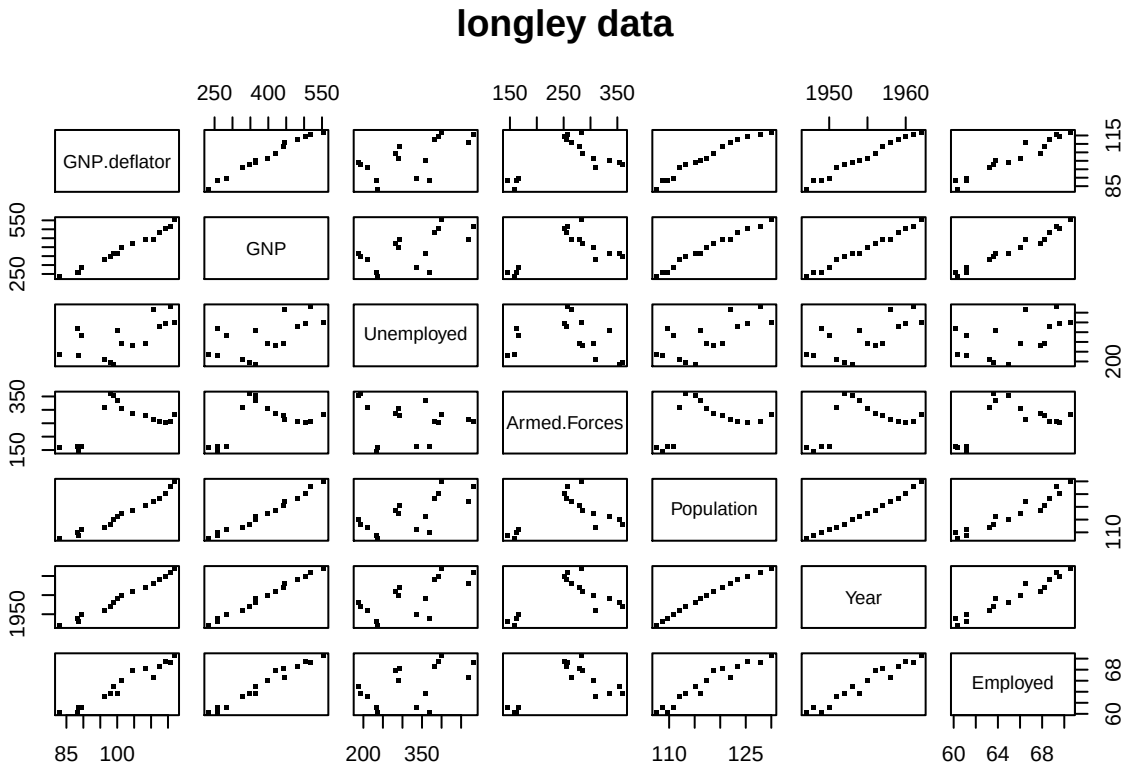
HW 7 PSTAT127

Marissa Santiago

05/29/2022

Problem 1

```
require(stats); require(graphics)
pairs( longley, main = "longley data", cex=0.5, pch=15)
```




```
#### ========================================
### fitting using lm
fit.lm1 = lm(Employed ~ ., data = longley)
summary(fit.lm1)
```

```
##
```

```
## Call:
## lm(formula = Employed ~ ., data = longley)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41011 -0.15767 -0.02816  0.10155  0.45539
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.482e+03  8.904e+02  -3.911 0.003560 **
## GNP.deflator  1.506e-02  8.492e-02   0.177 0.863141
## GNP          -3.582e-02  3.349e-02  -1.070 0.312681
## Unemployed   -2.020e-02  4.884e-03  -4.136 0.002535 **
## Armed.Forces -1.033e-02  2.143e-03  -4.822 0.000944 ***
## Population   -5.110e-02  2.261e-01  -0.226 0.826212
## Year          1.829e+00  4.555e-01   4.016 0.003037 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3049 on 9 degrees of freedom
## Multiple R-squared:  0.9955, Adjusted R-squared:  0.9925
## F-statistic: 330.3 on 6 and 9 DF,  p-value: 4.984e-10
```

