Zurich University
of Applied Sciences

School of
Engineering

IDP Institute of
Data Analysis and
Process Design

WBL Statistik 2024 — Nonlinear Regression

A Powerful Tool With Considerable Complexity

Half-Day 3: Bootstrap, Prediction, Calibration

Andreas Ruckstuhl
Institut für Datenanalyse und Prozessdesign
Zürcher Hochschule für Angewandte Wissenschaften

Outline:

Half-Day 1 Estimation and Standard Inference

- The Nonlinear Regression Model
- Iterative Estimation - Model Fitting
- Inference Based on Linear Approximations

Half-Day 2 Improved Inference and Visualisation

- Likelihood Based Inference
- Profile t Plot and Profile Traces
- Parameter Transformations

Half-Day 3 Bootstrap, Prediction and Calibration

- Bootstrap
- Prediction
- Calibration

Outlook

2.4 Bootstrap

An alternative to profile confidence intervals is to apply the **bootstrap** method.

- It is a resampling method
- It does not rely on a linear approximation.
- The error distibution may not be Gaussian

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.*

The Steps in Bootstrap

- The simplest bootstrap method involves
  - taking the original dataset of n observations and
  - sampling from it to form a new sample (called **bootstrap sample**) that is also of size n.
    The sampling is done with *replacement*
    ☞ 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),
  - For each of these bootstrap samples we compute its parameter estimates (each of these are called **bootstrap estimates**).
  - The bootstrap estimates of one parameter may be summarized by a histogram.
  - 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.

Nonparametric Bootstrap

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.
  - 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**.

The parametric bootstrap:

We assume that

- the residuals are Gaussian distributed and
- hence we resample from $\mathcal{N}\left\langle 0, \widehat{\sigma}^2 \right\rangle$
  $\widehat{\sigma}^2$ is the variance estimate from the nonlinear regression fit

Bootstrap with `nlsBoot`

To apply bootstrap with nonlinear regression estimation

- use, e.g., the R function `nlsBoot()` in the package `nlstools`
- `nlsBoot()` is based on a nonparametric bootstrap approach, where the mean centred residuals are bootstrapped

  "mean centred" because the residuals may have a non-zero mean with a *nonlinear* regression model

95% **bootstrap percentile confidence interval**
It is based on quantiles of 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.
- 95% **bootstrap percentile confidence interval**: take the 25th value and the 975th value among the 1'000 ordered estimates.

The bootstrap confidence interval has the advantage of lying entirely within the range of plausible parameter values.

Biochemical Oxygen Demand (cont'd)

95% confidence interval:

	Wald		Profile Likelihood		Bootstrap	
	2.5%	97.5%	2.5%	97.5%	2.5%	97.5%
Th1	12.2127987	26.072347	14.0845447	38.482510	16.0755863	25.699375
Th2	-0.0327549	1.094938	0.1356242	1.810125	0.2815905	1.088552

- The results of the three approaches differ considerably.
- We know from the likelihood profiles that the linear approximation approach is poor in this example.

R code to generate the above results:

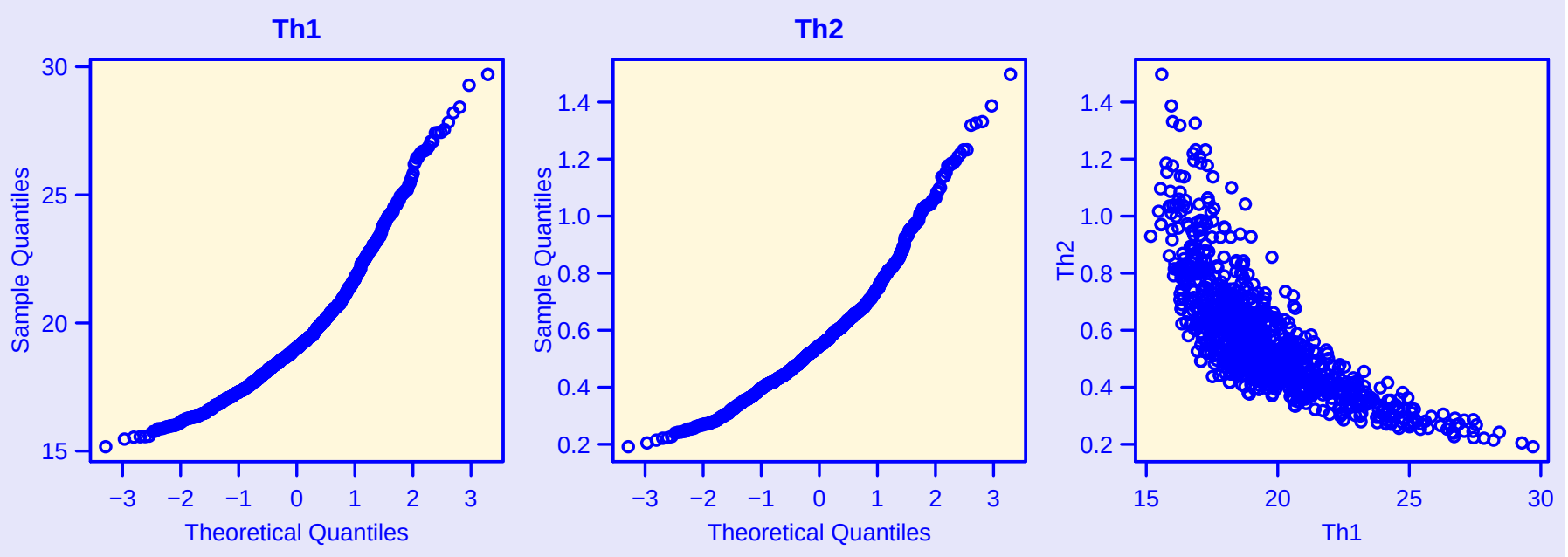
```
## Wald:
> require(nlstools)
> confint2(D.bod.nls, method="asymptotic")   # using t-quantiles
## 95% confidence interval see above

## Profile Likelihood:
> confint(D.bod.nls)
Waiting for profiling to be done...
## 95% confidence interval see above

## Bootstrap:
> library(nlstools)
> D.bod.nls.Boot <- nlsBoot(D.bod.nls)
> summary(D.bod.nls.Boot)
## 95% confidence interval see above
```

