Introduction to Statistical Learning - FINAL PROJECT

CHARITA TUMMALA

ctpmx@umsystem.edu

16338814

1) Regression :

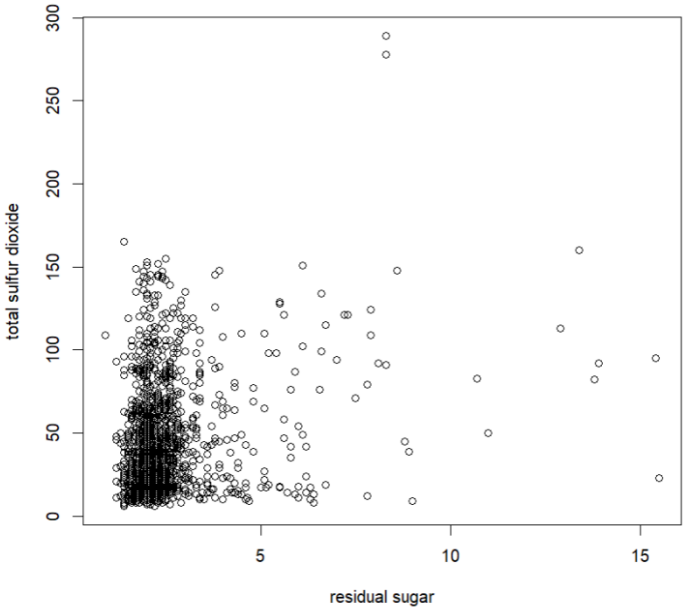
Final Project.R × winequality_red ×

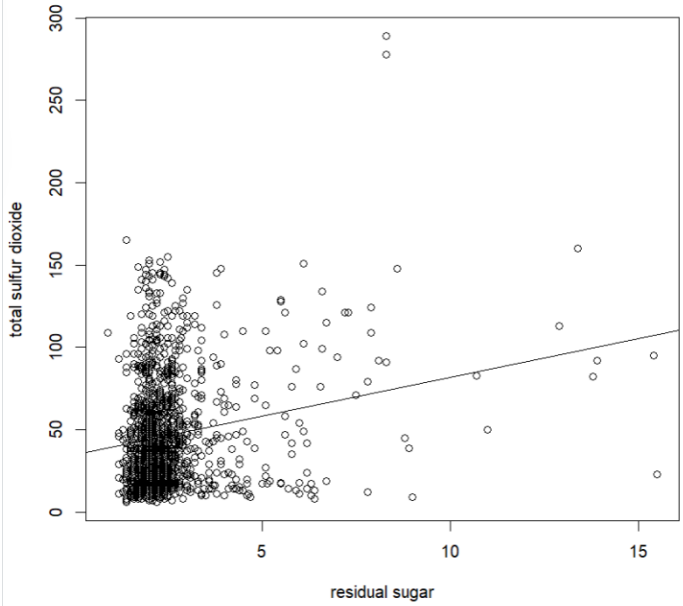
Filter

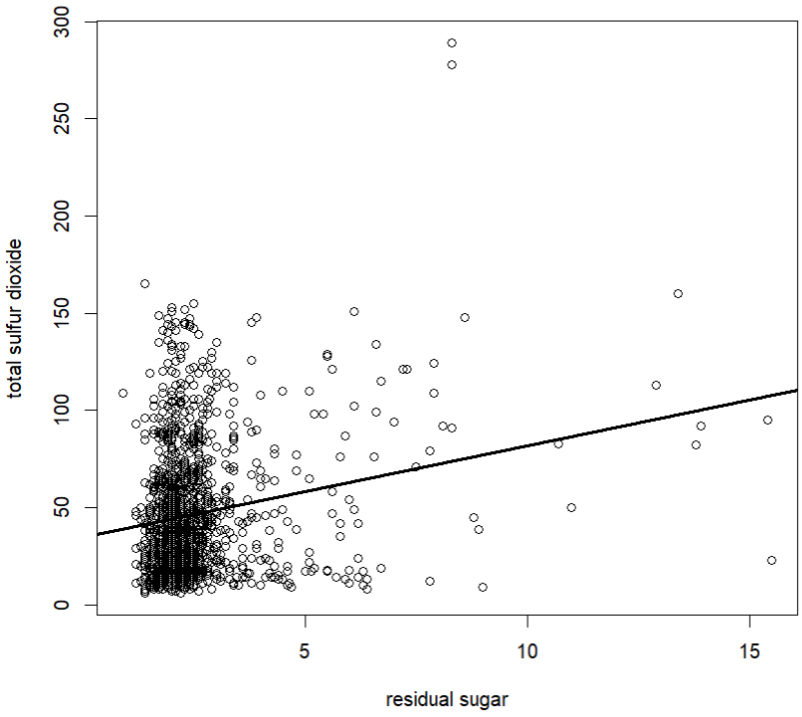
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
1	7.4	0.700	0.00	1.90	0.076	11	34	0.99780	3.51	0.56	9.4	
2	7.8	0.880	0.00	2.60	0.098	25	67	0.99680	3.20	0.68	9.8	
3	7.8	0.760	0.04	2.30	0.092	15	54	0.99700	3.26	0.65	9.8	
4	11.2	0.280	0.56	1.90	0.075	17	60	0.99800	3.16	0.58	9.8	
5	7.4	0.700	0.00	1.90	0.076	11	34	0.99780	3.51	0.56	9.4	
6	7.4	0.660	0.00	1.80	0.075	13	40	0.99780	3.51	0.56	9.4	
7	7.9	0.600	0.06	1.60	0.069	15	59	0.99640	3.30	0.46	9.4	
8	7.3	0.650	0.00	1.20	0.065	15	21	0.99460	3.39	0.47	10.0	
9	7.8	0.580	0.02	2.00	0.073	9	18	0.99680	3.36	0.57	9.5	
10	7.5	0.500	0.36	6.10	0.071	17	102	0.99780	3.35	0.80	10.5	
11	6.7	0.580	0.08	1.80	0.097	15	65	0.99590	3.28	0.54	9.2	
12	7.5	0.500	0.36	6.10	0.071	17	102	0.99780	3.35	0.80	10.5	
13	5.6	0.615	0.00	1.60	0.089	16	59	0.99430	3.58	0.52	9.9	
14	7.8	0.610	0.29	1.60	0.114	9	29	0.99740	3.26	1.56	9.1	
15	8.9	0.620	0.18	3.80	0.176	52	145	0.99860	3.16	0.88	9.2	
16	8.9	0.620	0.19	3.90	0.170	51	148	0.99860	3.17	0.93	9.2	
17	8.5	0.280	0.56	1.80	0.092	35	103	0.99690	3.30	0.75	10.5	
18	8.1	0.560	0.28	1.70	0.368	16	56	0.99680	3.11	1.28	9.3	
19	7.4	0.590	0.08	4.40	0.086	6	29	0.99740	3.38	0.50	9.0	
20	7.9	0.320	0.51	1.80	0.341	17	56	0.99690	3.04	1.08	9.2	
21	8.9	0.220	0.48	1.80	0.077	29	60	0.99680	3.39	0.53	9.4	
22	7.6	0.390	0.31	2.30	0.082	23	71	0.99820	3.52	0.65	9.7	
23	7.9	0.430	0.21	1.60	0.106	10	37	0.99660	3.17	0.91	9.5	
24	8.5	0.490	0.11	2.30	0.084	9	67	0.99680	3.17	0.53	9.4	

