

Solution to Series 5

1. a) When we look at the form of the nonlinear model

$$Y_i = \theta_1 + \theta_2 \{1 - \exp(\theta_3(\theta_4 x_{i,1} + x_{i,2}))\} + E_i, \quad (1)$$

where $\theta_3 < 0$ and $\theta_4 > 0$, we can see that for large doses $x_{i,k}$ the curves converge to $\theta_1 + \theta_2$ as the $\exp(\dots)$ -term converges to 0 in this case. As a starting value for $\theta_1 + \theta_2$ we can take the maximal weight gain which corresponds to ca. 800.

```
> d.body <- read.table("http://stat.ethz.ch/Teaching/Datasets/cas-das/body.dat",
  header = TRUE)
> max(d.body$weight)
[1] 799
```

For doses $x_{i,1} = 0$ and $x_{i,2} = 0$, the following holds: $h\langle x_{i,1}, x_{i,2} \rangle = \theta_1$. The intersection of the curve with the y-axis thus corresponds to the parameter θ_1 . As a starting value for θ_1 , we choose $\theta_1^{(0)} = 640$ and thus get $\theta_2^{(0)} = 800 - 640 = 160$.

To determine starting values for θ_3 and θ_4 as well, we use the fact that, given θ_1 and θ_2 , model (1) is transformable linear:

$$\log\left(1 - \frac{Y_i - \theta_1}{\theta_2}\right) = \theta_3(\theta_4 x_{i,1} + x_{i,2}) = \theta_3 \theta_4 x_{i,1} + \theta_3 x_{i,2} = \beta_1 x_{i,1} + \beta_2 x_{i,2}. \quad (2)$$

Using the robust MM-estimation method, we can now estimate the coefficients on the right hand side.

```
> library(robustbase)
> r.rlm <- lmrob(log(1 - (weight - 640) / 160) ~ -1 + sourceA + sourceB,
  data = d.body)
> r.rlm
Call:
lmrob(formula = log(1 - (weight - 640)/160) ~ -1 + sourceA + sourceB, data = d.body)
  --> method = "MM"
Coefficients:
sourceA  sourceB
  -8.05   -10.60
```

We get $\theta_3^{(0)} \cdot \theta_4^{(0)} = \widehat{\beta}_1 = -8.1$ and $\theta_3^{(0)} = \widehat{\beta}_2 = -10.6$, and thus we obtain also $\theta_4^{(0)} = -8.1/(-10.6) = 0.76$.

- b) We fit the nonlinear model (1) with the starting values found in a) and get:

```
> 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))
> summary(r.nls)
Formula: weight ~ theta1 + theta2 * (1 - exp(theta3 * (theta4 * sourceA +
  sourceB)))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
theta1 638.8393     6.5876  96.98 8.1e-11 ***
theta2 175.9041     6.2108  28.32 1.3e-07 ***
theta3  -6.3872     0.8019  -7.97 0.00021 ***
theta4  0.7911     0.0489  16.18 3.5e-06 ***
```

```

---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.94 on 6 degrees of freedom

Number of iterations to convergence: 7
Achieved convergence tolerance: 8.52e-06

The estimates for the parameters are thus:

```

$$\hat{\theta}_1 = 638.8, \quad \hat{\theta}_2 = 175.9, \quad \hat{\theta}_3 = -6.39, \quad \hat{\theta}_4 = 0.79.$$

