STAT5703 HW3 Ex2

Wen Fan(wf2255), Hanjun Li(hl3339), Banruo Xie(bx2168)

Exercise 2

Question 1

```r
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-28. For overview type 'help("mgcv-package")'.
```

```r
cars$speed_sqr <- (cars$speed)^2
lr = lm(dist ~ speed + speed_sqr , data=cars)
summary(lr)
```

```
##
## Call:
## lm(formula = dist ~ speed + speed_sqr, data = cars)
##
## 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_sqr    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
AIC(lr)
```

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

In this model, we can see from the summary table above that the model is significant and all features are significant as well. However, the R-squared value is not very high. Thus, we drop the relatively unsignificant variable 'speed' using 'stepwise' method. After dropping, we can see the AIC value of the model decreases, while both of the variables in the model become significant.

```r
lr2 = step(lm(dist ~ speed_sqr+speed , data=cars))
```

```
## Start:  AIC=274.88
## dist ~ speed_sqr + speed
##
##             Df Sum of Sq   RSS    AIC
## - speed      1     46.42 10871 273.09
## <none>                   10825 274.88
## - speed_sqr  1    528.81 11354 275.26
##
## Step:  AIC=273.09
## dist ~ speed_sqr
```

```
##
##             Df Sum of Sq   RSS    AIC
## <none>                   10871 273.09
## - speed_sqr  1     21668 32539 325.91
```

```r
summary(lr2)
```

```
##
## Call:
## lm(formula = dist ~ speed_sqr, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.448  -9.211  -3.594   5.076  45.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.86005    4.08633   2.168   0.0351 *
## speed_sqr    0.12897    0.01319   9.781  5.2e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.05 on 48 degrees of freedom
## Multiple R-squared:  0.6659, Adjusted R-squared:  0.6589
## F-statistic: 95.67 on 1 and 48 DF,  p-value: 5.2e-13
```

```r
AIC(lr2)
```

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

According to above results, we would choose speed-squared only model as the best model. #### Question 2

As we drop the variable 'speed' in point 1, we could only get value of 'reaction time' in the residual of the formulation.

$$time = \frac{dist - \hat{\beta}_0 - \hat{\beta}_1 * speed}{speed}$$

By getting its value, we can estimate its distribution. #### Question 3

```r
lr_1 <- function(y, X)
  {
  qrx <- qr(X)
  Q <- qr.Q(qrx,complete=TRUE)
  QR <- qr.R(qrx)
  return(backsolve(QR, (t(Q) %*% y)))
  }
```

Question 4

```r
newM2 <- model.matrix(dist ~speed + I(speed^2),cars)
lr_1(cars$dist,newM2)
```

```
##           [,1]
## [1,] 2.4701378
## [2,] 0.9132876
## [3,] 0.0999593
```

Therefore, the function gives the right result for the coffecient for the model. #### Question 5

```r
lr_2 <- function (X, y) {
  qrx <- qr(X) ## returns a QR decomposition object
  Q <- qr.Q(qrx,complete=TRUE) ## extract Q
  R <- qr.R(qrx) ## extract R
  f <- t(Q)%*%y
  f <- f[1:ncol(X),]
  beta <- solve(R)%*%f
  residual <- y-X%*%beta
  sigma <- as.vector(t(residual)%*%residual/(nrow(X)-ncol(X)))
  variance <- solve(R)%*%t(solve(R))*sigma
  list(coefficient=beta,std_error=sqrt(as.matrix(diag(variance),ncol=ncol(X))),
       residual_variance=sigma)
}
newM2 <- model.matrix(dist ~speed + I(speed^2),cars)
lr_2(newM2,cars$dist)
```

```
## $coefficient
##                  [,1]
## (Intercept) 2.4701378
## speed       0.9132876
## I(speed^2)  0.0999593
##
## $std_error
##                    [,1]
## (Intercept) 14.81716473
## speed        2.03422044
## I(speed^2)   0.06596821
##
## $residual_variance
## [1] 230.3131
```

Therefore, the function gives the right result for the model. #### Question 6

```r
lr_3 <- function (X, y) {
  qrx <- qr(X) ## returns a QR decomposition object
  Q <- qr.Q(qrx,complete=TRUE) ## extract Q
  R <- qr.R(qrx) ## extract R
  f <- t(Q)%*%y
  f <- f[1:ncol(X),]
  beta <- solve(R)%*%f
  residual <- y-X%*%beta
  sigma <- as.vector(t(residual)%*%residual/(nrow(X)-ncol(X)))
  variance <- solve(R)%*%t(solve(R))*sigma
  vrr <- solve(t(X)%*%X)
  dia <- as.matrix(diag(vrr))
  pvalue <- 2*pt(-abs(beta)/sqrt((sigma*dia)),df=nrow(X)-ncol(X))
  list(coefficient=beta,std_error=sqrt(as.matrix(diag(variance),ncol=ncol(X))),
       pvalue=pvalue,residual_variance=sigma)
}
newM2 <- model.matrix(dist ~speed + I(speed^2),cars)
lr_3(newM2,cars$dist)
```

```
## $coefficient
##                  [,1]
## (Intercept) 2.4701378
```

```
## speed       0.9132876
## I(speed^2)  0.0999593
##
## $std_error
##                    [,1]
## (Intercept) 14.81716473
## speed        2.03422044
## I(speed^2)   0.06596821
##
## $pvalue
##                  [,1]
## (Intercept) 0.8683151
## speed       0.6555224
## I(speed^2)  0.1364024
##
## $residual_variance
## [1] 230.3131
```

Therefore, the function gives the right result for the model.