1a)

The model: $Employed = \beta_0 + \beta_1 GNP.deflator_i + \beta_2 GNP_i + \beta_3 Unemployed_i + \beta_4 Armed.Forces_i + \beta_5 Population_i + \beta_6 Year_i + \epsilon_i$

1b)

Based on the pairs plot, I do think there is a concern for multicolinearity. In the plot, there are several graphs that show a linear positive trend between two specific variables showing high correlation. Using R, we see that GNP is highly correlated to variables GNP.deflator, Population, and Year.

```
cor(longley$GNP,longley$GNP.deflator)
```

```
## [1] 0.9915892
```

```
cor(longley$GNP,longley$Population)
```

```
## [1] 0.9910901
```

```
cor(longley$GNP,longley$Year)
```

```
## [1] 0.9952735
```

1c)

To find the point estimate and its 95% confidence interval we must use function: $CI = Estimate \pm Std.Error \times 1.96$

```r
cat("The Estimate is",1.829e+00,"\n")
```

```
## The Estimate is 1.829
```

```r
cat("The interval is",c(1.829 - 1.96*4.555e-01,1.829 + 1.96*4.555e-01),"\n")
```

```
## The interval is 0.93622 2.72178
```

1d) HELP

```r
### ridge regression
library(MASS)
fit.ridge0 = lm.ridge(Employed ~ ., data=longley, lambda=0 )
fit.ridge1 = lm.ridge(Employed ~ ., data=longley, lambda=0.02 )
```

```r
coef(fit.ridge0)
```

```
##                GNP.deflator           GNP    Unemployed  Armed.Forces
## -3.482259e+03  1.506187e-02 -3.581918e-02 -2.020230e-02 -1.033227e-02
##    Population          Year
## -5.110411e-02  1.829151e+00
```

```r
coef(fit.ridge1)
```

```
##                GNP.deflator           GNP    Unemployed  Armed.Forces
## -1.877437e+03  9.346751e-03  5.190160e-03 -1.376159e-02 -8.127275e-03
##    Population          Year
## -1.288085e-01  1.003545e+00
```

```r
-8.127275e-03
```