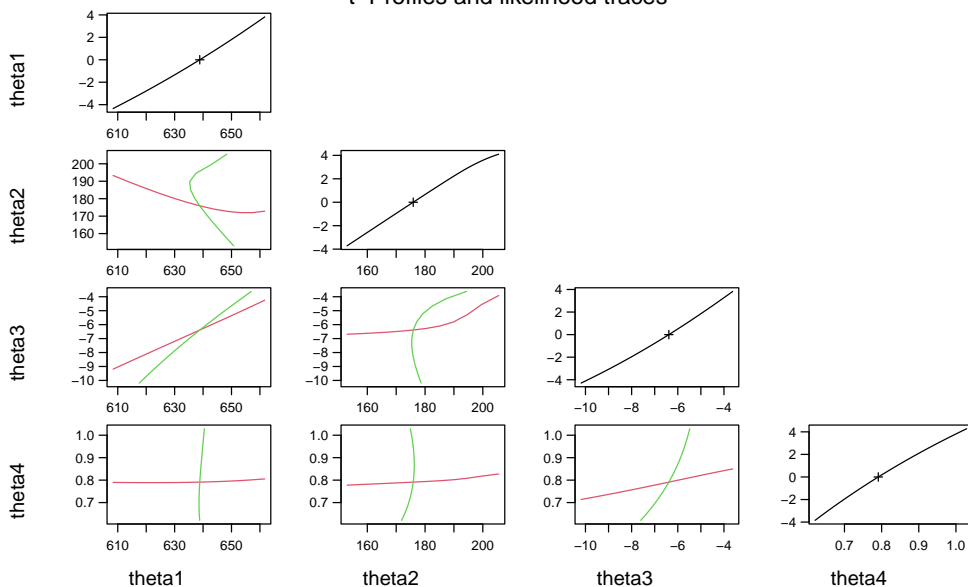
- c) We get the following *approximate* 95%-confidence-interval (based on the Wald statistic) for the parameter θ_4 (see Section 1.3.e. in the script):

$$\hat{\theta}_4 \pm \hat{\sigma}_{\hat{\theta}_4} \cdot q_{0.975}^{t_6} = 0.79 \pm 0.0489 \cdot 2.45 = [0.67, 0.91].$$

Since the value 1 does not lie inside the confidence-interval, θ_4 is significantly different from 1. This means that the growth rates of the turkeys for the two food additives are significantly different.

- d)

```
> library(sfsmisc)
> r.prof <- profile(r.nls)
> p.profileTraces(r.prof, subtitle = "t-Profiles and likelihood traces")
```



The likelihood profile traces show that the regression function tends to a strong nonlinear behaviour in the neighborhood of the parameter estimates (see the profile traces of θ_2 vs θ_1 and of θ_3 vs θ_2). The likelihood contours for the parameter θ_3 vs θ_2 are not corresponding well with the confidence ellipses of the asymptotic approximation (see Sections 2.1.c. and 2.2.c of the script). Since the t-profiles are only slightly curved in the interval $[-3, 3]$ (use scale on the y-axis), the one-dimensional confidence-intervals for θ_i are well approximated by the linear approximation.

- e) In c) we have calculated the approximate 95%-confidence-interval (based on the Wald-statistic) for the parameter θ_4 . The construction of this confidence-interval is based on the linear approximation of the model.

In the *nonlinear* model an *exact* 95%-confidence-interval for θ_4 would be given by the following equation: (see Script, Section 2.1.e.)

$$\{\theta_4 : q_{0.025}^{t_6} \leq T\langle\theta_4\rangle \leq q_{0.975}^{t_6}\} = \{\theta_4 : -2.45 \leq T\langle\theta_4\rangle \leq 2.45\},$$

where $T\langle\theta_4\rangle$ is the profile t -function of θ_4 , that is plotted in the profile t -plot of θ_4 (plot on the bottom right). Since this curve is only slightly curved, we can rely on the approximated 95%- confidence-interval calculated in c).

f) Confidence-intervals for the estimated parameters based on the profile t -function:

```
> confint(r.nls) # or confint(profile(r.nls))
                2.5%  97.5%
theta1 622.393 653.81
theta2 160.802 191.42
theta3  -8.427  -4.58
theta4   0.679   0.92
```

2. a) Model linearization:

$$y = \frac{\theta_1 \theta_3 (x_2 - x_3 / 1.632)}{1 + \theta_2 x_1 + \theta_3 x_2 + \theta_4 x_3}$$

$$\frac{1}{y} = \frac{1 + \theta_2 x_1 + \theta_3 x_2 + \theta_4 x_3}{\theta_1 \theta_3 (x_2 - x_3 / 1.632)}$$

$$\frac{x_2 - x_3 / 1.632}{y} = \frac{1}{\theta_1 \theta_3} + \frac{\theta_2}{\theta_1 \theta_3} x_1 + \frac{1}{\theta_1} x_2 + \frac{\theta_4}{\theta_1 \theta_3} x_3$$

$$\frac{x_2 - x_3 / 1.632}{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

Develop

$$\tilde{y}_i = \frac{x_{i2} - x_{i3} / 1.632}{y_i}$$

and compute the estimates for the regression:

$$\tilde{y}_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i$$

The relation between β_k and θ_k is:

$$\left. \begin{aligned} \beta_0 &:= \frac{1}{\theta_1 \theta_3} \\ \beta_1 &:= \frac{\theta_2}{\theta_1 \theta_3} \\ \beta_2 &:= \frac{1}{\theta_1} \\ \beta_3 &:= \frac{\theta_4}{\theta_1 \theta_3} \end{aligned} \right\} \Rightarrow \left\{ \begin{aligned} \theta_1 &= \frac{1}{\beta_2} \\ \theta_2 &= \frac{\beta_1}{\beta_0} \\ \theta_3 &= \frac{\beta_2}{\beta_0} \\ \theta_4 &= \frac{\beta_3}{\beta_0} \end{aligned} \right.$$

Determine starting values with R:

- Transformation of y :

```
> d.isomer <- read.table("http://stat.ethz.ch/Teaching/Datasets/cas-das/isomer.dat",
                        header = TRUE)
> d.isomer$ytilde <- (d.isomer$x2 - d.isomer$x3 / 1.632) / d.isomer$y
```

- Starting values from least-squares regression:

```
> r.lm <- lm(ytilde ~ x1 + x2 + x3, data = d.isomer)
> r.lm
```

Call:

```
lm(formula = ytilde ~ x1 + x2 + x3, data = d.isomer)
```

Coefficients:

(Intercept)	x1	x2	x3
-2.50789	0.06764	0.00519	0.18538

With these coefficients we get the following starting value for θ :

$$\theta_1^{(0)} = \frac{1}{0.0052} = 192.3, \quad \theta_2^{(0)} = \frac{0.0676}{-2.51} = -0.027,$$

$$\theta_3^{(0)} = \frac{0.0052}{-2.51} = -0.002, \quad \theta_4^{(0)} = \frac{0.1854}{-2.51} = -0.074.$$

- Starting values using a robust MM-estimation:

```
> library(robustbase)
> r.MM <- lmrob(ytilde ~ x1 + x2 + x3, data = d.isomer)
> r.MM
```

Call:

```
lmrob(formula = ytilde ~ x1 + x2 + x3, data = d.isomer)
  \--> method = "MM"
```

Coefficients:

(Intercept)	x1	x2	x3
2.9890	0.0507	0.0173	0.1313

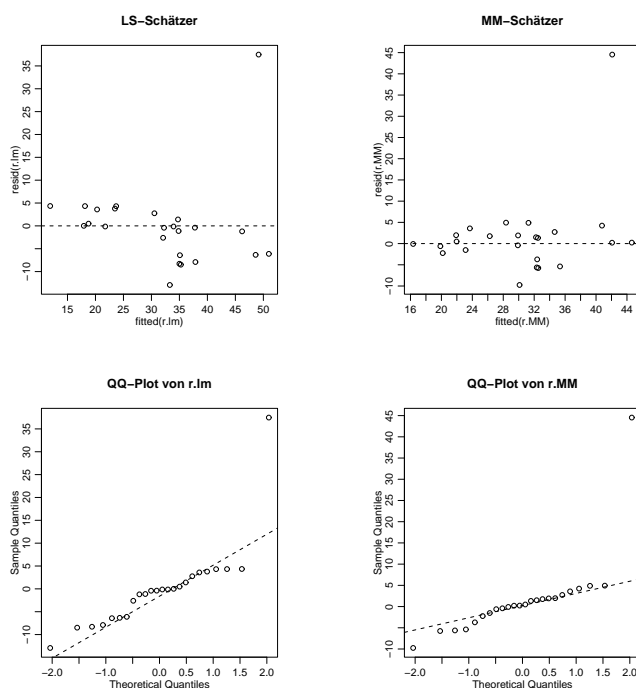
With these coefficients we get the following starting value for θ :

$$\begin{aligned}\theta_1^{(0)} &= \frac{1}{0.0173} = 58, & \theta_2^{(0)} &= \frac{0.0507}{2.989} = 0.017, \\ \theta_3^{(0)} &= \frac{0.0173}{2.989} = 0.006, & \theta_4^{(0)} &= \frac{0.1313}{2.989} = 0.044.\end{aligned}$$

