Using Fit

Perry Haaland Daniel Samarov Elaine McVey

March 26, 2013

Abstract

Robust and unbiased methods for curve fitting and calibration in the analysis of biological and chemical assays are of great importance for ensuring reliable and interpretable results. Two issues which repeatedly arise in the analysis of assay data are heteroscadisticity and failure to capture the entire range of values in the independent variable. The issue of heteroscedasticity is addressed through a variant on the weighted least squares model termed the variance function estimating (VFE) model (Davidian and Haaland 1990). The second issue is dealt with by implementing the latter model fitting technique to the family of four parameter logistic (FPL) and linear regression models. In this vignette we present a detailed overview of the calibFit package and its application to the latter type of problem.

Contents

1 Introduction

Robust and unbiased methods of curve fitting and calibration in the analysis of biological and chemical assays are of great importance for ensuring reliable and interpretable results. Two major issues which repeatedly arise in the analysis of such assays are heteroscedasticity in the response variable as a factor of the independent variable (dose, concentration, etc.) and failure to capture the entire range of values of the independent variable that represents the region of interest (for example, the failure to take measurements at higher levels of concentration resulting in the failure to capture the point of saturation). Improper handling of the heteroscasticity often observed in experimental data can lead to biased estimates of assay detection limits, and inflexible model choices can result in inappropriate fits leading to inaccurate inference.

The package we have developed, *calibFit*, performs univariate calibration and addresses the two aforementioned issues. A variant on the weighted least squares model termed the variance function estimating (VFE) model (Davidian and Haaland 1990) is implemented in order to model heteroscedasticity. This function is applied to several standard models (three and four parameter logistic models as well as linear and quadratic least squares models). The variety of models are flexible enough to allow analyses meant for data covering various portions of the response curve. The calibFit package is designed to be useful for both statisticians and experimentalists, with an option for user specification of the models or automated model selection. We present a detailed overview of the VFE model as wells as several demonstrations of application to experiment data.

A note to readers of this document. Section 2 presents a more detailed discussion of the model fitting in the *calibFit* package. Examples are worked in throughout. However, those interested in looking only at the examples can go directly to the Appendix in Section 3.

2 Model and Methodology

The *calibFit* package implements two methods for model fitting, the four parameter logistic (FPL) and linear regression models, equations (1) and (2) respectively:

$$f(x,\beta) = \frac{\beta_1 - \beta_2}{1 + (\frac{x}{\beta_3})^{\beta_4}} + \beta_2 + \epsilon \tag{1}$$

$$f(x,\beta) = \beta_0 + \beta_1 x + \epsilon \tag{2}$$

Extensions of these implemented in the package are the three parameter logistic (THPL) regression model, where β_4 in equation (1) is set to 1 and the quadratic linear regression model where a quadrattic term, $\beta_2 x^2$ is included in equation (2).

Once the parameters from the FPL or linear regression models have been estimated these are then used in the calibrating the independent as a function of the dependent variable. The calibration models for (1) and (2) are

$$x = \beta_3 (\frac{\beta_1 - y}{y - \beta_2})^{\frac{1}{\beta_4}} \tag{3}$$

and

$$x = \frac{y - \beta_0}{\beta 1} \tag{4}$$

We now discuss the general background and theory behind VFE models and provide examples in `R` illustrating the intuition of the methods and how they work.

In most assays with repeated measurements it is usually a safe assumption that the variability in the response at all levels of the independent variable (dose, concentration, etc.) is not constant

(heteroscadistic). Ignoring this can result in a poor modeling of the data and lead to inaccurate inference. A formal expression for a model which assumes constant variances can be written as

$$\begin{aligned} &Y_{ij} = f(x, \beta) + \sigma\epsilon_{ij} \\ &i = 1, \ldots, N, j = 1 \ldots, m_i \geq 1 \\ &n = \sum_{i=1}^{N} m_i \end{aligned} \tag{5}$$

where Y_{ij} is the response for the jth replicate at the ith setting of the $(k \times 1)$ vector of predictor variables $\{x_i\}$, $f(x_i, \beta)$ is the regression function and the β's are the coefficients associated with the independent variables x_i.The coefficients are estimated by minimizing (6), the ordinary least squares fit

$$\sum_{i=1}^{N} \sum_{j=1}^{m_i} (Y_{ij} - f(x_i, \beta))^2 \tag{6}$$

Standard assumptions are that the Y_{ij}'s are independent and identically distributed with independent random errors ϵ_{ij} having mean 0 and variance 1 such that $var(Y_{ij}) = \sigma^2$ for all i, j.

The object of interest in (6) is the regression function $f(x_i, \beta)$. Intuitively the regression function represents our underlying belief in how the data behaves. Typically we observe a sigmoidal (S shaped) curve with assay data. However, it is often the case that data may not have been collected at enough levels of the independent variable to capture this (as illustrated in Figure 1).

Consider the following examples (Figures 1). The first of the following two datasets relates the readings from a high performance liquid chromatography (HPLC) assay to the blood concentration (ng/ml) of a drug. The second dataset measures the presence of antibody in a sample at a given concentration of a particular drug using an Enzyme Linked Immunosorbent Assay (ELISA).

First load the *calibFit* library

```
> library(calibFit)
```

then the data

```
> data(HPLC)
> data(ELISA)
```

and assign variable names

```
> conc.hplc <- HPLC[,1]
> resp.hplc <- HPLC[,2]
> conc.elisa <- ELISA[,1]
> resp.elisa <- ELISA[,2]
```

The HPLC data is fit using an ordinary least squares regression model and the ELISA data is fit with a four parameter logistic (FPL) regression model. The data and model fits are shown in Figure (1).

A way to gain insight into what is happening with the variation in the data is to look at a plot of the predicted values against the (standardized) residuals. Ideally there should be no pattern in this plot. Any type of trend suggests an inconsistency in our model assumptions.

As stated earlier the assumption of constant variance in experimental sciences may often be incorrect. This is illustrated in Figures 1 and 2 where the variation in response increases with the concentration. It is sometimes the case that a log or square root transformation on the response variable can help control for non-constant variances. This may not always help however and it may alter the interpretability of the relationship between the dependent and independent variables (Figures 3 and 4).

The use of weighted least squares (WLS) is a standard approach to this type of problem. In WLS it is assumed that the experimental error of the response is proportional to some weights $\{w^{-1}\}$. An appropriate modification of (5) to more accurately reflect the character of the response is

$$Y_{ij} = f(x_i, \beta) + \sigma w_i^{-1/2} \epsilon_{ij}, i = 1, \ldots, N, j = 1, \ldots, m_i \tag{7}$$

and the coefficients β are estimated by minimizing (8)

$$\sum_{i=1}^{N} \sum_{j=1}^{m_i} w_i (Y_{ij} - f(x_i, \beta))^2 \tag{8}$$

where now $var(Y_{ij}) = \sigma^2 w_i^{-1}$.

2.1 Choosing a variance function

In order to provide a more general framework for finding an appropriate set of weights a variance function $g(\mu_i, z_i, \theta)$ is proposed such that

$$Y_{ij} = f(x_i, \beta) + \sigma g(\mu_i, z_i, \theta) \epsilon_{ij}, i = 1, \ldots, N, j = 1, \ldots, m_i \tag{9}$$

where μ_i is the mean response $f(x_i, \beta)$, $\{z_i\}$ is a sequence of known variables containing some or all of the values in x, and θ is a $(q \times 1)$ vector of parameters that may be known or unknown. The implied assumption becomes that $var(Y_{ij}) = \sigma^2 g^2(\mu_i, z_i, \theta)$ and the appropriate weights are $w_i = 1/g^2(\mu_i, z_i, \theta)$.

One common situation is that the experimental error is proportional to the mean response. This is also known as the constant coefficient of variation (CV) case. In this case the weight would be proportional to the mean response.