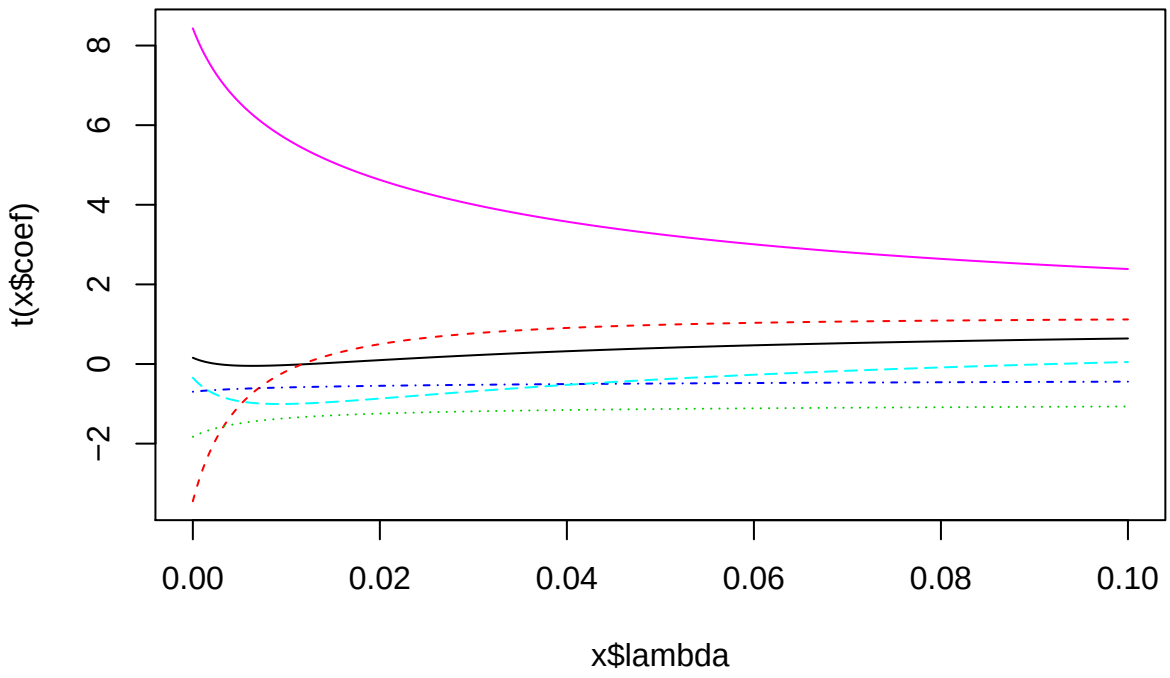
```
## [1] -0.008127275
```

The two theoretical ridge regression models are the same as their coefficents are only slightly different.

1e)

```r
# fit the ridge regression model
fit.ridge2 = lm.ridge(Employed ~ ., data=longley, lambda=seq(0, 0.1, 0.0001))

# plot the graph ridge coefficients vs lambda
plot(fit.ridge2)
```




```
#show first few rows
head( coef(fit.ridge2 )) ### Note that first row matches OLS estimates from fit.lm1
```

```
##                  GNP.deflator         GNP  Unemployed Armed.Forces  Population
## 0.0000 -3482.259   0.01506187 -0.03581918 -0.02020230  -0.01033227 -0.05110411
## 0.0001 -3456.458   0.01409837 -0.03490603 -0.02006969  -0.01029707 -0.05499472
## 0.0002 -3431.364   0.01317829 -0.03402282 -0.01994129  -0.01026284 -0.05872765
## 0.0003 -3406.945   0.01229950 -0.03316814 -0.01981689  -0.01022954 -0.06231062
## 0.0004 -3383.169   0.01145997 -0.03234064 -0.01969631  -0.01019711 -0.06575085
## 0.0005 -3360.009   0.01065781 -0.03153907 -0.01957938  -0.01016553 -0.06905511
##            Year
## 0.0000 1.829151
## 0.0001 1.816027
## 0.0002 1.803259
## 0.0003 1.790832
## 0.0004 1.778730
## 0.0005 1.766938
```

```
#choose ridge parameter given the vector lambda is a sequence from 0 (OLS) to .1
select( lm.ridge(Employed ~ ., data=longley, lambda=seq(0, 0.1, 0.0001)) )
```

```
## modified HKB estimator is 0.004275357
## modified L-W estimator is 0.03229531
## smallest value of GCV  at 0.0028
```

```r
## Now fit the ridge regression model with a much larger lambda value,
fit.ridge3 = lm.ridge(Employed ~ ., data=longley, lambda= 10e8)

#print the coefficent values for the new model
coef(fit.ridge3)
```

```
##              GNP.deflator          GNP   Unemployed Armed.Forces   Population
## 6.531698e+01 5.055457e-09 5.560367e-10 3.021637e-10 3.692493e-10 7.758049e-09
##         Year
## 1.146419e-08
```

```r
### Return to the results from fit.ridge2. Run the following code and explain:

#finds the index of minimum of GCV
which.min( fit.ridge2$GCV )
```

```
## 0.0028
##     29
```

