Statistical Models – October 2016 — Assignment 02

Chouliaras Georgios Christos , Jiayang Zhuo
Group 15

I. Computational Problems

Problem 1

(i) A scatterplot of the data can be seen in the figure below:

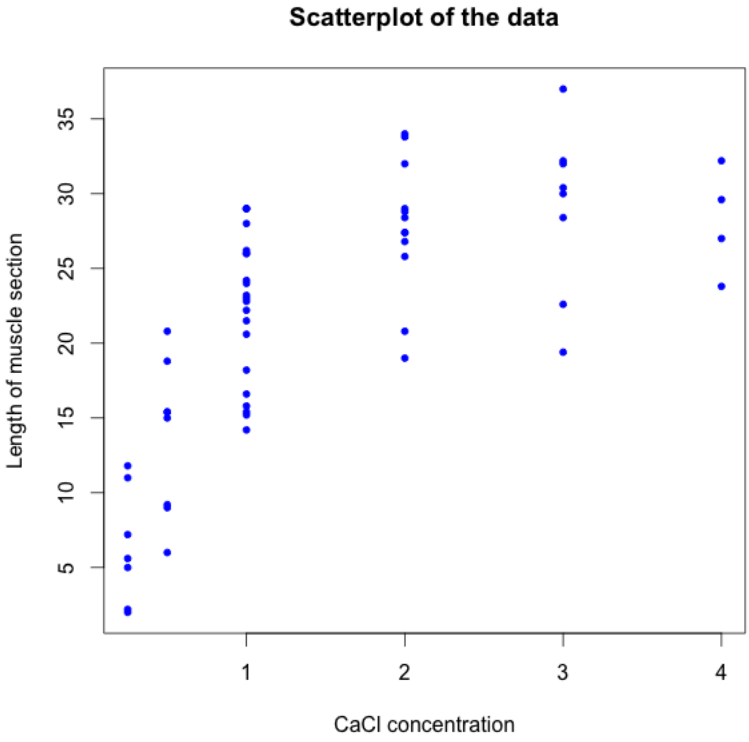


Figure 1: Scatterplot of muscle Length and CaCl concentration

The distribution of the points in the scatterplot shows a non-linear relationship between the *length of the muscle section* and the *CaCl concetration*, so a linear model would not work on these data. The plot's shape is similar to that of a logistic curve and thus a non-linear function of the form $f(x) = c_1 + c_2\ e^{(-x/c_3)}$ might be a good fit for these data.

(ii) For the estimation of the model's parameters, the Gauss Newton method with initial values $\theta_1 = 10, \theta_2 = -5, \theta_3 = 5$ using the `nls` function resulted to the following estimates after 8 iterations:

$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$
28.9633	-34.2274	0.6082

Table 1: Estimated parameters θ

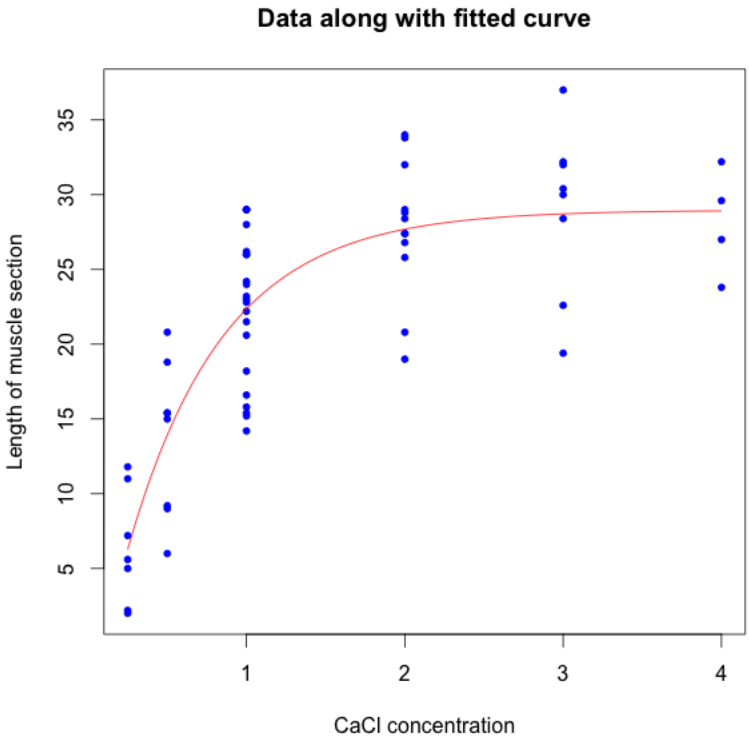


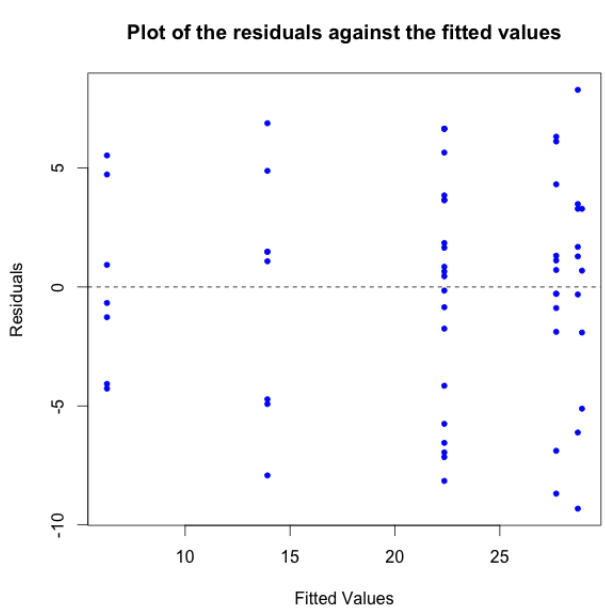
Figure 2: Plot of the data along with the fitted curve.

From an observation in the plot of the data along with the fitted curve containing the estimated parameter values in figure (2) it can be seen that the model fits the data well, as the fitted line follows the overall course of the *muscle length* as *CaCl concentration* increases. Ideally, there should be a slight fall in the curve for concentration over 3, however it might occur due to the limited data points for the group with concentration 4.

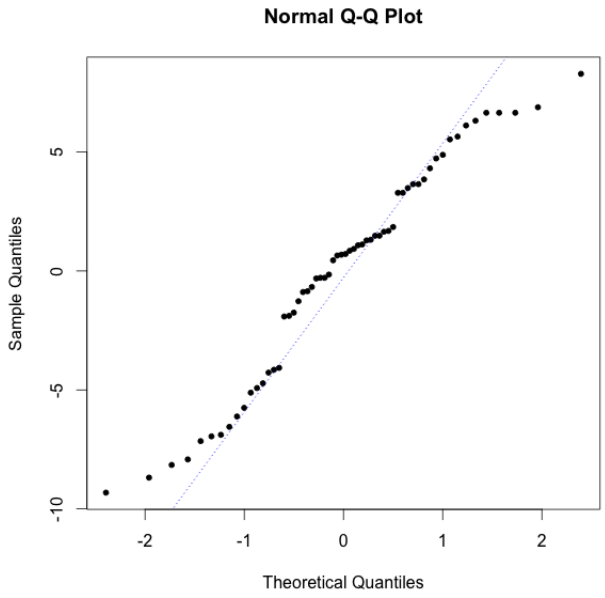
At this point it is essential to check the assumptions for the errors' normality and constant error variance. A plot of the residuals against the fitted values in figure (3a) does not show any visible pattern and the spread of the points around zero is fairly constant, so the variance of the residuals can be considered constant. A normal quantile plot of the residuals in figure (3b) shows a linear relationship between the theoretical and the sample quantiles, however there are many points that deviate from the line, but not in a great extent. A *Shapiro-Wilk* normality test produced a p value 0.052 which is very low, however it is greater than the significance level $\alpha = 0.05$ so the residuals can be considered normal.

(iii) In order to obtain different estimates of θ for each animal, the model reformed to:

$$Y_{jk} = \theta_{1,k} + \theta_{2,k}\, e^{-X_{jk}/\theta_3} + \varepsilon_{jk} \qquad j = 1, \ldots, n_k \qquad k = 1, \ldots, 21$$



(a) Plot of the residuals against the fitted values



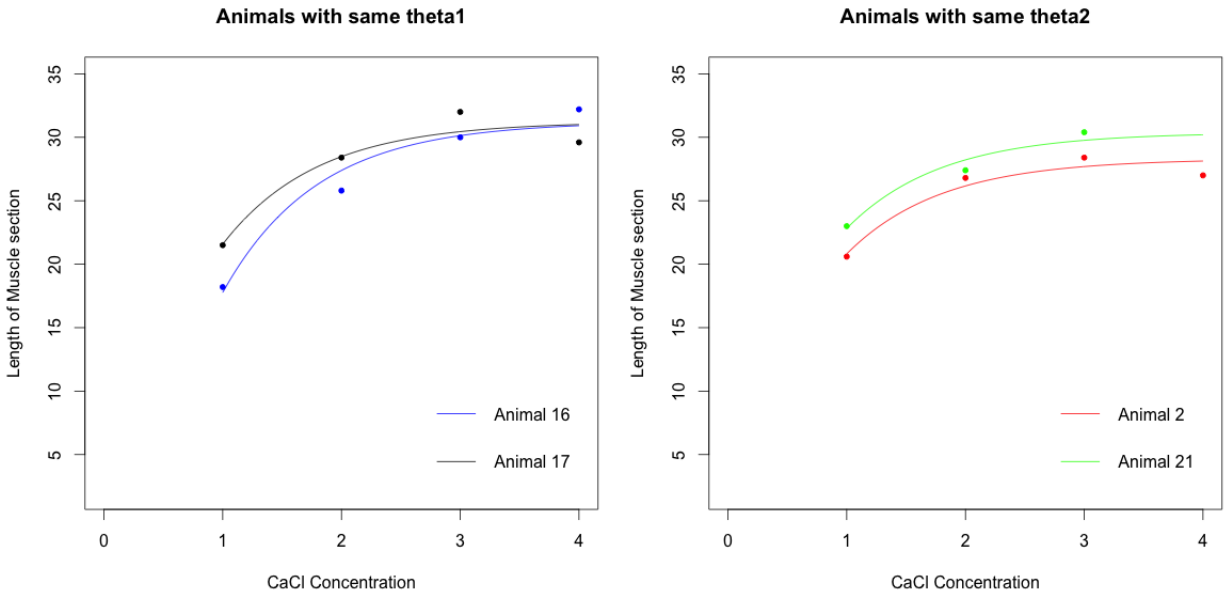
(b) Normal Q-Q plot of the residuals for confirmation of normality

Figure 3: Check for validity of normality and constant error variance of the residuals of the model.

