Contents

Task 1: Simple regression. Get a univariate dataset from sources 1

For this task a dataset “trees” was selected to build the simple regression between the girth and the volume of a tree.

a) Build a simple regression model (command lm). Provide the estimates of the model's parameters. Draw the scatter plot and the regression line.

The lm command in R resulted into the following simple linear regression line: y(volume) = -36.94 + 5.0659*x(girth)

According to the scatter plot below, the regression line almost perfectly suits the data with a little dispersity.

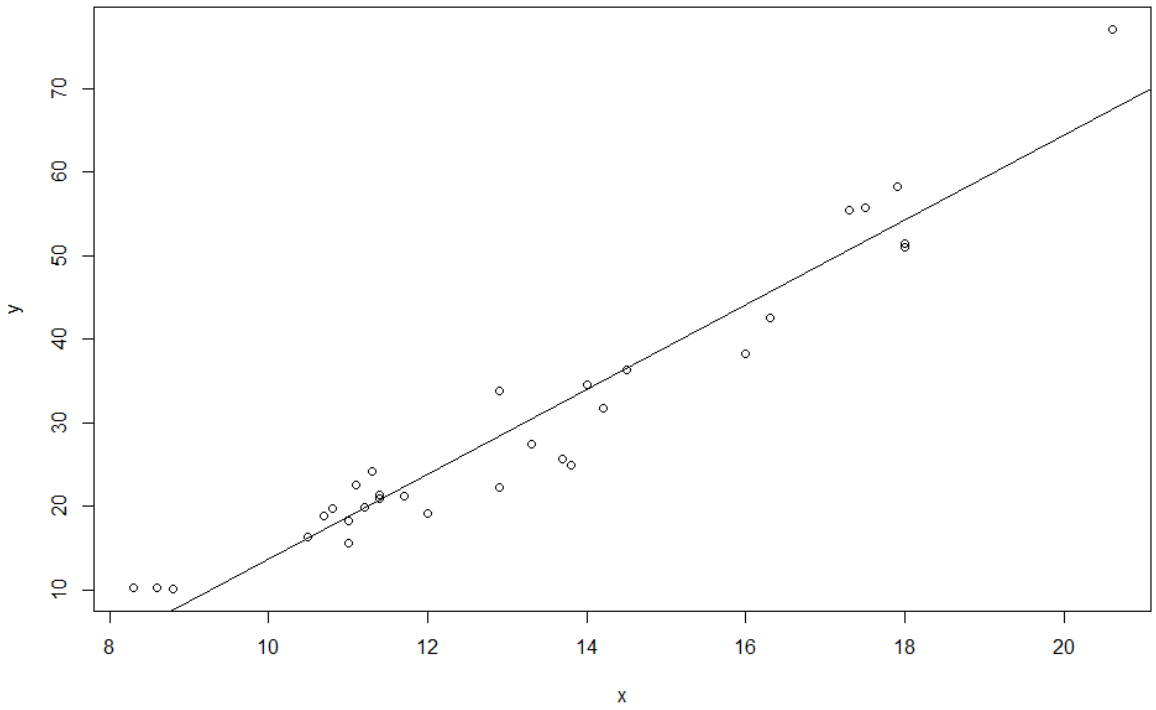

Figure 1, scatter plot of girth and volume along with regression line

b) Analyze the summary statistics (command summary()) focusing on:

i. The t-test for the slope. Explain.

ii. The F-test. Explain.

iii. R2 coefficient. Explain.

The p-values of both intercept (7.62e-12) and slope (< 2e-16) are very low that indicates the rejection of H_0 (intercept and slope are equal to 0).

The F-statistic p-value: < 2.2e-16 means the same as t-test that the H_0 (slope is equal to 0) can be rejected in favor of H_1 (slope is not equal to 0).

R-squared: 0.9331 which means that the data points lie almost perfectly on the regression line.

c) Plot the residuals against fitted values and comment on the model's adequacy. Examine the qq-plot for the residuals. Plot Cook's distances of the model. Explain.