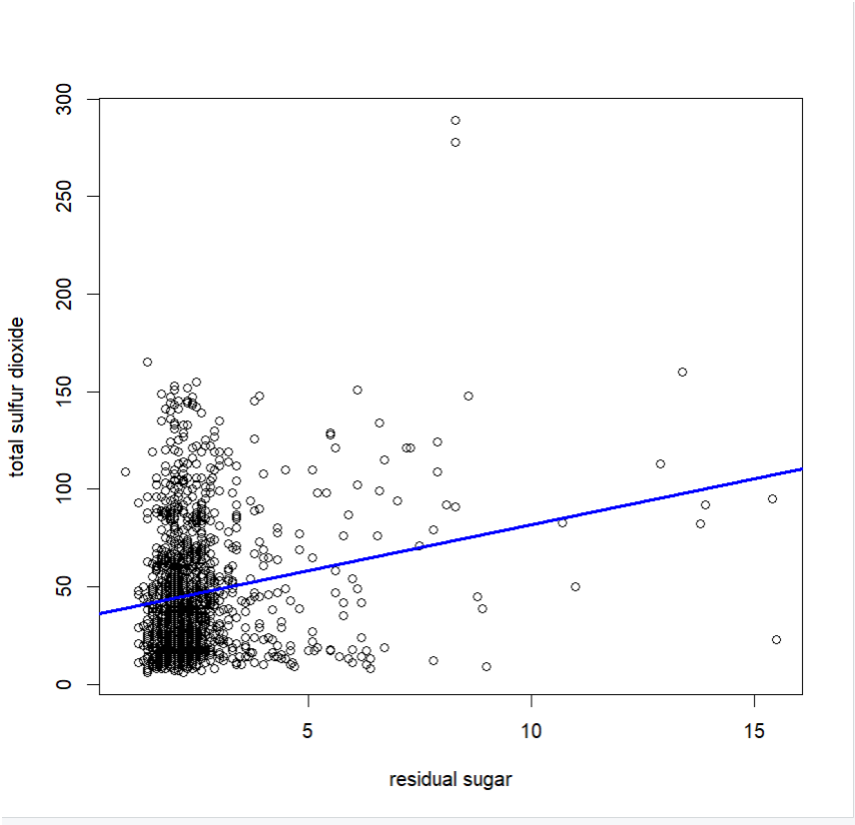
```
> library(readxl)
> winequality_red <- read_excel("C:/Users/tumma/Downloads/winequality_red.xlsx")
> View(winequality_red)
> attach(winequality_red)
> lm.fit <- lm(`total sulfur dioxide` ~ `residual sugar`, data = winequality_red)
> lm.fit <- lm(`total sulfur dioxide` ~ `residual sugar`)
> lm.fit

Call:
lm(formula = `total sulfur dioxide` ~ `residual sugar`)

Coefficients:
     (Intercept)  `residual sugar`
          34.442             4.737

>
> summary(lm.fit)

Call:
lm(formula = `total sulfur dioxide` ~ `residual sugar`)

Residuals:
    Min      1Q  Median      3Q     Max
-84.863 -23.021  -8.337  16.032 215.242

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       34.4418     1.6600  20.748  < 2e-16 ***
`residual sugar`   4.7369     0.5717   8.286 2.45e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 32.22 on 1597 degrees of freedom
Multiple R-squared:  0.04122,	Adjusted R-squared:  0.04062
F-statistic: 68.66 on 1 and 1597 DF,  p-value: 2.449e-16

> names(lm.fit)
 [1] "coefficients"  "residuals"     "effects"       "rank"          "fitted.values"
 [6] "assign"        "qr"            "df.residual"   "xlevels"       "call"
[11] "terms"         "model"
```

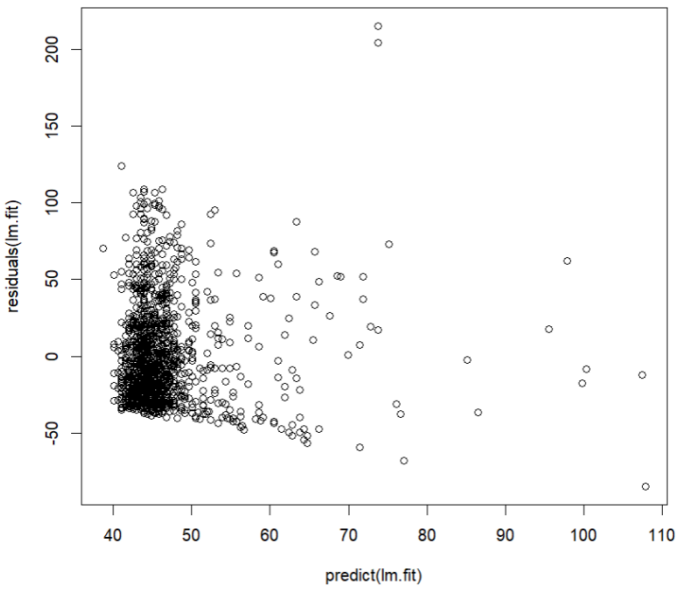
```
> coef(lm.fit)
     (Intercept) `residual sugar`
       34.441761         4.736886
> confint(lm.fit)
                     2.5 %    97.5 %
(Intercept)      31.185706 37.697815
`residual sugar`  3.615586  5.858185
> predict(lm.fit, data.frame(`residual sugar` = (c(5,10,15))),interval = "confidence")
           fit       lwr       upr
1     43.44184 41.70664  45.17704
2     46.75766 45.17572  48.33961
3     45.33660 43.73362  46.93958
4     43.44184 41.70664  45.17704
5     43.44184 41.70664  45.17704
6     42.96816 41.18374  44.75257
7     42.02078 40.12184  43.91972
8     40.12602 37.94624  42.30581
9     43.91553 42.22353  45.60753
10    63.33676 59.04221  67.63132
11    42.96816 41.18374  44.75257
12    63.33676 59.04221  67.63132
13    42.02078 40.12184  43.91972
14    42.02078 40.12184  43.91972
15    52.44193 50.32114  54.56271
16    52.91562 50.71847  55.11276
17    42.96816 41.18374  44.75257
18    42.49447 40.65531  44.33362
19    55.28406 52.66619  57.90193
20    42.96816 41.18374  44.75257
21    42.96816 41.18374  44.75257
22    45.33660 43.73362  46.93958
23    42.02078 40.12184  43.91972
24    45.33660 43.73362  46.93958
25    45.81029 44.22218  47.39839
> predict(lm.fit, data.frame(`residual sugar` = (c(5,10,15))),interval = "prediction")
           fit          lwr      upr
1     43.44184 -19.780488448 106.6642
2     46.75766 -16.460647875 109.9760
3     45.33660 -17.882243440 108.5554
4     43.44184 -19.780488448 106.6642
5     43.44184 -19.780488448 106.6642
6     42.96816 -20.255546839 106.1919
7     42.02078 -21.206260107 105.2478
8     40.12602 -23.110072098 103.3621
9     43.91553 -19.305628902 107.1367
10    63.33676  -0.007498758 126.6810
11    42.96816 -20.255546839 106.1919
12    63.33676  -0.007498758 126.6810
13    42.02078 -21.206260107 105.2478
14    42.02078 -21.206260107 105.2478
15    52.44193 -10.792163213 115.6760
16    52.91562 -10.321081698 116.1523
17    42.96816 -20.255546839 106.1919
18    42.49447 -20.730804064 105.7197
19    55.28406  -7.968654220 118.5368
20    42.96816 -20.255546839 106.1919
21    42.96816 -20.255546839 106.1919
22    45.33660 -17.882243440 108.5554
23    42.02078 -21.206260107 105.2478
24    45.33660 -17.882243440 108.5554
25    45.81029 -17.408179368 109.0288

> plot(`residual sugar`,`total sulfur dioxide`)
>
> abline(lm.fit)
>
> abline(lm.fit, lwd = 3)
>
> abline(lm.fit, lwd = 3, col = "blue")
>
> plot(predict(lm.fit), residuals(lm.fit))
>
> plot(predict(lm.fit), rstudent(lm.fit))
```