Biochemical Oxygen Demand (cont'd)

- The marginal bootstrap distributions and their joint distribution show:
  The linear approximation (i.e., Wald's Gaussian approximation) is unsuitable

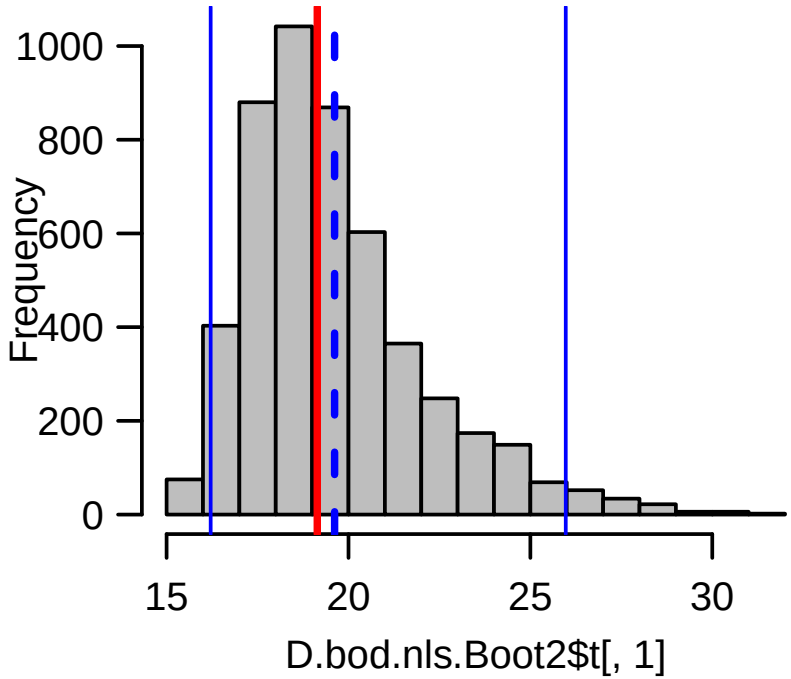


- The difference between the results of the profile likelihood approach and the result of the bootstrap approach is rather nebulous (at first).
  Note that the size of the dataset is rather small.

Bias-Corrected Bootstrap

Histogram of the marginal bootstrap distributions of θ_1

- vertical blue lines: 0.025 and 0.975 quantile
- vertical red line: $\widehat{\theta}_1$
- vertical blue dashed line: mean of θ_1^*



The bootstrap percentile confidence interval is reliable if

- the bias $\left|\widehat{\theta}_1 - \bar{\theta}_1^*\right|$ is small
- the bootstrap distribution is symmetric

If not, use the bias-corrected accelerated (bca) bootstrap interval.

Bias-Corrected Bootstrap with R

```
> library(boot)
> f.bod <- function(rs, ind){
   bsY <- fitted(D.bod.nls) + rs[ind]
   coef(nls(bsY ~ Th1*(1-exp(-Th2*days)), data=D.bod,
            start=coef(D.bod.nls)))
  }
> h.rs <- scale(resid(D.bod.nls), scale=FALSE)    ## mean centred residuals

>   ## bootstrapping
> set.seed(seed=117)
> D.bod.nls.Boot2 <- boot(h.rs, f.bod, R=4999, stype="i" )

>   ## bias-corrected accelerated (bca) bootstrap interval
> rbind(boot.ci(D.bod.nls.Boot2, conf=0.95, type="bca", index=1)$bca[4:5],
        boot.ci(D.bod.nls.Boot2, conf=0.95, type="bca", index=2)$bca[4:5])
             [,1]        [,2]
[1,]   16.5859883   27.947124
[2,]    0.2970484    1.233983
```

3 Prediction and Calibration

Example Cress

The concentration of a agro-chemical substance in soil samples is studied by the growth characteristics of cress (Nasturtium).

- 6 measurements of the response variable Y (weight of cress) were made on each of 7 soil samples with predetermined (or measured with the largest possible precision) concentrations x.

  ☞ We assume that the x-values have no measurement error.

