Chuvilina Anna, group MBD171

Assignment on Regression and Classification

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

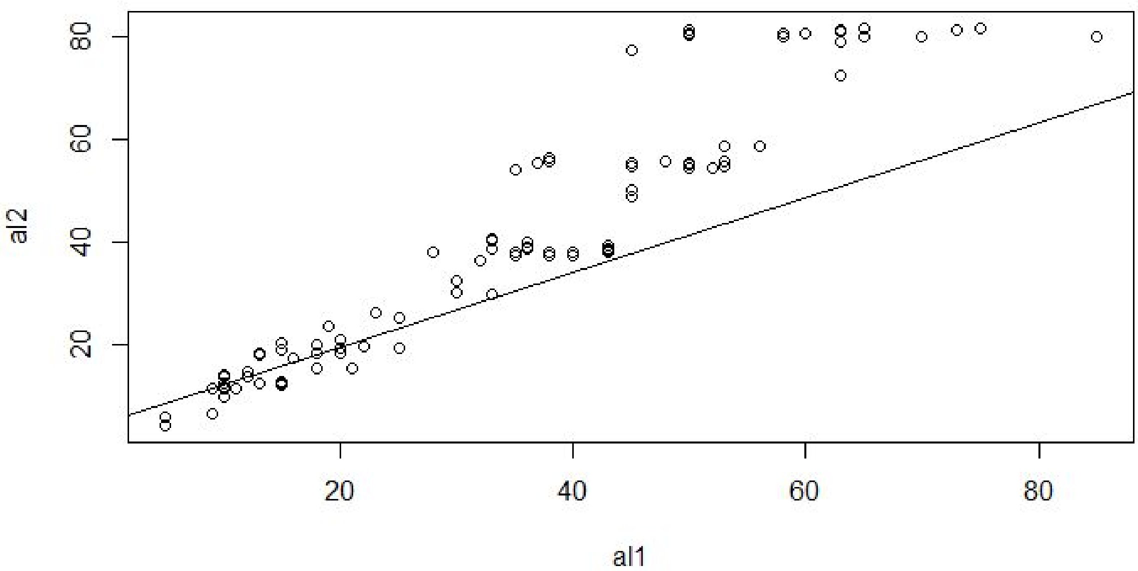
I have chosen data frame "alaska.pipeline" from UsingR. This data frame contains the following columns: field.defect (depth of defect as measured in field), lab.defect (depth of defect as measured in lab), batch (one of 6 batches). Length of data frame is 107.

```
al1 = alaska.pipeline$field.defect
al2 = alaska.pipeline$lab.defect
al3 = alaska.pipeline$batch
length(al1)
[1] 107
length(al2)
[1] 107
length(al3)
[1] 107
```

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

I use data about depth of defect as measured in field and about depth of defect as measured in lab for to build a simple regression model.

```
al = lm(al1~al2)
plot(al1, al2)
abline(al)
```