Again, with the Newton-Gauss iterative method using this time as initial values the θ estimates of the former model, the estimated values for the parameter vector $\boldsymbol{\theta^*} = (\theta_{1,1}, \ldots, \theta_{1,21}, \theta_{2,1}, \ldots, \theta_{2,21}, \theta_3)$ can be seen in table (2). The estimations for $\theta_{1,k}$ range from 19.99 to 40.05 while $\theta_{2,k}$ ranges from -47.20 to -15.89. This wide range of values in these parameters, means that the *CaCl concentration* affects each animal's *length of muscles* differently and thus, the creation of this second model makes sense. In order to inspect how these parameters affect each animal, the curves for two animals with almost the same $\hat{\theta}_1$ and two animals with almost the same $\hat{\theta}_2$ have been plotted.

More, specifically in figure (4a) the curves for the animals 16 and 17 have been plotted. These animals have $\hat{\theta}_{1,16} = 31.2258$ and $\hat{\theta}_{1,17} = 31.2303$ which are very similar, but different $\hat{\theta}_2$: $\hat{\theta}_{2,16} = -47.2068$ and $\hat{\theta}_{2,17} = -33.8746$. From the figure it can be observed that the curves have a different slope and thus, $\hat{\theta}_2$ is the parameter which affects the *slope* of the curves. On the other hand, in figure (4b) the curves of the animals 2 and 21 with the same $\hat{\theta}_2$ but with different $\hat{\theta}_1$ seem to have the same slope, but are in a different height. Hence, $\hat{\theta}_1$ is the parameter which affects the *vertical movement* of the curves. It has to be noted, that the effect of each parameter could be more clear if there were more data points for each animal, however even with limited data points, these plots explain why a new model with different θ values for different animals has been created.

Finally, it is necessary to check the full model's assumptions for normality and constant error variance of the residuals. Figure (5a) shows a plot of the model's residuals against the fitted values which does not indicate any pattern, so the assumption of constant error variance is valid. However, in the normal quantile plot of figure (5b) the majority of the points deviate from the line, something which indicates that the normal assumption of the errors might not be valid. Moreover, a *Shapiro-Wilk* normality test produced a p value 0.0089 which is low enough to reject the hypothesis for the normality of the errors.

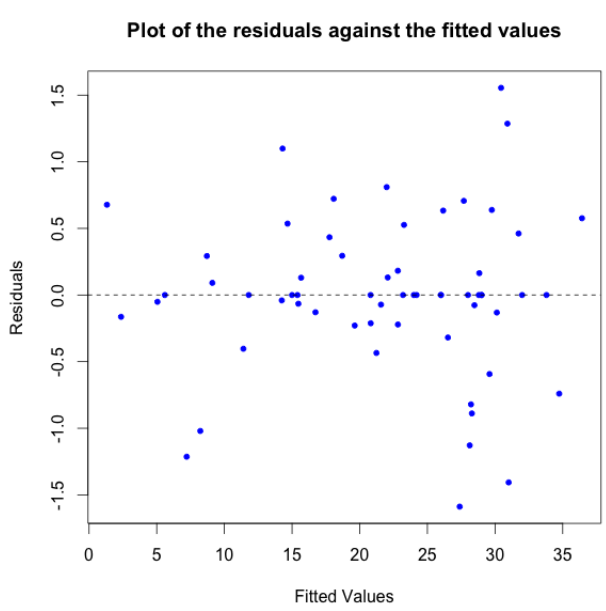


(a) Effect of parameter θ_2 *(b) Effect of parameter θ_1*

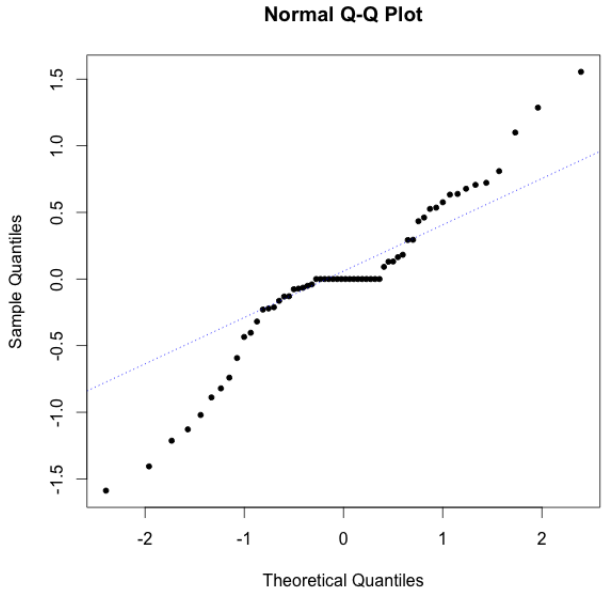
Figure 4: Affection of parameters θ_1 and θ_2 on the curves.

k	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$
1	23.4541	-27.3004	
2	28.3020	-26.2702	
3	30.8007	-30.9011	
4	25.9211	-32.2384	
5	25.9211	-29.9406	
6	20.1200	-20.6219	
7	33.5953	-19.6246	
8	39.0527	-45.7799	
9	32.1369	-31.3446	
10	40.0052	-38.5987	0.7969
11	36.1904	-33.9211	
12	36.9109	-38.2680	
13	30.6346	-22.5683	
14	34.3118	-36.1669	
15	38.3952	-32.9521	
16	31.2258	-47.2068	
17	31.2303	-33.8746	
18	19.9977	-15.8962	
19	37.0953	-28.9690	
20	32.5942	-36.9171	
21	30.3757	-26.5075	

Table 2: Estimated parameters θ for each animal.



(a) Plot of the residuals against the fitted values



(b) Normal Q-Q plot of the residuals for confirmation of normality

Figure 5: Check for validity of normality and constant error variance of the residuals of the full model

Problem 2

(i) The requested model was created,by selecting appropriate values for $\boldsymbol{\theta}$ and σ^2 as follows:

$$\boldsymbol{\theta} = (5, 90, 2) \qquad \sigma^2 = 0.5$$

The vector $x_i \in [0,3], i = 1, \ldots, n$ was created using the `runif(n, 0, 3)` function and the independent errors $\varepsilon_1, \ldots, \varepsilon_{100} \backsim N(0, \sigma^2)$ using the `rnorm` function with mean 0. A scatterplot of the simulated data can be seen in figure (6) in which the blue line indicates the plot of the true curve $f(x, \boldsymbol{\theta})$. The true curve follows the overall course of the data nicely, by decreasing rapidly after x=0.5 and then more slowly after x=2. This nice fit is reasonable and not surprising, as the data were simulated using this specific function.

In order to fit a model in the data, non-linear regression with initial values $\theta_1 = 10, \theta_2 = 30, \theta_3 = 9$ was used to estimate $\boldsymbol{\theta}$. The estimated parameters can be seen below:

$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$
4.899061	92.888626	2.081185

Table 3: Estimated parameters θ

These estimates are very close to the original vector $\boldsymbol{\theta} = (5, 90, 2)$ which means that the accuracy of the model is high. Moreover, the estimated variance as well as the estimated covariance matrix are:

$$\hat{\sigma}^2 = 0.4695968$$

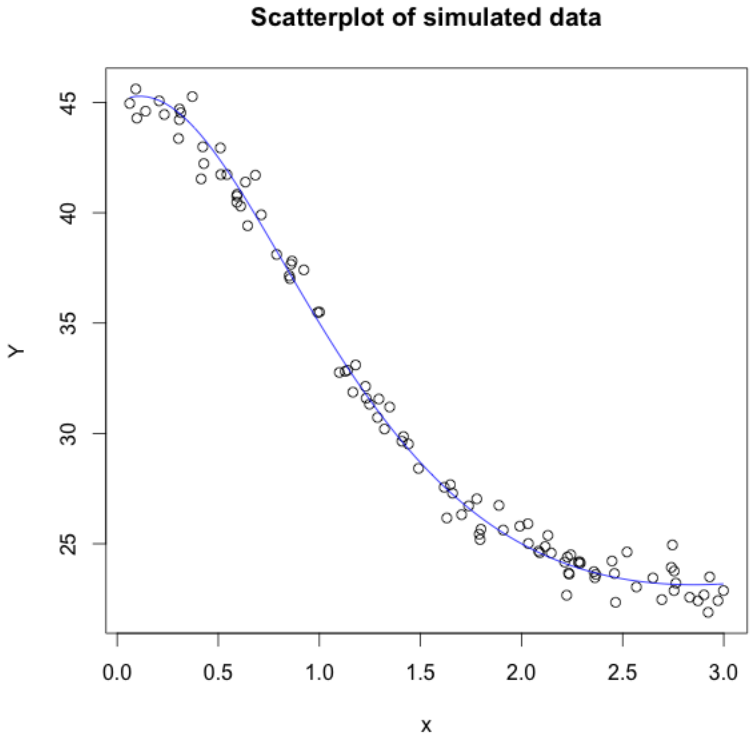


Figure 6: Scatterplot of simulated data along with true curve $f(x, \boldsymbol{\theta})$

$$\widehat{Cov(\hat{\boldsymbol{\theta}})} = \begin{bmatrix} 0.009389367 & -0.1802279 & -0.004308497 \\ -0.180227904 & 4.2163990 & 0.103125401 \\ -0.004308497 & 0.1031254 & 0.002574023 \end{bmatrix}$$

The high accuracy of the model is verified once again, as the estimated variance is close to the real variance $\sigma^2 = 0.5$. In order to check also the accuracy of the estimated covariance matrix, the real covariance matrix calculated using the following formula:

$$Cov(\theta) = \sigma^2 (V^T V)^{-1}$$

where V is the 100 x 3 derivative matrix containing the partial derivatives with respect to each of the θ's:

$$V = [\frac{\partial f}{\partial \theta_1}(x_i, \theta), \frac{\partial f}{\partial \theta_2}(x_i, \theta), \frac{\partial f}{\partial \theta_3}(x_i, \theta)]$$

for $i = 1, \ldots, 100$. So, the real covariance matrix is as follows:

$$Cov(\boldsymbol{\theta}) = \begin{bmatrix} 0.009502168 & -0.17550616 & -0.004171477 \\ -0.175506162 & 3.99137111 & 0.097126859 \\ -0.004171477 & 0.09712686 & 0.002414359 \end{bmatrix}$$

It can be observed, that the estimated covariance matrix is very similar to the real one and this is another indication of the model's high accuracy.

The fitted curve was plotted along with the true curve in the scatterplot of the simulated data and can be seen in figure (7). The model fitted the data well, and the fitted curve is really close to the real curve with a small deviation in the values of x between 0 and 0.5. This nice fit is also not surprising, as the estimated parameters $\hat{\boldsymbol{\theta}}$ are very close to the original parameters and thus, a plot of the fitted curve was expected to be similar to the original one.