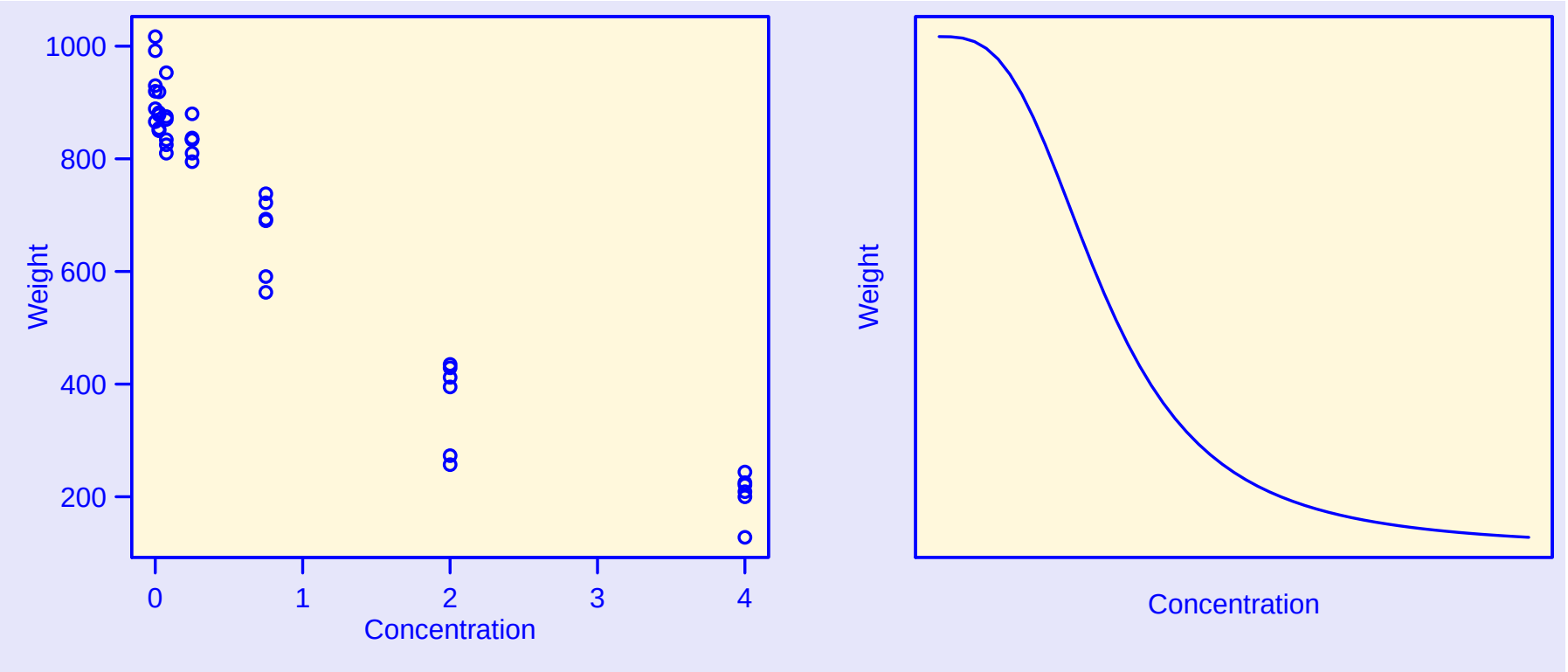
- The variable of interest is the weight of the cress per unit area after 3 weeks.

- A "logit-log" model is used to describe the relationship between concentration and weight:

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

Example Cress

Illustration of the dataset and the function $h\langle\cdot\rangle$:



Left: Scatter plot of the data.
Right: A typical form of the regression function to be used.

Example Cress

- Fitting of the nonlinear regression model:

```
> library(investr)   # contains dataset 'nasturtium'
> 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=nasturtium,
             start=list(Th1=900, Th2=-1, Th3=1))
```

Next steps:

- **calibration**
  Concentration known: ☞ x_i, $i = 1, \ldots, n$
  Weight of harvested cress is measured: ☞ $Y = y_i$, $i = 1, \ldots, n$
- **"Prediction"**
  which weight values will be observed at a concentration of e.g. $x_0 = 3$?
- **"Inversion"**
  Weight of the harvested cress is known (measured).
  How high is the concentration x when e.g. a weight values of 600 (y) is observed?

Calibration Versus Prediction

- First, run an experiment to obtain a learning sample
- Fit the regression function $h\langle \cdot, \underline{\theta} \rangle$ with the unkown parameters $\underline{\theta}$ to the data of the leaning sample ☞ $\widehat{\underline{\theta}}$

- Actually, we are interested in a new pair of observations $(x_o,\ Y_o)$
  - Prediction Problem: If x_o is observed, predict Y_o: $\widehat{Y}_o = h\left\langle x_o, \widehat{\underline{\theta}} \right\rangle$
  - Calibration Problem: If $Y_o = y_o$ is observed, predict x_o: $\widehat{x}_o = h^{-1}\left\langle y_o, \widehat{\underline{\theta}} \right\rangle$

3.1 Prediction Interval

Along the lines of linear regression, an **approximate** $(1 - \alpha/2)$ **prediction interval** can be specified by