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

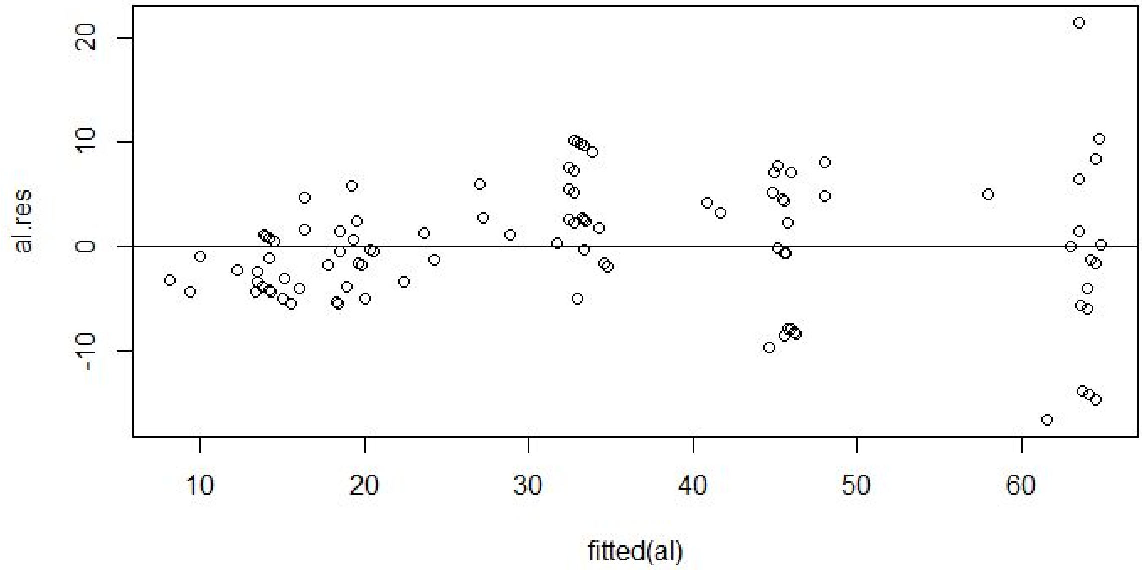
i. The t-test for the slope. Explain.
ii. The F-test. Explain.
iii. R2 coefficient. Explain.

```
> summary(al)

Call:
lm(formula = al1 ~ al2)

Residuals:
    Min       1Q   Median       3Q      Max
-16.5817  -3.8259   0.1283   3.7432  21.5174

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.99368    1.12566   4.436 2.26e-05 ***
al2          0.73111    0.02455  29.778  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.081 on 105 degrees of freedom
Multiple R-squared:  0.8941,  Adjusted R-squared:  0.8931
F-statistic: 886.7 on 1 and 105 DF,  p-value: < 2.2e-16
```

As we can see t-test value is 29.778 and p-value is < 2e-16, it means that al2 isn't significant R^2 is 0.8931, It means that we have a good model with correlated values. Estimate show us positive correlation.

(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.

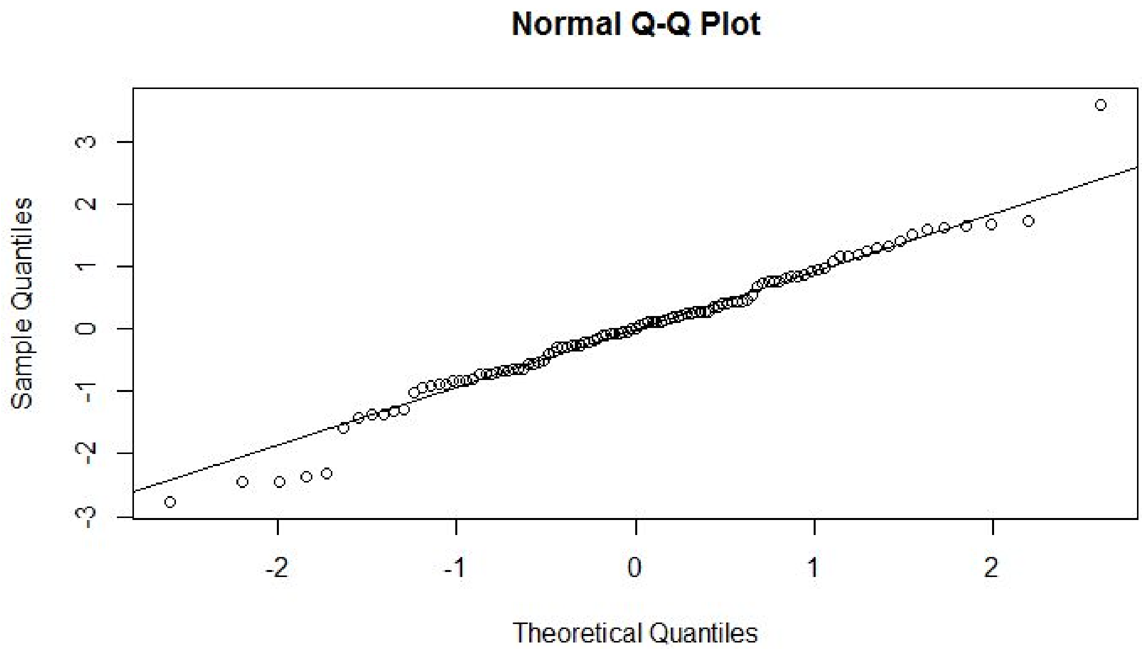
```
al.res = resid(al)
plot(fitted(al), al.res)
abline(0,0)
```



The plot shows the mean residual doesn't change with the fitted values, but the spread of the residuals is increasing as the fitted values changes. That is, the spread is not constant. Heteroskedasticity.

