Introduction to Nonlinear Regression

Andreas Ruckstuhl*
IDP Institute of Data Analysis and Process Design
ZHAW Zurich University of Applied Sciences in Winterthur

2024†

Contents

*E-Mail Address: Andreas.Ruckstuhl@zhaw.ch; Internet: http://www.idp.zhaw.ch
†The author thanks Werner Stahel for his valuable comments and Amanda Strong and Lukas Meier for their help in translating the original German version into English.

Goals

The *nonlinear regression model* block in the Weiterbildungslehrgang (WBL) in angewandter Statistik at the ETH Zurich should

1. introduce problems that are relevant to the fitting of nonlinear regression functions,
2. present graphical representations for assessing the quality of approximate confidence intervals, and
3. introduce some parts of the statistics software R that can help with solving concrete problems.

1 Estimation and Standard Inference

1.1 The Nonlinear Regression Model

a **The Regression Model.** Regression studies the relationship between a **variable of interest** Y and one or more **explanatory or predictor variables** $x^{(j)}$. The general model is

$$Y_i = h\langle x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(m)}; \theta_1, \theta_2, \ldots, \theta_p\rangle + E_i.$$

Here, h is an appropriate function that depends on the predictor variables and parameters, that we want to combine into vectors $\underline{x} = [x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(m)}]^T$ and $\underline{\theta} = [\theta_1, \theta_2, \ldots, \theta_p]^T$. We assume that the errors are all normally distributed and independent, i.e.

$$E_i \sim \mathcal{N}\left\langle 0, \sigma^2 \right\rangle, \text{ independent.}$$

b **The Linear Regression Model.** In (multiple) linear regression, we considered functions h that are linear in the parameters θ_j,

$$h\langle x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(m)}; \theta_1, \theta_2, \ldots, \theta_p\rangle = \theta_1 \widetilde{x}_i^{(1)} + \theta_2 \widetilde{x}_i^{(2)} + \ldots + \theta_p \widetilde{x}_i^{(p)},$$

where the $\widetilde{x}^{(j)}$ can be arbitrary functions of the original explanatory variables $x^{(j)}$. There the parameters were usually denoted by β_j instead of θ_j.

c **The Nonlinear Regression Model.** In nonlinear regression, we use functions h that are *not* linear in the parameters. Often, such a function is derived from theory. In principle, there are unlimited possibilities for describing the deterministic part of the model. As we will see, this flexibility often means a greater effort to make statistical statements.

Example **d** **Puromycin** The speed of an enzymatic reaction depends on the concentration of a substrate. As outlined in Bates and Watts (1988), an experiment was performed to examine how a treatment of the enzyme with an additional substance called Puromycin influences the reaction speed. The initial speed of the reaction is chosen as the response variable, which is measured via radioactivity (the unit of the response variable is count/min^2; the number of registrations on a Geiger counter per time period measures the quantity of the substance, and the reaction speed is proportional to the change per time unit).

The relationship of the variable of interest with the substrate concentration x (in ppm) is described by the Michaelis-Menten function

$$h\langle x; \underline{\theta}\rangle = \frac{\theta_1 x}{\theta_2 + x}\ .$$

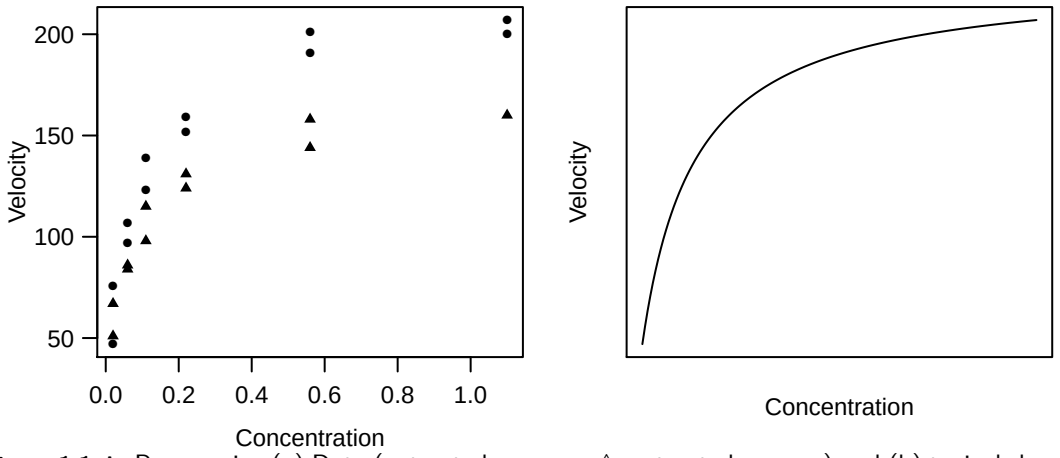

Figure 1.1.d.: Puromycin. (a) Data (• treated enzyme; △ untreated enzyme) and (b) typical shape of the regression function.

An infinitely large substrate concentration ($x \to \infty$) leads to the "asymptotic" speed θ_1. It was hypothesized that this parameter is influenced by the addition of Puromycin. The experiment is therefore carried out once with the enzyme treated with Puromycin and once with the untreated enzyme. Figure 1.1.d shows the result. In this section the data of the treated enzyme is used.

Example e **Biochemical Oxygen Demand** To determine the biochemical oxygen demand, stream water samples were enriched with soluble organic matter, with inorganic nutrients and with dissolved oxygen, and subdivided into bottles (Marske, 1967, see Bates and Watts, 1988). Each bottle was inoculated with a mixed culture of microorganisms, sealed and put in a climate chamber with constant temperature. The bottles were periodically opened and their dissolved oxygen concentration was analyzed, from which the biochemical oxygen demand [mg/l] was calculated. The model used to connect the cumulative biochemical oxygen demand Y with the incubation time x, is based on exponential decay:

$$h\left\langle x; \underline{\theta} \right\rangle = \theta_1 \left(1 - e^{-\theta_2 x}\right).$$

Figure 1.1.e shows the data and the regression function.

Example f **Membrane Separation Technology** (Rapold-Nydegger, 1994). The ratio of protonated to deprotonated carboxyl groups in the pores of cellulose membranes depends on the pH-value x of the outer solution. The protonation of the carboxyl carbon atoms can be detected by ^{13}C-NMR. We assume that the relationship can be described by the extended *"Henderson-Hasselbalch Equation"* for polyelectrolytes

$$\log_{10}\left\langle \frac{\theta_1 - y}{y - \theta_2} \right\rangle = \theta_3 + \theta_4 x \,,$$

where the unknown parameters are θ_1, θ_2 and $\theta_3 > 0$ and $\theta_4 < 0$. Solving for y leads to the model

$$Y_i = h\langle x_i; \underline{\theta}\rangle + E_i = \frac{\theta_1 + \theta_2\, 10^{\theta_3 + \theta_4 x_i}}{1 + 10^{\theta_3 + \theta_4 x_i}} + E_i \,.$$

The regression function $h\langle x_i, \underline{\theta}\rangle$ for a reasonably chosen $\underline{\theta}$ is shown in Figure 1.1.f next to the data.

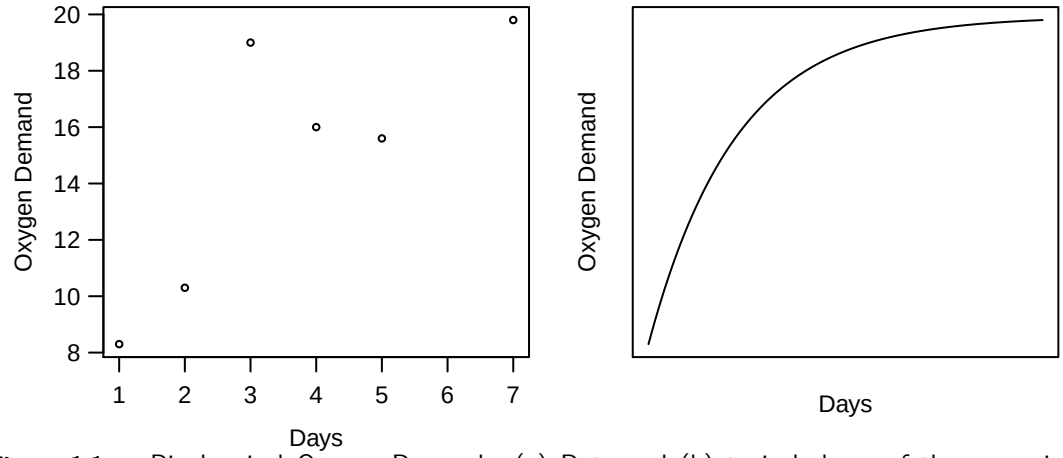

Figure 1.1.e.: Biochemical Oxygen Demand. (a) Data and (b) typical shape of the regression function.

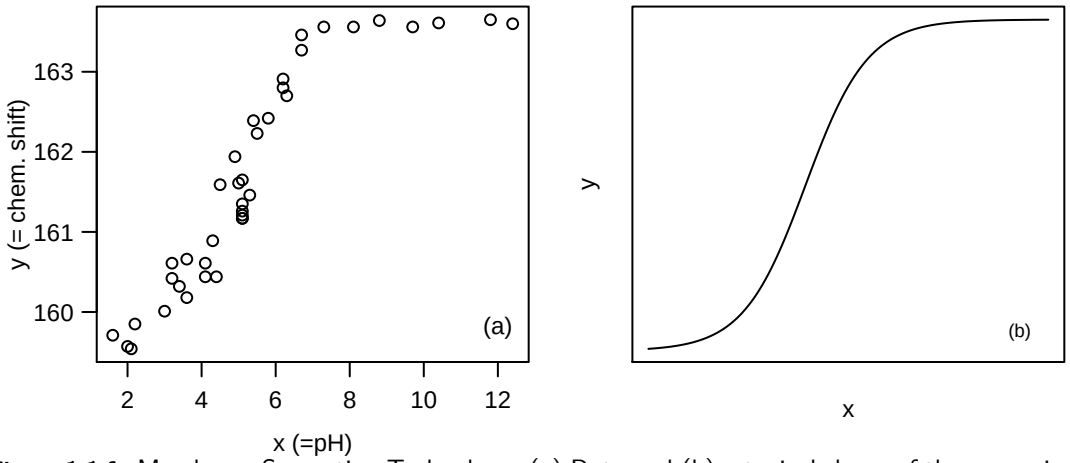

Figure 1.1.f.: Membrane Separation Technology. (a) Data and (b) a typical shape of the regression function.

g Some Further Examples of Nonlinear Regression Functions.

- Hill model (enzyme kinetics): $h\langle x_i, \underline{\theta}\rangle = \theta_1 x_i^{\theta_3}/(\theta_2 + x_i^{\theta_3})$
  For $\theta_3 = 1$ this is also known as the Michaelis-Menten model (1.1.d).
- Mitscherlich function (growth analysis): $h\langle x_i, \underline{\theta}\rangle = \theta_1 + \theta_2 \exp\langle\theta_3 x_i\rangle$.
- From kinetics (chemistry) we get the function

$$h\langle x_i^{(1)}, x_i^{(2)}; \underline{\theta}\rangle = \exp\langle -\theta_1 x_i^{(1)} \exp\langle -\theta_2/x_i^{(2)}\rangle\rangle.$$

- Cobbs-Douglas production function

$$h\left\langle x_i^{(1)}, x_i^{(2)}; \underline{\theta}\right\rangle = \theta_1 \left(x_i^{(1)}\right)^{\theta_2} \left(x_i^{(2)}\right)^{\theta_3}.$$

Since useful regression functions are often derived from the theoretical background of the application of interest, a general overview of nonlinear regression functions is of

very limited benefit. A compilation of functions from publications can be found in Appendix 7 of Bates and Watts (1988).

h **Transformably Linear Regression Functions.** Some nonlinear regression functions have a very favourable structure. For example, a power function

$$h\langle x;\underline{\theta}\rangle = \theta_1 x^{\theta_2}$$

can be **transformed to a linear** model by expressing the logarithm of $h\langle x;\underline{\theta}\rangle$ as a linear (in the parameters) function of the logarithm of the explanatory variable x

$$\ln\langle h\langle x;\underline{\theta}\rangle\rangle = \ln\langle\theta_1\rangle + \theta_2 \ln\langle x\rangle = \beta_0 + \beta_1\widetilde{x}\ ,$$

where $\beta_0 = \ln\langle\theta_1\rangle$, $\beta_1 = \theta_2$ and $\widetilde{x} = \ln\langle x\rangle$. We call such a regression function h **transformably linear**.

i **The Statistically Complete Model.** The "regression fitting" of the "linearized" regression function given in the previous paragraph is based on the model

$$\ln\langle Y_i\rangle = \beta_0 + \beta_1\widetilde{x}_i + E_i\ ,$$

where the random errors E_i all have the same normal distribution. We transform this model back and get

$$Y_i = \theta_1 \cdot x^{\theta_2} \cdot \widetilde{E}_i,$$

with $\widetilde{E}_i = \exp\langle E_i\rangle$. The errors $\widetilde{E}_i$, $i = 1,\ldots,n$, now have a multiplicative effect and are log-normally distributed! The assumption about the random errors is now clearly different than for a regression model based directly on h,

$$Y_i = \theta_1 \cdot x^{\theta_2} + E_i^*,$$

where the random error E_i^* contributes additively and is normally distributed.

A linearization of the regression function is therefore advisable only if the assumptions about the random errors can be better satisfied – in our example, if the errors actually act multiplicatively rather than additively and are log-normally rather than normally distributed. These assumptions must be checked with residual analysis (see, e.g., Figure 1.1.i).

* It should be noted that it is not sufficient to examine the fit in a scatter plot of response versus exploratory variable, as the non-linearity of the fitted curve prevents an assessment of the specified error structure. Consequently, the evaluation must be carried out in the Tukey-Anscombe diagram.

j * Note: In linear regression it has been shown that the variance can be stabilized with certain transformations of the response variable (e.g. $\log\langle\cdot\rangle, \sqrt{\cdot}$). If this is not possible, in certain circumstances one can also perform a weighted linear regression. The process is analogous in nonlinear regression.

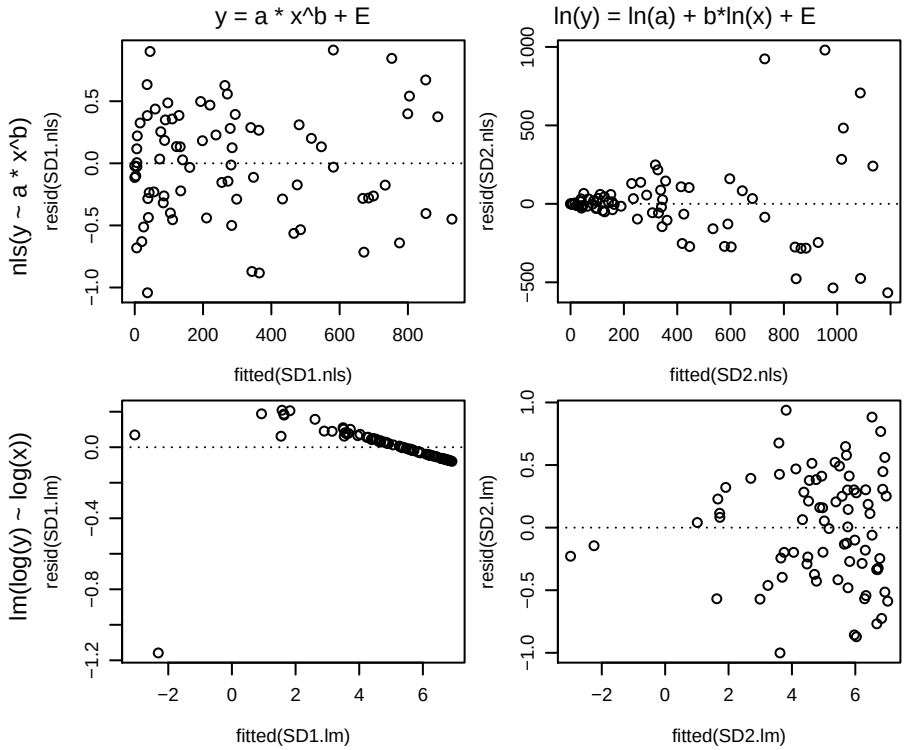

Figure 1.1.i.: Tukey-Anscombe plots of four different situations. In the left column, the data are simulated from an additive error model, whereas in the right column the data are simulated from a multiplicative error model. The top row shows the Tukey-Anscombe plots of fits of a multiplicative error model to both data sets and the bottom row shows the Tukey-Anscombe plots of fits of an additive error model to both data sets.

k Here are some more linearizable functions (see also Daniel and Wood, 1980):

$$
\begin{aligned}
h\langle x;\underline{\theta}\rangle &= 1/(\theta_1+\theta_2\exp\langle -x\rangle) && \longleftrightarrow && 1/h\langle x;\underline{\theta}\rangle &= \theta_1+\theta_2\exp\langle -x\rangle \\
h\langle x;\underline{\theta}\rangle &= \theta_1 x/(\theta_2+x) && \longleftrightarrow && 1/h\langle x;\underline{\theta}\rangle &= 1/\theta_1+\theta_2/\theta_1\tfrac{1}{x} \\
h\langle x;\underline{\theta}\rangle &= \theta_1 x^{\theta_2} && \longleftrightarrow && \ln\langle h\langle x;\underline{\theta}\rangle\rangle &= \ln\langle\theta_1\rangle+\theta_2\ln\langle x\rangle \\
h\langle x;\underline{\theta}\rangle &= \theta_1\exp\langle\theta_2 g\langle x\rangle\rangle && \longleftrightarrow && \ln\langle h\langle x;\underline{\theta}\rangle\rangle &= \ln\langle\theta_1\rangle+\theta_2 g\langle x\rangle \\
h\langle x;\underline{\theta}\rangle &= \exp\langle -\theta_1 x^{(1)}\exp\langle -\theta_2/x^{(2)}\rangle\rangle && \longleftrightarrow && \ln\langle\ln\langle h\langle x;\underline{\theta}\rangle\rangle\rangle &= \ln\langle -\theta_1\rangle+\ln\langle x^{(1)}\rangle-\theta_2/x^{(2)} \\
h\langle x;\underline{\theta}\rangle &= \theta_1\ \left(x^{(1)}\right)^{\theta_2}\ \left(x^{(2)}\right)^{\theta_3} && \longleftrightarrow && \ln\langle h\langle x;\underline{\theta}\rangle\rangle &= \ln\langle\theta_1\rangle+\theta_2\ \ln\langle x^{(1)}\rangle+\theta_3\ \ln\langle x^{(2)}\rangle\ .
\end{aligned}
$$

The last one is the Cobbs-Douglas Model from 1.1.g.

l We have almost exclusively seen regression functions that only depend on one predictor variable x. This was primarily because it was possible to graphically illustrate the model. The following theory also works well for regression functions $h\langle\underline{x};\underline{\theta}\rangle$ that depend on several predictor variables $\underline{x} = [x^{(1)}, x^{(2)}, \ldots, x^{(m)}]$.