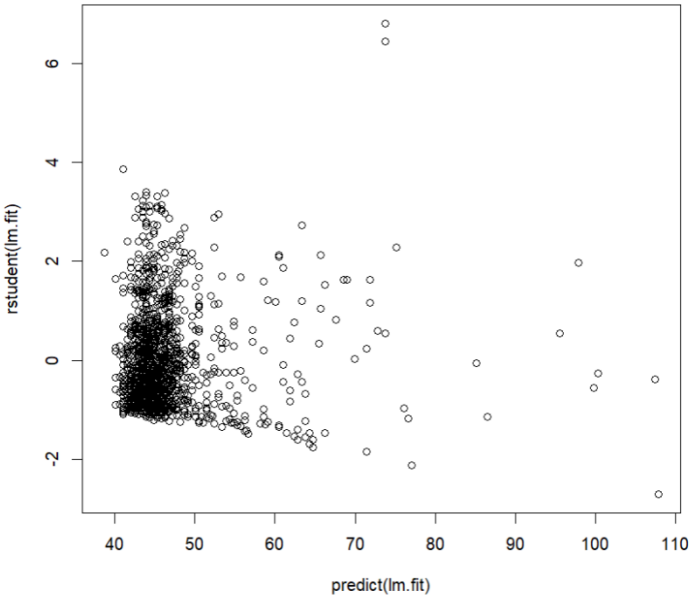

total sulfur dioxide
residual sugar
0
50
100
150
200
250
300
5
10
15


total sulfur dioxide
residual sugar
0
50
100
150
200
250
300
5
10
15


total sulfur dioxide
0
50
100
150
200
250
300
5
10
15
residual sugar


total sulfur dioxide
0
50
100
150
200
250
300
5
10
15
residual sugar



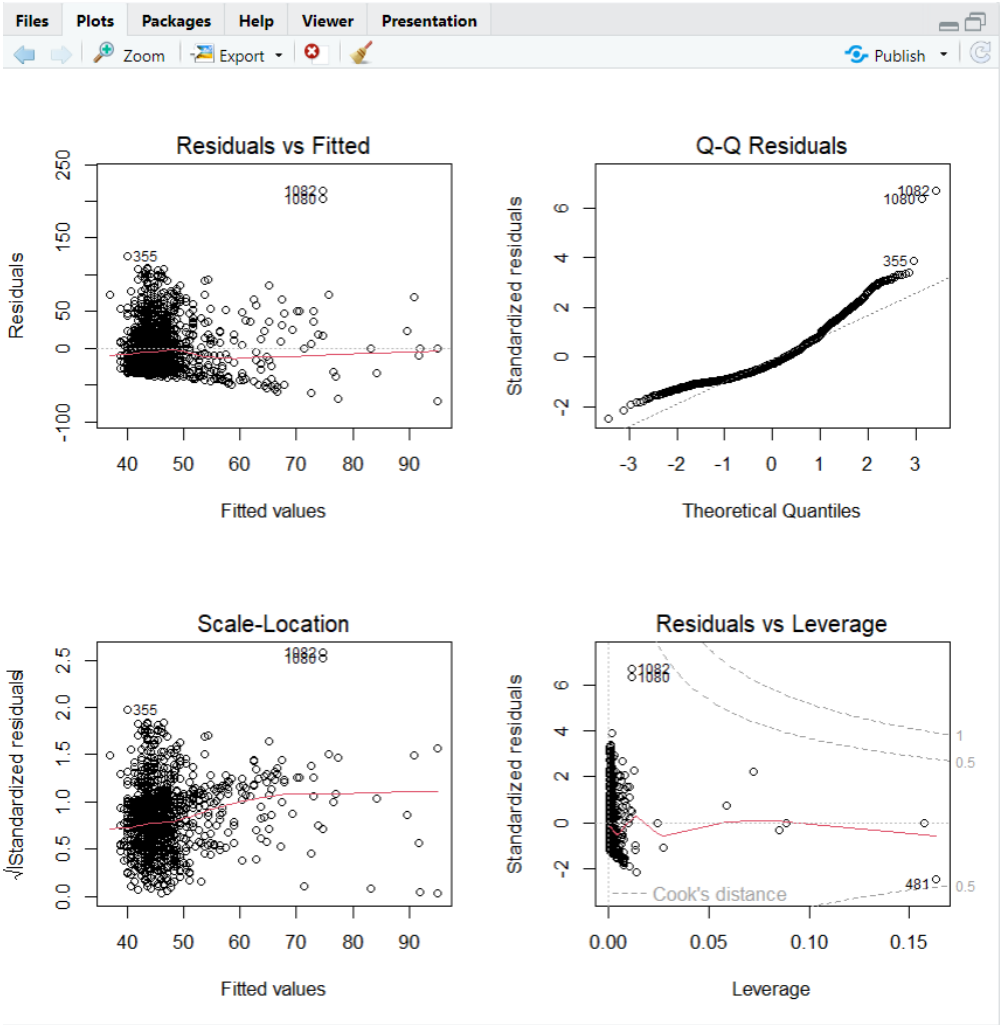



```
> lm.fit <- lm(`total sulfur dioxide` ~ `residual sugar`+ sulphates, data = winequality_red)
> summary(lm.fit)

Call:
lm(formula = `total sulfur dioxide` ~ `residual sugar` + sulphates,
    data = winequality_red)

Residuals:
    Min      1Q  Median      3Q     Max
-84.809 -23.102  -7.892  16.110 216.476

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       29.1133     3.5334   8.240 3.56e-16 ***
`residual sugar`   4.7315     0.5713   8.281 2.54e-16 ***
sulphates          8.1169     4.7523   1.708   0.0878 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 32.2 on 1596 degrees of freedom
Multiple R-squared:  0.04297,	Adjusted R-squared:  0.04177
F-statistic: 35.83 on 2 and 1596 DF,  p-value: 6.008e-16
```

```
> lm.fit <- lm(`total sulfur dioxide` ~ ., data = winequality_red)
> summary(lm.fit)

Call:
lm(formula = `total sulfur dioxide` ~ ., data = winequality_red)

Residuals:
    Min      1Q  Median      3Q     Max 
-55.355 -13.868  -4.457   8.186 177.013 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)          -1.969e+03  7.241e+02  -2.719  0.00663 ** 
`fixed acidity`      -8.052e+00  8.652e-01  -9.307  < 2e-16 ***
`volatile acidity`    3.099e+01  4.177e+00   7.420 1.90e-13 ***
`citric acid`         5.215e+01  4.867e+00  10.714  < 2e-16 ***
`residual sugar`      8.005e-01  5.133e-01   1.559  0.11912    
chlorides            -8.682e+01  1.428e+01  -6.081 1.50e-09 ***
`free sulfur dioxide` 1.967e+00  5.567e-02  35.335  < 2e-16 ***
density               2.241e+03  7.385e+02   3.035  0.00245 ** 
pH                   -5.193e+01  6.438e+00  -8.066 1.42e-15 ***
sulphates             8.628e+00  3.986e+00   2.164  0.03059 *  
alcohol              -1.938e+00  9.358e-01  -2.071  0.03855 *  
quality              -3.825e+00  8.539e-01  -4.480 8.00e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 22.18 on 1587 degrees of freedom
Multiple R-squared:  0.5484,	Adjusted R-squared:  0.5453 
F-statistic: 175.2 on 11 and 1587 DF,  p-value: < 2.2e-16
```

```
> lm.fit1 <- lm(`total sulfur dioxide` ~ . - sulphates, data = winequality_red)
> summary(lm.fit1)

Call:
lm(formula = `total sulfur dioxide` ~ . - sulphates, data = winequality_red)

Residuals:
    Min      1Q  Median      3Q     Max 
-56.906 -13.929  -4.403   8.001 176.087 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)          -2.363e+03  7.016e+02  -3.368 0.000774 ***
`fixed acidity`      -8.372e+00  8.535e-01  -9.810  < 2e-16 ***
`volatile acidity`    2.964e+01  4.134e+00   7.168 1.16e-12 ***
`citric acid`         5.265e+01  4.867e+00  10.817  < 2e-16 ***
`residual sugar`      5.778e-01  5.035e-01   1.148 0.251335    
chlorides            -7.569e+01  1.334e+01  -5.676 1.64e-08 ***
`free sulfur dioxide` 1.980e+00  5.543e-02  35.723  < 2e-16 ***
density               2.645e+03  7.155e+02   3.696 0.000226 ***
pH                   -5.372e+01  6.391e+00  -8.405  < 2e-16 ***
alcohol              -1.489e+00  9.136e-01  -1.629 0.103422    
quality              -3.469e+00  8.389e-01  -4.136 3.72e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 22.21 on 1588 degrees of freedom
Multiple R-squared:  0.5471,	Adjusted R-squared:  0.5442 
F-statistic: 191.8 on 10 and 1588 DF,  p-value: < 2.2e-16
```

```
> lm.fit1 <- update(lm.fit, ~. - sulphates)
> summary(lm(`total sulfur dioxide` ~ `residual sugar` * sulphates, data = winequality_red))

Call:
lm(formula = `total sulfur dioxide` ~ `residual sugar` * sulphates,
    data = winequality_red)

Residuals:
    Min      1Q  Median      3Q     Max
-84.594 -23.360  -7.872  15.719 209.351

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                 16.314      7.419   2.199 0.028017 *
`residual sugar`            10.087      2.790   3.616 0.000308 ***
sulphates                   27.588     11.004   2.507 0.012270 *
`residual sugar`:sulphates  -8.141      4.151  -1.961 0.049996 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 32.17 on 1595 degrees of freedom
Multiple R-squared:  0.04527,	Adjusted R-squared:  0.04348
F-statistic: 25.21 on 3 and 1595 DF,  p-value: 6.178e-16
> lm.fit2 <- lm(`total sulfur dioxide` ~ `residual sugar` + I(`residual sugar`^2))
> summary(lm.fit2)

Call:
lm(formula = `total sulfur dioxide` ~ `residual sugar` + I(`residual sugar`^2))

Residuals:
    Min      1Q  Median      3Q     Max
-72.136 -23.048  -8.398  15.879 214.380

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           31.2739     3.1361   9.972  < 2e-16 ***
`residual sugar`       6.4931     1.5820   4.104 4.26e-05 ***
I(`residual sugar`^2) -0.1531     0.1286  -1.191    0.234
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 32.22 on 1596 degrees of freedom
Multiple R-squared:  0.04207,	Adjusted R-squared:  0.04087
F-statistic: 35.05 on 2 and 1596 DF,  p-value: 1.27e-15
> lm.fit <- lm(`total sulfur dioxide` ~ `residual sugar`)
> anova(lm.fit, lm.fit2)
Analysis of Variance Table

Model 1: `total sulfur dioxide` ~ `residual sugar`
Model 2: `total sulfur dioxide` ~ `residual sugar` + I(`residual sugar`^2)
  Res.Df     RSS Df Sum of Sq      F Pr(>F)
1   1597 1657921
2   1596 1656450  1    1471.1 1.4174  0.234
> par(mfrow = c(2,2))
> plot(lm.fit2)
```




```
> lm.fit5 <- lm(`total sulfur dioxide` ~ poly(`residual sugar`, 5))
> summary(lm.fit5)

Call:
lm(formula = `total sulfur dioxide` ~ poly(`residual sugar`,
    5))

Residuals:
    Min      1Q  Median      3Q     Max
-90.677 -23.027  -8.536  16.041 198.493

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                46.4678     0.8029  57.872   <2e-16 ***
poly(`residual sugar`, 5)1 266.9797    32.1073   8.315   <2e-16 ***
poly(`residual sugar`, 5)2 -38.3549    32.1073  -1.195   0.2324
poly(`residual sugar`, 5)3 -51.2376    32.1073  -1.596   0.1107
poly(`residual sugar`, 5)4 -74.9902    32.1073  -2.336   0.0196 *
poly(`residual sugar`, 5)5  77.5085    32.1073   2.414   0.0159 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 32.11 on 1593 degrees of freedom
Multiple R-squared:  0.05032,	Adjusted R-squared:  0.04733
F-statistic: 16.88 on 5 and 1593 DF,  p-value: 2.748e-16
```