Starting values:

We use the estimates $\hat{\beta}_0, \dots, \hat{\beta}_3$ from the **robust regression** to get the starting values $\theta_1^{(0)}, \dots, \theta_4^{(0)}$ for $\theta_1, \dots, \theta_4$. Here are two reasons for using the robust regression:

1. The parameters $\theta_1, \dots, \theta_4$ must be positive. In contrast to the robust MM-regression, the least-squares regression yields negative starting values for $\theta_1, \dots, \theta_4$. This is not meaningful in a physical sense (see description of the exercise).
2. The errors are not normally distributed (check residual plots). In particular, the errors in the linearized model cannot be additive and normally distributed if the errors in the nonlinear model are additive and normally distributed.



b) Estimation of the nonlinear regression model:

```
> 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))
> summary(r.nls, correlation = TRUE)
```

Formula: $y \sim \theta_1 * \theta_3 * (x_2 - x_3 / 1.632) / (1 + \theta_2 * x_1 + \theta_3 * x_2 + \theta_4 * x_3)$

Parameters:

```

      Estimate Std. Error t value Pr(>|t|)
theta1  35.9202     8.2123   4.37 0.00029 ***
theta2   0.0708     0.1787   0.40 0.69596
theta3   0.0377     0.1001   0.38 0.71010
theta4   0.1671     0.4160   0.40 0.69210

```

```
---
```

```
Signif. codes:
```

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

```
Residual standard error: 0.402 on 20 degrees of freedom
```

```
Correlation of Parameter Estimates:
```

```

      theta1 theta2 theta3
theta2 -0.80
theta3 -0.84   1.00
theta4 -0.79   1.00   1.00

```

```
Number of iterations to convergence: 17
```

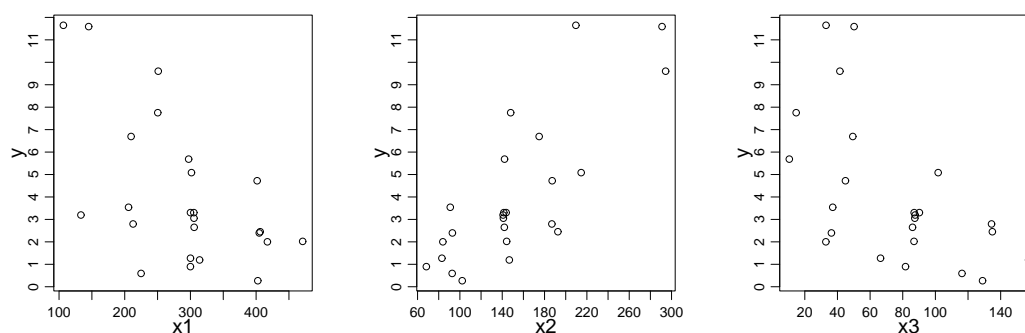
```
Achieved convergence tolerance: 2.26e-06
```

Problems:

1. High correlation between the parameter estimates (as can be seen from the "Correlation of Parameter Estimates" in the summary output).
2. Only θ_1 is significantly different from 0. If we were to omit the coefficients θ_2 , θ_3 and θ_4 (this means set $\theta_2 = \theta_3 = \theta_4 = 0$), we would get the model