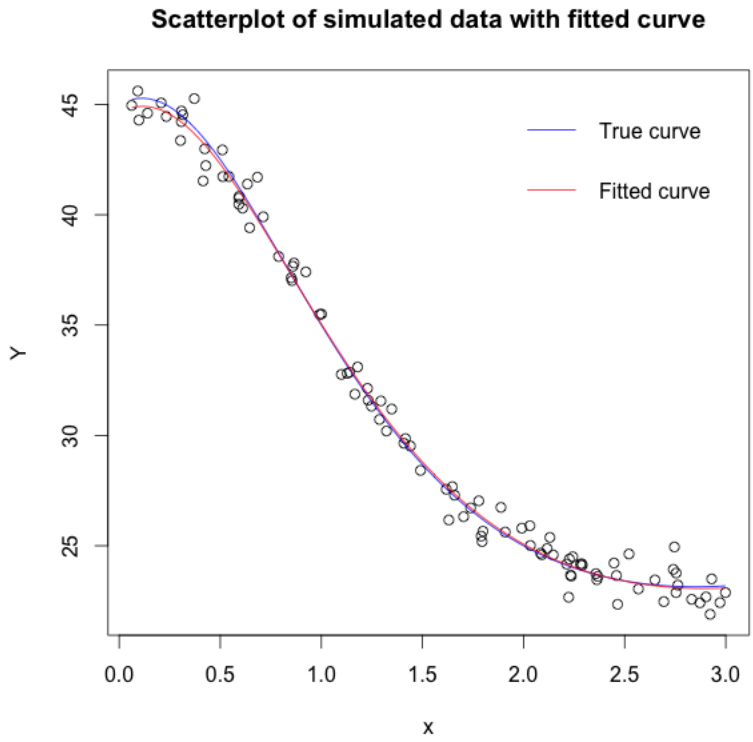


Figure 7: Scatterplot of simulated data along with true curve $f(x, \boldsymbol{\theta})$

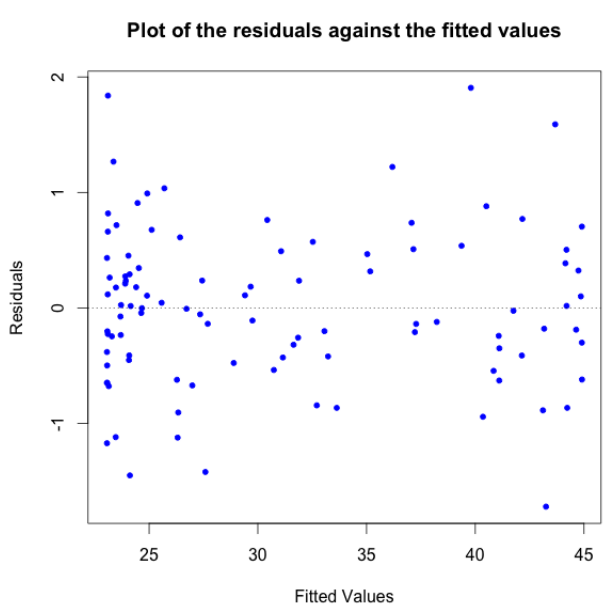
(ii) To confirm the model's assumptions the residuals against the fitted values were plotted, along with a normal Q-Q plot in figure (8). The residuals are nicely spread out without any indication of pattern, so the assumption of constant error variance is valid. Also the sample quantiles have a linear relationship with the theoretical quantiles, with a small deviation in extremely low and high values. Furthermore, a *Shapiro-Wilk* test on the residuals produced a p value 0.8004 which along with the aforementioned Q-Q plot, confirm the assumption of normality for the errors.

(iii) The approximate 98% confidence intervals which were calculated using *asymptotic normality* are:

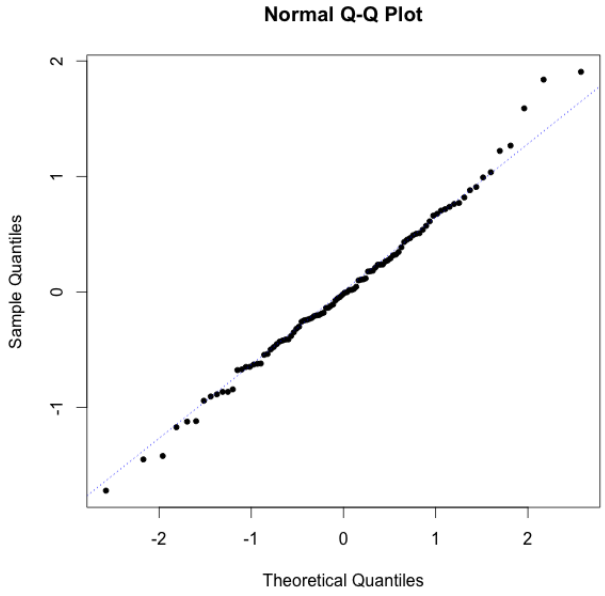
$$\hat{\theta}_l \pm t_{97;0.99}\ \hat{\sigma}\ \sqrt{(\mathbb{V}^{\hat{T}}\mathbb{V})^{-1}_{ll}}$$

with $l = 1, 2, 3$ and $\alpha = 0.02$, the calculated 98% confidence intervals are as follows:

$$4.891196 \leq \theta_1 \leq 4.906926 \qquad 89.356628 \leq \theta_2 \leq 96.420625 \qquad 2.079028 \leq \theta_3 \leq 2.083341$$



(a) Plot of the residuals against the fitted values



(b) Normal Q-Q plot of the residuals for confirmation of normality

Figure 8: Check for validity of normality assumptions on the residuals of the fitted model

Using the *bootstrap*, the approximate 98% bootstrap confidence intervals for B=1000 iterations are:

$$[2\hat{\theta}_l - \theta^*_{l;1+\lfloor 0.99\cdot 1000\rfloor}, 2\hat{\theta}_l - \theta^*_{l;1+\lfloor 0.02\cdot 1000\rfloor}]$$

and were calculated as:

$$4.684915 \leq \theta_1 \leq 5.128091 \qquad 88.010885 \leq \theta_2 \leq 97.125017 \qquad 1.965550 \leq \theta_3 \leq 2.191569$$

It can be noticed, that the *bootstrap* confidence intervals are very similar to those of the *asymptotic normality*. Moreover, it can also be observed, that the *bootstrap intervals* are more accurate than the *asymptotic normality* intervals, due to the fact that the real $\theta_1 = 5$ and $\theta_3 = 2$ lie between the *bootstrap intervals* but not in those of *asymptotic normality*. One reason for this, is the fact that the *asymptotic normality* intervals are a bit more narrower than the *bootstrap intervals*.

(iv) An approximate 98% confidence interval for the expected value of Y in the case that x=1 and assymptotic normality holds, can be obtained from:

$$f(\boldsymbol{x}, \hat{\boldsymbol{\theta}}) \pm t_{n-p;1-\alpha/2}\hat{\sigma}\sqrt{\hat{v}_x^T(\hat{\mathbb{V}}^T\hat{\mathbb{V}})^{-1}\hat{v}_x}$$

which for this specific case becomes:

$$f(1, \hat{\boldsymbol{\theta}}) \pm t_{97;0.99}\hat{\sigma}\sqrt{\hat{v}_1^T(\hat{\mathbb{V}}^T\hat{\mathbb{V}})^{-1}\hat{v}_1}$$

The gradient vector $\hat{v}_1$ calculated as:

$$\hat{v}_1 = [\frac{\partial f}{\partial \theta_1}(1, \hat{\boldsymbol{\theta}}), \frac{\partial f}{\partial \theta_2}(1, \hat{\boldsymbol{\theta}}), \frac{\partial f}{\partial \theta_3}(1, \hat{\boldsymbol{\theta}})]^T = [1, 0.3245506, -9.7842421]^T$$

and hence the approximate 98% confidence interval calculated as:

$$34.78367 < f(1, \hat{\boldsymbol{\theta}}) < 35.30855$$

The real mean response for x=1, $f(1, \boldsymbol{\theta})$ is 35 which lies in the calculated interval, something that verifies that the interval is indeed correct. The 98% confidence intervals also obtained using the bootstrap in order to conduct a comparison of accuracy. An approximate 98% bootstrap confidence interval with B=1000 iterations for the expected value of Y when x=1 is:

$$[2f(1, \hat{\boldsymbol{\theta}}) - f^*_{(1+\lfloor 0.99 \cdot 1000 \rfloor)}(1), 2f(1, \hat{\boldsymbol{\theta}}) - f^*_{(1+\lfloor 0.02 \cdot 1000 \rfloor)}]$$

which leads to the following bootstrap confidence intervals:

$$34.7745 < f(1, \hat{\boldsymbol{\theta}}) < 35.29297$$

It can be noted that the two confidence intervals are extremely similar and both of them are accurate, in the sense that the real mean response for x=1, $f(1, \boldsymbol{\theta}) = 35$ lies within these intervals. Furthermore, the 98% confidence intervals for many values of x in the interval (0,3] were calculated and plotted. The plot of these confidence intervals can be seen below:

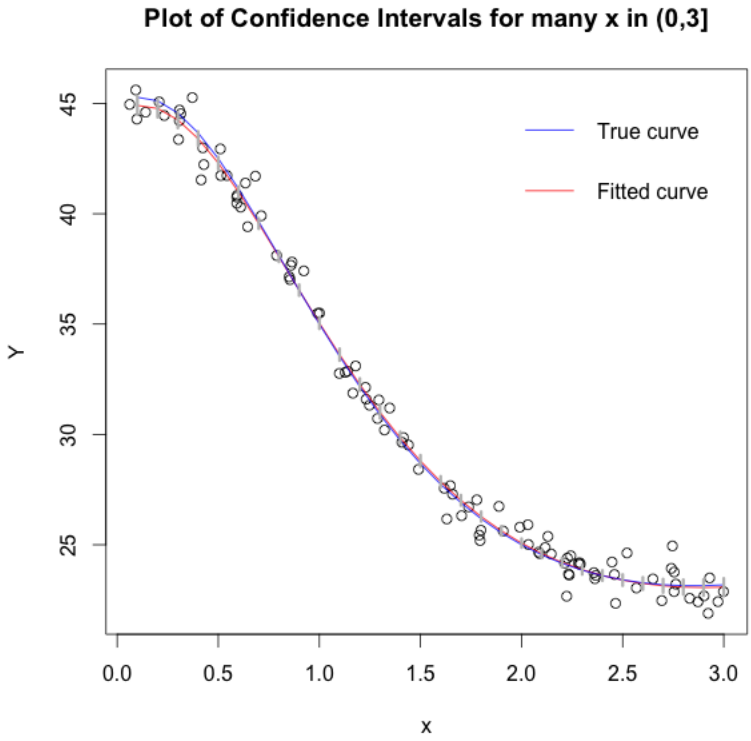


Figure 9: Plot of Confidence interval for many x's in (0,3].

From figure (9) it can be noticed that the true curve lies always within the approximate 98% confidence intervals something that indicates once again, that the accuracy of the model is high. However, this result is also not surprising, as the fitted curve is really similar to the true curve and thus, the confidence intervals were also expected to be accurate.