The variance function g is meant to reflect our assumption about the experimental error in the data. Thus, if the experimental error is taken to proportional to the mean response then weighting Y_{ij} by its mean response $\frac{1}{\mu_i}$ will yield a constant variance: $Var(\frac{1}{\mu_i} Y_{ij}) = (\frac{1}{\mu_i})^2 Var(Y_{ij}) = (\frac{1}{\mu_i})^2 \mu_i^2 \sigma^2 = \sigma^2$.

Another way to gain some insight into the experimental error is to first fit a model which generally describes the data and then plot the log of the the absolute values of the residuals $|r_{ij}|$ against the log of the fitted values. The reason for looking at a plot like this is that the residuals contain meaningful information about the variance and its relationship to the mean response. Putting everything on the log scale allows us to capture the general trend in the variance while controlling for for visual bias due to change of scale. From this visual representation some insight can be gained into what an appropriate variance function (VF) might be. If a generally linear relationship is apparent in this plot, as is the case in Figure 5, then a constant CV VF estimating model, discussed previously, would be appropriate (this turns out to be the case under most circumstances, for this reason *calibFit* implements the constant CV as the VFE).

A brief aside, if the underlying model is in fact (9) then it can be shown that (6) is no longer the best minimum variance unbiased estimator. Consider the following

$$\Sigma_{LS}^{\star} = \sigma^2 \Sigma_{LS}^{-1} Q \Sigma_{LS}^{-1}, Q = X^T G X \tag{10}$$

```
> ## Plot of the data with a std OLS fit
> par(mfrow=c(1,2))
> plot(conc.hplc,resp.hplc,
+         xlab = "Concentration (ng/ml)",
+         ylab = "Response",
+         main = "HPLC data")
> linmodel <- lm(resp.hplc~conc.hplc)
> # The predicted response
> linPredResp <- fitted(linmodel)
> # Linear regression fit
> lines(conc.hplc,linPredResp)
> ## Plot of the data with a std FPL fit
> plot(log(conc.elisa),resp.elisa,
+         xlab = "log(Concentration (ng/ml))",
+         ylab = "Response",
+         main = "ELISA data")
> fplmodel <- calib.fit(conc.elisa,resp.elisa,type="log.fpl")
> # The predicted response
> fplPredResp <- fplmodel@fitted.values
> # fpl regression fit
> lines(log(conc.elisa),fplPredResp)
```

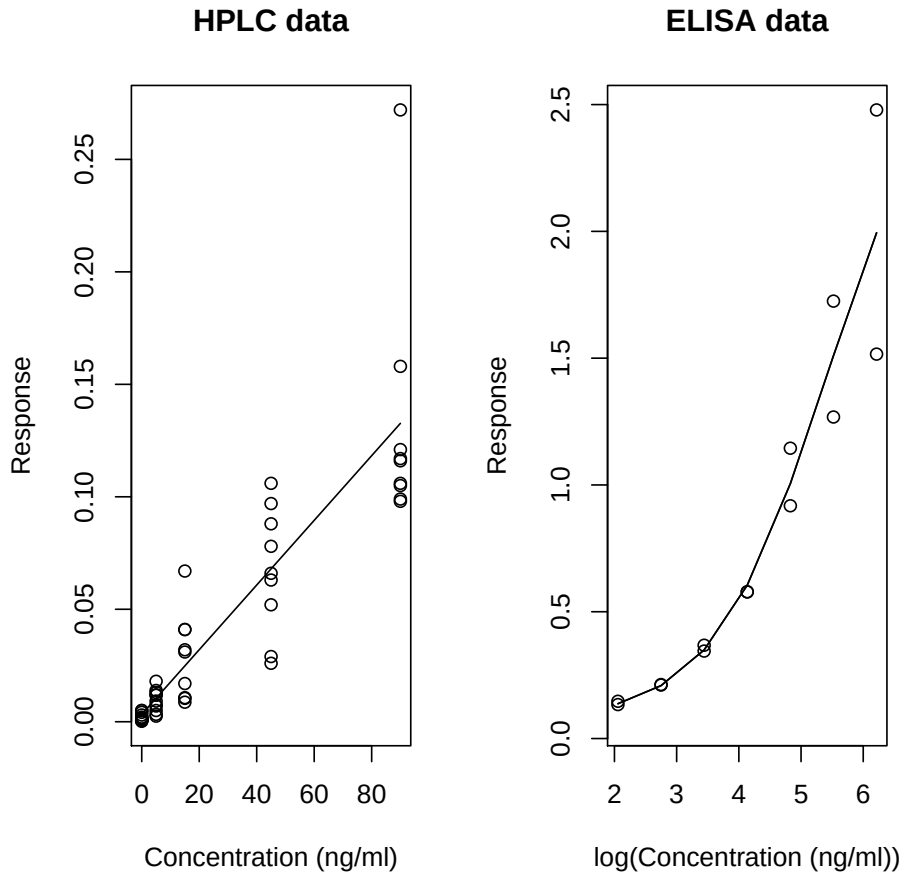


Figure 1: On the left is a plot of the HPLC data with standard least squares fit and on the right is a plot of the ELISA data with standard fpl regression fit. Note that in the ELISA data because the concentrations were serially diluted the log of the concentration is used

```
> par(mfrow=c(1,2))
> ## Residuals from linear fit
> linres <- residuals(linmodel)/summary(linmodel)[['sigma']]
> plot(linPredResp,linres,
+          xlab = "Predicted Value of Mean (LS)",
+          ylab = "Standardized Residuals",
+          main = "HPLC data",
+          ylim = c(-5,5))
> abline(h=0)
> ## Residuals from fpl fit
> fplres <- fplmodel@residuals/fplmodel@sigma
> plot(fplPredResp,fplres,
+         xlab = "Predicted Value of Mean (FPL)",
+         ylab = "Standardized Residuals",
+         main = "ELISA data",
+         ylim = c(-5,5))
> abline(h=0)
```

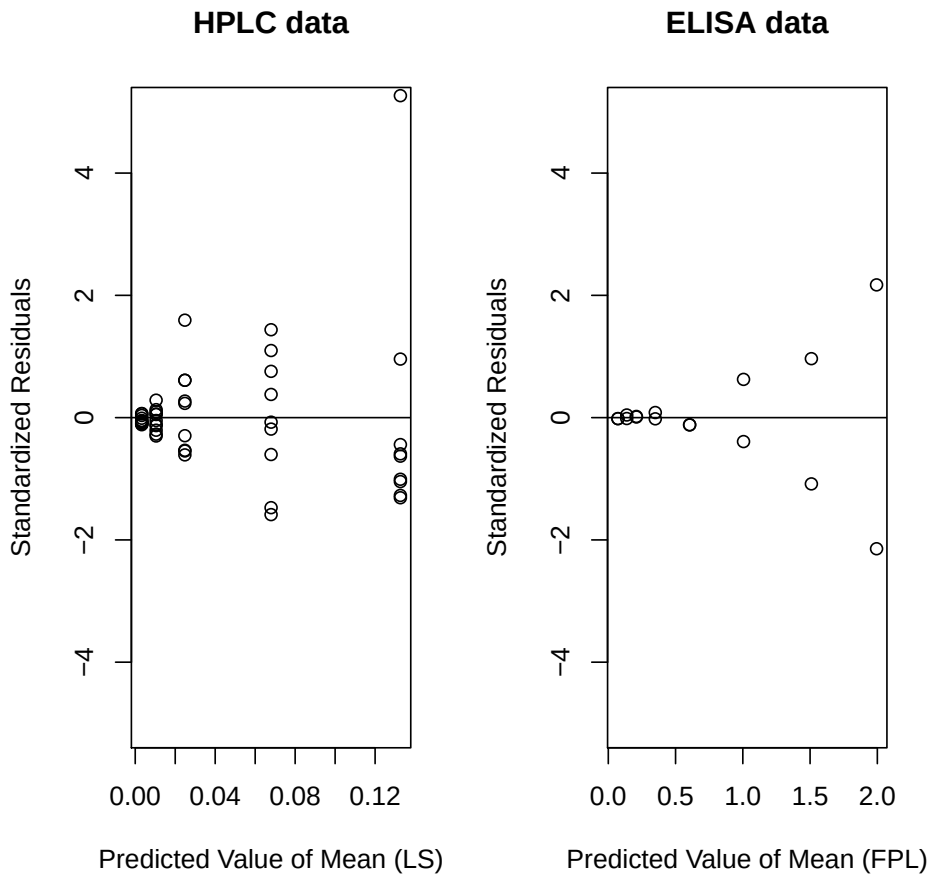


Figure 2: Plot of least squares residuals showing fan shape

```
> plot(conc.hplc,log(resp.hplc),
+          xlab = "Concentration (ng/ml)",
+          ylab = "log(Response)")
```