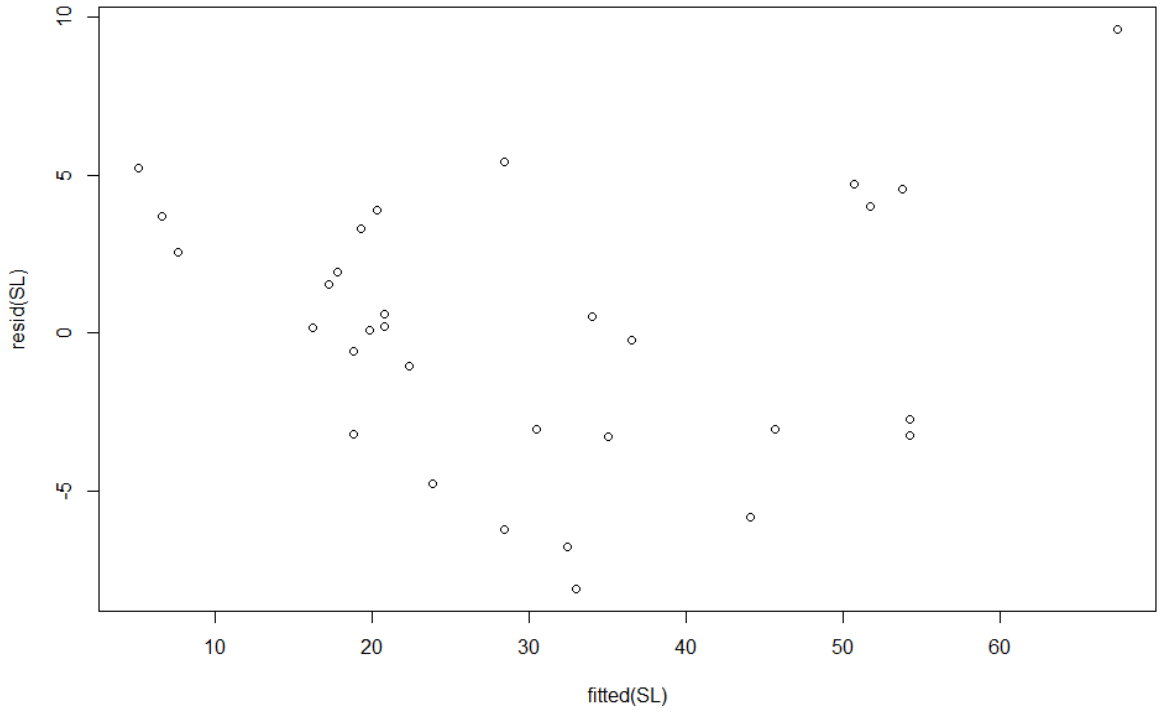

Figure 2, residuals against fitted values

From the figure above it can be concluded that the residuals (distances from actual data points to regression line) are on an acceptable level except of a few outliers, approximately (+,-) 5 difference.