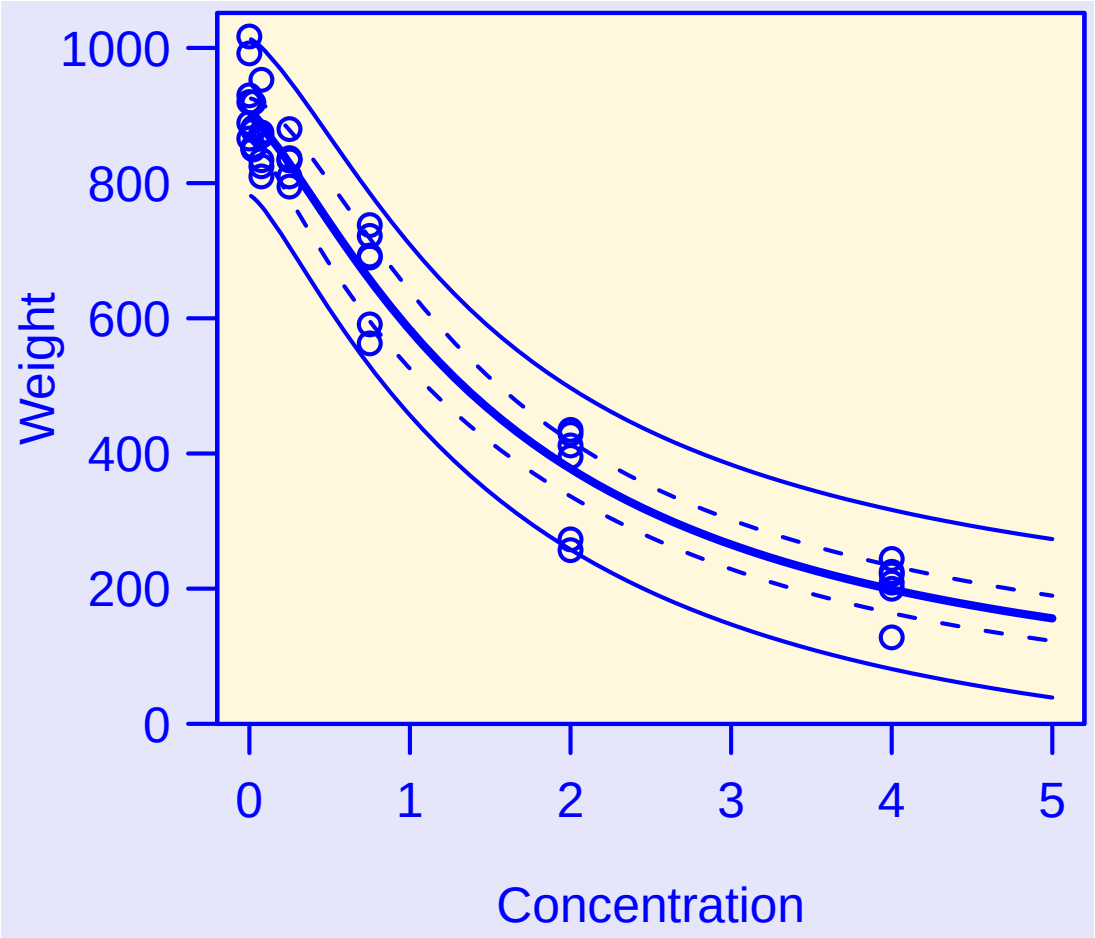
$$\widehat{\eta}_o \pm q_{1-\alpha/2}^{t_{n-p}} \sqrt{\widehat{\sigma}^2 + \mathrm{se}^2 \langle \widehat{\eta}_o \rangle} \quad \text{with } \widehat{\eta}_o := h\left\langle x_o, \underline{\widehat{\theta}} \right\rangle .$$

The variability in the prediction contains the variability in the error term E_i of the regression model as well as the variability of the parameter estimation in $\widehat{\eta}_o$ (cf. Lecture Notes 3.1.c).

Hence

- the prediction interval is wider than the confidence interval for the expectation $h\langle x_o, \underline{\theta} \rangle$.
- Even if the sample size n is very large, i.e., $\mathrm{se}^2 \langle \widehat{\eta}_o \rangle \to 0$, the variability will never drop below the variability of the error term $\widehat{\sigma}^2$.

Example Cress: Prediction Band (versus Confidence Band)



95% confidence band (dashed line) and 95% predition band (solid line) for the fitted regression curve (thick solid line).

Example Cress: Computation of the Confidence/Prediction Band

Note: The R function `predict.nls()` ignores the arguments `se.fit` and `interval`

Alternative:

```
## Fitting: see SLIDE 13

> require(investr)

## Setting up the graphic:
> N.new <- data.frame(conc=seq(0, 5, by=0.05))
> plot(weight ~ conc, data=N.new, type="l", col="blue", las=1,
       ylim=c(0,1000))
> points(weight ~ conc, data=D.kresse)

## Confidence band:
> N.nlsCI <- predFit(N.nls, newdata=N.new, interval="confidence")
> lines(N.new$conc, N.nlsCI[,"lwr"], col="orange")
> lines(N.new$conc, N.nlsCI[,"upr"], col="orange")

## Adding prediction band
> N.nlsPI <- predFit(N.nls, newdata=N.new, interval="prediction")
> lines(N.new$conc, N.nlsPI[,"lwr"], col="orange", lwd=2)
> lines(N.new$conc, N.nlsPI[,"upr"], col="orange", lwd=2)
```

3.2 Calibration Interval

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.

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.

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, this procedure is only correct if $h\langle\cdot\rangle$ is monotone increasing or decreasing.

Approximate Calibration Interval

Accuracy of $\widehat{x}_o$?

To answer this question a confidence interval is formed for x_o

we can view x_o as a parameter.

