400 Lab Assignment 2

Chuan Du (Sophie)

11/8/2018

```
#load data
redwine = read.table("redwine.txt", header = TRUE); head(redwine, 5)
```

```
##   QA   FA   VA   CA  RS    CH FS SD     DE   PH   SU  AL
## 1  5  7.4 0.70 0.00 1.9 0.076 11 34 0.9978 3.51 0.56 9.4
## 2  5  7.8 0.88 0.00 2.6 0.098 25 67 0.9968 3.20 0.68 9.8
## 3  5  7.8 0.76 0.04 2.3 0.092 15 54 0.9970 3.26 0.65 9.8
## 4  6 11.2 0.28 0.56 1.9 0.075 17 60 0.9980 3.16 0.58 9.8
## 5  5  7.4 0.70 0.00 1.9 0.076 11 34 0.9978 3.51 0.56 9.4
```

Problem 1

```
#remove NA
RS_avg = mean(redwine$RS, na.rm = TRUE); RS_avg
```

```
## [1] 2.537952
```

```
SD_avg = mean(redwine$SD, na.rm = TRUE); SD_avg
```

```
## [1] 46.29836
```

Answer: $avg(RS) = 2.537952$ and $avg(SD) = 46.29836$.

Problem 2

```
#find which obs in SD are NA
na_index = which(is.na(redwine$SD))
#remove these NA in SD
SD = na.omit(redwine$SD)
#remove FS obs with these indices
FS = redwine$FS[-na_index]
#fit the model
mod2 = lm(SD ~ FS)
mod2$coefficients
```

```
## (Intercept)          FS
##   13.185505    2.086077
```

Answer: The coefficients of the regression model is **13.185505** and **2.086077**.

Problem 3

```
FS.impute = redwine$FS[na_index]
SD.impute = coefficients(mod2)[1] + coefficients(mod2)[2] * FS.impute
redwine$SD[na_index] = SD.impute
mean(redwine$SD)
```

```
## [1] 46.30182
```

Answer: The average of SD after the imputation is **46.30182**.

Problem 4

```
#define avg value imputation
avg.imp = function(x, avg){
  missing = is.na(x)
  imputed = x
  imputed[missing] = avg
  return(imputed)
}

#apply the method to RS
RS_imp = avg.imp(redwine$RS, RS_avg)
mean(RS_imp)
```

```
## [1] 2.537952
```

Answer: The average of RS after the imputation is **2.537952**,

Problem 5

```
#fill in na of RS by avg imputation
redwine$RS = RS_imp
```

```
redwinemodel = lm(QA ~ ., data = redwine)
redwinemodel$coefficients
```

```
##   (Intercept)            FA            VA            CA            RS
##  47.202815335   0.068406796  -1.097686420  -0.178949797   0.025926958
##            CH            FS            SD            DE            PH
##  -1.631290466   0.003530106  -0.002854970 -44.816652166   0.035996993
##            SU            AL
##   0.944871182   0.247046550
```

Problem 6

```
summary(redwinemodel)
```

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

Answer: Based on the model summary, we could see that **PH** is a *non-significant* predictor and with the *largest p-value*, so **PH** is least likely to be related to QA.

Problem 7

```
CV_i = function(n, K){
  #n is sample size, k is number of folds
  #returns k-len lst of indices for each part
  m = floor(n/K) #approximate size of each part
  r = n - m*K
  I = sample(n, n) #random reordering of the indices
  Ind = list() #index for all k parts
  length(Ind) = K
  for (k in 1:K){
    if (k <= r)
      kpart = ((m+1)*(k-1)+1):((m+1)*k)
    else
      kpart = ((m+1)*r+m*(k-r-1)+1):((m+1)*r+m*(k-r))
    Ind[[k]] = I[kpart]  #indices for kth part of data
  }
  Ind
}
```

```
Nrep = 20 #repeat CV 20 times
K =  5 #5-fold cv
n = nrow(redwine)
y = redwine$QA
SSE = c()
for (j in 1:Nrep){
  Ind = CV_i(n, K)
  yhat = y
  for (k in 1:K){
    out = lm(QA ~., data = redwine[-Ind[[k]], ])
    yhat[Ind[[k]]] = as.numeric(predict(out, redwine[Ind[[k]], ]))
  }
  SSE = c(SSE, sum((y - yhat)^2))
}
SSE
```

```
##  [1] 682.2452 688.8997 688.2202 682.6521 679.1089 688.3628 679.0585
##  [8] 680.8830 681.6469 684.9561 683.9429 681.9997 681.2930 687.1228
## [15] 685.1354 680.3174 682.8281 681.0902 686.3080 681.2102
```

```
mean(SSE)
```

```
## [1] 683.3641
```

Problem 8

```
mu = mean(redwine$PH); mu
```

```
## [1] 3.306202
```

```
sigma = sd(redwine$PH); sigma
```

```
## [1] 0.3924948
```

```
redwine2 = subset(redwine, redwine$PH >= mu-3*sigma & redwine$PH <= mu+3*sigma)
dim(redwine2)
```

```
## [1] 1580   12
```

```
dim(redwine)[1] - dim(redwine2)[1]
```

```
## [1] 19
```

Answer: For the selected attribute *PH*, the average $\mu = 3.306202$, the standard deviation $\sigma = 0.3924948$. After removing observations that is outside the range $[\mu - 3\sigma, \mu + 3\sigma]$, we have the new dataset with dimension $1580 * 12$, and by comparing with the original dataset, we have removed **19** observations.

Problem 9

```
redwinemodel2 = lm(QA ~ ., data = redwine2)
summary(redwinemodel2)
```

```
##
## 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
```

Answer: By comparing the models, we could see that **the new model is better**, since R^2 increases, R^2_{adj} increases and F-statistics increases after we remove outliers and impute missing values. *VA, CH, SD, SU, AL* are the 5 attributes that are most likely to be related to QA based on p-values.