Results :

```
> summary(lm(`total sulfur dioxide` ~ log(`volatile acidity`), data = winequality_red))

Call:
lm(formula = `total sulfur dioxide` ~ log(`volatile acidity`),
    data = winequality_red)

Residuals:
    Min      1Q  Median      3Q     Max 
-47.446 -24.117  -8.779  15.330 246.895 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)              52.498      1.814  28.936  < 2e-16 ***
log(`volatile acidity`)   8.632      2.317   3.725 0.000202 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 32.76 on 1597 degrees of freedom
Multiple R-squared:  0.008614,	Adjusted R-squared:  0.007993 
F-statistic: 13.88 on 1 and 1597 DF,  p-value: 0.000202
```

2) Feature Selection / Model Optimization Methods :

Final Project.R × | Life_Expectancy_Data × | winequality_red ×

Filter

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	Bl
1	Afghanistan	2015	Developing	65.0	263	62	0.01	7.127962e+01	65	1154	
2	Afghanistan	2014	Developing	59.9	271	64	0.01	7.352358e+01	62	492	
3	Afghanistan	2013	Developing	59.9	268	66	0.01	7.321924e+01	64	430	
4	Afghanistan	2012	Developing	59.5	272	69	0.01	7.818422e+01	67	2787	
5	Afghanistan	2011	Developing	59.2	275	71	0.01	7.097109e+00	68	3013	
6	Afghanistan	2010	Developing	58.8	279	74	0.01	7.967937e+01	66	1989	
7	Afghanistan	2009	Developing	58.6	281	77	0.01	5.676222e+01	63	2861	
8	Afghanistan	2008	Developing	58.1	287	80	0.03	2.587393e+01	64	1599	
9	Afghanistan	2007	Developing	57.5	295	82	0.02	1.091016e+01	63	1141	
10	Afghanistan	2006	Developing	57.3	295	84	0.03	1.717152e+01	64	1990	
11	Afghanistan	2005	Developing	57.3	291	85	0.02	1.388648e+00	66	1296	
12	Afghanistan	2004	Developing	57.0	293	87	0.02	1.529607e+01	67	466	
13	Afghanistan	2003	Developing	56.7	295	87	0.01	1.108905e+01	65	798	
14	Afghanistan	2002	Developing	56.2	3	88	0.01	1.688735e+01	64	2486	
15	Afghanistan	2001	Developing	55.3	316	88	0.01	1.057473e+01	63	8762	
16	Afghanistan	2000	Developing	54.8	321	88	0.01	1.042496e+01	62	6532	
17	Albania	2015	Developing	77.8	74	0	4.60	3.649752e+02	99	0	
18	Albania	2014	Developing	77.5	8	0	4.51	4.287491e+02	98	0	
19	Albania	2013	Developing	77.2	84	0	4.76	4.308770e+02	99	0	
20	Albania	2012	Developing	76.9	86	0	5.14	4.124434e+02	99	9	
21	Albania	2011	Developing	76.6	88	0	5.37	4.370621e+02	99	28	
22	Albania	2010	Developing	76.2	91	1	5.28	4.182276e+01	99	10	
23	Albania	2009	Developing	76.1	91	1	5.79	3.480560e+02	98	0	
24	Albania	2008	Developing	75.3	1	1	5.61	3.662207e+01	99	0	

Showing 1 to 24 of 2,938 entries, 22 total columns

```
> library(readxl)
> Life_Expectancy_Data <- read_excel("C:/Users/tumma/Downloads/Life_Expectancy_Data.xlsx")
> View(Life_Expectancy_Data)
> library(ISLR2)
> library(readr)
> View(Life_Expectancy_Data)
> attach(Life_Expectancy_Data)
The following objects are masked from Life_Expectancy_Data (pos = 6):

    Adult Mortality, Alcohol, BMI, Country, Diphtheria, GDP, Hepatitis B, HIV/AIDS,
    Income composition of resources, infant deaths, Life expectancy, Measles,
    percentage expenditure, Polio, Population, Schooling, Status, thinness 1-19
    years, thinness 5-9 years, Total expenditure, under-five deaths, Year

> names(Life_Expectancy_Data)
 [1] "Country"                        "Year"
 [3] "Status"                         "Life expectancy"
 [5] "Adult Mortality"                "infant deaths"
 [7] "Alcohol"                        "percentage expenditure"
 [9] "Hepatitis B"                    "Measles"
[11] "BMI"                            "under-five deaths"
[13] "Polio"                          "Total expenditure"
[15] "Diphtheria"                     "HIV/AIDS"
[17] "GDP"                            "Population"
[19] "thinness  1-19 years"           "thinness 5-9 years"
[21] "Income composition of resources" "Schooling"
> dim(Life_Expectancy_Data)
[1] 2938   22
> sum(is.na(Life_Expectancy_Data$`Adult Mortality`))
[1] 10
> library(leaps)
```

Forward:

```
> regfit.fwd <- regsubsets(`Life expectancy` ~ ., data = Life_Expectancy_Data, nvmax = 11, method =
"forward")
Reordering variables and trying again:
Warning message:
In leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in = force.in,  :
  1  linear dependencies found
> summary(regfit.fwd)
Subset selection object
Call: regsubsets.formula(`Life expectancy` ~ ., data = Life_Expectancy_Data,
    nvmax = 11, method = "forward")
152 Variables  (and intercept)
                                Forced in Forced out
CountryAlbania                      FALSE      FALSE
CountryAlgeria                      FALSE      FALSE
CountryAngola                       FALSE      FALSE
CountryArgentina                    FALSE      FALSE
CountryArmenia                      FALSE      FALSE
CountryAustralia                    FALSE      FALSE
CountryAustria                      FALSE      FALSE
CountryAzerbaijan                   FALSE      FALSE
CountryBangladesh                   FALSE      FALSE
CountryBelarus                      FALSE      FALSE
CountryBelgium                      FALSE      FALSE
CountryBelize                       FALSE      FALSE
CountryBenin                        FALSE      FALSE
CountryBhutan                       FALSE      FALSE
CountryBosnia and Herzegovina       FALSE      FALSE
CountryBotswana                     FALSE      FALSE
CountryBrazil                       FALSE      FALSE
CountryBulgaria                     FALSE      FALSE
CountryBurkina Faso                 FALSE      FALSE
CountryBurundi                      FALSE      FALSE
CountryCabo Verde                   FALSE      FALSE
CountryCambodia                     FALSE      FALSE
CountryCameroon                     FALSE      FALSE
CountryCanada                       FALSE      FALSE
CountryCentral African Republic     FALSE      FALSE
CountryChad                         FALSE      FALSE
CountryChile                        FALSE      FALSE
CountryChina                        FALSE      FALSE
CountryColombia                     FALSE      FALSE
```

Backward :

```
> regfit.bwd <- regsubsets(`Life expectancy` ~ ., data = Life_Expectancy_Data, nvmax = 14, method =
"backward")
Reordering variables and trying again:
Warning message:
In leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in = force.in,  :
  1  linear dependencies found
> summary(regfit.bwd)
Subset selection object
Call: regsubsets.formula(`Life expectancy` ~ ., data = Life_Expectancy_Data,
    nvmax = 14, method = "backward")
152 Variables  (and intercept)
                                Forced in Forced out
CountryAlbania                      FALSE      FALSE
CountryAlgeria                      FALSE      FALSE
CountryAngola                       FALSE      FALSE
CountryArgentina                    FALSE      FALSE
CountryArmenia                      FALSE      FALSE
CountryAustralia                    FALSE      FALSE
CountryAustria                      FALSE      FALSE
CountryAzerbaijan                   FALSE      FALSE
CountryBangladesh                   FALSE      FALSE
CountryBelarus                      FALSE      FALSE
CountryBelgium                      FALSE      FALSE
CountryBelize                       FALSE      FALSE
CountryBenin                        FALSE      FALSE
CountryBhutan                       FALSE      FALSE
CountryBosnia and Herzegovina       FALSE      FALSE
CountryBotswana                     FALSE      FALSE
CountryBrazil                       FALSE      FALSE
CountryBulgaria                     FALSE      FALSE
CountryBurkina Faso                 FALSE      FALSE
CountryBurundi                      FALSE      FALSE
CountryCabo Verde                   FALSE      FALSE
CountryCambodia                     FALSE      FALSE
CountryCameroon                     FALSE      FALSE
CountryCanada                       FALSE      FALSE
CountryCentral African Republic     FALSE      FALSE
CountryChad                         FALSE      FALSE
CountryChile                        FALSE      FALSE
CountryChina                        FALSE      FALSE
```