It is based on a hypothesis test (as always):

- Null hypothesis: $x_o = x_H$ Alternative: $x_o \neq x_H$
- Test statistic: $Y_o = y_o$
- Acceptance interval is given by the prediction interval

$$h\left\langle x_o, \underline{\widehat{\theta}}\right\rangle \pm q_{0.975}^{t_{n-p}} \sqrt{\widehat{\sigma}^2 + \mathrm{se}^2\left\langle \widehat{\eta}_o\right\rangle}\,.$$

Hence, the interval defined by

$$\left\{x^* \,:\, |y_0 - h\left\langle x^*, \underline{\widehat{\theta}}\right\rangle| \leq \sqrt{\widehat{\sigma}^2 + \mathrm{se}^2\left\langle \widehat{\eta}_o\right\rangle}\; q_{1-\alpha/2}^{t_{n-p}}\right\}.$$

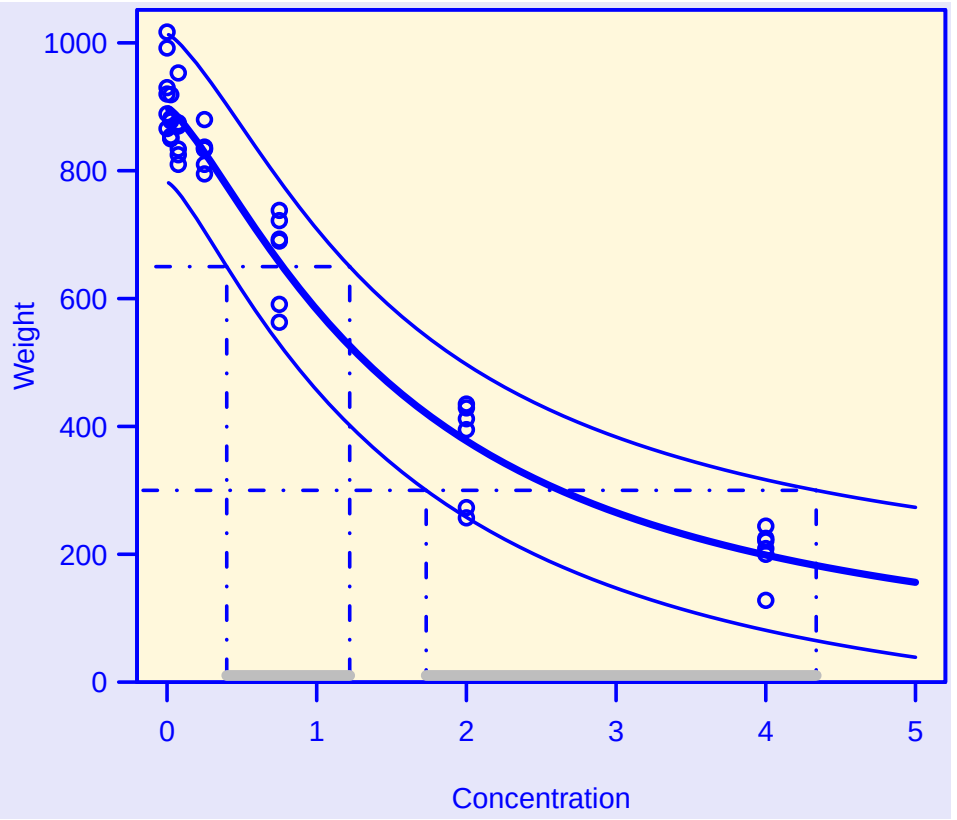
is the wanted confidence interval (also called **calibration interval**) for x_o.

Example Cress:

Visualization of the construction of calibration intervals

95% predition band (solid line) for the fitted regression curve (thick solid line).

The resulting calibration intervals are [0.4, 1.22] and [1.73, 4.34] for measured weights of $y_0 = 650$ and $y_0 = 350$, respectively.



Example Cress: Computing a Calibration Intervall

- Let us put the idea presented for the calculation of the calibration interval into practice.
- The calculation is based on the prediction intervals calculated with the **Wald approach**.

Suppose we have measured a value of *weight* = 650
How high is the concentration?

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

Suppose we have three replicated measures for the same lot (with a mean value of 650):

```
> invest(N.nls, y0=c(659, 571, 720), interval="inversion") ## default
   estimate       lower       upper
 0.7718271   0.5482369   1.0237440
```

As expected, the calibration interval is shorter.

- The calculation of the calibration interval is based on the prediction intervals using a **bca bootstrap approach**.

```
> invest(N.nls, y0=650, invest(N.nls, y0=650, interval="percentile",
        boot.type="parametric", nsim=300, seed=101)
  estimate       lower       upper          se         bias
 0.7718271   0.4250045   1.1521891   0.1928681   -0.0003627
```

Alternatively, we can use a nonparametric bootstrap:

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

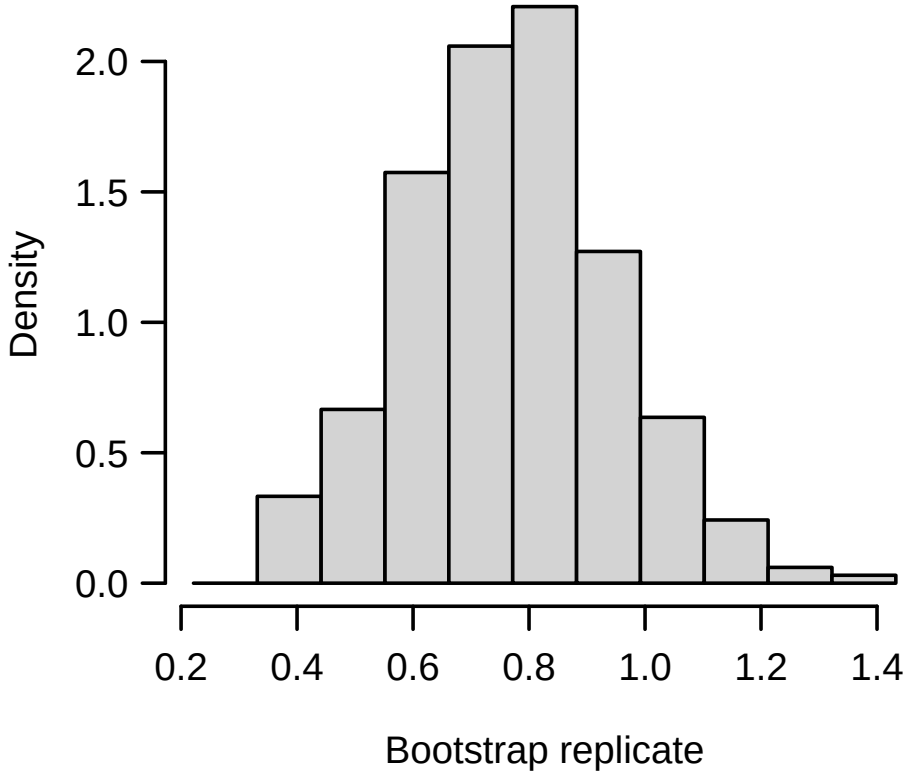
The difference between these two bootstrap approaches is small in this example, and also the differences to the Wald approach ist small.

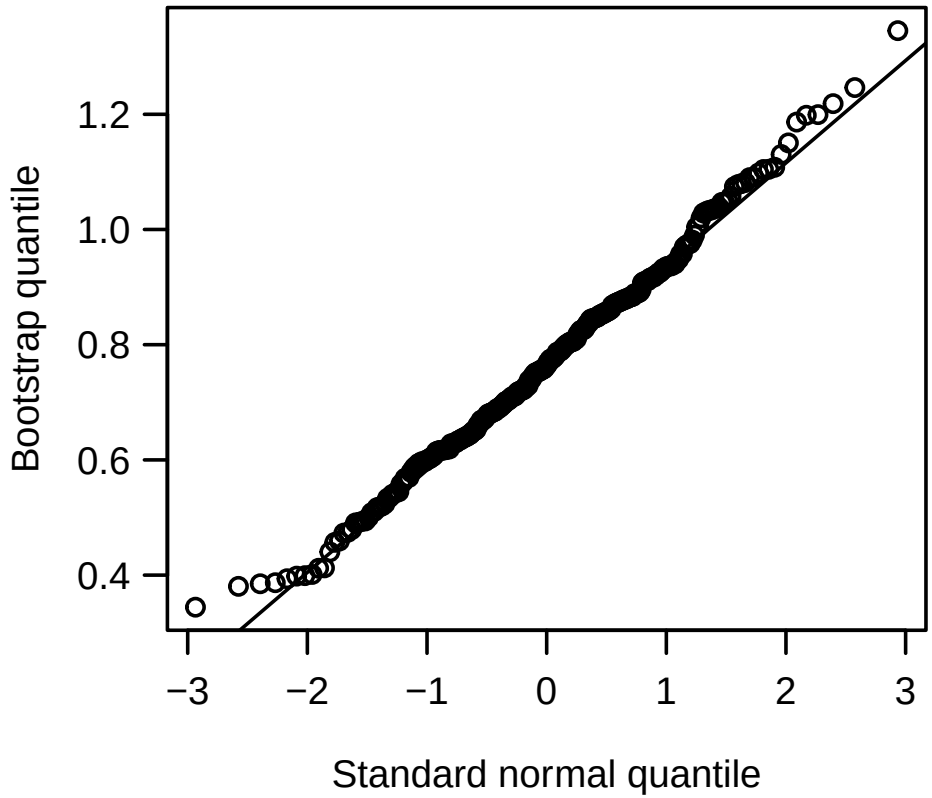
Suppose we have three replicated measures for the same lot

```
> invest(N.nls, y0=650, interval="percentile", boot.type="nonparametric",
        nsim=300, seed=101)
  estimate       lower       upper          se         bias
 0.7718271   0.5689410   1.0000296   0.1107456   -0.0049647
```

The `plot` of a "`bootCal`" objects produces plots for the bootstrap replicates of the inverse estimate.

```
> N.calIB2 <- invest(N.nls, y0=650, interval="percentile",
                     boot.type="nonparametric", nsim=300, seed=101)
> plot(I.calIB2)
```





Take Home Message Half-Day 3

- **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.

