Lab Assignment 2

Si Chen

11/8/2018

```
redwine <- read.table("redwine.txt", header = TRUE)
```

Problem 1

```
mean(redwine$RS, na.rm = TRUE)
```

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

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

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

The average of RS is 2.537952. The average of SD is 46.29836.

Problem 2

```
SD.obs <- na.omit(redwine$SD)
FS.obs <- redwine$FS[!is.na(redwine$SD)]
fit1 = lm(SD.obs ~ FS.obs)
coefficients(fit1)
```

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

Problem 3

```
FS.impute <- redwine$FS[is.na(redwine$SD)]
SD.impute <- coefficients(fit1)[2] * FS.impute + coefficients(fit1)[1]
redwine$SD[is.na(redwine$SD)] = SD.impute
mean(redwine$SD)
```

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

The average of SD after imputation is 46.30182.

Problem 4

```
avg.imp <- function (a, avg){
        missing <- is.na(a)
        imputed <- a
        imputed[missing] <- avg
```

```
        return (imputed)
   }
RS_avg = mean(na.omit(redwine$RS))
RS.avgimp = avg.imp(redwine$RS,RS_avg)
mean(RS.avgimp)
```

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

The average of RS after imputation is 2.537952.

Problem 5

```
redwine$RS <- RS.avgimp
fit = lm(QA ~ ., data = redwine)
coefficients(fit)
```

```
##   (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(fit)
```

```
##
## 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
```

PH has the largest p-value, therefore is the least likely to be related to QA.

Problem 7

```
CVInd <- function(n,K) {
   m<-floor(n/K)
   r<-n-m*K
   I<-sample(n,n)
   Ind<-list()
   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
K <- 5
n = nrow(redwine)
y<-redwine$QA
SSE <- c()
for (j in 1:Nrep) {
  Ind<-CVInd(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] 688.4381 687.9600 688.4864 686.4649 688.8080 684.7336 682.4908
##  [8] 683.4638 678.3263 685.2283 683.8695 686.1115 679.3683 685.0414
## [15] 679.7312 680.3740 685.8111 690.7125 683.0603 677.9944
```

```
mean(SSE)
```

```
## [1] 684.3237
```

I think this method is wrong.

CVInd() seperates y randomly, which means for every ordered $\hat{y}$, the corresponding y is wrong.

I will use the right method below.

```
CVInd <- function(n,K) {
   m<-floor(n/K)
   r<-n-m*K
```

```
   I<-sample(n,n)
   Ind<-list()
   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
K <- 5
n = nrow(redwine)
y<-redwine$QA
SSE <- c()
for (j in 1:Nrep) {
  Ind<-CVInd(n,K)
  for (k in 1:K) {
     out <- lm(QA~.,data = redwine[-Ind[[k]],])
     yhat <- as.numeric(predict(out,redwine[Ind[[k]],]))
     SSE <- c(SSE,sum((redwine$QA[Ind[[k]]]-yhat)^2))
  }
}
SSE
```

```
##   [1] 139.0222 140.2214 137.7259 156.1666 108.5568 141.3541 146.5847
##   [8] 138.8803 133.7565 124.9640 122.0318 136.3954 139.7025 140.1757
##  [15] 150.4647 112.8324 134.7392 135.5146 149.7586 149.7776 149.5035
##  [22] 129.2524 135.7468 127.4637 140.7371 157.1592 133.5216 119.7352
##  [29] 155.1237 116.6793 146.2935 136.6683 136.3813 138.5563 130.2270
##  [36] 115.8588 135.0473 156.0239 127.8010 150.5093 120.3105 146.7801
##  [43] 141.6756 124.2845 147.0880 134.9024 153.8569 124.4355 126.8530
##  [50] 140.7927 137.6697 125.8730 137.0744 127.6974 156.1366 140.4758
##  [57] 127.8666 143.0953 123.9153 145.6024 142.4115 150.6612 124.4579
##  [64] 129.3031 136.0975 135.6658 127.9075 135.9075 141.1818 140.4445
##  [71] 130.7105 155.8067 127.5383 128.4219 138.9777 140.0869 129.3457
##  [78] 138.4344 151.7981 125.0306 162.5100 137.4657 126.1681 132.9774
##  [85] 123.1415 142.4440 126.2539 151.7570 131.0720 127.6091 115.1421
##  [92] 148.5380 139.9354 143.0015 132.5959 128.9457 146.2129 121.3062
##  [99] 156.0572 129.6236
```

```
mean(SSE)
```

```
## [1] 136.5625
```

Probelem 8

```
PH.mean = mean(redwine$PH)
PH.sd = sd(redwine$PH)
PH.mean
```

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

```
PH.sd
```

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

```
PH.lb = PH.mean - 3 * PH.sd
PH.ub = PH.mean + 3 * PH.sd
redwine2 <- subset(redwine, redwine$PH < PH.ub & redwine$PH > PH.lb)
dim(redwine2)
```

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

```
dim(redwine)
```

```
## [1] 1599   12
```

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

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

The average of PH is 3.306202. The standard deviation of PH is 0.3924948. redwine2 has 1580 rows. 19 observations are removed.

Problem 9

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

```
##
## 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
```

The new model is better because the R-squared increases. VA, CH, SD, SU, AL are attributes having the 5 lowest p-values, so they are most likely to be related to QA.