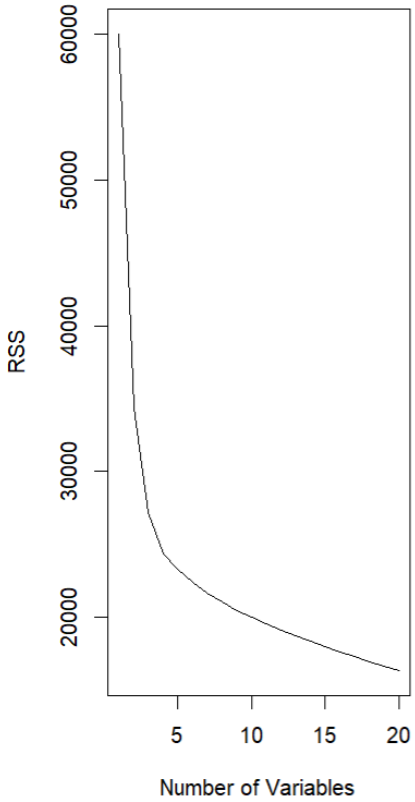
Forward:

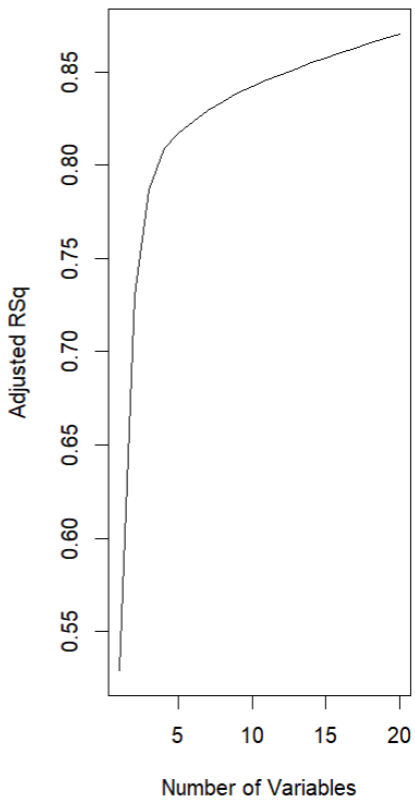
```
> regfit.fwd <- regsubsets(`Life expectancy` ~ ., data = Life_Expectancy_Data, nvmax = 19, method =
"forward")
Reordering variables and trying again:
Warning message:
In leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in = force.in,  :
  1  linear dependencies found
> summary(regfit.fwd)
Subset selection object
Call: regsubsets.formula(`Life expectancy` ~ ., data = Life_Expectancy_Data,
    nvmax = 19, method = "forward")
152 Variables  (and intercept)
                                Forced in Forced out
CountryAlbania                      FALSE      FALSE
CountryAlgeria                      FALSE      FALSE
CountryAngola                       FALSE      FALSE
CountryArgentina                    FALSE      FALSE
CountryArmenia                      FALSE      FALSE
CountryAustralia                    FALSE      FALSE
CountryAustria                      FALSE      FALSE
CountryAzerbaijan                   FALSE      FALSE
CountryBangladesh                   FALSE      FALSE
CountryBelarus                      FALSE      FALSE
CountryBelgium                      FALSE      FALSE
CountryBelize                       FALSE      FALSE
CountryBenin                        FALSE      FALSE
CountryBhutan                       FALSE      FALSE
CountryBosnia and Herzegovina       FALSE      FALSE
CountryBotswana                     FALSE      FALSE
CountryBrazil                       FALSE      FALSE
CountryBulgaria                     FALSE      FALSE
CountryBurkina Faso                 FALSE      FALSE
CountryBurundi                      FALSE      FALSE
CountryCabo Verde                   FALSE      FALSE
CountryCambodia                     FALSE      FALSE
CountryCameroon                     FALSE      FALSE
CountryCanada                       FALSE      FALSE
CountryCentral African Republic     FALSE      FALSE
CountryChad                         FALSE      FALSE
CountryChile                        FALSE      FALSE
CountryChina                        FALSE      FALSE
CountryColombia                     FALSE      FALSE
```

Backward:

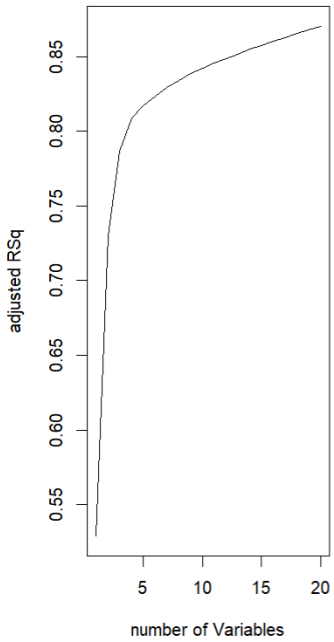
```
> regfit.bwd <- regsubsets(`Life expectancy` ~ ., data = Life_Expectancy_Data, nvmax = 19, method =
"backward")
Reordering variables and trying again:
Warning message:
In leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in = force.in,  :
  1  linear dependencies found
> summary(regfit.bwd)
Subset selection object
Call: regsubsets.formula(`Life expectancy` ~ ., data = Life_Expectancy_Data,
    nvmax = 19, method = "backward")
152 Variables  (and intercept)
                                Forced in Forced out
CountryAlbania                      FALSE      FALSE
CountryAlgeria                      FALSE      FALSE
CountryAngola                       FALSE      FALSE
CountryArgentina                    FALSE      FALSE
CountryArmenia                      FALSE      FALSE
CountryAustralia                    FALSE      FALSE
CountryAustria                      FALSE      FALSE
CountryAzerbaijan                   FALSE      FALSE
CountryBangladesh                   FALSE      FALSE
CountryBelarus                      FALSE      FALSE
CountryBelgium                      FALSE      FALSE
CountryBelize                       FALSE      FALSE
CountryBenin                        FALSE      FALSE
CountryBhutan                       FALSE      FALSE
CountryBosnia and Herzegovina       FALSE      FALSE
CountryBotswana                     FALSE      FALSE
CountryBrazil                       FALSE      FALSE
CountryBulgaria                     FALSE      FALSE
CountryBurkina Faso                 FALSE      FALSE
CountryBurundi                      FALSE      FALSE
CountryCabo Verde                   FALSE      FALSE
CountryCambodia                     FALSE      FALSE
CountryCameroon                     FALSE      FALSE
CountryCanada                       FALSE      FALSE
CountryCentral African Republic     FALSE      FALSE
CountryChad                         FALSE      FALSE
CountryChile                        FALSE      FALSE
CountryChina                        FALSE      FALSE
CountryColombia                     FALSE      FALSE
```

```
> reg.summaryfwd <- summary(regfit.fwd)
> names(reg.summaryfwd)
[1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
> reg.summaryfwd$rsq
 [1] 0.5294455 0.7303904 0.7870954 0.8091754 0.8174361 0.8242060 0.8302296 0.8349744 0.8394316
[10] 0.8433629 0.8467191 0.8499411 0.8529112 0.8561103 0.8588285 0.8617013 0.8643465 0.8668922
[19] 0.8693349 0.8717268
> par(mfrow = c(1,2))
> plot(reg.summaryfwd$rss, xlab = "Number of Variables", ylab = "RSS", type = "l")
> plot(reg.summaryfwd$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")
```

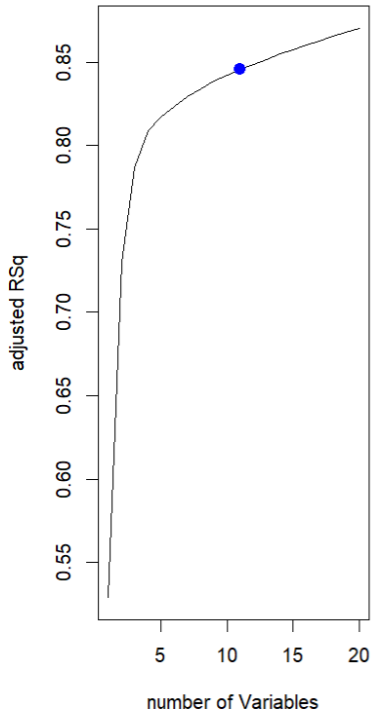




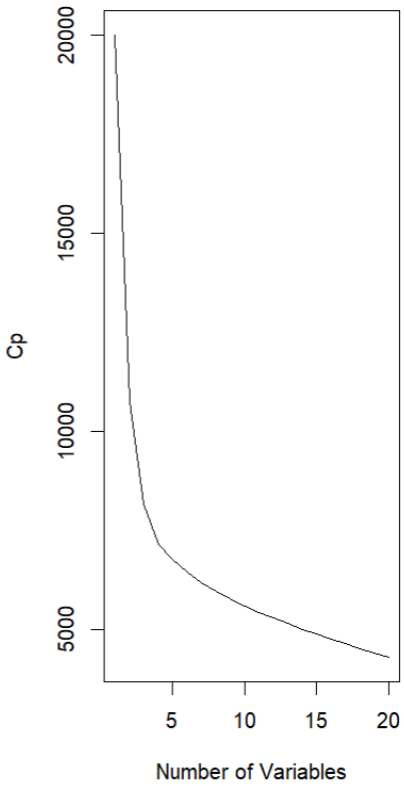

```
> which.max(reg.summaryfwd$adjr2)
[1] 20
> plot(reg.summaryfwd$adjr2, xlab = "number of Variables", ylab = "adjusted RSq", type = "l")
```