- The concepts of prediction intervals and calibration intervals were discussed

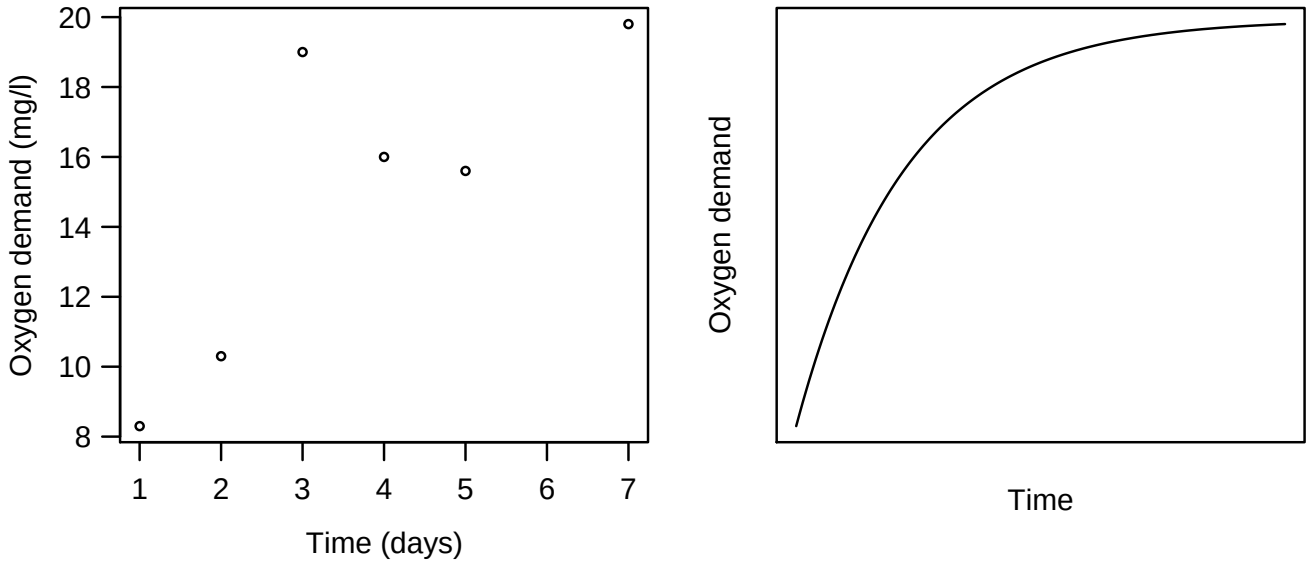
- Often nonlinear regression models are used for **Calibration**.

  A simple method of constructing **calibration intervals** is to "invert" the prediction interval.

8 Outlook

8.A Example Biochemical Oxygen Demand (BOD)

Troubles are caused by a too insufficient design of experiment (DoE)



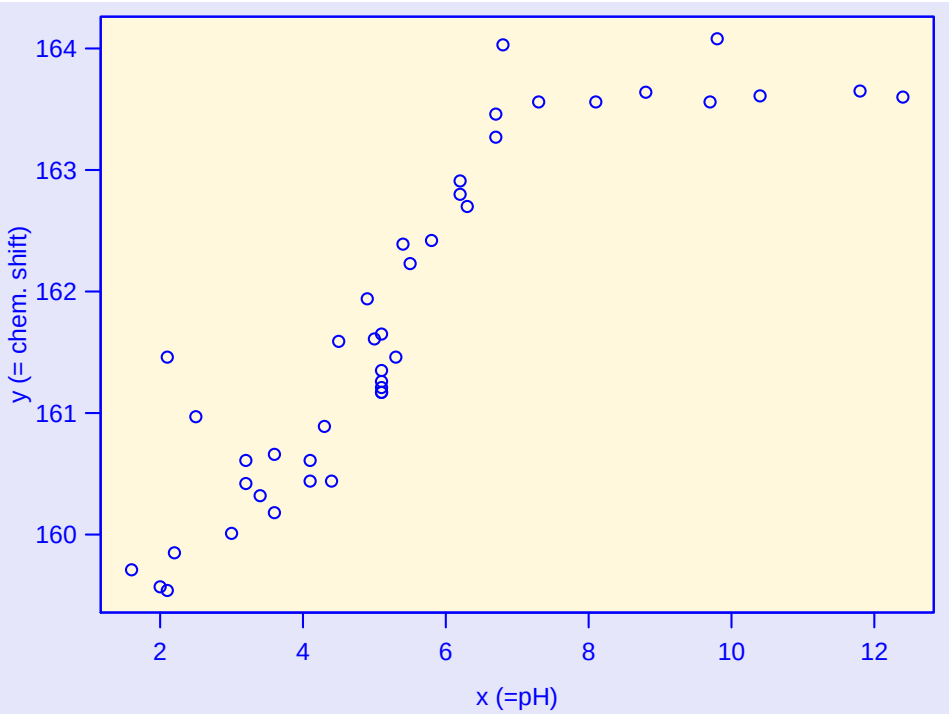
$$Y_i = \theta_1 \cdot \left(1 - e^{\theta_2 \cdot x_i}\right) + E_i \qquad \text{mit} \;\; E_i \; i.i.d. \sim \mathcal{N}\left\langle 0, \sigma^2 \right\rangle ,$$

where Y is the biochemical oxygen demand (BOD) $[mg/\ell]$ and

x the incubation time [days]

8.B Robust Fitting (I)

How the original data of the example 'Cellulose Membrane' shows: Robust methods are also need in nonlinear regression