1.2 Model Fitting Using an Iterative Algorithm

a **The Principle of Least Squares.** To get estimates for the parameters $\underline{\theta} = [\theta_1, \theta_2, \ldots, \theta_p]^T$, one applies – like in linear regression – the principle of least squares. The sum of the squared deviations

$$S(\underline{\theta}) := \sum_{i=1}^{n} (y_i - \eta_i\langle\underline{\theta}\rangle)^2 \qquad \text{where} \quad \eta_i\langle\underline{\theta}\rangle := h\langle x_i; \underline{\theta}\rangle$$

should be minimized. The notation that replaces $h\langle x_i; \underline{\theta}\rangle$ with $\eta_i\langle\underline{\theta}\rangle$ is reasonable because $[x_i, y_i]$ is given by the data and only the parameters $\underline{\theta}$ remain to be determined.

Unfortunately, the minimum of $S(\underline{\theta})$ and thus the estimater have no explicit solution as it was the case for linear regression. **Iterative numeric procedures** are therefore needed. We will sketch the basic ideas of the most common algorithm. It is also the basis for the easiest way to derive tests and confidence intervals.

b **Geometrical Illustration.** The observed values $\underline{Y} = [Y_1, Y_2, \ldots, Y_n]^T$ define a point in n-dimensional space. The same holds true for the "model values" $\underline{\eta}\langle\underline{\theta}\rangle = [\eta_1\langle\underline{\theta}\rangle, \eta_2\langle\underline{\theta}\rangle, \ldots, \eta_n\langle\underline{\theta}\rangle]^T$ for a given $\underline{\theta}$.

Please take note: In multivariate statistics where an observation consists of m variables $x^{(j)}$, $j = 1, 2, \ldots, m$, it's common to illustrate the observations in the m-dimensional space. Here, we consider the Y- and η-values of all n observations as points in the n-dimensional space.

Unfortunately, geometrical interpretation stops with three dimensions (and thus with three observations). Nevertheless, let us have a look at such a situation, first for simple linear regression.

c Let's start with three observed response values 8.3, 16, 19.9 (i.e., $\underline{Y} = [8.3, 16, 19.9]^T$) at the three x values $\underline{x} = [1, 4, 7]^T$. The two vectors each define a point in three-dimensional space. Let us first consider the fitting of a straight line, i.e. a simple linear regression. For given parameters $\beta_0 = 5$ and $\beta_1 = 1$ we can calculate the model values $\eta_i\langle\underline{\beta}\rangle = \beta_0 + \beta_1 x_i$ and represent the corresponding vector $\underline{\eta}\langle\underline{\beta}\rangle = \beta_0\underline{1} + \beta_1\underline{x}$ as a point. We now ask: Where are all the points that can be achieved by varying the parameters $\underline{\beta}$? These are the possible linear combinations of the two vectors $\underline{1}$ and $\underline{x}$ and thus form the plane "spanned by $\underline{1}$ and $\underline{x}$" (see Figure 1.2.c). By estimating the parameters according to the principle of least squares, the squared distance between $\underline{Y}$ and $\underline{\eta}\langle\underline{\beta}\rangle$ is minimized. So, we want the point on the plane that has the least distance to $\underline{Y}$. This is also called the **projection** of $\underline{Y}$ onto the plane and is called $\underline{\widehat{y}}$. The parameter values that correspond to this point $\underline{\widehat{\eta}}$ are therefore the estimated parameter values $\underline{\widehat{\beta}} = [\widehat{\beta}_0, \widehat{\beta}_1]^T$. An illustration can be found in Figure 1.2.c.

d Now we want to fit a nonlinear function, e.g. $h\langle\underline{x}; \underline{\theta}\rangle = \theta_1 \exp\langle 1 - \theta_2 x\rangle$ from the the example "Biochemical Oxygen Demand", to the same three observations. For $\theta_1 = 16$ and $\theta_2 = 0.4$, we obtain $\underline{\eta}\langle\underline{\theta}\rangle = h\langle\underline{x}, \underline{\theta}\rangle = [5.275, 12.770, 15.027]^T$. If you alter the two parameters, you get a two-dimensional *curved* surface, called **model surface**, in three-dimensional space.

The estimation problem again consists of finding the point $\underline{\widehat{\eta}}$ on the model surface that is closest to $\underline{Y}$. The parameter values that correspond to this point $\underline{\widehat{\eta}}$ are then the

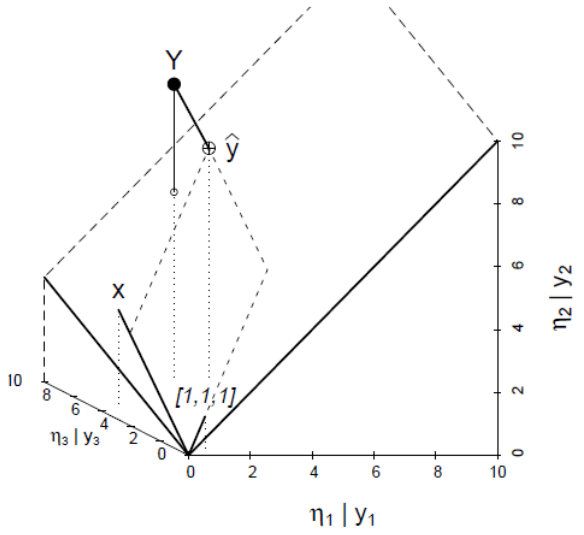

Figure 1.2.c.: Geometrical illustration of simple linear regression in case of three observations. Values of $\underline{\eta}\left\langle\underline{\beta}\right\rangle = \beta_0 + \beta_1\underline{x}$ for varying parameters $[\beta_0, \beta_1]$ given $\underline{x}$ lead to a plane in three-dimensional space. The diagram also shows the point on the surface that is closest to $\underline{Y} = [Y_1, Y_2, Y_3]$. It is the fitted value $\widehat{\underline{y}}$ and determines the estimated parameters $\widehat{\underline{\beta}}$.

estimated parameter values $\widehat{\underline{\theta}} = [\widehat{\theta}_1, \widehat{\theta}_2]^T$. Figure Figure 1.2.d illustrates the nonlinear case.

In this three-observations example, we can read the estimated parameters directly off the graph here: $\widehat{\theta}_1$ is a bit less than 21 and $\widehat{\theta}_2$ is a bit larger than 0.6.

e **Approach for the Minimization Problem.** The main idea of the usual algorithm for minimizing the sum of squares (see 1.2.a) is as follows: If a preliminary best value $\underline{\theta}^{(\ell)}$ exists, we approximate the model surface with the plane that touches the surface at the point $\underline{\eta}\left\langle\underline{\theta}^{(\ell)}\right\rangle = h\left\langle\underline{x}; \underline{\theta}^{(\ell)}\right\rangle$ (tangent plane). Now we are looking for the point on that plane that lies closest to $\underline{Y}$. This is the same as estimation in a linear regression problem. This new point lies on the plane, but not on the surface that corresponds to the nonlinear problem. However, it determines a parameter vector $\underline{\theta}^{(\ell+1)}$ that we use as starting value for the next iteration.

f **Linear Approximation.** To determine the tangent plane we need the partial derivatives

$$A_i^{(j)}\langle\underline{\theta}\rangle := \frac{\partial \eta_i\langle\underline{\theta}\rangle}{\partial \theta_j} ,$$

that can be summarized by an $n \times p$ matrix $\boldsymbol{A}$. The approximation of the model surface $\underline{\eta}\langle\underline{\theta}\rangle$ by the tangent plane at a parameter value $\underline{\theta}^*$ is

$$\eta_i\langle\underline{\theta}\rangle \approx \eta_i\langle\underline{\theta}^*\rangle + A_i^{(1)}\langle\underline{\theta}^*\rangle(\theta_1 - \theta_1^*) + \ldots + A_i^{(p)}\langle\underline{\theta}^*\rangle(\theta_p - \theta_p^*)$$

or, in matrix notation,

$$\underline{\eta}\langle\underline{\theta}\rangle \approx \underline{\eta}\langle\underline{\theta}^*\rangle + \boldsymbol{A}\langle\underline{\theta}^*\rangle(\underline{\theta} - \underline{\theta}^*) .$$

If we now add a random error, we get a linear regression model

$$\widetilde{\underline{Y}} = \boldsymbol{A}\langle\underline{\theta}^*\rangle\underline{\beta} + \underline{E}$$

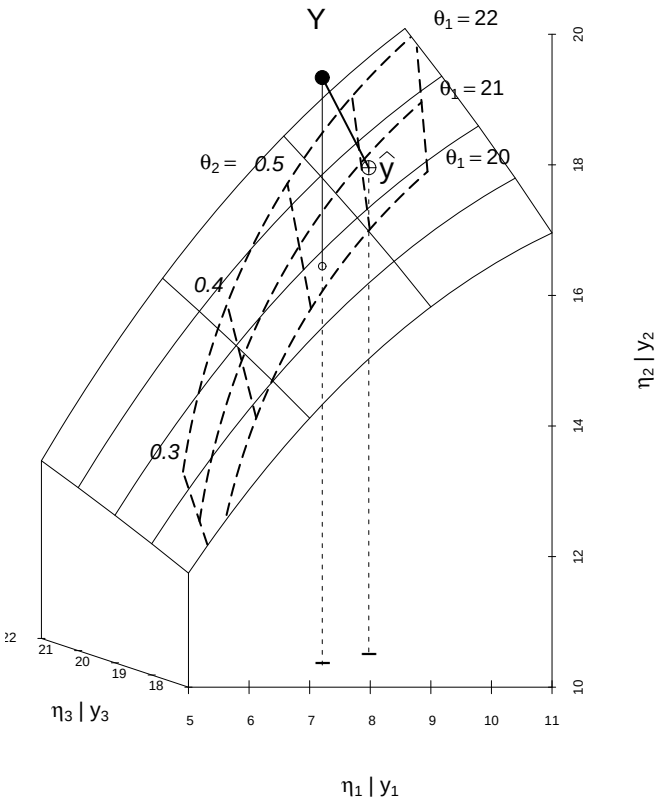

Figure 1.2.d.: Geometrical illustration of nonlinear regression. The values of $\underline{\eta}\langle\underline{\theta}\rangle := h\langle\underline{x};\theta_1,\theta_2\rangle$ for varying parameters $[\theta_1,\theta_2]$ lead to a two-dimensional "model surface" in three-dimensional space. The lines on the model surface correspond to constant θ_1 and θ_2, respectively. The vector of the estimated model values $\widehat{\underline{y}} = h\left\langle\underline{x};\widehat{\underline{\theta}}\right\rangle$ is the point on the model surface that is closest to $\underline{Y}$.

with "preliminary residuals" $\widetilde{Y}_i = Y_i - \eta_i\langle\underline{\theta}^*\rangle$ as response variable, the columns of $\boldsymbol{A}$ as predictors and the coefficients $\beta_j = \theta_j - \theta_j^*$ (a model without intercept β_0).

g **Gauss-Newton Algorithm.** The Gauss-Newton algorithm starts with an initial value $\underline{\theta}^{(0)}$ for $\underline{\theta}$, solving the just introduced linear regression problem for $\underline{\theta}^* = \underline{\theta}^{(0)}$ to find a correction $\underline{\beta}$ and hence an improved value $\underline{\theta}^{(1)} = \underline{\theta}^{(0)} + \underline{\beta}$. Again, the approximated model is calculated, and thus the "preliminary residuals" $\underline{Y} - \underline{\eta}\left\langle\underline{\theta}^{(1)}\right\rangle$ and the partial derivatives $\boldsymbol{A}\left\langle\underline{\theta}^{(1)}\right\rangle$ are determined, leading to $\underline{\theta}_2$. This iteration step is continued until the the correction $\underline{\beta}$ is small enough. (Further details can be found in Appendix A.1.)

It can not be guaranteed that this procedure actually finds the minimum of the sum of squares. The better the p-dimensional model surface at the minimum $\widehat{\underline{\theta}} = (\widehat{\theta}_1,\ldots,\widehat{\theta}_p)^T$ can be locally approximated by a p-dimensional plane and the closer the initial value $\underline{\theta}^{(0)}$ is to the solution, the higher are the chances of finding the optimal value.

* Algorithms usually determine the derivative matrix $\boldsymbol{A}$ numerically. In more complex problems the numerical approximation can be insufficient and cause convergence problems. For such situations it is an advantage if explicit expressions for the partial derivatives can be used to determine the derivative matrix more reliably (see also Section 2.3).

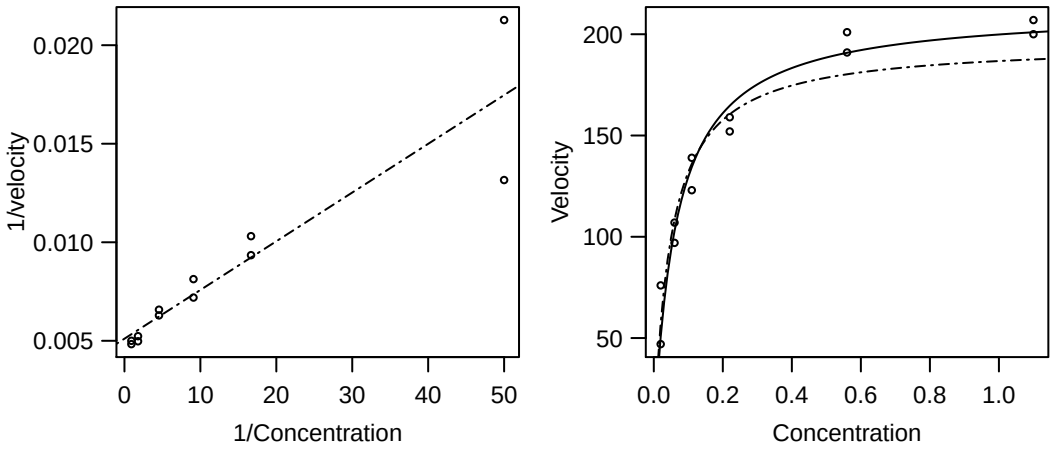

Figure 1.2.j.: Puromycin. Left: Regression function in the linearized problem. Right: Regression function $h\langle x;\underline{\theta}\rangle$ for the starting values $\underline{\theta} = \underline{\theta}^{(0)}$ (- - - - -) and for the least squares estimation $\underline{\theta} = \widehat{\underline{\theta}}$ (———).

h Starting Values. An iterative procedure always requires a starting value. Good starting values help to find a solution more quickly and more reliably.