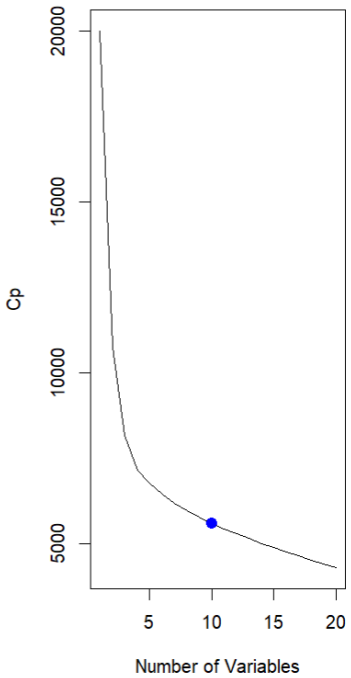
```
> points(11, reg.summaryfwd$adjr2[11], col = "blue",cex = 2, pch = 20)
```



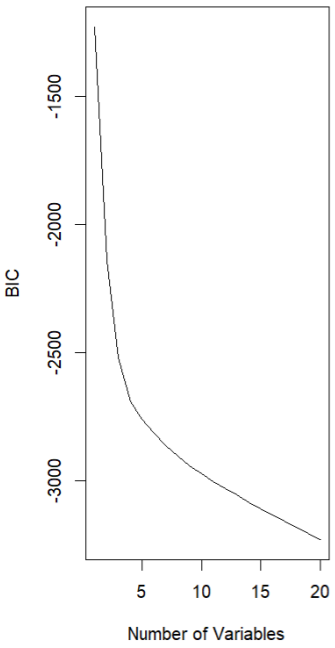

```
> plot(reg.summaryfwd$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")
```



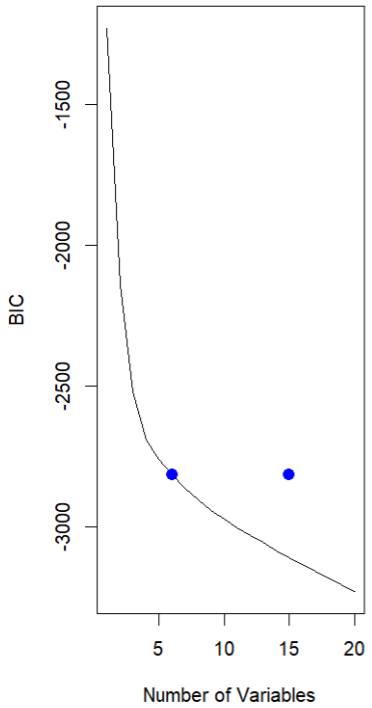
```
> which.min(reg.summaryfwd$cp)
[1] 20
> points(10, reg.summaryfwd$cp[10], col = "blue",cex = 2, pch = 20)
```



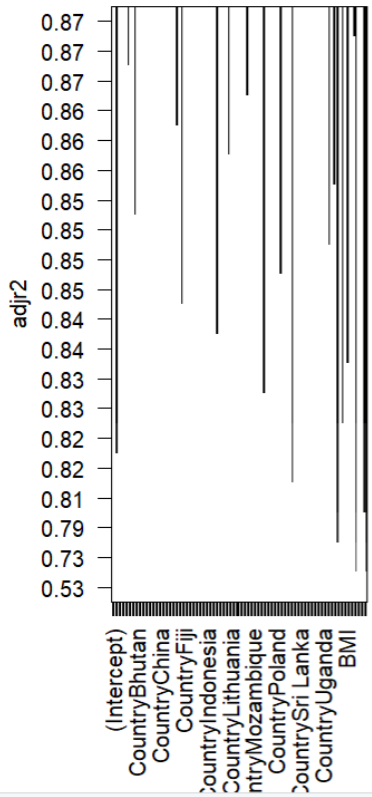

```
> which.min(reg.summaryfwd$bic)
[1] 20
> plot(reg.summaryfwd$bic, xlab = "Number of Variables", ylab = "BIC", type = "l")
```

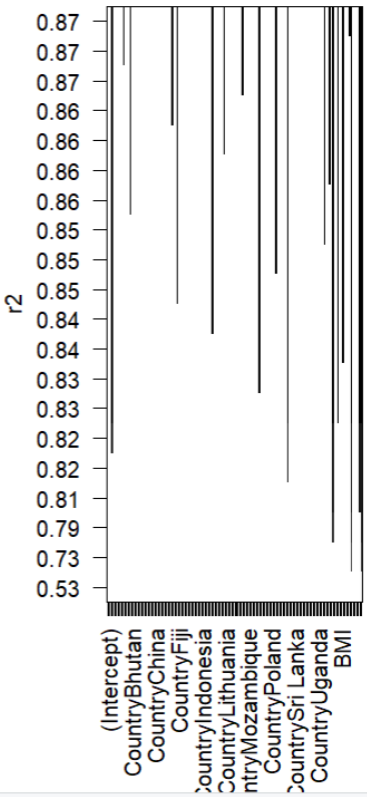


```
> points(15, reg.summaryfwd$bic[6], col = "blue",cex = 2, pch = 20)
> points(6, reg.summaryfwd$bic[6], col = "blue",cex = 2, pch = 20)
```

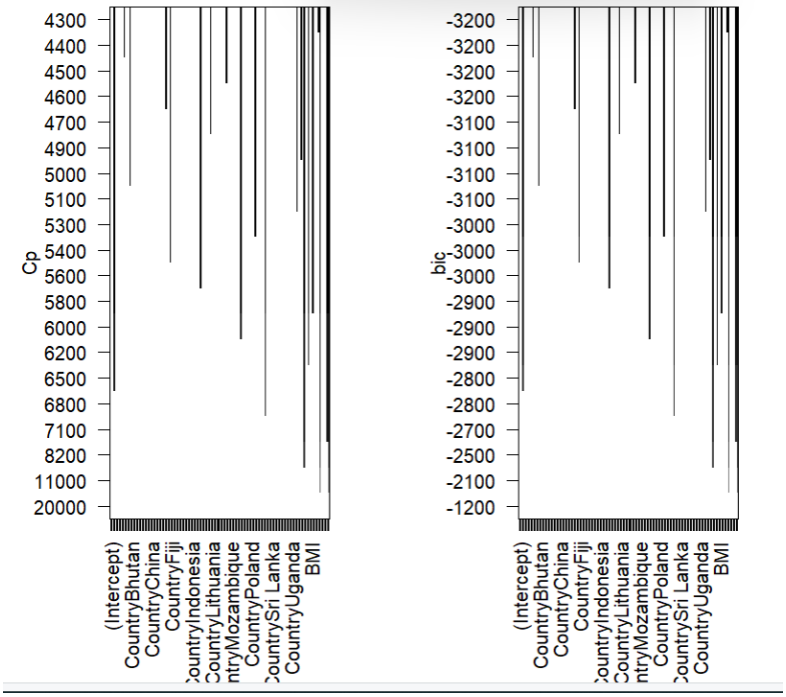



```
> plot(regfit.fwd,scale = "adjr2")
> plot(regfit.fwd,scale = "r2")
```





```
> plot(regfit.fwd,scale = "Cp")
> plot(regfit.fwd,scale = "bic")
```

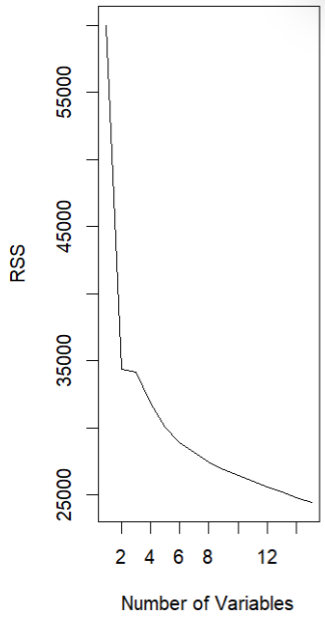


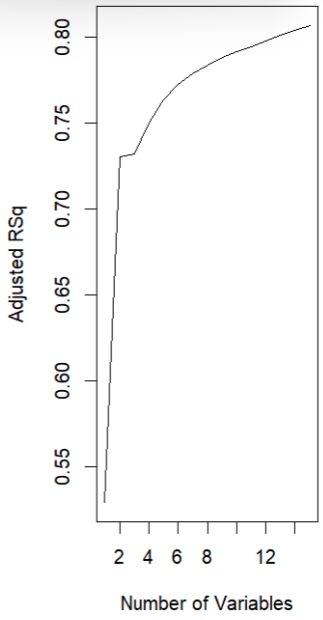

```
> coef(regfit.fwd, 6)
        (Intercept)       CountryAngola CountrySierra Leone      `infant deaths`
       4.642224e+01       -1.306146e+01       -1.383167e+01       -1.112165e-03
               GDP          Schooling     StatusDeveloping
       8.680005e-05        1.966364e+00       -1.486681e+00
```