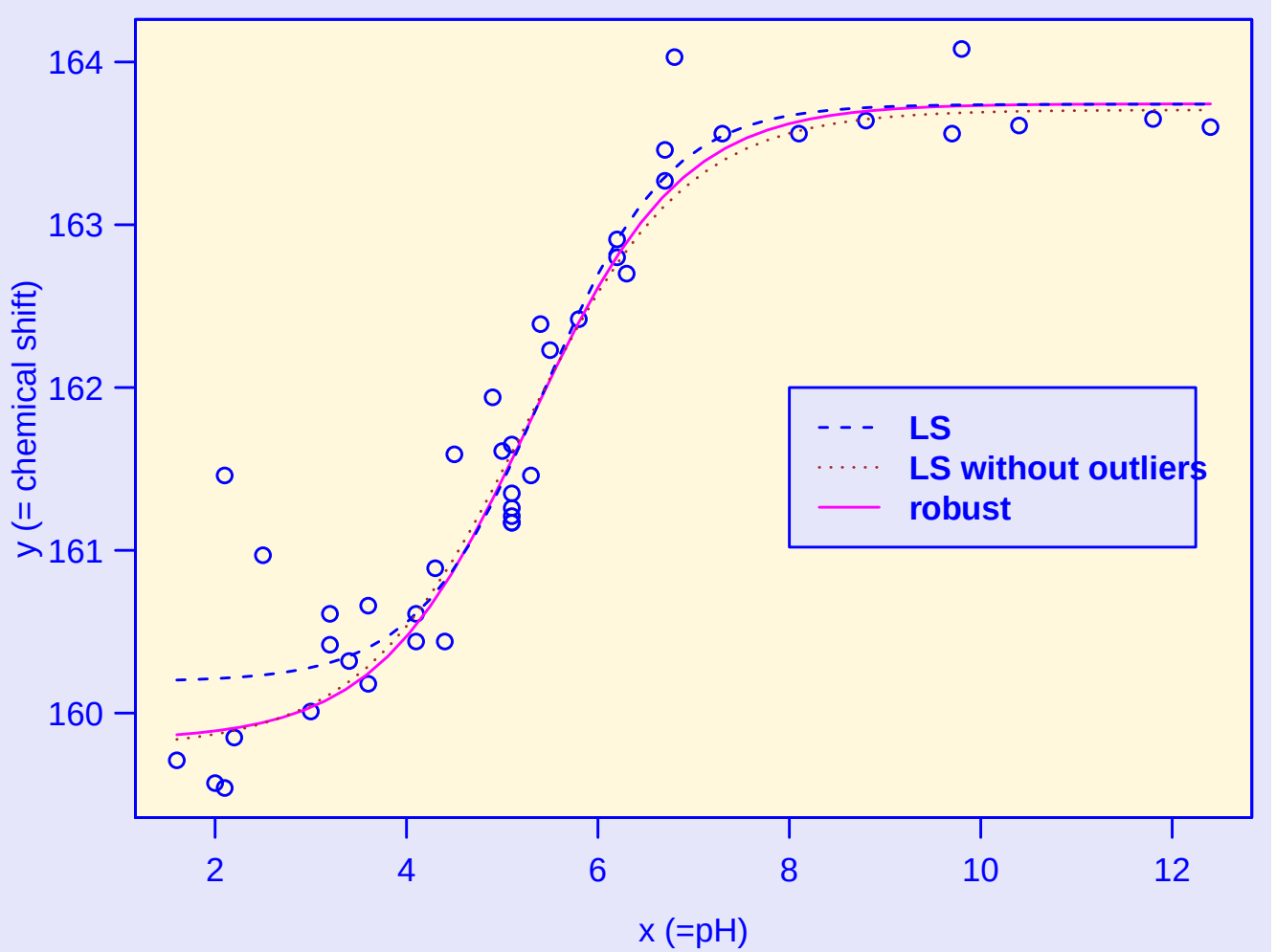


R implementation: `nlrob()` in the R package `robustbase` with

- `method="M"`: Computes an M-estimator using `nls(...)`
- `method="MM"`: Computes an MM-estimator, starting from `init`, either `"S"` or `"lts"`
- `method=...`: see help page of `nlrob()`

8.B Robust Fitting (II)

Fitted function using LS (`nls()`), LS without outliers (as in this lecture) and robust method (`nlrob()`).



8.D Correlated Errors

(cf. `gnls(...)` in the R package `nlme`)

8.E Random Effets - Nonlinear Mixed Effects-Models

(cf. `nlme(...)` in the package `nlme` – very challenging!)
Literature: Book "Mixed-Effects Models in S and S-Plus" by Pinheiro and Bates (2000).

8.F Software

Fitting and Wald-type inference is the standard implementation almost everywhere
Bootstrap and Profiling are not (yet) standard approaches apart from R

8.G Literature

- I recommend the book of Bates and Watts, "Nonlinear Regression Analysis & Its Application" (1988)
- A short introduction to this topic using the statistics program R can be found in
  - Venables and Ripley, "Modern Applied Statistics with S" (2002)
  - Ritz and Streibig, "Nonlinear Regression with R" (2008)
- Huet, Bouvier, Poursat and Jolivet (2010) focus in their book "Statistical Tools for Nonlinear Regression: A Practical Guide with S-Plus and R Examples" on application of the bootstrap method in nonlinear regression modelling.