Problem 3

In order to investigate the impact of weeds on the productivity of a crop, two candidate functions can be used to model the relationship between the number of healthy crops (N) which produced in 56 plots (n) of land and the rate of weeds (R) found in each plot. Thus, two models can be created, where each one uses a different function:

$$Y_{i,1} = \delta + \frac{\alpha - \delta}{1 + e^{\beta log(\gamma r)}} + \varepsilon_{i,1} \qquad i = 1, \ldots, n$$

$$Y_{i,2} = \gamma + \alpha e^{-\beta r} + \varepsilon_{i,2} \qquad i = 1, \ldots, n$$

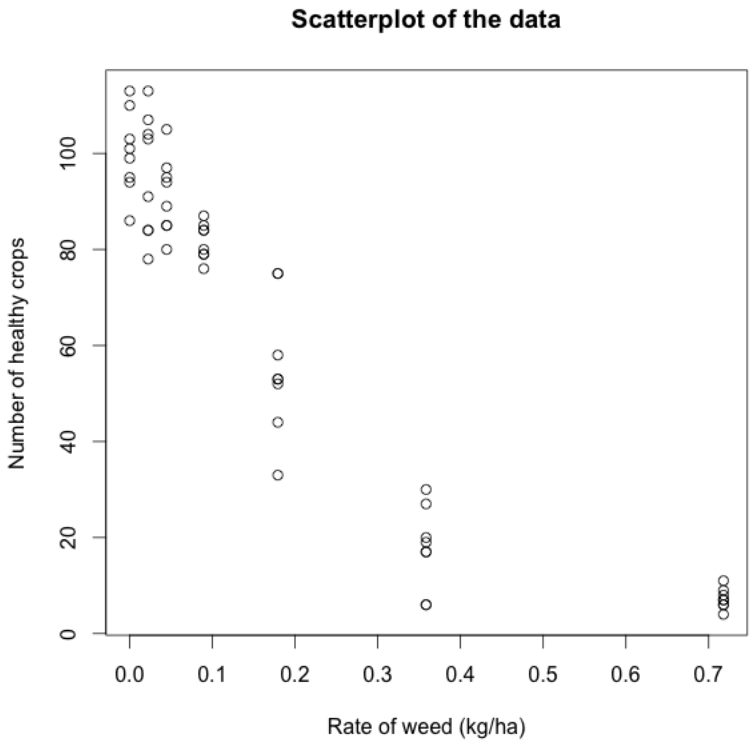


Figure 10: Scatterplot of the Number of healthy crops against the Rate of weed

From an observation of the scatterplot of the data in figure (10) it can be seen that the data decrease exponentially, so both the log-logistic function as well as the exponential function might be a good fit for these data. However, in order to confirm this assumption the estimates of the models' parameters have to be obtained and then the fitted curves have to be plotted along with the scatterplot of the *number* of the healthy crops against the *rate* of the weed. For the first model, using the *Gauss-Newton* method with initial values $\alpha = 20$, $\beta = 4$, $\gamma = 3$, $\delta = 5$ after 10 iterations the `nls` function produced the following parameter estimates and their corresponding p-values:

Parameter	Std. Error	Estimate	P-value
α	2.2526	97.09	$2 \cdot 10^{-16}$
β	0.3642	2.24	$1.12 \cdot 10^{-07}$
γ	0.4335	5.17	$2 \cdot 10^{-16}$
δ	5.4119	1.05	0.846

Table 4: Estimated parameters and p-values for the first model.

It can be noticed that while the first three parameters are significant, the parameter δ produced a p-value greater than the significance level $\alpha = 0.05$, which indicates that this parameter might be insignificant for the productivity of crop. Similarly, for the second model *Gauss-Newton* method with initial values $\alpha = 5$, $\beta = 3$, $\gamma = 4$ after 10 iterations produced the following estimates and p-values:

Parameter	Std. Error	Estimate	P-value
α	6.0464	110.03	$2 \cdot 10^{-16}$
β	0.5028	3.49	$5.5 \cdot 10^{-09}$
γ	6.4630	-5.22	0.422

Table 5: Estimated parameters and p-values for the second model.

Again, according to table (8) only the first two parameters are significant for the model, while γ has a p-value greater than the significance level. The fitted curves along with the scatterplot of the data have been plotted, in order to compare visually the fit of the two models.

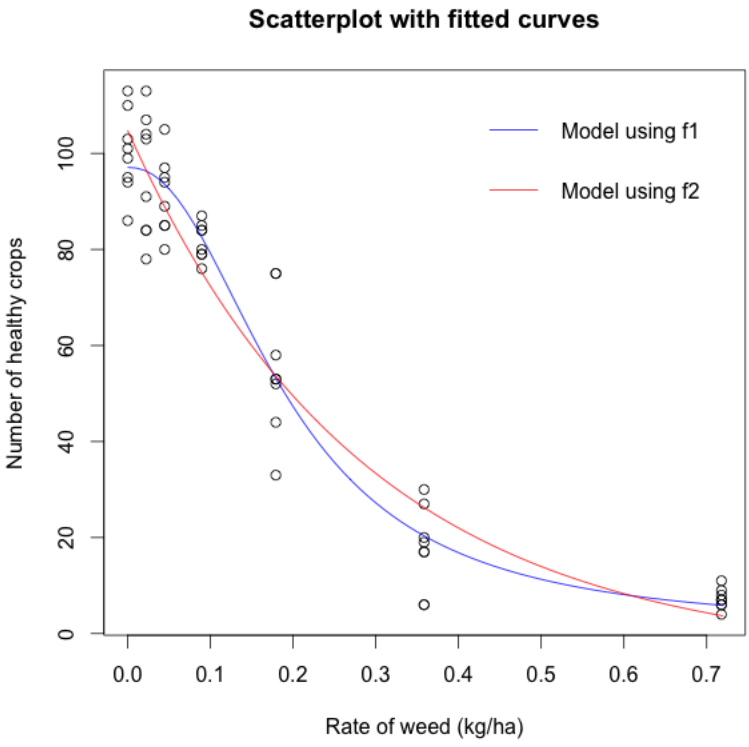
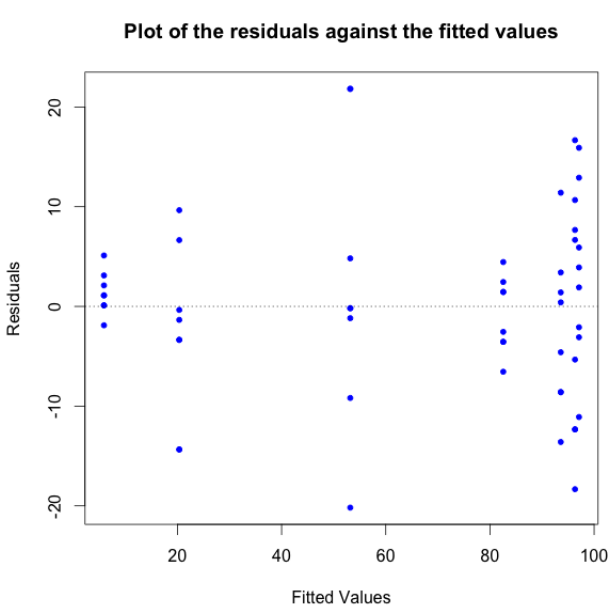
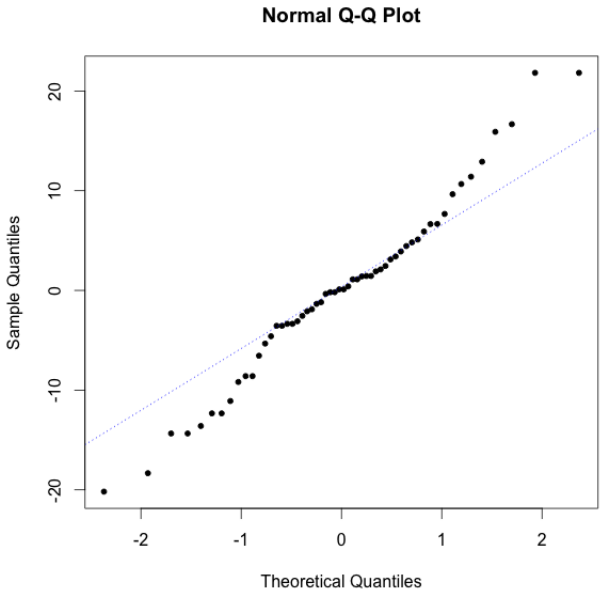


Figure 11: Scatterplot of the data along with fitted curves

From a visual comparison of the fitted curves in figure (11), it can be seen that both models fit the data in a nice way, as both of them follow the decay of the data points. However, the model with function 1 seems to fit the data in a better way than the model using function 2, as it decreases rapidly from 0.05 until 0.2 and after that point it starts to decrease more slowly. On the other hand the second function has a decreasing exponential curve which could be described as simpler than the curve of the first function. The second curve seems to fit the data slightly worse than the first one, as it does not cross the points where the most data points are concentrated, while the first one does. However the efficiency of each model has to be checked using information criteria for comparison. Moreover, before any further analysis, the assumptions of the models have to checked, in order to confirm if they can be assumed as valid.



(a) Plot of the residuals against the fitted values

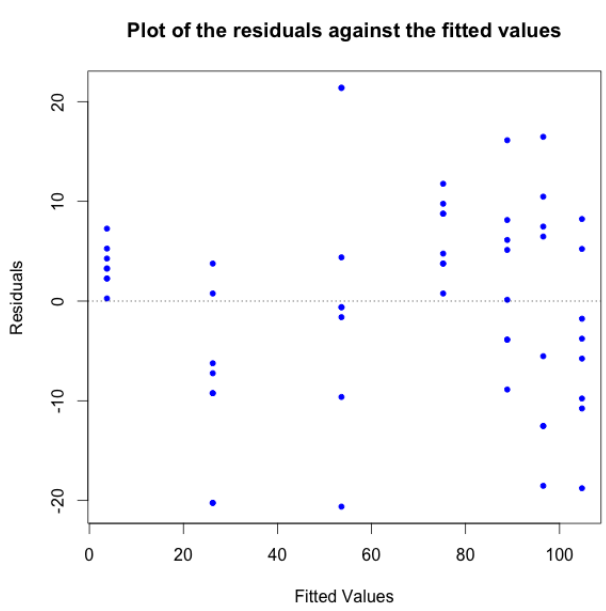


(b) Normal Q-Q plot of the residuals for confirmation of normality

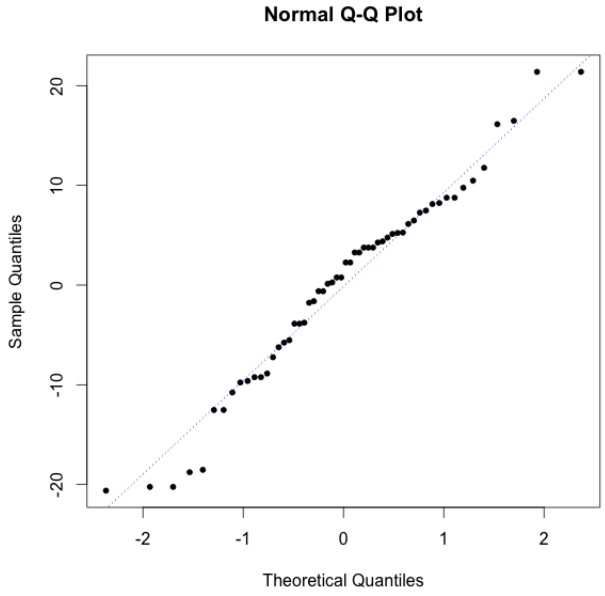
*Figure 12: Check for validity of normality and constant error variance of the residuals of the **first** model.*

From a plot of the residuals against the fitted values for the first model there is not a visible pattern in the points and thus, the assumption of constant error variance may be considered valid. From an observation on the normal quantile plot of the residuals, it can be seen that the majority of the points follows the line. There are some outliers, however they do not deviate too far from the line. The assumption of normality can be verified further by plotting a histogram of the residuals in figure (14a) where it seems that the distribution of the residuals is approximately normal. In addition, a *Shapiro-Wilk* normality test produced a p-value 0.42 which does not reject the normality assumption.

