```
> # ############################################################################
> # Title: MSIA 400 - Assignment 1
> # Date: 10/20/14
> # Author: Steven Lin
> ##############################################################################
>
>
> # Setup ####
>
> # My PC
> main = "C:/Users/Steven/Documents/Academics/3_Graduate School/2014-2015 ~ NU/"
>
> # Aginity
> #main = "\\\\nas1/labuser169"
>
> course = "MSIA_400_Analytics for Competitive Advantage"
> datafolder = "Lab/Assignment_01"
> setwd(file.path(main,course, datafolder))
>
> # Import data
> filename = "redwine.txt"
> redwine = read.table(filename, header=T)
>
>
> # Look at data
> names(redwine)
> head(redwine)
> nrow(redwine)
> summary(redwine)

>
```

## > # Problem 1

```
> 
> # Calculate the averages of RS and SD by ignoring the missing values
> RS_avg = mean(redwine$RS, na.rm = T)
> SD_avg = mean(redwine$SD, na.rm = T)
> RS_avg
[1] 2.537952
> SD_avg
[1] 46.29836
```

## > # Problem 2

```
> 
> # Create vectors of SD.obs and FS.obs by omitting observations
> # with missing values in SD
> 
> # T/F of missing values of SD
> missing_SD = is.na(redwine$SD)
> 
> # Create vectors for SD and FS
> SD.obs = redwine$SD[!missing_SD]
> FS.obs = redwine$FS[!missing_SD]
> 
> # Fit regression
> fit1 = lm(SD.obs ~ FS.obs)
> summary(fit1)

Call:
lm(formula = SD.obs ~ FS.obs)

Residuals:
    Min      1Q  Median      3Q     Max 
-54.489 -13.530  -7.155   7.252 197.587 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 13.18551    1.11502   11.82   <2e-16 ***
FS.obs       2.08608    0.05867   35.56   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 24.39 on 1580 degrees of freedom
Multiple R-squared:  0.4445,	Adjusted R-squared:  0.4441 
F-statistic:  1264 on 1 and 1580 DF,  p-value: < 2.2e-16

> 
> # Coefficients
> fit1$coeff
(Intercept)      FS.obs 
  13.185505    2.086077 
```

## > # Problem 3

```
> # FS values of the observations with missing SD values.
> FS.obs_missing_SD = redwine$FS[missing_SD]
> FS.obs_missing_SD
 [1] 15.0 12.0 11.0 12.0 40.5  1.0  7.0 35.0 15.0 36.0 23.0 12.0  8.0  7.0 15.0
18.0  5.0
> length(FS.obs_missing_SD)
[1] 17
>
> # Estimated SD values using the regression model
> SD.est = predict(fit1,data.frame(FS.obs =FS.obs_missing_SD))
> SD.est = as.vector(SD.est)
> SD.est
 [1] 44.47667 38.21843 36.13236 38.21843 97.67164 15.27158 27.78805 86.19821 44.
47667
[10] 88.28429 61.16528 38.21843 29.87412 27.78805 44.47667 50.73490 23.61589
>
> # Impute missing values of SD using the created vector.
> redwine.imputed = redwine
> redwine.imputed$SD[missing_SD] = SD.est

> # Print out the average of SD after the imputation
> mean(redwine.imputed$SD)
[1] 46.30182
```

## > # Problem 4

```
> # T/F of missing values of RS
> missing_RS = is.na(redwine.imputed$RS)
>
> # Impute missing values of RS using the average value imputation method
> redwine.imputed$RS[missing_RS] = RS_avg
> summary(redwine.imputed)
>
> # Print out the average of RS after the imputation
> mean(redwine.imputed$RS)
[1] 2.537952
```

## > # Problem 5

```
> # Build multiple linear regression model for the new data set
> # and save it as winemodel.
>
> winemodel = lm(QA ~ .,redwine.imputed)
>
> # Print out the coefficients of the regression model.
> coeff = winemodel$coeff
> coeff = as.matrix(winemodel$coeff)
> colnames(coeff) = 'Coefficient'
> coeff
              Coefficient
(Intercept)  47.202815335
FA            0.068406796
VA           -1.097686420
CA           -0.178949797
RS            0.025926958
CH           -1.631290466
FS            0.003530106
SD           -0.002854970
DE          -44.816652166
PH            0.035996993
SU            0.944871182
AL            0.247046550
```

