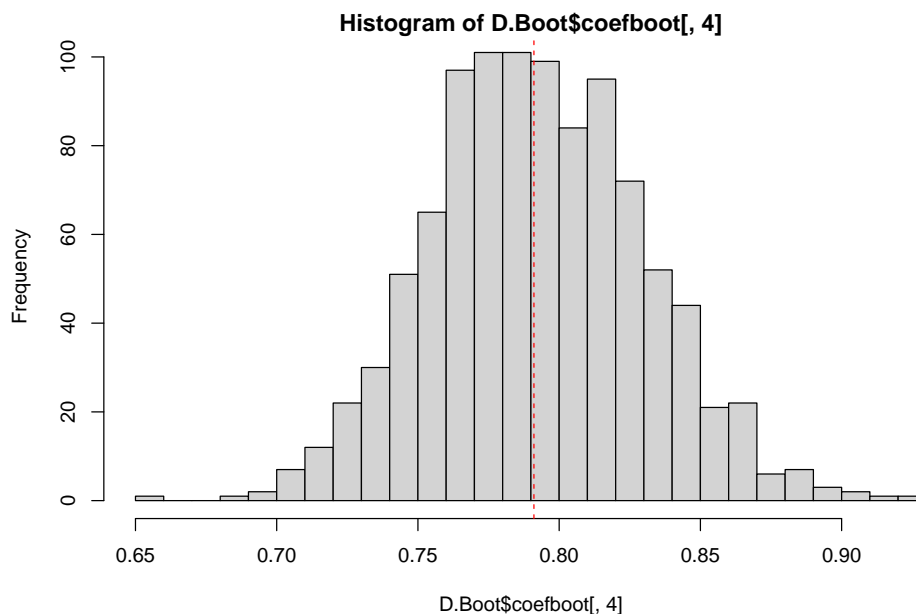


## Solution to Series 6

```
1. a) > d.body <- read.table("http://stat.ethz.ch/Teaching/Datasets/cas-das/body.dat",
                             header = TRUE)
> r.nls <- nls(weight ~ theta1 + theta2 * (1 - exp(theta3 * (theta4 * sourceA + sourceB))),
               data = d.body,
               start = list(theta1 = 640, theta2 = 160, theta3 = -10.6, theta4 = 0.76),
               control = nls.control(maxit = 1000))
```

```
b) > library(nlstools)
> set.seed(1234)
> D.Boot <- nlsBoot(r.nls, niter = 999)
> hist(D.Boot$coefboot[, 4], breaks = 20)
> abline(v = coefficients(r.nls)[4], lty = 2, col = "red")
> max(D.Boot$coefboot[, 4])
```

```
[1] 0.923
```



The parameter  $\theta_4$  is clearly significantly different from 1 since all the bootstrap estimates for  $\theta_4$  are smaller than 1 (and hence also the 25 largest ones (2.5% of 1000 = 25)).

```
c) > summary(D.Boot)
```

```
-----
```

Bootstrap statistics

	Estimate	Std. error
theta1	638.715	5.1290
theta2	176.701	4.8705
theta3	-6.373	0.6172
theta4	0.792	0.0378

```
-----
```

Median of bootstrap estimates and percentile confidence intervals

	Median	2.5%	97.5%
theta1	639.081	627.406	648.132
theta2	176.846	167.537	186.427
theta3	-6.346	-7.677	-5.245
theta4	0.791	0.721	0.864

The 95%-bootstrap confidence-interval for  $\theta_4$  is  $[0.72, 0.87]$ . It is shorter than the confidence-intervals obtained in Series 5.