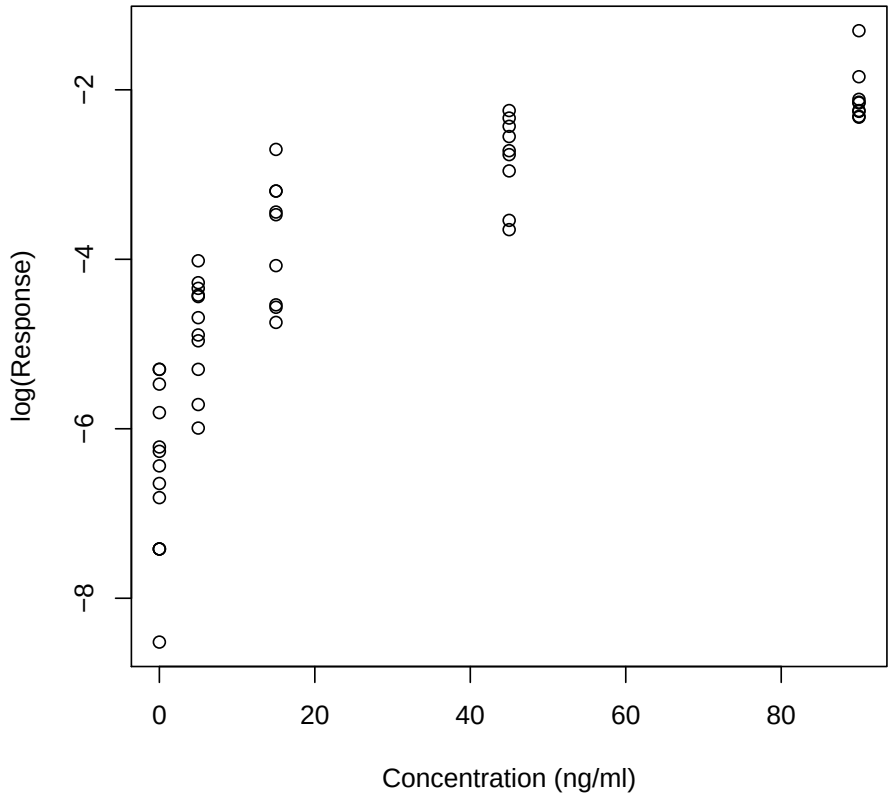


Figure 3: Plot of Concentration against log Response. This transformation still has not accounted for the non-constant variances and has also failed to preserve the linear relationship between the variables

```
> plot(conc.hplc,sqrt(resp.hplc),
+          xlab = "Concentration (ng/ml)",
+          ylab = "sqrt(Response)")
```

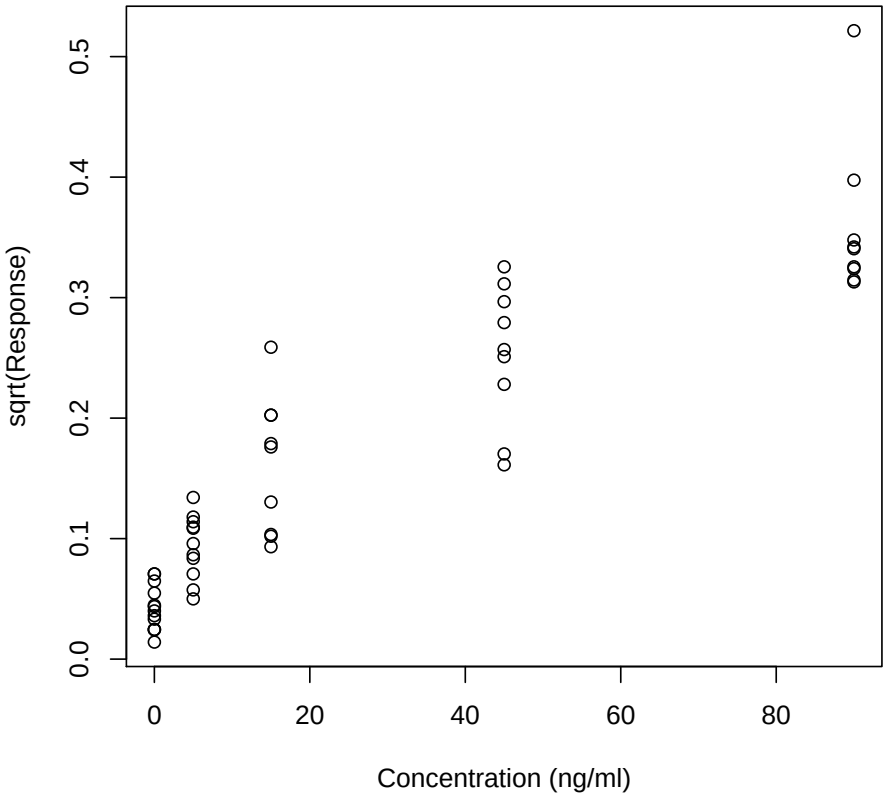


Figure 4: Plot of Concentration against the square root of the Response. The problems with this transformation are similar to those in Figure 3

```
> par(mfrow=c(1,2))
> plot(log(linPredResp),log(abs(linres)),
+         xlab = "Log(LS predicted values)",
+         ylab = "Log(absolute LS residuals)",
+         main = "HPLC data",
+         ylim = c(-6,2))
> linresmodel <- lm(log(abs(linres))~log(linPredResp))
> lines(log(linPredResp),fitted(linresmodel))
> plot(log(fplPredResp),log(abs(fplres)),
+         xlab = "Log(FPL predicted values)",
+         ylab = "Log(absolute FPL residuals)",
+         main = "ELISA data",
+         ylim = c(-6,2))
> fplresmodel <- lm(log(abs(fplres))~log(fplPredResp))
> lines(log(fplPredResp),fitted(fplresmodel))
```

