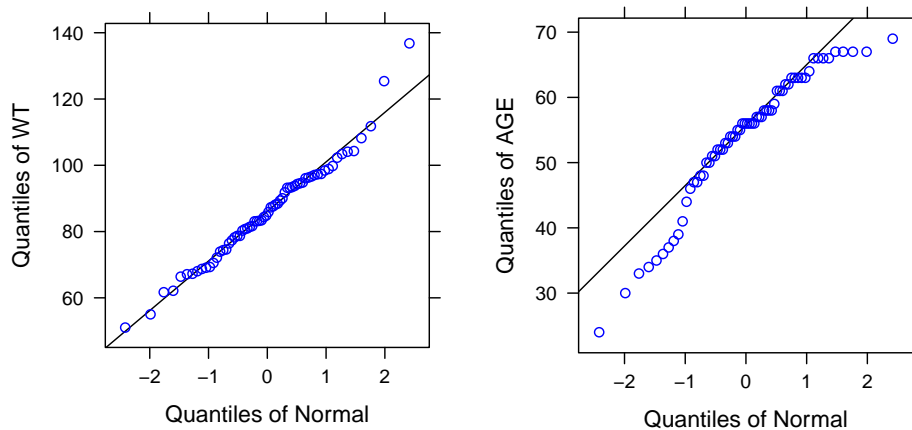
This graph is used for visualization of the distribution of the covariates. By default only the values on the first row of each individual is plotted, i.e. the display shows the distribution of the covariates at baseline (assuming that the first row is at time 0).

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "covariates") <- c("WT",
+     "SEX", "RACE", "AGE")
> xplot <- cov.hist(xpdb)
> print(xplot)
```

# cov.qq

## Distribution of covariates (Run 1)



**Figure 6:** QQ-plots of WT, SEX, RACE and AGE. The quantiles of the distribution of body weight and age are plotted versus quantiles of the Normal distribution. The black diagonal lines is the line of identity.

This graph is used for visualization of the distribution of the covariates. By default only the values on the first row of each individual is plotted, i.e. the display shows the distribution of the covariates at baseline (assuming that the first row is at time 0).

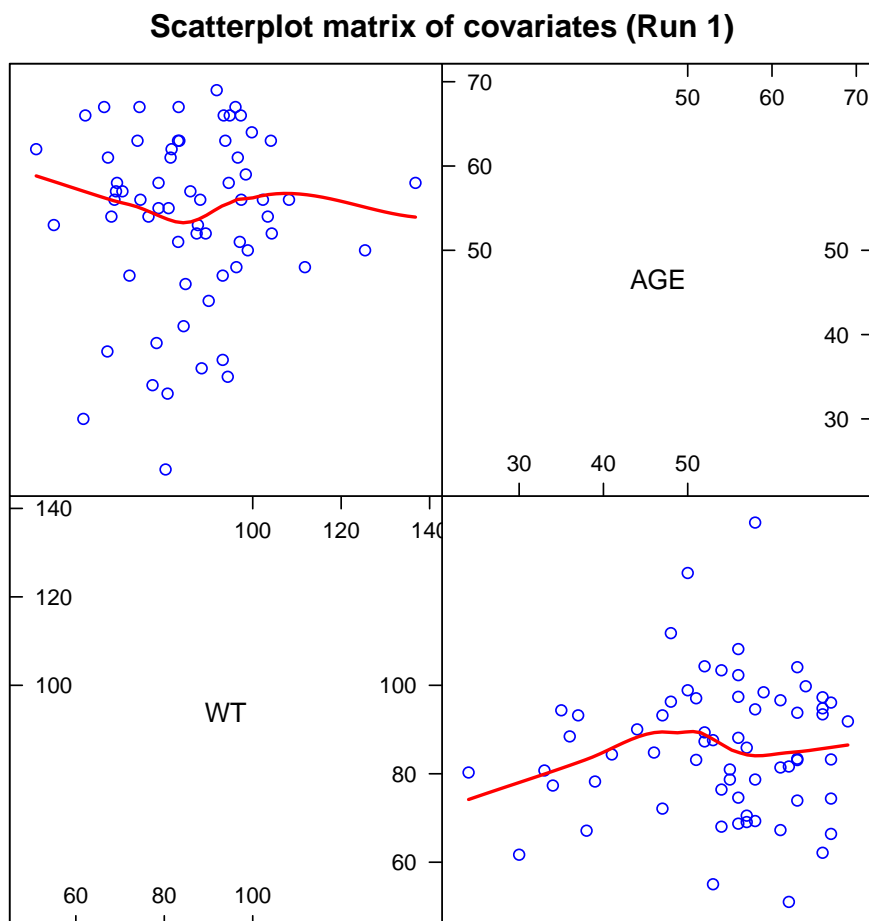
This is a very powerful way of comparing the distribution of a variable to the Normal distribution. For covariates we typically do not make any assumptions regarding the distribution of

covariates so deviations from the line of identity is usually no cause to worry.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "covariates") <- c("WT",
+    "AGE")
> xplot <- cov.qq(xpdb)
> print(xplot)
```

# cov.splom



**Figure 7:** Scatterplot matrix of WT, AGE and SEX. Each panel includes the data from each individual and each pair of covariates are plotted twice, once in the upper triangle and once in the lower. The red lines are smooths.

This graph is useful for investigating the correlation between covariates. As with all covariate plots, the default is to only use the data from the first row of each individual.

**Code used to generate the graph:**

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "covariates") <- c("WT",
+     "AGE", "SEX")
> xplot <- cov.splom(xpdb)
> print(xplot)
```



## cov.summary

```
+-----+-----+-----+
|      |Category|N |%   |
+-----+-----+-----+
|RACE|1      |43|67.2|
|      |2      |20|31.2|
|      |3      |1 |1.6 |
+-----+-----+-----+
|SEX |1      |42|65.6|
|      |2      |22|34.4|
+-----+-----+-----+
```

```
+-----+-----+-----+-----+-----+-----+-----+
|      |Mean|   SD|   Q1|Median|   Q3|      Range| N|
+-----+-----+-----+-----+-----+-----+-----+
| HT|171.2|10.38| 165|   173| 179|   140-188|64|
+-----+-----+-----+-----+-----+-----+-----+
| WT|86.07|15.54|75.97| 85.35|96.15|51.03-136.8|64|
+-----+-----+-----+-----+-----+-----+-----+
| AGE|53.88|10.34| 49.5|   56| 62|   24-69|64|
+-----+-----+-----+-----+-----+-----+-----+
```

This command can be used to summarize the covariate information in the data set. By default, only the covariate values present in the first data record of each individual are used, i.e. the baseline values of the covariates. This behavior can be changed by setting the argument `onlyfirst` to `FALSE`. It is also possible to

re-direct the output to a `csv` file which can be imported into spreadsheet or word processing programs.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> cov.summary(xpdb)
```

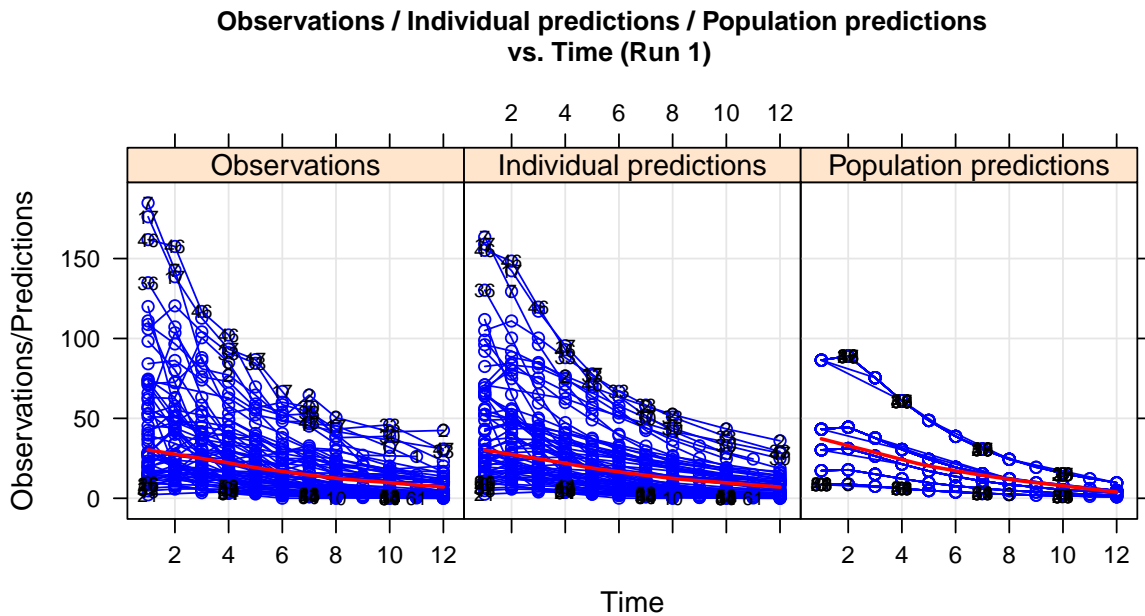
# Overall goodness of fit assessment

Once a model candidate has been fit to the data we need to check if it is any good. The first level of such checks can be quite high level; for example, does a plot of the individual predictions versus time look anything like the real observations versus time. If the model fails at this level it probably needs some substantial changes and we don't need to worry about minor deficiencies in the residual error model.

Sometimes one graph is not enough for this high level check of the model and we may need composite displays of two or more graphs, for example the output from the `basic.gof` function.

Overall goodness of fitness plots are also well suited for communicating the appropriateness of the model to a wider audience in a succinct manner.

# dv.preds.vs.idv



**Figure 8:** Observed data, individual and population predictions versus time. The data points from each individual are connected with a line and each data point is indicated by a circle. Extreme data points (outermost 5%) are labeled with the ID-number of the corresponding individual. The red line is a smooth.

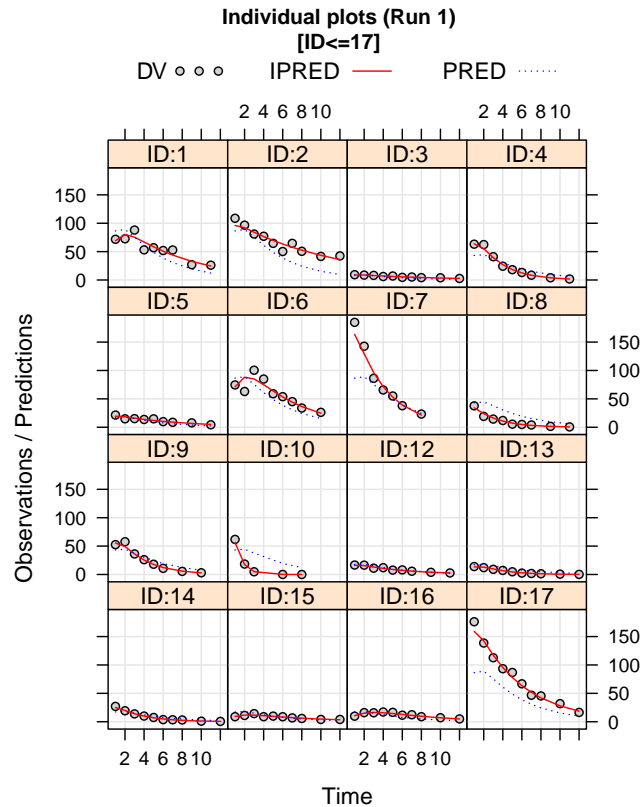
This graph is used for an overall visualization of the performance of the model. If the two right panels (the individual and population predictions respectively) closely resembles the left panel (the observed data), there is not much more the model can be improved. The difference between the middle and right hand panel gives an indication of the maxi-

mum improvement the inclusion of covariates may have.

## Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- dv.preds.vs.idv(xpdb,
+   grid = TRUE, ids = TRUE)
> print(xplot)
```

# ind.plots



**Figure 9:** The observed data, individual and population predictions versus time for each individual are plotted in separate panels. The ID-number of each individual is given in the strip above each panel. The predictions are connected with a line and the observed data points are indicated by symbols.

This graph is used for an overall visualization of the performance of the model. As an overall visualization of a fit it is useful, especially for single dose PK profiles, or if time after dose is used as the independent variable. The plots can be misleading, especially if there is high shrinkage, leading to the appearance of a “perfect fit” between the individual predictions and the observed data. Additionally, one “diagnostic signal” in the plot is the differences between observed and predicted values, some-

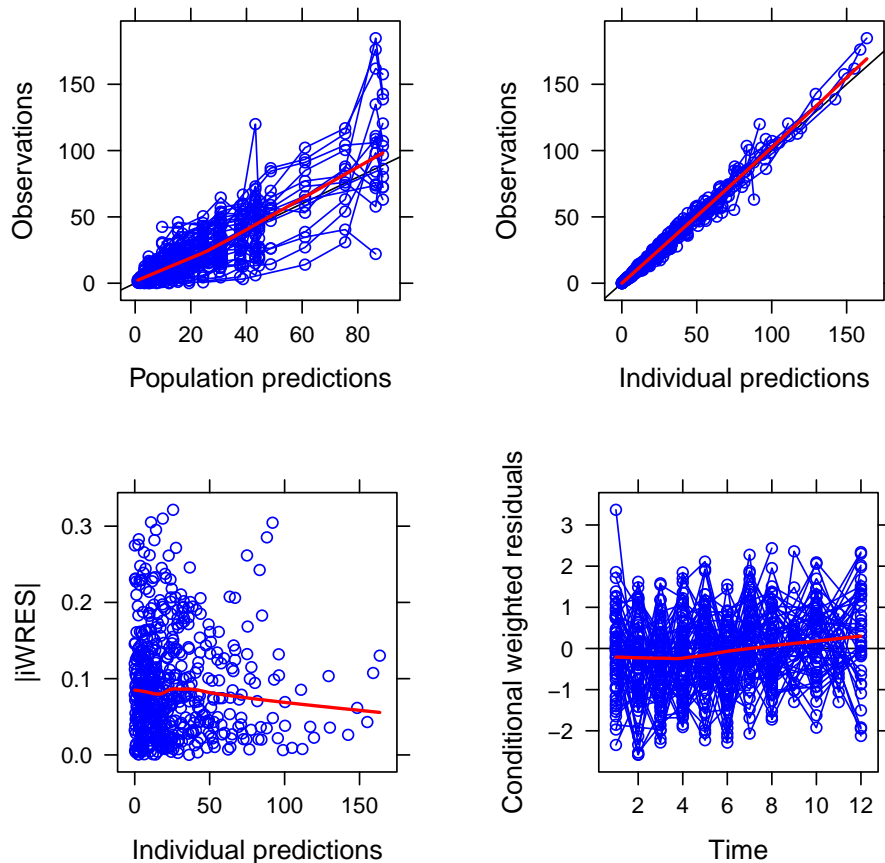
thing that is not the focus of this display. Residual and/or observed vs predicted plots are better diagnostic tools to highlight these differences.

## Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- ind.plots(xpdb, grid = TRUE,
+   subset = "ID<=17")
> print(xplot)
```

# basic.gof

## Basic goodness-of-fit plots (Run 1)



**Figure 10:** Basic goodness of fit plots. The top panels show the observed data versus the population and individual predictions, respectively. The bottom left panel shows the absolute values of the individually weighted residuals and the bottom right panel shows the population conditional weighted residuals. Individual data points are indicated by blue circles and are connected by blue lines (except bottom left panel). Extreme values are labeled by the corresponding ID-number. The red lines are smooths, the black diagonal lines (top panels) are the lines of identity and the black horizontal line (bottom right) is the zero line.

Overall goodness of fit assessment usually means considering multiple aspects of the performance of the model. Typically this involves inspecting multiple graphs. In this display four mainstay goodness of fit plots are collected on the same page (see below).

In this version of this display, the conditional weighted residuals are used in the bottom right

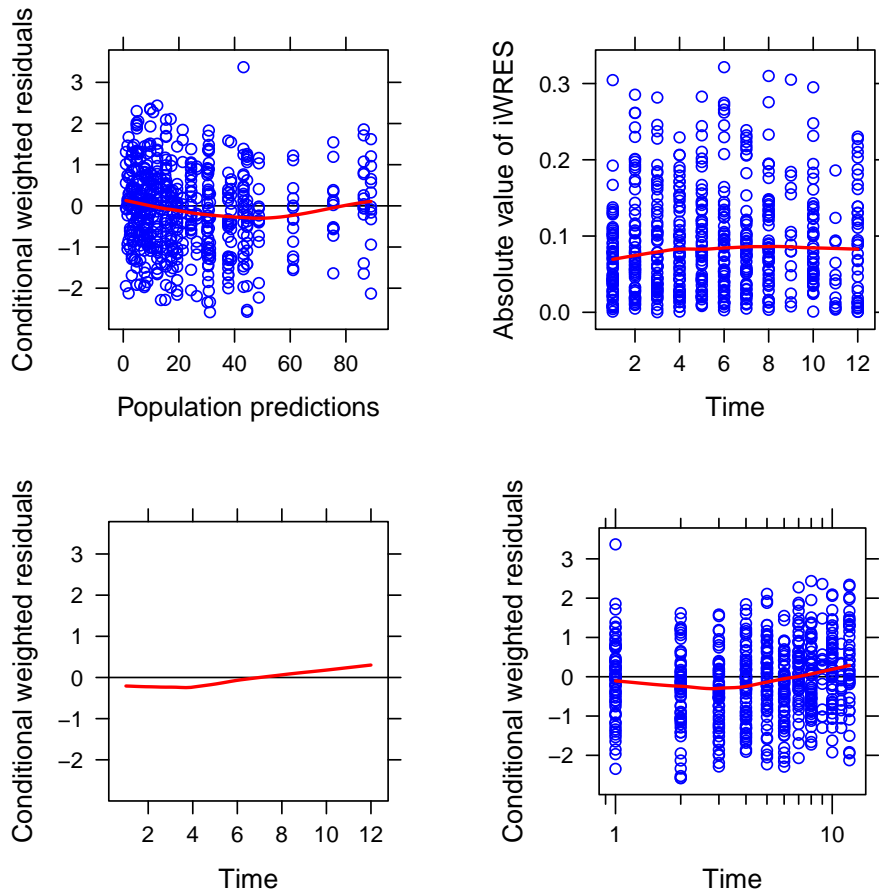
panel. The argument `force.wres=TRUE` to the `basic.gof` function determines which version of the weighted residuals are used.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- basic.gof(xpdb)
> print(xplot)
```

# addit.gof

## Additional goodness of fit plots (Run 1)



**Figure 11:** Additional goodness of fit plots. The top left panel show the conditional weighted residuals versus the population predictions, the top right panel shows the absolute values of the individual weighted residuals versus time, the bottom left panel shows the smooth of the conditional weighted residuals versus time (data points are suppressed) and the bottom right panel shows the conditional weighted residuals versus the logarithm of time. Individual data points are connected by blue circles and extreme values are labeled by the corresponding ID-number.

This display shows another set of goodness of fit plots. The two top panels are elaborated on elsewhere. The bottom panels are a complement to the bottom right panel of the basic goodness of fit plots. In this version the data points are omitted to facilitate the comparison of the smooth and the zero line (which ideally should overlap). The bottom right panel has

the `idv(=x)`-axis on the log-scale. The latter is useful if there are large x-values that dominate the display.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- addit.gof(xpdb)
> print(xplot)
```

# runsum

## Summary of run 1

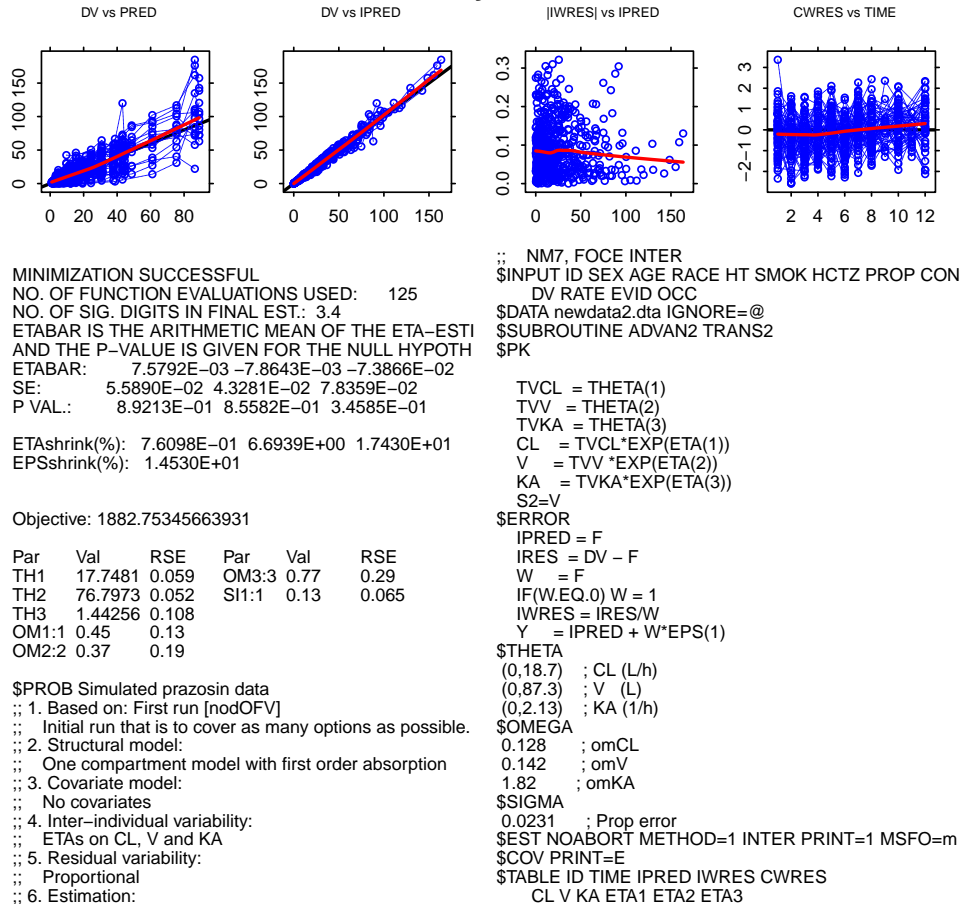


Figure 12: Summary of run 1.

This display is intended as a (printed) one side summary of a run. It combines the basic goodness of fit graphs (see section “basic.gof” on page 21), some NONMEM output and the model file. (The model file is allotted the remaining space after the graphs and NONMEM output have been included. If there is not enough space the model file will be truncated.) The parameter estimates and the relative standard

errors (RSE) are given, where the latter is the SE/estimate. The variability parameter estimates are converted to standard deviations and correlations (but the RSE will still refer to the variance and covariance estimates).

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> runsum(xpdb, dir = "../")
```

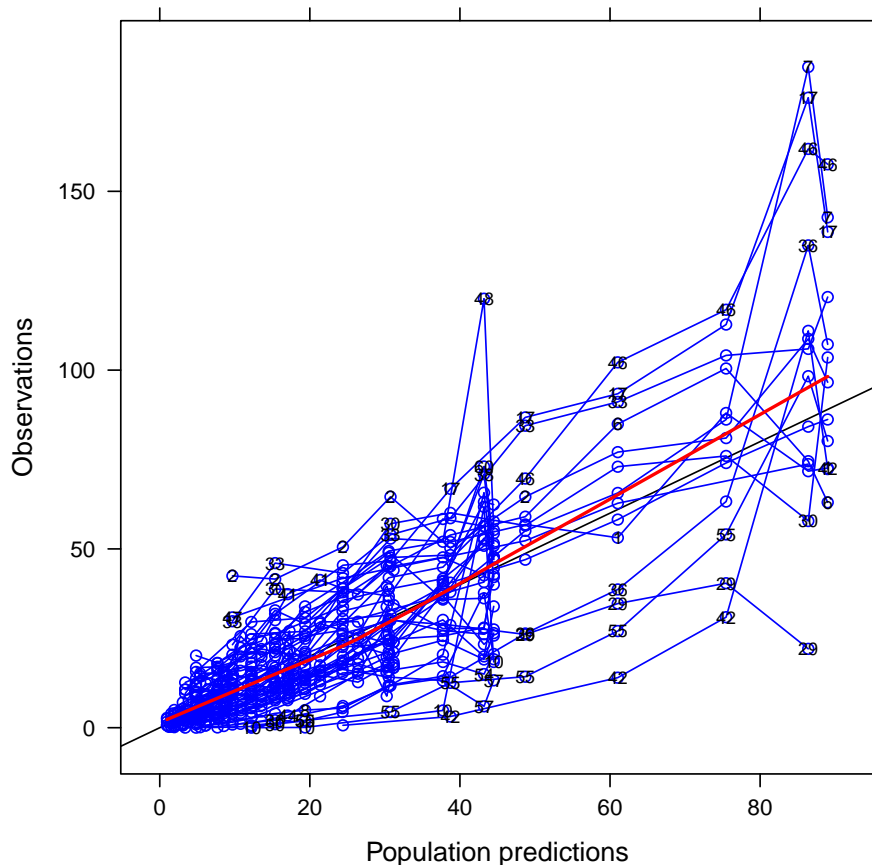
# Structural model diagnostics

The first step in a typical model development process is to get the structural (fixed effects) model describing the data well. If the overall diagnostic plots indicate that there are some issues with the model, a reasonable first step is typically to investigate if the structural model is appropriate.



# dv.vs.pred

## Observations vs. Population predictions (Run 1)



**Figure 13:** The observed data versus the population predictions. The data points from each individual are connected with a line and each data point is indicated by a circle. Extreme data points (outmost 5% of the data) are labeled with the ID-number of the corresponding individual. The black line is the line of identity and the red line is a smooth.

This graph is a mainstay goodness of fit plot. The aim is for the line of identity to go through the middle of the data density. A good sign is if the smooth and the line of identity overlap, although this may occur even if the model correctly describes the data. If the structural model does not include important covari-

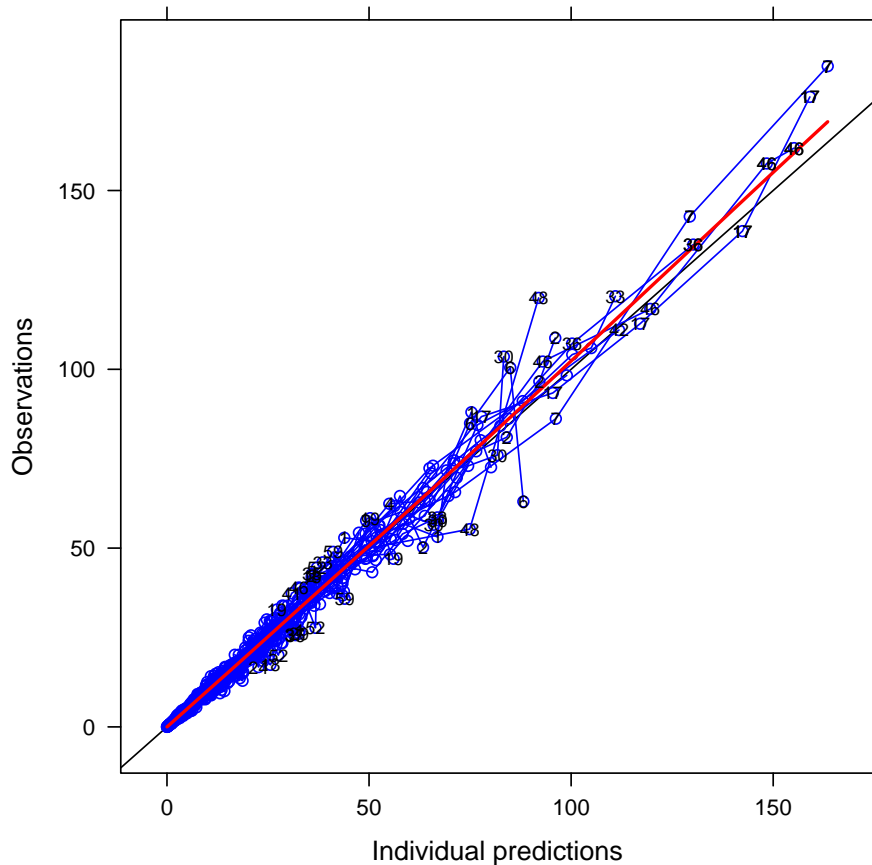
ates, the graph can have a marked vertical "chopped" look to the far right.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- dv.vs.pred(xpdb, ids = TRUE)
> print(xplot)
```

# dv.vs.ipred

### Observations vs. Individual predictions (Run 1)



**Figure 14:** The observed data versus the individual predictions. The data points from each individual are connected with a line and each data point is indicated by a circle. Extreme data points are labeled with the ID-number of the corresponding individual (outermost 5% of the data). The black line is the line of identity and the red line is a smooth.

