HW2

Problem 4

From the MASS package load the Boston dataset. (Remember that the MASS package has a select command just like the tidyverse, so if you are going to use the tidyverse select command you need to load tidyverse last.)

a. Start by building a model for the median household value (medv) as explained by everything else in the dataset. Save this model so you can compare the coefficients in future steps of this problem.

```
model<-lm(medv~.,data=Boston)
summary(model)
```

```
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas         2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## black        9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

b. Explicitly build the normal equations and use them to find the coefficients for the linear model. Compare to what you found in part 1. Be sure to include your code in your writeup.

```
X<-matrix(nrow=nrow(Boston),ncol=ncol(Boston))

y<-Boston$medv
X[,1]<-rep(1)
for(j in 1:13){
  X[,j+1]<-Boston[,j]
```

```
}
A<-t(X)%*%X
b<-t(X)%*%y
beta<-solve(A,b)
beta
```

```
##                [,1]
##  [1,]  3.645949e+01
##  [2,] -1.080114e-01
##  [3,]  4.642046e-02
##  [4,]  2.055863e-02
##  [5,]  2.686734e+00
##  [6,] -1.776661e+01
##  [7,]  3.809865e+00
##  [8,]  6.922246e-04
##  [9,] -1.475567e+00
## [10,]  3.060495e-01
## [11,] -1.233459e-02
## [12,] -9.527472e-01
## [13,]  9.311683e-03
## [14,] -5.247584e-01
```

```
norm(beta-model$coefficients,"f")
```

```
## [1] 1.244974e-11
```

c. Find the eigenvalues of the matrix XTX and compute the ratio between the largest and the smallest eigenvalues. This ratio is called the condition number of the matrix XTX (technically the condition number is the square root of this ratio, but we don't need the root here). The condition number in this case should be huge! What does this number mean about the matrix XTX? (Hint: think about what it means geometrically for the ratio of largest to smallest eigenvalues to be very very large)

```
evalA<-eigen(A)
condA<-max(evalA$values)/min(evalA$values)
condA
```

```
## [1] 228418414
```

d. Now pre-process the predictors by doing a z transformation on each. Recall that

$$z_j = \frac{x_j - \bar{x_j}}{\sigma_{x_j}}$$

e. Perform this pre-processing on the data and build the normal equations from this pre-processed data. What does the condition number tell you?

```
Z<-X
for(j in 1:13){
  Z[,j+1]<-scale(Boston[,j])
}
B<-t(Z)%*%Z
evalB<-eigen(B)
condB<-max(evalB$values)/min(evalB$values)
condB
```

```
## [1] 96.47174
```