lab2-Yueying Zhang

```
redwine = read.table('redwine.txt', sep='\t', header=TRUE)
```

Problem 1

```
# calculate average of RS and SD
rs_mean <- mean(redwine$RS, na.rm=TRUE)
sd_mean <- mean(redwine$SD, na.rm=TRUE)
print(paste0('Average for RS: ', rs_mean))
```

```
## [1] "Average for RS: 2.53795180722892"
```

```
print(paste0('Average for SD: ', sd_mean))
```

```
## [1] "Average for SD: 46.2983565107459"
```

Problem 2

```
# simple linear regression of SD and FS
sd_vector <- as.vector(redwine[!is.na(redwine$SD), 'SD'])
fs_vector <- as.vector(redwine[!is.na(redwine$SD), 'FS'])
fit1 <- lm(sd_vector~fs_vector)
print(coefficients(fit1))
```

```
## (Intercept)   fs_vector
##   13.185505    2.086077
```

Problem 3

```
# caculated fitted value for missing value in SD
fs_17 <- as.vector(redwine[is.na(redwine$SD), 'FS'])
sd_missing <- coefficients(fit1)[1] + coefficients(fit1)[2] * fs_17
# impute missing SD
redwine$SD[which(is.na(redwine$SD))] <- sd_missing
# average of SD
sd_mean2 <- mean(redwine$SD)
print(paste0('Average of SD after imputation: ', sd_mean2))
```

```
## [1] "Average of SD after imputation: 46.3018196746507"
```

Problem 4

```
# replace missing value for RS with its mean
redwine$RS[which(is.na(redwine$RS))] <- rs_mean
rs_mean2 <- mean(redwine$RS)
print(paste0('Average of RS after imputation: ', rs_mean2))
```

```
## [1] "Average of RS after imputation: 2.53795180722892"
```

Problem 5

```
# fit multiple linear regression
winemodel <- lm(QA~FA+VA+CA+RS+CH+FS+SD+DE+PH+SU+AL, data=redwine)
print(coefficients(winemodel))
```

```
##   (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(winemodel)
```

```
##
## Call:
## lm(formula = QA ~ FA + VA + CA + RS + CH + FS + SD + DE + PH +
##     SU + AL, 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
```

PH is least likely to be related to QA since it has largest p-value (0.414413).

Problem 7

```
# 5-fold cross validation
CVInd <- function(n, K) {  #n is sample size; K is number of parts; returns K-length
 list 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()  #will be list of indices 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 #number of replicates of CV
K <- 5  #K-fold CV on each replicate
n <- nrow(redwine)
y <- redwine$QA
SSE <- matrix(0, Nrep, 1)
for (j in 1:Nrep) {
  Ind <- CVInd(n, K)
  yhat <- y
  for (k in 1:K) {
     winemodel_cv <- lm(QA~FA+VA+CA+RS+CH+FS+SD+DE+PH+SU+AL,
                        data=redwine[-Ind[[k]],])
     yhat[Ind[[k]]] <- as.numeric(predict(winemodel_cv, redwine[Ind[[k]],]))
  } #end of k loop
  SSE[j,] <- c(sum((y-yhat)^2))
} #end of j loop
print(paste0('Average SSE: ', apply(SSE, 2, mean)))
```

```
## [1] "Average SSE: 682.765408054435"
```

Problem 8

```
# identify and remove outliers in PH
ph_mean <- mean(redwine$PH)
ph_sd <- sd(redwine$PH)
upper <- ph_mean + 3 * ph_sd
lower <- ph_mean - 3 * ph_sd
redwine2 <- redwine[which(redwine$PH>=lower & redwine$PH<=upper),]
print(paste0('Mean: ', ph_mean))
```

```
## [1] "Mean: 3.30620200125078"
```

```
print(paste0('SD: ', ph_sd))
```

```
## [1] "SD: 0.39249479939923"
```

```
print(paste0('Previous dimension: ', dim(redwine)))
```

```
## [1] "Previous dimension: 1599" "Previous dimension: 12"
```

```
print(paste0('New dimension: ', dim(redwine2)))
```

```
## [1] "New dimension: 1580" "New dimension: 12"
```

```
print(paste0('Row numbers after deleting outliers: ', nrow(redwine2)))
```

```
## [1] "Row numbers after deleting outliers: 1580"
```

```
print(paste0('Rows removed: ', nrow(redwine)-nrow(redwine2)))
```

```
## [1] "Rows removed: 19"
```

Problem 9

```
winemodel2 <- lm(QA~FA+VA+CA+RS+CH+FS+SD+DE+PH+SU+AL, data=redwine2)
summary(winemodel2)
```

```
##
## Call:
## lm(formula = QA ~ FA + VA + CA + RS + CH + FS + SD + DE + PH +
##     SU + AL, 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
```

Model 2 with outlier removed has larger F-statistics and adjusted R-squared, so I would prefer model 2.

VA, CH, SD, SU, AL are most likely to correlate with QA.