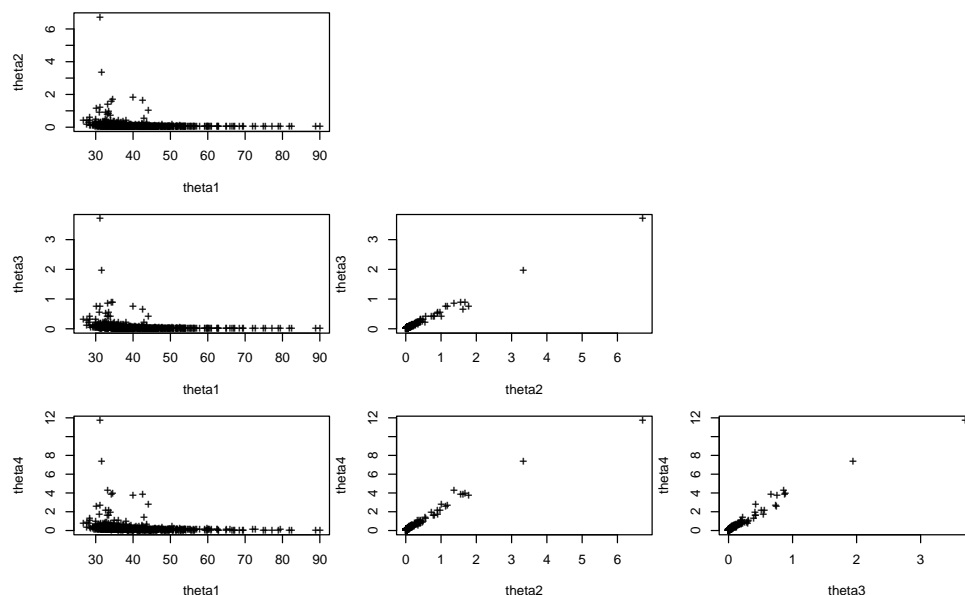
```
d) > d.isomer <- read.table("http://stat.ethz.ch/Teaching/Datasets/cas-das/isomer.dat",
                             header = TRUE)
> r.nls <- nls(y ~ theta1 * theta3 * (x2 - x3 / 1.632) /
              (1 + theta2 * x1 + theta3 * x2 + theta4 * x3),
              data = d.isomer,
              start = list(theta1 = 58, theta2 = 0.017, theta3 = 0.006, theta4 = 0.044))

e) > set.seed(4567)
> D.Boot <- nlsBoot(r.nls)
```

Warning message:

In nlsBoot(r.nls) : The fit did not converge 322 times during bootstrapping

```
> plot(D.Boot)
```