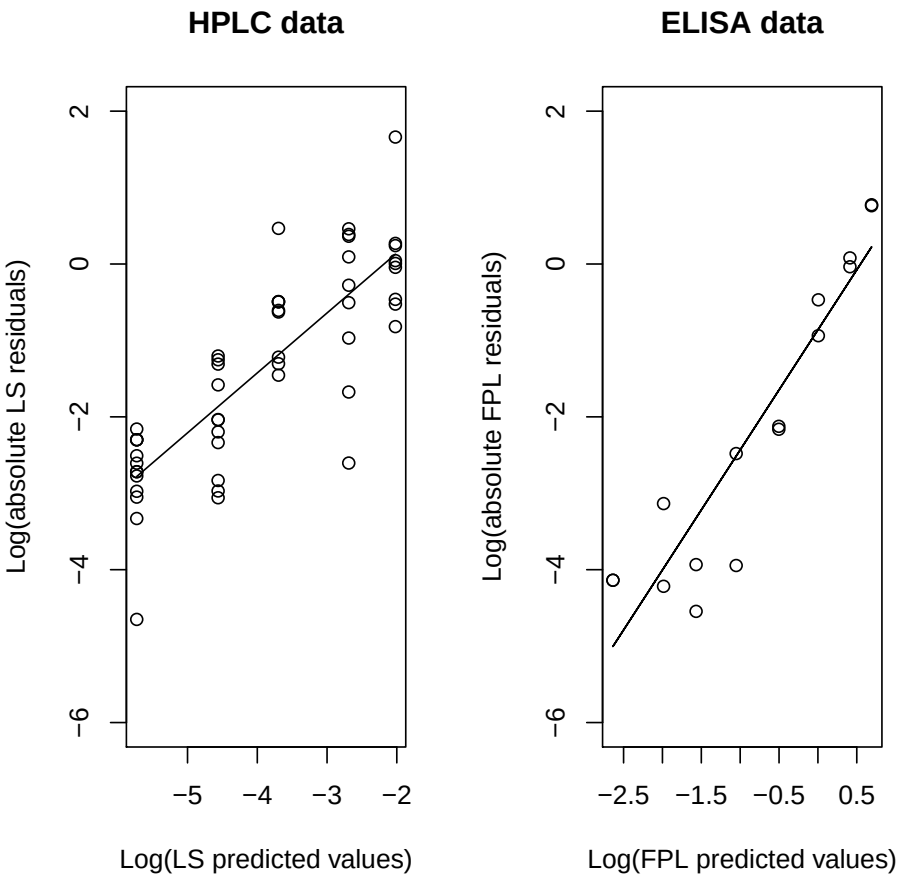


Figure 5: Plot of Log(absolute LS residuals) versus log(LS predicted values)

where Σ_{LS} is the variance covariance matrix of β_{LS} under standard assumptions. It can then be shown that $\Sigma^{\star}_{LS} - \Sigma_{GLS}$ is non-negative definite, where $\Sigma_{GLS} = \sigma^2(X^T G^{-1} X)^{-1}$ is the GLS estimate of the variance covariance matrix for β_{GLS} and G is a $(n \times n)$ matrix with the weights g along its diagonal.

This comes back to the issue of how reliable the results of the assay are. If the noise from experimentation is not taken into consideration then the above result implies that inference will be potentially unreliable.

2.2 Estimation of the Parameters

Next we discuss the estimation of σ and θ. Casting the problem in terms of the regression of the squared residuals against the variance function provides a nice framework for modeling the VFE model. For any estimate $\hat{\beta}$ the residuals are calculated as $r_{ij} = Y_{ij} - \hat{\mu}_i$, where $\hat{\mu}_i = f(x_i, \hat{\beta})$. Note that the VF as described in (9) implies $E(Y_{ij} - \mu_i)^2 = \sigma^2 g^2(\mu_i, z_i, \theta)$ and

$$E(r_{ij}^2) \approx \sigma^2 g^2(\hat{\mu}_i, z_i, \theta) \tag{11}$$

A brief aside. In standard regression analysis under the assumption of normality and constant variances equation (11) would be $E(r_{ij}^2) = (1 - h_{ii})\sigma^2$, where h_{ii} is the i^{th} element along the diagonal of the hat matrix $X(X^T X)^{-1} X^T$ and represents the amount of "leverage" a point has. The rule of thumb is that any point with an h_{ii} greater than $\frac{2p}{n}$, where p is the number of parameters in the model, is considered to be a point of high leverage and may be biasing the results of the regression analysis.

For the case where non-constant variances are assumed (Eq. (9)), the $1 - h_{ii}$'s are replaced by the values along the diagonal of the matrix

$$(I - X(X^T G X)^{-1} X^T G)^T (I - X(X^T G X)^{-1} X^T G) \tag{12}$$

In the paper by Davidian and Haaland and currently in the *calibFit* package this is not taken into consideration. For future work it may be worth exploring the effects of including these terms in the model.

Continuing, consider $\hat{\mu}_i$ to be known then (11) can be thought of as a regression problem where r_{ij} is the response and $\sigma^2 g^2(\hat{mu}_i, z_i, \theta)$ is the predictor. For normal data

$$var(r_{ij}^2) \approx (2\sigma^4) g^4(\hat{\mu}_i, z_i, \theta) \tag{13}$$

An appropriate model for this problem takes the form of a GLS regression where σ and θ are estimated by minizing

$$\sum_{i=1}^{N} \sum_{j=1}^{m_i} v_i (r_{ij}^2 - \sigma^2 g^2(\hat{\mu}_i, z_i, \theta))^2 \tag{14}$$

where $v_i = g^{-4}(\hat{\mu}_i, z_i, \theta)$. Thus the estimation of θ is based on the residuals from the previous fit in the iteration. Finally σ is estimated using the estimated value of θ.

One of the problems that may arise from using squared residuals as in (14) is that it increases the influence an outlying point has on the estimation of θ and σ. An alternative approach would be to use the absolute values of the residuals. It can be shown that $E|Y_{ij} - \mu_i| = \eta g(\mu_i, z_i, \theta)$ where $\eta = \sigma E(|\epsilon_{ij}|)$, $E(|r_{ij}|) \approx \eta g(\hat{\mu}_i, z_i, \theta)$ and $Var(|r_{ij}|) \propto g^2(\hat{\mu}_i, z_i, \theta)$. The function to be minimized is

$$\sum_{i=1}^{N} \sum_{j=1}^{m_i} u_i (|r_{ij}| - \eta g(\hat{\mu}_i, z_i, \theta))^2 \tag{15}$$

where $u_i = g^{-2}(\hat{\mu}_i, z_i, \theta)$.

Finally $\hat{\sigma}$ is estimated by

$$\hat{\sigma}^2 = (n-p)^{-1} \sum_{i=1}^{N} \sum_{j=1}^{m_i} (Y_{ij} - f(x_i, \hat{\beta}_{GLS}))^2 / g^2(f(x_i, \hat{\beta}_{GLS}), z_i, \hat{\theta}) \tag{16}$$

where the weight matrix G now has $g^2(f(x_i, \hat{\beta}_{GLS}), z_i, \hat{\theta})$ along its diagonal.

Applying the above methods for estimating the VF causes a considerable change in the distribution of the residuals. Specifically the systematic trend that was apparent before has been almost completely removed (Figure 6).

An even greater change can be seen in a plot of the confidence intervals (Figure 7 and 8). These plots can be generated as follows using `calib.fit`

```
> cal.fpl <- calib.fit(conc.elisa,resp.elisa,type="log.fpl")
> cal.lin.pom <- calib.fit(conc.hplc,resp.hplc,type="lin.pom")
> cal.fpl.pom <- calib.fit(conc.elisa,resp.elisa,type="log.fpl.pom")
> linpom.fit <- cal.lin.pom@fitted.values
> fplpom.fit <- cal.fpl.pom@fitted.values
> sig.lin <- cal.lin.pom@sigma
> sig.fpl <- cal.fpl.pom@sigma
> theta.lin <- cal.lin.pom@theta
> theta.fpl <- cal.fpl.pom@theta
> linpom.res <- cal.lin.pom@residuals*(1/((linpom.fit^theta.lin)*sig.lin))
> fplpom.res <- cal.fpl.pom@residuals*(1/((fplpom.fit^theta.fpl)*sig.fpl))
```

Next plot the standardized residuals
and look at a plot of the fitted model as well as the confidence intervals before and after the adjustment by POM.

2.3 Calibration

Calibration (or inverse regression) is the process of calculating the value of the independent variable for a given value of the dependent variable, in essence reversing the role of the two.