Concerning the second model, a plot of the residuals against the fitted values in figure (13a) does not show any particular pattern, so the assumption of constant error variance might be considered valid. However, the spread of the point around 0, is not very constant and the most possible reason for this is the limited number of observations in the data set. A normal quantile plot in figure (13b) along with a histogram in figure (14b) can verify that the residuals can be considered approximately normal. Furthermore a *Shapiro-Wilk* normality test produced a p-value 0.21 which verifies the normality assumption.

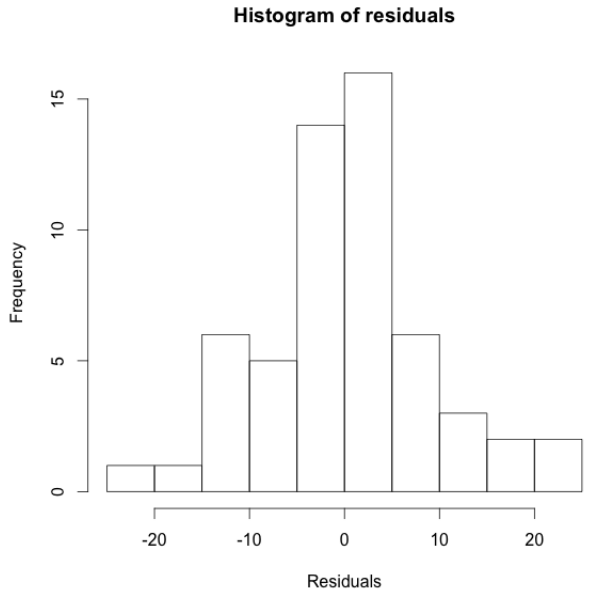


(a) Plot of the residuals against the fitted values

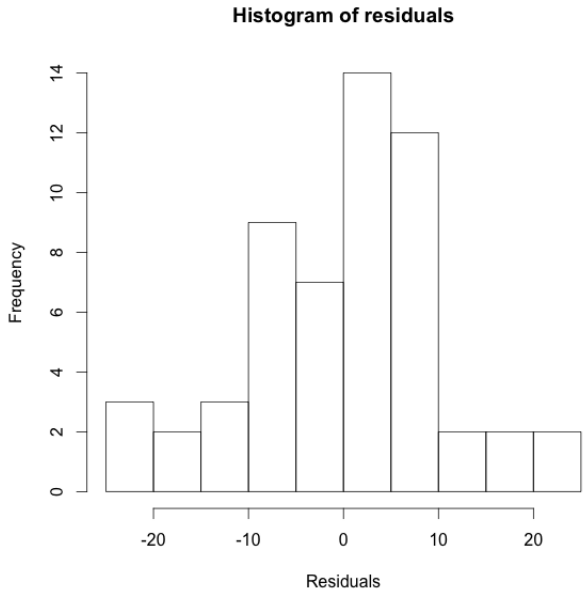


(b) Normal Q-Q plot of the residuals for confirmation of normality

Figure 13: Check for validity of normality and constant error variance of the residuals of the ***second*** *model.*



(a) Histogram of the residuals for the ***first*** *model*



(b) Histogram of the residuals for the ***second*** *model*

Figure 14: Histograms of the residuals

In order to compare these two non-nested models, information criteria such as Akaike Information Criterion (AIC) and Bayes Information Criterion (BIC) can be used. A comparison of the AIC and BIC values for both models is summarized in table (6).

Model	AIC	BIC
1	414.11	424.24
2	423.74	431.84

Table 6: AIC and BIC values for both models.

For both the information criteria the first model produced smaller values than the second, which means that the model which uses the first function can model the relationship between the *number* of healthy crops and the *rate* of weeds in a better way than the model which uses the second function. These information criteria, confirm the initial guess from the visual comparison of the two curves, that the first curve fits the data in a slightly better way than the second.

A further interesting investigation, would be to set the insignificant parameters of the two models to zero, in order to observe whether it improves the fit. Such an action results to the following models:

$$Y'_{i,1} = \frac{\alpha}{1 + e^{\beta log(\gamma r)}} + \varepsilon_{i,1} \qquad i = 1, \ldots, n$$

$$Y'_{i,2} = \alpha e^{-\beta r} + \varepsilon_{i,2} \qquad i = 1, \ldots, n$$

The new parameter estimates for the *reduced* models are:

Parameter	Std. Error	Estimate	P-value
α	2.1453	97.266717	$2 \cdot 10^{-16}$
β	0.2243	2.181393	$2.21 \cdot 10^{-07}$
γ	0.2807	5.111663	$2 \cdot 10^{-16}$

*Table 7: Estimated parameters and p-values for the first **reduced** model.*

Parameter	Std. Error	Estimate	P-value
α	2.4116	105.418925	$2 \cdot 10^{-16}$
β	0.2707	3.857306	$2 \cdot 10^{-16}$

*Table 8: Estimated parameters and p-values for the second **reduced** model.*

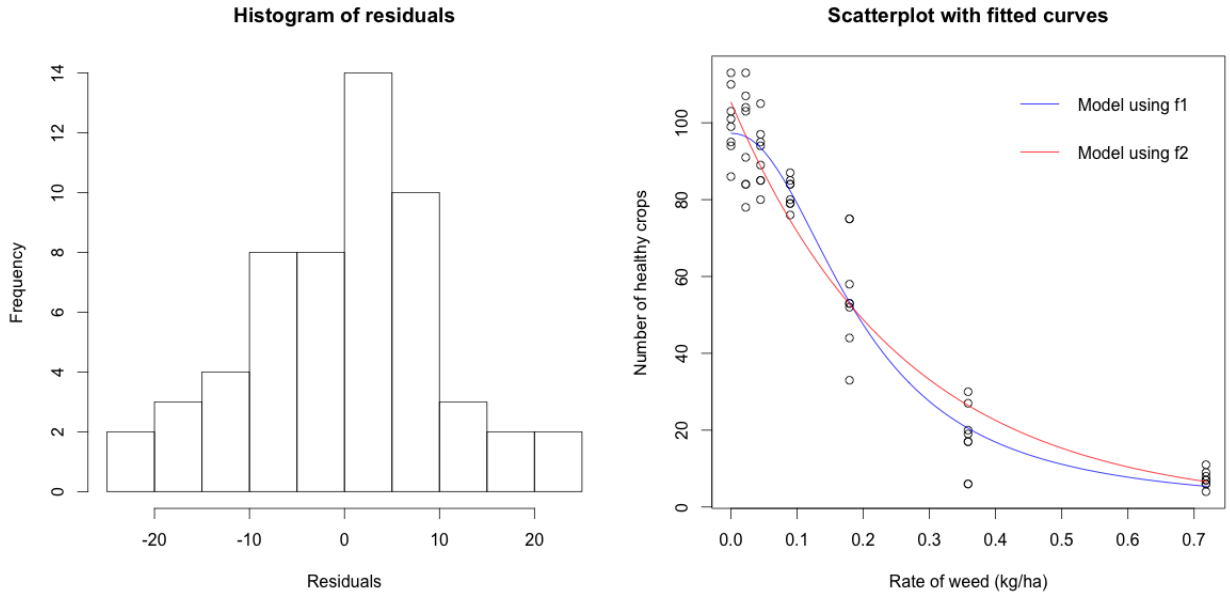
The parameter estimates are similar to the estimates of the full models, but this time all of them are significant. Moreover, the standard errors of the parameters of both reduced models are lower than in the full models. The assumptions of the residuals did not had any significant improvement, however there were some minor improvements in some of the plots. For example, the histogram of the second model is now slightly better, as it can be seen from figure (15a). The rest of the plots are skipped, due to similarity with the previous plots, but they are contained in the R code in Appendix.

Furthermore, the information criteria have been conducted again in order to observe improvements in their values. The results can be seen in the table (9).

Model	AIC	BIC
1	412.15	420.25
2	422.64	428.71

*Table 9: AIC and BIC values for both **reduced** models.*

It is clear, that the first model produced lower AIC and BIC values, so once again it can be considered that this is the model which produced the best fit for these data. Moreover, the values of both the reduced models are lower than the values of the full models (table 6). This observation verifies the fact, that indeed the reduced functions which include only the significant parameters can model the relationship between the *number* of healthy crops and the *rate*, in a better way than the full initial functions. Finally, a scatterplot with the fitted curves of the reduced models is plotted, in order to observe differences with the previous scatterplot. It can be seen from figure (15b) that while the first curve does not show any particular difference, this time the second curve does not cross the first in 0.6 and it decays more smoothly until 0.7. From this plot it is easier to visually observe that the first curve fits the data slightly better than the second one.



(a) Histogram of the residuals for the second ***reduced*** *model* *(b) Scatterplot of data along with fitted curves for the* ***reduced models***

Figure 15

Conclusively, the best of the created models can be considered to be the *first reduced* model, as it is the most accurate as well as more simple than its' full version. Furthermore, the residuals of the *first* model seem a bit more random and spread out than those of the second model and this is another indication that this model should be preferred.