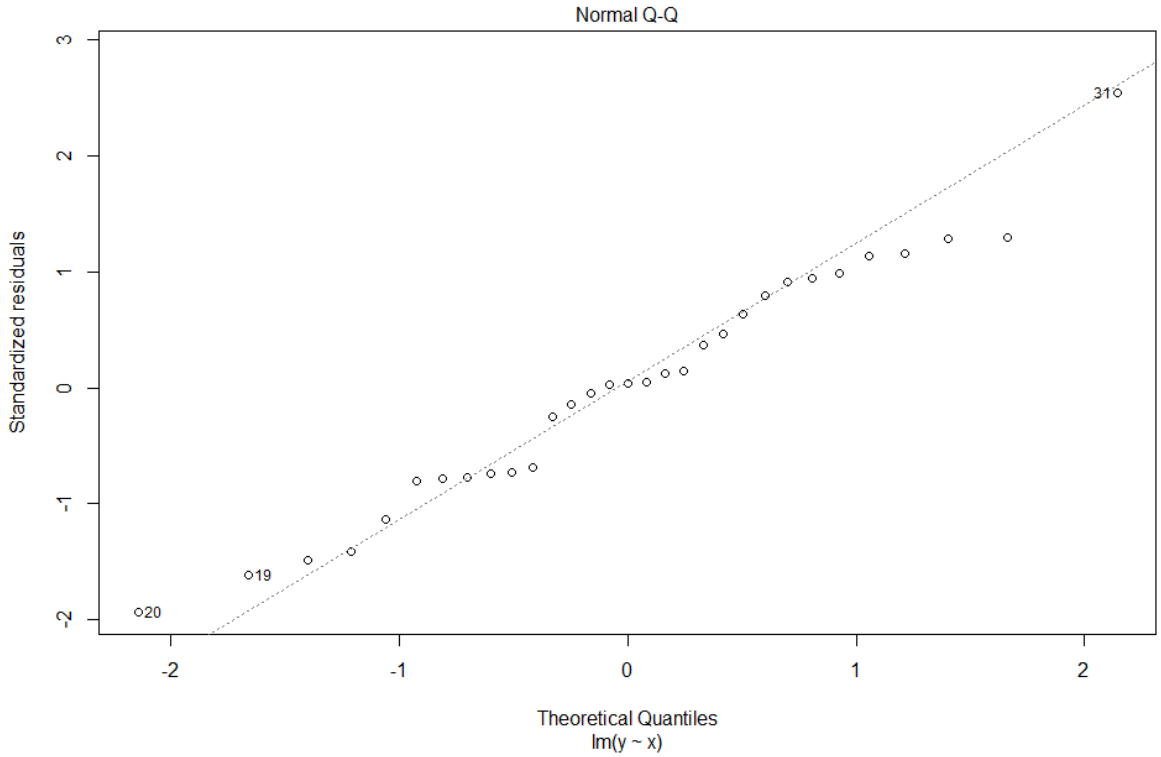

Figure 3, QQ-plot of residuals

From the qq-plot it can be concluded that there are some deviations from normality. But still in general these deviations are not on unacceptable level. Hence we can say that the data is more less normally distributed.

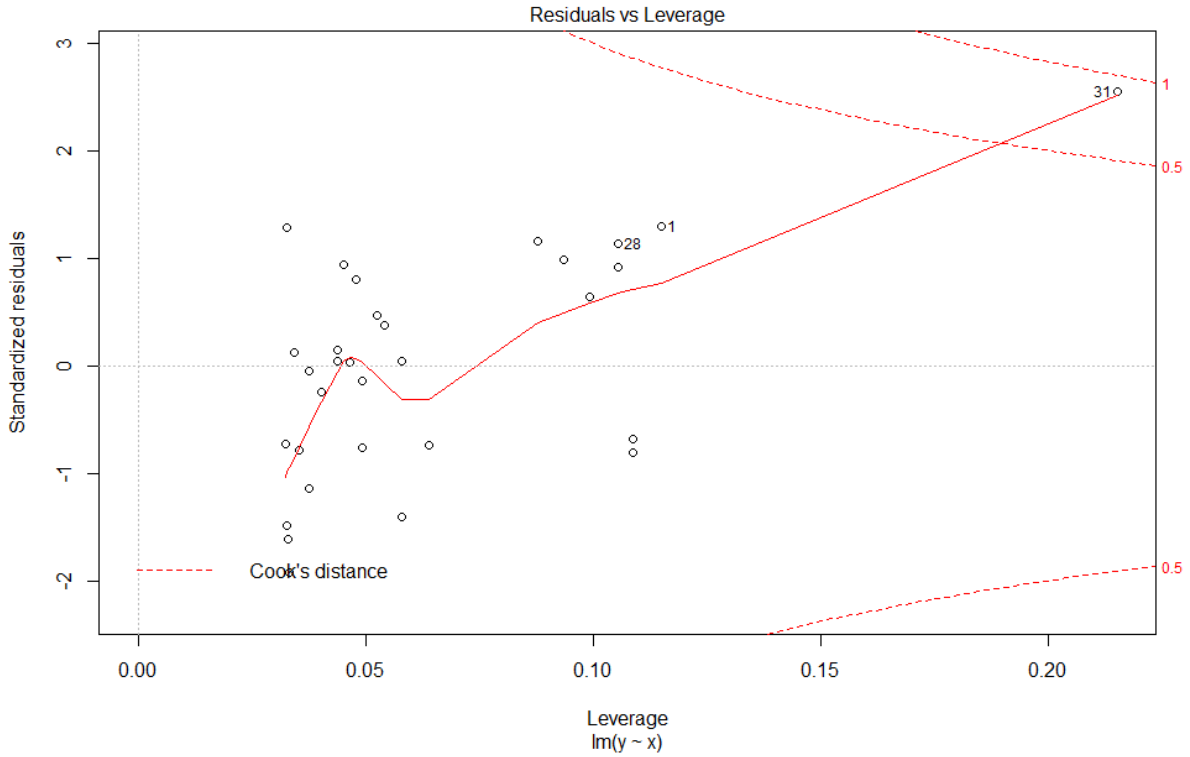

Figure 4, Cook's distance

???

d) Make predictions for several new values of the explanatory (independent) variable. For each predicted value, compute and plot the confidence intervals for the mean and single value.