The inverse of Equations (1) and (2) give

$$x = \beta_3 \left(\frac{\beta_1 - y}{y - \beta_2}\right)^{\frac{1}{\beta_4}} \tag{17}$$

and

$$x = \frac{y - \beta_0}{\beta 1} \tag{18}$$

Calculation of the inverse in the THPL and quadratic regression cases is straightforward.

calibFit has two options for calculating the confidence intervals for the calibrated estimates. The first of the two uses the confidence intervals calculated from the fit of the y's as a function of the x's

$$\hat{y} \pm t_{1-\alpha/2,df} \hat{\sigma} \sqrt{f(x, \hat{\beta})^T \hat{\Sigma} f(x, \hat{\beta})} \tag{19}$$

where α is the Type I error, df is the degrees of freedom and $\hat{\Sigma}$ is the covariance matrix of the coefficients β. The problem then becomes finding the width of this confidence interval at the point associated with the x being estimated.

```
> par(mfrow=c(1,2))
> plot(linpom.fit,linpom.res,
+         xlab = "Fitted Values (LS)",
+         ylab = "Standardized Residuals",
+         main = "HPLC data")
> plot(fplpom.fit,fplpom.res,
+         xlab = "Fitted Values (FPL)",
+         ylab = "Standardized Residuals",
+         main = "ELISA data")
```

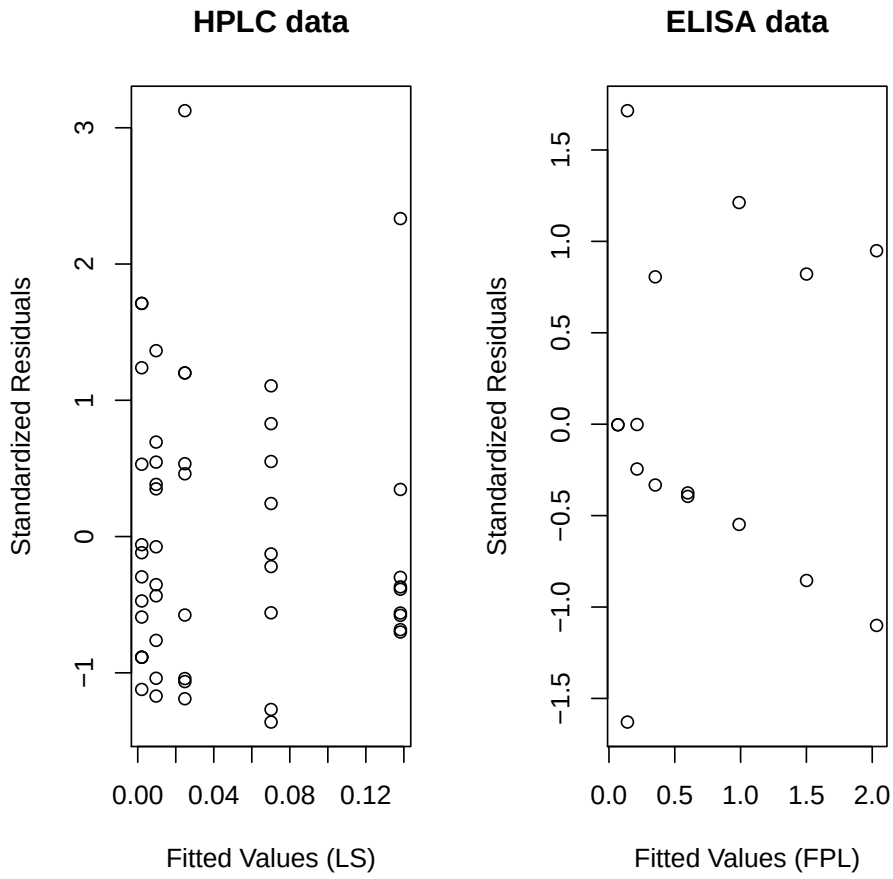


Figure 6: Plot of Log(absolute LS residuals) versus log(LS predicted values)

```
> par(mfrow=c(1,2))
> plot(HPLC, main = "HPLC data", sub="Without POM fit",col="blue",pch=16)
> lines(conc.hplc,fitted(linmodel),col="lightblue")
> lines(conc.hplc,ciu,col="lightblue",lty=2)
> lines(conc.hplc,cil,col="grey",lty=2)
> plot(cal.lin.pom,print=FALSE,main = "HPLC data", sub = "With POM fit",xlab = "Concentration", ylab =
```

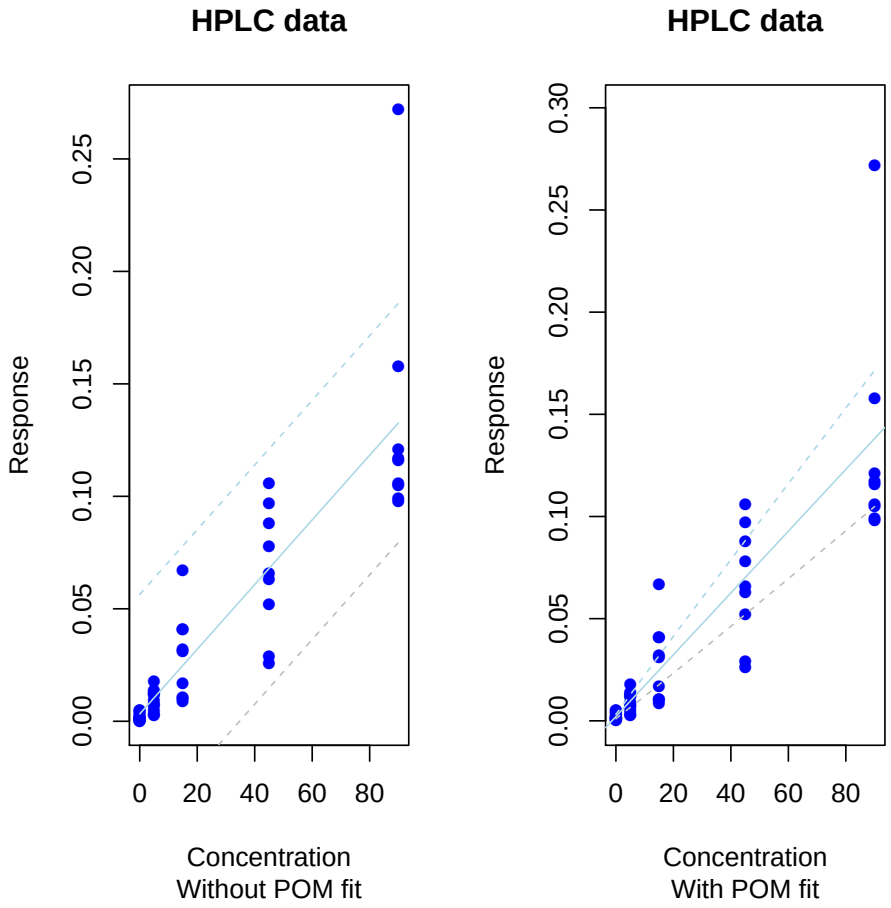


Figure 7: Plot of fitted models for HPLC data with and without POM

```
> par(mfrow=c(1,2))
> plot(cal.fpl,print=FALSE,main = "ELISA data", sub = "Without POM fit",xlab = "Concentration", ylab =
> plot(cal.fpl.pom,print=FALSE,main = "ELISA data", sub = "With POM fit",xlab = "Concentration", ylab
```