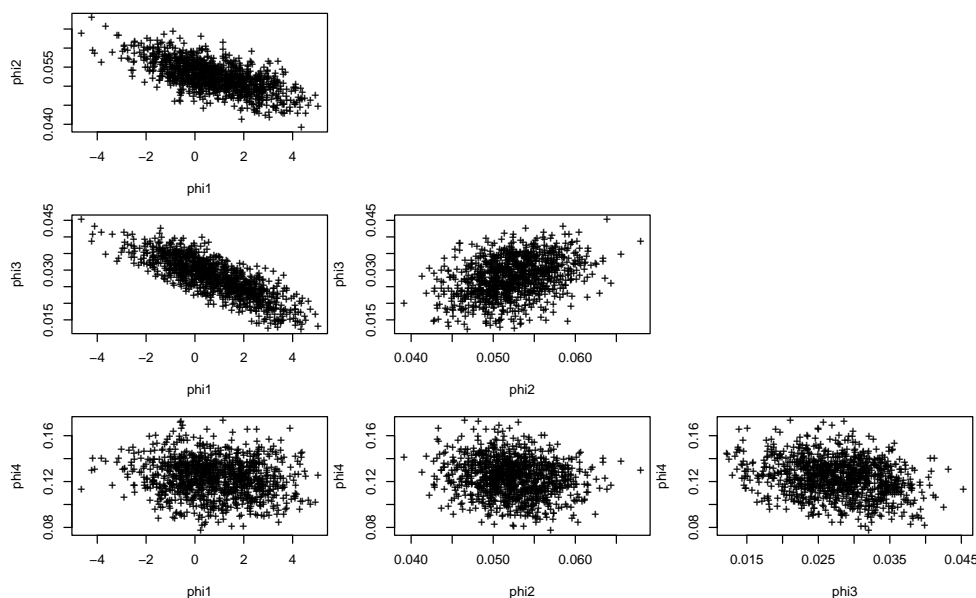


The high correlation between the parameters is clearly visible in the plots. This is probably also the reason why the fit does not converge for many of the bootstrapping samples.

```
f) > set.seed(5689)
> r.nls <- nls(y ~ (x2 - x3 / 1.632) /
              (phi1 + phi2 * x1 + phi3 * x2 + phi4 * x3),
              data = d.isomer,
              start = list(phi1 = 2.99, phi2 = 0.051, phi3 = 0.017, phi4 = 0.131))
> D.Boot <- nlsBoot(r.nls, niter = 999)

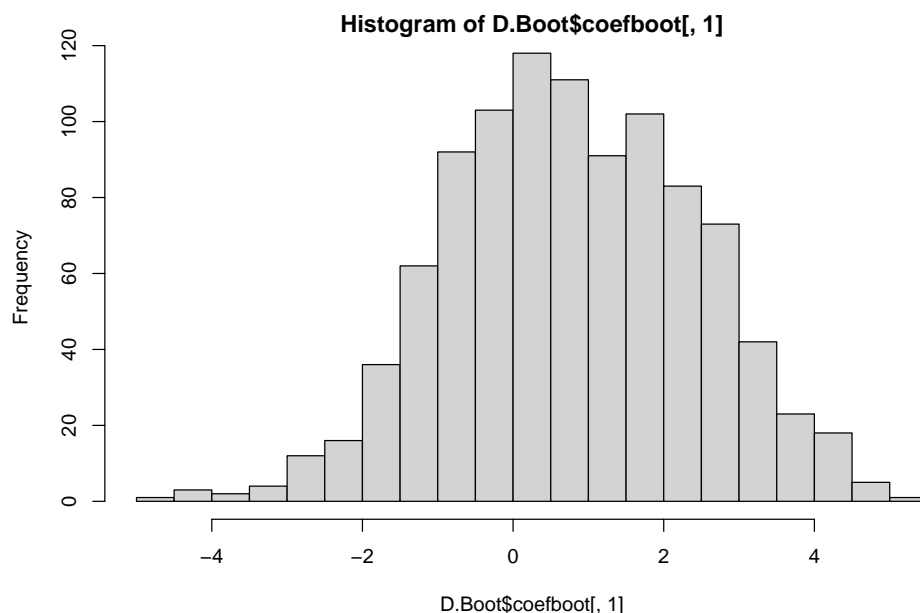
Warning message:
In nlsBoot(r.nls, niter = 999) :
  The fit did not converge 1 times during bootstrapping

> plot(D.Boot)
```



Now the correlations look much better, only  $\varphi_1$  and  $\varphi_2$ , as well as  $\varphi_1$  and  $\varphi_3$  are still quite highly correlated.

g) `> hist(D.Boot$coefboot[, 1], breaks = 20)`



We have to find the values such that 2.5% (i.e. about 25) of the replicates are below and above them. This yields the following confidence-interval:  $[-2.5, 4.25]$ .

h) `> phi1.sort <- sort(c(D.Boot$coefboot[, 1], summary(r.nls)$coefficients[1, 1]))`  
`> phi1.sort[25]`  
`[1] -2.48`  
`> phi1.sort[975]`  
`[1] 3.95`

The 95%-bootstrap confidence-interval will be about  $[-2.48, 4.05]$ . (The values will not be exactly the same as with the `summary()` argument since there are only 998 bootstrap replicates because one of the fits didn't converge).

i) The reparametrized model is:

$$y = \frac{x_2 - x_3/1.632}{\varphi_1 + \varphi_2 x_1 + \varphi_3 x_2 + \varphi_4 x_3}.$$