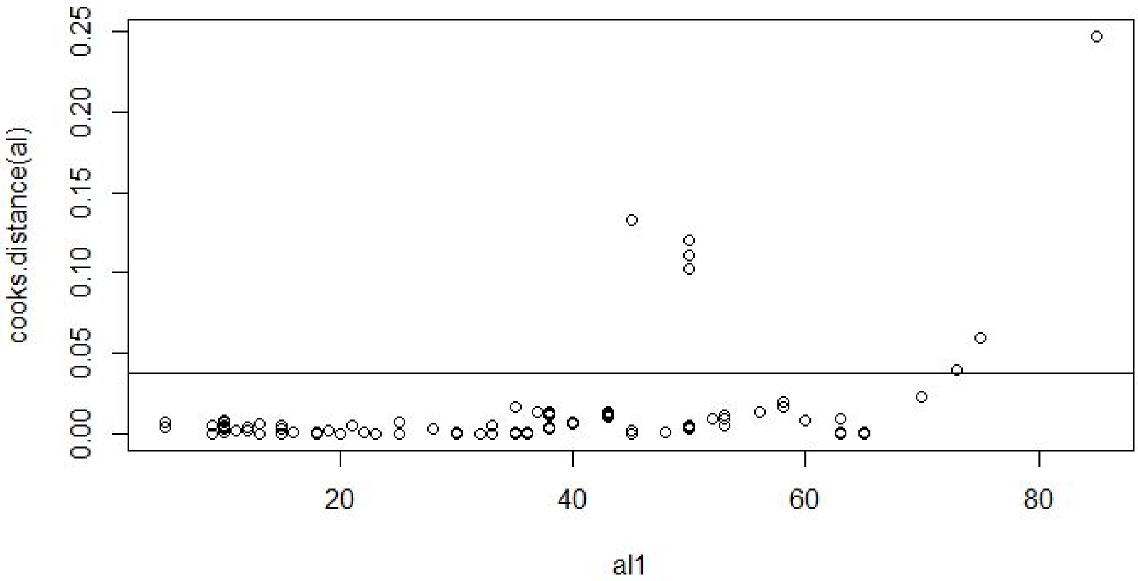
```
al.stdres = rstandard(al)
qqnorm(al.stdres)
qqline(al.stdres)
```

We can observe heavy-tailed distribution on qqplot.



```
plot(al1, cooks.distance(al))
n = 4/107
abline(n,0)
```

Цу

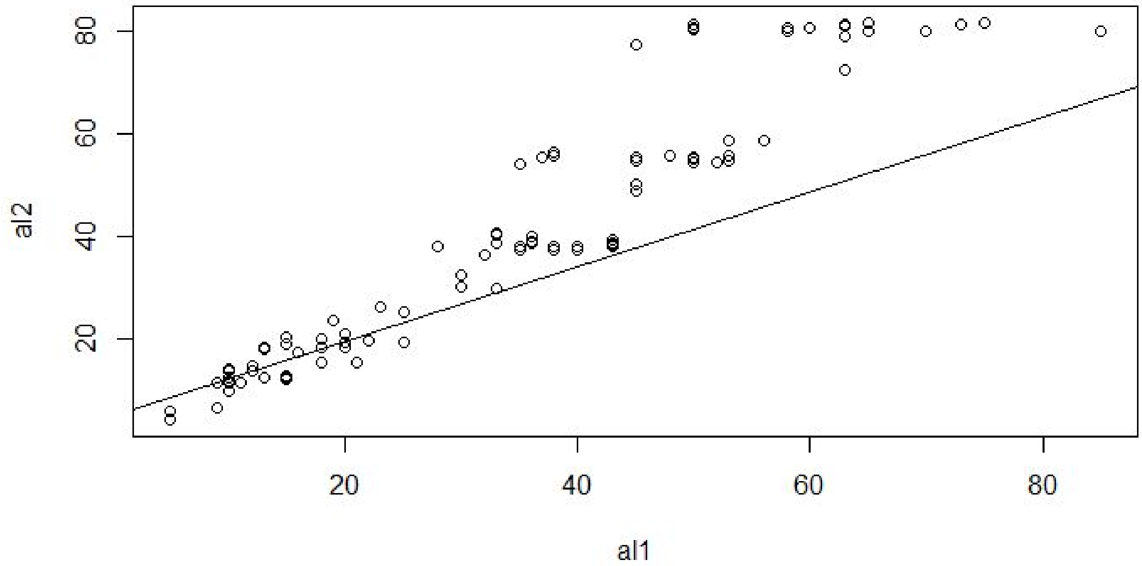


The cut off is 0.037. We can see that model has 7 outliers which can negatively affect regression model.