The following predictions were made for the independent variable x (seq(10, 30, 1)):

Value	10	11	12	13	14	15	16	17	18	19	20
Prediction	13.72	18.78	23.84	28.91	33.97	39.04	44.11	49.17	54.24	59.30	64.37
Value	21	22	23	24	25	26	27	28	29	30	
Prediction	69.43	74.50	79.57	84.63	89.70	94.76	99.83	104.90	109.96	115.03	

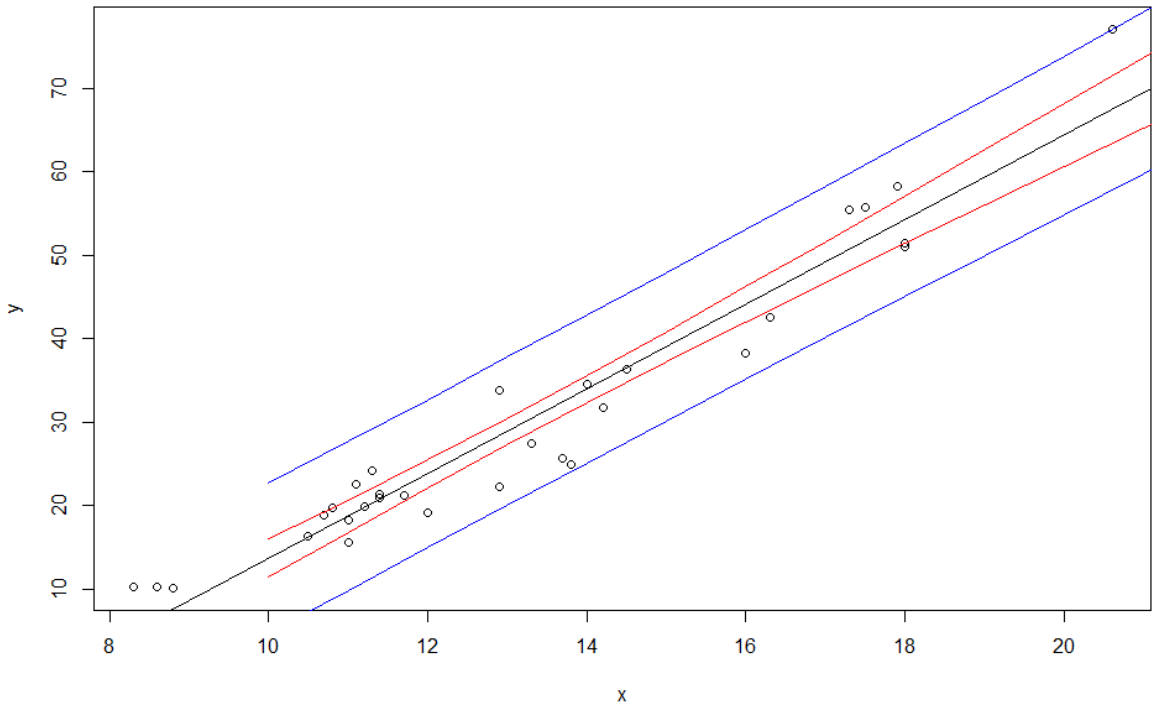

Figure 5, 95% confidence intervals for observations and regression lines

According to the graph above the area between two blue lines shows with 95% confidence possible observations for the values between 10 and 30. Additionally, the area between two red lines indicate 95% confidence interval for the mean of possible values (potential regression lines).

Task 2: Multivariate regression. Get a multivariate dataset (at least 3 variables) from 2.

a) Choose the response and explanatory variables.

For this task "BeefDemand.txt" file was used to apply the multivariate regression. Three variables including "Consumer Price Index" (x1), "RealDPI" (x2), and "RealBeefPrice" (x3) were used to derive multivariate regression model for predicting "BeefConsumption" level.

b) Build a multivariate linear model (command lm). Provide the estimates of the model's parameters.