As  $x_1$  is in the model only once (in the term  $\varphi_2 \cdot x_1$ ) we can check if it has an influence on  $y$  by checking whether  $\varphi_2$  is significantly different from 0.

```
> summary(D.Boot)

-----
Bootstrap statistics
      Estimate Std. error
phi1    0.7829    1.63314
phi2    0.0524    0.00411
phi3    0.0277    0.00581
phi4    0.1231    0.01624

-----
Median of bootstrap estimates and percentile confidence intervals
      Median    2.5%  97.5%
phi1 0.7090 -2.4658  3.9265
phi2 0.0522  0.0444  0.0605
phi3 0.0278  0.0151  0.0381
phi4 0.1233  0.0907  0.1557

The parameter  $\varphi_2$  is indeed significantly different from 0.
```

## 2. a) Parameter estimation:

```
> d.cort <- read.table("http://stat.ethz.ch/Teaching/Datasets/cas-das/cortisol.dat",
                      header = TRUE, sep = ",")
> r.nls <- nls(y ~ theta1 + (theta2 - theta1) / (1 + exp(theta3 + theta4 * dose))^theta5,
              data = d.cort,
              start = list(theta1 = 5, theta2 = 8, theta3 = 3, theta4 = 3, theta5 = 0.6))
> summary(r.nls, corr = TRUE)

Formula: y ~ theta1 + (theta2 - theta1)/(1 + exp(theta3 + theta4 * dose))^theta5

Parameters:
      Estimate Std. Error t value Pr(>|t|)
theta1  4.8596    0.0190  255.86 < 2e-16 ***
theta2  7.9319    0.0104  760.28 < 2e-16 ***
theta3  1.4997    0.1646   9.11 7.5e-13 ***
theta4  2.6470    0.1024  25.85 < 2e-16 ***
theta5  0.4871    0.0408  11.94 < 2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0313 on 59 degrees of freedom

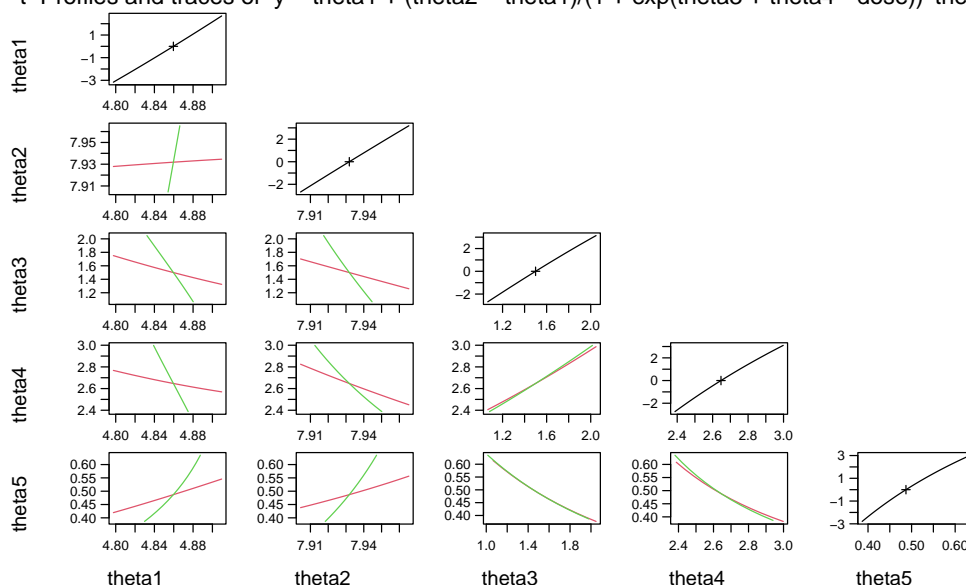
Correlation of Parameter Estimates:
      theta1 theta2 theta3 theta4
theta2  0.11
theta3 -0.42 -0.46
theta4 -0.31 -0.63  0.95
theta5  0.52  0.48 -0.99 -0.94

Number of iterations to convergence: 7
Achieved convergence tolerance: 2.53e-06
```

## b) Profile $t$ -Plot and likelihood profile traces:

```
> t.prof <- profile(r.nls)
> p.profileTraces(t.prof)
```

t-Profiles and traces of  $y \sim \theta_1 + (\theta_2 - \theta_1)/(1 + \exp(\theta_3 + \theta_4 * \text{dose}))^{\theta_5}$