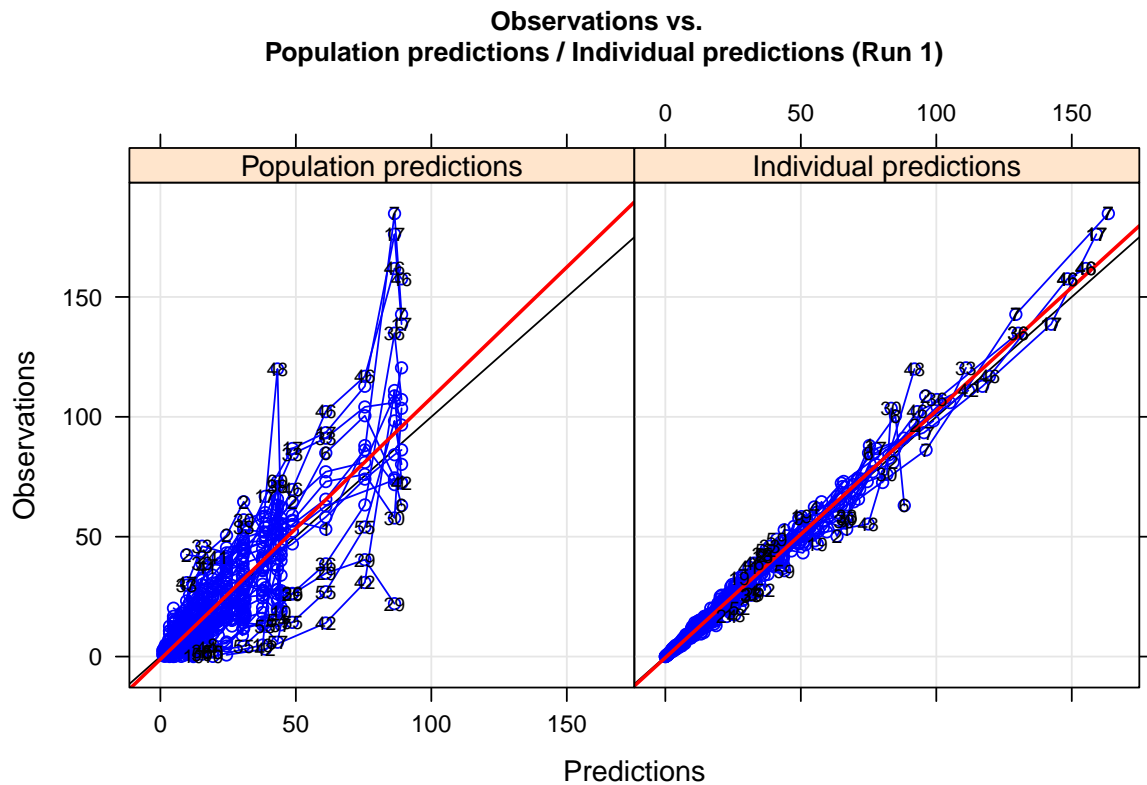
This graph can be used for overall goodness of fit assessment if the epsilon shrinkage is low. The aim is for the line of identity to go through the middle of the data density. A good sign is if the smooth and the line of identity overlap. If the degree of shrinkage is low this can be a very informative graph. If there is high shrink-

age (over 30%) then this plot may show the misleading “perfect fit” phenomena.

**Code used to generate the graph:**

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- dv.vs.ipred(xpdb, ids = TRUE)
> print(xplot)
```

# dv.vs.pred.ipred



**Figure 15:** The observed data versus the population and individual individual predictions. The data points from each individual are connected with a line and each data point is indicated by a circle. Extreme data points are labeled with the ID-number of the corresponding individual. The black line is the line of identity and the red line is a smooth.

This is the combination of the previous two graphs and is also used for overall goodness of fit assessment. Apart from the obvious compactness of the display, this graph has the added benefit of providing a direct comparison of the populations and individual prediction. For example, it may give an indication as to how much (or little) of the variability the fixed

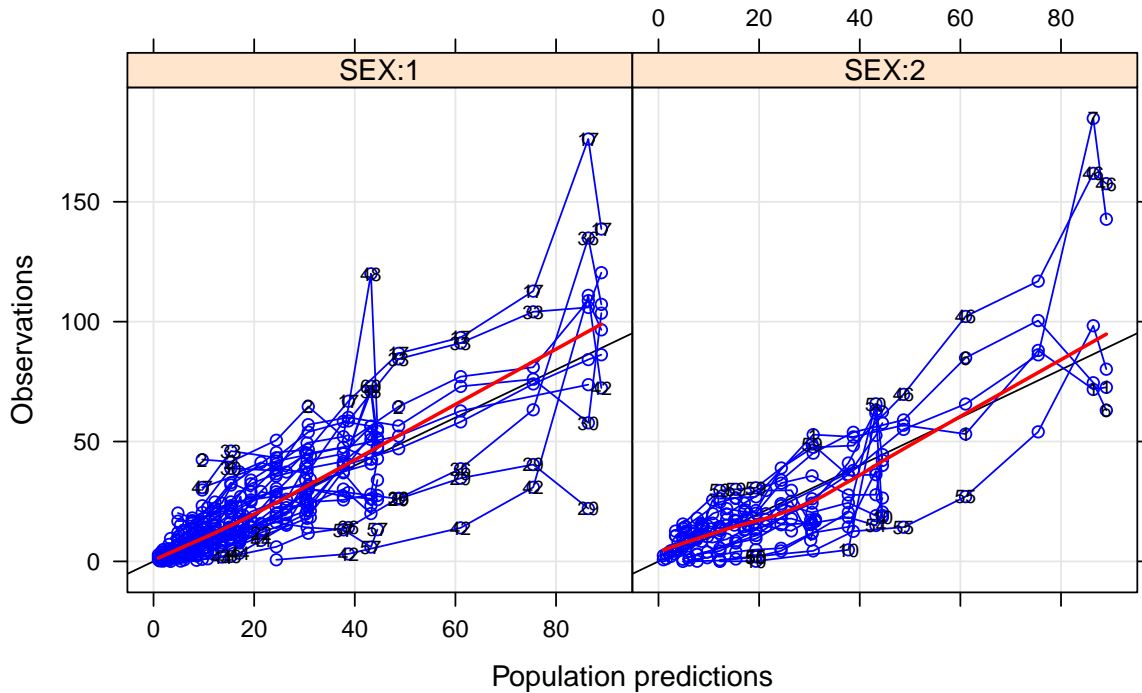
effect part of the model explains.

## Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- dv.vs.pred.ipred(xpdb,
+   grid = TRUE, ids = TRUE)
> print(xplot)
```

# dv.vs.pred.by.cov

## Observations vs Population predictions (Run 1)



**Figure 16:** The observed data versus the population predictions given SEX. The data points from each individual are connected with a line and each data point is indicated by a circle. Extreme data points are labeled with the ID-number of the corresponding individual. The black line is the line of identity and the red line is a smooth.

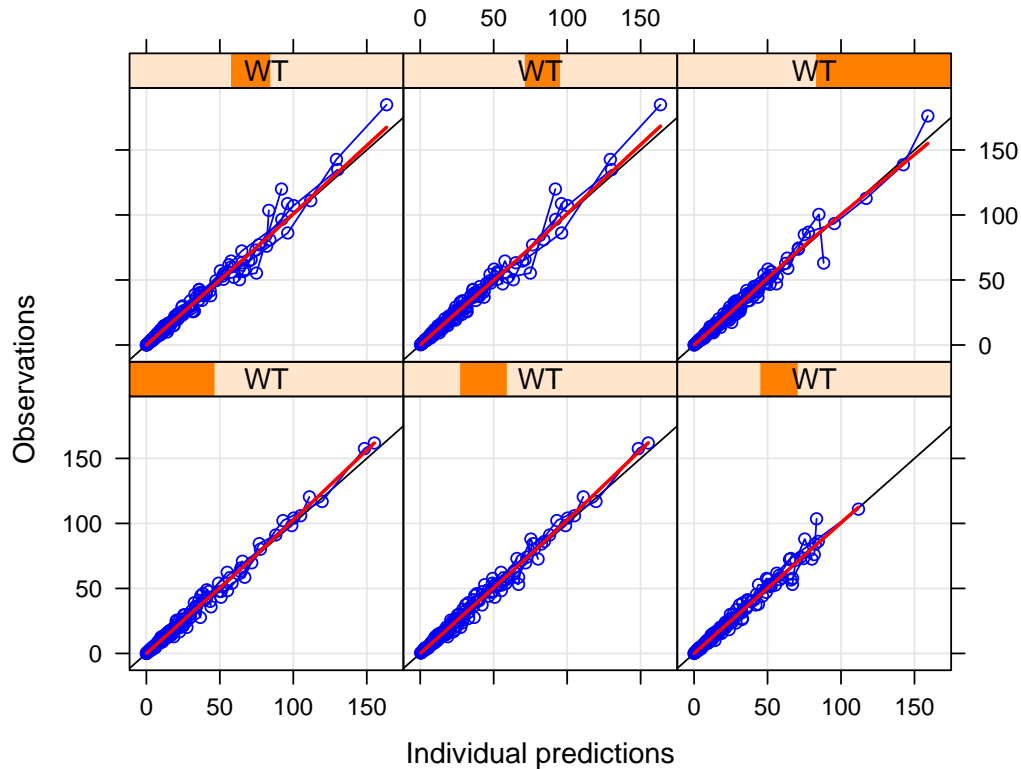
This graph is an extension to the DV versus PRED graph (Section “dv.vs.pred” on page 25). The purpose is to check the goodness of fit in different subsets of the data as defined by one or more covariates.

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "covariates") <- c("SEX")
> xplot <- dv.vs.pred.by.cov(xpdb,
+   grid = TRUE, ids = TRUE)
> print(xplot)
```

**Code used to generate the graph:**

# dv.vs.ipred.by.cov

## Observations vs Individual predictions (Run 1)



**Figure 17:** The observed data versus the population predictions given intervals of WT. The data points from each individual are connected with a line and each data point is indicated by a circle. The black line is the line of identity and the red line is a smooth.

This graph is an extension to the DV versus IPRED graph (Section “dv.vs.ipred” on page 26). The purpose is to check the goodness of fit in different subsets of the data as defined by one or more covariates. In this case the covariate is continuous so the subsets are defined by intervals of the covariate. The intervals are defined using the `equal.count` function in R and have by default approximately the

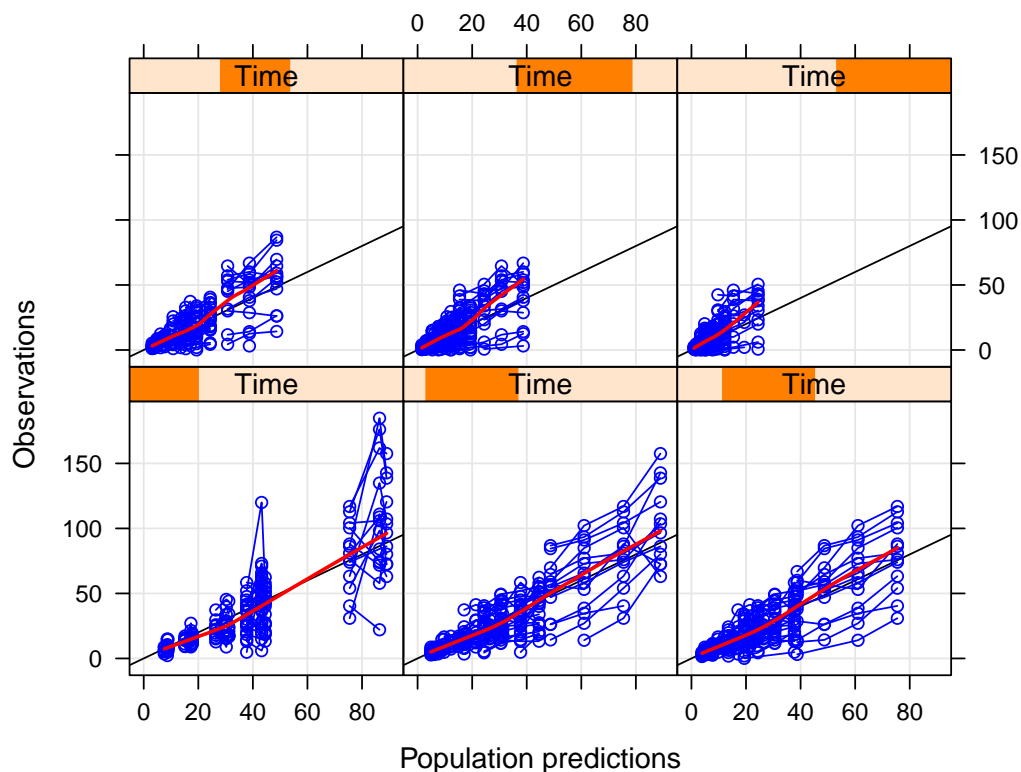
same number of data points in each interval (intervals overlap by 50%).

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "covariates") <- c("WT")
> xplot <- dv.vs.ipred.by.cov(xpdb,
+   grid = TRUE)
> print(xplot)
```

## dv.vs.pred.by.idv

Observations vs. Population predictions (Run 1)



**Figure 18:** The observed data versus the population predictions given intervals of TIME. The data points from each individual are connected with a line and each data point is indicated by a circle. The black line is the line of identity and the red line is a smooth.

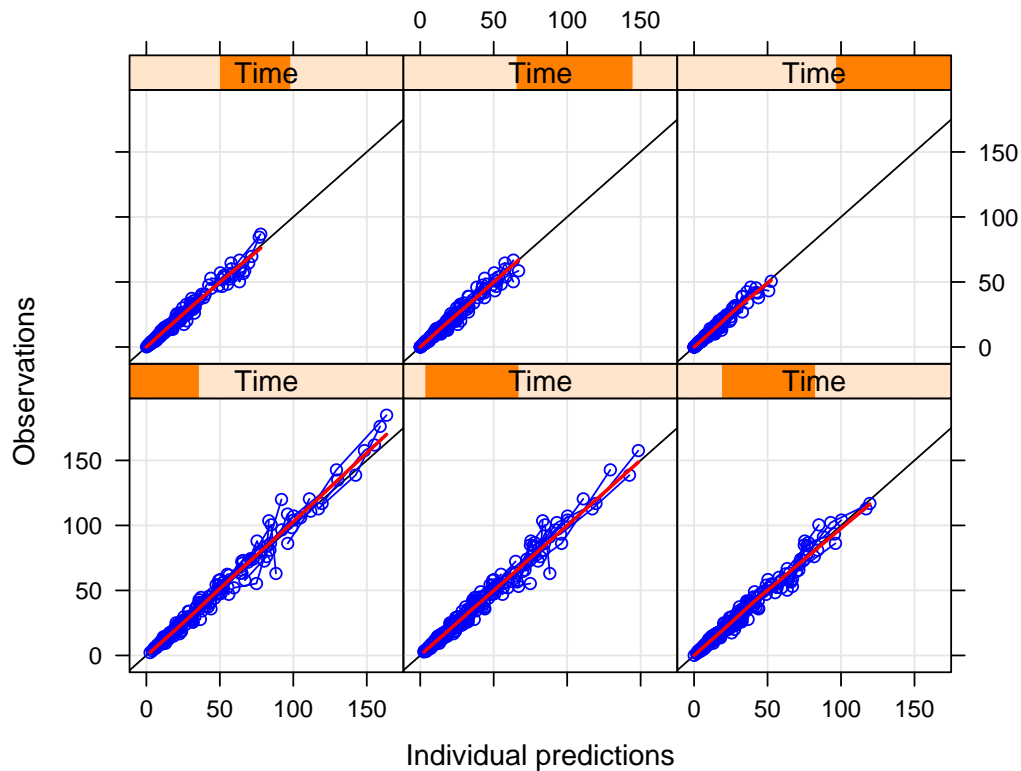
This graph is an extension to the DV versus PRED by covariates graph (Section “dv.vs.pred.by.cov” on page 28). The purpose is to check the goodness of fit in different subsets of the data as defined by intervals if the independent variable. The intervals are defined as described on page 28.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- dv.vs.pred.by.idv(xpdb,
+   grid = TRUE)
> print(xplot)
```

## dv.vs.ipred.by.idv

### Observations vs. Individual predictions (Run 1)



**Figure 19:** The observed data versus the population predictions given intervals of TIME. The data points from each individual are connected with a line and each data point is indicated by a circle. The black line is the line of identity and the red line is a smooth.

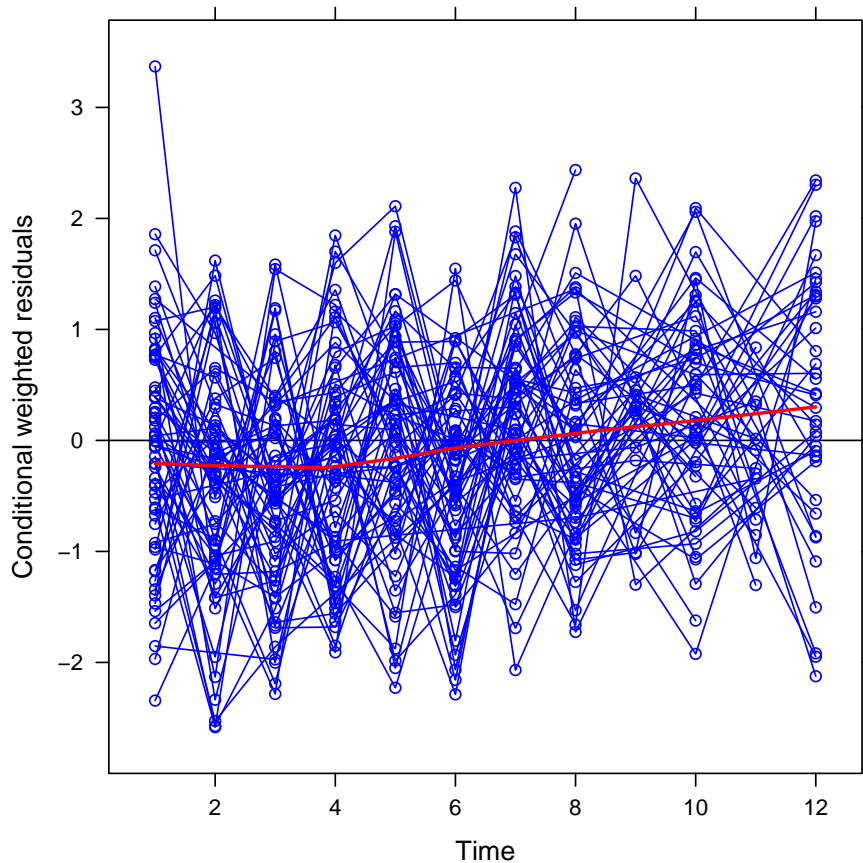
This graph is an extension to the DV versus IPRED by covariates graph (Section “dv.vs.ipred.by.cov” on page 29). The purpose is to check the goodness of fit in different subsets of the data as defined by intervals of the independent variable. The intervals are defined as described on page 28.

#### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- dv.vs.ipred.by.idv(xpdb,
+   grid = TRUE)
> print(xplot)
```

## cwres.vs.idv

Conditional weighted residuals vs. Time (Run 1)



**Figure 20:** *The conditional weighted residuals versus time (the independent variable). Each residual is indicated by a blue circle. The red line is a smooth.*

This graph is intended to check if the CWRES are independent of the independent variable (the independence of the residuals to the main predictor is a fundamental assumption in all regression analyses). A horizontal smooth centered on the horizontal zero line is an indica-

tion that this is the case.

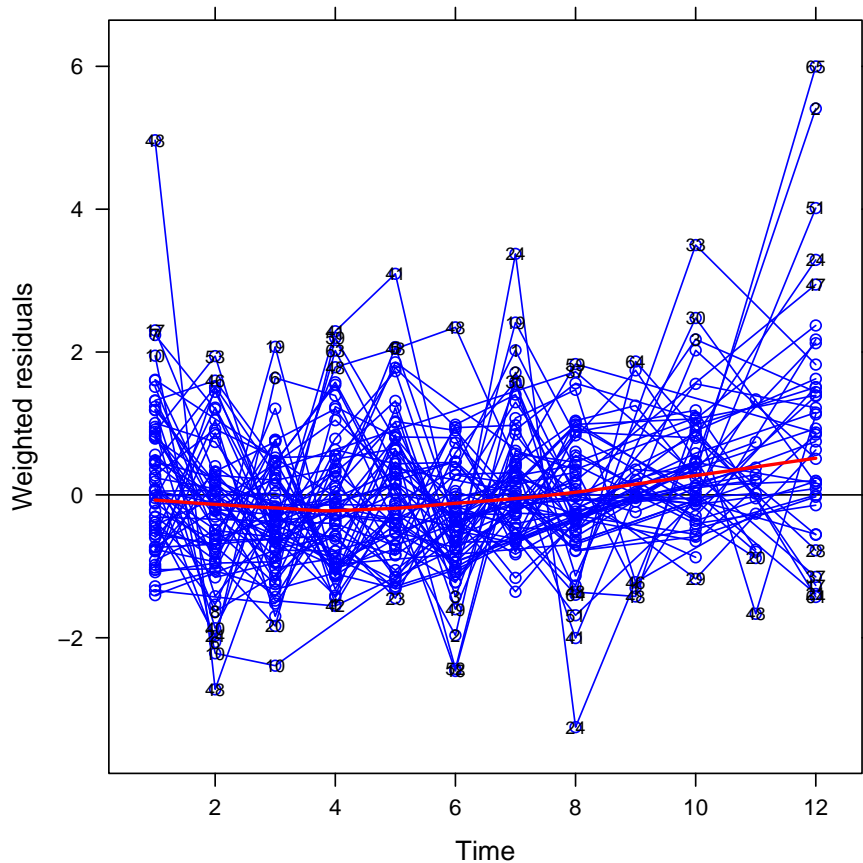
### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- cwres.vs.idv(xpdb)
> print(xplot)
```



## wres.vs.idv

Weighted residuals vs. Time (Run 1)



**Figure 21:** The weighted residuals versus time (the independent variable). Each residual is indicated by a blue circle. Extreme data points are labeled with the ID-number of the corresponding individual. The red line is a smooth.

This graph is intended to check if the WRES are independent of the independent variable (the independence of the residuals to the main predictor is a fundamental assumption in all regression analyses). A horizontal smooth centered on the horizontal zero line is an indication that this is the case.

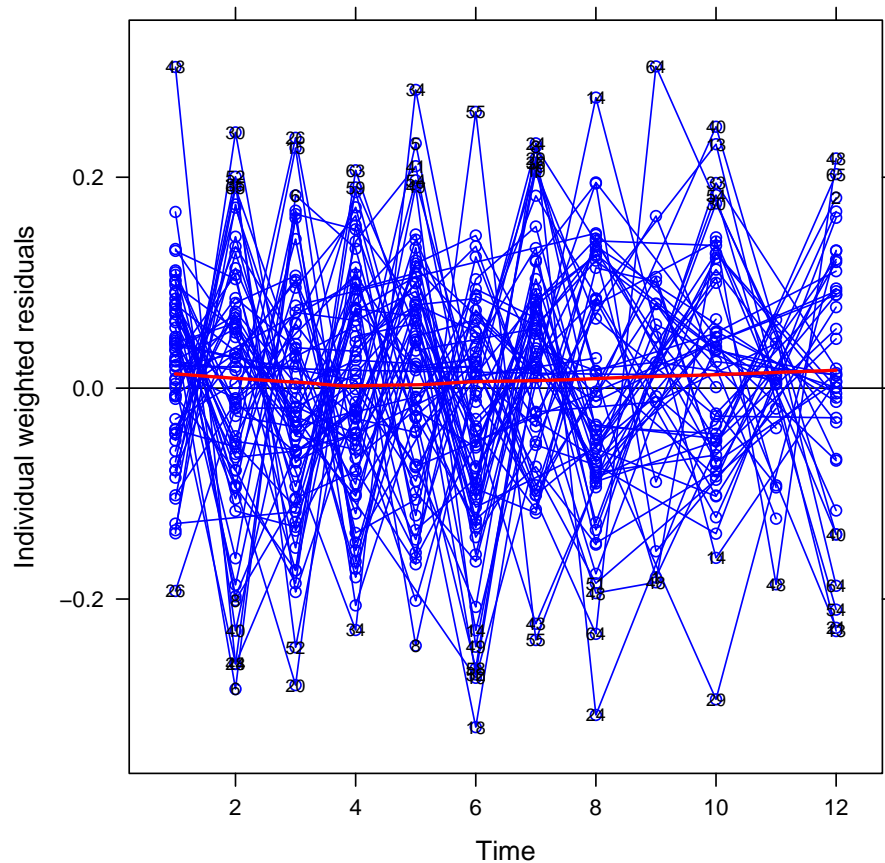
It is more correct to use CWRES instead of WRES (Section “cwres.vs.cov” on page 43).

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- wres.vs.idv(xpdb, ids = TRUE)
> print(xplot)
```

# iwres.vs.idv

### Individual weighted residuals vs. Time (Run 1)



**Figure 22:** The individual weighted residuals versus time (the independent variable). Each residual is indicated by a blue circle. Extreme data points are labeled with the ID-number of the corresponding individual. The red line is a smooth.

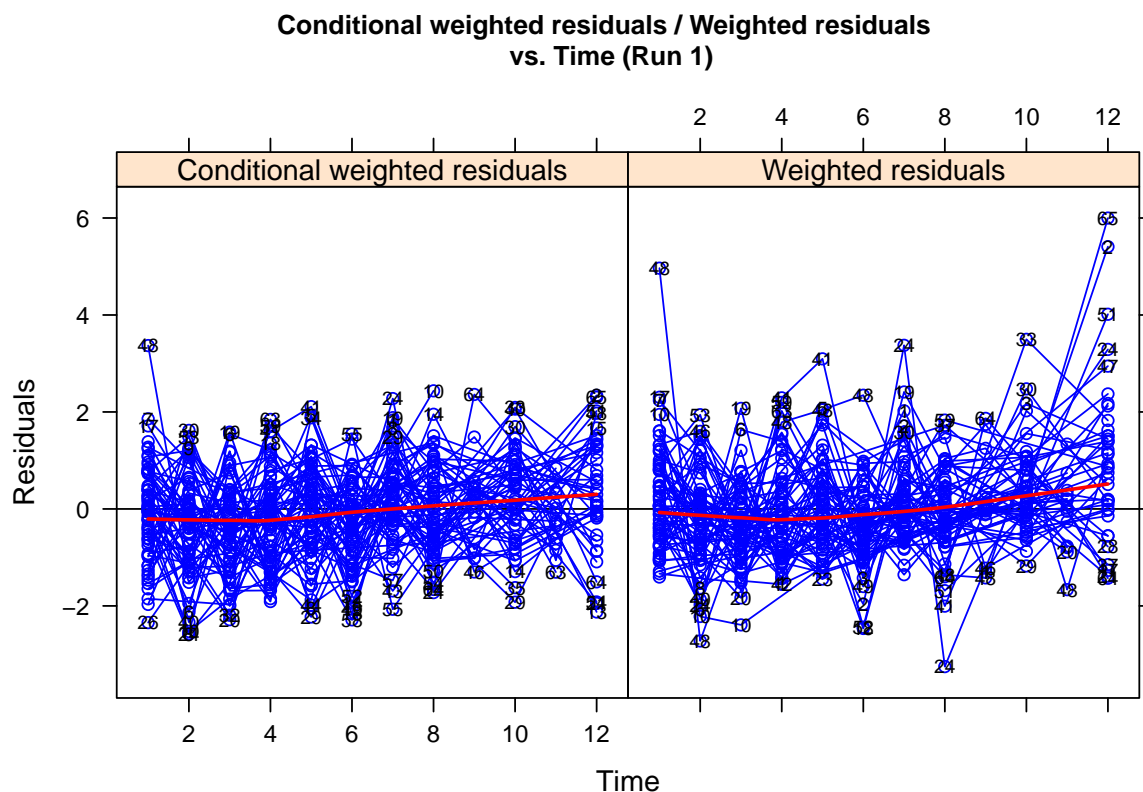
This graph is intended to check if the IWRES are independent of the independent variable (the independence of the residuals to the main predictor is a fundamental assumption in all regression analyses). A horizontal smooth centered on the horizontal zero line is an indica-

tion that this is the case.

**Code used to generate the graph:**

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- iwres.vs.idv(xpdb, ids = TRUE)
> print(xplot)
```

## cwres.wres.vs.idv



**Figure 23:** *The conditional weighted residuals and weighted residuals versus time (the independent variable). Each residual is indicated by a blue circle. Extreme data points are labeled with the ID-number of the corresponding individual. The red line is a smooth.*

This graph is intended to check if the weighted residuals (conditional or not) are independent of the independent variable. Horizontal smooths centered on the horizontal zero lines are an indication that this is the case.

Apart from the model development usage of this graph, it also provides an evaluation of the necessity of using CWRES instead of WRES. This can be useful if the computation of the CWRES is cumbersome. With modern ver-

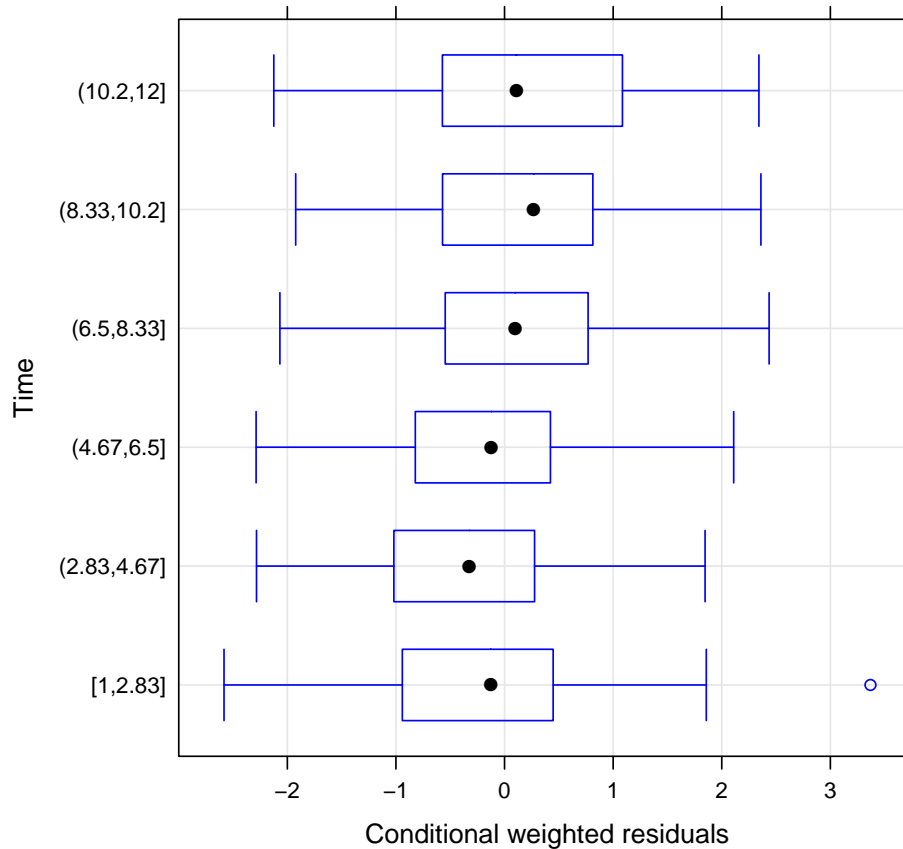
sions of NONMEM (7 and beyond), CWRES can be easily obtained and there is really no reason not to use CWRES.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- cwres.wres.vs.idv(xpdb,
+   ids = TRUE)
> print(xplot)
```

## cwres.vs.idv.bw

Time vs. Conditional weighted residuals (Run 1)



**Figure 24:** *Intervals of time (the independent variable) versus the conditional weighted residuals. The distribution of the residuals for each time interval is visualized by a box and whisker plot.*

This graph is intended to check if the CWRES are independent of the independent variable (the independence of the residuals to the main predictor is a fundamental assumption in all regression analyses). It is similar to the xyplot with CWRES versus time (see Section “cwres.vs.idv” on page 32) but may be more useful when there is a lot of data points. The solid dots in the middle of the boxes indicate the medians of the residual distributions and if the residuals are independent of time, they should fall close and randomly around the ver-

tical zero line.