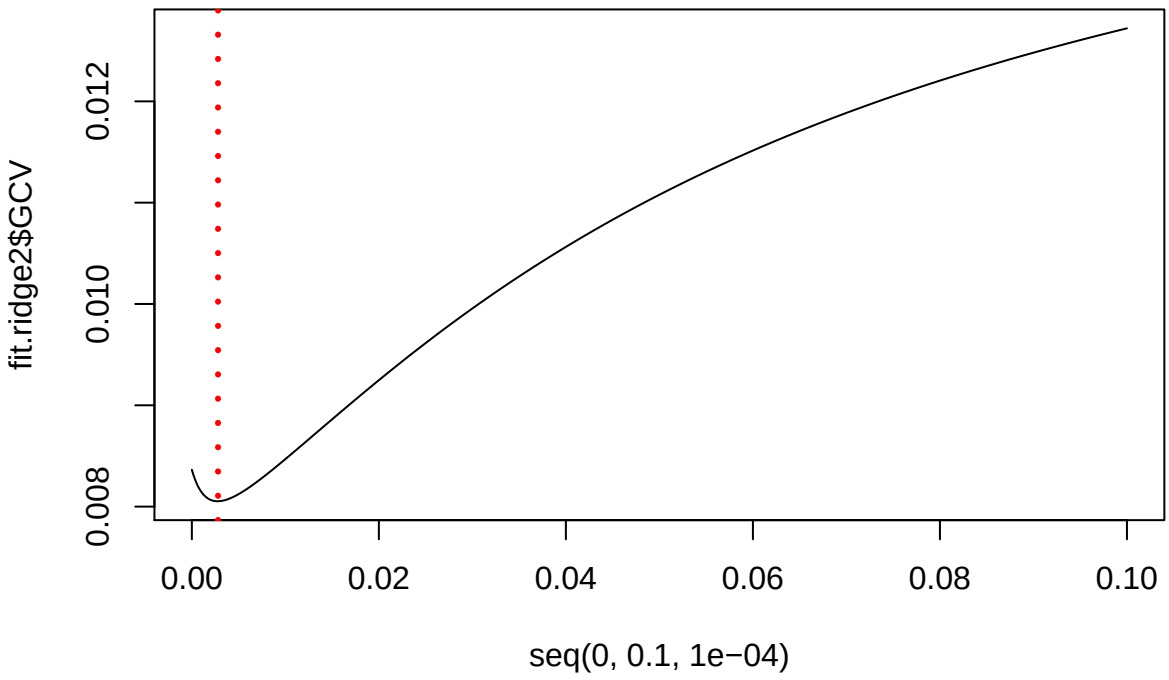
```r
#turns the index into a number
as.numeric( which.min( fit.ridge2$GCV ) )
```

```
## [1] 29
```

```r
#gets the value of  the minimum lambda
best.lambda.byGCV = as.numeric( names( which.min( fit.ridge2$GCV )))

#plots the line of lambda vs x from 0 to 0.1
plot( y=fit.ridge2$GCV, x=seq(0, 0.1, 0.0001), type="l" )

#plots the place of best lambda
abline( v= best.lambda.byGCV, col="red", lwd=3, lty=3 )
```