Summary:

```
> regfit.bwd <- regsubsets(`Life expectancy` ~ ., data = Life_Expectancy_Data, nvmax = 14, method =
"backward")
Reordering variables and trying again:
Warning message:
In leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in = force.in,  :
  1  linear dependencies found
> summary(regfit.bwd)
Subset selection object
Call: regsubsets.formula(`Life expectancy` ~ ., data = Life_Expectancy_Data,
    nvmax = 14, method = "backward")
152 Variables  (and intercept)
                                 Forced in Forced out
CountryAlbania                       FALSE      FALSE
CountryAlgeria                       FALSE      FALSE
CountryAngola                        FALSE      FALSE
CountryArgentina                     FALSE      FALSE
CountryArmenia                       FALSE      FALSE
CountryAustralia                     FALSE      FALSE
CountryAustria                       FALSE      FALSE
CountryAzerbaijan                    FALSE      FALSE
CountryBangladesh                    FALSE      FALSE
CountryBelarus                       FALSE      FALSE
CountryBelgium                       FALSE      FALSE
CountryBelize                        FALSE      FALSE
CountryBenin                         FALSE      FALSE
CountryBhutan                        FALSE      FALSE
CountryBosnia and Herzegovina        FALSE      FALSE
CountryBotswana                      FALSE      FALSE
CountryBrazil                        FALSE      FALSE
CountryBulgaria                      FALSE      FALSE
```

```
> reg.summarybwd <- summary(regfit.bwd)
> plot(reg.summarybwd$rss, xlab = "Number of Variables", ylab = "RSS", type = "l")
> plot(reg.summarybwd$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")
```





3) Classification :

Final Project.R × | heart_disease × | Life_Expectancy_Data × | winequality_red ×

Filter

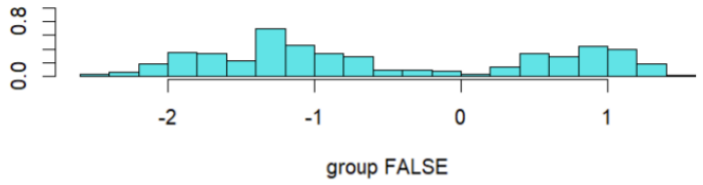
	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch	exang	oldpeak	slope
1	1	63	Male	Cleveland	typical angina	145	233	TRUE	lv hypertrophy	150	FALSE	2.3	down
2	2	67	Male	Cleveland	asymptomatic	160	286	FALSE	lv hypertrophy	108	TRUE	1.5	flat
3	3	67	Male	Cleveland	asymptomatic	120	229	FALSE	lv hypertrophy	129	TRUE	2.6	flat
4	4	37	Male	Cleveland	non-anginal	130	250	FALSE	normal	187	FALSE	3.5	down
5	5	41	Female	Cleveland	atypical angina	130	204	FALSE	lv hypertrophy	172	FALSE	1.4	upslo
6	6	56	Male	Cleveland	atypical angina	120	236	FALSE	normal	178	FALSE	0.8	upslo
7	7	62	Female	Cleveland	asymptomatic	140	268	FALSE	lv hypertrophy	160	FALSE	3.6	down
8	8	57	Female	Cleveland	asymptomatic	120	354	FALSE	normal	163	TRUE	0.6	upslo
9	9	63	Male	Cleveland	asymptomatic	130	254	FALSE	lv hypertrophy	147	FALSE	1.4	flat
10	10	53	Male	Cleveland	asymptomatic	140	203	TRUE	lv hypertrophy	155	TRUE	3.1	down
11	11	57	Male	Cleveland	asymptomatic	140	192	FALSE	normal	148	FALSE	0.4	flat
12	12	56	Female	Cleveland	atypical angina	140	294	FALSE	lv hypertrophy	153	FALSE	1.3	flat
13	13	56	Male	Cleveland	non-anginal	130	256	TRUE	lv hypertrophy	142	TRUE	0.6	flat
14	14	44	Male	Cleveland	atypical angina	120	263	FALSE	normal	173	FALSE	0.0	upslo
15	15	52	Male	Cleveland	non-anginal	172	199	TRUE	normal	162	FALSE	0.5	upslo
16	16	57	Male	Cleveland	non-anginal	150	168	FALSE	normal	174	FALSE	1.6	upslo
17	17	48	Male	Cleveland	atypical angina	110	229	FALSE	normal	168	FALSE	1.0	down
18	18	54	Male	Cleveland	asymptomatic	140	239	FALSE	normal	160	FALSE	1.2	upslo
19	19	48	Female	Cleveland	non-anginal	130	275	FALSE	normal	139	FALSE	0.2	upslo
20	20	49	Male	Cleveland	atypical angina	130	266	FALSE	normal	171	FALSE	0.6	upslo
21	21	64	Male	Cleveland	typical angina	110	211	FALSE	lv hypertrophy	144	TRUE	1.8	flat
22	22	58	Female	Cleveland	typical angina	150	283	TRUE	lv hypertrophy	162	FALSE	1.0	upslo
23	23	58	Male	Cleveland	atypical angina	120	284	FALSE	lv hypertrophy	160	FALSE	1.8	flat
24	24	58	Male	Cleveland	non-anginal	132	224	FALSE	lv hypertrophy	173	FALSE	3.2	upslo
25	25	60	Male	Cleveland	asymptomatic	130	206	FALSE	lv hypertrophy	132	TRUE	2.4	flat

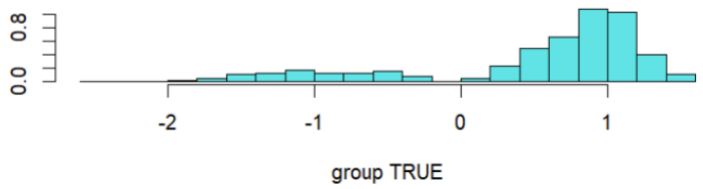
Showing 1 to 25 of 909 entries, 15 total columns

Linear Discriminant Analysis :

```
library('caTools')
library("ggplot2")
library(Metrics)
library(e1071)
df = read.csv('./heart_disease.csv')
library(MASS)
split = sample.split(df$cp, SplitRatio = 0.7)
trainingset = subset(df, split == TRUE)
testset = subset(df, split == FALSE)
lda.fit <- lda(exang ~ cp + age + sex, data = trainingset)
plot(lda.fit)
lda.pred <- predict(lda.fit, testset)
lda.class <- lda.pred$class
table(lda.class, testset$exang)
mean(lda.class == testset$exang)


glm.fits <- glm(
  exang ~  sex+age+chol,
  data = trainingset, family = binomial
)
summary(glm.fits)
glm.probs <- predict(glm.fits,testset, type = "response")
glm.pred <- rep("FALSE", 260)
glm.pred[glm.probs > .5] = "TRUE"
table(glm.pred, testset$exang)
print(mean(glm.pred == testset$exang))
print(mean(glm.pred != testset$exang))
```

Results :





Summary :

```
Call:
glm(formula = exang ~ sex + age + chol, family = binomial, data = trainingset)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5230  -1.0200  -0.6933   1.1862   2.1869

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.0150841  0.6282288  -6.391 1.65e-10 ***
sexMale      1.1193095  0.2468553   4.534 5.78e-06 ***
age          0.0479181  0.0097234   4.928 8.30e-07 ***
chol         0.0005631  0.0007916   0.711    0.477
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 783.19  on 584  degrees of freedom
Residual deviance: 733.34  on 581  degrees of freedom
  (51 observations deleted due to missingness)
AIC: 741.34

Number of Fisher Scoring iterations: 4

> glm.probs <- predict(glm.fits,testset, type = "response")
> glm.pred <- rep("FALSE", 260)
> glm.pred[glm.probs > .5] = "TRUE"
> table(glm.pred, testset$exang)

glm.pred FALSE TRUE
   FALSE   125   60
   TRUE     34   38
> print(mean(glm.pred == testset$exang))
[1] NA
> print(mean(glm.pred != testset$exang))
[1] NA
>
```

Logistic Regression Code :

```
library('caTools')
library("ggplot2")
library(Metrics)
library(e1071)
df = read.csv('./heart_disease.csv')
library(MASS)
split = sample.split(df$cp, SplitRatio = 0.7)
trainingset = subset(df, split == TRUE)
testset = subset(df, split == FALSE)
glm.fits <- glm(
  exang ~  sex+age+chol,
  data = trainingset, family = binomial
)
summary(glm.fits)
glm.probs <- predict(glm.fits,testset, type = "response")
glm.pred <- rep("FALSE", 260)
glm.pred[glm.probs > .5] = "TRUE"
table(glm.pred, testset$exang)
print(mean(glm.pred == testset$exang))
print(mean(glm.pred != testset$exang))
```

