HW3 - Exercise 5

Zining Fan(zf2234), Mutian Wang(mw3386), Siyuan Wang(sw3418)

4/13/2020

```
knitr::opts_chunk$set(echo = TRUE)
```

Question 1:

```
df <- cars
```

```
df$speed_2 <- df$speed*df$speed
```

```
model <- lm(formula = dist ~ speed+speed_2, data = df)
summary(model)
```

```
##
## Call:
## lm(formula = dist ~ speed + speed_2, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.720  -9.184  -3.188   4.628  45.152
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.47014   14.81716   0.167    0.868
## speed        0.91329    2.03422   0.449    0.656
## speed_2      0.09996    0.06597   1.515    0.136
##
## Residual standard error: 15.18 on 47 degrees of freedom
## Multiple R-squared:  0.6673, Adjusted R-squared:  0.6532
## F-statistic: 47.14 on 2 and 47 DF,  p-value: 5.852e-12
```

If we judged by the hypothesis testing methods, we can see that the p values of all the terms are not small enough for us to reject the null hypothesis. We now explore them with AIC methods.

```
AIC(model)
```

```
## [1] 418.7721
```

```
model_1 <- lm(formula = dist ~ 1, data = df)
model_2 <- lm(formula = dist ~ speed, data = df)
model_3 <- lm(formula = dist ~ speed_2, data = df)
model_4 <- lm(formula = dist ~ speed+speed_2, data = df)
model_5 <- lm(formula = dist ~ 0+speed, data = df)
model_6 <- lm(formula = dist ~ 0+speed_2, data = df)
model_7 <- lm(formula = dist ~ 0+speed+speed_2, data = df)
```

```r
print(AIC(model_1))
```

```
## [1] 469.8024
```

```r
print(AIC(model_2))
```

```
## [1] 419.1569
```

```r
print(AIC(model_3))
```

```
## [1] 416.986
```

```r
print(AIC(model_4))
```

```
## [1] 418.7721
```

```r
print(AIC(model_5))
```

```
## [1] 423.7498
```

```r
print(AIC(model_6))
```

```
## [1] 419.6579
```

```r
print(AIC(model_7))
```

```
## [1] 416.8016
```

We can now observed that model_7 has the smallest AIC value. So the most appropriate model is to throw away the intercept term and keep speed and square of speed.

Question 2: Based on the selected model from Question 1, we now observe the parameters from model_7.

```r
summary(model_7)
```

```
##
## Call:
## lm(formula = dist ~ 0 + speed + speed_2, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.836  -9.071  -3.152   4.570  44.986
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## speed    1.23903    0.55997   2.213  0.03171 *
## speed_2  0.09014    0.02939   3.067  0.00355 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.02 on 48 degrees of freedom
## Multiple R-squared:  0.9133, Adjusted R-squared:  0.9097
## F-statistic: 252.8 on 2 and 48 DF,  p-value: < 2.2e-16
```

So the esimate of the speed term represent the reaction time of the driver, which is 1.23903 s.

Question 3:

```
library(pracma)
QR_ls <- function(y,X) {
  qrx <- qr(X)
  Q <- qr.Q(qrx,complete=TRUE)
  R <- qr.R(qrx)
  R_inv <- inv(R)
  p <- ncol(X)
  f_raw <- t(Q) %*% y
  f <- head(f_raw,p)
  return(R_inv %*% f)
}
```

Here we finish writing the R function.

Question 4:

```
X <- model.matrix(dist ~speed + I(speed^2),cars)
y <- cars$dist
QR_ls(y,X)
```

```
##                   [,1]
## (Intercept) 2.4701378
## speed       0.9132876
## I(speed^2)  0.0999593
```

```
model <- lm(formula = dist ~ speed+speed_2, data = df)
print(model)
```

```
##
## Call:
## lm(formula = dist ~ speed + speed_2, data = df)
##
## Coefficients:
## (Intercept)        speed      speed_2
##     2.47014      0.91329      0.09996
```

Here we see that the result of QR decomposition method is exactly the same as those from the lm function. Our answer is validated.

Question 5:

```
summary(model)
```

```
##
## Call:
## lm(formula = dist ~ speed + speed_2, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -28.720  -9.184  -3.188   4.628  45.152
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.47014   14.81716   0.167    0.868
## speed        0.91329    2.03422   0.449    0.656
## speed_2      0.09996    0.06597   1.515    0.136
##
## Residual standard error: 15.18 on 47 degrees of freedom
## Multiple R-squared:  0.6673, Adjusted R-squared:  0.6532
## F-statistic: 47.14 on 2 and 47 DF,  p-value: 5.852e-12
```

```r
library(pracma)
QR_se <- function(y,X) {
  qrx <- qr(X)
  Q <- qr.Q(qrx,complete=TRUE)
  R <- qr.R(qrx)
  R_inv <- inv(R)
  p <- ncol(X)
  n <- nrow(X)
  f_raw <- t(Q) %*% y
  f <- head(f_raw,p)
  r <- tail(f_raw,n-p)
  beta <- R_inv %*% f
  sigma_2 <- Norm(r,p=2)**2/(n-p)
  se_beta <- (R_inv %*% t(R_inv))*sigma_2
  return(sqrt(diag(se_beta)))
}
QR_sigma <- function(y,X) {
  qrx <- qr(X)
  Q <- qr.Q(qrx,complete=TRUE)
  R <- qr.R(qrx)
  R_inv <- inv(R)
  p <- ncol(X)
  n <- nrow(X)
  f_raw <- t(Q) %*% y
  f <- head(f_raw,p)
  r <- tail(f_raw,n-p)
  sigma_2 <- Norm(r,p=2)**2/(n-p)
  return(sigma_2)
}
QR_beta <- function(y,X) {
  qrx <- qr(X)
  Q <- qr.Q(qrx,complete=TRUE)
  R <- qr.R(qrx)
  R_inv <- inv(R)
  p <- ncol(X)
  n <- nrow(X)
  f_raw <- t(Q) %*% y
  f <- head(f_raw,p)
  r <- tail(f_raw,n-p)
  beta <- R_inv %*% f
  return(beta)
}
```

```r
X <- model.matrix(dist ~speed + I(speed^2),cars)
y <- cars$dist
print("coefficient")
```

```
## [1] "coefficient"
```

```r
print(QR_beta(y,X))
```

```
##                  [,1]
## (Intercept) 2.4701378
## speed       0.9132876
## I(speed^2)  0.0999593
```

```r
print("estimated residual variance")
```

```
## [1] "estimated residual variance"
```

```r
print(QR_sigma(y,X))
```

```
## [1] 230.3131
```

```r
print("standard error of parameter estimators")
```

```
## [1] "standard error of parameter estimators"
```

```r
print(QR_se(y,X))
```

```
## (Intercept)       speed  I(speed^2)
## 14.81716473  2.03422044  0.06596821
```

Here we can see that our function produce the same results from the lm.

Question 6:

```r
q <- QR_beta(y,X)/QR_se(y,X)
n <- nrow(X)
p <- ncol(X)
print(2*pt(q, n-p, lower.tail = FALSE))
```

```
##                  [,1]
## (Intercept) 0.8683151
## speed       0.6555224
## I(speed^2)  0.1364024
```

Here we produce the same results as those from the lm function.