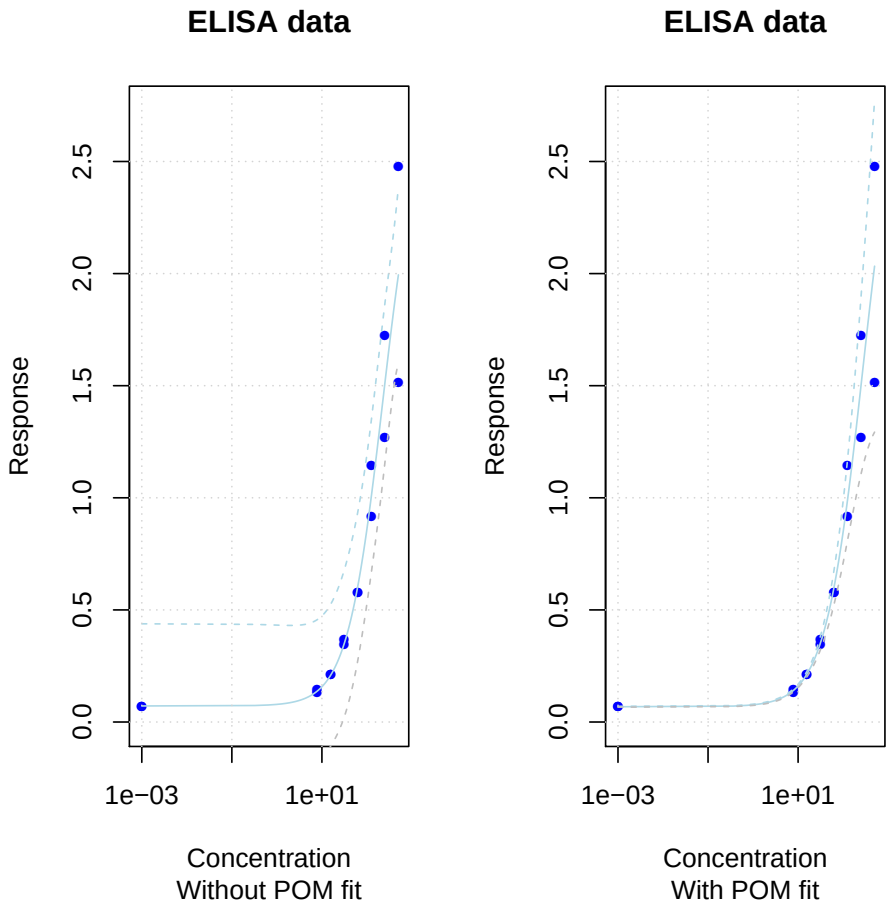


Figure 8: Plot of fitted models for HPLC data with and without POM

```
> plot(calib.lin,main="HPLC data")
```

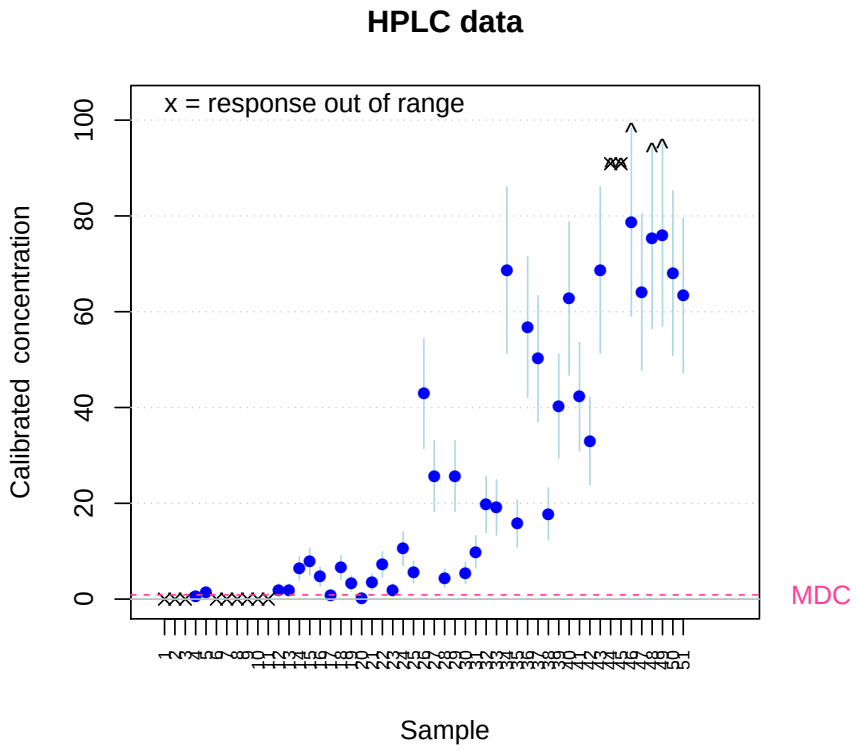


Figure 9: Plot of calibrated estimates for the HPLC data

The other approach is to calculate the Wald Intervals as shown in (20)

$$\begin{aligned} &\hat{x} \pm t_{1-\alpha/2,df} \sqrt{\hat{Var}(\hat{x})} \\ &\hat{Var}(\hat{x}) = \frac{\partial \hat{x}}{\partial \beta}^T Cov(\hat{Y}, \hat{\beta}) \frac{\partial \hat{x}}{\partial \beta} \end{aligned} \tag{20}$$

Continuing with the previous example

```
> calib.lin <- calib(cal.lin.pom,resp.hplc)
> calib.fpl <- calib(cal.fpl.pom,resp.elisa)
```

In addition we can plot the calibrated estimates along with their confidence intervals

2.4 Statistics

There are also several useful statistics calculated. The first of these is the minimum detectable concentration (MDC). MDC is lowest concentration where the curve is increasing (decreasing) which results

```
> plot(calib.fpl,main="ELISA data")
```

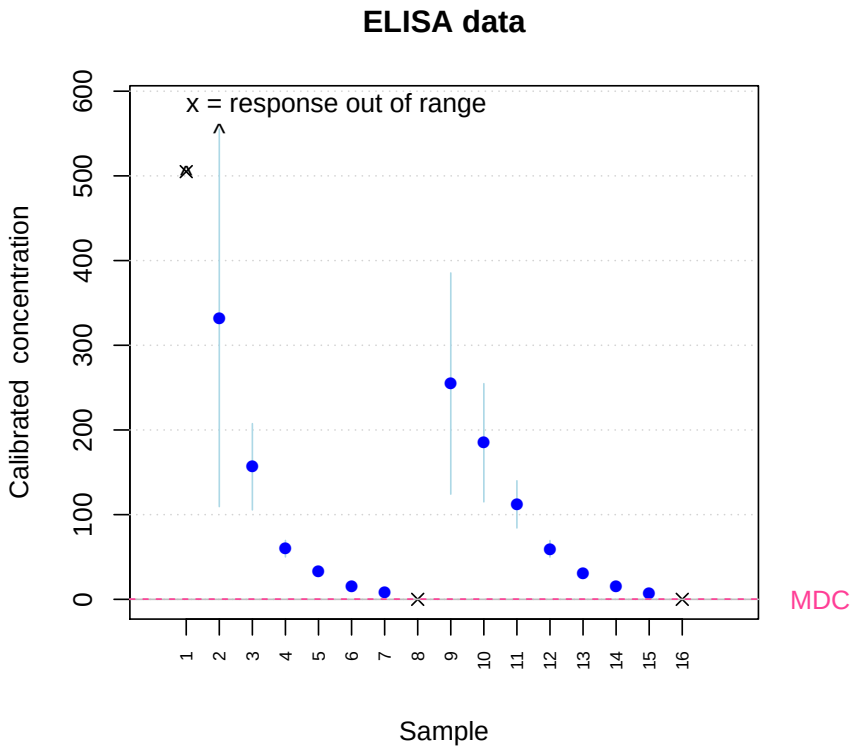


Figure 10: Plot of calibrated estimates for the ELISA data

in an expected response significantly greater (less) than the expected response at 0 concentration. For an increasing curve this is

$$x_{MDC} = min\{x : f(x, \hat{\beta}) \leq LCL_0\} \tag{21}$$

and for a decreasing curve this is

$$x_{MDC} = min\{x : f(x, \hat{\beta}) \geq UCL_0\} \tag{22}$$

Where LCL_0 and UCL_0 are respectively the lower and upper confidence limits at 0. The equations for these are shown in equation (19).

Next is the reliable detection limit (RDL). The RDL for an increasing (decreasing) curve, is the lowest concentration that has a high probability of producing a response that is significantly greater (less) than the response at 0.

$$x_{RDL} = min\{x : UCL_x \leq LCL_0\} \tag{23}$$

and for a decreasing curve this is

$$x_{RDL} = min\{x : LCL_x \geq UCL_0\} \tag{24}$$

Last is the limit of quantitization (LOQ). LOQ is the lowest concentration at which the coefficient of variation of dose is less than a fixed percent, the default is 20 in the *calibFit* package.