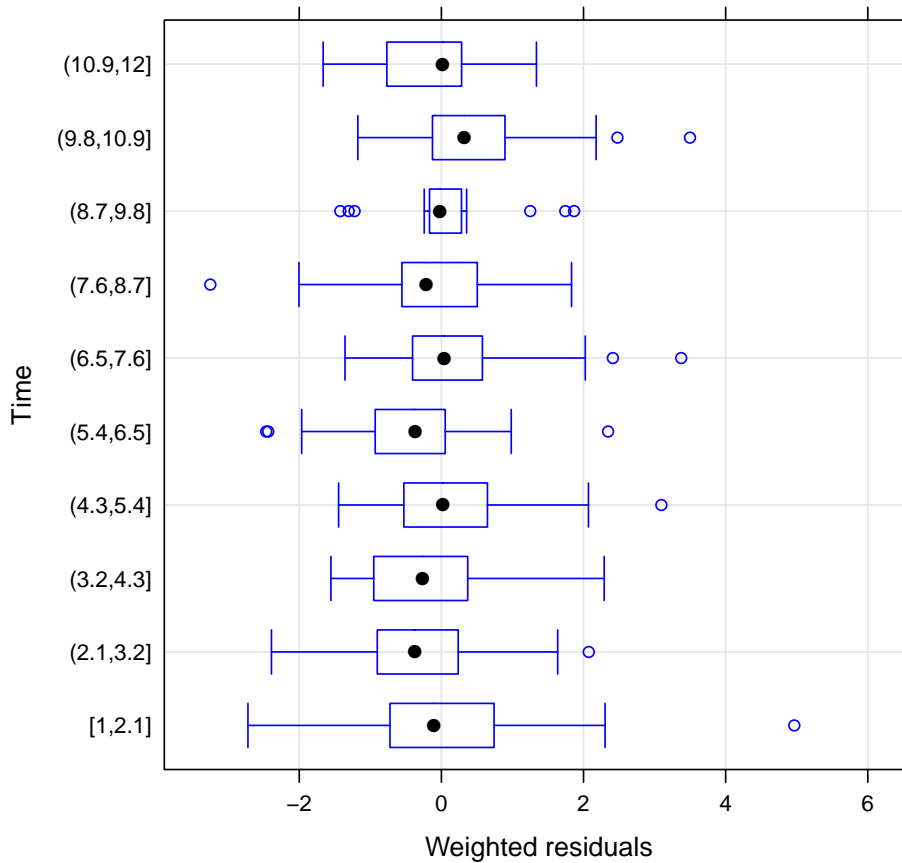
The number of time intervals may have to be adjusted (with the `bins` argument) to something that is appropriate in relation to the number of observations.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- cwres.vs.idv.bw(xpdb,
+   bins = 6)
> print(xplot)
```

# wres.vs.idv.bw

## Time vs. Weighted residuals (Run 1)



**Figure 25:** *Intervals of time (the independent variable) versus the weighted residuals. The distribution of the residuals for each time interval is visualized by a box and whisker plot.*

This graph is intended to check if the WRES are independent of the independent variable (the independence of the residuals to the main predictor is a fundamental assumption in all regression analyses). It is similar to the xplot with WRES versus time (see Section “wres.vs.idv” on page 33) but may be more useful when there is a lot of data points. The solid dots in the middle of the boxes indicate the medians of the residual distributions and if the residuals are independent of time, they should fall close and randomly around the vertical zero line.

The time intervals are constructed such that there is an approximately equal number of data points in each boxplot.

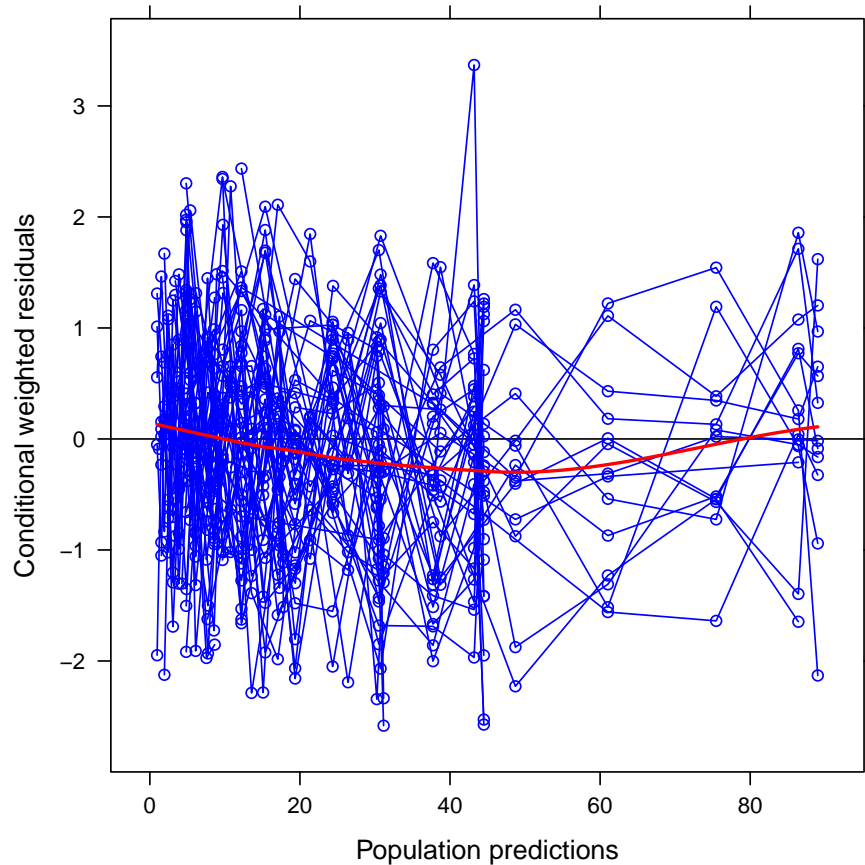
It is more correct to use CWRES instead of WRES (Section “cwres.vs.idv.bw” on the previous page).

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- wres.vs.idv.bw(xpdb)
> print(xplot)
```

## cwres.vs.pred

Conditional weighted residuals vs. Population predictions (Run 1)



**Figure 26:** *The conditional weighted residuals versus the population predictions. Each residual is indicated by a blue circle. The red line is a smooth.*

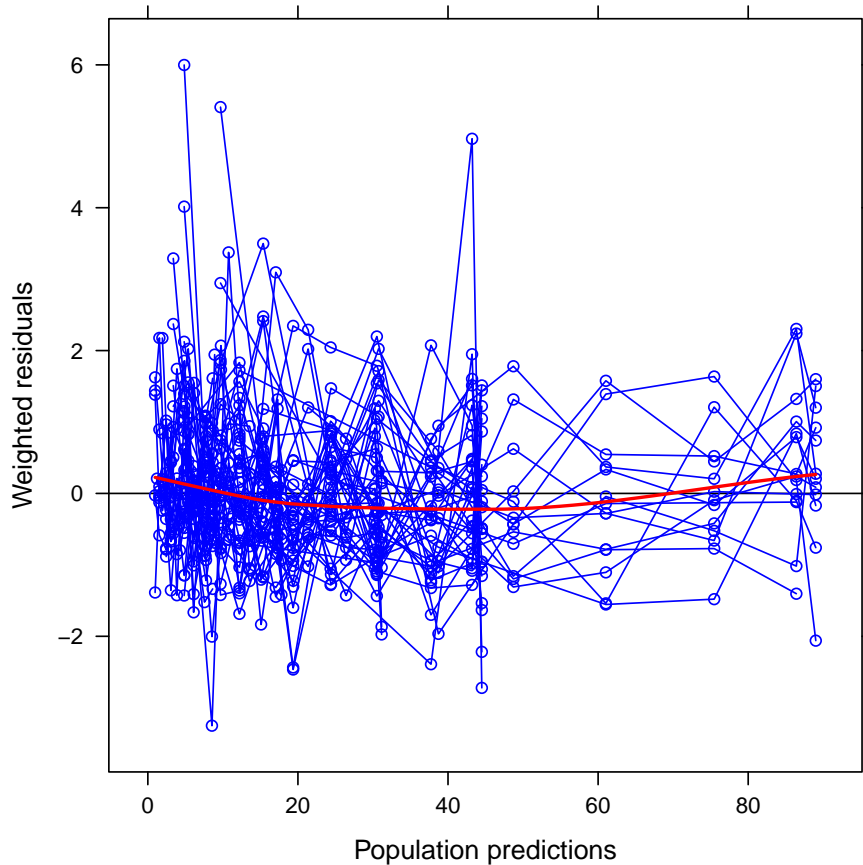
This graph is intended to check if there are any trends in the residuals with the population predictions. The aim is to have no trends, i.e. that the smooth is horizontal.

**Code used to generate the graph:**

```
> xpdb <- xpose.data(1, directory = "../")  
> xplot <- cwres.vs.pred(xpdb)  
> print(xplot)
```

# wres.vs.pred

## Weighted residuals vs. Population predictions (Run 1)



**Figure 27:** The weighted residuals versus the population predictions. Each residual is indicated by a blue circle. The red line is a smooth.

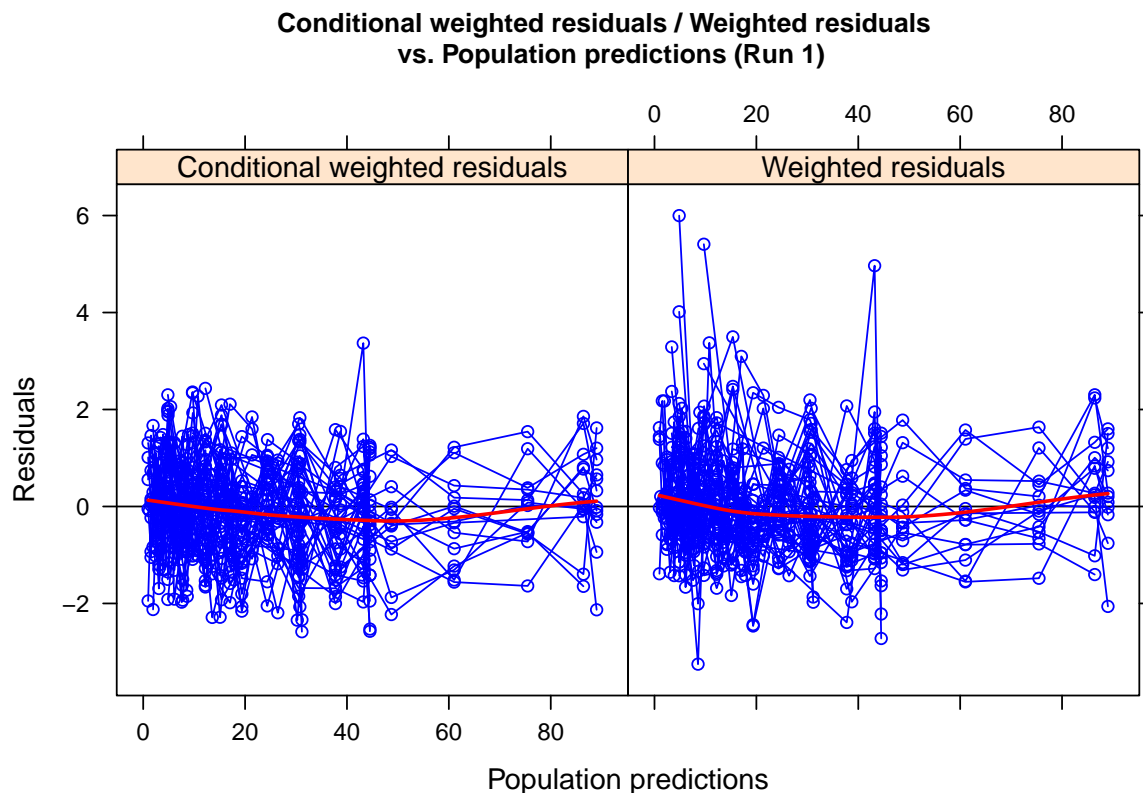
This graph is intended to check if there are any trends in the residuals with the predictions. The aim is to have no trends, i.e. that the smooth is horizontal.

It is more correct to use the CWRES (see Section “cwres.vs.pred” on the preceding page).

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")  
> xplot <- wres.vs.pred(xpdb)  
> print(xplot)
```

## cwres.wres.vs.pred



**Figure 28:** The conditional weighted residuals and the weighted residuals versus the population predictions. Each residual is indicated by a blue circle. The red line is a smooth.

This graph is intended to check for trends in the residuals over the predictions. Horizontal smooths centered on the horizontal zero lines are an indication that there are no unexplained trends.

Apart from the model development usage of this graph, it also provides an evaluation of the necessity of using CWRES instead of WRES. This can be useful if the computation of the

CWRES is cumbersome. With modern versions of NONMEM (7 and beyond), CWRES can be easily obtained and there is really no reason not to use CWRES.

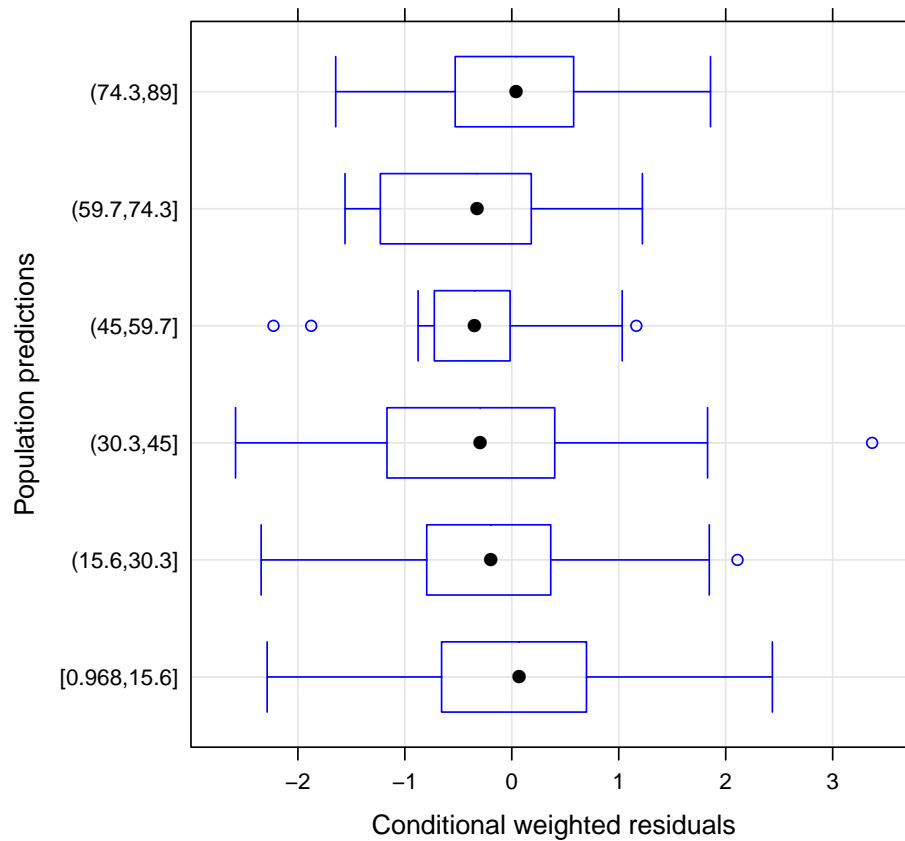
### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- cwres.wres.vs.pred(xpdb)
> print(xplot)
```



# cwres.vs.pred.bw

## Population predictions vs. Conditional weighted residuals (Run 1)



**Figure 29:** *Intervals of the population predictions versus the conditional weighted residuals. The distribution of the residuals for each time interval is visualized by a box and whisker plot.*

This graph is intended to check if there are any trends in the CWRES over the population predictions. It is similar to the xy-plot with CWRES versus PRED ("cwres.vs.pred" on page 38) but may be more useful when there is a lot of data points. The solid dots in the middle of the boxes indicate the medians of the residual distributions and if they fall close and randomly around the vertical zero line, it is an indication that the model is performing adequately. The

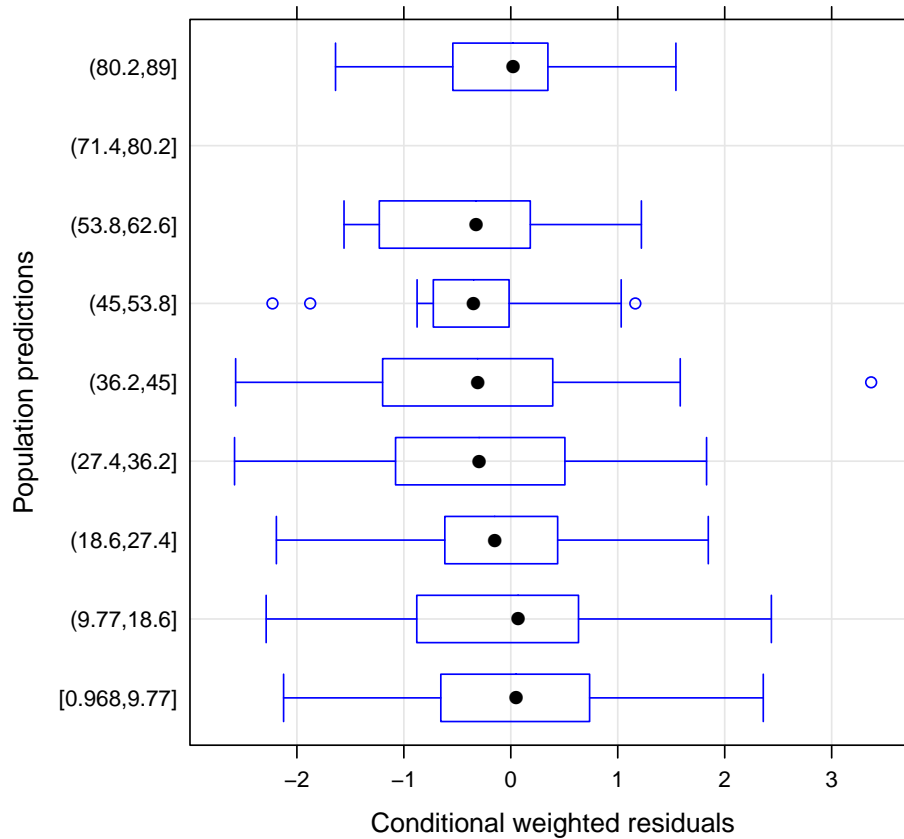
number of time intervals may have to be adjusted (with the `bins` argument) to something that is appropriate in relation to the number of observations.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- cwres.vs.pred.bw(xpdb,
+   bins = 6)
> print(xplot)
```

# wres.vs.pred.bw

## Population predictions vs. Conditional weighted residuals (Run 1)



**Figure 30:** *Intervals of the population predictions versus the weighted residuals. The distribution of the residuals for each time interval is visualized by a box and whisker plot.*

This graph is intended to check if there are any trends in the WRES over the population predictions. It is similar to the xy-plot with WRES versus PRED (“wres.vs.pred” on page 39) but may be more useful when there is a lot of data points. The solid dots in the middle of the boxes indicate the medians of the residual distributions and if they fall close and randomly

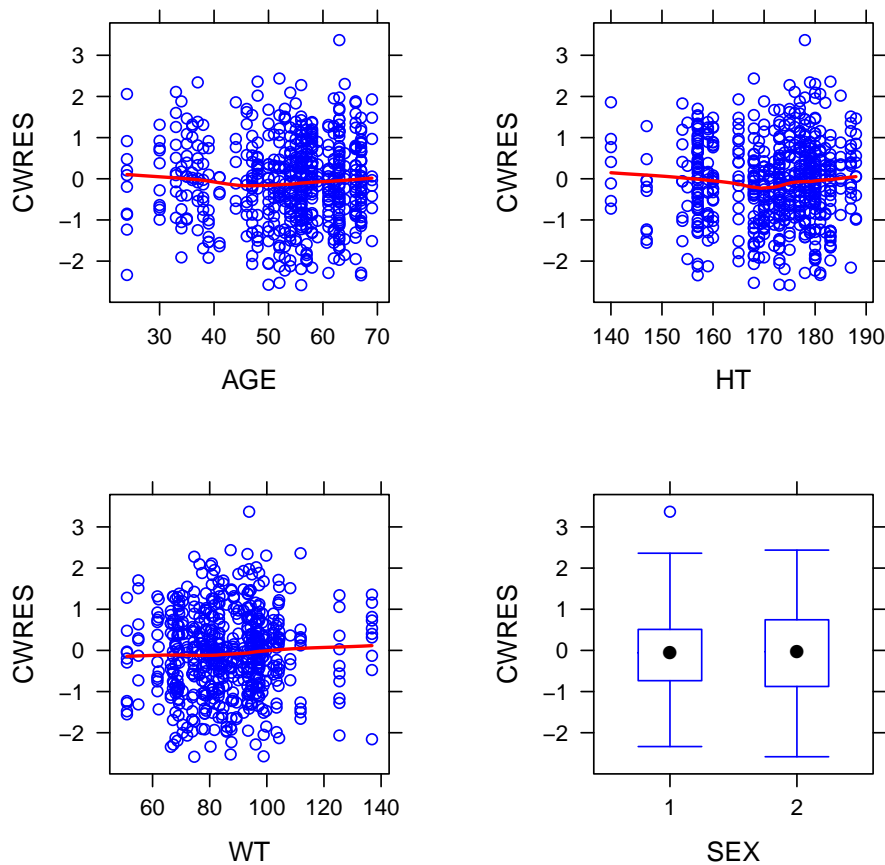
around the vertical zero line, it is an indication that the model is performing adequately.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- cwres.vs.idv.bw(xpdb,
+   bins = 6)
> print(xplot)
```

## cwres.vs.cov

### Conditional weighted residuals vs Covariates (Run 1)



**Figure 31:** *The conditional weighted residuals versus covariates. In the panels with continuous covariates, each residual is indicated by a blue circle. The red lines are smooths. In panels with categorical covariates, the distribution of the residuals displayed using box and whiskers plots.*

This graph is intended to check if the CWRES are independent of the covariates. Horizontal smooths (categorical covariates) or medians (categorical covariates) centered on the horizontal zero line suggest that this is the case.

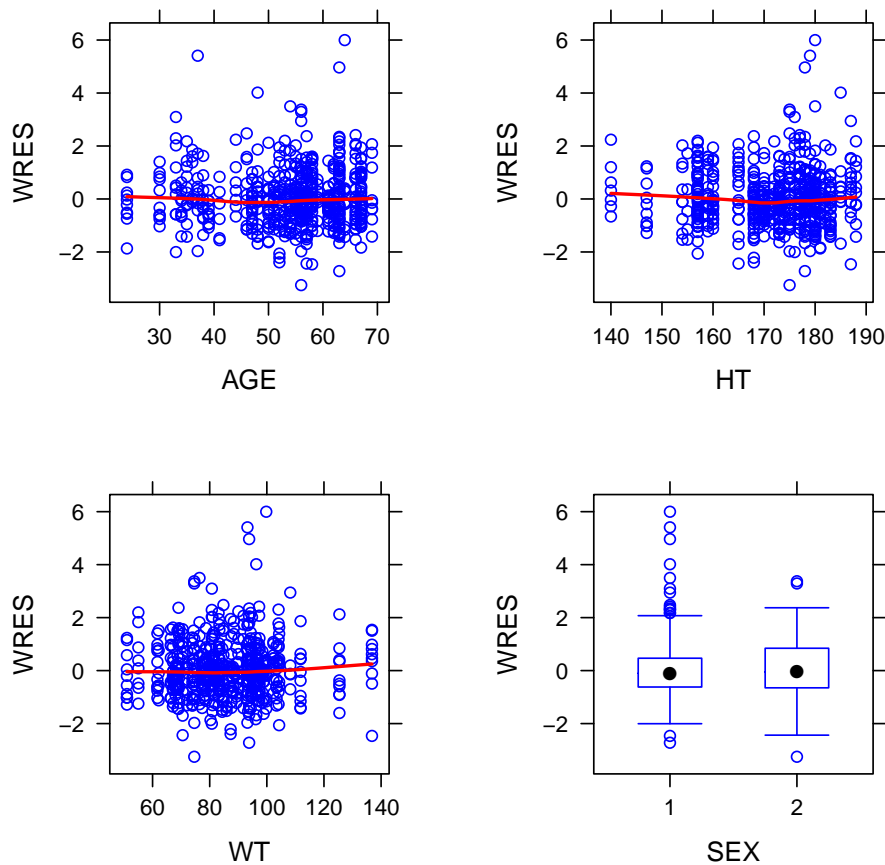
If the shrinkage is low it may be more informative to use eta versus covariate plots instead.

#### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "covariates") <- c("AGE",
+     "HT", "WT", "SEX")
> xplot <- cwres.vs.cov(xpdb)
> print(xplot)
```

## WRES.VS.COV

### Weighted residuals vs Covariates (Run 1)



**Figure 32:** The weighted residuals versus covariates. In the panels with continuous covariates, each residual is indicated by a blue circle. The red lines are smooths. In panels with categorical covariates, the distribution of the residuals displayed using box and whiskers plots.

This graph is intended to check if the WRES are independent of the covariates. Horizontal smooths (continuous covariates) or medians (categorical covariates) centered on the horizontal zero line suggest that this is the case.

If the shrinkage is low it may be more informative to use eta versus covariate plots instead. It is more correct to use CWRES instead of WRES (Section “cwres.vs.cov” on the previous

page).

#### Code used to generate the graph:

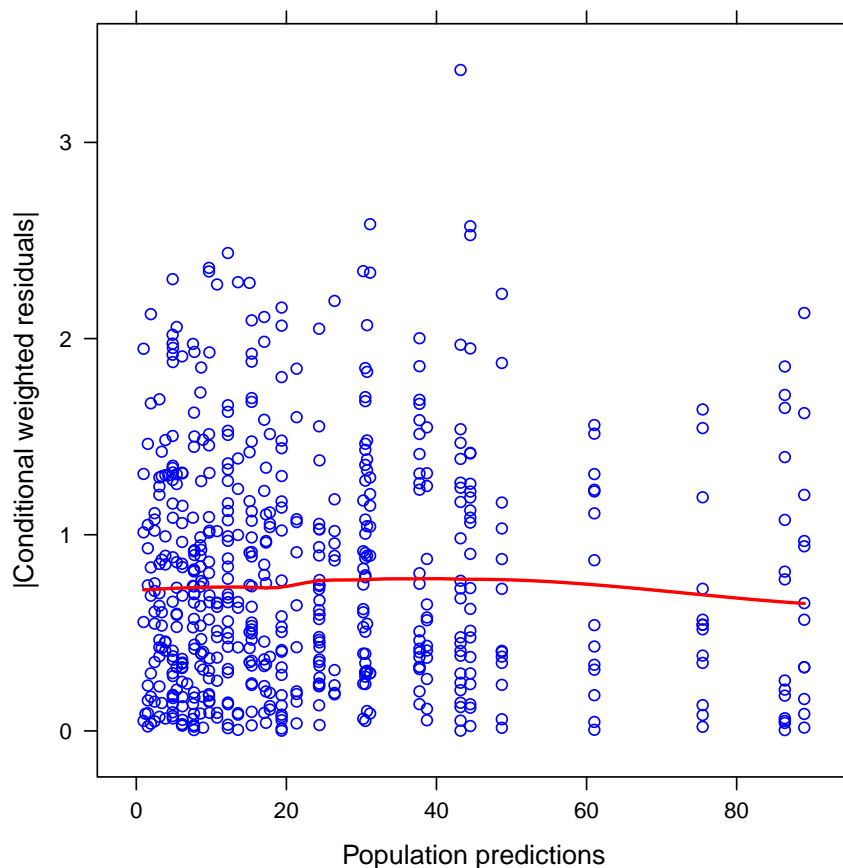
```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "covariates") <- c("AGE",
+     "HT", "WT", "SEX")
> xplot <- wres.vs.cov(xpdb)
> print(xplot)
```

# Residual model diagnostics

It is necessary to correctly characterise the statistical model (residual error, between subject variability, etc.) before any firm conclusions can be drawn from the model. Sometimes it may also be hard to judge the appropriateness of the structural model before the statistical model is in reasonably good shape (and vice versa). Additionally, to use statistical criteria (like p-values) to discriminate between contending models, it is necessary to have a good statistical model. One primary way of diagnosing the statistical model is by using residuals, the difference between observations and model predictions (both population and individual predictions). This section utilizes different types of residuals to investigate, primarily, the statistical portion of a population model.

# absval.cwres.vs.pred

Conditional weighted residuals| vs. Population predictions (Run 1



**Figure 33:** *The absolute values of the conditional weighted residuals versus the population predictions. Each residual is indicated by a blue circle. The red line is a smooth.*

This graph is intended to check if the CWRES are independent of the population predictions. Horizontal smooths suggest that this is the case. This graph is useful for diagnosing both the total variability explained in the model (residual unexplained variability, between sub-

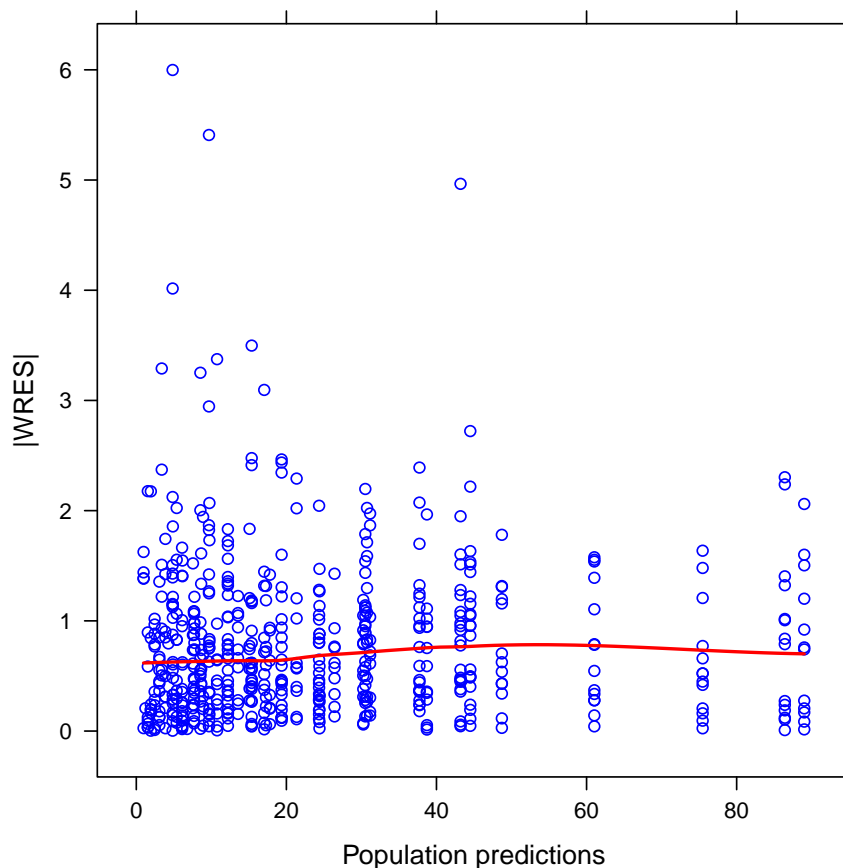
ject variability, etc.).

## Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")  
> xplot <- absval.cwres.vs.pred(xpdb)  
> print(xplot)
```

## absval.wres.vs.pred

[Weighted residuals] vs. Population predictions (Run 1)



**Figure 34:** The absolute values of the weighted residuals versus the population predictions. Each residual is indicated by a blue circle. The red line is a smooth.

This graph is intended to check if the WRES are independent of the population predictions. Horizontal smooths suggest that this is the case. This graph is useful for diagnosing both the total variability explained in the model (residual unexplained variability, between subject variability, etc.).

In general it is better to use the CWRES in-

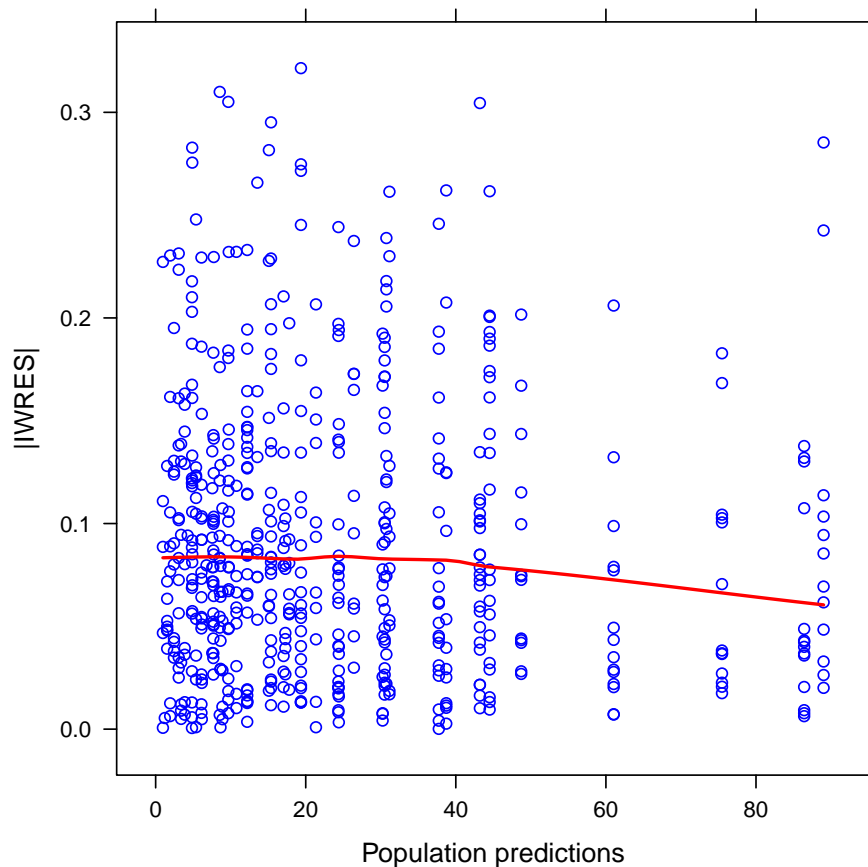
stead (see “absval.cwres.vs.pred” on the preceding page)

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")  
> xplot <- absval.wres.vs.pred(xpdb)  
> print(xplot)
```

# absval.iwres.vs.pred

**[Individual weighted residuals] vs. Population predictions (Run 1)**



**Figure 35:** *The absolute values of the individual weighted residuals versus the population predictions. Each residual is indicated by a blue circle. The red line is a smooth.*

This graph is intended to check if the IWRES are independent of the population predictions. Horizontal smooths suggest that this is the case. This graph is useful for diagnosing the residual unexplained variability (RUV) model.

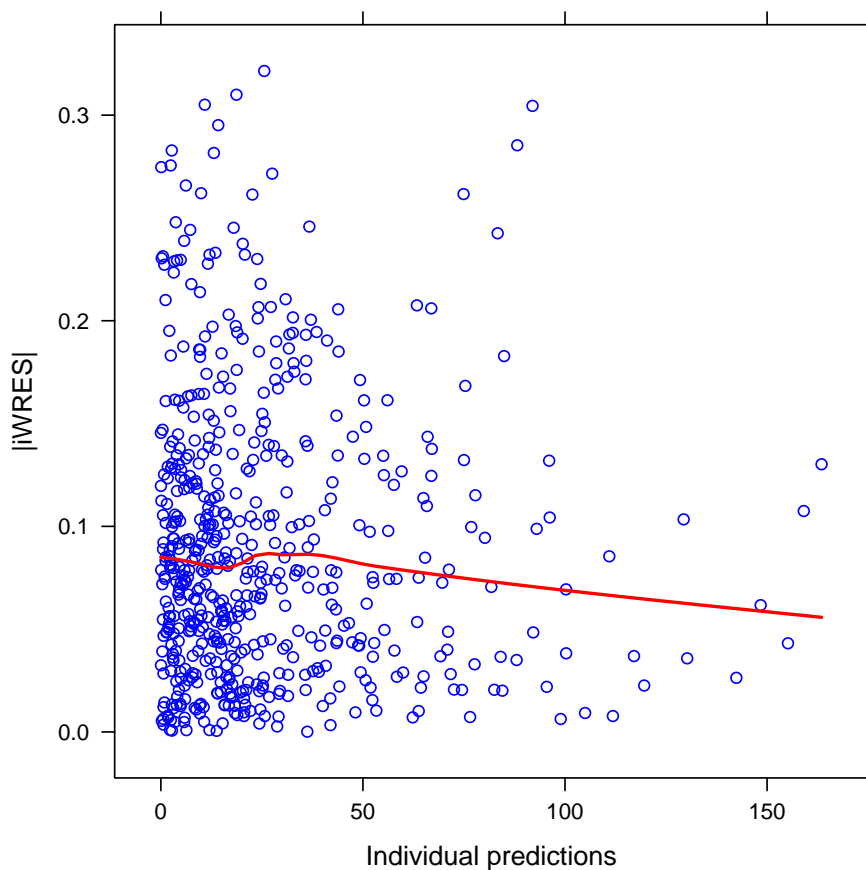
**Code used to generate the graph:**

```
> xpdb <- xpose.data(1, directory = "../")  
> xplot <- absval.iwres.vs.pred(xpdb)  
> print(xplot)
```



## absval.iwres.vs.ipred

**|Individual weighted residuals| vs. Individual predictions (Run 1)**



**Figure 36:** *The absolute values of the individual weighted residuals versus the individual predictions. Each residual is indicated by a blue circle. The red line is a smooth.*

This graph is intended to check if the IWRES are independent of the individual predictions. Horizontal smooths suggest that this is the case. This graph is useful for diagnosing the residual unexplained variability (RUV) model.

If the  $\epsilon$ -shrinkage is high, this graph may not

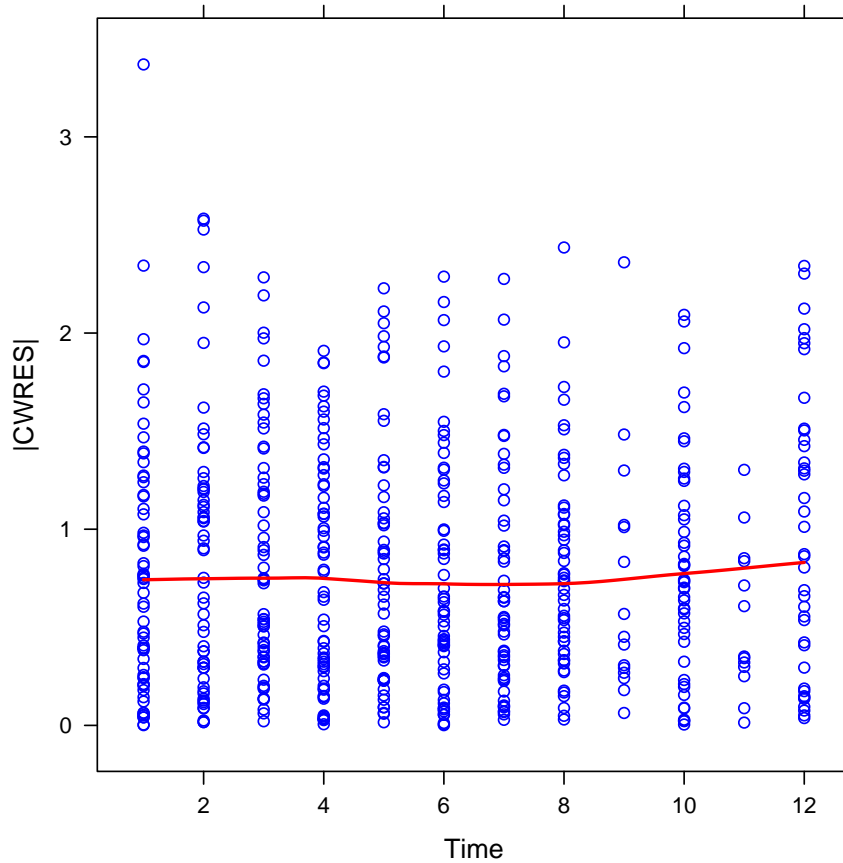
be very informative.

**Code used to generate the graph:**

```
> xpdb <- xpose.data(1, directory = "../")  
> xplot <- absval.iwres.vs.ipred(xpdb)  
> print(xplot)
```

## absval.wres.vs.idv

|Conditional weighted residuals| vs. Time (Run 1)



**Figure 37:** The absolute values of the conditional weighted residuals versus the independent variable. Each residual is indicated by a blue circle. The red line is a smooth.

This graph is intended to check that the variability of the CWRES are independent of the independent variable. Horizontal smooths suggest that this is the case. This graph is useful for diagnosing the total variability explained in the model (residual unexplained variability,

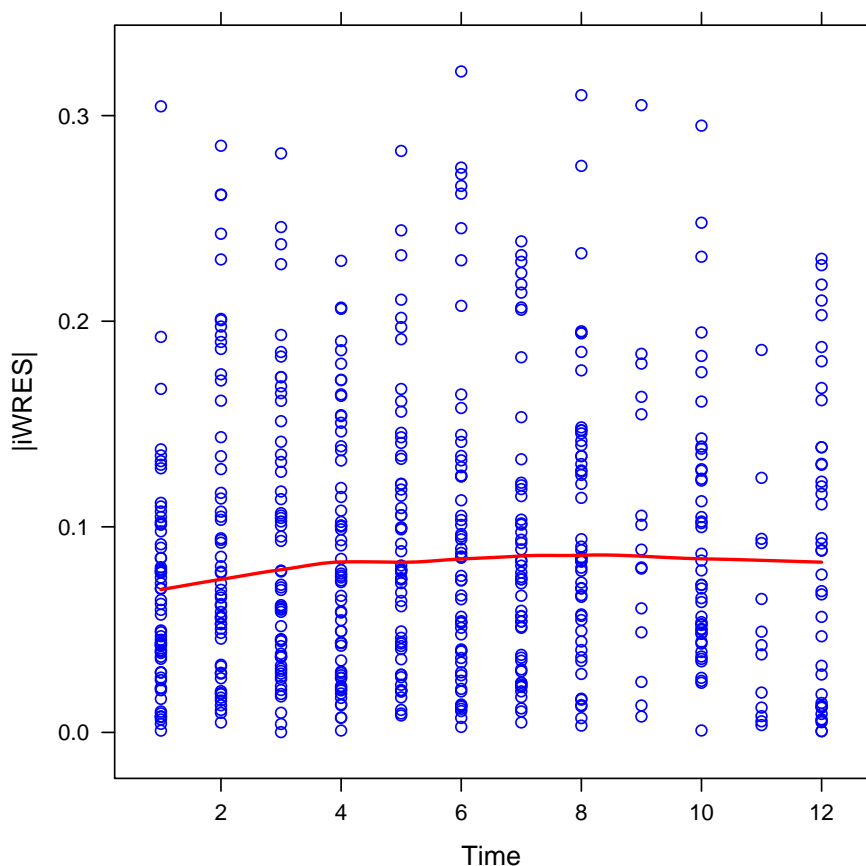
between subject variability, etc.).

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- absval.wres.vs.idv(xpdb)
> print(xplot)
```

# absval.iwres.vs.idv

|Individual weighted residuals| vs. Time (Run 1)



**Figure 38:** The absolute values of the individual weighted residuals versus the independent variable. Each residual is indicated by a blue circle. The red line is a smooth.

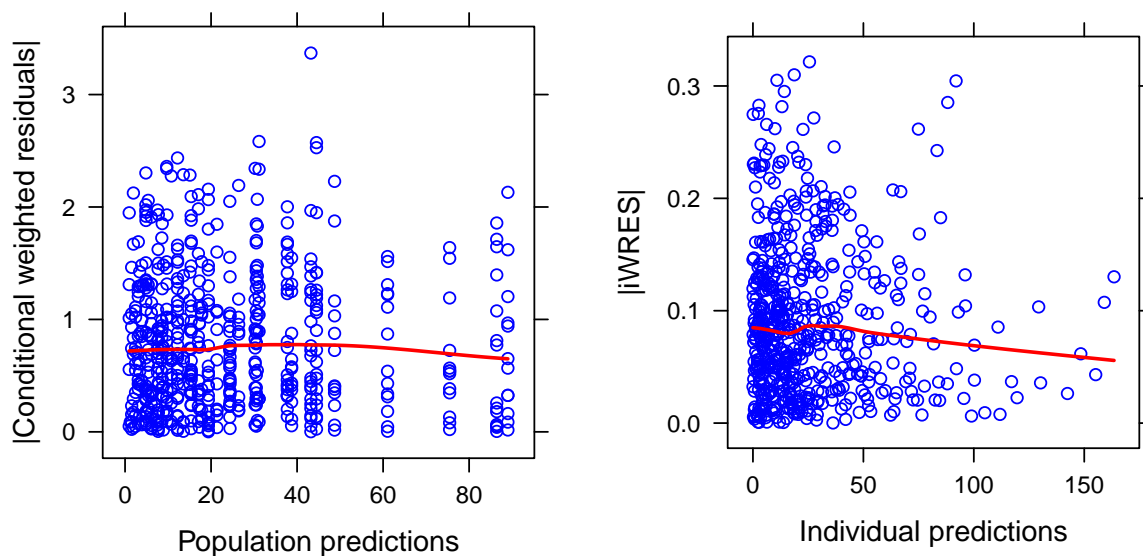
This graph is intended to check if the variability of the IWRES are independent of the independent variable. Horizontal smooths suggest that this is the case.

**Code used to generate the graph:**

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- absval.iwres.vs.idv(xpdb)
> print(xplot)
```

# absval.iwres.cwres.vs.ipred.pred

(Conditional) Weighted residuals vs. Predictions (Run 1)



**Figure 39:** The absolute values of the population and individual weighted residuals versus the population and individual predictions. Each residual is indicated by a blue circle. The red line is a smooth.

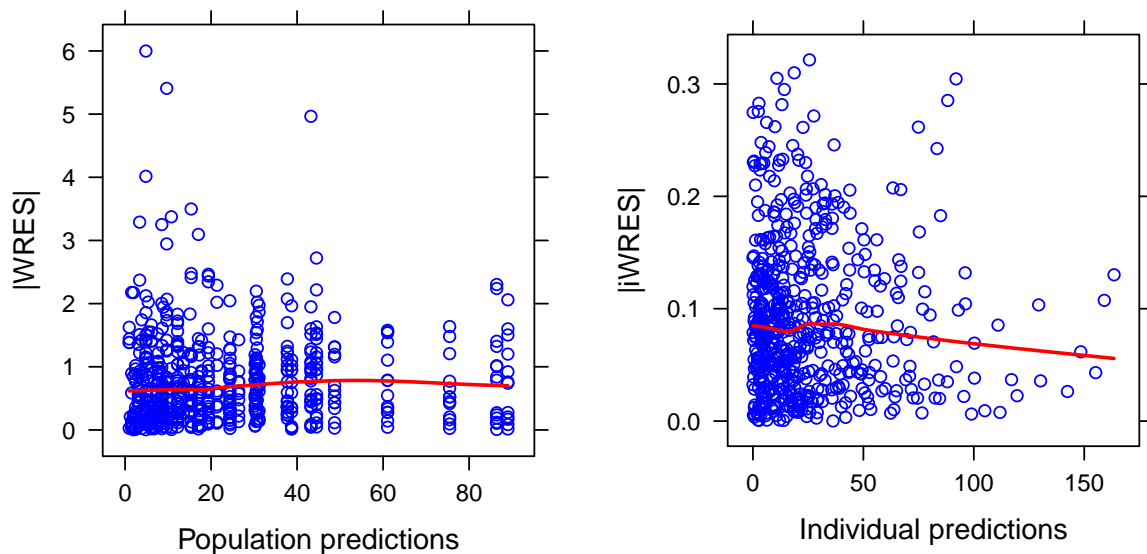
This graph is a composite of “abs-  
sval.cwres.vs.pred” on page 46 and “ab-  
sval.iwres.vs.ipred” on page 49.

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- absval.iwres.cwres.vs.ipred.pred(xpdb)
> print(xplot)
```

**Code used to generate the graph:**

# absval.iwres.wres.vs.ipred.pred

## Weighted residuals vs. Predictions (Run 1)



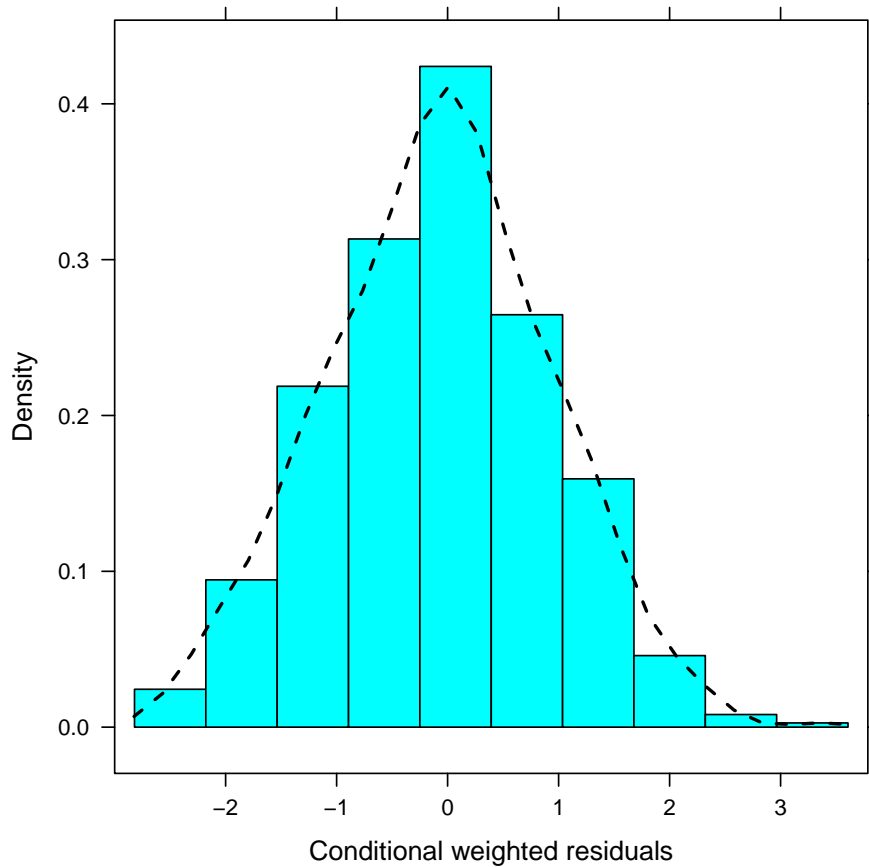
**Figure 40:** The absolute values of the population and individual weighted residuals versus the population and individual predictions. Each residual is indicated by a blue circle. The red line is a smooth.

This graph is a composite of “ab- > xpdb <- xpose.data(1, directory = "../")  
sval.wres.vs.pred” on page 47 and “ab- > xplot <- absval.iwres.wres.vs.ipred.pred(xpdb)  
sval.iwres.vs.ipred” on page 49. > print(xplot)

**Code used to generate the graph:**

## cwres.dist.hist

Distribution of Conditional weighted residuals (Run 1)



**Figure 41:** *The distribution of the weighted residuals. The height of the bars represent the density of the distribution and the line is a smoothed representation of the density.*

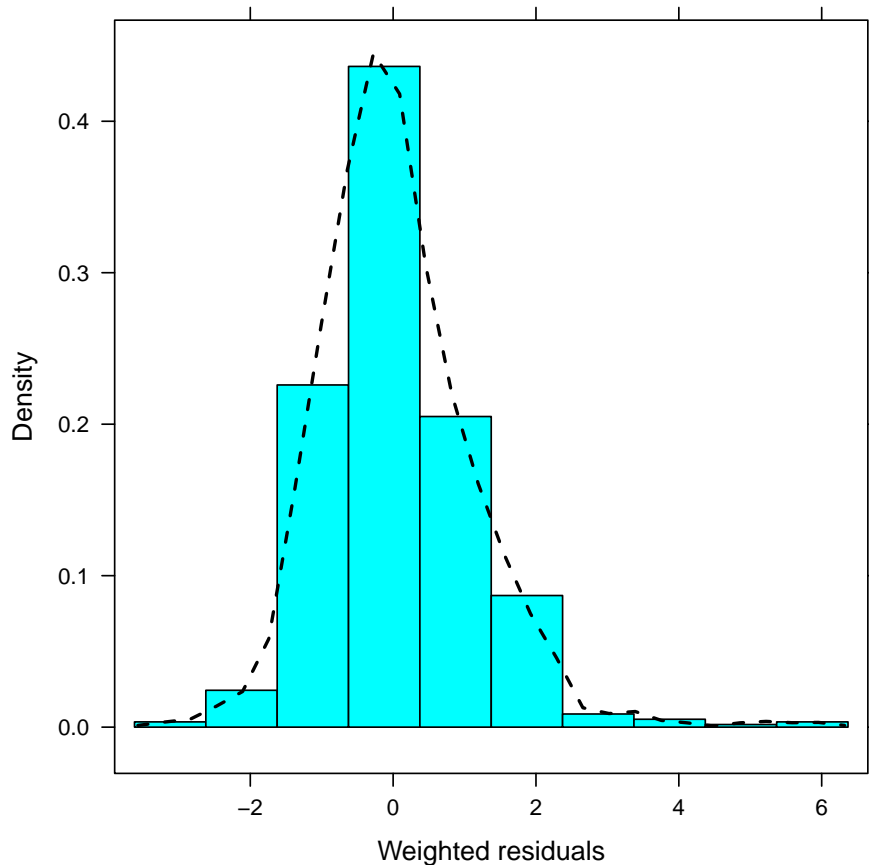
The purpose is to visualize the distribution of the CWRES. Nominally the CWRES should be normally distributed ( $CWRES_i = N(0,1)$ ), hence 95% of the CWRES points should be between -2 and 2. The center of the distribution should be 0.

**Code used to generate the graph:**

```
> xpdb <- xpose.data(1, directory = "../")  
> xplot <- cwres.dist.hist(xpdb)  
> print(xplot)
```

# wres.dist.hist

Distribution of Weighted residuals (Run 1)



**Figure 42:** The distribution of the weighted residuals. The height of the bars represent the density of the distribution and the line is a smoothed representation of the density.

The purpose is to visualize the distribution of the WRES. Nominally the WRES should be normally distributed ( $WRES_i = N(0,1)$ ), hence 95% of the CWRES points should be between -2 and 2. The center of the distribution should be 0. Typically the CWRES should be used instead of the WRES as the WRES

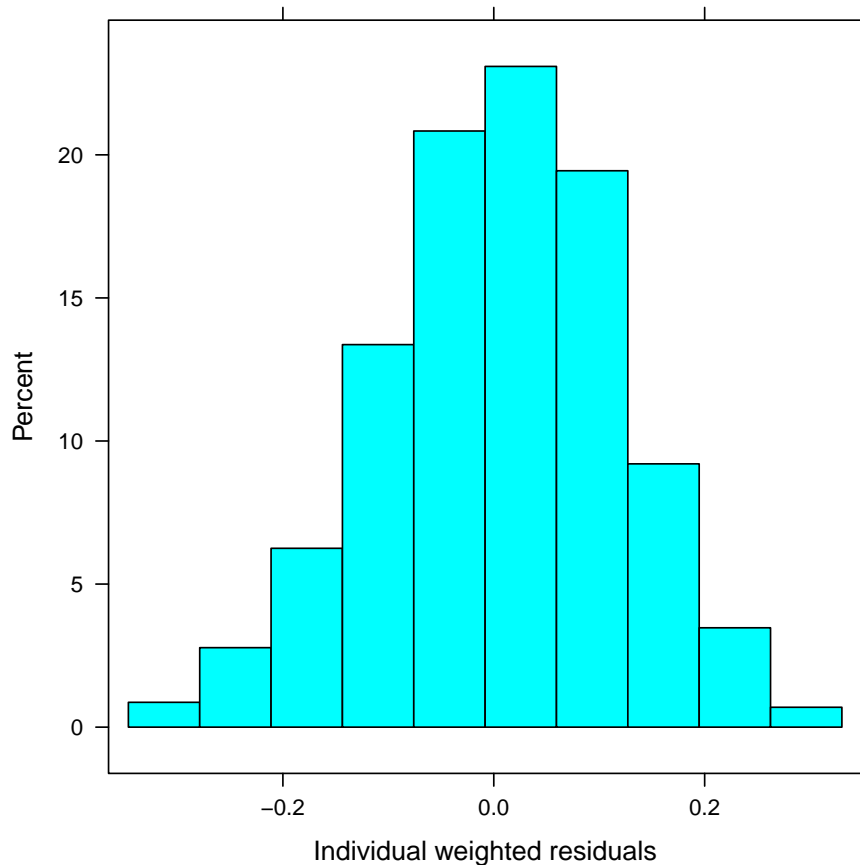
may not be normally distributed even if the model is describing the data well.

## Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- wres.dist.hist(xpdb)
> print(xplot)
```

# iwres.dist.hist

## Distribution of Individual weighted residuals (Run 1)



**Figure 43:** *The distribution of the individual weighted residuals. The height of the bars represent the percentage of points in the interval.*

The purpose is to visualize the distribution of the IWRES. The IWRES should be normally distributed. In this particular graph, the y-axis is set to “percent” and the density line is made invisible by using a transparent color.

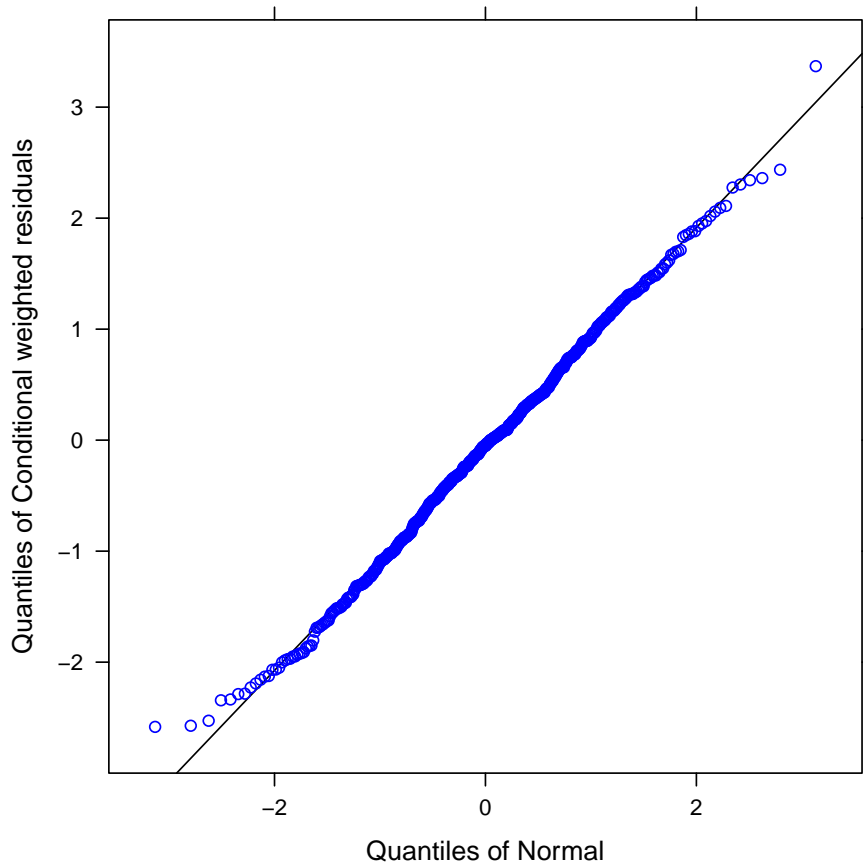
```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- iwres.dist.hist(xpdb,
+   type = "percent", hidcol = "transparent",
+   ylb = "Percent")
> print(xplot)
```

**Code used to generate the graph:**



## cwres.dist.qq

Distribution of Conditional weighted residuals (Run 1)



**Figure 44:** The quantiles of the conditional weighted residuals versus the quantiles of the standard normal distribution. Each data point is represented by a blue circle. The black solid line is the line of identity.

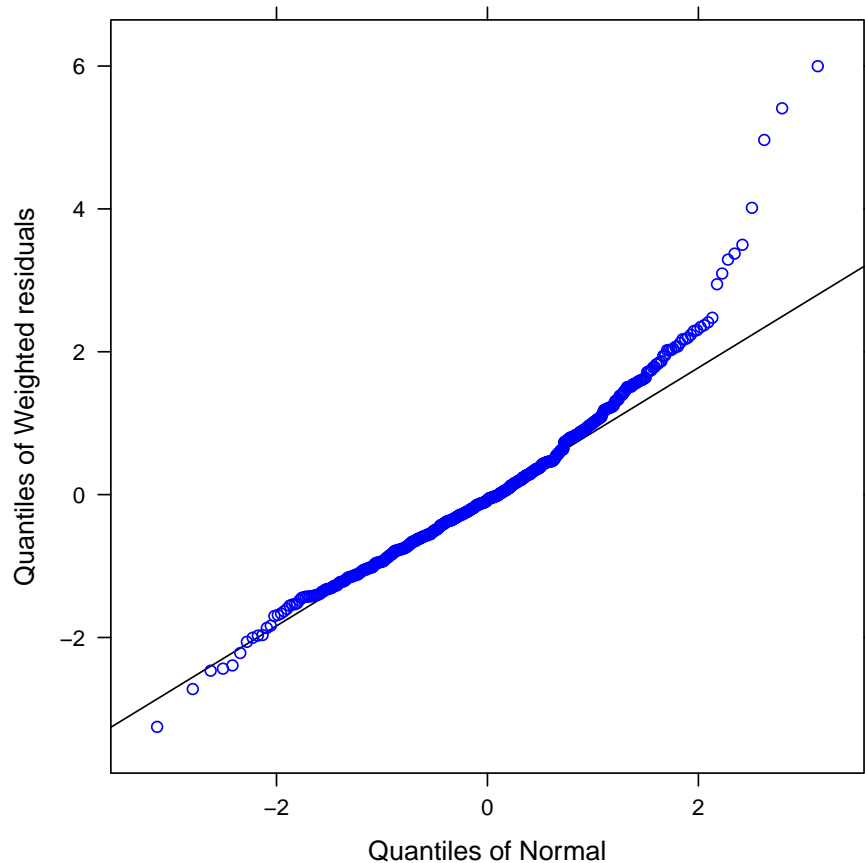
The purpose is to directly compare the shape of the distribution of the CWRES to a normal distribution. If all points fall on the line of identity, the shape of the CWRES distribution is exactly normal.

**Code used to generate the graph:**

```
> xpdb <- xpose.data(1, directory = "../")  
> xplot <- cwres.dist.qq(xpdb)  
> print(xplot)
```

## wres.dist.qq

Distribution of Weighted residuals (Run 1)



**Figure 45:** *The quantiles of the weighted residuals versus the quantiles of a normal distribution. Each data point is represented by a blue circle. The black solid line is the line of identity.*

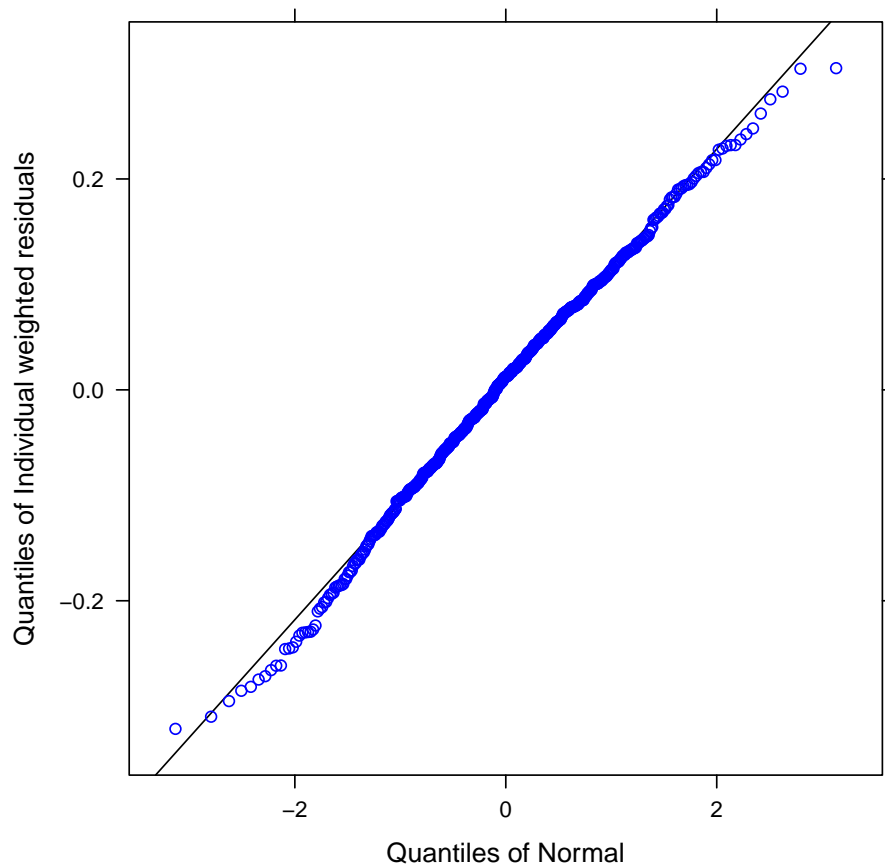
The purpose is to directly compare the shape of the distribution of the WRES to a normal distribution. If all points fall on the line of identity, the shape of the WRES distribution is exactly normal.

**Code used to generate the graph:**

```
> xpdb <- xpose.data(1, directory = "../")  
> xplot <- wres.dist.qq(xpdb)  
> print(xplot)
```

## iwres.dist.qq

Distribution of Individual weighted residuals (Run 1)



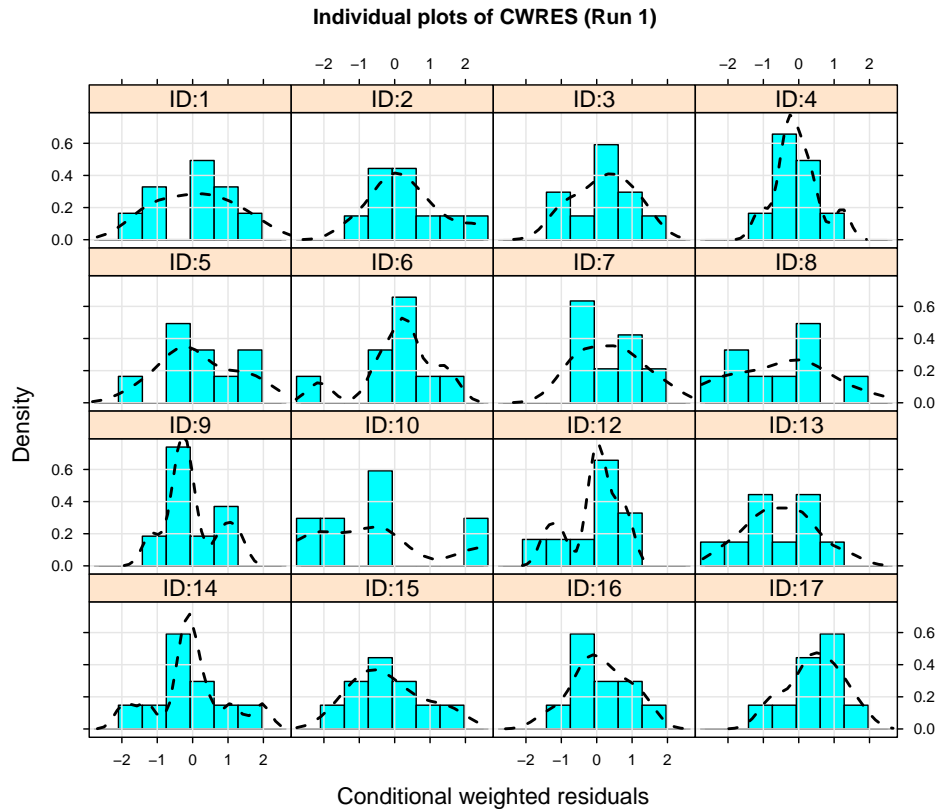
**Figure 46:** The quantiles of the individual weighted residuals versus the quantiles of a standard normal distribution. Each data point is represented by a blue circle. The black solid line is the line of identity.

The purpose is to directly compare the shape of the distribution of the IWRES to a normal distribution. If all points fall on the line of identity, the shape of the IWRES distribution is exactly normal.

**Code used to generate the graph:**

```
> xpdb <- xpose.data(1, directory = "../")  
> xplot <- iwres.dist.qq(xpdb)  
> print(xplot)
```

# ind.plots.cwres.hist



**Figure 47:** The individual distributions of the CWRES. The height of the bars represent the density of the distribution and line is a smoothed representation of the density.

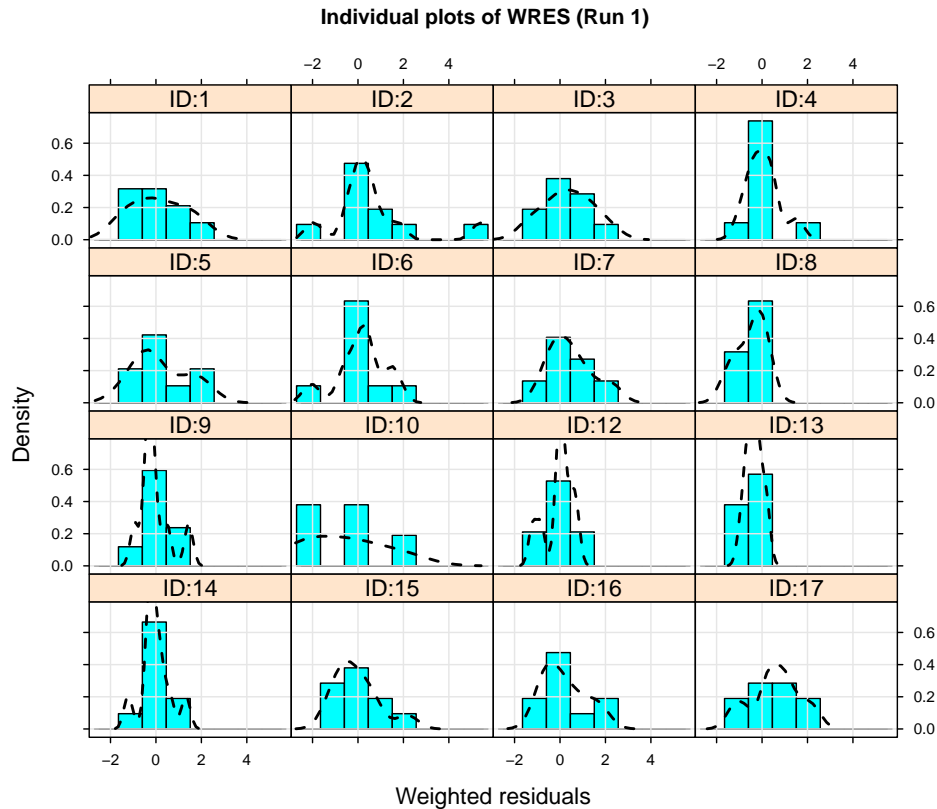
This graph can be used to visualize the individual distributions of the CWRES. The underlying assumption is that the CWRES should be normally distributed. If all *individual* CWRES distributions are symmetrically/normally distributed then this will be the case. On the other hand, the overall distribution can still be normally distributed even if not all individual distributions are. It may then be useful to iden-

tify the individuals that appears to have non-symmetric/non-normal distributions, as these may be outliers or carry unique information.

## Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- ind.plots.cwres.hist(xpdb,
+   grid = TRUE, subset = "ID<=17")
> print(xplot)
```

# ind.plots.wres.hist



**Figure 48:** The individual distribution of the WRES. The height of the bars represent the density of the distribution and the line is a smoothed representation of the density.

This graph can be used to visualize the individual distributions of the WRES. The underlying assumption is that the WRES should be normally distributed. If all *individual* WRES distributions are symmetrically/normally distributed then this will be the case. On the other hand, the overall distribution can still be symmetrically/normally distributed even if not all individual distributions are. It may then be useful to identify the individuals that appears to have non-symmetric/non-normal distributions, as these may be outliers or carry unique infor-

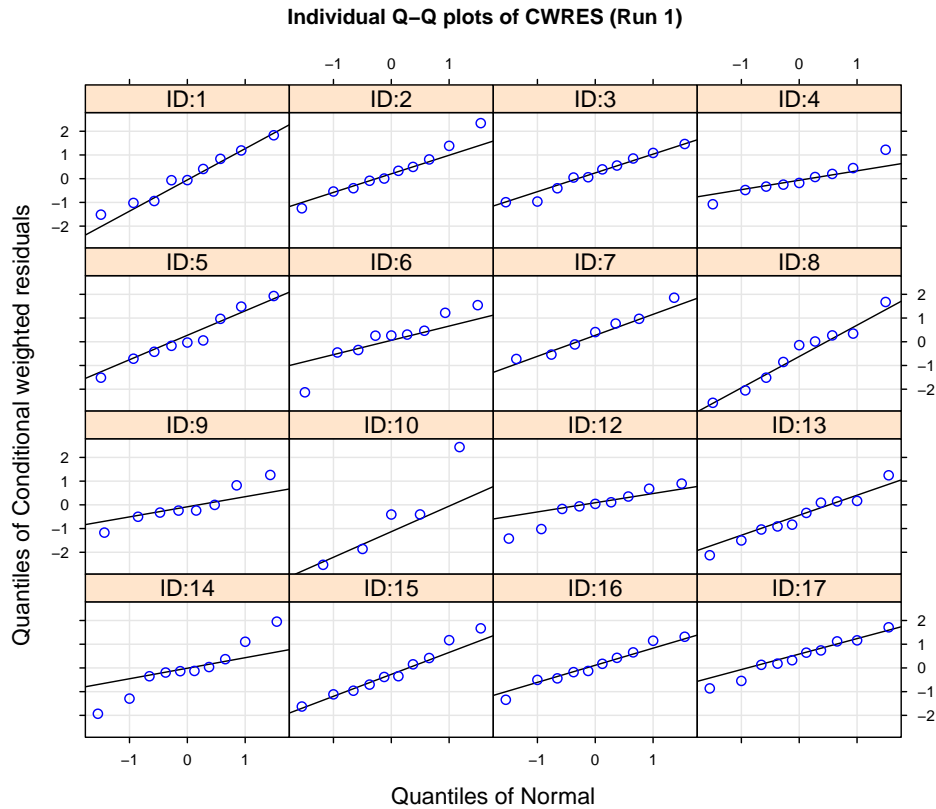
mation.

It is probably better to use the `ind.plots.cwres` function ( on the previous page).

## Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- ind.plots.wres.hist(xpdb,
+   grid = TRUE, subset = "ID<=17")
> print(xplot)
```

# ind.plots.cwres.qq



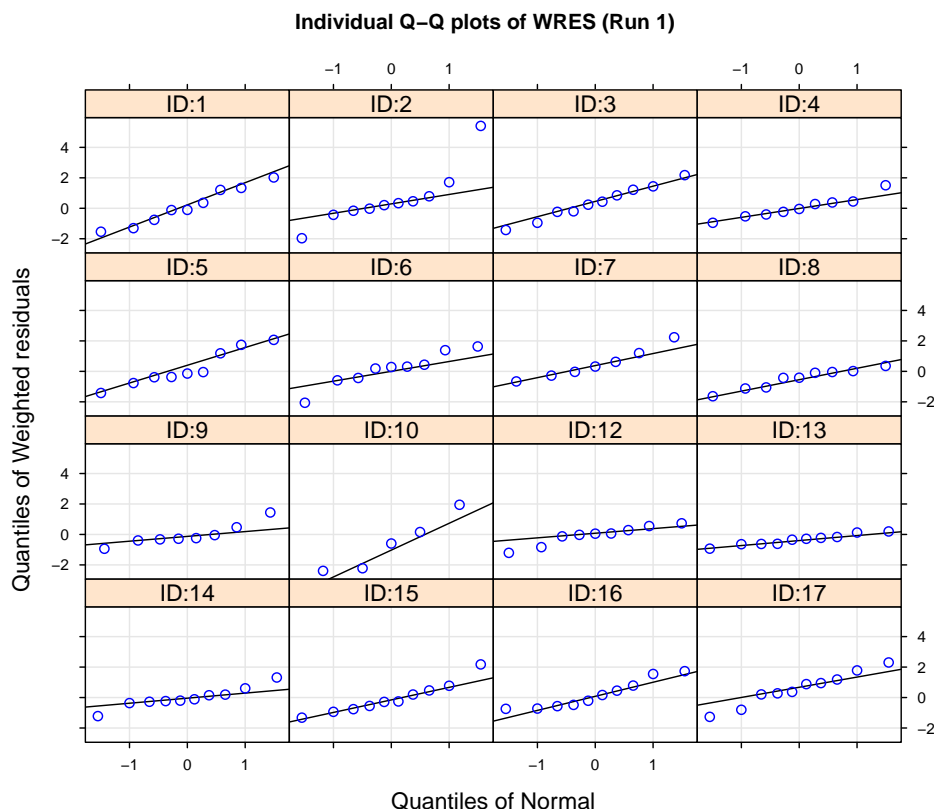
**Figure 49:** QQ plots of the individual CWRES distributions. The quantiles of the CWRES for each individual are plotted against the quantiles of the normal distribution. The diagonal black line is the line of identity.

This graph can be used to compare the CWRES distribution of each individual against the normal distribution. If the individual distribution is normally distributed the points should fall on the line of identity.

## Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- ind.plots.cwres.qq(xpdb,
+   grid = TRUE, subset = "ID<=17")
> print(xplot)
```

# ind.plots.wres.qq



**Figure 50:** QQ plots of the individual WRES distributions. The quantiles of the WRES for each individual are plotted against the quantiles of the normal distribution. The diagonal black line is the line of identity.

This graph can be used to compare the WRES distribution of each individual against the normal distribution. If the individual distribution is normally distributed the points should fall on the line of identity.

If the FOCE estimation method is used, it is probably better to use the `ind.plots.qq` func-

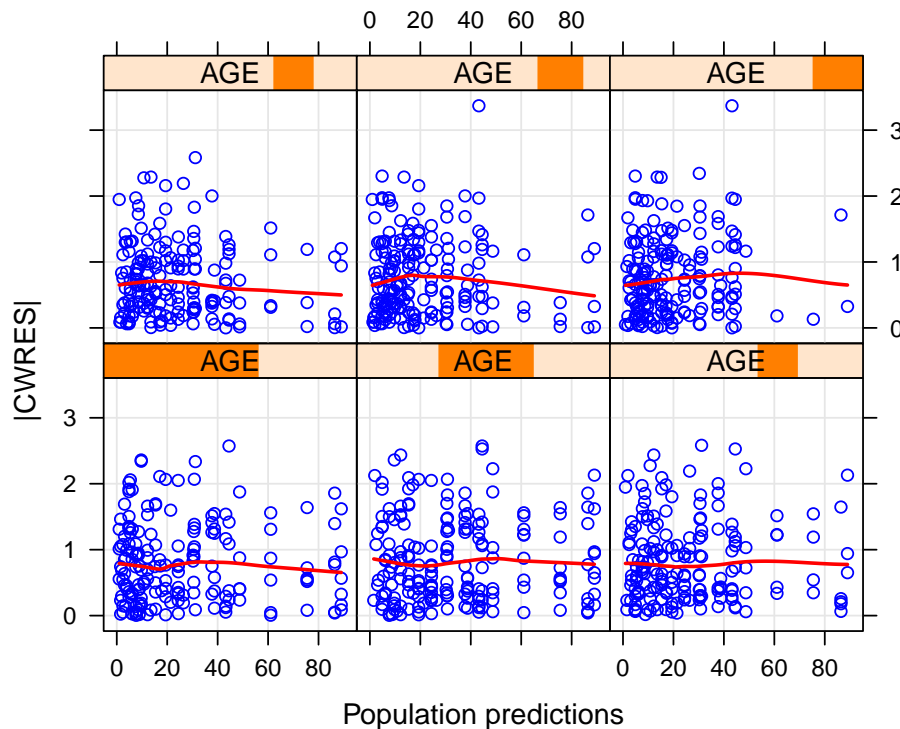
tion ( on the preceding page).

## Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> xplot <- ind.plots.wres.qq(xpdb,
+   grid = TRUE, subset = "ID<=17")
> print(xplot)
```

# absval.cwres.vs.pred.by.cov

[Conditional weighted residuals]  
vs Population predictions (Run 1)



**Figure 51:** The absolute values of the conditional weighted residuals versus the population predictions by age. Each residual is indicated by a blue circle. The red line is a smooth.

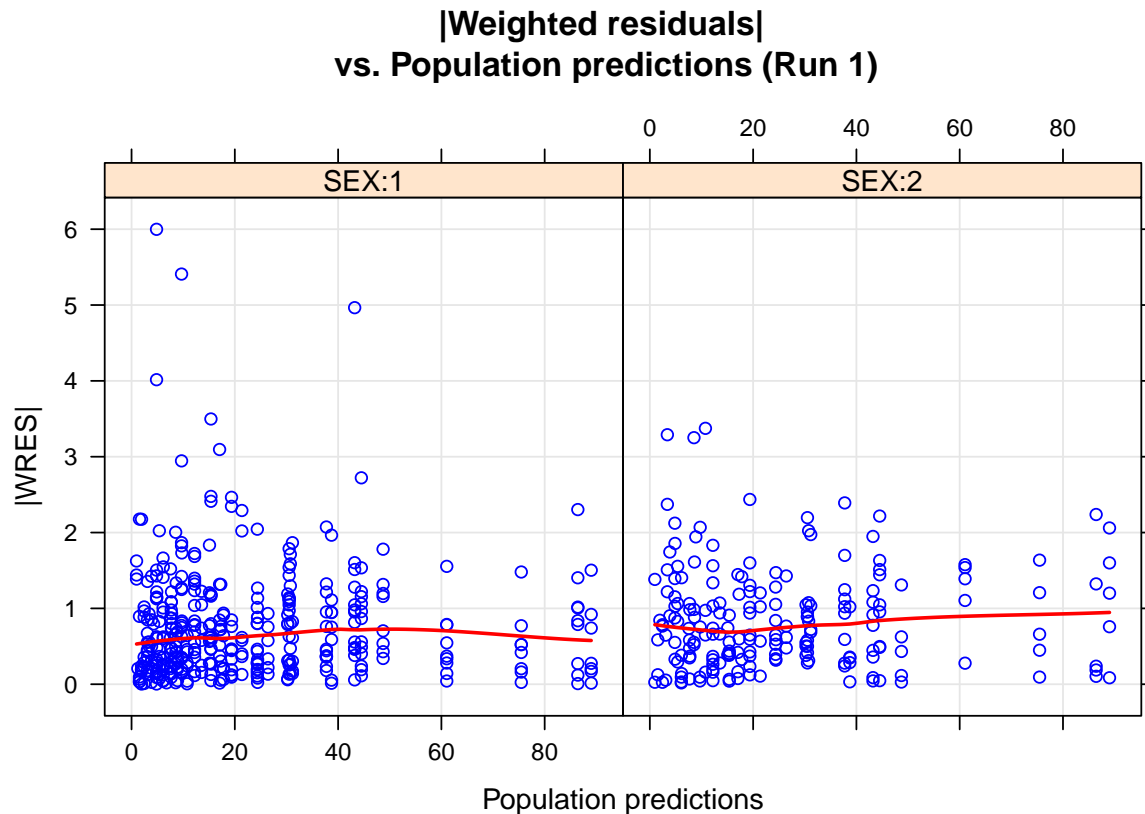
This graph is intended to check if the CWRES are independent of the population predictions in each interval of the covariate. Horizontal smooths in each panel would suggest that this is the case.

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "covariates") <- c("AGE")
> xplot <- absval.cwres.vs.pred.by.cov(xpdb,
+   grid = TRUE)
> print(xplot)
```

**Code used to generate the graph:**



## absval.wres.vs.pred.by.cov



**Figure 52:** *The absolute values of the weighted residuals versus the population predictions by sex. Each residual is indicated by a blue circle. The red line is a smooth.*

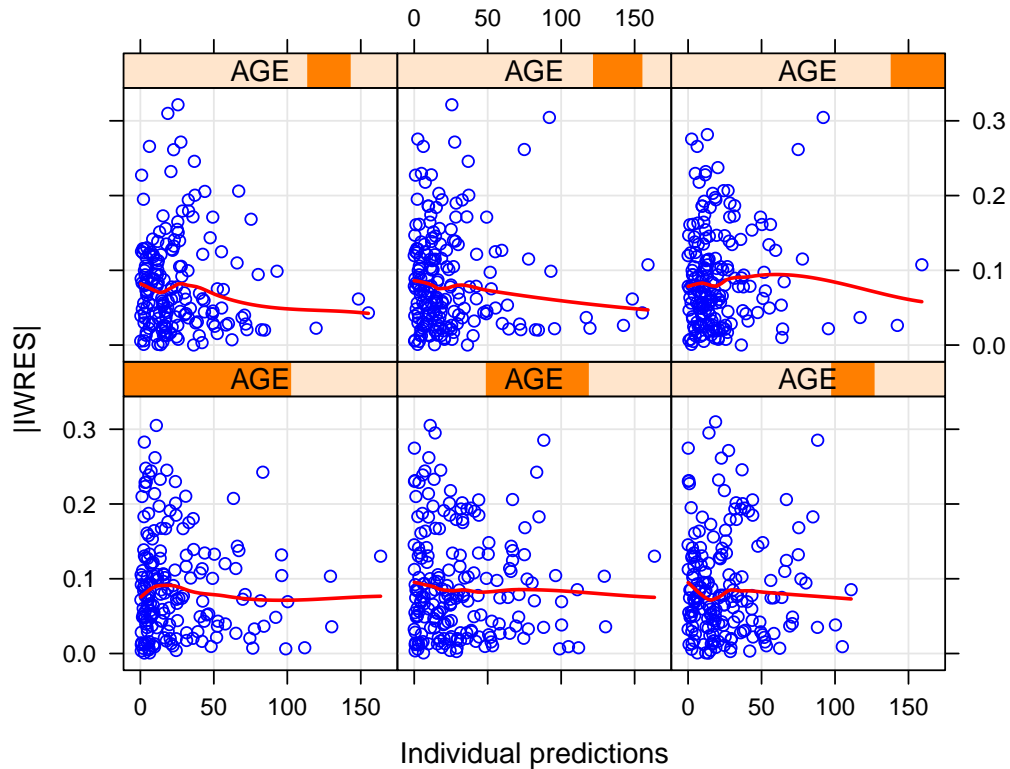
This graph is intended to check if the WRES are independent of the population predictions in each of the covariate categories (similarly to “absval.wres.vs.pred” on page 47). Horizontal smooths in each panel would suggest that this is the case.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "covariates") <- c("SEX")
> xplot <- absval.wres.vs.pred.by.cov(xpdb,
+   grid = TRUE)
> print(xplot)
```

# absval.iwres.vs.ipred.by.cov

|Individual weighted residuals| vs Individual predictions (Run 1)



**Figure 53:** The absolute values of the individually weighted residuals versus the population predictions by age. Each residual is indicated by a blue circle. The red line is a smooth.

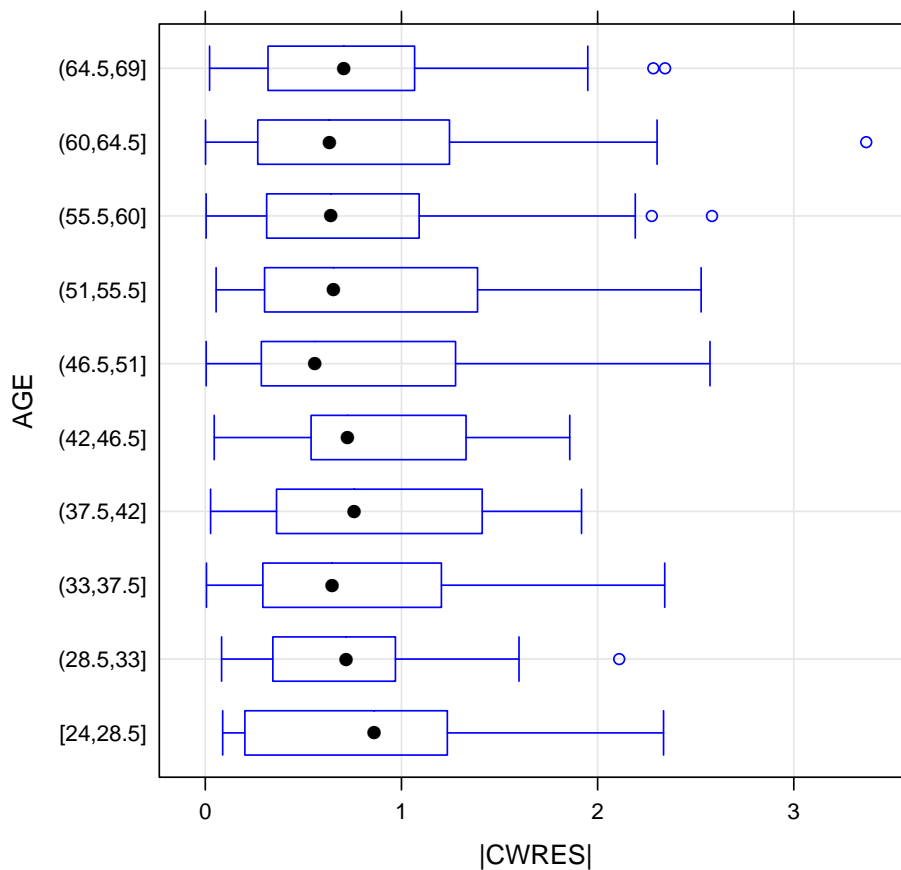
This graph is intended to check if the IWRES are independent of the population predictions in each of the interval of the covariate. Horizontal smooths in each panel would suggest that this is the case.

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "covariates") <- c("AGE")
> xplot <- absval.iwres.vs.ipred.by.cov(xpdb,
+   grid = TRUE)
> print(xplot)
```

**Code used to generate the graph:**

# absval.cwres.vs.cov.bw

[Conditional weighted residuals] vs Covariates (Run 1)



**Figure 54:** Box and whiskers plot of the absolute values of the conditional weighted residuals for each interval of age.