Several simple but useful principles for determining starting values can be used:

- use prior knowledge;
- interpret the behavior of the expectation function $h\langle\rangle$ in terms of the parameters analytically or graphically;
- transform the expectation function $h\langle\rangle$ analytically to obtain linear behavior;
- reduce dimensionality by substituting values for some parameters or by evaluating the function at specific design values; and
- use conditional linearity.

i Starting Value from Prior Knowledge. As already noted in the introduction, nonlinear models are often based on theoretical considerations of the corresponding application area. Already existing **prior knowledge** from similar experiments can be used to get a starting value. To ensure the quality of the chosen starting value, it is advisable to graphically represent the regression function $h\langle x;\underline{\theta}\rangle$ for various possible starting values $\underline{\theta} = \underline{\theta}^0$ together with the data (e.g., as in Figure 1.2.j, right).

j Starting Values From Transform the Expectation Function. Often – because of the distribution of the error term – one is forced to use a nonlinear regression function even though the expectation function $h\langle\rangle$ cpuld be transformed to a linear function. However, the linearized expectation function can be used to get starting values.

In the Puromycin example the regression function is linearizable: The reciprocal values of the two variables fulfill

$$\widetilde{y} = \frac{1}{y} \approx \frac{1}{h\langle x;\underline{\theta}\rangle} = \frac{1}{\theta_1} + \frac{\theta_2}{\theta_1}\frac{1}{x} = \beta_0 + \beta_1\widetilde{x}\,.$$

The least squares solution for this modified problem is $\widehat{\underline{\beta}} = [\widehat{\beta}_0,\ \widehat{\beta}_1]^T = (0.00511,\ 0.000247)^T$ (Figure 1.2.j, left). This leads to the starting values

$$\theta_1^{(0)} = 1/\widehat{\beta}_0 = 196, \quad \theta_2^{(0)} = \widehat{\beta}_1/\widehat{\beta}_0 = 0.048.$$

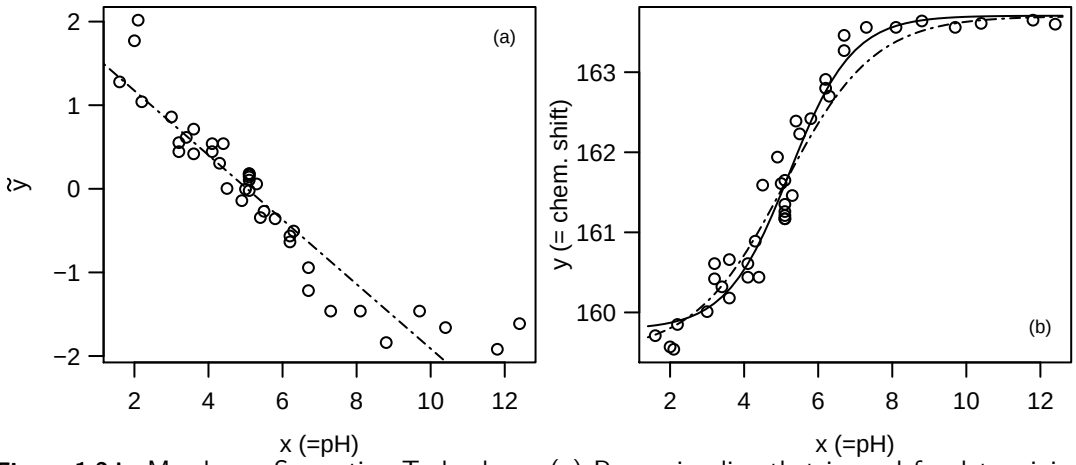

Figure 1.2.l.: Membrane Separation Technology. (a) Regression line that is used for determining the starting values for θ_3 and θ_4. (b) Regression function $h\langle x; \underline{\theta}\rangle$ for the starting value $\underline{\theta} = \underline{\theta}^{(0)}$ (- - - - -) and for the least squares estimator $\underline{\theta} = \widehat{\underline{\theta}}$ (———).

k **Starting Values from Interpreting the Behavior of $h\langle\rangle$.** It is often helpful to consider the geometrical features of the regression function.

In the Puromycin Example we can derive a starting value in another way: θ_1 is the response value for $x = \infty$. Since the regression function is monotonically increasing, we can use the maximal y_i-value or a visually determined "asymptotic value" $\theta_1^{(0)} = 207$ as starting value for θ_1. The parameter θ_2 is the x-value, such that y reaches half of the asymptotic value θ_1. This leads to $\theta_2^{(0)} = 0.06$.

The starting values thus result from a geometrical interpretation of the parameters and a rough estimate can be determined by "fitting by eye".

Example l **Membrane Separation Technology (cont'd)** In the Membrane Separation Technology example we let $x \to \infty$, so $h\langle x; \underline{\theta}\rangle \to \theta_1$ (since $\theta_4 < 0$); for $x \to -\infty$, $h\langle x; \underline{\theta}\rangle \to \theta_2$. From Figure 1.1.f (a) we see that $\theta_1 \approx 163.7$ and $\theta_2 \approx 159.5$. Once we know θ_1 and θ_2, we can linearize the regression function by

$$\tilde{y} := \log_{10}\left(\frac{\theta_1^{(0)} - y}{y - \theta_2^{(0)}}\right) = \theta_3 + \theta_4 x\ .$$

This is called **conditional linearity** of the expectation function. The linear regression model leads to the starting values $\theta_3^{(0)} = 1.83$ and $\theta_4^{(0)} = -0.36$, respectively.

With this starting value the algorithm converges to the solution $\widehat{\theta}_1 = 163.7$, $\widehat{\theta}_2 = 159.8$, $\widehat{\theta}_3 = 2.675$ and $\widehat{\theta}_4 = -0.512$. The functions $h\langle\cdot; \underline{\theta}^{(0)}\rangle$ and $h\langle\cdot; \widehat{\underline{\theta}}\rangle$ are shown in Figure 1.2.l (b).

* The property of conditional linearity of a function can also be useful to develop an algorithm specifically suited for this situation (see e.g. Bates and Watts, 1988).

m **Self-Starter Function.** For repeated use of the same nonlinear regression model some automated way of providing starting values is demanded nowadays. Basically, we should be able to collect all the manual steps which are necessary to obtain the initial values for a nonlinear regression model into a function, and use it to generate the starting values. Such a function is called a **self-starter function** and should allow

Model	Mean Function	Name of Self-Starter Function
Biexponential	$A1 \cdot e^{-x \cdot e^{lrc1}} + A2 \cdot e^{-x \cdot e^{lrc2}}$	`SSbiexp(x, A1, lrc1, A2, lrc2)`
Asymptotic regression	$Asym + (R0 - Asym) \cdot e^{-x \cdot e^{lrc}}$	`SSasymp(x, Asym, R0, lrc)`
Asymptotic regression with offset	$Asym \cdot (1 - e^{-(x-c0) \cdot e^{lrc}})$	`SSasympOff(x, Asym, lrc, c0)`
Asymptotic regression (c0 = 0)	$Asym \cdot (1 - e^{-x \cdot e^{lrc}})$	`SSasympOrig(x, Asym, lrc)`
First-order compartment	$x1 \cdot \frac{e^{lKe+lKa-lCl}}{e^{lKa}-e^{lKe}} \cdot (e^{-x2 \cdot e^{lKe}} - e^{-x2 \cdot e^{lKa}})$	`SSfol(x1, x2, lKe, lKa, lCl)`
Gompertz	$Asym \cdot e^{-b2 \cdot b3x}$	`SSgompertz(x, Asym, b2, b3)`
Logistic	$A + \frac{B-A}{1+e^{(xmid-x)/scal}}$	`SSfpl(x, A, B, xmid, scal)`
Logistic (A = 0)	$\frac{Asym}{1+e^{(xmid-x)/scal}}$	`SSlogis(x, Asym, xmid, scal)`
Michaelis-Menten	$Vm \cdot \frac{x}{K+x}$	`SSmicmen(x, Vm, K)`
Weibull	$Asym - Drop \cdot e^{-e^{lrc} \cdot xpwr}$	`SSweibull(x, Asym, Drop, lrc, pwr)`

Table 1.2.m.: Available self-starter functions for `nls()` which come with the standard installation of R.

to run the estimating procedure easily and smoothly.

Self-starter functions are specific for a given mean function and calculate starting values for a given dataset. The challenge is to design a self-starter function robustly. That is, its resulting values must allow the estimation algorithm to converge to the parameter estimates.

One of the self-starter knowledge bases comes with the standard installation of R. You find an overview in Table 1.2.m and some more details, including informations how to write a self-starter yourself, see, e.g. in Ritz and Streibig (2008).

1.3 Inference Based on Linear Approximations

a The estimator $\widehat{\underline{\theta}}$ is the value of $\underline{\theta}$ that optimally fits the data. We now ask *which parameter values $\underline{\theta}$ are compatible with the observations.* The **confidence region** is the set of all these values. For an individual parameter θ_j the confidence region is a **confidence interval**.

These concepts have been discussed in great detail in the modul "Linear Regression Analysis". As a look on the summary output of a `nls` object shows, it looks very similar to the summary output of a fitted linear regression model.

Example b **Membrane Separation Technology (cont'd)** The R summary output for the Membrane Separation Technology example can be found in R Output 1.3.b. The parameter estimates are in column `Estimate`, followed by the estimated approximate standard errors (`Std. Error`) and the test statistics (`t value`), that are approximately t_{n-p} distributed. The corresponding p-values can be found in column `Pr(>|t|)`. The estimated standard deviation $\widehat{\sigma}$ of the random error E_i is here labelled as "residual standard error".

```
> Mem.fit <- nls(delta ~ (T1 + T2*10^(T3+T4*pH))/(10^(T3+T4*pH)+1),
           D.membran, start=list(T1=163.7, T2=159.5, T3=1.83, T4=-0.36))
> summary(Mem.fit)
Formula: delta ~ (T1 + T2 * 10^(T3 + T4 * pH)) / (10^(T3 + T4 * pH) + 1)

Parameters:
     Estimate  Std. Error   t value  Pr(>|t|)  Residual standard error:
 T1  163.7056      0.1262  1297.256   < 2e-16
 T2  159.7846      0.1594  1002.194   < 2e-16
 T3    2.6751      0.3813     7.015  3.65e-08
 T4   -0.5119      0.0703    -7.281  1.66e-08

0.2931 on 35 degrees of freedom
Number of iterations to convergence: 7
Achieved convergence tolerance: 5.517e-06
```

R-Output 1.3.b: Summary of the fit of the Membrane Separation Technology example.

c Going into details, it is immediately clear that the inference is a matter more complex. It is not possible to write down exact results. However, one can derive inference results if the number n of observations goes to infinity. Then they look like in linear regression analysis. Such results can be used now for finite n, but hold just *approximately*.

In this section, we will explore these so-called asymptotic results in some more details.

d The **asymptotic properties** of the estimator can be derived from the linear approximation. Indeed, the inference in nonlinear regression is approximately the same as in the linear regression problem mentioned in 1.2.f

$$\underline{\widetilde{Y}} = \boldsymbol{A}\left\langle \underline{\theta}^* \right\rangle \underline{\beta} + \underline{E}\,,$$

when the parameter vector $\underline{\theta}^*$ used in the linearization is at the solution. If the estimation procedure has converged (i.e. $\underline{\theta}^* = \widehat{\underline{\theta}}$), then $\underline{\beta} = 0$ (otherwise this would not be the solution). The standard errors of the coefficients $\widehat{\underline{\beta}}$ – or more generally the covariance matrix of $\widehat{\underline{\beta}}$ – then approximate the corresponding quantities for $\widehat{\underline{\theta}}$.

* More precisely: The standard errors characterize the uncertainties that are generated by the random fluctuations of the data. The available data have led to the estimator $\widehat{\underline{\theta}}$. If the data would have been slightly different, then $\widehat{\underline{\theta}}$ would still be approximately correct, thus we accept the fact that it is good enough for the linearization. The estimation of $\underline{\beta}$ for the new data set would thus lie as far from the estimated value for the available data, as is quantified by the distribution of the parameters in the linear problem.

e **Asymptotic Distribution of the Least Squares Estimator.** It follows that the least squares estimator $\widehat{\underline{\theta}}$ is asymptotically normally distributed

$$\widehat{\underline{\theta}} \overset{as.}{\sim} \mathcal{N}\left\langle \underline{\theta}, \boldsymbol{V}\left\langle \underline{\theta} \right\rangle \right\rangle,$$

with asymptotic covariance matrix $\boldsymbol{V}\left\langle \underline{\theta} \right\rangle = \sigma^2 (\boldsymbol{A}\left\langle \underline{\theta} \right\rangle^T \boldsymbol{A}\left\langle \underline{\theta} \right\rangle)^{-1}$, where $\boldsymbol{A}\left\langle \underline{\theta} \right\rangle$ is the $n \times p$ matrix of partial derivatives (see 1.2.f).

To explicitly determine the covariance matrix $V\left\langle \underline{\theta} \right\rangle$, $\boldsymbol{A}\left\langle \underline{\theta} \right\rangle$ is calculated using $\widehat{\underline{\theta}}$ instead of the unknown $\underline{\theta}$ and is denoted as $\widehat{\boldsymbol{A}}$. For the error variance σ^2 we plug-in the usual estimator. Hence, we can write

$$\widehat{\boldsymbol{V}} = \hat{\sigma}^2 \left(\widehat{\boldsymbol{A}}^T \widehat{\boldsymbol{A}} \right)^{-1}$$

where

$$\widehat{\sigma}^2 = \frac{S(\underline{\widehat{\theta}})}{n-p} = \frac{1}{n-p}\sum_{i=1}^{n}\left(y_i - \eta_i\left\langle\underline{\widehat{\theta}}\right\rangle\right)^2 \qquad \text{and} \qquad \widehat{\boldsymbol{A}} = \boldsymbol{A}\left\langle\underline{\widehat{\theta}}\right\rangle .$$

Hence, the distribution of the estimated parameters is approximately determined and we can (like in linear regression) derive standard errors and confidence intervals, or confidence ellipses (or ellipsoids) if multiple variables are considered jointly.

The denominator $n-p$ in the estimator $\widehat{\sigma}^2$ was already introduced in linear regression to ensure that the estimator is unbiased. Tests and confidence intervals were not based on the normal and Chi-square distribution but on the **t- and F-distribution**. They take into account that the estimation of σ^2 causes additional random fluctuation. Even if the distributions are no longer exact, the approximations are more exact if we do this in nonlinear regression too. Asymptotically the difference goes to zero.

Based on these considerations we can construct approximate $(1-\alpha)\cdot 100\%$ confidence intervals:

$$\widehat{\theta}_k \pm q_{1-\alpha/2}^{t_{n-p}} \cdot \sqrt{\widehat{V}_{kk}}\,,$$

where $\widehat{V}_{kk}$ is the kth diagonal element of $\widehat{\boldsymbol{V}}$.

Example f **Puromycin (cont'd)** Based on the summary output given in 1.3.b the approximate 95% confidence interval for the parameter θ_1 is

$$163.706 \pm q_{0.975}^{t_{35}} \cdot 0.1262 = 163.706 \pm 0.256\,.$$

Based on analogues calculation we obtain the following 95% confidence intervall for the parameters θ_1, θ_2, θ_3, and θ_4:

θ_1:	[163.45,	163.96]
θ_2:	[159.46,	160.11]
θ_3:	[1.90,	3.45]
θ_4:	[-0.65,	-0.37]

Example g **Puromycin (cont'd)** In order to check the influence of treating an enzyme with Puromycin a general model for the data (with and without treatment) can be formulated as follows:

$$Y_i = \frac{(\theta_1 + \theta_3 z_i)x_i}{\theta_2 + \theta_4 z_i + x_i} + E_i,$$

where z is the indicator variable for the treatment ($z_i = 1$ if treated, $z_i = 0$ otherwise).