3 Appendix

First load the *calibFit* library

```
> library(calibFit)
```

then the data

```
> data(HPLC)
> data(ELISA)
```

and assign variable names

```
> conc.hplc <- HPLC[,1]
> resp.hplc <- HPLC[,2]
> conc.elisa <- ELISA[,1]
> resp.elisa <- ELISA[,2]
```

The HPLC data is fit using an ordinary least squares regression model and the ELISA data is fit with a four parameter logistic (FPL) regression model. The data and model fits are shown in Figure 1.

```
> par(mfrow=c(1,2))
> #par(mar = c(3.5,3.5,1.5,1.5))
> plot(conc.hplc,resp.hplc,
+         xlab = "Concentration (ng/ml)",
+         ylab = "Response",
+         main = "HPLC data")
> linmodel <- lm(resp.hplc~conc.hplc)
> # The predicted response
> linPredResp <- fitted(linmodel)
> # Linear regression fit
```

```
> lines(conc.hplc,linPredResp)
> ## Plot of the data with a std FPL fit
> #par(mar = c(3.5,3.5,1.5,1.5))
> plot(log(conc.elisa),resp.elisa,
+         xlab = "log(Concentration (ng/ml))",
+         ylab = "Response",
+         main = "ELISA data")
> fplmodel <- calib.fit(conc.elisa,resp.elisa,type="log.fpl")
> # The predicted response
> fplPredResp <- fplmodel@fitted.values
> # fpl regression fit
> lines(log(conc.elisa),fplPredResp)
```

On the left is a plot of the HPLC data with standard linear regression model fit with least squares and on the right is a plot of the ELISA data with standard FPL regression model also fit with least squares. Note that in the ELISA data because the concentrations were serially diluted the log of the concentration is used

A way to gain insight into what is happening with the variation in the data is to look at a plot of the predicted values against the (standardized) residuals. Ideally there should be no pattern in this plot. Any type of trend suggests an inconsistency in our model assumptions.

```
> par(mfrow=c(1,2))
> ## Residuals from linear fit
> linres <- residuals(linmodel)/summary(linmodel)[['sigma']]
> plot(linPredResp,linres,
+          xlab = "Predicted Value of Mean (LS)",
+          ylab = "Standardized Residuals",
+          main = "HPLC data",
+          ylim = c(-5,5))
> abline(h=0)
> ## Residuals from fpl fit
> fplres <- fplmodel@residuals/fplmodel@sigma
> plot(fplPredResp,fplres,
+         xlab = "Predicted Value of Mean (FPL)",
+         ylab = "Standardized Residuals",
+         main = "ELISA data",
+         ylim = c(-5,5))
> abline(h=0)
```

This is a plot of least squares residuals showing fan shape

As stated earlier the assumption of constant variance in experimental sciences may often be incorrect. This is illustrated in Figures 1 and 2 where the variation in response increases with the concentration. It is sometimes the case that a log or square root transformation on the response variable can help control for non-constant variances. This may not always help however and it may alter the interpretability of the relationship between the dependent and independent variables (Figures 3 and 4).

```
> plot(conc.hplc,log(resp.hplc),
+          xlab = "Concentration (ng/ml)",
+          ylab = "log(Response)")
```

This is a plot of Concentration against log Response. This transformation still has not accounted for the non-constant variances and has also failed to preserve the linear relationship between the variables.

```
> plot(conc.hplc,sqrt(resp.hplc),
+          xlab = "Concentration (ng/ml)",
+          ylab = "sqrt(Response)")
```

This is a plot of Concentration against the square root of the Response. The problems with this transformation are similar to those in Figure 3.

If a generally linear relationship is apparent in this plot, as is the case in Figure 5, then a constant CV VFE model, discussed previously, would be appropriate (this turns out to be the case under most circumstances, for this reason *calib* implements the constant CV as the VFE).

```
> par(mfrow=c(1,2))
> plot(log(linPredResp),log(abs(linres)),
+         xlab = "Log(LS predicted values)",
+         ylab = "Log(absolute LS residuals)",
+         main = "HPLC data",
+         ylim = c(-6,2))
> linresmodel <- lm(log(abs(linres))~log(linPredResp))
> lines(log(linPredResp),fitted(linresmodel))
> plot(log(fplPredResp),log(abs(fplres)),
+         xlab = "Log(FPL predicted values)",
+         ylab = "Log(absolute FPL residuals)",
+         main = "ELISA data",
+         ylim = c(-6,2))
> fplresmodel <- lm(log(abs(fplres))~log(fplPredResp))
> lines(log(fplPredResp),fitted(fplresmodel))
```

A plot of Log(absolute LS residuals) versus log(LS predicted values)

An even greater change can be seen in a plot of the confidence intervals (Figure 7 and 8).

These plots can be generated as follows using `calib.fit`

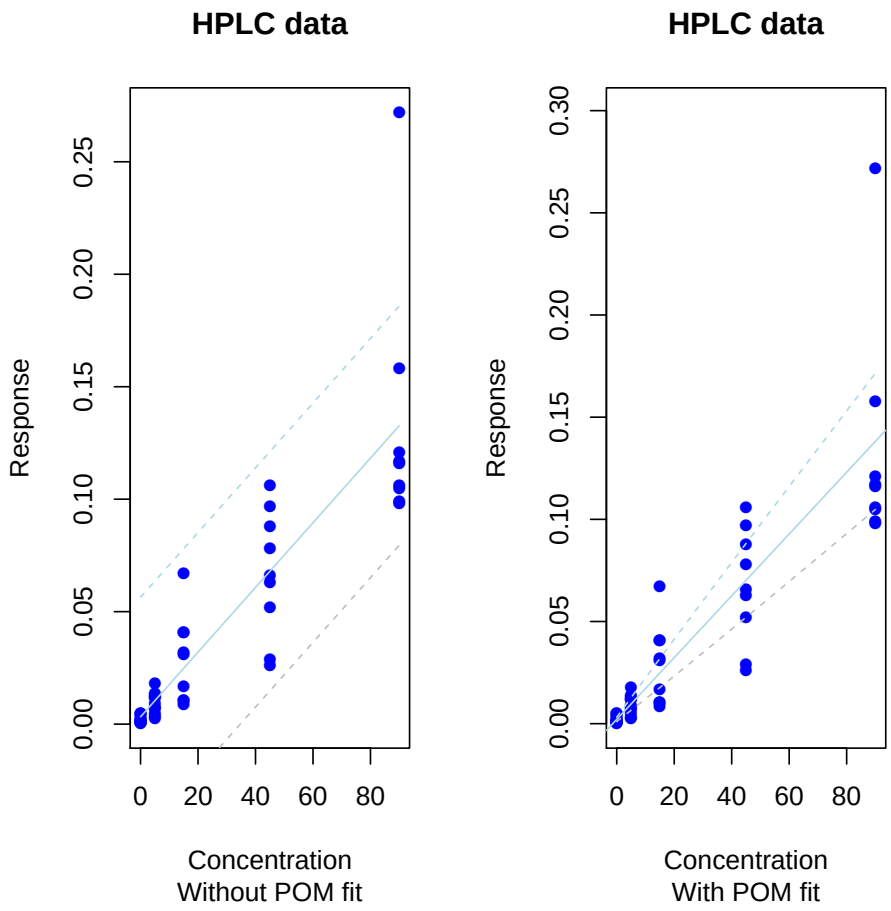
```
> cal.fpl <- calib.fit(conc.elisa,resp.elisa,type="log.fpl")
> cal.lin.pom <- calib.fit(conc.hplc,resp.hplc,type="lin.pom")
> cal.fpl.pom <- calib.fit(conc.elisa,resp.elisa,type="log.fpl.pom")
> linpom.fit <- cal.lin.pom@fitted.values
> fplpom.fit <- cal.fpl.pom@fitted.values
> sig.lin <- cal.lin.pom@sigma
> sig.fpl <- cal.fpl.pom@sigma
> theta.lin <- cal.lin.pom@theta
> theta.fpl <- cal.fpl.pom@theta
> linpom.res <- cal.lin.pom@residuals*(1/((linpom.fit^theta.lin)*sig.lin))
> fplpom.res <- cal.fpl.pom@residuals*(1/((fplpom.fit^theta.fpl)*sig.fpl))
```