The curvature in the profile  $t$ -plot of  $\theta_5$  is considerable, so the approximated confidence-interval based on the asymptotic theory (using the Wald-statistic) will show an approximation error. Also, a high correlation between  $\theta_3$  and  $\theta_4$ , between  $\theta_3$  and  $\theta_5$  and between  $\theta_4$  and  $\theta_5$  is evident. The correlations are due to an overparametrization of the model. If we assume  $\theta_5$  to be given ( $\theta_5 = 0.5$ ) all the problems seen on the likelihood profile traces disappear (see next plot).

Assume  $\theta_5 = 0.5$  is given. We then have:

$$y \approx \tilde{h}(x, \boldsymbol{\theta}) = \theta_1 + \frac{\theta_2 - \theta_1}{(1 + \exp(\theta_3 + \theta_4 x))^{0.5}}.$$

```
> r.nls2 <- nls(y ~ theta1 + (theta2 - theta1) / (1 + exp(theta3 + theta4 * dose))^0.5,
               data = d.cort, start = list(theta1 = 5, theta2 = 8, theta3 = 3, theta4 = 3))
> summary(r.nls2, corr = TRUE)
```

Formula:  $y \sim \theta_1 + (\theta_2 - \theta_1)/(1 + \exp(\theta_3 + \theta_4 * \text{dose}))^{0.5}$

Parameters:

	Estimate	Std. Error	t value	Pr(> t )
theta1	4.86270	0.01603	303.3	<2e-16 ***
theta2	7.93339	0.00908	873.8	<2e-16 ***
theta3	1.44903	0.02221	65.2	<2e-16 ***
theta4	2.61768	0.03555	73.6	<2e-16 ***

---

Signif. codes:

0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0311 on 60 degrees of freedom

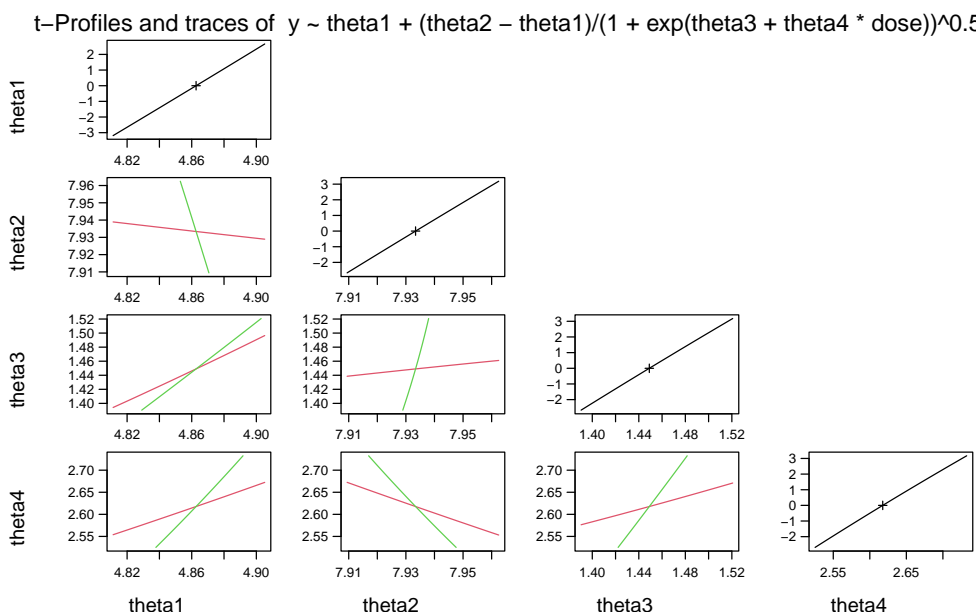
Correlation of Parameter Estimates:

	theta1	theta2	theta3
theta2	-0.19		
theta3	0.79	0.17	
theta4	0.58	-0.58	0.46

Number of iterations to convergence: 5

Achieved convergence tolerance: 1.1e-06

```
> t.prof <- profile(r.nls2)
> p.profileTraces(t.prof)
```