R Output 1.3.g shows that the parameter θ_4 is not significantly different from 0 at the 5% level since the p-value of 0.167 is larger then the level (5%). However, the treatment has a clear influence that is expressed through θ_3; the 95% confidence interval covers the region $52.398 \pm 9.5513 \cdot 2.09 = [32.4, 72.4]$ (the value 2.09 corresponds to the 97.5% quantile of the t_{19} distribution).

```
Formula: velocity ~ (T1 + T3 * (treated == T)) * conc/(T2 + T4
                     * (treated == T) + conc)
Parameters:
     Estimate  Std. Error  t value  Pr(>|t|)  Residual standard error: 10.4
 T1   160.280       6.896   23.242  2.04e-15
 T2     0.048       0.008    5.761  1.50e-05
 T3    52.404       9.551    5.487  2.71e-05
 T4     0.016       0.011    1.436     0.167

on 19 degrees of freedom
Number of iterations to convergence: 6
Achieved convergence tolerance: 4.267e-06
```

R-Output 1.3.g: Computer output of the fit for the Puromycin example.

h Confidence Intervals for Function Values. Besides the parameters, the function value $h\langle \underline{x}_0, \underline{\theta}\rangle$ for a given $\underline{x}_0$ is often of interest. In linear regression the function value $h\left\langle \underline{x}_0, \underline{\beta}\right\rangle = \underline{x}_0^T\underline{\beta} =: \eta_0$ is estimated by $\widehat{\eta}_0 = \underline{x}_0^T\widehat{\underline{\beta}}$ and the corresponding $(1-\alpha)\cdot 100\%$ confidence interval is

$$\widehat{\eta}_0 \pm q_{1-\alpha/2}^{t_{n-p}} \cdot \operatorname{se}\langle\widehat{\eta}_0\rangle$$

where

$$\operatorname{se}\langle\widehat{\eta}_0\rangle = \widehat{\sigma}\sqrt{\underline{x}_0^T\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\underline{x}_0}.$$

Using asymptotic approximations we can specify confidence intervals for the function values $h\langle \underline{x}_0; \underline{\theta}\rangle$ for nonlinear h. If $\widehat{\eta}_0 := h\left\langle x_0, \widehat{\underline{\theta}}\right\rangle$ is linearly approximated at $\underline{\theta}$ we get

$$\widehat{\eta}_0 \approx h\langle x_0, \underline{\theta}\rangle + \underline{a}_0^T\left(\widehat{\underline{\theta}} - \underline{\theta}\right) \text{ where } \underline{a}_0 = \frac{\partial h\langle x_0, \underline{\theta}\rangle}{\partial \underline{\theta}}.$$

Based on the asymptotic results of 1.3.e, we obtain the following asymptotic distribution of $\widehat{\eta}_0$:

$$\widehat{\eta}_0 \overset{as.}{\sim} \mathcal{N}\left(h\langle x_0, \underline{\theta}\rangle, \underline{a}_0^T\,\boldsymbol{V}\langle\underline{\theta}\rangle\,\underline{a}_0\right).$$

To be able to use this expression for the explicit calculation of an approximate $(1-\alpha)$ confidence interval for the function value $\eta_0\langle\underline{\theta}\rangle := h\langle\underline{x}_0, \underline{\theta}\rangle$, we have to replace the unknown parameter $\underline{\theta}$ by its estimate $\widehat{\underline{\theta}}$. This yields

$$\eta_0\left\langle\widehat{\underline{\theta}}\right\rangle \pm q_{1-\alpha/2}^{t_{n-p}} \cdot \operatorname{se}\left\langle\eta_0\left\langle\widehat{\underline{\theta}}\right\rangle\right\rangle,$$

where

$$\operatorname{se}\left\langle\eta_0\left\langle\widehat{\underline{\theta}}\right\rangle\right\rangle = \widehat{\sigma}\sqrt{\widehat{\underline{a}}_0^T\left(\widehat{\boldsymbol{A}}^T\widehat{\boldsymbol{A}}\right)^{-1}\widehat{\underline{a}}_0} = \sqrt{\widehat{\underline{a}}_0^T\widehat{\boldsymbol{V}}\widehat{\underline{a}}_0}$$

with

$$\widehat{\underline{a}}_0 = \frac{\partial h\langle x_0, \underline{\theta}\rangle}{\partial \underline{\theta}}\Big|_{\underline{\theta}=\widehat{\underline{\theta}}}.$$

(definition of $\widehat{\boldsymbol{V}}$ see 1.3.e). If $\underline{x}_0$ is equal to an observed $\underline{x}_i$, $\widehat{\underline{a}}_0$ equals the corresponding row of the matrix $\widehat{\boldsymbol{A}}$.

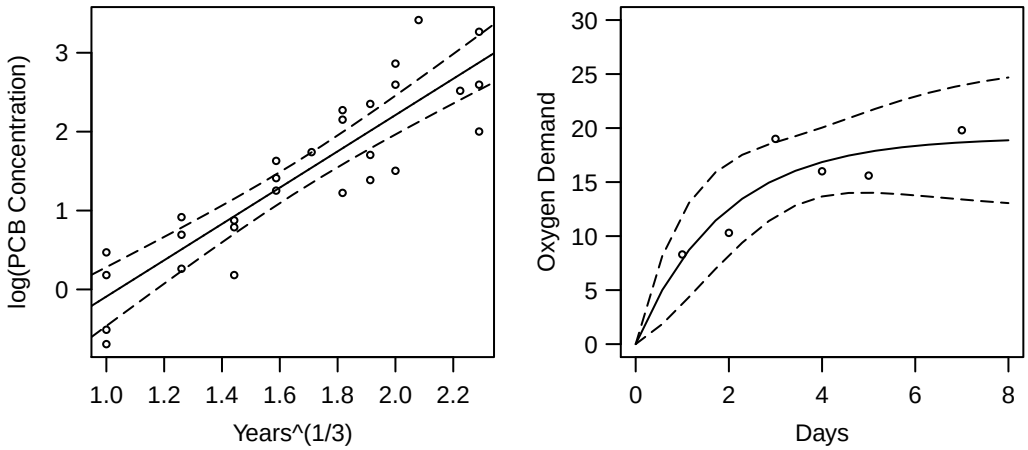

Figure 1.3.i.: Left: Confidence band for an estimated line for a linear problem. Right: Confidence band for the estimated curve $h\langle x, \underline{\theta}\rangle$ in the oxygen demand example.

i Confidence Band. The expression for the $(1-\alpha)$ confidence interval for $\eta_0\langle\underline{\theta}\rangle := h\langle x_0, \underline{\theta}\rangle$ also holds for arbitrary x_0. As in linear regression, it is illustrative to represent the limits of these intervals as a "confidence band" that is a function of x_0. See Figure 1.3.i for the confidence bands for the examples "Puromycin" and "Biochemical Oxygen Demand".

Confidence bands for linear and nonlinear regression functions behave differently: For linear functions the confidence band has minimal width at the center of gravity of the predictor variables and gets wider the further away one moves from the center (see Figure 1.3.i, left). In the nonlinear case, the bands can have arbitrary shape. Because the functions in the "Puromycin" and "Biochemical Oxygen Demand" examples must go through zero, the interval shrinks to a point there. Both models have a horizontal asymptote and therefore the band reaches a constant width for large x (see Figure 1.3.i, right).

j Prediction Interval. The confidence band gives us an idea of the **function values** $h\langle x\rangle$ (the expected values of Y for a given x). However, it does not answer the question where **future observations** Y_0 for given $\underline{x}_0$ will lie. This is often more interesting than the question of the function value itself; for example, we would like to know where the measured value of oxygen demand will lie for an incubation time of 6 days.

Such a statement is a prediction about a **random variable** and should be distinguished from a confidence interval, which says something about a **parameter**, which is a fixed (but unknown) number. Hence, we call the region **prediction interval**. More about this in Chapter 3.1.

k Variable Selection. In nonlinear regression, unlike in linear regression, variable selection is usually not an important topic, because

- there is no one-to-one relationship between parameters and predictor variables. Usually, the number of parameters is different than the number of predictors.
- there are seldom problems where we need to clarify whether an explanatory variable is necessary or not – the model is derived from the underlying theory (e.g., "enzyme kinetics").

However, there is sometimes the reasonable question whether a subset of the parameters in the nonlinear regression model can appropriately describe the data (see example "Puromycin").

- **Model Selection.** Sometimes, there is the situation where we need to find the most appropriate model for a dataset, for which a collection of candidate models is available. Most of the time, these models are *not* nested submodels of each other. Then one can use Akaike's information criterion (AIC) to select the best model and run a residual analysis to confirm the selection.

2 Improved Inference and Its Visualisation

2.1 Likelihood Based Inference

a The quality of the approximate confidence region strongly depends on the quality of the linear approximation. Also, the convergence properties of the optimization algorithm are influenced by the quality of the linear approximation. With a somewhat larger computational effort, linearity can be checked graphically and – at the same time – we get more precise confidence intervals.

b **F-Test for Model Comparison.** To test a null hypothesis $\underline{\theta} = \underline{\theta}^*$ for the whole parameter vector or also $\theta_j = \theta_j^*$ for an individual component, we can use an **F-test for model comparison** like in linear regression. Here, we compare the sum of squares $S\langle\underline{\theta}^*\rangle$ that arises under the null hypothesis with the sum of squares $S\langle\widehat{\underline{\theta}}\rangle$ (for $n \to \infty$ the F- test is the same as the so-called likelihood-ratio test, and the sum of squares is, up to a constant, equal to the negative log-likelihood).

Let us first consider a null-hypothesis $\underline{\theta} = \underline{\theta}^*$ for the whole parameter vector. The test statistic is

$$T = \frac{n-p}{p} \frac{S\langle\underline{\theta}^*\rangle - S\langle\widehat{\underline{\theta}}\rangle}{S\langle\widehat{\underline{\theta}}\rangle} \overset{(as.)}{\sim} F_{p,n-p}.$$

Searching for all null-hypotheses that are not rejected leads us to the confidence region

$$\left\{ \underline{\theta} \;\middle|\; S\langle\underline{\theta}\rangle \le S\langle\widehat{\underline{\theta}}\rangle \left(1 + \tfrac{p}{n-p}\, q\right) \right\},$$

where $q = q_{1-\alpha}^{F_{p,n-p}}$ is the $(1-\alpha)$ quantile of the F-distribution with p and $n-p$ degrees of freedom.

In linear regression we get the same (exact) confidence region if we use the (multivariate) normal distribution of the estimator $\widehat{\underline{\beta}}$. In the nonlinear case the results are different. The region that is based on the F-test is *not* based on the linear approximation in 1.2.f and hence is (much) more exact.

c **Confidence Regions for p=2.** For $p = 2$, we can find the confidence regions by calculating $S\langle\underline{\theta}\rangle$ on a grid of $\underline{\theta}$ values and determine the borders of the region through interpolation, as is common for contour plots. Figure 2.1.c illustrates both the confidence region based on the linear approximation and based on the F-test for the example "Puromycin" (left) and for "Biochemical Oxygen Demand" (right).

For $p > 2$ contour plots do not exist. In the next chapter we will introduce graphical tools that also work in higher dimensions. They depend on the following concepts.

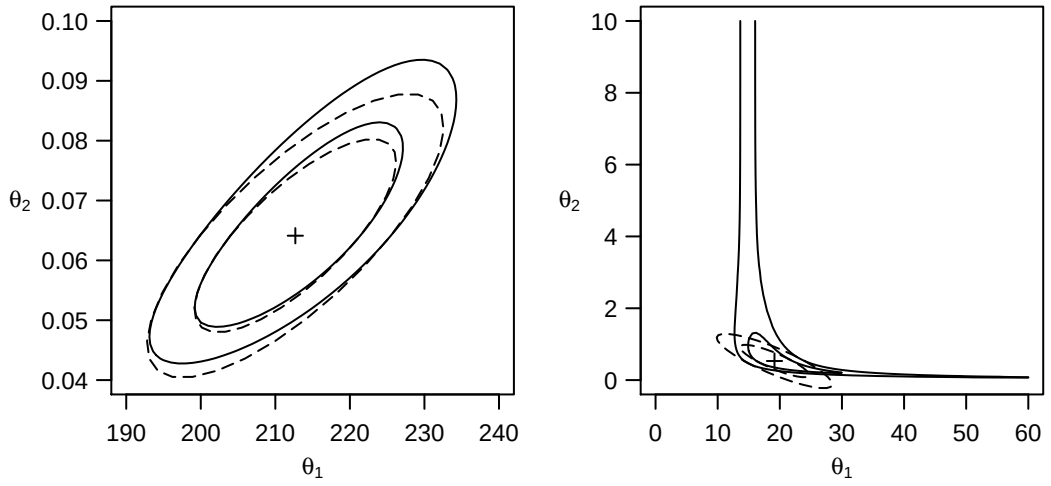

Figure 2.1.c.: Nominal 80 and 95% likelihood contours (——) and the confidence ellipses from the asymptotic approximation (– – – –). + denotes the least squares solution. In the Puromycin example (left) the agreement is good and in the oxygen demand example (right) it is bad.

d F-Test for Individual Parameters. Now the question of interest is whether an individual parameter θ_k can be equal to a certain value θ_k^*. Such a null hypothesis makes no statement about the remaining parameters. The model that corresponds to this null hypothesis and fits the data best is determined by a least squares estimation of the remaining parameters with θ_k fixed at $= \theta_k^*$. So, $S\langle\theta_1, \ldots, \theta_k^*, \ldots, \theta_p\rangle$ is minimized with respect to θ_j, $j \neq k$. We denote the minimum by $\widetilde{S}_k$ and the minimizer θ_j by $\widetilde{\theta}_j$. Both values depend on θ_k^*. We therefore write $\widetilde{S}_k\langle\theta_k^*\rangle$ and $\widetilde{\theta}_j\langle\theta_k^*\rangle$.

The test statistic for the F-test with null hypothesis $H_0 : \theta_k = \theta_k^*$ is

$$\widetilde{T}_k = (n-p)\,\frac{\widetilde{S}_k\langle\theta_k^*\rangle - S\left\langle\underline{\widehat{\theta}}\right\rangle}{S\left\langle\underline{\widehat{\theta}}\right\rangle}.$$

It follows (approximately) an $F_{1,n-p}$ distribution.

We can now construct a confidence interval by (numerically) solving the equation $\widetilde{T}_k = q_{0.95}^{F_{1,n-p}}$ for θ_k^*. It has a solution that is less than $\widehat{\theta}_k$ and one that is larger.

e t-Test via F-Test. In linear regression and in the previous chapter we have calculated tests and confidence intervals from a test value that follows a t-distribution (t-test for the coefficients). Is this another test?

It turns out that the test statistic of the t-test in linear regression turns into the test statistic of the F-test if we square it. Hence, both tests are equivalent. In nonlinear regression, the F-test is not equivalent to the t-test discussed in the last chapter (1.3.e). However, we can transform the F-test to a t-test that is more accurate than the one of the last chapter:

From the test statistic of the F-test, we take the square-root and add the sign of $\widehat{\theta}_k - \theta_k^*$,

$$T_k\langle\theta_k^*\rangle := \operatorname{sign}\left\langle\widehat{\theta}_k - \theta_k^*\right\rangle\,\frac{\sqrt{\widetilde{S}_k\langle\theta_k^*\rangle - S\left\langle\underline{\widehat{\theta}}\right\rangle}}{\widehat{\sigma}}.$$

$\operatorname{sign}\langle a\rangle$ denotes the sign of a and $\widehat{\sigma}^2 = S\left\langle \underline{\widehat{\theta}} \right\rangle / (n-p)$. This test statistic is (approximately) t_{n-p} distributed.

In the linear regression model, T_k is – as already pointed out – equal to the test statistic of the usual t-test,

$$T_k\langle \theta_k^* \rangle = \frac{\widehat{\theta}_k - \theta_k^*}{\operatorname{se}\left\langle \widehat{\theta}_k \right\rangle} .$$

f Confidence Intervals for Function Values via F-test. With this technique we can also determine confidence intervals for a function value at a point x_0. For this we reparameterize the original problem so that a parameter, say ϕ_1, represents the function value $h\langle x_0\rangle$ and proceed as in 2.1.d.

2.2 Profile t-Plot and Profile Traces

a Profile t-Function and Profile t-Plot. The graphical tools for checking the linear approximation are based on the just discussed t-test, that actually doesn't use this approximation. We consider the test statistic T_k (2.1.e) as a function of its arguments θ_k and call it **profile t-function** (in the last chapter the arguments were denoted with θ_k^*, now for simplicity we leave out the *). For linear regression we get, as can be seen from 2.1.e, a straight line, while for nonlinear regression the result is a monotone increasing function. The graphical comparison of $T_k\langle\theta_k\rangle$ with a straight line is the so-called **profile t-plot**. Instead of θ_k, it is common to use a standardized version

$$\delta_k\langle\theta_k\rangle := \frac{\theta_k - \widehat{\theta}_k}{\operatorname{se}\left\langle \widehat{\theta}_k \right\rangle}$$

on the horizontal axis because it is used in the linear approximation. The comparison line is then the "diagonal", i.e. the line with slope 1 and intercept 0.

The more the profile t-function is curved, the stronger the nonlinearity in a neighborhood of θ_k. Therefore, this representation shows how good the linear approximation is in a neighborhood of $\widehat{\theta}_k$ (the neighborhood that is statistically important is approximately determined by $|\delta_k\langle\theta_k\rangle| \leq 2.5$). In Figure 2.2.a it is apparent that in the Puromycin example the nonlinearity is minimal, while in the Biochemical Oxygen Demand example it is large.

In Figure 2.2.a we can also read off the confidence intervals according to 2.1.e. For convenience, the probabilites $P\langle T_k \leq t\rangle$ of the corresponding t-distributions are marked on the right vertical axis. For the Biochemical Oxygen Demand example this results in a confidence interval without upper bound!