II. Theoretical Problems

Problem 1

The *likelihood* function for the non-linear regression model is given by:

$$\mathcal{L}(\theta, Y_i, x) = \prod_{i=1}^{n}(\theta, Y_i, x) = \prod_{i=1}^{n}(2\pi\sigma^2)^{-1/2}exp(-\frac{1}{2\sigma^2}(Y_i - f(x_i, \theta))^2) \Leftrightarrow$$

$$\mathcal{L}(\theta, Y_i, x) = (2\pi\sigma^2)^{-n/2}exp(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - f(x_i, \theta))^2)$$

The *log-likelihood* function is:

$$\begin{aligned}
\ell(\theta, Y_i; x) &= log(\mathcal{L}(\theta, Y_i, x)) \\
&= log[(2\pi\sigma^2)^{-n/2}exp(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - f(x_i, \theta))^2] \\
&= log[(2\pi\sigma^2)^{-n/2}] + log[exp(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - f(x_i, \theta))^2] \\
&= -\frac{n}{2}log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - f(x_i, \theta))^2 \qquad (1)
\end{aligned}$$

The third equality holds because $log(a \cdot b) = loga + logb$ for a,b positive numbers. In this case both of the quantities are positive, so this formula holds. Concerning the relation of MLE and LSE, if the errors ε_i are normal: $\varepsilon_1, \ldots, \varepsilon_n \sim N(0, \sigma^2)$ then the least squares estimator is also the maximum likelihood estimator. In this case, the errors are assumed normal and thus, MLE equals LSE.
The maximum likelihood estimator for σ^2 is the value of σ^2 that maximizes $\ell(\theta, Y_i; x)$. In other words, σ^2_{mle} is a solution to:

$$\max_{\sigma^2} \ell(\theta, Y_i; x)$$

The maximum likelihood estimator for σ^2 can be found by differentiating $\ell(\theta, Y_i; x)$ with respect to σ^2 and set it equal to zero:

$$\begin{aligned}
\frac{\partial\ell(\hat{\theta}_{mle}, Y_i; x)}{\partial\sigma^2} = -\frac{n}{2}(\frac{2\pi}{2\pi\hat{\sigma}^2_{mle}}) + \frac{1}{2(\hat{\sigma}^2_{mle})^2}\sum_{i=1}^{n}(Y_i - f(x_i, \theta))^2 = 0 \Leftrightarrow \\
\frac{n}{2\hat{\sigma}^2_{mle}} = \frac{\sum_{i=1}^{n}(Y_i - f(x_i, \theta))^2}{2(\hat{\sigma}^2_{mle})^2} \Leftrightarrow \\
\hat{\sigma}_{mle} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - f(x_i, \theta))^2 \qquad (2)
\end{aligned}$$

Problem 2

In order for the model to be linear, the regression function must be able to be written in the form:

$$f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 \tag{3}$$

It can be noticed that by setting the parameter $\theta_2 = 0$ the model can be written in the form (3) as:

$$Y_i = \theta_0' + \theta_1' x_1 + \theta_3' x_3 + \varepsilon_i$$

with $\theta_0' = \theta_5, \theta_1' = \theta_1, \theta_3' = 2\theta_3, x_1 = x, x_3 = x^2$ and the model takes the form:

$$Y_i = \theta_1 x + 2\theta_3 x^2 + \theta_5 + \varepsilon_i \tag{4}$$

which is *linear with respect to* θ. Similarly, by setting the parameter $\theta_4 = 0$ the model takes the form:

$$Y_i = \theta_1 x + 2\theta_3 x^2 + \frac{\theta_2 x^3}{3 + e^{x^2}} + \theta_5 + \varepsilon_i \tag{5}$$

which is again *linear with respect to* θ. Then, the hypothesis that the model is in fact linear, is equivalent to the hypotheses that θ_2 and/or θ_4 are insignificant for the model, and thus they can be set equal to zero. Methods for checking the significance of θ_2 are provided below, the exact same methods hold also for θ_4.
One way to check if θ_2 can be set to zero is by comparing the sum of squares of the two nested models:

$$q = 5 \qquad \text{global model: } Y_{i,1} = \theta_1 x + 2\theta_3 x^2 + \frac{\theta_2 x^3}{3 + e^{-\theta_4 x + x^2}} + \theta_5 + \varepsilon_i$$

$$p = 4 \qquad \text{nested model: } Y_{i,1} = \theta_1 x + 2\theta_3 x^2 + \theta_5 + \varepsilon_i$$

It has to be noted, that the comparison can be done only if the normality assumptions for the errors are valid. In order for the comparison to take place, an F-test can be used, which is in fact a comparison of residual sum of squares. The hypotheses now are:

H_0: the smaller model with p=4 parameters holds
H_1: the global model with q=5 parameters holds

The reduced model is *adequate* for describing the data at significance level α if and only if:

$$\frac{[S(\hat{\theta}_p) - S(\hat{\theta}_q)]/(q-p)}{S(\hat{\theta}_q)/(n-q)} < \mathcal{F}_{q-p,n-q;1-\alpha}$$

which for this case where p=4 and q=5 becomes:

$$\frac{S(\hat{\theta}_4) - S(\hat{\theta}_5)}{S(\hat{\theta}_5)/(n-5)} < \mathcal{F}_{1,n-5;1-\alpha}$$

If this holds, then the reduced model can be used instead of the global model which means that the parameter θ_2 is insignificant and the model is in fact linear. The exact same test can be conducted also to the parameter θ_4 and the only difference will be that the nested model this time will have the form (5).

Alternatively, it can be checked whether the p-value of θ_2 parameter is greater than 0.1. This can be tested using the approximate 95% confidence intervals:

$$\hat{\theta}_2 \pm t_{n-5;0.975}\hat{\sigma}\sqrt{[\hat{\mathbb{V}}^T\hat{\mathbb{V}}^{-1}]_{22}}$$

The hypotheses now are:

$$H_0\text{: } \theta_2 = 0$$
$$H_1\text{: } \theta_2 \neq 0$$

If zero lies in the 95% confidence intervals it means that the null hypothesis holds and thus, θ_2 is insignificant and can be set to zero. If this is the case the model is in fact linear. The exact same test can be conducted to the parameter θ_4, in which case the confidence intervals are:

$$\hat{\theta}_4 \pm t_{n-5;0.975}\hat{\sigma}\sqrt{[\hat{\mathbb{V}}^T\hat{\mathbb{V}}^{-1}]_{44}}$$

Similarly, the *t-ratio* can be checked:

$$t - ratio : \frac{\theta_2 \; parameter \; estimate}{estimated \; standard \; error}$$

If the t-ratio is not significantly different from zero, then the parameter θ_2 can be considered insignificant and thus, equal to zero. The exact same test can be conducted also to the parameter θ_4.

Moreover, the correlation between θ_2 and any other parameters can be checked. If the correlation between θ_2 and any other parameter is high (>0.9) it means that the two parameters are highly dependent to each other, so one of the parameters has to be set equal to zero in order to prevent over-parametrization. If this is the case, θ_2 can again be set equal to zero and the model is linear. Exactly the same test can also be conducted to the parameter θ_4, by checking the correlation between θ_4 and the other parameters.

Problem 3

The normal equations can be obtained by differentiating f with respect to θ_ℓ in the sum of squares $S(\hat{\theta})$ and setting this derivative equal to zero.

$$S(\theta) = \sum_{i=1}^{n}(Y_i - f(x_i, \theta))^2$$

Hence the normal equations are:

$$-2\sum_{i=1}^{n}\frac{\partial f}{\partial \theta_\ell}(x_i, \theta)(Y_i - f(x_i, \theta)) = 0 \Leftrightarrow$$

$$\sum_{i=1}^{n}\frac{\partial f}{\partial \theta_\ell}(x_i, \theta)(Y_i - f(x_i, \theta)) = 0 \qquad \ell = 1, \ldots, p$$

In this specific case, $\ell = 2$ so the normal equations are the following:

$$\sum_{i=1}^{n} \frac{\partial f}{\partial \theta_1}(x_i, \theta)(Y_i - f(x_i, \theta)) = 0 \tag{6}$$

$$\sum_{i=1}^{n} \frac{\partial f}{\partial \theta_2}(x_i, \theta)(Y_i - f(x_i, \theta)) = 0 \tag{7}$$

But $\frac{\partial f}{\partial \theta_1}(x_i, \theta)$ and $\frac{\partial f}{\partial \theta_2}(x_i, \theta)$ are:

$$\frac{\partial f}{\partial \theta_1}(x_i, \theta) = \theta_1 cos(\theta_1 x) + e^{-\theta_2 x_i}$$
$$\frac{\partial f}{\partial \theta_2}(x_i, \theta) = -\theta_1 x_i e^{-\theta_2 x_i}$$

Hence the normal equations become:

$$\sum_{i=1}^{n} (\theta_1 cos(\theta_1 x_i) + e^{-\theta_2 x_i})(Y_i - f(x_i, \theta)) = 0 \tag{8}$$

$$\sum_{i=1}^{n} (-\theta_1 x_i e^{-\theta_2 x_i})(Y_i - f(x_i, \theta)) = 0 \tag{9}$$

Solving these equations, $\hat{\theta}_1$ and $\hat{\theta}_2$ which minimize the sum of squares can be found.
In the case that $n = 200, x_1 = x_2 = \ldots = x_{100} = 0$ and $x_{101} = x_{102} = \ldots = x_{200} = 1$ equations (8) and (9) become:

$$\sum_{i=1}^{100} (\hat{\theta}_1 cos(0) + e^0)(Y_i - f(0, \theta)) + \sum_{i=101}^{200} (\hat{\theta}_1 cos(\hat{\theta}_1) + e^{-\hat{\theta}_2})(Y_i - f(1, \hat{\theta})) = 0 \Leftrightarrow$$
$$\sum_{i=1}^{100} (\hat{\theta}_1 + 1)(Y_i - f(0, \theta)) + \sum_{i=101}^{200} (\hat{\theta}_1 cos(\hat{\theta}_1) + e^{-\hat{\theta}_2})(Y_i - f(1, \hat{\theta})) = 0 \Leftrightarrow$$
$$(\hat{\theta}_1 + 1) \sum_{i=1}^{100} (Y_i - f(0, \theta)) + (\hat{\theta}_1 cos(\hat{\theta}_1) + e^{-\hat{\theta}_2}) \sum_{i=101}^{200} (Y_i - f(1, \hat{\theta})) = 0 \tag{10}$$

$$\sum_{i=1}^{100} 0 + \sum_{i=101}^{200} (-\hat{\theta}_1 e^{-\hat{\theta}_2})(Y_i - f(1, \hat{\theta})) = 0 \Leftrightarrow$$
$$\sum_{i=101}^{200} (-\hat{\theta}_1 e^{-\hat{\theta}_2})(Y_i - f(1, \hat{\theta})) = 0 \Leftrightarrow$$
$$(-\hat{\theta}_1 e^{-\hat{\theta}_2}) \sum_{i=101}^{200} (Y_i - f(1, \hat{\theta})) = 0 \tag{11}$$

In the equation (11) $\hat{\theta}_1$ and $e^{-\hat{\theta}_2}$ are non negative and thus,

$$\sum_{i=101}^{200}(Y_i - f(1,\hat{\theta})) = 0 \tag{12}$$

$$\sum_{i=101}^{200} Y_i - \sum_{i=101}^{200} f(1,\hat{\theta}) = 0 \Leftrightarrow$$

$$\sum_{i=101}^{200} Y_i = \sum_{i=101}^{200} f(1,\hat{\theta}) \tag{13}$$

But, the left hand side can be written in terms of the grand mean μ because:

$$\frac{\sum_{i=1}^{n} Y_i}{n} = \mu \Leftrightarrow \sum_{i=1}^{n} Y_i = \mu n$$