ML=lm(y~x1+x2+x3) command was used to derive the multivariate linear model.

c) Analyze the summary statistics (command summary()) with the emphasis on:

i. t-test for slopes. Explain.

ii. Overall F-test. Explain.

iii. R2 and adjusted R2 coefficients. Explain.

The summary of the derived model indicated very low p-values for all intercept and the three variables (slopes) that indicate that none of the parameters are equal to 0 (H_0 hypothesis rejected in favor to H_1).

F-test's p-value is also 2.973e-11. Therefore we can reject H_0 (x1=x2=x3=0) in favor to H_1 (at least one of x1, x2, x3 are not = 0)

Both Multiple R-squared: 0.7989 and Adjusted R-squared: 0.7801 are close to 1 that indicate "goodness" of the model.

d) Plot the residuals against fitted values and comment on the model's adequacy.

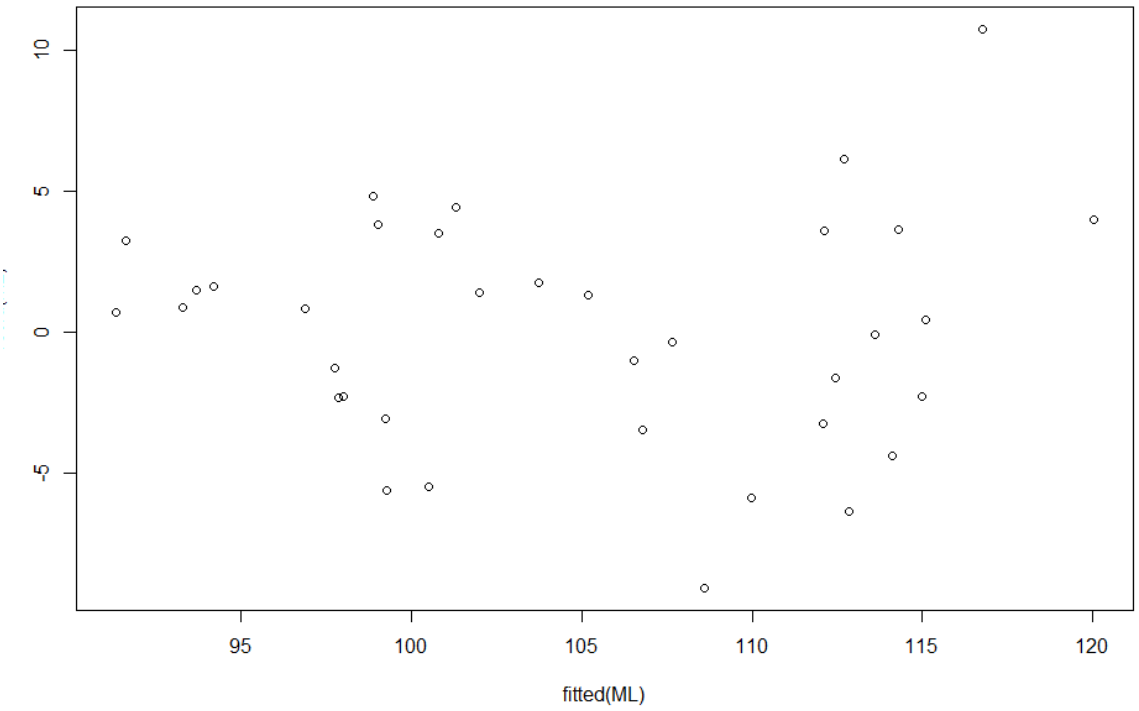

Figure 6, residuals against fitted values

Apart from a few outliers the residual values lie between -5 and 5 that are on an acceptable level since for example for the fitted value of 100 residual of -5 would make 5% difference which is tolerable.

e) Play with your model by adding or removing the explanatory variables. Alternatively, add a non-linear term(s) to your model:

i. Choose the best one by the partial F-test criterion (command anova)

ii. Choose the best one by the AIC criterion (command stepAIC) (deals with the trade-off between the goodness of fit of the model and the complexity of the model. Given a set of candidate models for the data, the preferred model is the one with the minimum AIC value. AIC rewards goodness of fit (as assessed by the likelihood function), but it also includes a penalty that is an increasing function of the number of estimated parameters. The penalty

discourages overfitting, because increasing the number of parameters in the model almost always improves the goodness of the fit.)

iii. For each model, watch the value of the adjusted R2.