Example b Membrane Separation Technology (cont'd) As 2.2.a shows, from the profile t-plot we can graphically read out corresponding confidence intervals that are based on the profile t-function. The R function `confint(...)` numerically calculates the desired confidence interval on the basis of the profile t-function. R Output 2.2.b shows the corresponding R output from the membrane separation example. In this case, no large differences from the classical calculation method are apparent (cf. 1.3.f).

```
> confint(Mem.fit, level=0.95)
Waiting for profiling to be done...
             2.5%        97.5%
 T1   163.4661095  163.9623685
 T2   159.3562568  160.0953953
 T3     1.9262495    3.6406832
 T4    -0.6881818   -0.3797545
```

R-Output 2.2.b: Membrane separation technology example: R output for the confidence intervals that are based on the profile t-function.

c Likelihood Profile Traces. The **likelihood profile traces** are another useful graphical tool. Here the estimated parameters $\widetilde{\theta}_j$, $j \neq k$ for fixed θ_k (see 2.1.d) are considered as functions $\widetilde{\theta}_j^{(k)}\langle\theta_k\rangle$.

The graphical representation of these functions would fill a whole matrix of diagrams, but without diagonals. It is worthwhile to combine the "opposite" diagrams of this matrix: Over the representation of $\widetilde{\theta}_j^{(k)}\langle\theta_k\rangle$ we superimpose $\widetilde{\theta}_k^{(j)}\langle\theta_j\rangle$ in mirrored form so that the axes have the same meaning for both functions.

Figure 2.2.c shows one of these diagrams for both our two examples. Additionally, contours of confidence regions for $[\theta_1, \theta_2]$ are plotted. It can be seen that that the profile traces intersect the contours at points where they have horizontal or vertical tangents.

The representation does not only show the nonlinearities, but is also useful for the understanding of **how the parameters influence each other**. To understand this, we go back to the case of a linear regression function. The profile traces in the individual diagrams then consist of two lines, that intersect at the point $[\widehat{\theta}_1, \widehat{\theta}_2]$. If we standardize the parameter by using $\delta_k\langle\theta_k\rangle$ from 2.2.a, one can show that the slope of the trace $\widetilde{\theta}_j^{(k)}\langle\theta_k\rangle$ is equal to the correlation coefficient c_{kj} of the estimated coefficients $\widehat{\theta}_j$ and $\widehat{\theta}_k$. The "reverse line" $\widetilde{\theta}_k^{(j)}\langle\theta_j\rangle$ then has, compared with the horizontal axis, a slope

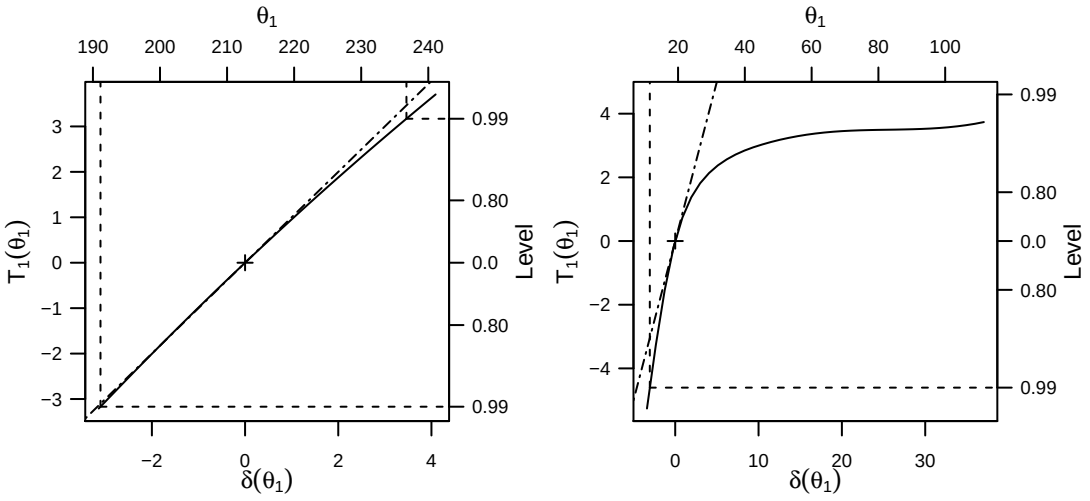

Figure 2.2.a.: Profile t-plot for the first parameter for both the Puromycin (left) and the Biochemical Oxygen Demand example (right). The dashed lines show the applied linear approximation and the dotted line the construction of the 99% confidence interval with the help of $T_1\langle\theta_1\rangle$.

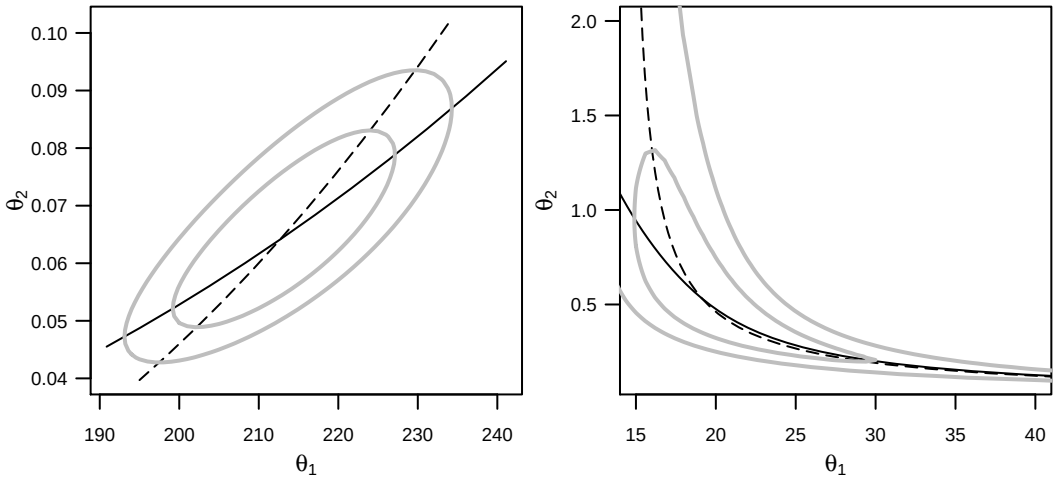

Figure 2.2.c.: Likelihood profile traces for the Puromycin and Oxygen Demand examples, with 80%- and 95% confidence regions (gray curves).

of $1/c_{kj}$. The angle between the lines is thus a monotone function of the correlation. It therefore measures the **collinearity** between the two predictor variables. If the correlation between the parameter estimates is zero, then the traces are orthogonal to each other.

For a nonlinear regression function, both traces are curved. The angle between them still shows how strongly the two parameters θ_j and θ_k interplay, and hence how their estimators are correlated.

Example d **Membrane Separation Technology (cont'd)** All profile t-plots and profile traces can be put in a triangular matrix, as can be seen in Figure 2.2.d. Most profile traces are strongly curved, meaning that the regression function tends to a strong nonlinearity around the estimated parameter values. Even though the profile traces for θ_3 and θ_4 are straight lines, a further problem is apparent: The profile traces lie on top of each other! This means that the parameters θ_3 and θ_4 are strongly collinear. Parameter θ_2 is also collinear with θ_3 and θ_4, although more weakly.

e * **Good Approximation of Two Dimensional Likelihood Contours.** The profile traces can be used to construct very accurate approximations for two dimensional projections of the likelihood contours (see Bates and Watts, 1988). Their calculation is computationally less demanding than for the corresponding exact likelihood contours.

2.3 Parameter Transformations

a In this section we study the effects of transforming the parameters. This topic is based on the fact that the mean regression function can usually be written down by mathematically equivalent expressions. For example, the two expression for the Michaelis-Menten function are equivalent:

$$\frac{\theta_1 x}{\theta_2 + x} = \frac{x}{\phi_1 + \phi_2 x}\,.$$

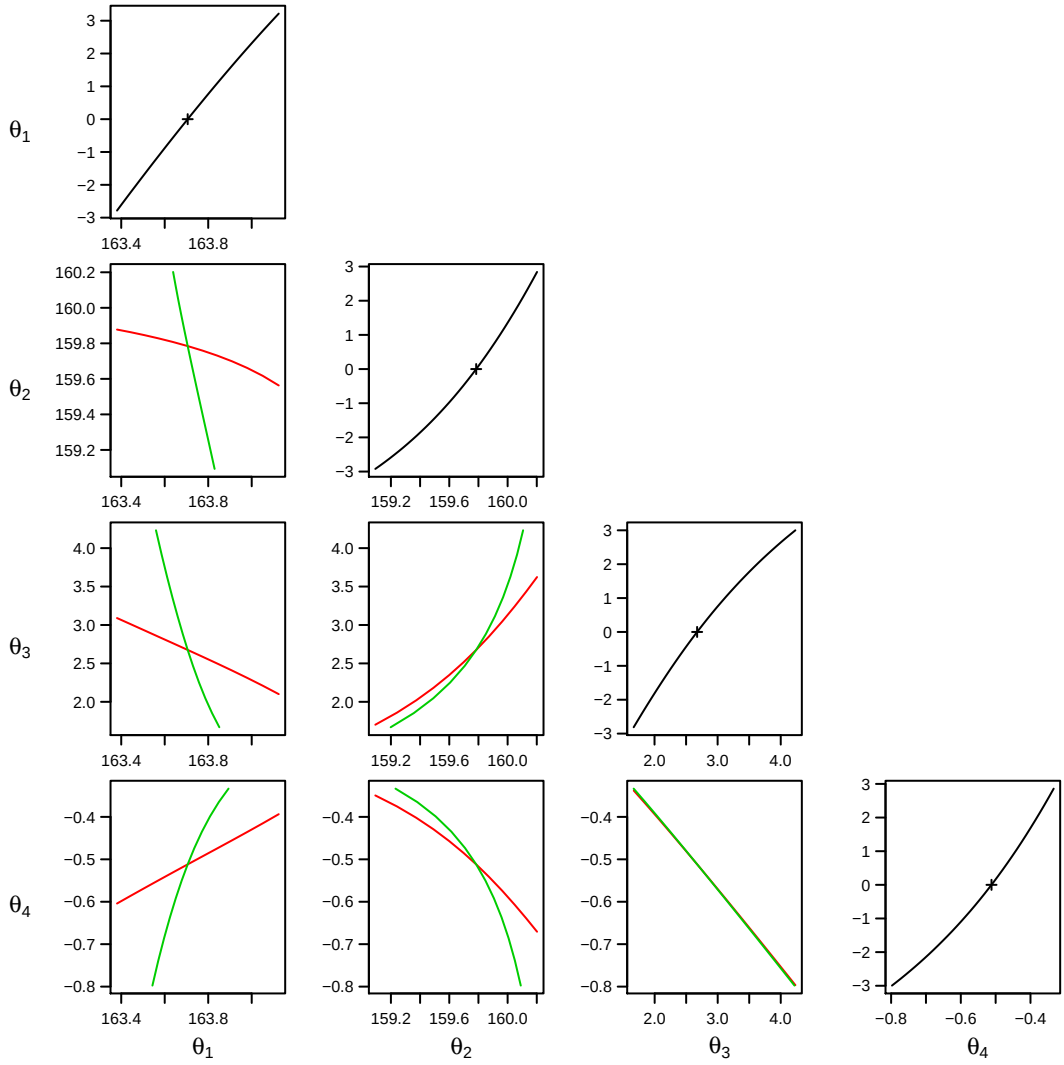

Figure 2.2.d.: Profile t-plots and Profile Traces for the Example "Membrane Separation Technology". The $+$ in the profile t-plot denotes the least squares solution.

Hence, we must have the following relations between the two sets of parameters:

$$\phi_1 = \frac{\theta_2}{\theta_1} \quad \text{and} \quad \phi_1 = \frac{1}{\theta_1}\,.$$

In another example we have the two equivalent expressions

$$\theta_1 e^{\theta_2 x} = \phi_1 \phi_2^x$$

and the following relations between the two sets of parameters:

$$\phi_1 = \theta_1 \quad \text{and} \quad \phi_1 = e^{\theta_2}\,.$$

Why should such equivalent expressions of the mean regression function be of any interest to us?

b **Parameter transformations** are primarily used to improve the linear approximation and therefore improve the convergence behavior and the **quality of the confidence interval**.

We point out that parameter transformations, unlike transformations of the response variable (see 1.1.k), do *not* change the error part of the model. Hence, they are not helpful if the assumptions about the distribution of the random error are violated. It is the quality of the linear approximation and the statistical statements based on it that are being changed.

Sometimes the transformed parameters are very difficult **to interpret**. The important questions often concern individual parameters of the original parameter set. Nevertheless, we can work with transformations: We derive more accurate confidence regions for the transformed parameters and can transform them back to get results for the original parameters.

c **Restricted Parameter Regions.** Often the admissible region of a parameter is restricted, e.g. because the regression function is only defined for positive values of a parameter. Usually, such a constraint is ignored to begin with and we wait to see whether and where the algorithm converges. According to experience, parameter estimation will end up in a reasonable range if the model describes the data well and the data contain enough information for determining the parameters.

Sometimes, though, problems occur in the course of the computation, especially if the parameter value that best fits the data lies near the border of the admissible region. The simplest way to deal with such problems is via transformation of the parameter.

Examples

- The parameter θ should be positive. Through a transformation $\theta \longrightarrow \phi = \ln\langle\theta\rangle$, $\theta = \exp\langle\phi\rangle$ is always positive for all possible values of $\phi \in \mathbb{R}$:

$$h\langle x, \theta\rangle \longrightarrow h\langle x, \exp\langle\phi\rangle\rangle.$$

- The parameter should lie in the interval (a, b). With the log transformation $\theta = a + (b - a)/(1 + \exp\langle\phi\rangle)$, θ can (for arbitrary $\phi \in \mathbb{R}$) only take values in (a, b).
- In the model

$$h\langle x, \underline{\theta}\rangle = \theta_1 \exp\langle-\theta_2 x\rangle + \theta_3 \exp\langle-\theta_4 x\rangle$$

with $\theta_2, \theta_4 > 0$ the parameter pairs (θ_1, θ_2) and (θ_3, θ_4) are interchangeable, i.e. $h\langle x, \underline{\theta}\rangle$ does not change. This can create uncomfortable optimization problems, because the solution is not unique. The constraint $0 < \theta_2 < \theta_4$ that ensures the uniqueness is achieved via the transformation $\theta_2 = \exp\langle\phi_2\rangle$ und $\theta_4 = \exp\langle\phi_2\rangle(1 + \exp\langle\phi_4\rangle)$. The function is now

$$h\langle x, (\theta_1, \phi_2, \theta_3, \phi_4)\rangle = \theta_1 \exp\left\langle - \exp\langle\phi_2\rangle x\right\rangle + \theta_3 \exp\left\langle - \exp\langle\phi_2\rangle(1 + \exp\langle\phi_4\rangle)x\right\rangle .$$

d **Parameter Transformation for Collinearity.** A simultaneous variable and parameter transformation can be helpful to weaken **collinearity** in the partial derivative vectors. For example, the model $h\langle x, \underline{\theta}\rangle = \theta_1 \exp\langle -\theta_2 x\rangle$ has derivatives

$$\frac{\partial h}{\partial \theta_1} = \exp\langle -\theta_2 x\rangle \ , \qquad \frac{\partial h}{\partial \theta_2} = -\theta_1 x \exp\langle -\theta_2 x\rangle \ .$$

If all x values are positive, both vectors

$$\begin{aligned} \underline{a}_1 &:= (\exp\langle -\theta_2 x_1\rangle, \ldots, \exp\langle -\theta_2 x_n\rangle)^T \\ \underline{a}_2 &:= (-\theta_1 x_1 \exp\langle -\theta_2 x_1\rangle, \ldots, -\theta_1 x_n \exp\langle -\theta_2 x_n\rangle)^T \end{aligned}$$

tend to disturbing collinearity. This collinearity can be avoided if we use **centering**. The model can be written as $h\langle x; \underline{\theta}\rangle = \theta_1 \exp\langle -\theta_2 (x - x^* + x^*)\rangle$ With the re-parameterization $\phi_1 := \theta_1 \exp\langle -\theta_2 x^*\rangle$ and $\phi_2 := \theta_2$ we get

$$h\left\langle x; \underline{\phi}\right\rangle = \phi_1 \exp\langle -\phi_2 (x - x^*)\rangle \ .$$

The derivative vectors are approximately orthogonal if we chose the mean value of the x_i for x^*.

Example e **Membrane Separation Technology (cont'd)** In this example it is apparent from the approximate correlation matrix (Table 2.3.e, left half) that the parameters θ_3 and θ_4 are strongly correlated (we have already observed this in 2.2.d using the profile traces). If the model is re-parameterized to

$$y_i = \frac{\theta_1 + \theta_2\, 10^{\phi_3 + \theta_4 (x_i - \mathrm{med}\langle x_j\rangle)}}{1 + 10^{\phi_3 + \theta_4 (x_i - \mathrm{med}\langle x_j\rangle)}} + E_i, \ i = 1 \ldots n \,,$$

where $x^* := \mathrm{med}\,\langle x_j\rangle$, with $\phi_3 := \theta_3 + \theta_4 \,\mathrm{med}\,\langle x_j\rangle$, an improvement is achieved (right half of Table 2.3.e).