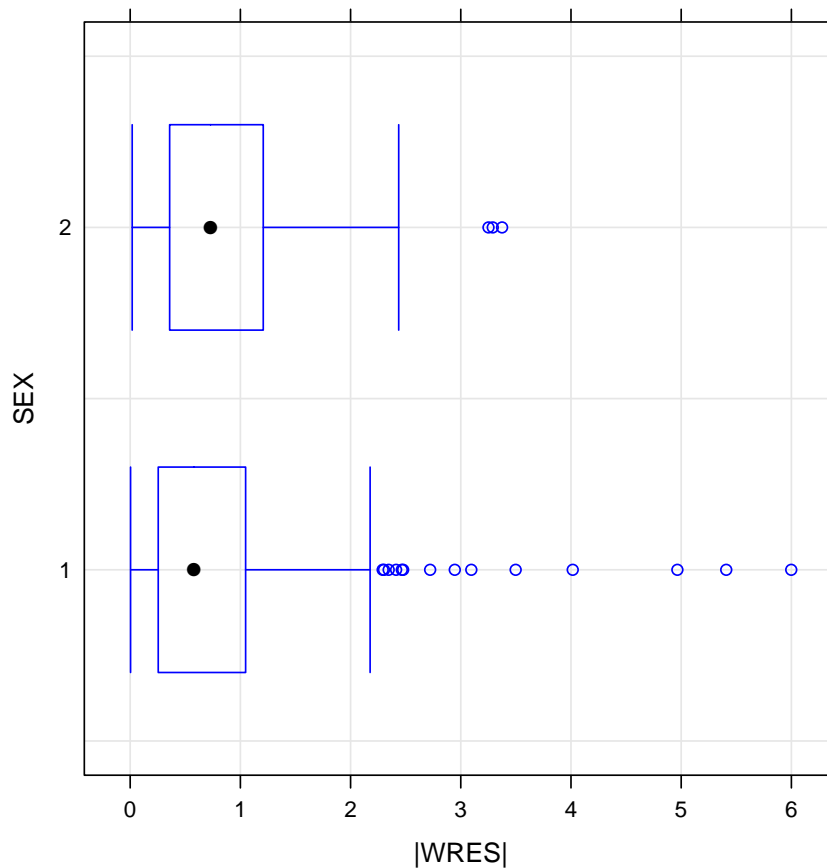
This graph is intended to check if the CWRES are similarly distributed between the covariate categories. Black dots (the medians) located at the same x-value would suggest that this is the case.

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "covariates") <- c("AGE")
> xplot <- absval.cwres.vs.pred.by.cov(xpdb,
+   grid = TRUE)
> print(xplot)
```

**Code used to generate the graph:**

# absval.wres.vs.cov.bw

**[Weighted residuals] vs Covariates (Run 1)**



**Figure 55:** *Box and whiskers plot of the absolute values of the weighted residuals for each level of sex.*

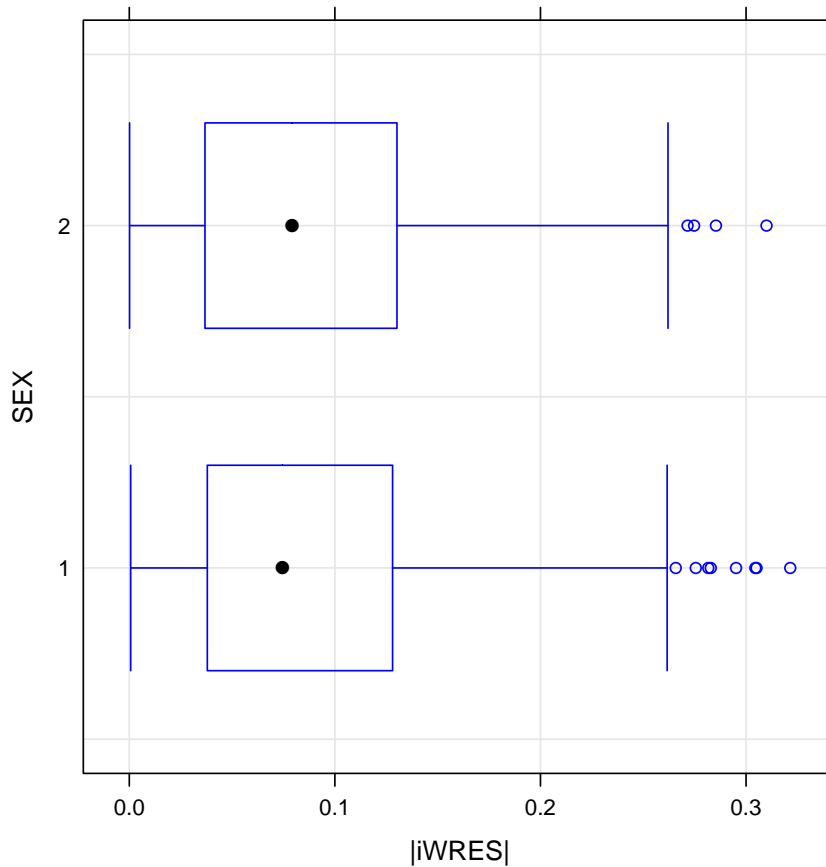
This graph is intended to check if the WRES are similarly distributed between the covariate categories. Black dots (the medians) located at the same x-value would suggest that this is the case.

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "covariates") <- c("SEX")
> xplot <- absval.wres.vs.cov.bw(xpdb,
+   grid = TRUE)
> print(xplot)
```

**Code used to generate the graph:**

# absval.iwres.vs.cov.bw

|Individual weighted residuals| vs Covariates (Run 1)



**Figure 56:** Box and whiskers plot of the absolute values of the weighted residuals for each level of sex.

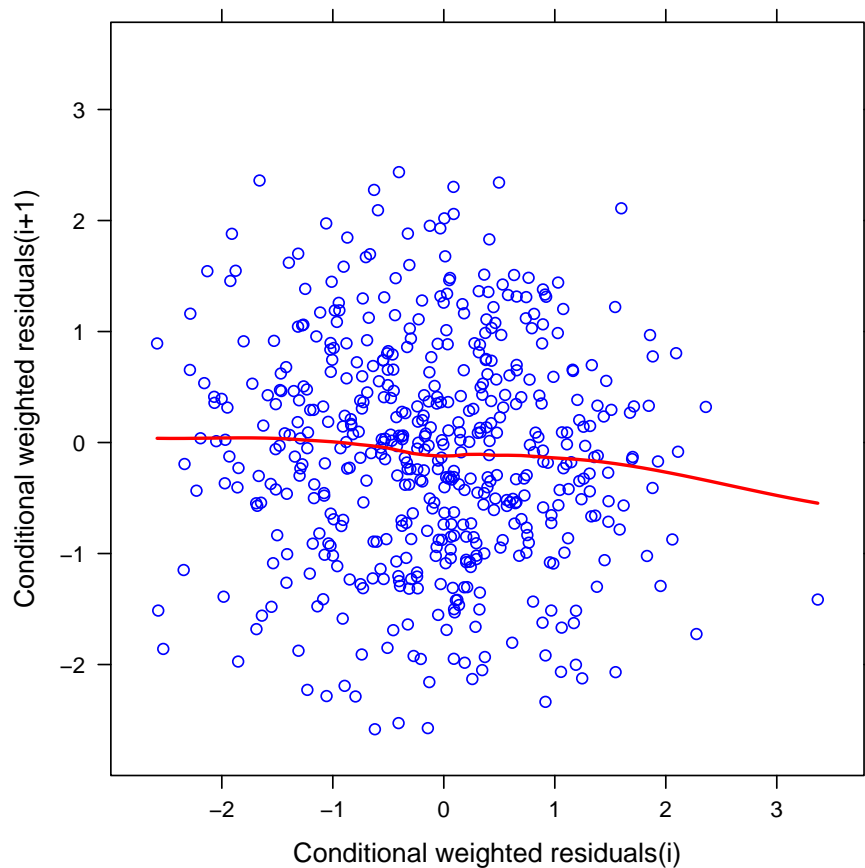
This graph is intended to check if the IWRES are similarly distributed between the covariate categories. Black dots (the medians) located at the same x-value would suggest that this is the case.

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "covariates") <- c("SEX")
> xplot <- absval.iwres.vs.cov.bw(xpdb,
+   grid = TRUE)
> print(xplot)
```

**Code used to generate the graph:**

# autocorr.cwres

## Autocorrelation of Conditional weighted residuals (Run 1)



**Figure 57:** *The conditional weighted residuals versus the previous conditional weighted residual. Data points are indicated by blue circles and the red line is a smooth.*

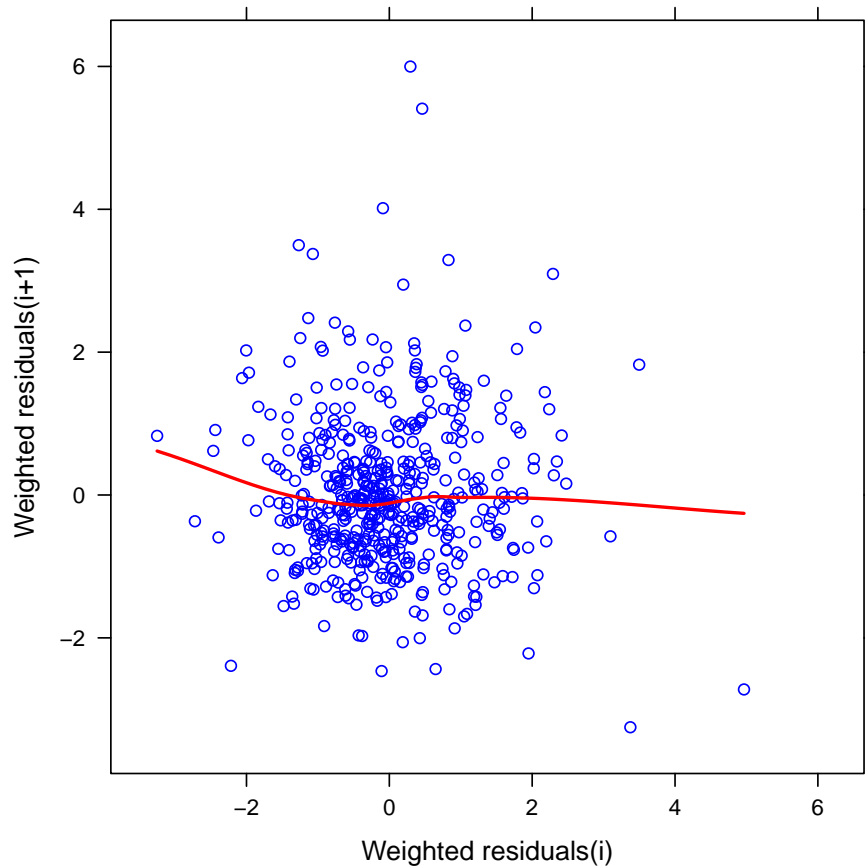
This graph is intended to check the degree of correlation among the residuals. A pronounced trend in the smooth around the center of the data would suggest that there is autocorrelation present.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")  
> xplot <- autocorr.cwres(xpdb)  
> print(xplot)
```

# autocorr.wres

## Autocorrelation of Weighted residuals (Run 1)



**Figure 58:** *The weighted residuals versus the previous weighted residual. Data points are indicated by blue circles and the red line is a smooth.*

This graph is intended to check the degree of correlation among the residuals. A pronounced trend in the smooth around the center of the data would suggest that there is autocorrelation present.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")  
> xplot <- autocorr.wres(xpdb)  
> print(xplot)
```

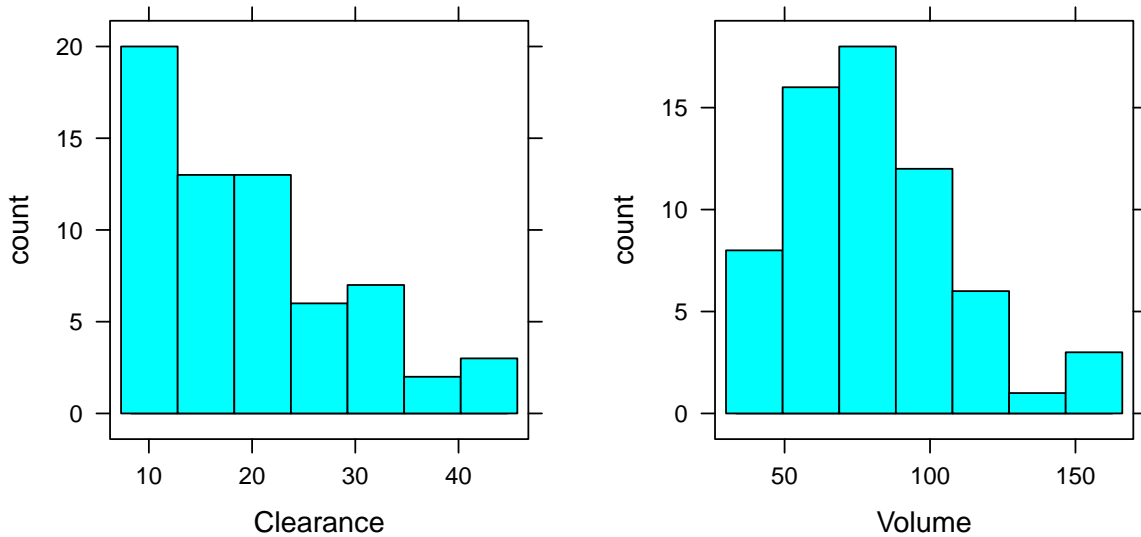
# Parameter distribution diagnostics

Statistical comparisons of competing models depends on the correct description of the statistical sub-parts of the model, including both the residual error distribution as well as the distributions of the between individual variability (BSV) terms in the model. The distributions of the BSV are also important for the simulation properties of the model. NONMEM will generate normally distributed BSV when simulating from the model. If the BSV distributions estimated from data are asymmetrically distributed, it is likely that the simulated data will not mimic the observed.



# parm.hist

## Distribution of parameters (Run 1)



**Figure 59:** Histograms of the posterior Bayes estimates of Clearance and Volume. Each bar represent the count of values in the corresponding interval on the x-axis.

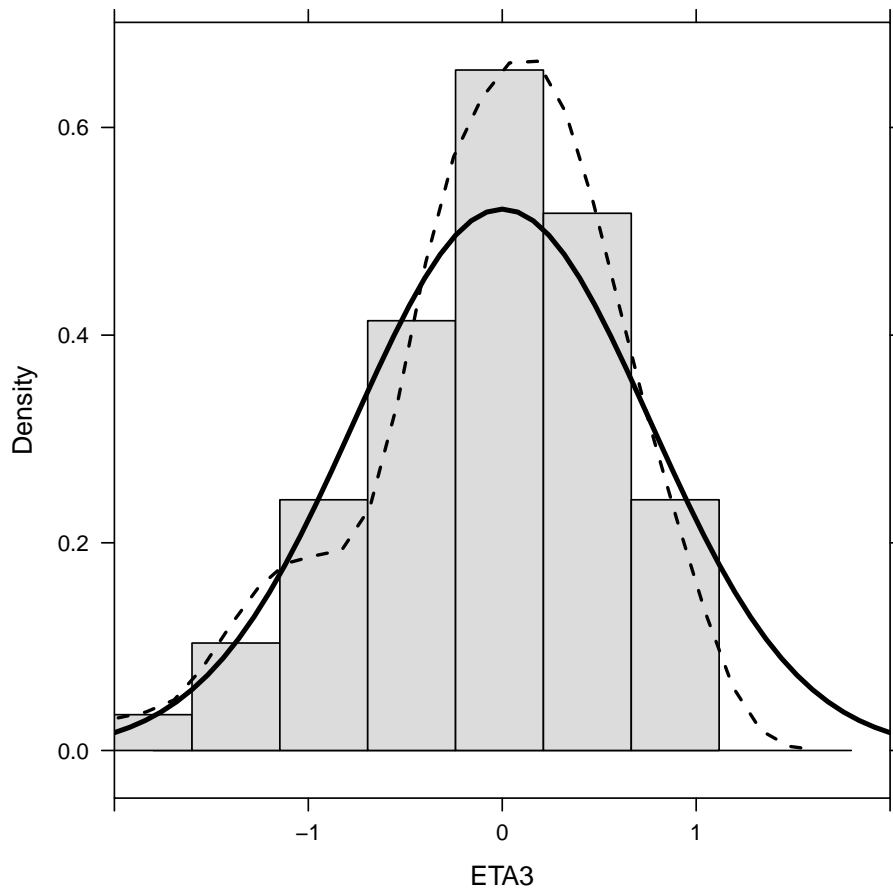
This graph is intended for visualization of the distribution(s) of the parameters. The parameters are typically the data items included in the patab or using the default xpose definition of parameters.

The expected shape of the distribution is given by the way that the parameter has been coded in the NONMEM model file.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "parms") <- c("CL",
+   "V")
> xplot <- parm.hist(xpdb, hidcol = "transparent",
+   type = "count", ylb = "count")
> print(xplot)
```

## ranpar.hist



**Figure 60:** Histogram of the posterior Bayes estimates of ETA3. The black dashed line is a smooth of the probability density while the solid black line is the theoretical density based on mean=0 and sd=0.7652.

This graph function is similar to histograms of the parameters (Section “parm.hist” on the preceding page). In this particular case the theoretical density has been superposed and can be used to judge the impact of shrinkage. The estimated sd for this parameter was 0.7652 and the shrinkage was 17%. From the graph it can be concluded that the shrinkage is mainly affecting the positive side of the dis-

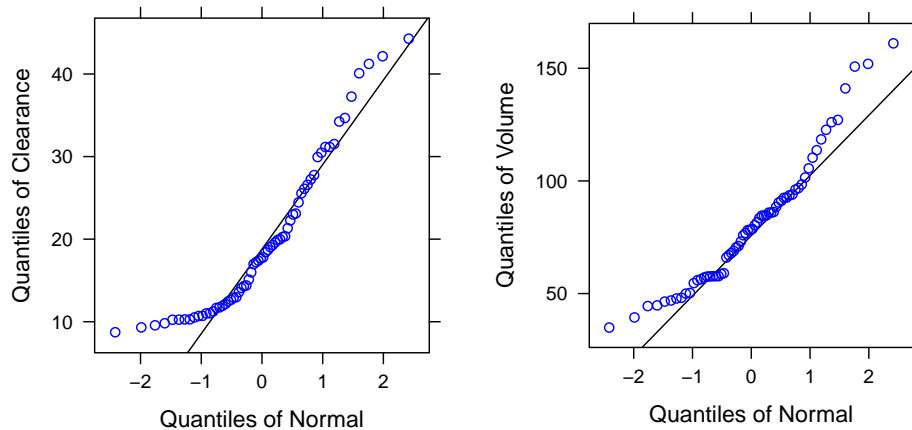
tribution.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "ranpar") <- c("ETA3")
> xplot <- ranpar.hist(xpdb, math.dens = list(mean
+   sd = 0.7652), xlim = c(-2,
+   2), hicol = "gainsboro", main = NULL)
> print(xplot)
```

# parm.qq

## Distribution of parameters (Run 1)



**Figure 61:** *The quantiles of the distributions if CL and V are plotted versus the quantiles of the Normal distribution. The diagonal black line is the line of identity.*

This graph can be used to compare the distribution of the POSTHOC parameters to the Normal distribution. If the parameter is normally distributed the data points will fall on the line of identity. This graph is very sensitive and will easily detect non-normalities. Since the individual parameters are usually modelled as being non-normal, this display may not be particularly useful (but see “ranpar.qq” on the next

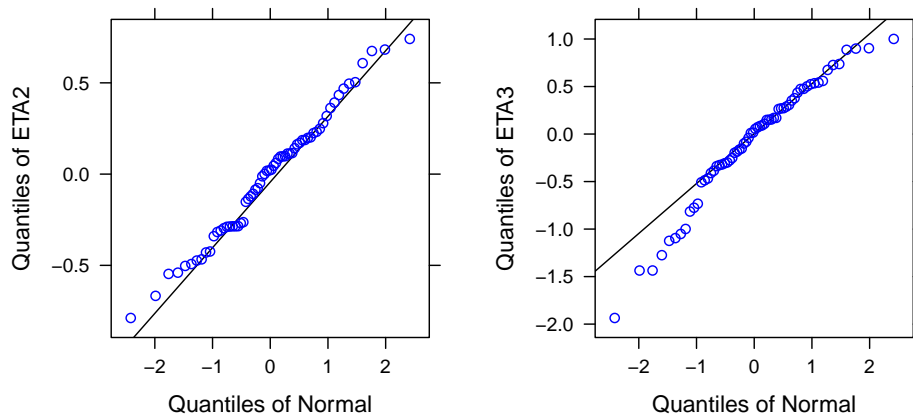
page).

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "parms") <- c("CL",
+   "V")
> xplot <- parm.qq(xpdb)
> print(xplot)
```

# ranpar.qq

## Distribution of random parameters (Run 1)



**Figure 62:** The quantiles of the distributions if ETA1 and ETA2 are plotted versus the quantiles of the Normal distribution. The diagonal black line is the line of identity.

This graph can be used to compare the distribution of the POSTHOC parameters or random effects to the Normal distribution. If the parameter is normally distributed the data points will fall on the line of identity. This graph is very sensitive and will easily detect non-normalities. Since  $\eta$ s are typically assumed normally or symmetrically distributed, this graph is a useful diagnostic for this as-

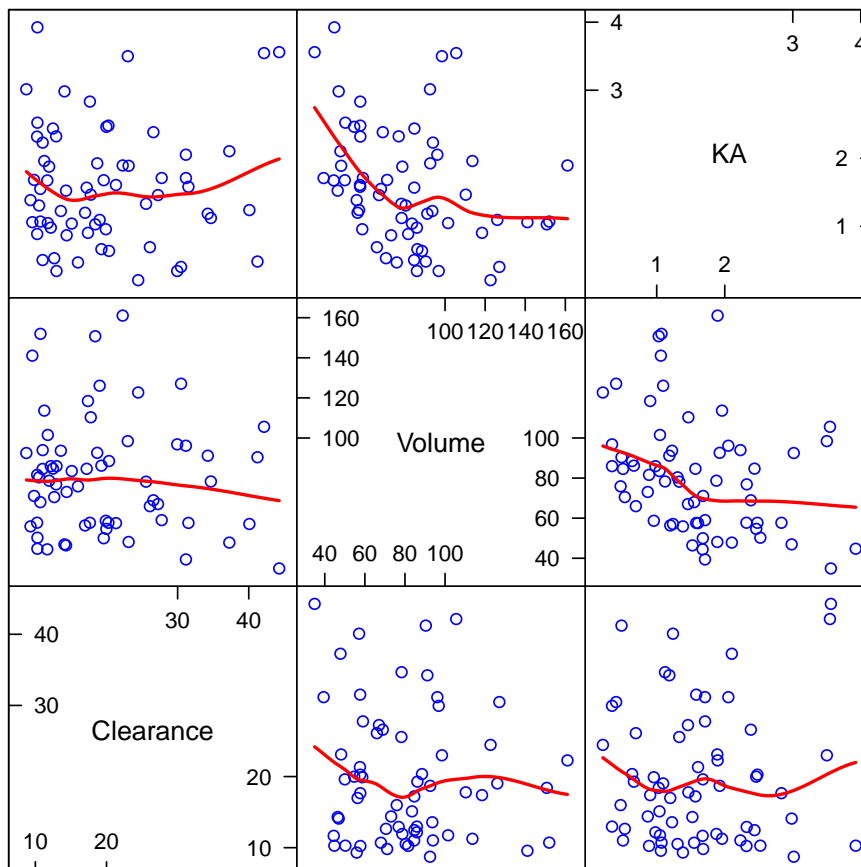
sumption.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "ranpar") <- c("ETA2",
+   "ETA3")
> xplot <- ranpar.qq(xpdb)
> print(xplot)
```

# parm.splom

Scatterplot matrix of parameters (Run 1)



**Figure 63:** The posterior Bayes estimates (blue circles) of the parameters are plotted versus each other. The red line is a smooth.

This graph is used to explore correlations between the parameters. Each pair of parameters are plotted twice, once in the upper triangle and once in the lower. However, since the parameter regarded as the x-variable is different for the upper and lower plot, the smooths will be different. Often, if a trend is only visible with one of the two (upper and lower) plots then the trend may be an artifact of the way the smooth is calculated. Since the individual parameters are typically not normally distributed, this display will look slightly different from the same one using the POSTHOC  $\eta$ s (see “ran-

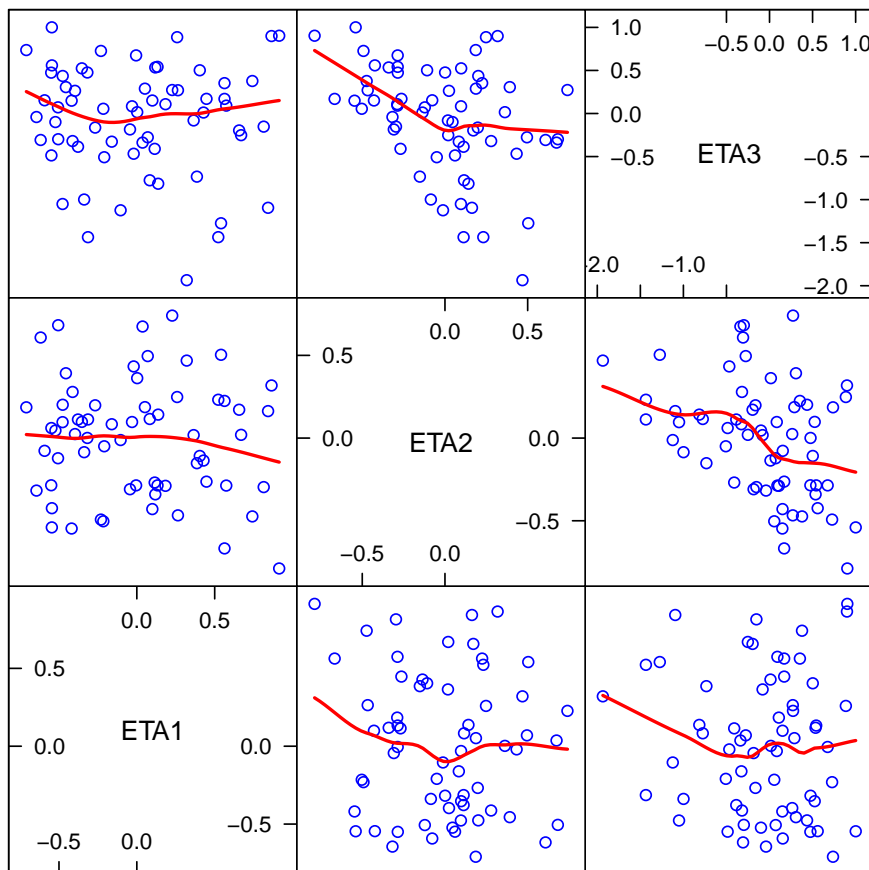
par.splom” on the following page). If covariates have already been included in the model, it is more appropriate to use the  $\eta$ -estimates instead of the individual parameters for this graph.

## Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "parms") <- c("CL",
+   "V", "KA")
> xplot <- parm.splom(xpdb)
> print(xplot)
```

# ranpar.splom

Scatterplot matrix of random parameters (Run 1)



**Figure 64:** The posterior Bayes estimates (blue circles) of the  $\eta$ s are plotted versus each other. The red line is a smooth.