So equation (13) becomes:

$$\begin{aligned} 100\mu_2 &= 100 f(1,\hat{\theta}) \Leftrightarrow \\ \mu_2 &= f(1,\hat{\theta}) \Leftrightarrow \\ \mu_2 &= sin(\hat{\theta_1}) + \hat{\theta}_1 e^{-\theta_2} \end{aligned} \tag{14}$$

where $\mu_2 = \frac{Y_{101}+\ldots+Y_{200}}{100}$ is the mean value of the second half of the observations (i.e. the mean value of the observations $Y_{101}, \ldots, Y_{200}$). Now, using equation (12) in equation (10), the latter becomes:

$$(\hat{\theta}_1 + 1)\sum_{i=1}^{100}(Y_i - f(0,\theta)) = 0$$

This equation has 2 possible solutions: either $\hat{\theta}_1 = -1$ or:

$$\sum_{i=1}^{100}(Y_i - f(0,\theta)) = 0 \Leftrightarrow$$

$$\sum_{i=1}^{100} Y_i - \sum_{i=1}^{100} f(0,\theta) = 0 \Leftrightarrow$$

$$\sum_{i=1}^{100} Y_i = \sum_{i=1}^{100} f(0,\theta) \tag{15}$$

Using $\sum_{i=1}^{100} Y_i = \mu_1 n_1$ equation (15) becomes:

$$\begin{aligned} 100\mu_1 &= 100 f(0,\hat{\theta}) \Leftrightarrow \\ sin0 + \hat{\theta}_1 e^0 &= \mu_1 \Leftrightarrow \\ \hat{\theta}_1 &= \mu_1 \end{aligned} \tag{16}$$

Where $\mu_1 = \frac{Y_1+\ldots+Y_{100}}{100}$ is the mean value of the first 100 observations, which is a known quantity given the real data. Now for the grand mean it holds that:

$$\begin{aligned}
\mu &= \frac{\mu_1 + \mu_2}{2} \overset{(16),(14)}{\Longleftrightarrow} \\
\mu &= \frac{\hat{\theta}_1 + sin(\hat{\theta}_1) + \hat{\theta}_1 e^{-\theta_2}}{2} \Leftrightarrow \\
2\mu &= \hat{\theta}_1 + sin(\hat{\theta}_1) + \hat{\theta}_1 e^{-\hat{\theta}_2} \Leftrightarrow \\
\hat{\theta}_1 e^{-\hat{\theta}_2} &= 2\mu - \hat{\theta}_1 - sin(\hat{\theta}_1) \Leftrightarrow \\
e^{-\hat{\theta}_2} &= \frac{2\mu - \hat{\theta}_1 - sin(\hat{\theta}_1)}{\hat{\theta}_1} \Leftrightarrow \\
\hat{\theta}_2 &= -log(\frac{2\mu - \hat{\theta}_1 - sin(\hat{\theta}_1)}{\hat{\theta}_1}) \qquad (17)
\end{aligned}$$

Concerning $\hat{\theta}_1$, the solution $\hat{\theta}_1 = -1$ is not possible to hold, due to the fact that this is a constant value, which does not depend on the values of the real observations. On the other hand, $\hat{\theta}_1 = \mu_1$ depends on the real observations, as it is the mean value of the first 100 observations, so this value is much more meaningful than the constant -1.

Using $\hat{\theta}_1 = \mu_1$, $\hat{\theta}_2$ can be determined from equation (17) as follows:

$$\hat{\theta}_2 = -log(\frac{2\mu - \mu_1 - sin(\mu_1)}{\mu_1}) \qquad (18)$$

In fact, these values for the parameters $\hat{\theta}_1$ and $\hat{\theta}_2$ are the *final estimates* which are produced by *Gauss-Newton* method (the R code which reproduces this theoretical part and shows that these values are indeed the final estimates, can be found in the Appendix). Now, when it comes to initial values, in order for the iteration method to converge smoothly, one can choose as initial value for $\hat{\theta}_1$, a value which is close to μ_1, and then using equation (17) a starting value for $\hat{\theta}_2$ can be determined. For example, if the mean value of the first half of the observations is $\mu_1 = 50$, while the grand mean is $\mu = 25$, then an appropriate initial value for $\hat{\theta}_1$ would be $\hat{\theta}_1 = 40$ and then using equation (17) it can be found that an appropriate starting value for $\hat{\theta}_2$ is $\hat{\theta}_2 = 1.46$.