	θ_1	θ_2	θ_3		θ_1	θ_2	ϕ_3
θ_2	−0.256			θ_2	−0.256		
θ_3	−0.434	0.771		ϕ_3	0.323	0.679	
θ_4	0.515	−0.708	−0.989	θ_4	0.515	−0.708	−0.312

Table 2.3.e.: Correlation matrices for the Membrane Separation Technology example for the original parameters (left) and the transformed parameters $\widetilde{\theta}_3$ (right).

f **Reparameterization.** In Chapter 2.2 we have presented means for graphical evaluation of the linear approximation. If the approximation is considered inadequate we would like to improve it. An appropriate reparameterization can contribute to this. So, for example, for the model

$$h\left\langle \underline{x}, \underline{\theta} \right\rangle = \frac{\theta_1 \theta_3 (x^{(2)} - x^{(3)})}{1 + \theta_2 x^{(1)} + \theta_3 x^{(2)} + \theta_4 x^{(3)}}$$

the reparameterization

$$h\langle \underline{x}, \underline{\phi} \rangle = \frac{x^{(2)} - x^{(3)}}{\phi_1 + \phi_2 x^{(1)} + \phi_3 x^{(2)} + \phi_4 x^{(3)}}$$

with $\phi_1 := 1/(\theta_1\theta_3)$, $\phi_2 := \theta_2/(\theta_1\theta_3)$, $\phi_3 := 1/\theta_1$, and $\phi_4 := \theta_4/(\theta_1\theta_3)$ is recommended by (Ratkowsky, 1985) (see also exercises).

Example g **Membrane Separation Technology (cont'd)** The parameter transformation in 2.3.e leads to a satisfactory result, as far as correlation is concerned. If we look at the likelihood contours or the profile t-plot and the profile traces, the parameterization is still not satisfactory.

An intensive search for further improvements leads to the following transformations that turn out to have satisfactory profile traces (see Figure 2.3.g):

$$\begin{aligned} \phi_1 &:= \frac{\theta_1 + \theta_2\, 10^{\phi_3}}{10^{\phi_3} + 1}\,, & \phi_2 &:= \log_{10}\left(\frac{\theta_1 - \theta_2}{10^{\phi_3} + 1}\, 10^{\phi_3}\right)\,, \\ \phi_3 &:= \theta_3 + \theta_4 \operatorname{med}\langle x_j \rangle & \phi_4 &:= 10^{\theta_4}\,. \end{aligned}$$

The model now reads

$$Y_i = \phi_1 + 10^{\phi_2}\, \frac{1 - \phi_4{}^{(x_i - \operatorname{med}\langle x_j \rangle)}}{1 + 10^{\phi_3}\, \phi_4{}^{(x_i - \operatorname{med}\langle x_j \rangle)}} + E_i\,.$$

and we get the result shown in R Output 2.3.g

h It turns out that a **successful reparametrization is very data set specific**. A reason is that nonlinearities and correlations between estimated parameters depend on the (estimated) parameter vector itself. Therefore, no generally valid recipe can be given. This often makes the search for appropriate reparametrizations very challenging.

i **Failure of Gaussian Error Propagation.** Even if a parameter transformation helps us deal with difficulties with the convergence behavior of the algorithm or the quality of the confidence intervals, the **original parameters** often have a physical interpretation. We take the simple transformation example $\theta \longrightarrow \phi := \ln\langle\theta\rangle$ from 2.3.c. The fitting of the model opens with an estimator $\widehat{\phi}$ with estimated standard error $\widehat{\sigma}_{\widehat{\phi}}$. An obvious estimator for θ is then $\widehat{\theta} = \exp\langle\widehat{\phi}\rangle$. The standard error for $\widehat{\theta}$ can be determined with the help of the Gaussian error propagation law (see, e.g., Stahel, 2007, Sec. 6.11):

$$\widehat{\sigma}^2_{\widehat{\theta}} \approx \left(\left. \frac{\partial \exp\langle\phi\rangle}{\partial \phi} \right|_{\phi = \widehat{\phi}} \right)^2 \widehat{\sigma}^2_{\widehat{\phi}} = \left(\exp\langle\widehat{\phi}\rangle\right)^2 \widehat{\sigma}^2_{\widehat{\phi}}$$

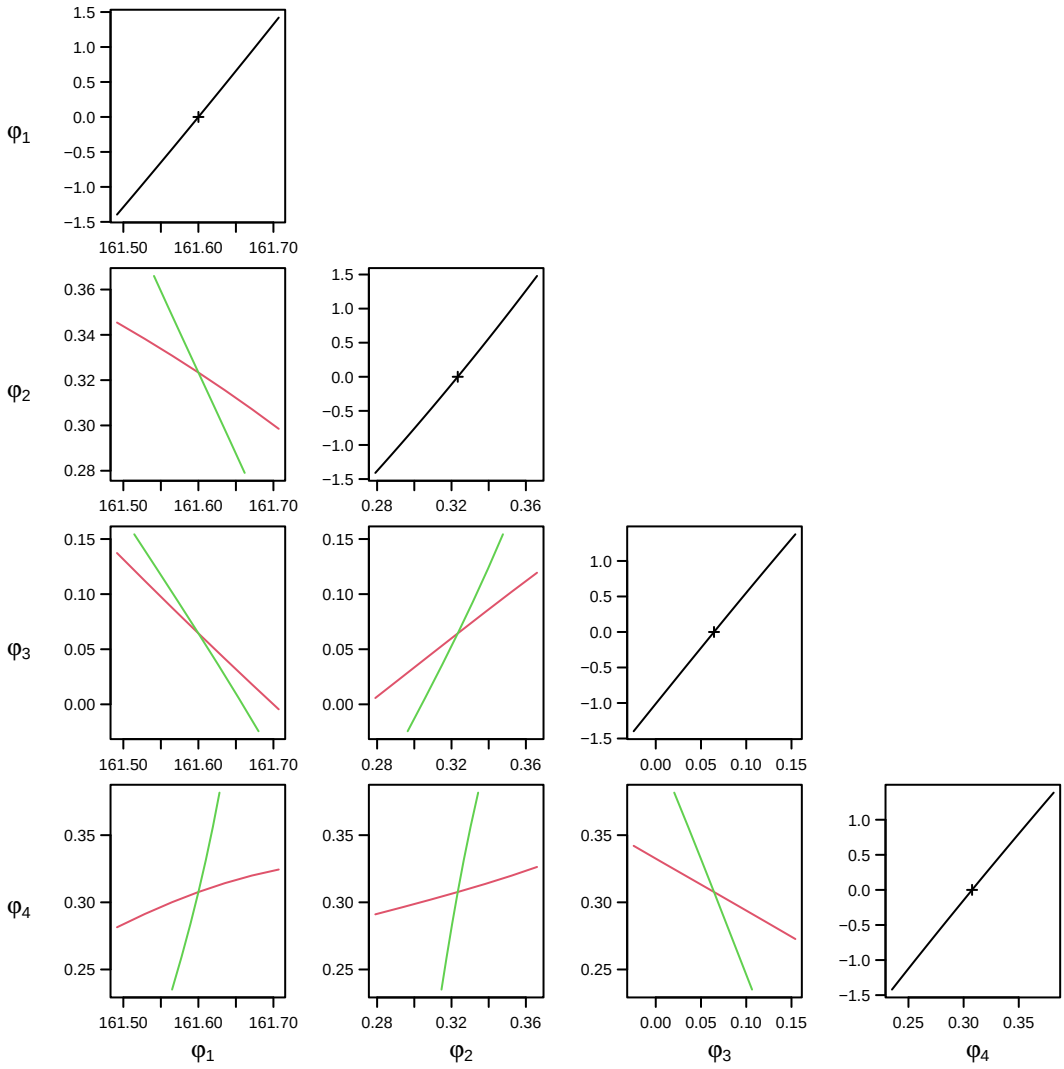

Figure 2.3.g.: Profile t-plot and profile traces for the Membrane Separation Technology example according to the given transformations.

or

$$\widehat{\sigma}_{\widehat{\theta}} \approx \exp\langle\widehat{\phi}\rangle \, \widehat{\sigma}_{\widehat{\phi}} \, .$$

From this we get the approximate 95% confidence interval for θ:

$$\exp\langle\widehat{\phi}\rangle \pm \widehat{\sigma}_{\widehat{\theta}} \, q_{0.975}^{t_{n-p}} = \exp\langle\widehat{\phi}\rangle \left(1 \pm \widehat{\sigma}_{\widehat{\phi}} \, q_{0.975}^{t_{n-p}}\right) \, .$$

The Gaussian error propagation law is based on the linearization of the transformation function $g\langle\cdot\rangle$; concretely on $\exp\langle\cdot\rangle$. We have carried out the parameter transformation because the quality of the confidence intervals left a lot to be desired, then unfortunately this linearization negates what has been achieved and we are back where we started before the transformation.

```
Formula: delta ~ TT1 + 10^TT2 * (1 - TT4^pHR)/(1 + 10^TT3 * TT4^pHR)

Parameters:
       Estimate  Std. Error   t value  Pr(> |t|)  Residual standard error:
 TT1  161.60008     0.07389  2187.122   < 2e-16
 TT2    0.32336     0.03133    10.322  3.67e-12
 TT3    0.06437     0.05951     1.082     0.287
 TT4    0.30767     0.04981     6.177  4.51e-07

0.2931 on 35 degrees of freedom

Correlation of Parameter Estimates:
        TT1   TT2    TT3  Number of iterations to convergence: 5
 TT2  -0.56
 TT3  -0.77  0.64
 TT4   0.15  0.35  -0.31

Achieved convergence tolerance: 9.838e-06
```

R-Output 2.3.g: Membrane Separation Technology: Summary of the fit after parameter transformation.

The correct approach in such cases is to determine the confidence interval as presented in Chapter 2.1. If this is impossible for whatever reason, we can fall back on the following approximation.

j Confidence Intervals on the Original Scale (Alternative Approach). Even though parameter transformations help us in situations where we have problems with convergence of the algorithm or the quality of confidence intervals, the original parameters often remain the quantity of interest (e.g., because they have a nice physical interpretation). Consider the transformation $\theta \longrightarrow \phi = \ln\langle\theta\rangle$. Fitting the model results in an estimator $\widehat{\phi}$ and an estimated standard error $\widehat{\sigma}_{\widehat{\phi}}$. Now we can construct a confidence interval for θ. We have to search all θ for which $\ln\langle\theta\rangle$ lies in the interval

$$\widehat{\phi} \pm \widehat{\sigma}_{\widehat{\phi}} q_{0.975}^{t_{df}}.$$

Generally formulated: Let g be the transformation of ϕ to $\theta = g\langle\phi\rangle$. Then

$$\left\{\theta \,:\, g^{-1}\langle\theta\rangle \in \left[\widehat{\phi} - \widehat{\sigma}_{\widehat{\phi}} q_{0.975}^{t_{df}},\, \widehat{\phi} + \widehat{\sigma}_{\widehat{\phi}} q_{0.975}^{t_{df}}\right]\right\}$$

is an approximate 95% confidence interval for θ. If $g^{-1}\langle\cdot\rangle$ is strictly monotone increasing, this confidence interval is identical to

$$\left[g\left\langle\widehat{\phi} - \widehat{\sigma}_{\widehat{\phi}} q_{0.975}^{t_{df}}\right\rangle,\, g\left\langle\widehat{\phi} + \widehat{\sigma}_{\widehat{\phi}} q_{0.975}^{t_{df}}\right\rangle\right].$$

This procedure also ensures that, unlike the Gaussian error propagation law, the confidence interval is entirely in the region that is predetermined for the parameter. It is thus impossible that, for example, the confidence interval in the example $\theta = \exp\langle\phi\rangle$, unlike the interval from 2.3.i, can contain negative values.

However, this approach should only be used if the calculation based on the F-test from Chapter 2.1 is not possible. On the other hand, the author does prefer this approach to the Gaussian error propagation law.

2.4 Bootstrap

a An alternative to profile confidence intervals is to apply the **bootstrap** method. It is a resampling method and does not rely on the linear approximation as well. Bootstrap allows an estimation of the sampling distribution of almost any statistic using only very simple techniques.

The basic idea of bootstrapping is that inference about a parameter from sample data can be modelled by resampling the sample data and performing inference based on these resampled datasets. That is, bootstrapping treats inference of a parameter θ, given the sample data, as being analogous to inference of the estimated parameter $\widehat{\theta}$, given the resampled data.

The simplest bootstrap method involves taking the original dataset of n observations and sampling from it to form a new sample (called a 'resample' or bootstrap sample) that is also of size n. The bootstrap sample is taken from the original dataset using sampling with *replacement* so it is not identical with the original "real" sample. This process is repeated a large number of times (typically 1,000 or 10,000 times), and for each of these bootstrap samples we compute its parameter estimation (each of these are called bootstrap estimates). We now have a histogram of bootstrap estimates. This provides an estimate of the shape of the distribution of the parameter estimates from which we can answer questions about how much the parameter estimates varies.

b In regression analysis, one may resample from the complete observations, that are pairs of response and explanatory variables $(y_i, \underline{x}_i)$, or just from the residuals r_i. In the latter case the bootstrap observations $((y_i^*, \underline{x}_i))$ are constructed by $y_i^* = h\left\langle \underline{x}_i, \widehat{\underline{\theta}} \right\rangle + r_i^*$, where r_i^* is a resampled residual. In either case, we will call the bootstrap method **nonparametric**. Hence, there are also parametric bootstrap methods. A parametric bootstrap version is to assume that the residuals are Gaussian distributed and hence we resample from a Gaussian distribution with expectation 0 and variance $\widehat{\sigma}^2$, the variance estimate from the nonlinear regression fit.

c We will use the R function `nlsBoot()` in the package `nlstools`. This means that we use a nonparametric bootstrap approach where the mean centred residuals are bootstrapped. The residuals are mean centred because the residuals may have a non-zero mean with a *nonlinear* regression model (Venables and Ripley, 2002, Sec 8.4,). By default, `nlsBoot()` generates $B = 999$ datasets, and for each dataset the considered nonlinear regression model is fitted and the resulting parameter estimates stored.

The bootstrapped values are useful for assessing the marginal distributions of the parameters estimates. So we can start out by comparing them with a normal distribution, which they should ideally follow if the linear approximation holds.

d A basic method to construct confidence limits is based on quantiles in the empirical distribution of the bootstrap parameter estimates. If $B = 999$ bootstrap simulations are used, then the empirical distribution is based on 1'000 values: The original estimate

and the 999 bootstrap estimates. Thus, to construct a 95% bootstrap confidence interval, we take the 25th value and the 975th value among the 1'000 ordered estimates (ordered from smallest to largest) as left and right endpoints, respectively. This type of bootstrap confidence interval is called a **bootstrap percentile confidence interval**. There are better ways of constructing bootstrap confidence interval based on a natural derivation for statisticians.

The bootstrap confidence interval has the advantage of lying entirely within the range of plausible parameter values. This is in contrast to Wald confidence intervals, which for small sample size occasionally may give unrealistic lower and upper limits.

The bootstrap approach is somewhat computer-intensive, as the nonlinear regression model considered has to be refitted numerous times, but for many applications it will still be a feasible approach. Moreover, the linear approximation will improve as the sample size increases, and therefore the bootstrap approach may be most useful for small datasets.

Example e Biochemical Oxygen Demand (cont'd). In R Output 2.4.e the three introduced

```
## Wald:
> h <- summary(D.bod.nls)$coefficients[,2]
> coef(D.bod.nls) + qt(0.975, 19)*cbind(``2.5%''=-h, ``97.5%''=h)
            2.5%       97.5%  ## Profile Likelihood:
 Th1  13.9185602  24.3665858
 Th2   0.1060357   0.9561475


> confint(D.bod.nls)
Waiting for profiling to be done...
            2.5%      97.5%  ## Bootstrap:
 Th1  14.0845447  38.482510
 Th2   0.1356242   1.810125


> library(nlstools)
> D.bod.nls.Boot <- nlsBoot(D.bod.nls)
> summary(D.bod.nls.Boot)
------
Bootstrap estimates
        Th1        Th2  ------
 19.0667361  0.5365059

Bootstrap confidence intervals
            2.5%      97.5%
 Th1  16.0755863  25.699375
 Th2   0.2815905   1.088552
```

R-Output 2.4.e: Biochemical Oxygen Demand. 95% confidence interval determined by three different approaches: Based on Wald, profile likelihoods and bootstrap, respectively.

procedure to determine confidence intervals for the parameters in the Biochemical Oxygen Demand model are calculated. The results of the three approaches differ considerably. As we know from 2.1.c, the linear approximation approach is poor in this example. Hence, we are not surprised to see the difference between the result of the Wald approach and the result of the profile likelihood approach.

