Gas Flaring Regression Code

```
library(tidyverse) #loading necessary packages
library(ggplot2)
library(rlist)
library(ISLR2)
library(glmnet)
library(readxl)
require(methods)
library(caret)
library(knitr)
```

```
#loading in Africa and Middle East Data
flaringdata2 <- read_excel("C:/Users/nandi/OneDrive - UW/FinalProject484/MiddleEast_Africa_Filtered.xlsx
#ensuring proper years range as prices start from 1984 and some data is unaviable for 2021
flaringdata2 <- flaringdata2%>%filter(Year<2021, Year>1983)
 #loading in prices data
PricesData <- read_excel("C:/Users/nandi/OneDrive - UW/FinalProject484/PricesData.xlsx")
```

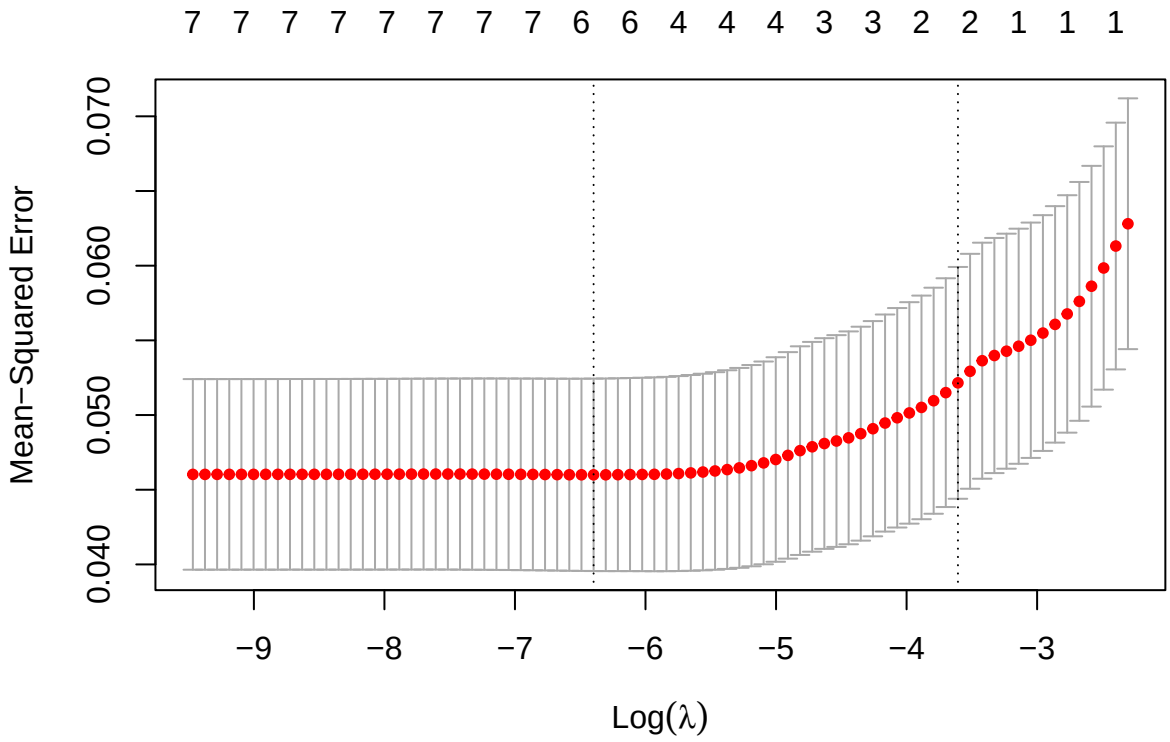
```
oil_prices <- list() #for loop to add prices to the flaringdata2 dataset

for(i in 1:nrow(flaringdata2)) {
  for(j in 1:nrow(PricesData)) {
    if (flaringdata2[i, 2] == PricesData[j, 1]) {
      oil_prices <- append(oil_prices, (PricesData[j, 2]))
    }
  }
}

gas_prices <- list()
for(i in 1:nrow(flaringdata2)) {
  for(j in 1:nrow(PricesData)) {
    if (flaringdata2[i, 2] == PricesData[j, 1]) {
      gas_prices <- append(gas_prices, (PricesData[j, 3]))
    }
  }
}

flaringdata2$oil_price <- oil_prices
flaringdata2$gas_price <- gas_prices

#ensuring column is counted as numeric
flaringdata2$oil_price <- as.numeric(as.character(flaringdata2$oil_price))
flaringdata2$gas_price <- as.numeric(as.character(flaringdata2$gas_price))
```

```r
#creating matrix for explanatory variables and
#removing unnecessary non-numeric identifiers and NA values such as
#country name, year, region, and whether the country is an OPEC member
x <- (flaringdata2[, -c(1, 2, 3, 4, 6, 11, 12)])
#creating dependent variable vector, gas flared
y <- flaringdata2$gasflared_bcm


#normalizing between 0 and 1 as the variables have very difference ranges
#as they are in different units
process <- preProcess(as.data.frame(x), method=c("range"))
x_scale <- predict(process, as.data.frame(x))


#normalizing the y matrix and turning it into a vector
process <- preProcess(as.data.frame(y), method=c("range"))
y_scale <- predict(process, as.data.frame(y))
y_vector <- y_scale$y


#creating a model matrix for x for LASSO regression
x_mat <- model.matrix(~ ., x_scale) [, -1]
```

```r
#setting the seed
set.seed (1)
#splitting the data into training and test
train <- sample (1: nrow(x_mat), nrow(x_mat) / 1.25)
test <- (-train)
y_vector.test <- y_vector[test]
```

```r
#fitting LASSO regression
flare_lasso <- glmnet(x_mat[train, ], y_vector[train], alpha = 1)
#plot(flare_lasso) (for reference)
```

```r
 #picking best lambda for LASSO by minimizing the loss function
set.seed (1)
cv.out <- cv.glmnet(x_mat[train , ], y_vector[train], alpha = 1)
plot(cv.out)
```




```
bestlam <- cv.out$lambda.min
lasso.pred <- predict(flare_lasso , s = bestlam ,
newx = x_mat[test , ])
#checking mean error
mean (( lasso.pred - y_vector.test)^2)
```

```
## [1] 0.02940755
```

```
#checking coefficients on LASSO regression
out <- glmnet (x_mat, y_vector, alpha = 1)
lasso.coef <- predict (out , type = "coefficients",
s = bestlam)[1:8, ]
lasso.coef
```

```
##     (Intercept)      refcap_kbd     oilprod_kbd gasreserves_tcm     gasprod_bcm
##      0.16481759     -0.41271304      0.25022227      0.85071290      0.13198561
##     gascons_bcm       oil_price       gas_price
##     -0.51320830      0.00000000      0.03558375
```

```
#printing coefficients
print.data.frame(lasso.coef)
```

```
## [1] (Intercept)     refcap_kbd      oilprod_kbd     gasreserves_tcm
## [5] gasprod_bcm     gascons_bcm     oil_price       gas_price
## <0 rows> (or 0-length row.names)
```

```
#creating a table of the coefficients
kable(lasso.coef)
```

	x
(Intercept)	0.1648176
refcap_kbd	-0.4127130
oilprod_kbd	0.2502223
gasreserves_tcm	0.8507129
gasprod_bcm	0.1319856
gascons_bcm	-0.5132083
oil_price	0.0000000
gas_price	0.0355837