(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.

```
> al1P= seq(0, 53, 0.5)
> predict(al, new = data.frame(al1 = al1P))
        1         2         3         4         5         6         7         8         9
19.762123 45.935900 14.132568 20.493234 16.325901 33.507011 20.347012 32.922123
45.643455
       10        11        12        13        14        15        16        17        18
64.871676 33.872567 46.228344 34.603678 15.448568 64.579232 15.009902 64.579232
19.981457
       19        20        21        22        23        24        25        26        27
45.935900 63.994343 19.615901 46.301455 13.840124 19.323457 16.325901 33.360789
19.250346
       28        29        30        31        32        33        34        35        36
```

```
32.775900 45.204788 63.482565 33.141456 45.789677 33.360789 14.132568 63.775009
14.278790
       37        38        39        40        41        42        43        44        45
64.140565 19.981457 45.204788 18.884790 45.570344 13.986346 18.446124 13.401457
32.775900
       46        47        48        49        50        51        52        53        54
18.519235 32.775900 45.424122 33.287678 44.839233 32.775900 13.767013 64.725454
13.401457
       55        56        57        58        59        60        61        62        63
63.482565 18.373012 45.424122 63.628787 63.994343 45.789677 15.960346 64.213676
13.767013
       64        65        66        67        68        69        70        71        72
64.506121 14.132568 32.922123 44.619900 62.970787 18.299901 45.570344 13.328346
19.250346
       73        74        75        76        77        78        79        80        81
16.325901 32.410345 19.250346 32.410345 45.570344 63.482565 32.410345 16.325901
22.321012
       82        83        84        85        86        87        88        89        90
12.158568 34.823011 17.788124  8.137458 31.679234 24.221901 27.219456 41.695455
27.000123
       91        92        93        94        95        96        97        98        99
23.637012 15.083013 48.056122 34.238122  9.380346 57.999232 33.360789 19.177235
64.579232
      100       101       102       103       104       105       106       107
61.581676 44.912344  9.965235 28.827901 19.469679 47.983010 14.425013 40.818122

> conf = predict(al, new = data.frame(al1 = al1P), int = "conf")
> prdct = predict(al, new = data.frame(al1 = al1P), int = "predict")
> plot(al1, al2)
> abline(al)
```



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

I have chosen data frame Efficiency of Muscle Work - Case 2: Algerian Subjects from http://www.stat.ufl.edu. Measurements of Heat Production (calories) at various Body Masses (kgs) and Work levels (Calories/hour) on a stationary bike. This data frame contains the following columns: Body Mass(V1), Work Level(V2), Heat Output(V3). Length of data frame is 37.

(a) Choose the response and explanatory variables.

Response variable is Heat Production. Explanatory variables are Body Masses and Work levels.

```
v1 = musc.data$V1
v2 = musc.data$V2
v3 = musc.data$V3
```

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

```
musc.lm = lm(v3~v1+v2)
```

(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.

```
> summary(musc.lm)
```