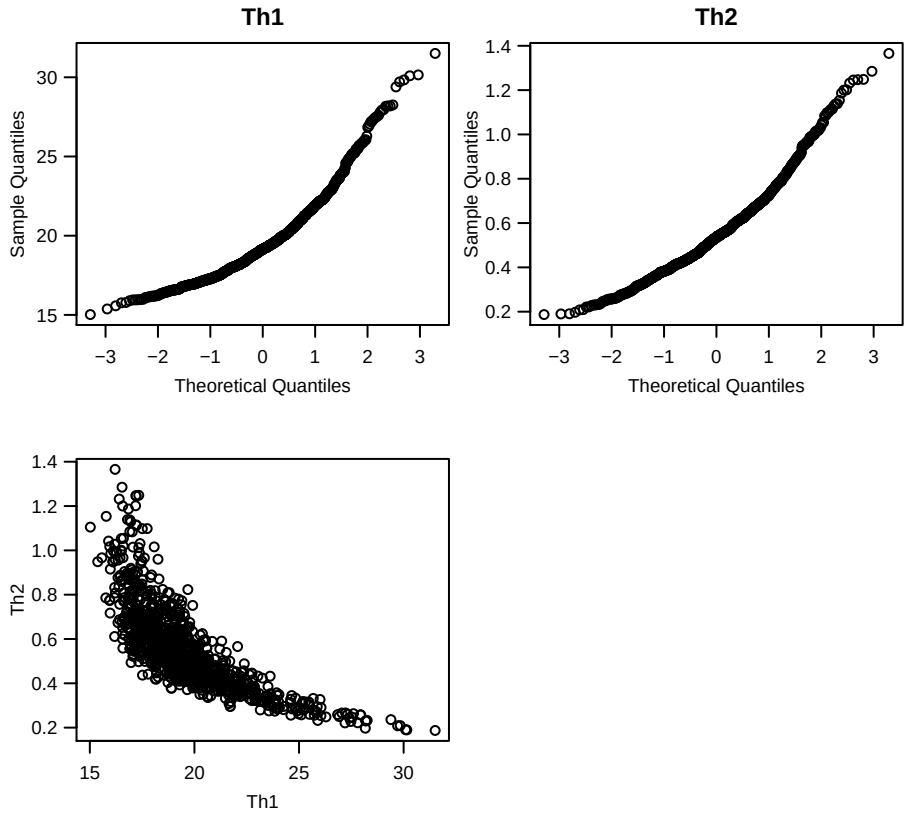

Figure 2.4.e.: Biochemical Oxygen Demand. Both the normal plots of the marginal bootstrap distributions of the parameters estimates in the top row and the scatterplot of the joint bootstrap distribution (the bottom left graphic) clearly shows the distortion from a Gaussian distribution.

The difference between the result of the profile likelihood approach and the result of the bootstrap approach is rather nebulous. Taking a look at the marginal bootstrap distributions of both parameter estimates and their joint distribution in Figure 2.4.e, we once again can reassure that the linear approximation is unsuitable. Compared to the likelihood contours in 2.1.c, which rely on the Gaussian error assumption, the joint bootstrap distribution does not identify a vertical "funnel" at the left-hand-side of the joint distribution. This difference yields different confidence intervals.

f Recap of this chapter.

- The commonly used **confidence intervals** in nonlinear regression analysis are based on a (crude) linear **approximation**.
- Use **graphical tools like profile t plots and profile traces** to assess the quality of the approximated confidence intervals (and hence the linear approximation).
- If insufficient:
  More accurate confidence intervals can be calculated for single parameters θ_k^* by using **profile t functions** (as in `confint()` implemented anyway).
- Convergence properties of the estimating algorithm and the quality of the Wald-type conference intervals can be improved by applying **suitable reparametrizations** (parameter transformations).

If the interpretation of the original parameters is crucial, then the confidence interval should also be backtransformed
* and not be determined by Gaussian error propagation rule.

- **Bootstrap confidence intervals** are an alternative to profile confidence intervals.
  - They are based on resampling techniques.
  - They do not rely on a linear approximation and
  - even do not assume that the errrors are Gaussian distributed

  Bootstrap allows an estimation of the sampling distribution of almost any statistic.

3 Prediction and Calibration

3.1 Prediction

a Besides the question of the set of plausible parameters (with respect to the given data, which we also call training data set), the question of the range of future observations is often of central interest. The difference between these two questions was already discussed in 1.3.j. In this chapter we want to answer the second question. We assume that the parameter $\underline{\theta}$ is estimated using the least squares method. What can we now say about a future observation Y_0 at a given point x_0?

Example b **Cress.** The concentration of an agrochemical material in soil samples can be studied through the growth behavior of a certain type of cress (nasturtium). 6 measurements of the response variable Y were made on each of 7 soil samples with predetermined (or measured with the largest possible precision) concentrations x. Hence, we assume that the x-values have no measurement error. The variable of interest is the weight of the cress per unit area after 3 weeks. A "log-logistic" model is used to describe the relationship between concentration and weight:

$$h\left\langle x; \underline{\theta} \right\rangle = \begin{cases} \theta_1 & \text{if } x = 0 \\ \frac{\theta_1}{1+\exp\langle \theta_2 + \theta_3 \ln\langle x \rangle \rangle} & \text{if } x > 0. \end{cases}$$

The data and the function $h\langle\cdot\rangle$ are illustrated in Figure 3.1.b. We can now ask ourselves which weight values will we see at a concentration of e.g. $x_0 = 3$? The answer can be given by a prediction using the fitted function.

* the dataset is available for example in the R package `investr` as `nasturtium`. In this lecture notes, the dataset is called `D.kresse`

Example c **Cress - Model Fitting.** Suitable start values are required to initialize the fitting process. The one for θ_1 is the mean value of weight at $x = 0$. The other two starting values can be determined by the linearization

$$\ln\left\langle \frac{\theta_1}{y} - 1 \right\rangle = \theta_2 + \theta_3 \ln x\,,$$

whereby the maximum weight value at $x = 0$ plus a small constant must be selected for θ_1 so that all values are defined. However, as the scatter diagram shows (not shown here), it is only appropriate to consider the linearization for x greater than 0.2. This results in the starting values -0.15 and 1 for θ_2 and θ_3. Then the nonlinear regression model can be fitted by

```
> N.cfn <- function(x, Th1, Th2, Th3){
        ifelse(x==0, Th1, Th1/(1+exp(Th2 + Th3*log(x))))}
> N.nls <- nls(weight ~ N.cfn(conc, Th1, Th2, Th3), data=D.kresse,
               start=list(Th1=900, Th2=-0.15, Th3=1))
```

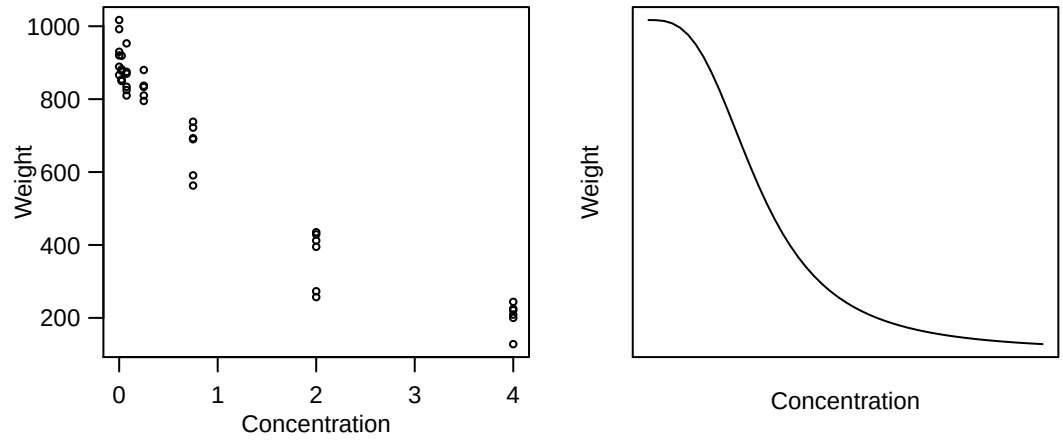

Figure 3.1.b.: Cress Example. Left: Representation of the data. Right: A typical shape of the applied regression function.

```
> summary(N.nls)

--- shortened ---

Parameters:
      Estimate  Std. Error  t value  Pr(>|t|)
 Th1  897.8629     13.7137    65.47   < 2e-16  ***
 Th2   -0.6144      0.1069    -5.75  1.15e-06  ***
 Th3    1.3503      0.1088    12.41  4.04e-15  ***
---
Residual standard error: 55.56 on 39 degrees of freedom

Number of iterations to convergence: 6
Achieved convergence tolerance: 1.007e-06
```

d Approximate Prediction Intervals - The Delta-Method Approach.. We can estimate the expected value $E\langle Y_0\rangle = h\langle x_0, \theta\rangle$ of the variable of interest Y at the point x_0 by $\widehat{\eta}_0 := h\langle x_0, \widehat{\theta}\rangle$. We also want to get an interval where a future observation will lie with high probability. So, we do not only have to take into account the randomness of the estimate $\widehat{\eta}_0$, but also the random error E_0. Analogous to linear regression, an at least approximate $(1-\alpha/2)$ prediction interval is given by

$$\widehat{\eta}_0 \pm q_{1-\alpha/2}^{t_{n-p}} \cdot \sqrt{\widehat{\sigma}^2 + \big(\operatorname{se}\langle\widehat{\eta}_0\rangle\big)^2}.$$

The calculation of $\operatorname{se}\langle\widehat{\eta}_0\rangle$ can be found in 1.3.h.

* **Derivation** The random variable Y_0 is the value of interest for an observation with predictor variable value x_0. Since we do not know the true curve (actually only the parameters), we have no choice but to study the deviations of the observations from the estimated curve,

$$R_0 = Y_0 - h\big\langle x_0, \widehat{\underline{\theta}}\big\rangle = \big(Y_0 - h\langle x_0, \underline{\theta}\rangle\big) - \big(h\big\langle x_0, \widehat{\underline{\theta}}\big\rangle - h\langle x_0, \underline{\theta}\rangle\big).$$

Even if $\underline{\theta}$ is unknown, we know the distribution of the expressions in the parentheses: Both are normally distributed random variables and they are independent because the first only depends on the "future" observation Y_0, the second only on the observations $Y_1, \ldots, Y_n$ that led to the estimated curve. Both have expected value 0; the variances add up to

$$\operatorname{var}\langle R_0\rangle \approx \sigma^2 + \sigma^2 \underline{a}_0^T (A^T A)^{-1} \underline{a}_0.$$

The described prediction interval follows by replacing the unknown values by their corresponding estimates.

Example e **Cress - Prediction.** As usual, a prediction can be calculated using the function `predict(...)` (more precisely with `predict.nls(...)`):

```
> predict(N.nls, newdata=data.frame(conc=c(0.5, 1.5, 2.5)))
[1] 740.6999 463.9442 313.4767
```

This function has arguments for the standard error and for intervals, but these are currently ignored because the method proposed in 3.1.d is not considered reliable enough. But often it is better to have some idea about the size of the prediction interval than none at all. Hence,

```
> require(investr)
> predFit(N.nls, newdata=data.frame(conc=c(0.5, 1.5, 2.5)), interval="prediction")
            fit       lwr       upr
 [1,]  740.6999  623.0639  858.3359
 [2,]  463.9442  347.0244  580.8640
 [3,]  313.4767  196.2272  430.7261
```

f **Prediction Versus Confidence Intervals.** If the sample size n of the training data set is very large, the estimated variance is dominated by the error variance $\widehat{\sigma}^2$. This means that the uncertainty in the prediction is then primarily caused by the random error. The second term in the expression for the variance reflects the uncertainty that is caused by the estimation of $\underline{\theta}$.

It is therefore clear that the prediction interval is wider than the confidence interval for the expected value, since the random error of the observation must be taken into account. The endpoints of such intervals are shown in Figure 3.2.c (left).

g * **Quality of the Approximation.** The derivation of the prediction interval in 3.1.d is based on the same approximation as in Chapter 1.3. The quality of the approximation can again be checked graphically.

h **Interpretation of the "Prediction Band".** The interpretation of the "prediction band" (as shown in Figure 3.2.c), is not straightforward. From our derivation it holds that

$$P\big(\, V_0^*\langle x_0\rangle \le Y_0 \le V_1^*\langle x_0\rangle \,\big) = 0.95,$$

where $V_0^*\langle x_0\rangle$ is the lower and $V_1^*\langle x_0\rangle$ the upper bound of the prediction interval for $h\langle x_0\rangle$. However, if we want to make a prediction about more than one future observation, then the number of the observations in the prediction interval is *not* binomially distributed with $\pi = 0.95$. The events that the individual future observations fall in the band are not independent; they depend on each other through the random borders V_0 and V_1. If, for example, the estimation of $\widehat{\sigma}$ randomly turns out to be too small, the band is too narrow for *all* future observations, and too many observations would lie outside the band.

3.2 Calibration

a The actual goal of the experiment in the **cress example** is to estimate the concentration of the agrochemical material from the weight of the cress. This means that we

would like to use the regression relationship in the "wrong" direction. This will cause problems with statistical inference. Such a procedure is often desired to **calibrate** a measurement method or to predict the result of a more expensive measurement method from a cheaper one. The regression curve in this relationship is often called a **calibration curve**. Another keyword for finding this topic is **inverse regression**.

Here, we would like to present a simple method that gives a useable result if simplifying assumptions hold.

b **Procedure under Simplifying Assumptions.** We assume that the predictor values x have no measurement error. In our example this is achieved if the concentrations of the agrochemical material are determined very carefully. For several soil samples with many different possible concentrations we carry out several independent measurements of the response value Y. This results in a training data set that is used to estimate the unknown parameters and the corresponding parameter errors.

Now, for a given value y_0 it is obvious to determine the corresponding x_0 value by simply inverting the regression function:

$$\widehat{x}_0 = h^{-1}\langle y_0, \widehat{\underline{\theta}}\rangle.$$

Here, h^{-1} denotes the inverse function of h. However, this procedure is only correct if $h\langle\cdot\rangle$ is monotone increasing or decreasing. Usually, this condition is fulfilled in calibration problems.

c **Accuracy of the Obtained Values.** Of course we now face the question about the accuracy of $\widehat{x}_0$. The problem seems to be similar to the prediction problem. However, here we observe y_0 and the corresponding value x_0 has to be estimated.

The answer can be formulated as follows: We treat x_0 as a *parameter* for which we want a confidence interval. Such an interval can be constructed (as always) from a test. We take as null hypothesis $x = x_0$. As we have seen in 3.1.d, Y lies with probability 0.95 in the prediction interval

$$\widehat{\eta}_0 \pm q_{1-\alpha/2}^{t_{n-p}} \cdot \sqrt{\widehat{\sigma}^2 + \left(\operatorname{se}\langle\widehat{\eta}_0\rangle\right)^2},$$

where $\widehat{\eta}_0$ was a compact notation for $h\langle x_0, \widehat{\theta}\rangle$. Therefore, this interval is an acceptance interval for the value Y_0 (which here plays the role of a test statistic) under the null hypothesis $x = x_0$. Figure 3.2.c illustrates all prediction intervals for all possible values of x_0 for the given interval in the Cress example.

d **Illustration.** Figure 3.2.c (right) illustrates the approach for the Cress example: Measured values y_0 are compatible with parameter values x_0 in the sense of the test, if the point $[x_0, y_0]$ lies in the (prediction interval) band. Hence, we can thus determine the set of x_0 values that are compatible with a given observation y_0. They form the dashed interval, which can also be described as the set

$$\left\{x \,:\, |y_0 - h\langle x, \widehat{\underline{\theta}}\rangle| \le q_{1-\alpha/2}^{t_{n-p}} \cdot \sqrt{\widehat{\sigma}^2 + \left(\operatorname{se}\left\langle h\langle x, \widehat{\underline{\theta}}\rangle\right\rangle\right)^2}\right\}.$$

This interval has been constructed by inverting the prediction interval of 3.2.c. It is the desired confidence interval (or **calibration interval**) for x_0.