Appendix

```
### Exercise 1 ###

library(MASS)
library(ggplot2)
data  = muscle

#scatterplot of the data
plot(Length ~ Conc ,xlab="CaCl concentration",
ylab="Length of muscle section",
main= "Scatterplot of the data", pch=20, cex=1, col="blue", data = data)



#fit model

data.fit = nls(Length ~ th1 + th2 * exp(-Conc/th3),
 data, list(th1 = 10, th2 = -5, th3 = 5))
summary(data.fit)

coef = coef(data.fit)

plot(Length ~ Conc ,xlab="CaCl concentration",
ylab="Length of muscle section",
main= "Data along with fitted curve",
 pch=20, cex=1, col="blue", data = data)
x=seq(from=0.25,to=4,by=0.01)
lines(x,coef[1] + coef[2]*exp(-x/coef[3]),col="red")
res1 = residuals(data.fit)

#Second model

data2.fit = nls(Length ~ th1[Strip] +
th2[Strip] * exp(-Conc/th3), data, list(th1 = rep(coef[1],21),
 th2 = rep(coef[2],21), th3 = coef[3]))
coef2 = coef(data2.fit)
summary(data2.fit)

res2 = residuals(data2.fit)

#Compare curves of different animals


#same theta1

x3 = seq(from=1,to=4,by=0.01)
plot(data$Conc[data$Strip == "S16"],
```

```
 data$Length[data$ Strip == "S16"],
 pch=20, col="blue",ylim=c(2, 35), xlim=c(0, 4), xlab= "CaCl Concentration",
 ylab = "Length of Muscle section",main = 'Animals with same theta1')
legend('bottomright',
legend = c('Animal 16','Animal 17'), lty=c(1,1),lwd=c(1,1),
col=c('blue','black'),bty = 'n')
lines(x3, coef2[16] + coef2[37] * exp(-x3 / coef2[43]),
 col="blue ")


points(data$Conc[data$Strip == "S17"],
 data$Length[data$ Strip == "S17"], pch=20, col="black")
lines(x3, coef2[17] + coef2[38] * exp(-x3 / coef2[43]),
 col="black ")

#same theta2
plot(data$Conc[data$Strip == "S02"],
 data$Length[data$ Strip == "S02"], pch=20,
 col="red",ylim=c(2, 35), xlim=c(0, 4),
 xlab= "CaCl Concentration",
  ylab = "Length of Muscle section",
  main = 'Animals with same theta2')
legend('bottomright',
legend = c('Animal 2','Animal 21'), lty=c(1,1),lwd=c(1,1),
col=c('red','green'),bty = 'n')
lines(x3, coef2[2] + coef2[23] * exp(-x3 / coef2[43]), col="red ")

points(data$Conc[data$Strip == "S21"],
 data$Length[data$ Strip == "S21"], pch=20, col="green")
lines(x3, coef2[21] + coef2[42] * exp(-x3 / coef2[43]), col="green ")

#Check Assumptions
#1st model
plot(fitted(data.fit),res1,
xlab="Fitted Values",ylab="Residuals",
main= "Plot of the residuals against the fitted values",
 pch=20, cex=1, col="blue") # not good
abline(a=0, b=0, lty= 2)
hist(res1)
hist(res2)
qqnorm(summary(data.fit)$res, cex = 1, pch= 20)
qqline(summary(data.fit)$res,lty=3,col="blue")


shapiro.test(summary(data.fit)$res)# H0:normality
#2nd Model
plot(fitted(data2.fit),
res2,xlab="Fitted Values",ylab="Residuals",
```

```
main= "Plot of the residuals against the fitted values",
 pch=20, cex=1, col="blue")
abline(a=0, b=0, lty= 2)


qqnorm(summary(data2.fit)$res, cex = 1, pch= 20)
qqline(summary(data2.fit)$res,lty=3,col="blue")


shapiro.test(summary(data2.fit)$res)# H0:normality
```

```
### Exercise 2 ###

n = 100
mean=0
sd = sqrt(0.5)

th1 = 5
th2 = 90
th3 = 2

theta_real = c(th1,th2,th3)



set.seed(88)
x = runif(n, 0, 3)
e = rnorm(100,mean,sd)

#create vector Y

Y = th1*x + th2/(th3 + x^2) + e


plot(Y~  x, main = 'Scatterplot of simulated data')
curve(th1*x+(th2/(th3+x^2)), col="blue", add=TRUE)


#fit model

fit = nls(Y ~ theta1*x+(theta2/(theta3+x^2)),
start = list(theta1 = 10, theta2 = 30, theta3 = 9))
coef = coef(fit)

#new curve
plot(Y ~  x, main = 'Scatterplot of simulated data with fitted curve')
legend('topright', legend = c('True curve','Fitted curve'), lty=c(1,1),lwd=c(1,1),
col=c('blue','red'),bty = 'n')
#using lines
x1=seq(from=0.0,to=3,by=0.01)
lines(x1,coef[1]*x1 + coef[2]/(coef[3] + x1^2),col="red")#fitted curve
lines(x1,th1*x1 + th2/(th3 + x1^2),col="blue")#true curve


#the estimated covariance matrix
res = residuals(fit)
SSE = sum((res)^2)
p=3
```

```
var = SSE/(n-p)
var
cov = vcov(fit)
cov

#real covariance matrix
expr = expression(th1*x + th2/(th3 + x^2))
#compute partial derivatives with respect to th1, th2 and th3 using function
der1 = eval(deriv(expr, 'th1'))
der2 = eval(deriv(expr, 'th2'))
der3 = eval(deriv(expr, 'th3'))
V =matrix(NA, nrow = 100, ncol =3)

#here we use attr(,'gradient)
#to obtain the appropriate partial derivatives from deriv function

V =(cbind(attr(der1,'gradient'),attr(der2,'gradient'),attr(der3,'gradient')))
#calculate real covariance matrix
cov_real =  sd^2 * solve((t(V) %*% V))
cov_real


#check residuals

plot(fitted(fit),res,xlab="Fitted Values",
ylab="Residuals",main= "Plot of the residuals against the fitted values", pch=20,
cex=1, col="blue") # not good
abline(a=0, b=0, lty= 3)


# qq-plot
qqnorm(res, cex = 1, pch= 20)
qqline(res,lty=3,col="blue")
hist(res)   # not good


shapiro.test(res)# H0:normality

# confidence intervals using Asymptotic normality
f=function(x,theta)return(theta[1]*x+(theta[2]/(theta[3]+x^2)))


ub1 = coef[1] + qt(0.99,n-3)*sqrt(cov[1,1])
lb1 = coef[1] - qt(0.99,n-3)*sqrt(cov[1,1])

ub2 = coef[2] + qt(0.99,n-3)*sqrt(cov[2,2])
lb2 = coef[2] - qt(0.99,n-3)*sqrt(cov[2,2])
```

```
ub3 = coef[3] + qt(0.99,n-3)*sqrt(cov[3,3])
lb3 = coef[3] - qt(0.99,n-3)*sqrt(cov[3,3])

v1 = c(lb1,ub1)
v2 = c(lb2,ub2)
v3 = c(lb3,ub3)

interval = rbind (v1,v2,v3)
interval


lb=numeric(3);
ub=numeric(3);
rownames(lb)=names(coef(fit))
for(i in 1:3) {lb[i]=coef(fit)[i]-qt(0.99,n-length(coef(fit)))*sqrt(cov[i,i])
               ub[i]=coef(fit)[i]+qt(0.99,n-length(coef(fit)))*sqrt(cov[i,i])}
ci=cbind(lb,ub); rownames(ci)=names(coef(fit)); ci



#using bootstrap



B=1000
par.boot=matrix(NA,B,length(coef(fit)))
rownames(par.boot)=paste("B",1:B,sep="")
colnames(par.boot)=names(coef(fit))
e_tilde = res - mean(res)

for(b in 1:B){

  # Bootstrap samples from centered residuals
  res_tilde=sample(e_tilde,replace=T)
  # Calculate bootstrap values for the response
  yboot=fitted(fit)+res_tilde #Y*_1,...Y*_n
  # Fit model using new response and get bootstrap estimates for parameter
  modelBoot=nls(yboot ~ theta1*x+(theta2/(theta3+x^2)),
  start=list(theta1 = 10, theta2 = 30, theta3 = 9))
  # Store estimated (by bootstrap) parameters
  par.boot[b,]=coef(modelBoot) #\theta*_1,..., \theta*_B
}
# Compute and display bootstrap confidence interval for the coordinates of theta
lb.boot=2*coef(fit)-apply(par.boot,2,quantile,prob=0.99)
ub.boot=2*coef(fit)-apply(par.boot,2,quantile,prob=0.01)
cbind(lb.boot,ub.boot)
```

```
#Confidence interval for the expected value of Y when x = 1

## the real mean response f(1,theta)
f1=f(1,theta_real)
f1
## confidence interval for the mean response f(1,theta)
grad<-function(x,theta){rbind(x,
                              1/(theta[3] + x^2),
                              -theta[2]/(theta[3]+x^2)^2)}
gradvec=grad(1,coef(fit))
se=sqrt(t(gradvec)%*%vcov(fit)%*%gradvec)
lb=f1-qt(0.99,n-3)*se
ub=f1+qt(0.99,n-3)*se
c(lb,ub) # approximate confidence interval for f(1,theta)


# estimates and confidence approximate intervals of
# the mean response for many x's from (0,3]
x1=seq(from=0.1,to=3,by=0.1)
f2=f(x1,coef(fit))
grad<-function(x1,theta){rbind(x1,
                              1/(theta[3] + x1^2),
                              -theta[2]/(theta[3]+x1^2)^2)}
gradvec=grad(x1,coef(fit))
se<-sqrt(apply(gradvec,2,function(xx) t(xx)%*%vcov(fit)%*%xx))


#se=sqrt(t(gradvec)%*%vcov(fit)%*%gradvec)
lb=f2-qt(0.99,n-length(coef(fit)))*se
ub=f2+qt(0.99,n-length(coef(fit)))*se
c(lb,ub)

#plot confidence intervals


plot(Y ~  x, main = 'Plot of Confidence Intervals for many x in (0,3]')
legend('topright', legend = c('True curve','Fitted curve'), lty=c(1,1),lwd=c(1,1),
col=c('blue','red'),bty = 'n')
#using lines
lines(x1,coef[1]*x1 + coef[2]/(coef[3] + x1^2),col="red")#fitted curve
lines(x1,th1*x1 + th2/(th3 + x1^2),col="blue")#true curve
segments(x1,lb,x1,ub,lty=1,lwd =3, col="grey") # confidence intervals



#confidence interval for the mean response f(1,theta) using bootstrap
```

```
B=1000
par.boot=matrix(NA,B,length(1))
rownames(par.boot)=paste("B",1:B,sep="")
e_tilde = res - mean(res)

for(b in 1:B){
  res_tilde=sample(e_tilde,replace=T)
  # Calculate bootstrap values for the response
  yboot=fitted(fit)+res_tilde #Y*_1,...Y*_n
  # Fit model using new response and get bootstrap estimates for parameter
  modelBoot=nls(yboot ~ theta1*x+(theta2/(theta3+x^2)),
  start=list(theta1 = 10, theta2 = 30, theta3 = 9))
  par.boot[b,] = c(f(1,coef(modelBoot)))
}
lb.boot=2*f1-apply(par.boot,2,quantile,prob=0.99)
ub.boot=2*f1-apply(par.boot,2,quantile,prob=0.01)
cbind(lb.boot,ub.boot)
```

```
### Exercise 3 ###

weeds = read.table("Weeds.txt",header=TRUE)



#scatterplot of data

plot(weeds$Rate,weeds$Number,xlab="Rate of weed (kg/ha)",
ylab="Number of healthy crops", main = 'Scatterplot of the data')

form1=as.formula(Number~d + (a - d)/(1 + exp(b*log(g*Rate))))
form2=as.formula(Number~g + a*exp(-b*Rate))




f1=function(r,a,b,g,d)return(d + (a - d)/(1 + exp(b*log(g*r))))
f2=function(r,a,b,g)return(g + a*exp(-b*r))

#fit model using f1

model1 = nls(form1,data = weeds, start=c(a=20,b=4,g=3,d=5))
summary(model1)
coef1 = coef(model1)
coef1

plot(weeds$Rate,weeds$Number,xlab="Rate of weed (kg/ha)",
ylab="Number of healthy crops", main = 'Scatterplot with fitted curves')
legend('topright', legend = c('Model using f1','Model using f2'), lty=c(1,1),
lwd=c(1,1),col=c('blue','red'),bty = 'n')

#using lines
x=seq(from=0.0,to=0.72,by=0.01)
lines(x,f1(x,coef1[1],coef1[2],coef1[3],coef1[4]),col="blue")


#fit model using f2

model2 = nls(form2, data=weeds, start = c(a=5,b=3,g=4))
summary(model2)
coef2 = coef(model2)
coef2
lines(x,f2(x,coef2[1],coef2[2],coef2[3]),col="red")


#use AIC and BIC to determine which model performs better
AIC(model1)
```

```
AIC(model2)
BIC(model1)
BIC(model2)
#1st model wins!

#Check assumptions

#1st model

plot(fitted(model1),residuals(model1),xlab="Fitted Values",ylab="Residuals",
main= "Plot of the residuals against the fitted values", pch=20, cex=1, col="blue")
abline(a=0, b=0, lty= 3)


# qq-plot
qqnorm(residuals(model1), cex = 1, pch= 20)
qqline(residuals(model1),lty=3,col="blue")
hist(residuals(model1),xlab = 'Residuals', ylab = 'Frequency',
main = 'Histogram of residuals')   # not good


shapiro.test(residuals(model1))# H0:normality


#2nd model
plot(fitted(model2),residuals(model2),xlab="Fitted Values",ylab="Residuals",
main= "Plot of the residuals against the fitted values", pch=20, cex=1, col="blue")
abline(a=0, b=0, lty= 3)


# qq-plot
qqnorm(residuals(model2), cex = 1, pch= 20)
qqline(residuals(model2),lty=3,col="blue")
hist(residuals(model2),xlab = 'Residuals', ylab = 'Frequency',
main = 'Histogram of residuals')   # not good


shapiro.test(residuals(model2))# H0:normality

#new models with deleted parameters

#1st model with deleted d

form3=as.formula(Number~(a)/(1 + exp(b*log(g*Rate))))
f3=function(r,a,b,g)return((a)/(1 + exp(b*log(g*r))))
f4=function(r,a,b)return(a*exp(-b*r))

model3 = nls(form3,data = weeds, start=c(a=20,b=4,g=3))
```

```
summary(model3)
coef3 = coef(model3)
coef3

AIC(model3) #better than before
BIC(model3)

plot(fitted(model3),residuals(model3),xlab="Fitted Values",ylab="Residuals",
main= "Plot of the residuals against the fitted values", pch=20, cex=1, col="blue")
abline(a=0, b=0, lty= 3)


# qq-plot
qqnorm(residuals(model3), cex = 1, pch= 20)
qqline(residuals(model3),lty=3,col="blue")
hist(residuals(model3),xlab = 'Residuals', ylab = 'Frequency',
main = 'Histogram of residuals')   # not good

#2nd model with deleted g

form4=as.formula(Number~a*exp(-b*Rate))
model4 = nls(form4,data = weeds, start=c(a=50,b=4))
summary(model4)
coef4 = coef(model4)
coef4

AIC(model4)
BIC(model4)

plot(fitted(model4),residuals(model4),xlab="Fitted Values",ylab="Residuals",
main= "Plot of the residuals against the fitted values", pch=20, cex=1, col="blue")
abline(a=0, b=0, lty= 3)


# qq-plot
qqnorm(residuals(model4), cex = 1, pch= 20)
qqline(residuals(model4),lty=3,col="blue")
hist(residuals(model4),xlab = 'Residuals', ylab = 'Frequency',
main = 'Histogram of residuals')


#plot with fitted curves

plot(weeds$Rate,weeds$Number,xlab="Rate of weed (kg/ha)",ylab="Number of healthy crops",
main = 'Scatterplot with fitted curves')
legend('topright', legend = c('Model using f1','Model using f2'), lty=c(1,1),
lwd=c(1,1),col=c('blue','red'),bty = 'n')
```

```
x=seq(from=0.0,to=0.72,by=0.01)
lines(x,f3(x,coef3[1],coef3[2],coef3[3]),col="blue")

lines(x,f4(x,coef4[1],coef4[2]),col="red")




###### Code for the 3rd Theoretical Problem ########

n=200
x = numeric(n)
for(i in 1:100){
  x[i]=0
}
for(i in 101:200){
  x[i]=1
}
e = rnorm(200,0,2)

th1=50
th2=5

Y = sin(th1*x) + th1*exp(-th2*x) + e
plot(Y~  x, main = 'Scatterplot of simulated data')
curve(sin(th1*x) + th1*exp(-th2*x), col="blue", add=TRUE)

fit = nls(Y ~ sin(theta1*x) + theta1*exp(-theta2*x),
start = list(theta1 = 40, theta2 = 1.46))
summary(fit)
coef = coef(fit)
coef
k=0
for(i in 1:100){
 k =  k + Y[i]
}
mu1 = k/100
mu1 # equal to the final theta1 estimate!!!!

theta2estimate = -log((2*mean(Y) - mu1 -sin(mu1))/mu1) #equal to the final
# theta2 estimate!!!
theta2estimate
```