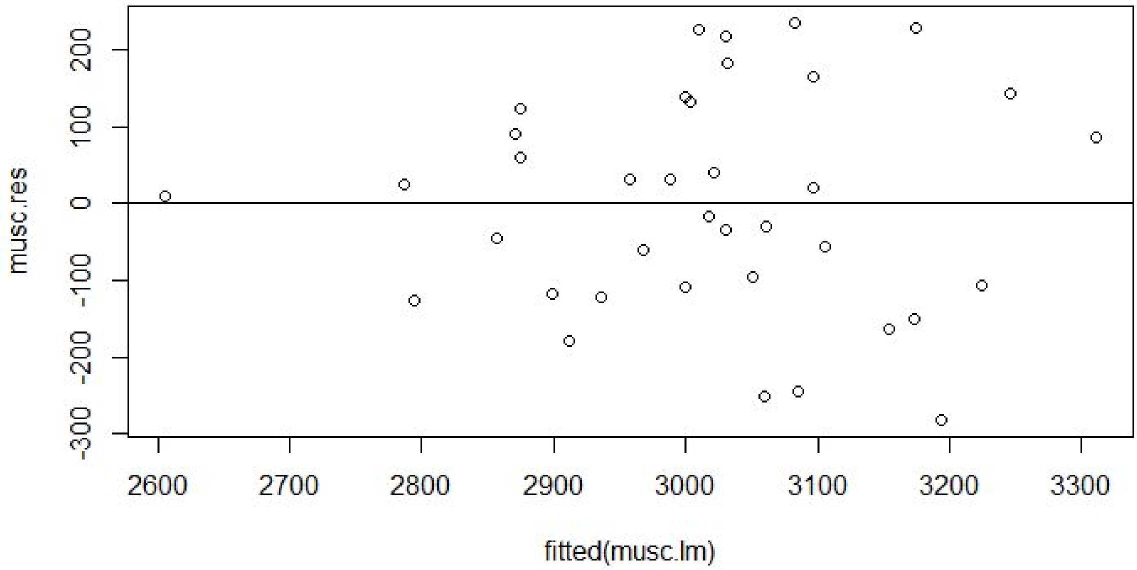
```
Call:
lm(formula = v3 ~ v1 + v2)

Residuals:
   Min     1Q Median     3Q    Max
-282.0 -109.2    9.1  123.9  235.9

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  977.425    376.053   2.599 0.013723 *
v1            17.778      4.943   3.597 0.001011 **
v2             6.244      1.522   4.102 0.000242 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 147.1 on 34 degrees of freedom
Multiple R-squared:  0.4922,  Adjusted R-squared:  0.4624
F-statistic: 16.48 on 2 and 34 DF,  p-value: 9.914e-06
```

---

v2 isn't significant, v1 has not good value of significant
R^2 is 0.4624, It means that we have not good model with low correlated values. Estimate show us positive correlation.

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

```
musc.res = resid(musc.lm)
plot(fitted(musc.lm),musc.res)
 abline(0,0)
```



The plot shows the mean residual doesn't change with the fitted values, but the spread of the residuals is increasing as the fitted values changes. That is, the spread is not constant. Heteroskedasticity.

(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), see p. 294 of [1].

```
> musc.lm1 =lm (v3~v1)
> musc.lm2 = lm(v3~v2)
> anova(musc.lm1)
Analysis of Variance Table

Response: v3
          Df  Sum Sq Mean Sq F value   Pr(>F)
v1         1  349284  349284  11.111 0.002037 **
Residuals 35 1100259   31436
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> anova(musc.lm2)
Analysis of Variance Table

Response: v3
          Df  Sum Sq Mean Sq F value    Pr(>F)
v2         1  433482  433482  14.932 0.0004616 ***
Residuals 35 1016061   29030
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
 anova(musc.lm1, musc.lm2)
Analysis of Variance Table

Model 1: v3 ~ v1
Model 2: v3 ~ v2
  Res.Df     RSS Df Sum of Sq F Pr(>F)
1     35 1100259
2     35 1016061  0     84198
```

---

ii. Choose the best one by the AIC criterion (command stepAIC), see p. 295 of [1].

```
> stepAIC(musc.lm1)
Start:  AIC=385.11
v3 ~ v1

       Df Sum of Sq     RSS    AIC
<none>              1100259 385.11
- v1    1    349284 1449544 393.31

Call:
lm(formula = v3 ~ v1)

Coefficients:
(Intercept)           v1
    1668.97        19.76

> stepAIC(musc.lm2)
Start:  AIC=382.16
v3 ~ v2

       Df Sum of Sq     RSS    AIC
<none>              1016061 382.16
- v2    1    433482 1449544 393.31

Call:
lm(formula = v3 ~ v2)

Coefficients:
(Intercept)           v2
   2118.255        6.779
```