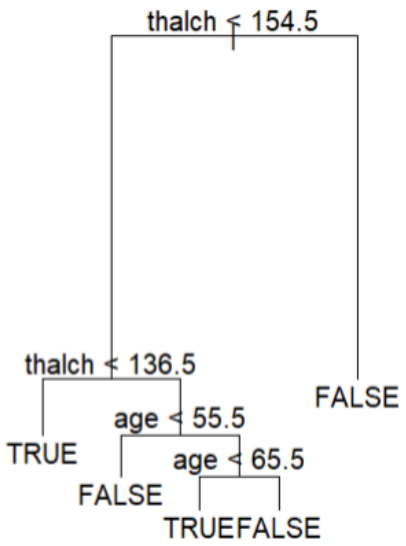
Results :

```
Call:
glm(formula = exang ~ sex + age + chol, family = binomial, data = trainingset)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6155  -1.0221  -0.7141   1.1608   2.1597

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.1090979  0.6336152  -6.485 8.86e-11 ***
sexMale      0.9887953  0.2291391   4.315 1.59e-05 ***
age          0.0535514  0.0099937   5.359 8.39e-08 ***
chol         0.0004553  0.0008069   0.564    0.573
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 793.09  on 585  degrees of freedom
Residual deviance: 741.80  on 582  degrees of freedom
  (50 observations deleted due to missingness)
AIC: 749.8

Number of Fisher Scoring iterations: 4

> glm.probs <- predict(glm.fits,testset, type = "response")
> glm.pred <- rep("FALSE", 260)
> glm.pred[glm.probs > .5] = "TRUE"
> table(glm.pred, testset$exang)

glm.pred FALSE TRUE
   FALSE   123   50
   TRUE     43   39
```

Tree Classifier :

```
library(tree)
df = read.csv('./heart_disease.csv')
head(df)

split = sample.split(df$cp, SplitRatio = 0.7)
trainingset = subset(df, split == TRUE)
testset = subset(df, split == FALSE)

tree.exang <- tree(as.factor(exang) ~ cp+age+sex+chol+fbs+thalch, trainingset)
summary(tree.exang)
plot(tree.exang)
text(tree.exang, pretty = 0)

tree.pred <- predict(tree.exang, testset,
                     type = "class" )
table(tree.pred, testset$exang)
```

Test Results :



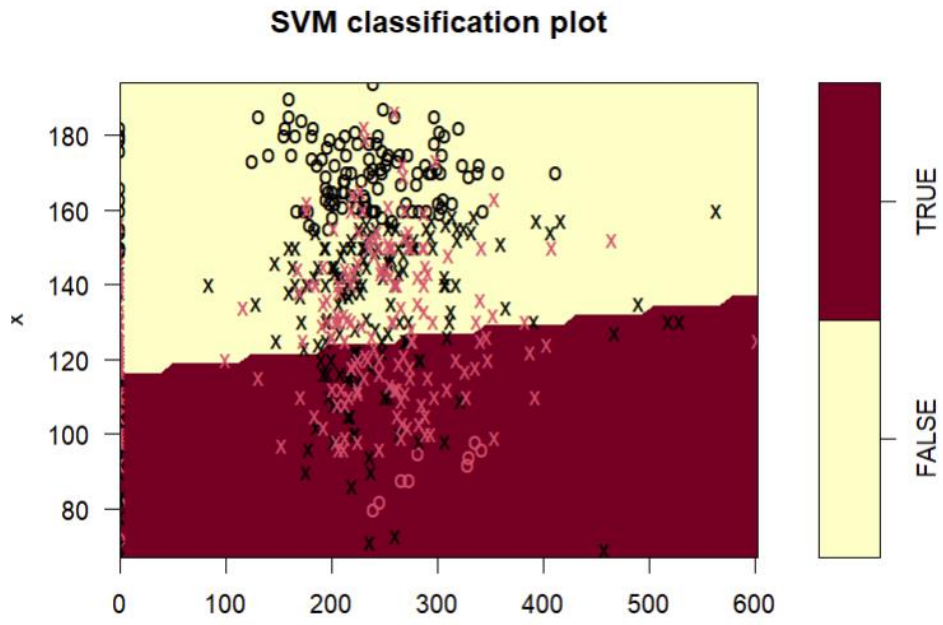
Summary :

```
Classification tree:
tree(formula = as.factor(exang) ~ cp + age + sex + chol + fbs +
    thalch, data = trainingset)
Variables actually used in tree construction:
[1] "thalch" "age"
Number of terminal nodes:  5
Residual mean deviance:  1.135 = 588.9 / 519
Misclassification error rate: 0.292 = 153 / 524
> plot(tree.exang)
> text(tree.exang, pretty = 0)
>
> tree.pred <- predict(tree.exang, testset,
+                      type = "class" )
Warning message:
In pred1.tree(object, tree.matrix(newdata)) : NAs introduced by coercion
> table(tree.pred, testset$exang)

tree.pred FALSE TRUE
    FALSE    85   21
    TRUE     65   84
```

SVM Linear Classifier :
Code :

```
library('caTools')
library("ggplot2")
library(Metrics)
library(e1071)
df = read.csv('./heart_disease.csv')
df<- df[-which(is.na(df$exang)), ]
data = data.frame(x = df$thalch , y = as.factor(df$exang), z = df$chol);
split = sample.split(data$z, SplitRatio = 0.6)
trainingset = subset(data, split == TRUE)
testset = subset(data, split == FALSE)
svmfit <- svm(y ~ x+z , data = trainingset, kernel = "linear",  cost = 1, scale = TRUE)
#generate a plot for submission in the next step.
summary(svmfit)
plot(svmfit, trainingset)
#using the built in tune function, we adjust the value of "cost" a few times (7 values) and see the
tune.out <- tune(svm, y ~ x+z, data = data, kernel = "linear",
                 ranges = list(cost = c(0.001, 0.01, 0.1, 1, 5, 10, 100)))
#we output the "best" performance model from the k-fold above
summary(tune.out)
#we copy that model and use summary again to show the parameters of that training
bestmod <- tune.out$best.model
summary(bestmod)
```

Test Results :



```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost
    1

- best performance: 0.3426493

- Detailed performance results:
   cost     error dispersion
1 1e-03 0.3946917 0.04673288
2 1e-02 0.3711531 0.05570206
3 1e-01 0.3462358 0.03581091
4 1e+00 0.3426493 0.03351449
5 5e+00 0.3438398 0.03382664
6 1e+01 0.3438398 0.03382664
7 1e+02 0.3426493 0.03534398

> #we copy that model and use summary again to show the parameters of that training
> bestmod <- tune.out$best.model
> summary(bestmod)

Call:
best.tune(METHOD = svm, train.x = y ~ x + z, data = data, ranges = list(cost = c(0.001,
    0.01, 0.1, 1, 5, 10, 100)), kernel = "linear")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  1

Number of Support Vectors:  614

 ( 308 306 )


Number of Classes:  2

Levels:
 FALSE TRUE
```

Part 4 Models

a) (5 points) A friend is starting a company and wants your help to see if they can figure out what factors most closely relate to the relative level of success for key competitors. They have gathered a few factors about each company such as total inventory, number of employees, annual operation budget and total profits. What method might you use to help your friend determine if their business model might be a success? Why did you choose this model?

   Total inventory, the number of employees, the annual operational budget, and overall earnings are the four indicators. Every variable appears to be linearly dependent. I would suggest using multiple linear regression analysis to gauge the likelihood that his business idea will be successful.

b) (5 points) An advertising firm has hired you to help them optimize their mailing list. They currently are looking to promote their client's store by sending packages of coupons to select areas. We want to know which postal codes the company should mail

to for maximum impact (shoppers come to the store with coupons). They currently have some survey data randomly sampled from homes in the area indicating how likely they were to shop at the client's location. What method might you try first to generate the mailing map? Why?

Logistic regression is a technique that can be used to estimate the likelihood that a customer would visit the client's store. Based on overall results, we can pinpoint the specific areas where mailing can have the biggest impact.

c) (5 points) A large company has been collecting data about their customers preferences for many years. They've hired you to help them transform the millions of samples and thousands of search and behavior features into a set of simplified features they can use to build a model which provides suggestions to their customers for future services. What method might you suggest first? Why?

We must translate the enormous volumes of data we now possess into simpler features. To gain additional insights, we may select the most effective regression procedures after utilizing PCA to determine a linear combination of variables.

d) (5 points) A company that specializes in shipping fruit to grocery stores wants to save money by sorting out bad fruit from good fruit before it goes on the truck. They have presented you with a device that can measure features like weight, color, size, and look for possible bad spots. Each of these measurements is imprecise, and there is significant overlap between the classes for most of the features. What supervised learning methods might you try? Why?

As was already said, there is a significant amount of class overlap in this categorization issue. I would have suggested multi-dimensional SVM if the characteristics could be easily separated. However, it seems that the random forest classifier can more accurately handle these overlaps.