## > # Problem 6

```
>
> # Printout the summary of the model.
> summary(winemodel)

Call:
lm(formula = QA ~ ., data = redwine.imputed)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.78010 -0.36249 -0.06331  0.44595  1.98828 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.720e+01  1.782e+01   2.649 0.008151 ** 
FA           6.841e-02  1.872e-02   3.654 0.000267 ***
VA          -1.098e+00  1.213e-01  -9.053  < 2e-16 ***
CA          -1.789e-01  1.474e-01  -1.214 0.224954    
RS           2.593e-02  1.419e-02   1.827 0.067944 .  
CH          -1.631e+00  4.097e-01  -3.982 7.14e-05 ***
FS           3.530e-03  2.159e-03   1.635 0.102262    
SD          -2.855e-03  7.248e-04  -3.939 8.54e-05 ***
DE          -4.482e+01  1.789e+01  -2.505 0.012329 *  
PH           3.600e-02  4.409e-02   0.816 0.414413    
SU           9.449e-01  1.136e-01   8.321  < 2e-16 ***
AL           2.470e-01  2.265e-02  10.906  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6491 on 1587 degrees of freedom
Multiple R-squared:  0.3584,	Adjusted R-squared:  0.354 
F-statistic:  80.6 on 11 and 1587 DF,  p-value: < 2.2e-16

>
> # Pick one attribute that is least likely to be related to QA based on p-value
s.
> p = as.matrix(sort(summary(winemodel)$coeff[-1,c("Pr(>|t|)")]))
> colnames(p)="p-value"
> p
        p-value
AL 9.316541e-27
VA 3.978528e-19
SU 1.859395e-16
CH 7.144969e-05
SD 8.544428e-05
FA 2.669015e-04
DE 1.232865e-02
RS 6.794396e-02
FS 1.022624e-01
CA 2.249543e-01
PH 4.144133e-01
>
> # CA, RS, FS and PH are insignificant at 0.05 level since p-values > 0.05,
> # suggesting that the coefficients are not significantly different than zero
> # and the effects on QA are insignificant
>
> # PH has the largest p-value = 0.414, indicating that is the attribute that
> # is least likely to be related to QA based on p-values ("the most insignifica
nt")
```

## > # Problem 7

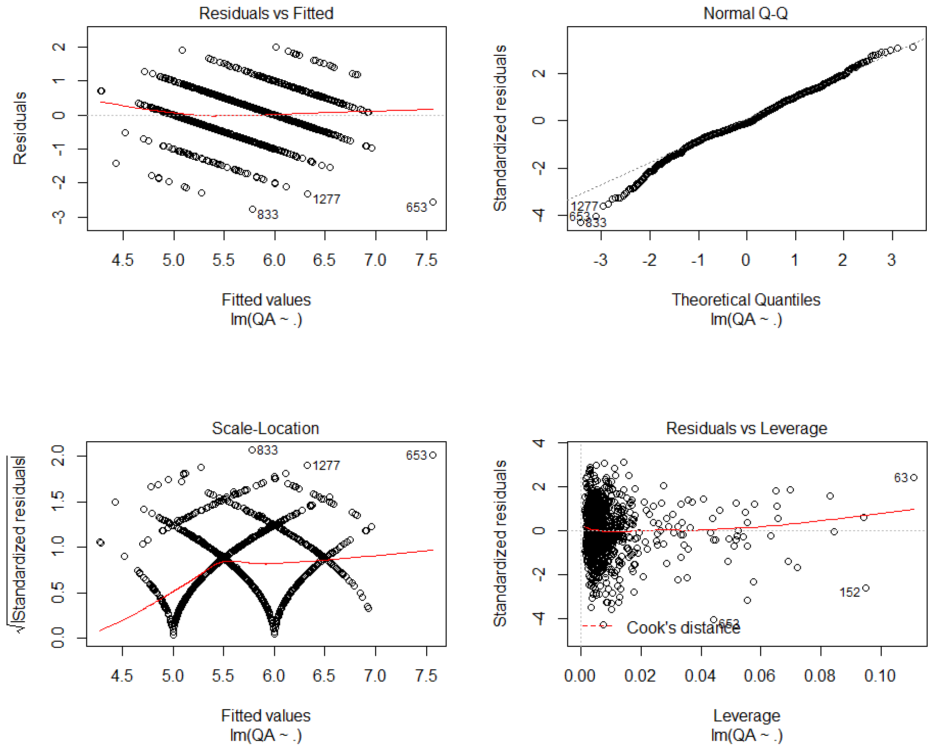
```
>
> # Calculate the average and standard deviation of the selected attribute
> PH_avg = mean(redwine.imputed$PH)
> PH_sd = sd(redwine.imputed$PH)
> PH_avg
[1] 3.306202
> PH_sd
[1] 0.3924948
>
> # boxplot(redwine.imputed$PH)
>
> # Create a new data set after removing observations that is outside of the
> # range [ m - 3s;m  + s3] and name the data set as redwine2.
>
> redwine2 = subset(redwine.imputed, (PH > PH_avg - 3*PH_sd) & (PH < PH_avg + 3*PH_sd))
>
> # Print out the dimension of redwine2 to know how many observations are removed.
> dim(redwine2)
[1] 1580   12
> dim(redwine2)[1]-dim(redwine)[1]
[1] -19
>
> # 19 observations removed
```