$$y \approx h(\mathbf{x}, \boldsymbol{\theta}) = 0.$$

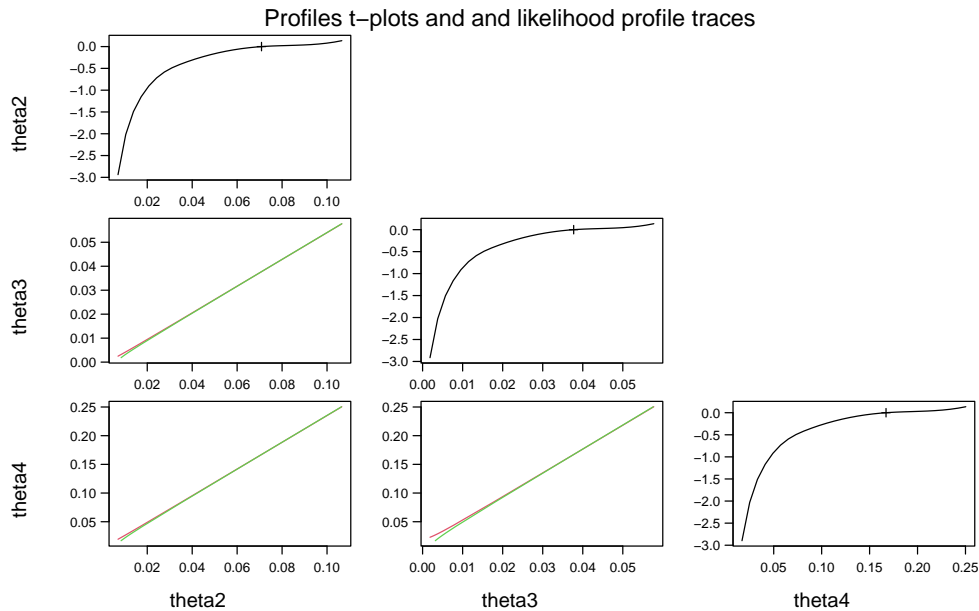
This surely is not meaningful. We can see clearly in the scatterplot that there are relationships between the target variable and the explanatory variables. The standard errors are probably much too big. This can be an indicator that the linear approximation is not good in the neighborhood of the parameter estimates $\hat{\boldsymbol{\theta}}$.



```

c) > ### Profile t-plots and likelihood profile traces
> library(sfsmisc)
> r.prof <- profile(r.nls, which = 2:4, delta.t = 0.2)
> p.profileTraces(r.prof, subtitle = "Profiles t-plots and and likelihood profile traces")

```



Profile t -Plots:

There is a strong deviation from linearity in the neighborhood of $\hat{\theta}_k$ (for $k = 2, 3, 4$). This confirms that the linear approximation in the neighborhood of $\hat{\theta}_k$ (for $k = 2, 3, 4$) is not suited to determine the confidence interval.

Likelihood profile traces:

The curves are virtually on top of each other which means that the parameters are highly correlated. This confirms the findings from the summary output in **b)**.

Note: The parameters are so highly correlated, that it is not possible to run the function `profile()` for all 4 parameters. Thus we can only see the profile t -plots and likelihood profile traces for three parameters.

3. a) Reparametrization of the model:

The original model

$$y = h(\mathbf{x}, \boldsymbol{\theta}) = \frac{\theta_1 \theta_3 (x_2 - x_3 / 1.632)}{1 + \theta_2 \cdot x_1 + \theta_3 \cdot x_2 + \theta_4 \cdot x_3}$$

can be rewritten as

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

when using

$$\varphi_1 = \frac{1}{\theta_1 \theta_3}, \quad \varphi_2 = \frac{\theta_2}{\theta_1 \theta_3}, \quad \varphi_3 = \frac{1}{\theta_1}, \quad \varphi_4 = \frac{\theta_4}{\theta_1 \theta_3}.$$

Note that the coefficients φ_k are equivalent to the coefficients β_k from **2a)**. The only difference is that we do not linearize the equation here.

As starting values for $\varphi_1, \dots, \varphi_4$ we can use the estimations from the MM-regression obtained in **2a)**:

$$\varphi_1^{(0)} = \hat{\beta}_0 = 2.99, \quad \varphi_2^{(0)} = \hat{\beta}_1 = 0.051, \quad \varphi_3^{(0)} = \hat{\beta}_2 = 0.017, \quad \varphi_4^{(0)} = \hat{\beta}_3 = 0.131$$

```
> 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))
> summary(r.nls, corr = TRUE)
```

Formula: $y \sim (x_2 - x_3 / 1.632) / (\text{phi1} + \text{phi2} \cdot x_1 + \text{phi3} \cdot x_2 + \text{phi4} \cdot x_3)$

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
phi1	0.73784	1.81740	0.41	0.68906
phi2	0.05227	0.00458	11.41	3.3e-10 ***
phi3	0.02784	0.00636	4.37	0.00029 ***

```
phi4 0.12332    0.01768    6.98 9.0e-07 ***
```

```
---
```

```
Signif. codes:
```

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

```
Residual standard error: 0.402 on 20 degrees of freedom
```

```
Correlation of Parameter Estimates:
```

```
      phi1 phi2 phi3
phi2 -0.67
phi3 -0.81  0.35
phi4 -0.15 -0.13 -0.24
```

```
Number of iterations to convergence: 5
```

```
Achieved convergence tolerance: 4.3e-06
```

All pairwise correlation coefficients are now smaller than 0.9.