Model	Pr(>F) compared to ML	AIC	adjusted R2
ML(y~x1+x2+x3)		108.08	0.7801
ML1 (y~x2+x3)	3.227e-08	140.96	0.4379
ML2(y~x1+x3)	2.628e-05	126.26	0.6264
ML3 (y~x1+x2)	5.111e-05	124.82	0.6411

In conclusion, we can state first that none of the independent variables are likely to be equal to 0 due to low Pr(>F) values. Secondly, according to AIC value (lowest) the best model is the initial one with three explanatory variables. Finally, adjusted R^2 is the biggest for the first model, hence the best one according to this criteria.

Task 3 Logistic regression. Get a binary response regression dataset from 1 or 2. Briefly describe the data.

For this task "apple_juice.dat" dataset was used to classify "Growth" column. Modeling the Growth Limit of Alicyclobacillus Acidoterrestris CRA7152 in Apple Juice: Effect of pH, Brix, Temperature, and Nisin Concentration.

a) Build a logistic regression model (command glm). Comment on the significance of the coefficients.

After building a logistic regression model with four independent variables (ph, Nisin, Temperature, and Brix) in order to classify Growth column, we get from the summary of the model some suspicious medium p-values: Intercept=>2.4%, x3=>2.1%, x4=>3%

It is difficult to judge whether we should exclude the above coefficients from the model solely from the p-values. Hence we should refer to the setpAIC function.

b) Use stepAIC command to select the best model.

Start: AIC=62.33

y ~ x + x2 + x3 + x4

	none	x4	x3	x	x2
AIC	62.331	66.153	67.219	78.148	81.637

According to the Akaike information criterion we should not eliminate any of the coefficients and hence the best model is "y ~ x + x2 + x3 + x4".

c) Make a prediction based on the entire dataset. State the threshold of acceptance. Compare the forecast with the actual observations. Comment on the results.

Threshold of acceptance will be 0,5 and above.

After making the forecast on the entire dataset and comparing with the actual values 62 out of 74 correct predictions were made. Overall accuracy of the model is (62/74)*100 = 84% which is on an acceptable level.

d) Divide the entire set into training and test subsets. Rebuild the model using only the training subset. Make predictions for the test subset. Comment.

The training subset is chosen to be 80% of total dataset: 59 observations and test subset: 15 observations.

	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
Pred	0.93	0.01	0.01	0.15	0.02	0.30	0.02	0.04	0.60	0.73	0.45	0.68	0.97	0.13	0.21
Actl	1	0	0	0	0	1	0	0	1	1	0	1	1	0	0

After rebuilding the model on the training subset and making forecasts on the test subset, only one false prediction was made. Accuracy rate on the test set is (14/15)*100=93% that indicates "goodness" of the model.

Task 4: Discriminant analysis. Use the same dataset as for the logistic regression.

a) Conduct the linear discriminant analysis (command lda, package MASS) using training and test subsets. Compare the forecast with the actual observations. Comment on the results.

	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
Pred	1	0	0	0	0	0	0	0	1	1	0	1	1	0	0
Actl	1	0	0	0	0	1	0	0	1	1	0	1	1	0	0

The linear discriminant regression analysis gave the same result as the previous logistic regression model with exactly the same row being misclassified.

b) Conduct the quadratic discriminant analysis (command qda). Comment.

	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
Pred	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0
Actl	1	0	0	0	0	1	0	0	1	1	0	1	1	0	0

The quadratic discriminant regression model showed slightly worse result by misclassifying two observations: 65 and 71. Accuracy = 87%

Task 5: The KNN classifier. Use the same dataset as for the logistic regression and discriminant analysis.

a) Conduct the KNN classification (command knn(), package class) using training and test subsets. Compare the forecast with the actual observations. Comment on the results.

k=5

	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
Pred	1	0	0	0	0	0	0	0	1	1	1	1	1	0	0
Actl	1	0	0	0	0	1	0	0	1	1	0	1	1	0	0

The KNN model gave a result of accuracy = (13/15)*100 = 87%

b) Play with a number of nearest neighbors K.

The best result is with the number of neighbours considered: 5

For other k values the number of falsely classified observations are from three and more.

Task 6: Compare the quality of classification obtained by algorithms 3-5 for the test subset.