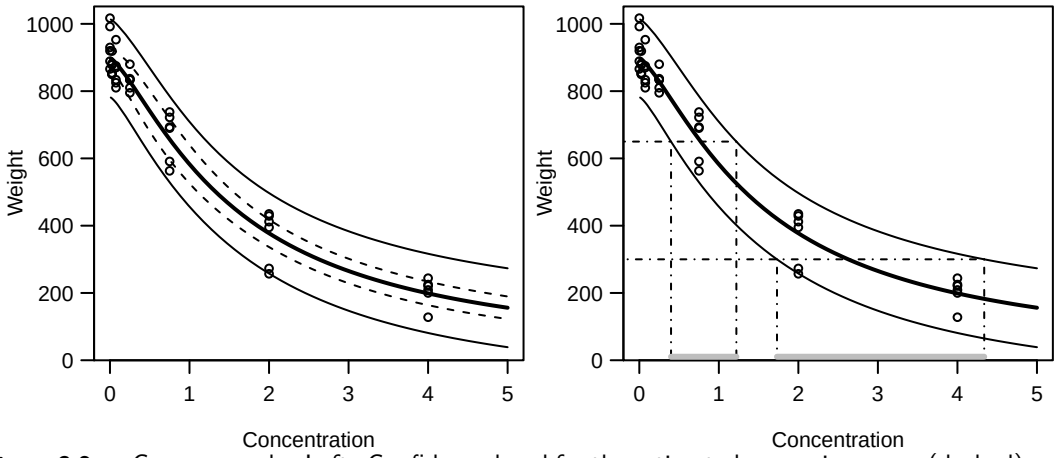

Figure 3.2.c.: Cress example. Left: Confidence band for the estimated regression curve (dashed) and prediction band (solid). Right: Schematic representation of how a calibration interval is determined, at the points $y_0 = 650$ and $y_0 = 300$. The resulting intervals are $[0.4, 1.22]$ and $[1.73, 4.34]$, respectively.

If we have m values to determine y_0, we apply the above method to $\bar{y}_0 = \sum_{j=0}^{m} y_{0j}/m$:

$$\left\{ x \,:\, |\bar{y}_0 - h\langle x, \widehat{\underline{\theta}}\rangle| \leq \sqrt{\frac{\widehat{\sigma}^2}{m} + \left(\mathrm{se}\langle h\langle x, \widehat{\underline{\theta}}\rangle\rangle\right)^2} \cdot q_{1-\alpha/2}^{t_{n-p}} \right\}.$$

The R function `invest(...)` from the package `investr` uses this approach in its default version.

Example e Cress - Calibration. We calculate the calibration value and the 95% calibration interval for an observed weight value of 650:

```
> require(investr)
> invest(N.nls, y0=650, interval="inversion") # default
  estimate      lower      upper
 0.7718271  0.4252540  1.1922483
```

If there are several (true) replicates of the measurements of y_o, you can enter it as follows:

```
> invest(N.nls, y0=c(c(309, 221, 370)), interval="inversion")
 estimate     lower     upper
 2.626680  2.034845  3.529605
```

As a result the length of the calibration interval is shorter as expected.

f Alternatively, one can calculate the calibration intreval by a bootstrap approach, either a parametric or a nonparametric one. In general, `nsim` (called B in this notes) should be as large as reasonably possible (say, `nsim = 4999`).

```
> I.calIB1 <- invest(N.nls, y0=650, interval="percentile",
                     boot.type="parametric", nsim=300, seed=101)
> I.calIB1   # print bootstrap summary
  estimate      lower      upper         se        bias
 0.7718271  0.4250045  1.1521891  0.1928681  -0.0003627
> plot(I.calIB1) # plot results, but not shown in this notes
```

```
> I.calIB2 <- invest(N.nls, y0=650, interval="percentile",
                     boot.type="parametric", nsim=300, seed=101)
> I.calIB2
  estimate      lower      upper         se        bias
 0.7718271  0.4062382  1.1196747  0.1792624  -0.0028426
```

g In this chapter, only one of many possibilities for determining a calibration interval was presented. Some details to Bootstrap and Likelihood Ratio Calibration Intervals can be found in Huet, Bouvier, Poursat and Jolivet (2010, p. 139ff).

4 Closing Comments

a **Reason for the Difficulty in the Biochemical Oxygen Demand Example.** Why did we have so many problems with the Biochemical Oxygen Demand example? Let us have a look at Figure 1.1.e and remind ourselves that the parameter θ_1 represents the expected oxygen demand for infinite incubation time, so it is clear that it is difficult to estimate θ_1, because the horizontal asymptote is badly determined by the given data. If we had more observations with longer incubation times, we could avoid the difficulties with the quality of the confidence intervals of θ.

Also in nonlinear models, a good (statistical) **experimental design** is essential. The information content of the data is determined through the choice of the experimental conditions and no (statistical) procedure can deliver information that is not contained in the data.

b **Robust Fitting.** The original data set of the example 'Cellulose Membrane' is presented in the scatter plot of Figure 4.0.b. It shows clearly that there are outliers with respect to a sigmoid curve. Since many of such experiments had to be analysed, removing the outliers case by case by hand is cumbersome. Robust methods as they have been introduced in Ruckstuhl (2024) is very timesafing and even more objective.

The function `nlrob(...)` of the R package `robustbase` implements several methods (see help page of `nlrob(...)` for details). Two of the methods are introduced in Ruckstuhl (2024) in case of the multiple linear regression model: The M-estimator (`method="M"`) and the MM-estimator (`method="MM"`). The MM-estimator has as initial estimator an S-estimator. In Figure 4.0.b (right) the fit of a robust M-estimator with bisquared ψ function ist compared with the result of a least-squares estimator. The function call is given in R Output 4.0.b.

```
library(MASS)
library(robustbase)
Mem.rFit0A <- nlrob(delta ~ (T1 + T2*10^(T3+T4*pH))/(10^(T3+T4*pH)+1),
                    I.nmr, start=list(T1=163.7, T2=160, T3=3.3, T4=-0.6),
                    psi=function(u, c, derive) psi.bisquare(u, c=4, deriv=0))
```

R-Output 4.0.b: R code to fit a robust M-estimator with bisquared ψ function to the Membrane Separation Technology dataset.

c **Correlated Errors.** Here we always assumed that the errors E_i are independent. Like in linear regression analysis, nonlinear regression models can also be extended to handle **correlated errors** and **random effects**.

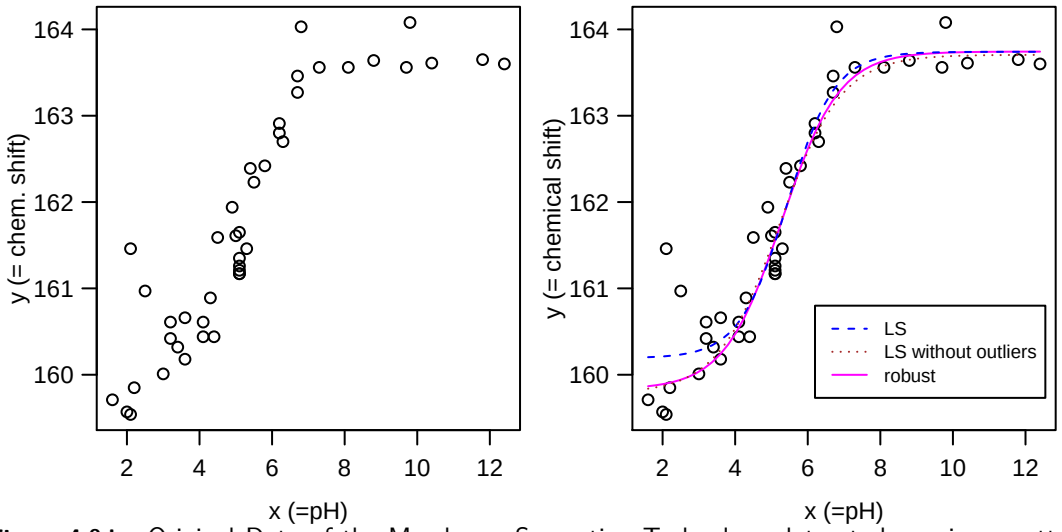

Figure 4.0.b.: Original Data of the Membrane Separation Technology dataset shown in a scatter plot (left). On the right-hand side, fits of using different estimators are shown: Least-squares (LS), Least-squares without outliers as in the lecture, and a robust M-estimator with bisquared ψ function.

d Statistics Programs. Today most statistics packages contain a procedure that can calculate asymptotic confidence intervals for the parameters. In principle it is then possible to calculate "t-profiles" and profile traces because they are also based on the fitting of nonlinear models (on a reduced set of parameters).

In R, the function `nls` is available, that is based on the work of Bates and Watts (1988). The "library" `nlme` contains R functions that can fit nonlinear regression models with correlated errors (`gnls`) and random effects (`nlme`). These implementations are based on the book "Mixed Effects Models in S and S-Plus" from Pinheiro and Bates (2000).

e Literature Notes. These notes are mainly based on the book of Bates and Watts (1988). A mathematical discussion about the statistical and numerical methods in nonlinear regression can be found in Seber and Wild (1989). The book of Ratkowsky (1989) contains many nonlinear functions $h\langle\cdot\rangle$ that are primarily used in biological applications.

A short introduction to this topic using the statistics program R can be found in Venables and Ripley (2002) and in Ritz and Streibig (2008).

More details on the application of the bootstrap method in nonlinear regression modelling can be found, e.g., in Ritz and Streibig (2008) and Huet et al. (2010). In the latter there is also a discussion about non-constant variance (i.e., heteroscedastic) models. It is also worth taking a look at the book of Carroll and Ruppert (1988).

A Appendix

A.1 The Gauss-Newton Method

a **The Optimization Problem..** To determine the least squares estimator we must minimize the squared sum

$$S(\underline{\theta}) := \sum_{i=1}^{n} (y_i - \eta_i\langle\underline{\theta}\rangle)^2.$$

Unfortunately this can not be carried out explicitly like in linear regression. With local linear approximations of the regression function, however, the difficulties can be overcome.

b **Solution Procedure..** The procedure is set out in four steps. We consider the $(j+1)$-th iteration in the procedure.To simplify, we assume that the regression function $h\langle\rangle$ has only one unknown parameter. The solution from the previous iteration is denoted by $\theta^{(j)}$. $\theta^{(0)}$ is then the notation for the starting value.

1. Approximation: The regression function $h\langle x_i, \theta\rangle = \eta\langle\theta\rangle_i$ with the one dimensional parameter θ at the point $\theta^{(j)}$ is approximated by a line:

$$\eta_i\langle\theta\rangle \approx \eta_i\langle\theta^{(j)}\rangle + \left.\frac{\partial\eta_i\langle\theta\rangle}{\partial\theta}\right|_{\theta^{(j)}} (\theta - \theta^{(j)}) = \eta_i\langle\theta^{(j)}\rangle + a_i\langle\theta^{(j)}\rangle(\theta - \theta^{(j)}) \,.$$

* For a multidimensional $\underline{\theta}$ with help of a hyper plane it is approximated:

$$h\langle x_i, \underline{\theta}\rangle = \eta_i\langle\underline{\theta}\rangle \approx \eta_i\langle\underline{\theta}^{(j)}\rangle + a_i^{(1)}\langle\underline{\theta}^{(j)}\rangle(\theta_1 - \theta_1^{(j)}) + \ \ldots\ + a_i^{(p)}\langle\underline{\theta}^{(j)}\rangle(\theta_p - \theta_p^{(j)}),$$

where

$$a_i^{(k)}\langle\underline{\theta}\rangle := \frac{\partial h\langle x_i, \underline{\theta}\rangle}{\partial\theta_k}, \quad k = 1, \ldots, p.$$

With vectors and matrices the above equation can be written as

$$\underline{\eta}\langle\underline{\theta}\rangle \approx \underline{\eta}\langle\underline{\theta}^{(j)}\rangle + \boldsymbol{A}^{(j)}(\underline{\theta} - \underline{\theta}^{(j)}) \,.$$

Here the $(n \times p)$ derivative matrix $\boldsymbol{A}^{(j)}$ consists of the j-th iteration of the elements $\{a_{ik} = a_i^{(k)}\langle\underline{\theta}\rangle\}$ at the point $\underline{\theta} = \underline{\theta}^{(j)}$.

2. Local Linear Model: We now assume that the approximation in the 1st step holds exactly for the true model. With this we get for the residuals

$$r_i^{(j+1)} = y_i - \{\eta\langle\theta^{(j)}\rangle_i + a_i\langle\theta^{(j)}\rangle(\theta - \theta^{(j)})\} = \widetilde{y}_i^{(j+1)} - a_i\langle\theta^{(j)}\rangle\beta^{(j)}$$

with $\widetilde{y}_i^{(j+1)} := y_i - \eta\langle\theta^{(j)}\rangle$ and $\beta^{(j+1)} := \theta - \theta^{(j)}$.

* For a multidimensional $\underline{\theta}$ it holds that:

$$\underline{r}^{(j+1)} = \underline{y} - \{\underline{\eta}(\underline{\theta}^{(j)}) + \boldsymbol{A}^{(j)}(\underline{\theta} - \underline{\theta}^{(j)})\} = \widetilde{\mathbf{y}}^{(j+1)} - \boldsymbol{A}^{(j)}\underline{\beta}^{(j+1)}$$

with $\tilde{\mathbf{y}}^{(j+1)} := \mathbf{y} - \underline{\eta}(\underline{\theta}^{(j)})$ and $\underline{\beta}^{(j+1)} := \underline{\theta} - \underline{\theta}^{(j)}$.

3. **Least Square Estimation in the Locally Linear Model:** To find the best-fitting $\widehat{\beta}^{(j+1)}$ for the data, we minimize the sum of squared residuals: $\sum_{i=1}^{n}(r_i^{(j+1)})^2$. This is the usual linear least squares problem with $\tilde{y}_i^{(j+1)} \equiv y_i$, $a_i\langle\theta^{(j)}\rangle \equiv x_i$ and $\beta^{(j+1)} \equiv \beta$ (The line goes through the origin). The solution to this problem, $\widehat{\beta}^{(j+1)}$, gives the best solution on the approximated line.

   * To find the best-fitting $\widehat{\underline{\beta}}^{(j+1)}$ in the multidimensional case, we minimize the sum of squared residuals: $\|r^{(j+1)}\|^2$. This is the usual linear least squares problem with $\tilde{\mathbf{y}}^{(j+1)} \equiv \mathbf{y}$, $\boldsymbol{A}^{(j)} \equiv X$ and $\underline{\beta}^{(j+1)} \equiv \underline{\beta}$. The solution to this problem gives a $\widehat{\underline{\beta}}^{(j+1)}$, so that the point

   $$\underline{\eta}\langle\underline{\theta}^{(j+1)}\rangle \quad \text{with } \underline{\theta}^{(j+1)} = \underline{\theta}^{(j)} + \widehat{\underline{\beta}}^{(j)}$$

   lies nearer to $\mathbf{y}$ than $\underline{\eta}\langle\underline{\theta}^{(j)}\rangle$.

4. **Iteration:** Now with $\theta^{(j+1)} = \theta^{(j)} + \widehat{\beta}^{(j)}$ we return to step 1 and repeat steps 1, 2, and 3 until this procedure converges. The converged solution minimizes $S\langle\theta\rangle$ and thus corresponds to the desired estimation value $\widehat{\theta}$.

c Further Details.. This minimization procedure that is known as the Gauss-Newton method, can be further refined. However, there are also other minimization methods available. It must be addressed in more detail what "converged" should mean and how the convergence can be achieved. The details about these technical questions can be read in, e.g. Bates and Watts (1988). For us it is of primary importance to see that iterative procedures must be applied to solve the optimization problems in 1.2.a.

Bibliography

Bates, D. M. and Watts, D. G. (1988). *Nonlinear Regression Analysis & Its Applications*, John Wiley & Sons.

Carroll, R. and Ruppert, D. (1988). *Transformation and Weighting in Regression*, Wiley, New York.

Daniel, C. and Wood, F. S. (1980). *Fitting Equations to Data*, John Wiley & Sons, New York.

Huet, S., Bouvier, A., Poursat, M.-A. and Jolivet, E. (2010). *Statistical Tools for Nonlinear Regression: A Practical Guide with S-Plus and R Examples*, 2nd edn, Springer-Verlag, New York.

Pinheiro, J. C. and Bates, D. M. (2000). *Mixed-Effects Models in S and S-PLUS*, Statistics and Computing, Springer.

Rapold-Nydegger, I. (1994). *Untersuchungen zum Diffusionsverhalten von Anionen in carboxylierten Cellulosemembranen*, PhD thesis, ETH Zurich.

Ratkowsky, D. A. (1985). A statistically suitable general formulation for modelling catalytic chemical reactions, *Chemical Engineering Science* **40**(9): 1623–1628.

Ratkowsky, D. A. (1989). *Handbook of Nonlinear Regression Models*, Marcel Dekker, New York.

Ritz, C. and Streibig, J. C. (2008). *Nonlinear Regression with R*, Use R!, Springer Verlag.

Ruckstuhl, A. (2024). Robust fitting of parametric models based on M-estimation. Lecture Note in WBL Angewandte Statistik, ETHZ.

Seber, G. and Wild, C. (1989). *Nonlinear regression*, Wiley, New York.

Stahel, W. A. (2007). *Statistische Datenanalyse: Eine Einführung für Naturwissenschaftler*, 5. Auflage edn, Vieweg+Teubner Verlag.

Venables, W. N. and Ripley, B. (2002). *Modern Applied Statistics with S*, Statistics and Computing, fourth edn, Springer-Verlag, New York.