---

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

```
> summary(musc.lm1)
```

```
Call:
lm(formula = v3 ~ v1)

Residuals:
   Min      1Q  Median      3Q     Max 
-333.04 -137.55    3.78  118.78  321.07 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1668.970    405.072   4.120  0.00022 ***
v1            19.759      5.928   3.333  0.00204 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 177.3 on 35 degrees of freedom
Multiple R-squared:  0.241,	Adjusted R-squared:  0.2193 
F-statistic: 11.11 on 1 and 35 DF,  p-value: 0.002037

> summary(musc.lm2)

Call:
lm(formula = v3 ~ v2)

Residuals:
   Min      1Q  Median      3Q     Max 
-303.90 -132.51  -30.33  151.04  318.10 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2118.255    233.930   9.055 1.07e-10 ***
v2             6.779      1.754   3.864 0.000462 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 170.4 on 35 degrees of freedom
Multiple R-squared:  0.299,	Adjusted R-squared:  0.279 
F-statistic: 14.93 on 1 and 35 DF,  p-value: 0.0004616
```

We can see that model with different variables have close values of R^2: 0.2193 and 0.279. It means that choice of explanatory variable has not affect the quality of the model.

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

I have chosen data frame Presence of Growth of CRA7152 in Apple Juice from http://www.stat.ufl.edu. Absence of growth of CRA7152 in apple juice
as a function of pH (3.5-5.5), Brix (11-19), temperature (25-50C), and Nisin concentration

(0-70).

```
 apple.ph = apple.data$V1
apple.nisin = apple.data$V2
apple.temp = apple.data$V3
apple.brix = apple.data$V4
apple.growth = apple.data$V5
```

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

```
apple.log = glm(apple.growth~apple.ph+apple.nisin+apple.temp+apple.brix, family =
binomial)
> summary(apple.log)

Call:
glm(formula = apple.growth ~ apple.ph + apple.nisin + apple.temp +
    apple.brix, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3614  -0.3990  -0.1585   0.6306   1.6200

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.24633    3.21864  -2.251 0.024362 *
apple.ph     1.88595    0.54123   3.485 0.000493 ***
apple.nisin -0.06628    0.01905  -3.479 0.000503 ***
apple.temp   0.11042    0.04769   2.316 0.020585 *
apple.brix  -0.31173    0.14317  -2.177 0.029458 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 95.945  on 73  degrees of freedom
Residual deviance: 52.331  on 69  degrees of freedom
AIC: 62.331

Number of Fisher Scoring iterations: 6
```

We can see that apple.ph, apple.temp, apple.brix correlated, but without good value.
Apple.nisin and apple.ph don't correlated.
(b) Use stepAIC command to select the best model.

```
> stepAIC(apple.log)
Start:  AIC=62.33
apple.growth ~ apple.ph + apple.nisin + apple.temp + apple.brix

              Df Deviance    AIC
<none>            52.331 62.331
- apple.brix   1   58.153 66.153
- apple.temp   1   59.219 67.219
- apple.ph     1   70.148 78.148
- apple.nisin  1   73.637 81.637

Call:  glm(formula = apple.growth ~ apple.ph + apple.nisin + apple.temp +
    apple.brix, family = binomial)

Coefficients:
(Intercept)     apple.ph  apple.nisin   apple.temp   apple.brix
   -7.24633      1.88595     -0.06628      0.11042     -0.31173

Degrees of Freedom: 73 Total (i.e. Null);  69 Residual
Null Deviance:     95.95
Residual Deviance: 52.33      AIC: 62.33
```

We observe model with lowest AIC 62.33 and optimal model with considering AIC will be formula = apple.growth ~ apple.ph + apple.temp

(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 is 0.5

```
> predict(apple.log, type = "response")
          1           2           3           4           5           6           7           8
0.640865388 0.063703458 0.624390499 0.269213240 0.721189987 0.615292098
0.938462821 0.770470752
          9          10          11          12          13          14          15          16
0.005417622 0.007774388 0.076511360 0.002429956 0.367785386 0.071650011
0.021713675 0.012476633
         17          18          19          20          21          22          23          24
0.023025058 0.429165305 0.013483055 0.272967506 0.819688780 0.974717132
0.932083860 0.008631542
         25          26          27          28          29          30          31          32
0.010669862 0.154826624 0.024486574 0.300339180 0.023000986 0.048209979
0.591373886 0.729702272
         33          34          35          36          37          38          39          40
0.447747132 0.675506680 0.970759620 0.128449650 0.211818044 0.640865388
0.063703458 0.624390499
```

```
        41          42          43          44          45          46          47          48
0.269213240 0.721189987 0.615292098 0.938462821 0.770470752 0.005417622
0.007774388 0.076511360
        49          50          51          52          53          54          55          56
0.002429956 0.367785386 0.071650011 0.021713675 0.012476633 0.023025058
0.429165305 0.013483055
        57          58          59          60          61          62          63          64
0.272967506 0.819688780 0.974717132 0.932083860 0.008631542 0.010669862
0.154826624 0.024486574
        65          66          67          68          69          70          71          72
0.300339180 0.023000986 0.048209979 0.591373886 0.729702272 0.447747132
0.675506680 0.970759620
        73          74
0.128449650 0.211818044

> apple.growth
 [1] 0 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 1 1 1 0 0 1 0
[47] 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 0 1 0 0 1 1 0 1 1 0 0
```