> # Problem 8

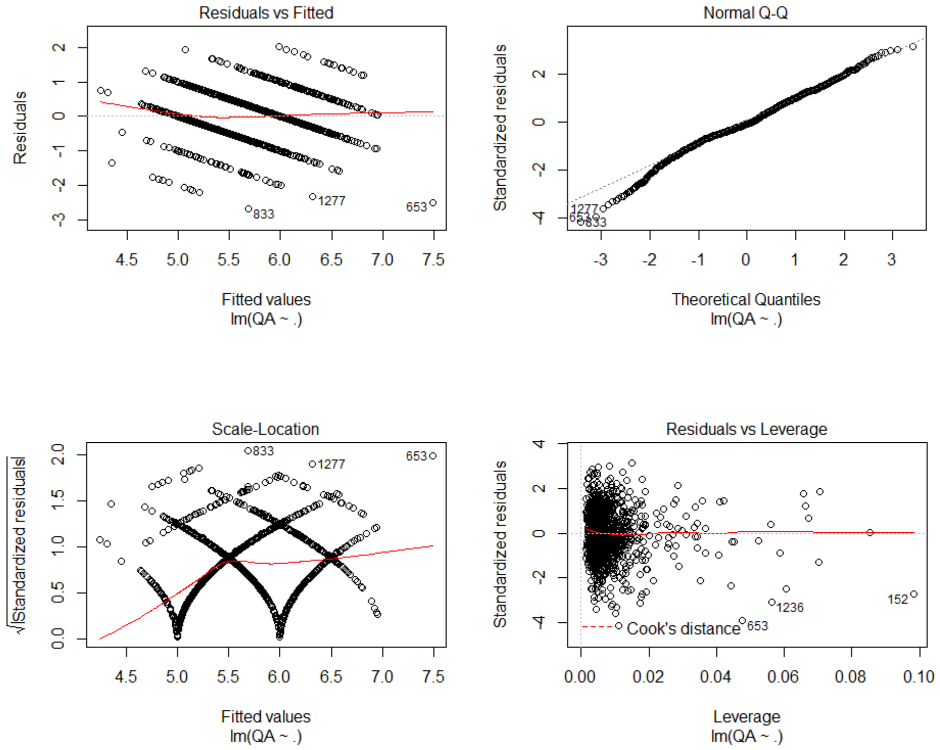
```
>
> # Build regression model winemodel2 using the new data set from Problem 7
> winemodel2 = lm(QA ~ . , redwine2)
>
> # print out the summary.
> summary(winemodel2)

Call:
lm(formula = QA ~ ., data = redwine2)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.68933 -0.36336 -0.04368  0.45221  2.01272 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  19.036170  21.211609   0.897   0.3696    
FA            0.024613   0.026019   0.946   0.3443    
VA           -1.072147   0.122031  -8.786  < 2e-16 ***
CA           -0.178017   0.148120  -1.202   0.2296    
RS            0.012955   0.014968   0.866   0.3869    
CH           -1.902552   0.420766  -4.522 6.60e-06 ***
FS            0.004421   0.002182   2.026   0.0429 *  
SD           -0.003145   0.000738  -4.261 2.16e-05 ***
DE          -14.973653  21.652465  -0.692   0.4893    
PH           -0.424704   0.192653  -2.205   0.0276 *  
SU            0.913456   0.114860   7.953 3.46e-15 ***
AL            0.282744   0.026553  10.648  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6475 on 1568 degrees of freedom
Multiple R-squared:  0.3629,	Adjusted R-squared:  0.3585 
F-statistic: 81.21 on 11 and 1568 DF,  p-value: < 2.2e-16

>
> # Compare this model with the model obtained in Problem 6 and decide
> # which one is better.
>
> # compare r squared
> summary(winemodel)$r.sq
[1] 0.3584256
> summary(winemodel2)$r.sq
[1] 0.3629441
>
> # both r.squared are too low, but model 2 has higher r-squared,
> # meaning it explains more variation in QA
>
> # both have 4 insignificant attributes at 0.05 level
> # model 1: CA, RS, FS, PH
> # model 2: FA, CA, RS, DE
>
> # both model have p-value < 2.2e-16 for the overall fit
>
> # looking at residuals
> plot(winemodel)

# leverage plot shows a higher influence of some data points in model 1


> # from the above discussion, both models are not very good, but
> # model 2 seems better than model 1.
```

```
>
> # Pick 5 attributes that is most likely to be related to QA based on p-values
> as.matrix(sort(summary(winemodel2)$coeff[-1,c("Pr(>|t|)")]))
           [p-values]
AL 1.298628e-25
VA 3.987768e-18
SU 3.461990e-15
CH 6.597172e-06
SD 2.158380e-05
PH 2.763384e-02
FS 4.292722e-02
CA 2.296053e-01
FA 3.443096e-01
RS 3.868817e-01
DE 4.893255e-01
> # The attributes with 5 lowest p-values (most signficant effect on QA) are:
> # AL, VA, SU, CH, SD
```

Model 1



Model 2


Residuals vs Fitted
Residuals
Fitted values
lm(QA ~ .)
833
1277
653
Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(QA ~ .)
1277
653
833
Scale-Location
√|Standardized residuals|
Fitted values
lm(QA ~ .)
833
1277
653
Residuals vs Leverage
Standardized residuals
Leverage
lm(QA ~ .)
1236
152
653
Cook's distance