Problem 2

```
library(datasets)
library(glmnet)
```

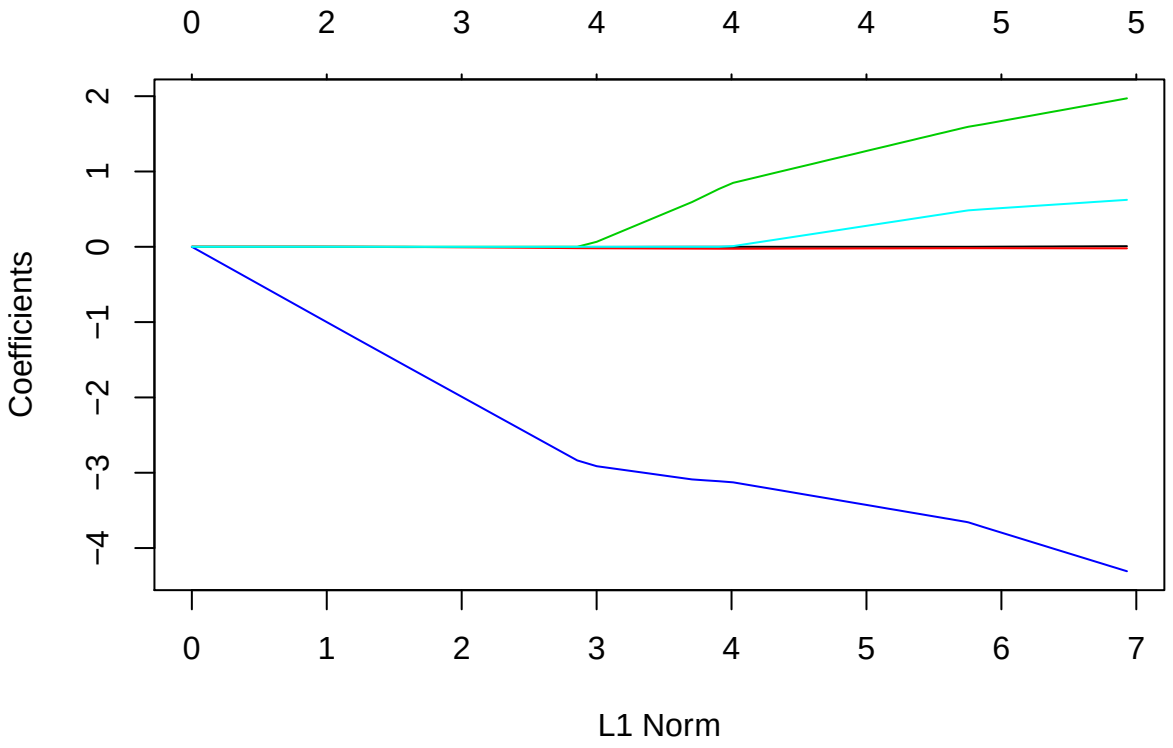
```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-1
```

```
#creates a matrix of our values since glmnet cannot take in a dataset
carsCols = as.matrix(mtcars[,c(3,4,5,6,7)])

#fit a lasso regression model on our matrix
fit.glmnet = glmnet( x= carsCols, y= mtcars[,1], alpha=1)

#plots the lasso regression model
plot(fit.glmnet)
```




```
#perform k-fold cross-validation to find optimal lambda value
fit.glmnet.5foldCV = cv.glmnet( x= carsCols, y= mtcars[,1], alpha=1, nfolds=5 )

#find optimal lambda value that minimizes test MS
coef( fit.glmnet.5foldCV, s= "lambda.min" )
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept) 30.721042817
## disp         .
## hp          -0.025846707
## drat         0.849281835
## wt          -3.127504027
## qsec         0.009436974
```

This code runs a lasso regression on linear model: $Y_i = \beta_o + \beta_1 hp_i + \beta_2 drat_i + \beta_3 wt_i + \beta_4 qsec_i + \epsilon_i$ from the mtcars dataset. This is done with R command "glmnet" which runs a generalized linear model via a penalized maxima likelihood to shrink λ. We know it is a lasso regression specifically because in "glmnet" alpha must always equal to 1. Also, the glmnet function does not work with dataframes, that is why we need to create a numeric matrix for the training features and a vector of target values which is given the name 'carsCols'. In the plotted graph of our lasso regression model we see that each different colored line represents the value taken by a different coefficient in the model. L1 norm in the x-axis is the regularization term for LASSO. As regularization decreases (aka lambda decreases) there is less restraint on the coefficent and we see the values within the model increase as well.

We then run a 5-fold cross validation on the same ridge regression model to find our best lambda for each given coefficent. The best lambda value produces the lowest MSE and controls the amount of shrinkage that

is done on each coefficent. The last line of code helps us deltermine the best lambda value for each cocefficent value. In the end we found no coefficent for 'disp' meaning the lasso regression shrunk the coefficent to zero and therefore dropped from the model due to it being insignificant if equal to zero.