This graph is used to explore the correlations between the random effects, and may be informative with respect to the OMEGA BLOCK structure in NONMEM. Since the distribution of the individual random effects are typically assumed normally or symmetrically distributed, it will look different to the same graph using the parameters (see “parm.splom” on the previous page). If covariates have been included in the model, the individual parameter estimates

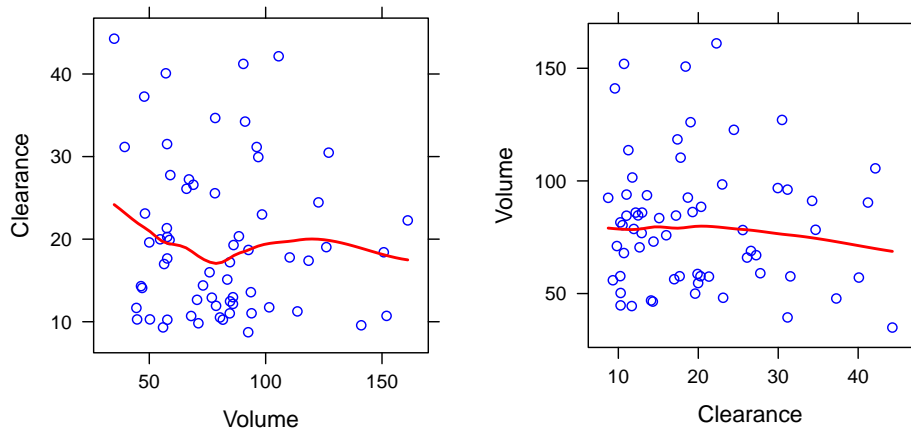
should not be used for investigating the correlation structure.

## Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "ranpar") <- c("ETA1",
+   "ETA2", "ETA3")
> xplot <- ranpar.splom(xpdb)
> print(xplot)
```

# parm.vs.parm

## Parameters vs. parameters (Run 1)



**Figure 65:** The posterior Bayes estimates (blue circles) of the parameters are plotted versus each other. The red line is a smooth. Extreme data points are labeled by the corresponding ID-number.

This graph is used to explore correlations between the parameters and fills the same role as the `parm.splom` function on page 77. The benefit of this display lies in the identification of extreme individuals. Additionally, if many parameters are compared then this function will produce a multiple page plot. It is possible to display the  $\eta$ s with this function by first changing the variable definition to the  $\eta$ s instead of

the parameters.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "parms") <- c("CL",
+   "V")
> xplot <- parm.vs.parm(xpdb)
> print(xplot)
```

## tabulate.parameters

```
+-----+-----+-----+
|Parameter| Value | RSE |
+-----+-----+-----+
|    TH1   |17.7481|0.059|
+-----+-----+-----+
|    TH2   |76.7973|0.052|
+-----+-----+-----+
|    TH3   |1.44256|0.108|
+-----+-----+-----+
|  OM1:1   |    0.45| 0.13|
+-----+-----+-----+
|  OM2:2   |    0.37| 0.19|
+-----+-----+-----+
|  OM3:3   |    0.77| 0.29|
+-----+-----+-----+
|  SI1:1   |    0.13|0.065|
+-----+-----+-----+
```

This command extracts the population parameters estimates and standard errors from the NONMEM list file. It is assumed that the list file is named: run#.lst, where # is the run-number extracted from the Xpose data object. The standard error information is presented as the

relative standard errors, i.e. SE/estimate.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> tabulate.parameters(xpdb, dir = "../")
```



## parm.summary

	Mean	SD	Q1	Median	Q3	Range	N
ETA3	-0.07386	0.6318	-0.3498	0.03522	0.3588	-1.935-1.001	64
ETA2	-0.007865	0.349	-0.2862	0.02196	0.1991	-0.7883-0.7406	64
ETA1	0.007579	0.4507	-0.4011	-0.00123	0.37	-0.7097-0.9142	64
KA	1.588	0.8672	1.017	1.495	2.065	0.2083-3.925	64
V	80.9	28.77	57.68	78.5	93.72	34.91-161.1	64
CL	19.81	9.374	11.88	17.73	25.7	8.728-44.28	64

This command tabulates the individual parameter estimates. Only the parameter values on the first individual data records are used unless `onlyfirst` is set to `FALSE`. It is also possible to re-direct the output to a `csv` file which can be imported into spreadsheet or word pro-

cessing programs.

### Code used to generate the graph:

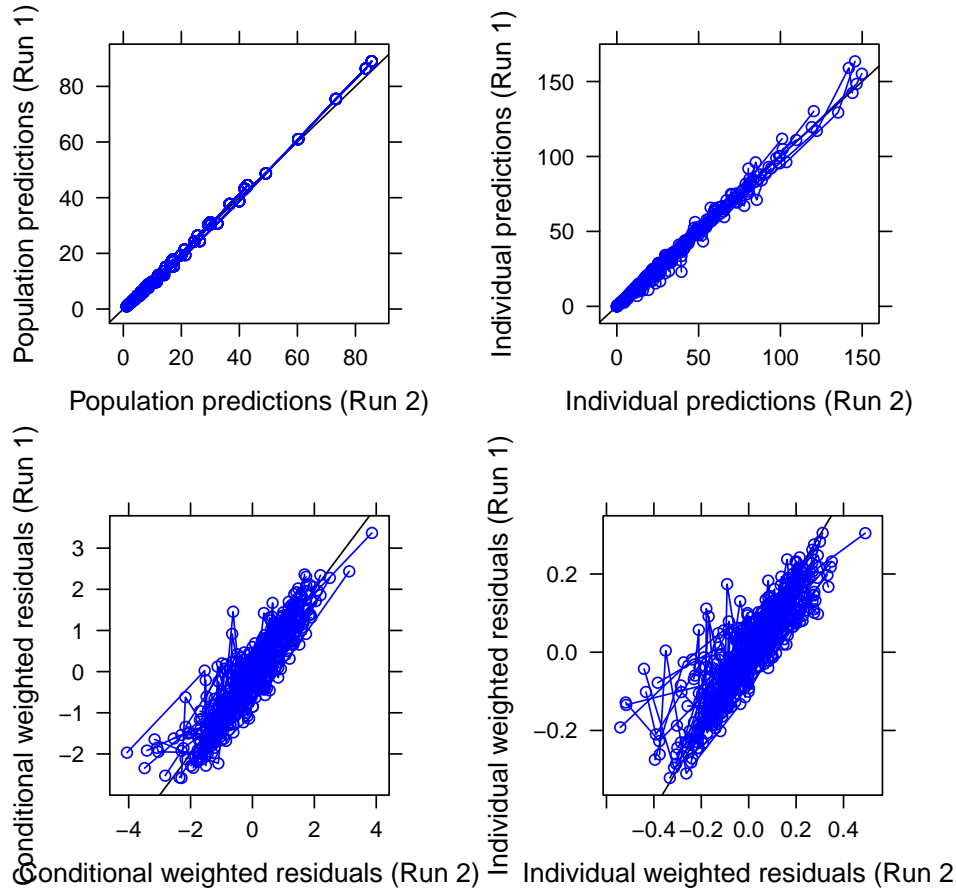
```
> xpdb <- xpose.data(1, directory = "../")
> parm.summary(xpdb)
```

# Model comparison

Simply put, building a model is the task of identifying the least bad model from a set of model candidates (remember that “All models are wrong but some are useful”). It is also important to understand the relative performance of two or more model candidates (for example, if one of the alternatives perform less well at high concentrations then this model should not be carried forward if it is important to characterise  $C_{max}$ ) or if the apparent benefit in terms of  $\Delta OFV$  with one model is dependent on only one or two individuals.

# basic.model.comp

lual weighted residuals (Run 1) vs Individual weighted residuals (



**Figure 66:** Comparison of Run 1 to Run 2. The top panels show the population and individual predictions for Run 1 and Run 2 versus each other. The bottom panels show the population and individual weighted residuals versus each other. Individual data points are indicated by blue circles and are connected by blue lines. The black diagonal lines are the lines of identity.

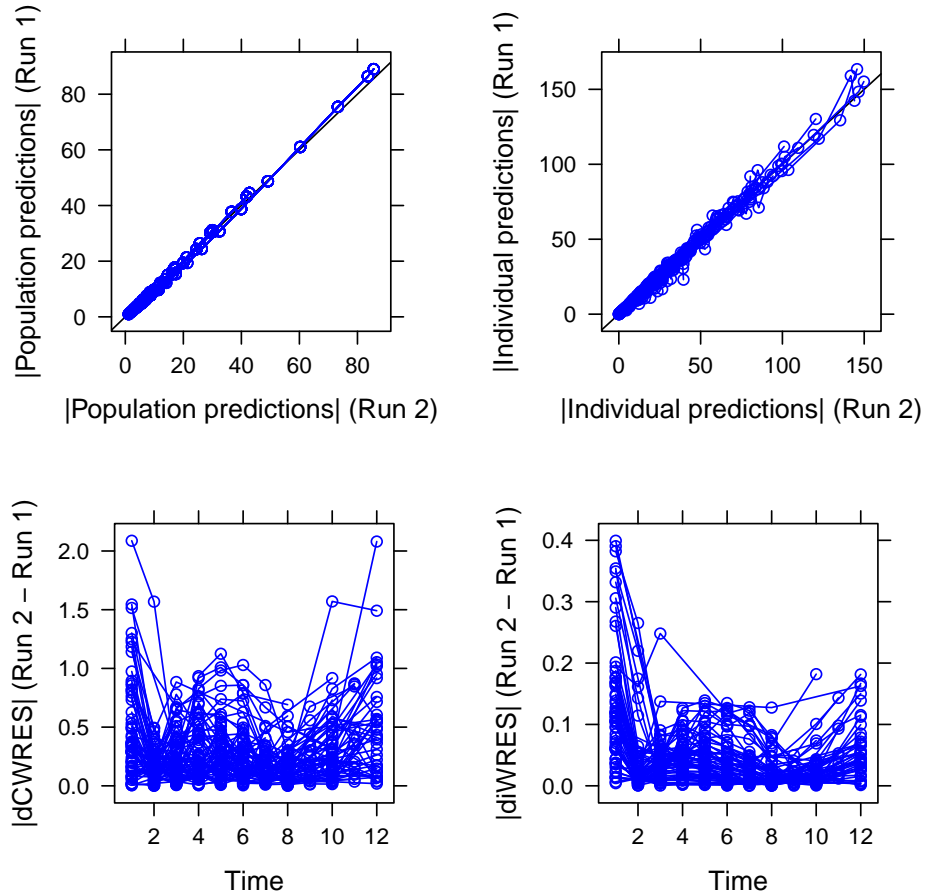
This graph is intended for visualization of the differences between two runs. In this particular case one OMEGA has been removed (KA), with a corresponding increase in OFV of about 110 points. It can be seen that the individual and population predictions are very similar and does not seem to depend on the removed  $\eta$ .

**Code used to generate the graph:**

```
> xpdb1 <- xpose.data(1, directory = "../")
> xpdb2 <- xpose.data(2, directory = "../")
> xplot <- basic.model.comp(xpdb2,
+   xpdb1)
> print(xplot)
```

# add.model.comp

## Additional model comparison plots (Run 2)



**Figure 67:** Comparison of Run 1 to Run 2. The top panels show the absolute values of the population and individual predictions for Run 1 and Run 2 versus each other. The bottom panels show the absolute values of the differences between the population and individual weighted residuals (CWRES and IWRES) of each run versus time. Individual data points are indicated by blue circles and are connected by blue lines. The black diagonal lines are the lines of identity.

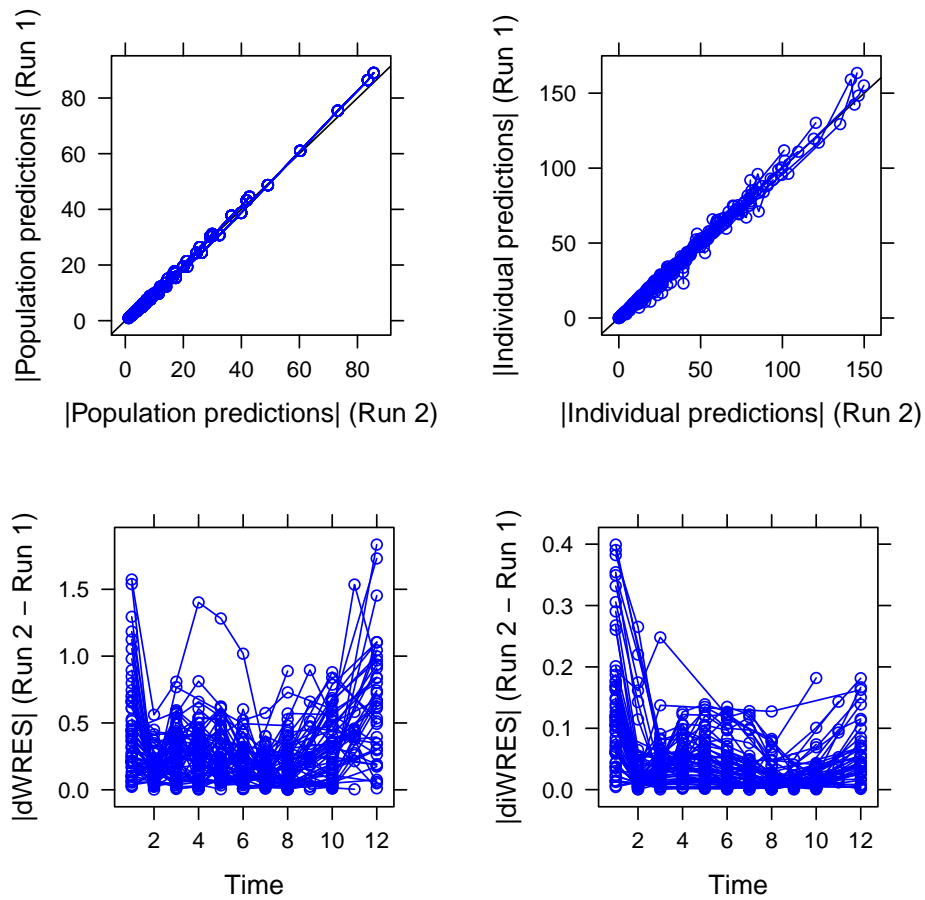
This graph is intended for visualization of the differences between two runs. The two top panels in this display is, in this example, the same as the two top panels in “basic.model.comp” on the preceding page (all predictions have the same sign). The two lower panels display the “residuals of the residuals”, i.e. the difference between the corresponding residual given the two models. The idea is that it is easier to see patterns in the dif-

ferences, if they are small, than if the residuals are plotted versus each other.

### Code used to generate the graph:

```
> xpdb1 <- xpose.data(1, directory = "../")
> xpdb2 <- xpose.data(2, directory = "../")
> xplot <- add.model.comp(xpdb2,
+   xpdb1)
> print(xplot)
```

### Additional model comparison plots (Run 2)



**Figure 68:** Comparison of Run 1 to Run 2. The top panels show the absolute values of the population and individual predictions for Run 1 and Run 2 versus each other. The bottom panels show the absolute values of the differences between the population and individual weighted residuals (WRES and IWRES) of each run versus time. Individual data points are indicated by blue circles and are connected by blue lines. The black diagonal lines are the lines of identity.

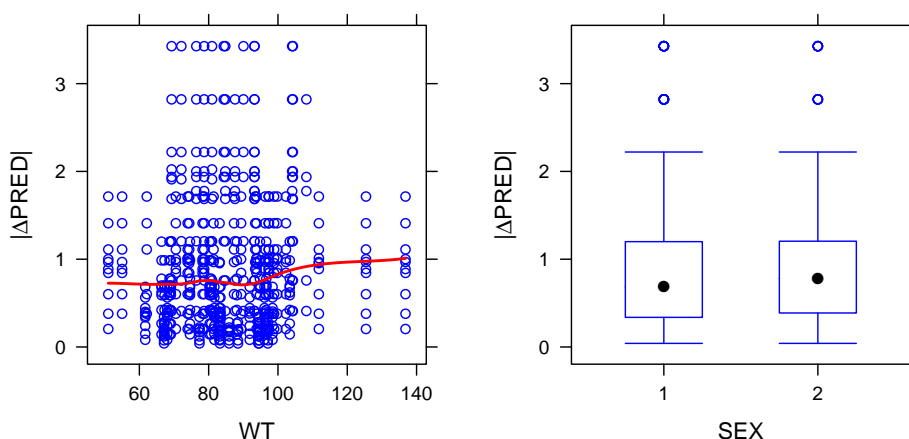
This display is identical to “add.model.comp” on the previous page except for the bottom left panel, which shows the weighted residuals rather than the conditional weighted residuals.

**Code used to generate the graph:**

```
> xpdb1 <- xpose.data(1, directory = "../")
> xpdb2 <- xpose.data(2, directory = "../")
> xplot <- add.model.comp(xpdb2,
+   xpdb1, force.wres = TRUE)
> print(xplot)
```

# absval.dpred.vs.cov.model.comp

**|PRED\_(Run2) – PRED\_(Run1)| vs. Covariates**



**Figure 69:** Comparison of Run 1 to Run 2. The absolute differences between the population predictions are plotted versus body weight and sex, respectively. The red line is a smooth.

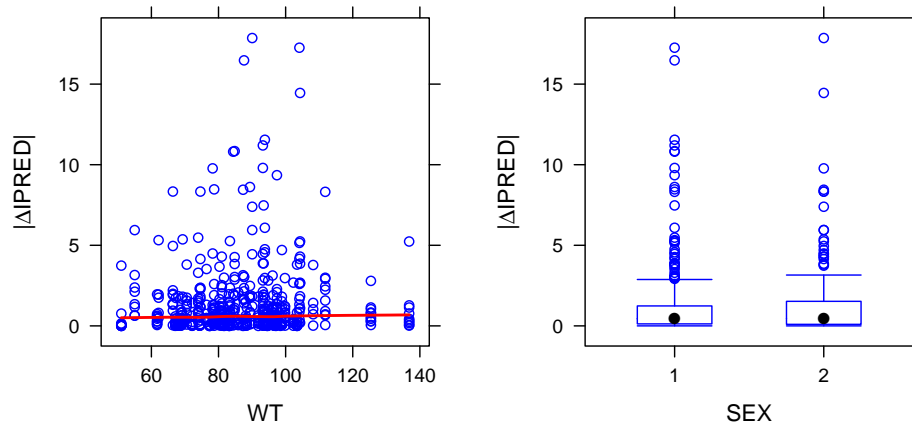
This display is used to investigate if the differences in population predictions between two models can be explained by covariates. If so, it may be possible to include this covariate effect in the model.

**Code used to generate the graph:**

```
> xpdb1 <- xpose.data(1, directory = "../")
> xpdb2 <- xpose.data(2, directory = "../")
> change.xvardef(xpdb2, "covariates") <- c("WT",
+     "SEX")
> xplot <- absval.dpred.vs.cov.model.comp(xpdb2,
+     xpdb1, smooth = TRUE)
> print(xplot)
```

# absval.dipred.vs.cov.model.comp

|IPRED\_(Run2) – IPRED\_(Run1)| vs. Covariates



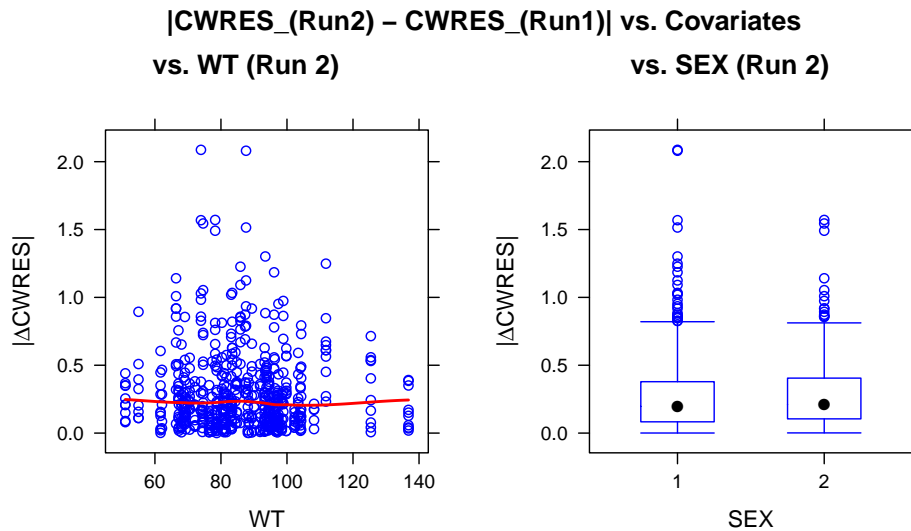
**Figure 70:** Comparison of Run 1 to Run 2. The absolute differences between the individual predictions are plotted versus body weight and sex, respectively. The red line is a smooth.

This display is used to investigate if the differences in individual predictions between two models can be explained by covariates. If so, it may be possible to include this covariate effect in the model.

**Code used to generate the graph:**

```
> xpdb1 <- xpose.data(1, directory = "../")
> xpdb2 <- xpose.data(2, directory = "../")
> change.xvardef(xpdb2, "covariates") <- c("WT",
+     "SEX")
> xplot <- absval.dipred.vs.cov.model.comp(xpdb2,
+     xpdb1, smooth = TRUE)
> print(xplot)
```

## absval.dcwres.vs.cov.model.comp



**Figure 71:** Comparison of Run 1 to Run 2. The absolute differences between the conditional weighted residuals are plotted versus body weight and sex, respectively. The red line is a smooth.

This display is used to investigate if the differences in conditional weighted residuals between two models can be explained by covariates. If so, it may be possible to include this covariate effect in the model.

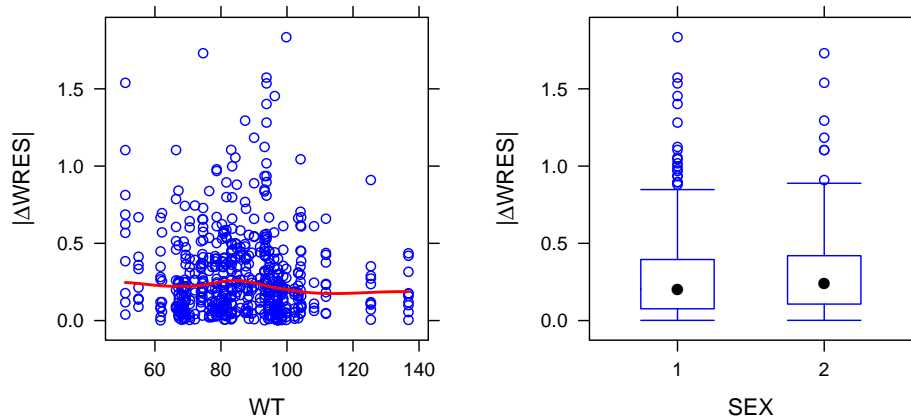
**Code used to generate the graph:**

```
> xpdb1 <- xpose.data(1, directory = "../")
> xpdb2 <- xpose.data(2, directory = "../")
> change.xvardef(xpdb2, "covariates") <- c("WT",
+     "SEX")
> xplot <- absval.dcwres.vs.cov.model.comp(xpdb2,
+     xpdb1, smooth = TRUE)
> print(xplot)
```



# absval.dwres.vs.cov.model.comp

**|WRES\_(Run2) – WRES\_(Run1)| vs. Covariates**



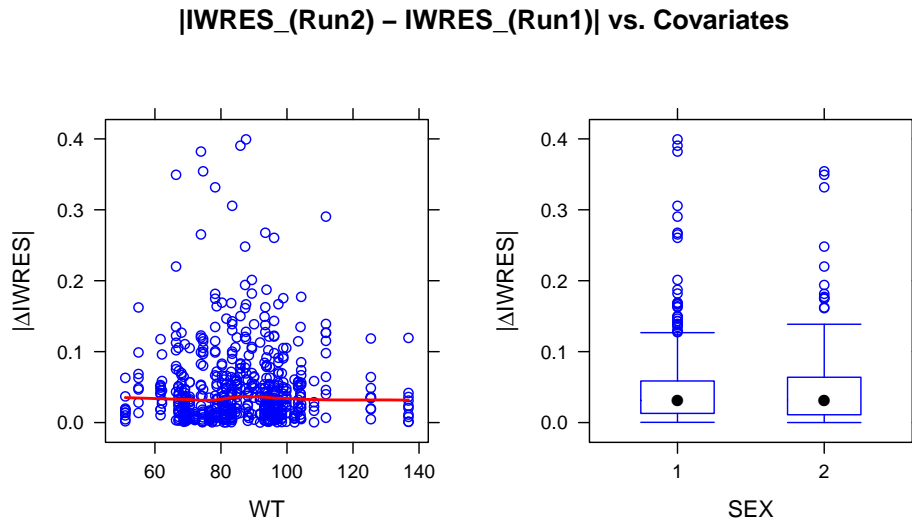
**Figure 72:** Comparison of Run 1 to Run 2. The absolute differences between the population weighted residuals are plotted versus body weight and sex, respectively. The red line is a smooth.

This display is used to investigate if the differences in population weighted residuals between two models can be explained by covariates. If so, it may be possible to include this covariate effect in the model.

**Code used to generate the graph:**

```
> xpdb1 <- xpose.data(1, directory = "../")
> xpdb2 <- xpose.data(2, directory = "../")
> change.xvardef(xpdb2, "covariates") <- c("WT",
+     "SEX")
> xplot <- absval.dwres.vs.cov.model.comp(xpdb2,
+     xpdb1, smooth = TRUE)
> print(xplot)
```

# absval.diwres.vs.cov.model.comp



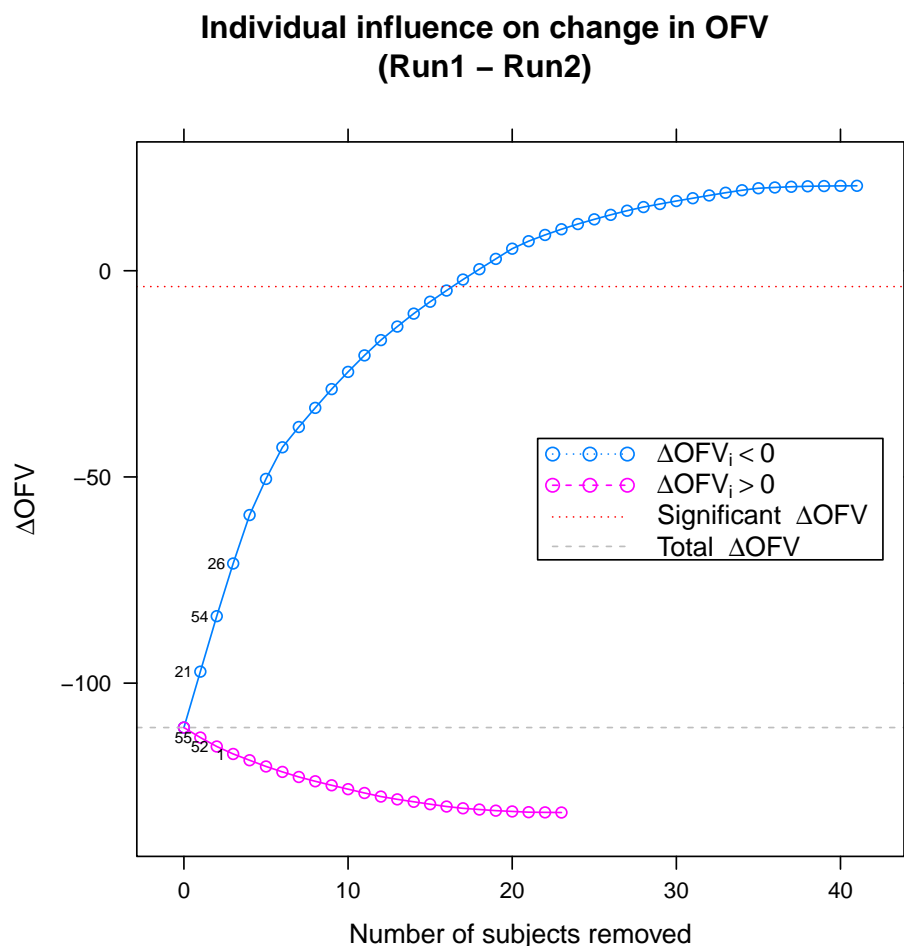
**Figure 73:** Comparison of Run 1 to Run 2. The absolute differences between the individual weighted residuals are plotted versus body weight and sex, respectively. The red line is a smooth.

This display is used to investigate if the differences in individual weighted residuals between two models can be explained by covariates. If so, it may be possible to include this covariate effect in the model.

**Code used to generate the graph:**