Next plot the standardized residuals

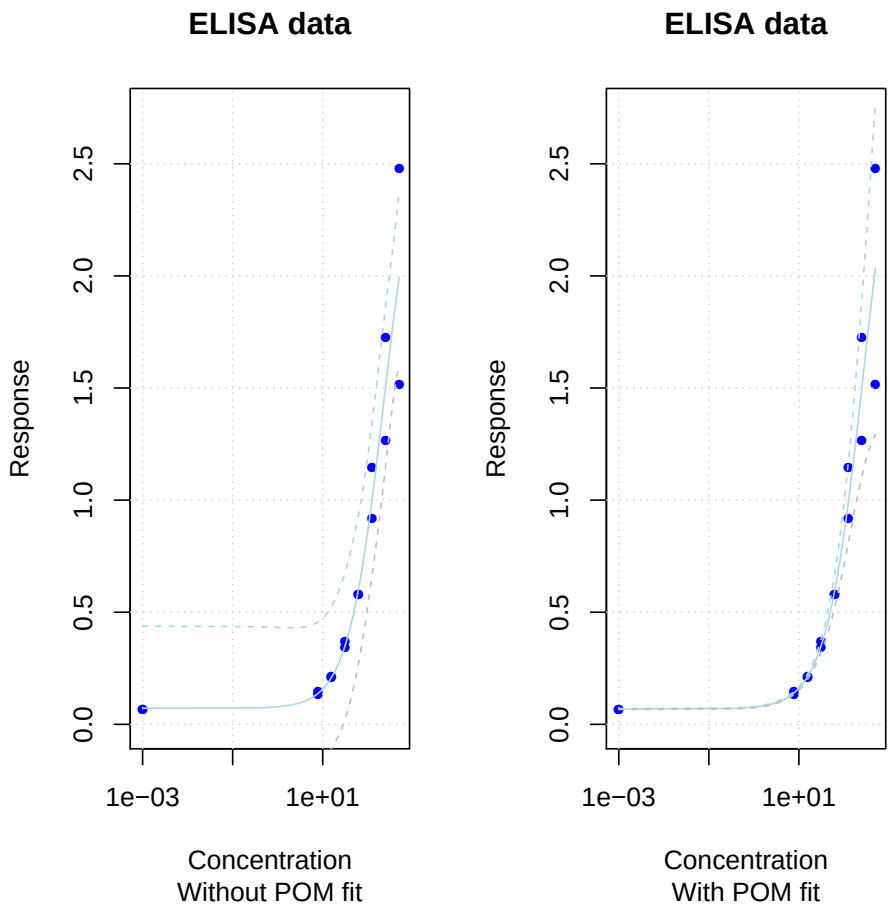
```
> par(mfrow=c(1,2))
> #par(mar = c(3.5,3.5,1.5,1.5))
> plot(linpom.fit,linpom.res,
+         xlab = "Fitted Values (LS)",
+         ylab = "Standardized Residuals",
+         main = "HPLC data")
> #par(mar = c(3.5,3.5,1.5,1.5))
> plot(fplpom.fit,fplpom.res,
```

```
+          xlab = "Fitted Values (FPL)",
+          ylab = "Standardized Residuals",
+          main = "ELISA data")
```

This is a plot of Log(absolute LS residuals) versus log(LS predicted values)
and look at a plot of the fitted model as well as the confidence intervals before and after the adjustment by POM.



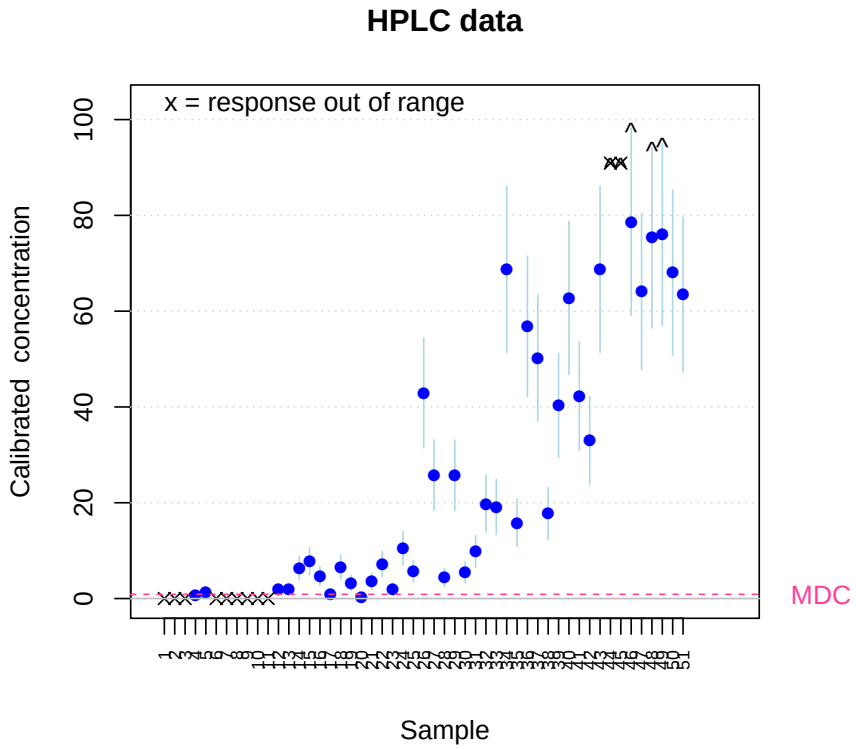
This is a plot of fitted models for HPLC data with and without POM



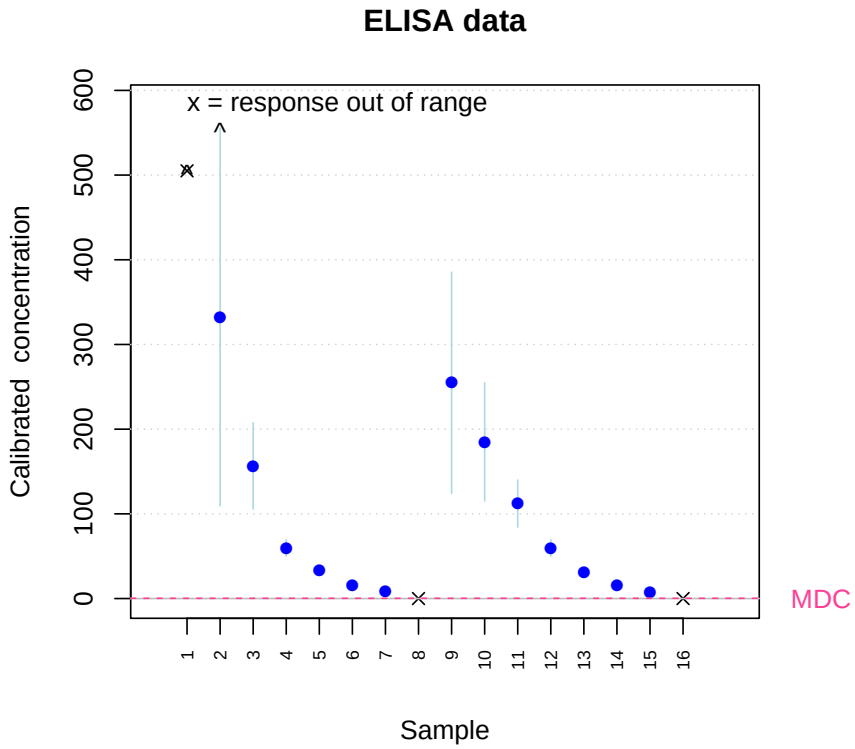
This is a plot of fitted models for HPLC data with and without POM
Continuing with the previous example

```
> calib.lin <- calib(cal.lin.pom,resp.hplc)
> calib.fpl <- calib(cal.fpl.pom,resp.elisa)
```

In addition we can plot the calibrated estimates along with their confidence intervals



This is a plot of calibrated estimates for the HPLC data.



This is a plot of calibrated estimates for the ELISA data.