The profile  $t$ -curves look much better: they are more or less linear. The correlation between the parameters is no longer apparent as can be seen from the likelihood profile traces.

c) **95%-prediction interval for  $h(0, \hat{\theta})$**

```
> require(investr)
> new.x0 <- data.frame(dose=c(-1.5, 0, 2))
> DC.nls2 <- r.nls2
> DC.nls <- r.nls
> (DC.nls2p <- predFit(DC.nls2, newdata=new.x0, interval="prediction")) # based on (2b)
      fit lwr upr
[1,] 7.81 7.75 7.88
[2,] 6.20 6.14 6.27
[3,] 4.97 4.90 5.04
> DC.nls2p[, "upr"] - DC.nls2p[, "lwr"]
[1] 0.127 0.127 0.134
> (DC.nlsp <- predFit(DC.nls, newdata=new.x0, interval="prediction")) # based on (2a)
      fit lwr upr
[1,] 7.81 7.75 7.88
[2,] 6.20 6.14 6.27
[3,] 4.97 4.90 5.04
> DC.nlsp[, "upr"] - DC.nlsp[, "lwr"]
[1] 0.128 0.128 0.136
```

d) **Calibration interval:**

The calibration intervals are the intersections between the horizontal lines at specified  $y$ -values and the prediction band. The second calibration interval is longer. The calibration is optimal for  $y \in [5.5, 7.7]$ .

```
> ## Creating the graphic:
> f.x <- function(x, theta=coef(DC.nls2)){
  theta[1] + (theta[2] - theta[1]) / (1 + exp(theta[3] + theta[4] * x))~0.5
}
> new.x <- data.frame(dose=seq(-3, 3, by=0.05))
> plot(h.x, f.x(new.x$dose), type="l", col="blue")
> ## points(y~dose, data=d.cort)
> DC.nls2px <- predFit(DC.nls2, newdata=new.x, interval="prediction")
> lines(new.x$dose, DC.nls2px[, "lwr"], col="orange")
> lines(new.x$dose, DC.nls2px[, "upr"], col="orange")
> ## The calibration intervals are the intersections between the horizontal lines
> ## at specified y-values and the prediction band. The second calibration
> ## interval is longer.
```

```

> abline(h=c(5.1,7), col="gray")
>
> ## The calibration is optimal for y in the range of [5.5; 7.7].
e) Calculating the calibration intervals for  $y_0 = 7$  and  $y_0 = 5.1$ :
> (DC.nls2C7 <- invest(DC.nls2, y0=7, interval="inversion")) ## default
estimate    lower    upper
   -0.530   -0.575   -0.486
> (DC.nls2C51 <- invest(DC.nls2, y0=5.1, interval="inversion"))
estimate    lower    upper
    1.40     1.21     1.65
> DC.nls2C7$upper - DC.nls2C7$lower
[1] 0.0888
> DC.nls2C51$upper - DC.nls2C51$lower
[1] 0.439
Thus we have the calibration interval  $[-0.58, -0.49]$  for  $y = 7$  ( $x_0 = -0.53$ ) and  $[1.21, 1.65]$  for  $y = 5.1$  ( $x_0 = 1.40$ ).
f) > (DC.nls2pbC7 <- invest(DC.nls2, y0=7, interval="percentile",
                           boot.type = "parametric", nsim=600))
estimate    lower    upper      se    bias
   -0.530   -0.576   -0.484   0.023   0.000
> (DC.nls2pbC51 <- invest(DC.nls2, y0=5.1, interval="percentile",
                           boot.type = "parametric", nsim=600))
estimate    lower    upper      se    bias
    1.400     1.205     1.657   0.111   0.010
> DC.nls2pbC7$upper - DC.nls2pbC7$lower
[1] 0.0919
> DC.nls2pbC51$upper - DC.nls2pbC51$lower
[1] 0.451
The values do not differ essentially from those of 2 e).

```