```
> xpdb1 <- xpose.data(1, directory = "../")
> xpdb2 <- xpose.data(2, directory = "../")
> change.xvardef(xpdb2, "covariates") <- c("WT",
+     "SEX")
> xplot <- absval.diwres.vs.cov.model.comp(xpdb2,
+     xpdb1, smooth = TRUE)
> print(xplot)
```

# dOFV.vs.id



**Figure 74:** Comparison of Run 1 to Run 2. Individuals are ranked based on the difference in individual OFV between the two runs. Then the individual that most benefits from run 2 (blue) and most benefit from run 1 (magenta) are removed from the  $\Delta OFV$  calculation along the x-axis. The dashed red line indicates the critical value for a significance at  $p < 0.05$ . The dashed grey indicates the total change in OFV going from run 1 to run 2.

This display, also known as a “shark plot” is used to investigate how the OFV differences between the new model (run 1 in this case) and the reference model (run 2) is distributed across individuals. If a few individuals are driving the difference between run1 and run2 then the plot will quickly move from below the red line to above the red line as the x-axis in-

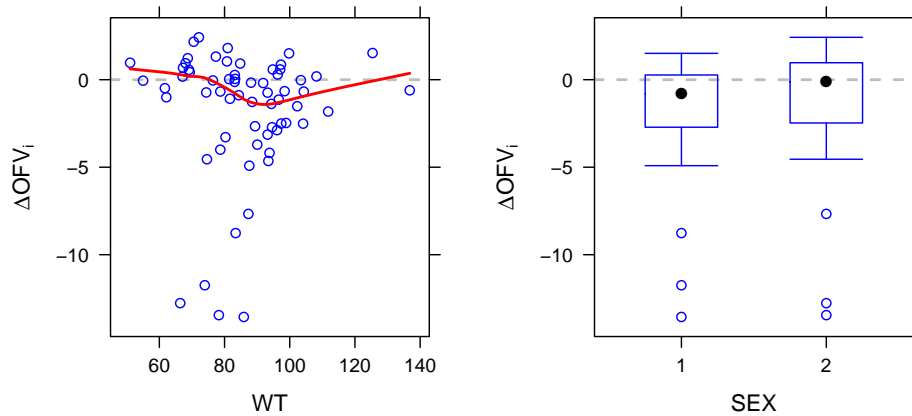
creases.

## Code used to generate the graph:

```
> xpdb1 <- xpose.data(1, directory = "../")
> xpdb2 <- xpose.data(2, directory = "../")
> xplot <- dOFV.vs.id(xpdb2, xpdb1)
> print(xplot)
```

# dOFV.vs.cov

Individual change in OFV vs. Covariate(s)  
(Run1 – Run2)



**Figure 75:** Comparison of the change in individual OFV between Run 1 and Run 2 over covariates. The gray dashed line indicates no difference and the red line is a smooth.

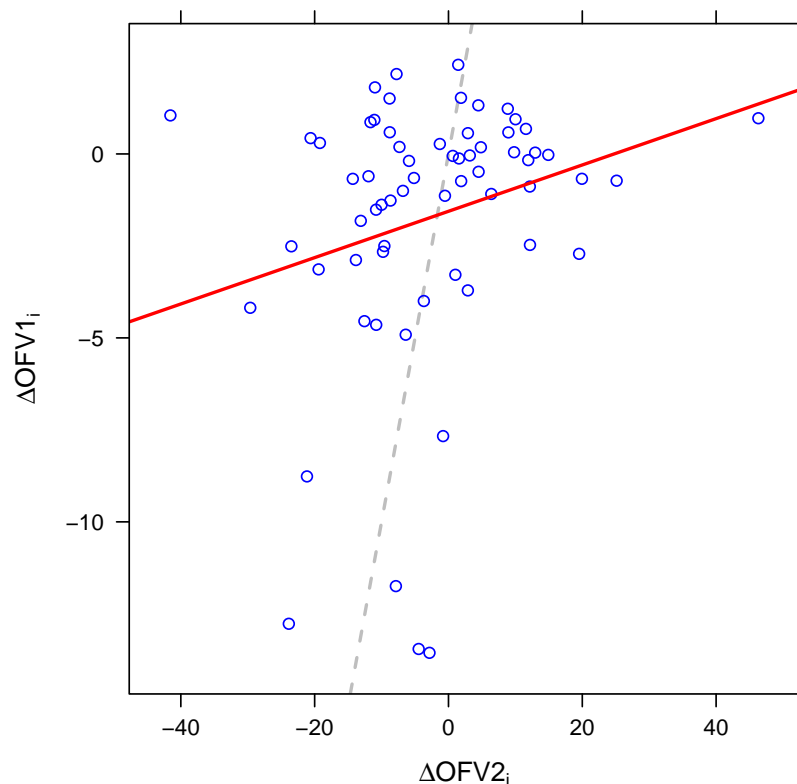
This display can be used to identify and evaluate if certain covariate related sub-groups of individuals benefit more from the model change.

**Code used to generate the graph:**

```
> xpdb1 <- xpose.data(1, directory = "../")
> xpdb2 <- xpose.data(2, directory = "../")
> change.xvardef(xpdb2, "covariates") <- c("WT",
+     "SEX")
> xplot <- dOFV.vs.cov(xpdb2, xpdb1,
+     smooth = TRUE, max.plots = 4)
> print(xplot)
```

# dOFV1.vs.dOFV2

Individual change in OFV vs. OFV2  
(OFV 1 = Run1 – Run2)  
(OFV 2 = Run3 – Run2)



**Figure 76:** Comparison of the improvements with Run 1 to Run 2, and Run 3 to Run 2. The individuals differences in OFV for Run 1 to Run 2 are plotted versus the individual differences in OFV for Run 3 versus Run 2. The grey dashed line is the line of identity and the red solid line is a linear regression line.

This graph can be used to investigate to what extent the difference in OFV obtained with one model are correlated to the differences obtained with another model. A possible conclusion can be that the two contending models are equivalent.

```
> xpdb1 <- xpose.data(1, directory = "../")
> xpdb2 <- xpose.data(2, directory = "../")
> xpdb3 <- xpose.data(3, directory = "../")
> xplot <- dOFV1.vs.dOFV2(xpdb2,
+   xpdb1, xpdb3)
> print(xplot)
```

**Code used to generate the graph:**

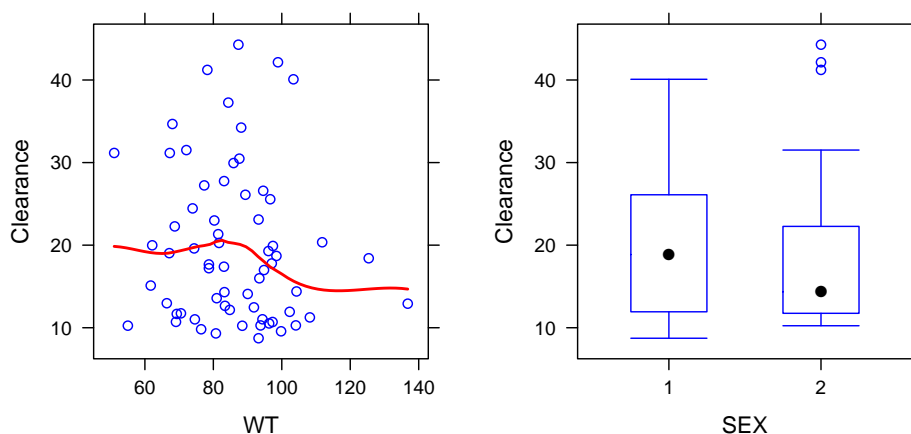
# Model development

Model development usually involves trying many alternative candidate models. This can be a tedious and time consuming task. Conversely, if this process can be made more efficient, model development time can be shortened and results and conclusions obtained more quickly.

The time required for developing a model can be reduced by excluding unlikely model candidates by graphical examination or by automating model building tasks. The functions in this section are specifically designed to do both of these tasks.

# parm.vs.cov

## Parameters vs. covariates (Run 1)



**Figure 77:** The individual parameter estimates (blue circles) are plotted versus the covariates. The red line is a smooth.

This graph is used to investigate if there are any trends between the individual parameters and the covariates. Note that there will be trends in these graphs if the covariates are included as predictors for the parameters in the model. To investigate if there are any remaining trends between the parameters and covariates *after the covariates have been included in the model* it is better to use the `ranpar.vs.cov` (96). function. Shrinkage may decrease the

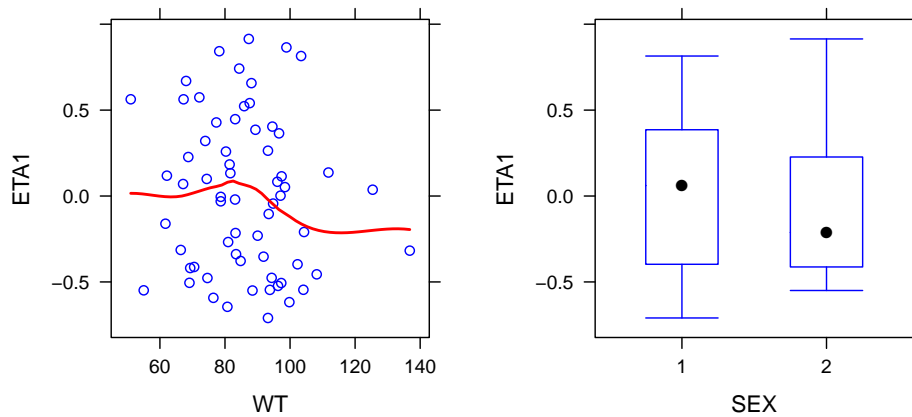
usefulness of this graph.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "parms") <- c("CL")
> change.xvardef(xpdb, "covariates") <- c("WT",
+    "SEX")
> xplot <- parm.vs.cov(xpdb)
> print(xplot)
```

# ranpar.vs.cov

## Parameters vs. covariates (Run 1)



**Figure 78:** The posterior Bayes estimates (blue circles) of the  $\eta$ s are plotted versus the covariates. The red line is a smooth.

This graph is used to investigate if there are any trends between the posterior Bayes estimates of the  $\eta$ s and the covariates. If no covariates have been included in the model, this graph will provide the same information as `parm.vs.cov` on page 95, except for differences caused by the parameterization of the relationship between the individual parameter and the  $\eta$ . Since it is what is remaining after the covariates have been accounted for that is plotted versus the covariate, this graph is useful if one wish to assess to what extent

the covariate model is appropriate. However, shrinkage may be an issue as with all posterior Bayes estimates.

### Code used to generate the graph:

```
> xpdb <- xpose.data(1, directory = "../")
> change.xvardef(xpdb, "ranpar") <- c("ETA1")
> change.xvardef(xpdb, "covariates") <- c("WT",
+    "SEX")
> xplot <- ranpar.vs.cov(xpdb)
> print(xplot)
```



## xpose.gam

The `xpose.gam` function can be used to run a stepwise GAM analysis (Mandema et al; *J Pharmacokinetic Biopharm*, 20:511-528, **1992**). It is typically run from the Xpose classic menu interface but can also be invoked on the command line.

To run the `xpose.gam` on the command line:

```
Looking for NONMEM table files.
  Reading ../xptab3
Table files read.
  Reading ../run3.phi

Looking for NONMEM simulation table files.
No simulated table files read.

> gobj1 <- xpose.gam(xpdb, parnam = "ETA1")

Start:  ETA1 ~ 1; AIC= 48.5707
Trial:  ETA1 ~ HT + 1 + 1 + 1 + 1; AIC= 45.8927
Trial:  ETA1 ~ 1 + WT + 1 + 1 + 1; AIC= 43.632
Trial:  ETA1 ~ 1 + 1 + AGE + 1 + 1; AIC= 49.2928
Trial:  ETA1 ~ 1 + 1 + 1 + RACE + 1; AIC= 52.5597
Trial:  ETA1 ~ 1 + 1 + 1 + 1 + SEX; AIC= 41.48
Step :  ETA1 ~ SEX ; AIC= 41.48

Trial:  ETA1 ~ HT + 1 + 1 + 1 + SEX; AIC= 43.4297
Trial:  ETA1 ~ 1 + WT + 1 + 1 + SEX; AIC= 39.9721
Trial:  ETA1 ~ 1 + 1 + AGE + 1 + SEX; AIC= 41.9986
Trial:  ETA1 ~ 1 + 1 + 1 + RACE + SEX; AIC= 43.6867
Step :  ETA1 ~ WT + SEX ; AIC= 39.9721

Trial:  ETA1 ~ HT + WT + 1 + 1 + SEX; AIC= 41.8201
Trial:  ETA1 ~ 1 + ns(WT, df = 2) + 1 + 1 + SEX; AIC= 40.0188
Trial:  ETA1 ~ 1 + WT + AGE + 1 + SEX; AIC= 40.1546
Trial:  ETA1 ~ 1 + WT + 1 + RACE + SEX; AIC= 42.3149

> gobj1

Call:
gam(formula = ETA1 ~ WT + SEX, data = gamdata, trace = FALSE)

Degrees of Freedom: 63 total; 61 Residual
Residual Deviance: 6.17538
```

The function returns an Xpose gam object, which can be used for plotting purposes.

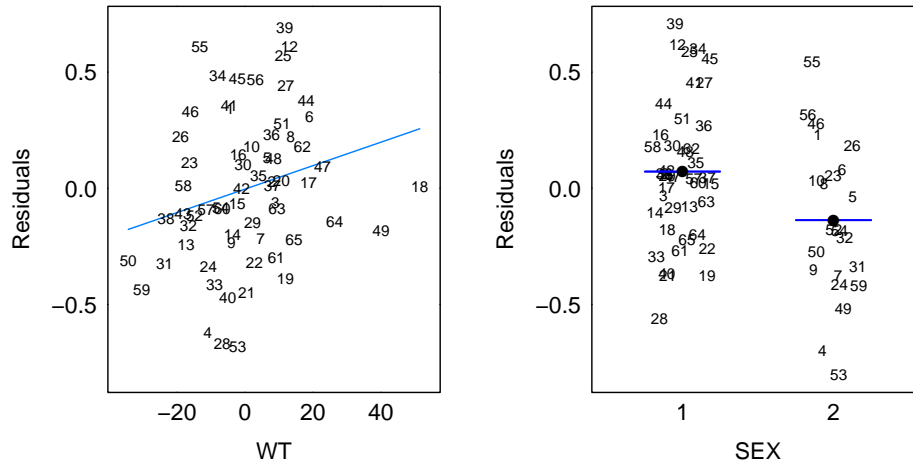
The on-screen output from the `xpose.gam` function is the trace information from the search, from which it is possible to deduce which covariates that were judged to be significant according to the AIC criterion. However, it is easier to assess this graphically using the `xp.plot` function (page 99).

By default all covariates defined in the Xpose database will be used in the search. To limit this, one can either change the covariate scope using the `xvardef` function before running `step.gam` or by using the argument `covnams`, e.g. `covnams=c("CL", "WT")`.

There are a number of search parameters that can be changed as well. See the `xpose.gam` help page (`>=?xpose.gam`).

# xp.plot

## GAM results for covariates on ETA1 (Run 3)



**Figure 79:** The partial residual of each term in the final stepwise GAM model versus the covariates. The continuous covariate (WT) is centered around its median value. The blue line in the left panel is a smooth and the blue lines in the right panel are the medians in each covariate group. Each data point is labeled by an ID number.

This display visualizes the fit of the final stepwise GAM model. The partial residuals, i.e. the residuals obtained when the covariate is removed from the model, versus each covariate are plotted. Continuous covariates are centered around the median value (to mimic what is typically done in the NONMEM model).

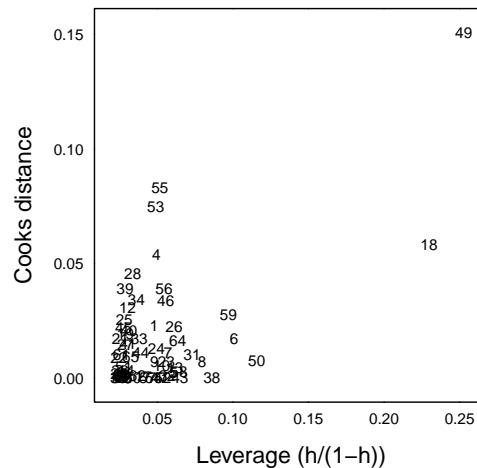
The plots can be used to hint at the covariate model one might use when testing in NONMEM.

### Code used to generate the graph:

```
> xplot <- xp.plot(gamobj = gobj1)
> print(xplot)
```

# xp.ind.inf.fit

Individual influence on the GAM fit for ETA1 (run 3)



**Figure 80:** The Cooks distance versus the leverage. Each data point is labeled by the ID number.

This graph is used to identify individuals that are overly influential in the GAM model. Such individuals have a high Cooks distance (large impact on parameter estimates when left out of the data set) and a high leverage (a high impact on the uncertainty of the parameter estimates when left out of the data set). If there are such individuals, they can be omitted (using the `subset` argument to the `xp.gam` func-

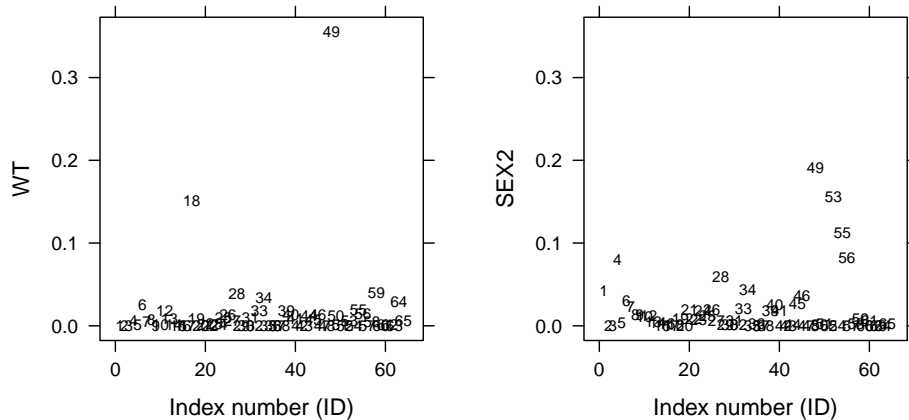
tion) to investigate if they would also influence the covariate selection. Note that an individuals may be highly influential without being incorrect.

## Code used to generate the graph:

```
> xplot <- xp.ind.inf.fit(gamobj = gobj1,  
+   aspect = 1)  
> print(xplot)
```

# xp.ind.inf.terms

Individual influence (Cooks distance) on each GAM term  
for ETA1 (Run 3)



**Figure 81:** The Cooks distance versus the ID number. Each data point is labeled by the ID number.

The `xp.ind.inf.fit` graph on page 100 visualizes the individual influence on the overall fit of the GAM model. In contrast, what the `xp.ind.inf.terms` function does is to investigate the Cooks distances for each term in the final GAM model. In other words, it is possible to identify if there is any specific covariate effect

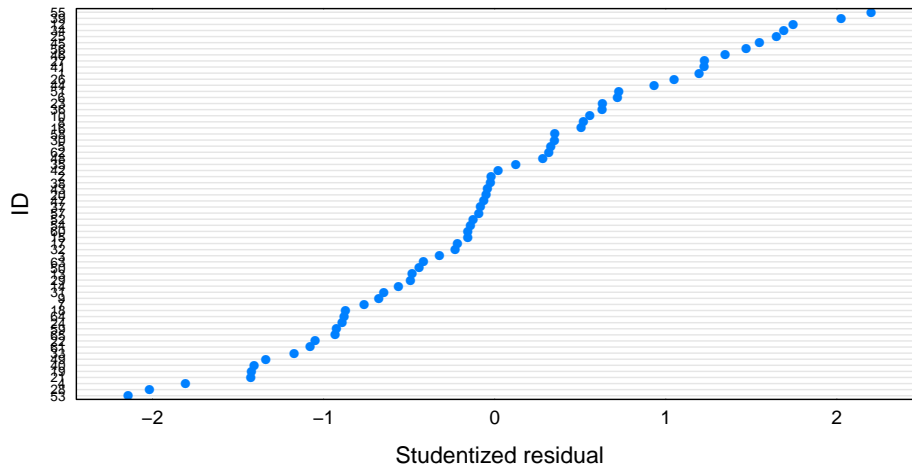
that is driven by one or a few individuals.

**Code used to generate the graph:**

```
> xplot <- xp.ind.inf.terms(gamobj = gobj1,
+   aspect = 1)
> print(xplot)
```

# xp.ind.stud.res

Studentized residual of the GAM fit for ETA1 (Run 3)



**Figure 82:** *The ID number versus the Studentized residuals. The ID numbers are ordered according to the residuals.*

Studentized residuals are scaled residuals such that their standard deviation is 1. This means that we can interpret the scale of the residuals and judge to what extent any residuals are much larger than what should be expected. In this graph, the ID numbers, ordered according to the Studentized residuals, are plotted against the latter. The scale on the x-axis is in standard deviation meaning that

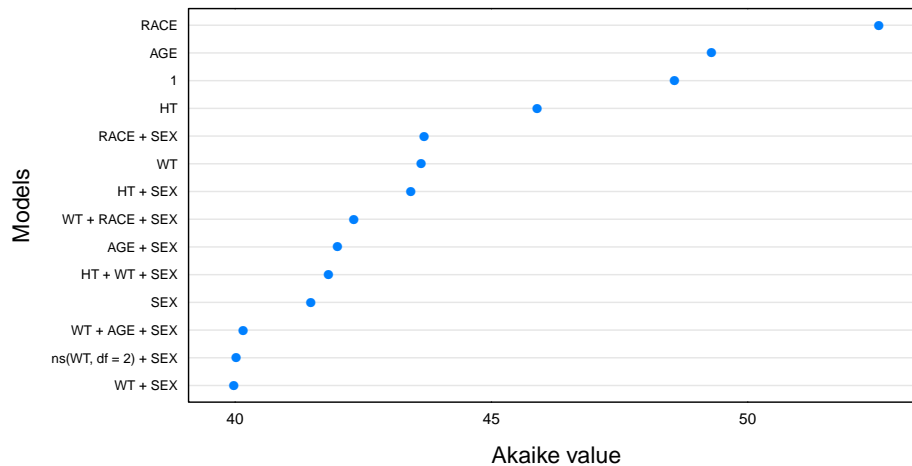
residuals outside -2 and 2 have a probability that is less than  $\approx 2.5\%$ . It may be perfectly reasonable for an individual to fall outside -2 or 2 but it may be worthwhile investigating these individuals more closely.

## Code used to generate the graph:

```
> xplot <- xp.ind.stud.res(gamobj = gobj1)
> print(xplot)
```

# xp.akaike.plot

AIC values from stepwise GAM search on ETA1 (Run 3)



**Figure 83:** The best models tested in the stepwise GAM analysis, ordered according to their Akaike values, versus the Akaike value.

This is a diagnostic for the stepwise GAM analysis. It visualizes the (up to 30) best models tried during the search. If the final model has an Akaike value that is well separated from other models, one may be more confident that the GAM search found the most important predictors in the correct functional form. Conversely, if the separation is small, one should

probably also regard the models close to the final model as likely indicators of which the important covariates and/or functional forms might be.

## Code used to generate the graph:

```
> xplot <- xp.akaike.plot(gamobj = gobj1)
> print(xplot)
```

# Getting more help

All `xpose.specific` functions have R-help files. They are accessible from the various R GUIs, and how you find them then depend on the GUI you are using. A more generic way to find help in R is to type `?function-name` on the R command line, e.g.:

```
> ?dv.vs.idv
```



# What is not included?

The intention with this document is to demonstrate all the functions available in `xpose.specific`. However, this version of the document will not completely fulfill that objective. A majority of the functions not covered in this document require simulated data in addition to typical NONMEM table files. These functions require additional NONMEM runs to produce; a topic we will cover in a future version of this document.

Below is a list, with a short description, of the functions that are not included. They also have the usual help pages in R so you are not left completely alone should you want to use them.

`cat.dv.vs.idv.sb` This function generates a goodness of fit graph for ordered categorical data. It is basically a stacked bar chart with the independent variable, for example dose or intervals of time, as the x-variable and stacked bars of the observations as the y-variable.

`kaplan.plot` A function that visualizes the data and VPC from survival type models in NONMEM.

`xpose.VPC` This is the function that forms the post-processor of the VPC output from PsN.

`xpose.VPC.categorical` This is the PsN post-processor for VPCs of categorical variables.

`xpose.VPC.both` Another VPC post-processor. It is a combination of `xpose.VPC` and `xpose.categorical`.

`cat.pc` Categorical (visual) predictive checks, without the need for PsN.

`npc.coverage` Implements the graphical back-end to the PsN npc output.

In addition to the above functions, there are also functionality, common to all `xpose.specific` functions that have been left out of this document. Most notably the mirror functionality has not been mentioned. Mirror functionality allows xpose to generate a reference plot to a goodness of fit plot by simulating data. The basic idea is that we don't really know what a goodness of fit plot should look like *if the model is correct* (admittedly, some plots we have a fair expectation for, but there are cases when we don't), but by simulating data from the model and then making the same goodness of fit plot based on the simulated data, we can learn what the plot should look like *if the model is correct*.

This functionality, which is available for all `xpose.specific` functions, can be invoked by specifying `mirror=T` as an argument. It is expected that there are simulated versions of the table files in the same directory as the real table files are. PsN can be asked to set these simulations up, e.g. execute `-mirror_plots='integer'`, where integer is the number of simulated data sets to generate.

# How to make NONMEM generate input to Xpose

## How Xpose identifies NONMEM runs

Xpose recognizes NONMEM runs, and files associated to a particular run, through the *run number*. This is a number that is used in the name of NONMEM model files, output files and table files.

The fundamental input to Xpose is one or more NONMEM table files. Even though Xpose recognizes and can read more than one table file from a single NONMEM run, the most convenient approach is to use only one table file. This table file should be named `xptab` followed by the run number, for example `xptab1` for run number 1.

The other table files that Xpose reads are `sdtab`, `patab`, `catab`, `cotab`, `mytab`, `extra` and `cwtab`. The contents of some of these files influence the way that Xpose interprets the data (see below).

## Xpose data variables

### Default variable definitions

Within the functions in the Xpose specific library all graphs are defined using *Xpose data variables*. These variables can be viewed as pointers to one or more columns in the NONMEM table file(s). For example, by default the `id` data variable points to the ID column in the NONMEM table file. The default definitions of the Xpose data variables are shown in Table “Xpose variable definitions” on the following page.

Xpose uses the data variable definitions in the specific functions to determine which data to plot or tabulate. For example, the function `dv.vs.idv` will plot the column in the NONMEM table files indicated by the `dv` data variable (default DV) versus the column indicated by the `idv` data variable (default TIME). This allows for a great deal of flexibility since the variable definitions can be changed by the user.

### Changing the default variable definitions

One useful thing to remember when using the `xpose.specific` functions is that they are defined in terms of the Xpose variable definitions (Table “Xpose variable definitions” on the next page). In other words, the functions `dv.vs.idv` and `pred.vs.idv` only deviate from each other in which Xpose variables they use as the y-variable (`dv` or `pred`).

It is possible to change the definition of these variable and, in effect, change the type of graph a particular function would generate. For example, the code

**Table 1: Xpose variable definitions**

Name	Default value	Description
id	ID	Used to identify individuals.
idlab	ID	Used to label data points.
idv	TIME, TAD or CP (in that order)	Used as the independent variable.
dv	DV or the variable in the fourth column from the right in the table files (checked in the order sdtab, patab, catab and cotab)	Used as the dependent variable.
occ	OCC	Identifies occasions. Presently not used in any Xpose specific functions.
pred	PRED	Used as the population predictions.
ipred	IPRED or IPRE	Used as the individual predictions.
wres	WRES	Used as the weighted residuals.
iwres	IWRES or IWRE	Used as the individual weighted residuals.
cwres	CWRES	Used as the conditional weighted residuals.
res	RES	Identifies the residuals. Presently not used in any Xpose specific functions.
parms	CL, V, V1, V2, V3, Q, Q1, Q2, Q3, KA, ETA1, ETA2, ETA3, ETA4, ETA5, ETA6, ETA7, ETA8, ETA9, ET10, ET11, ET12, ET13, ET14, ET15, ET16, ET17, ET18, ET19 and ET20 + any variables in the 2nd to (n-4)th columns in the patab	Used as the parameters.
ranpar	ETA1, ETA2, ETA3, ETA4, ETA5, ETA6, ETA7, ETA8, ETA9, ET10, ET11, ET12, ET13, ET14, ET15, ET16, ET17, ET18, ET19 and ET20	Used as the random effects parameters.
tvparms	TVCL, TVV, TVV1, TVV2, TVV3, TVQ, TVQ1, TVQ2, TVQ3 and TVKA	Used as the typical individual parameters.
covariates	GEN0, SEX, RACE, DOSE, FLAG, DAY, PAT, GEND, AGE, WT, HT, CRCL and CLCR + any variables in the 2nd to the (n-4)th columns in the patab	Used as the covariates.

```
> change.xvardef(xpdb, "dv") <- c("PRED")  
> xplot <- dv.vs.idv(xpdb)  
> print(xplot)
```

would produce a plot that looks like Figure “pred.vs.idv” on page 12 even though the function usually produces the graph in Figure “dv.vs.idv” on page 11.

# How Xpose identifies data items

The `xptab` should contain all data items that Xpose will need to produce the desired graphs.

Xpose recognize data items by name and will treat variables as either continuous or categorical based on the number of unique values in the variable.

# How to extend and modify the behavior of Xpose specific functions

Basically, every aspect of the Xpose plots can be influenced by specifying arguments to the function being called. In the examples above, we have tried to exemplify some of these arguments.

If you feel that an Xpose plot is close to, but not exactly what you want, check the relevant help page (104). Please also note that most standard lattice arguments can be passed as arguments to these functions.