Note that the estimates of φ_k and of β_k are different, although the φ_k 's correspond to the β_k 's in the deterministic part of the models. The statistical parts of the models (i.e. the error terms) are different in **2a)** and here.

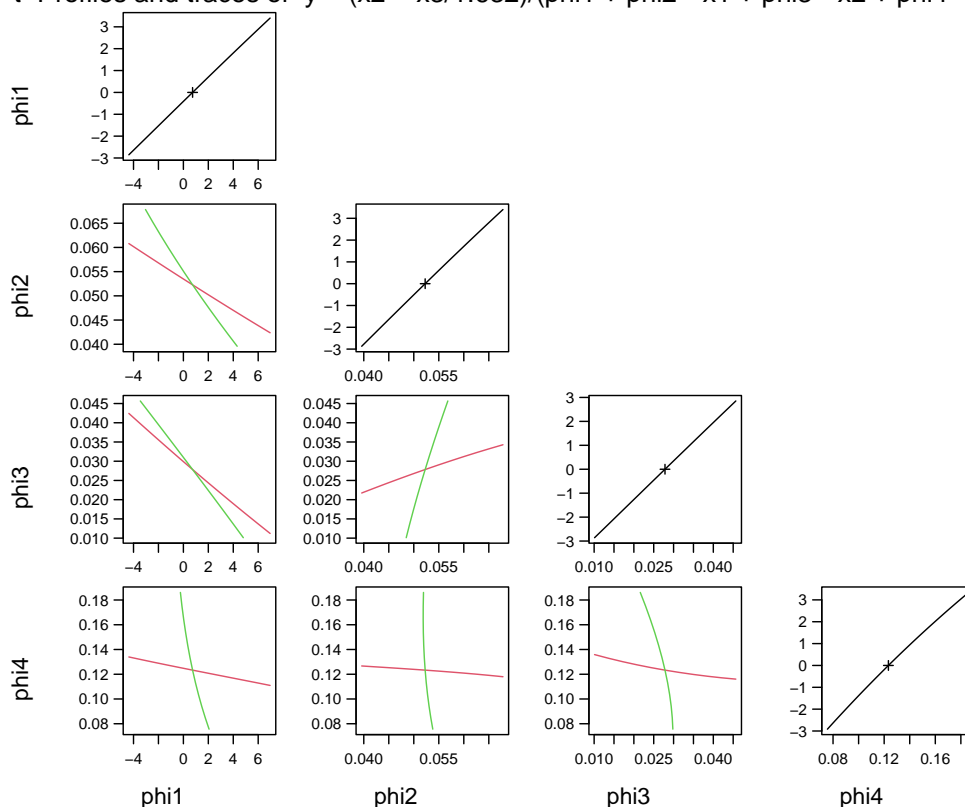
b) Likelihood profile traces:

```
> library(sfsmisc)
```

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

```
> p.profileTraces(t.prof)
```

```
t-Profiles and traces of y ~ (x2 - x3/1.632)/(phi1 + phi2 * x1 + phi3 * x2 + phi4 * x3)
```



- **Profiles t -plots:**

The curves are almost linear. Thus the linear approximation for determining the standard errors is good.

- **Likelihood profile traces:**

The likelihood profile traces are almost perpendicular (except for φ_1 vs φ_2 and φ_1 vs φ_3), this means that the parameter estimations are uncorrelated. Since the profile traces are almost linear, the confidence regions are well described by ellipses.

Note: The plots of *all four* parameters can be produced after this reparametrization (in contrast to before).

c) **Confidence intervals for $\varphi_1, \dots, \varphi_4$:**

A 95%-confidence-interval for φ_k based on the Wald-statistics is given by

$$\hat{\varphi}_k \pm t_{0.975}^{20} \cdot \hat{\sigma}_k,$$

where $t_{0.975}^{20} = 2.1$ is the 97.5%-quantile of the t -distribution with 20 degrees of freedom and $\hat{\sigma}_k$ is the standard error of $\hat{\varphi}_k$.