More than 80% of predicted values correspond to actual observations.

(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.

```
apple.data = read.table("apple_juice_dat.txt")
names(apple.data) = c('ph','nisin', 'temp', 'brix', 'growth')
apple.log = glm(growth ~ ph + nisin + temp + brix, data = apple.data, family =
binomial)
set.seed(101)
tr.index = sample(1:nrow(apple.data), nrow(apple.data)*0.8)
trSet = apple.data[tr.index, ]
testSet = apple.data[-tr.index, ]
apple.log1 = glm(growth ~ ph+ nisin + temp + brix, trSet, family = binomial)
fitted_results_test = predict(apple.log1, newdata= testSet, type = "response")
fitted_results_test = ifelse(fitted_results_test > 0.5,1,0)
fitted_results_test
5  6 17 20 26 27 29 37 43 44 49 54 56 58 70
 1  1  0  0  0  0  0  0  1  1  0  0  0  1  1
```

---

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.

```
> apple.log2 = lda(growth ~ ph+ nisin + temp + brix, trSet)
> apple.log2
Call:
lda(growth ~ ph + nisin + temp + brix, data = trSet)

Prior probabilities of groups:
        0         1
0.5932203 0.4067797

Group means:
      ph    nisin     temp  brix
0 4.2000 47.42857 38.97143 15.00
1 4.8125 17.50000 40.91667 13.25

Coefficients of linear discriminants:
              LD1
ph     1.14212031
nisin -0.03855885
temp   0.03817777
brix  -0.16672426

> apple.log2 = lda(growth ~ ph+ nisin + temp + brix, testSet)
> apple.log2p = predict(apple.log2, trSet)$class
> apple.log2p
 [1] 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0
[47] 1 0 0 0 0 0 1 0 0 1 0 0 0
Levels: 0 1
```

The LDA output is 0.593 and 0.406, it means that 59,3% of the training observations correspond presence of Growth of CRA7152.

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

```
> apple.qda = qda(growth ~ ph+ nisin + temp + brix, trSet)
> predict(apple.qda, testSet)
$class
 [1] 1 1 0 0 0 0 0 0 1 0 0 0 0 1 0
Levels: 0 1

$posterior
            0            1
5  0.02246616 9.775338e-01
6  0.10345915 8.965409e-01
17 0.99958814 4.118634e-04
20 0.97856839 2.143161e-02
```

```
26 0.99686579 3.134205e-03
27 0.99999911 8.854489e-07
29 0.99809355 1.906453e-03
37 0.99739444 2.605557e-03
43 0.10345915 8.965409e-01
44 0.99988304 1.169638e-04
49 0.99996654 3.345766e-05
54 0.99958814 4.118634e-04
56 0.99866290 1.337101e-03
58 0.07475746 9.252425e-01
70 0.83690817 1.630918e-01
```

---

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.
(b) Play with a number of nearest neighbors K.

```
> apple.label = apple.data[1:59, 1]
> knn(trSet, testSet, apple.label, 3)
 [1] 3.5 3.5 5.5 5   5   3.5 3.5 3.5 5   5   3.5 5.5 5   4   4
Levels: 3.5 4 5 5.5
> knn(trSet, testSet, apple.label, 5)
 [1] 5   5   5.5 5   5   3.5 5   3.5 5   5   5.5 5.5 3.5 5   5.5
Levels: 3.5 4 5 5.5
> knn(trSet, testSet, apple.label, 1)
 [1] 3.5 5   5   3.5 5   3.5 5   3.5 4   4   5   5   4   5.5 5.5
Levels: 3.5 4 5 5.5
```