Since the observation 65 was misclassified by all of the algorithms, we can conclude that this example is an outlier.

	Logistic	Linear discriminant	Quadratic discriminant	KNN
Accuracy %	93	93	87	87

Judging solely from the accuracy level, we could conclude that the logistic and linear discriminant regression models are the best ones. However, since the data availability was limited (only 59 observations for training subset) it is hard to conclude whether either of those models would behave with similar accuracy on actual data. Therefore, it would make sense to train the models and compare them based on larger datasets.

Appendix_R_Codes

```
library(MASS)
data()
x=trees[,c("Girth")]
y=trees[,c("Volume")]
```

#Task1

#a)

```
SL=lm(y~x)
#plot scatter diagram
plot(x,y)
# add regression line to the diagram
abline(SL)
```

#b)

```
summary(SL)
```

#c)

```
plot(x)
plot(fitted(SL), resid(SL))
plot(SL)
plot(x, cooks.distance(SL))
```

#d)

```
values = seq(10, 30, 1)
predict(SL, new=data.frame(x = values))
meanConfInterval = predict(SL, new=data.frame(x = values), int="conf")
meanConfInterval
observConfInterval = predict(SL, new=data.frame(x = values), int="predict")
observConfInterval
plot(x,y)
abline(SL)
# conf intervals for possible observations except some outliers
lines(values, observConfInterval[,2], col='blue')
lines(values, observConfInterval[,3], col='blue')
```

```
# conf interval for possible regression lines

lines(values, meanConfInterval[,2], col = 'red')

lines(values, meanConfInterval[,3], col = 'red')
```

#Task 2

```
#loading data

beef <- read.table('C:/FAU/Semester_3_HSE/Data Analysis/Homework3/BeefDemand.txt', header = TRUE)
```

#a)

```
y=beef[,c("BeefConsump")]

x1=beef[,c("CPI")]

x2=beef[,c("RealDPI")]

x3=beef[,c("RealBeefPrice")]
```

#b)

```
ML=lm(y~x1+x2+x3)
```

#c)

```
summary(ML)

stepAIC(ML)
```

#d)

```
plot(fitted(ML), resid(ML))
```

#e)

```
stepAIC(ML)

ML1=lm(y~x2+x3)

anova(ML, ML1)

stepAIC(ML1)

ML2=lm(y~x1+x3)

anova(ML, ML2)

stepAIC(ML2)

summary(ML2)

ML3=lm(y~x1+x2)

anova(ML, ML3)

stepAIC(ML3)
```

```
summary(ML3)
```

Task 3 Logistic regression

#a)

```
apple <- read.table("C:/FAU/Semester_3_HSE/Data Analysis/Homework3/apple_juice.dat", header = FALSE)

colnames(apple) <- c("pH", "Nisin", "Temp", "Brix", "Growth")

y <- apple[,c("Growth")]

x <- apple[,c("pH")]

x2 <- apple[,c("Nisin")]

x3 <- apple[,c("Temp")]

x4 <- apple[,c("Brix")]

#glm generalized linear model from MASS package

logR = glm(y~x+x2+x3+x4, family = binomial)

summary(logR)
```

#b)

```
stepAIC(logR)
```

#c)

```
f = predict(logR, type="response")

dif = abs(y - f)

length(dif[dif>0.5])
```

#d)

```
LogR_Training = glm(Growth~pH+Nisin+Temp+Brix, data=apple[1:59,], family = binomial)

new=apple[60:74,]

predict(LogR_Training, new, type="response")

tail(apple$Growth, 15)
```

#Task 4 Discriminant analysis

#a)

```
LDA1 = lda(Growth~pH+Nisin+Temp+Brix, data=apple[1:59,])

predict(LDA1, new=apple[60:74,])$class

tail(apple$Growth, 15)
```

#b)

```
QDA1 = qda(Growth~pH+Nisin+Temp+Brix, data=apple[1:59,])
```

```
predict(QDA1, new=apple[60:74,])
```

#Task 5 KNN

#a)

```
library(class)

train=apple[1:59,c("pH", "Nisin", "Temp", "Brix")]

test=apple[60:74, c("pH", "Nisin", "Temp", "Brix")]

res=apple[1:59, c("Growth")]

knn(train, test, res, 10)

tail(apple$Growth, 15)
```