We thus get following approximate 95%-confidence-intervals:

$$\begin{array}{ll} \varphi_1: & \text{CI} = 0.74 \pm 3.79 \\ \varphi_2: & \text{CI} = 0.052 \pm 0.0096 \\ \varphi_3: & \text{CI} = 0.028 \pm 0.013 \\ \varphi_4: & \text{CI} = 0.12 \pm 0.037. \end{array}$$

We can note that φ_1 is not significantly different from 0.

Using `confint()` we get the confidence interval from the profile t -functions. The differences between the confidence intervals obtained by both methods here are very small, since the curves are almost linear (see profile t -plot in part b)).

```
> library(MASS)
> confint(t.prof) # or confint(r.nls)
      2.5%  97.5%
phi1 -2.9984 4.5206
phi2  0.0430 0.0617
phi3  0.0149 0.0408
phi4  0.0887 0.1613
```

d) We can calculate the 95%-confidence-intervals for the original parameters θ_k using the ones of φ_k and the formula in the Section 2.3.j. of the script.

We have $\theta_1 = \frac{1}{\varphi_3}$, i.e. $\theta_1 = g(\varphi_3)$, where $g(x) = \frac{1}{x}$.

We thus get the following 95%-confidence interval for θ_1 (with correction for non-linearity):

$$CI_{\theta_1} = g(CI_{\varphi_3}) = \frac{1}{CI_{\varphi_3}} = \frac{1}{0.028 \mp 0.013} = [24, 69].$$

The gaussian interval (see Section 2.3.i. in the script) is not appropriate here due to the non-linearity of the original parameters. It is considerably different:

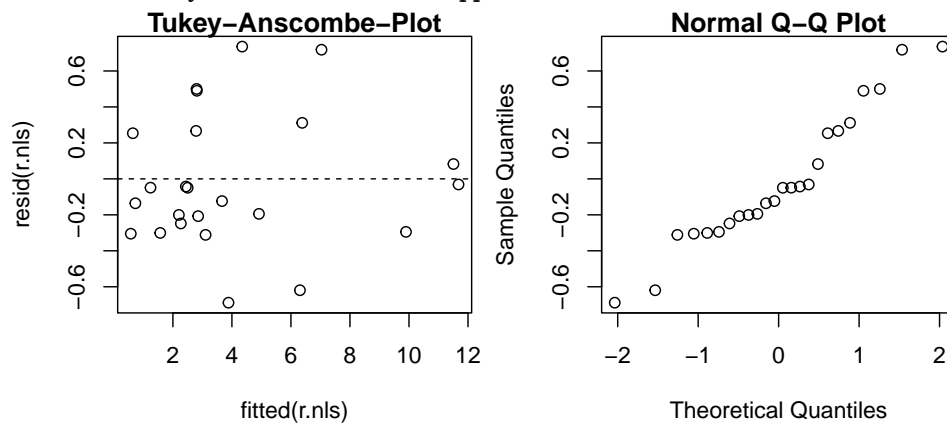
$$\hat{\sigma}_{\theta_1}^2 \approx \left(\frac{\partial}{\partial \varphi_3} \frac{1}{\varphi_3} \right)^2 \hat{\sigma}_{\varphi_3}^2 = \left(-\frac{1}{\varphi_3^2} \right)^2 \hat{\sigma}_{\varphi_3}^2 \implies \hat{\sigma}_{\theta_1} = \frac{\hat{\sigma}_{\varphi_3}}{(\hat{\varphi}_3)^2} = \frac{0.00637}{0.028^2} = 8.1$$

We thus get as 95%-confidence interval for θ_1 (by Gaussian approximation):

$$\frac{1}{\hat{\varphi}_3} \pm \hat{\sigma}_{\theta_1} t_{0.975}^{20} = [19, 53]$$

e) **Residual analysis:**

```
> par(mfrow = c(1, 2))
> plot(fitted(r.nls), resid(r.nls)); abline(h = 0, lty = 2)
> title("Tukey-Anscombe-Plot"); qqnorm(resid(r.nls))
```



The Tukey-Anscombe Plot does not show any special structure. The normal plot looks good. The assumptions (normally distributed errors, constant variance) seem to be fulfilled.