HW5 - Basic Linear Regression

Question 8.1 Describe a situation or problem from your job, everyday life, current events, etc., for which a linear regression model would be appropriate. List some (up to 5) predictors that you might use.

Linear regression model could be used to predict the death number per 1 million people due to covid-19. (The response) Using linear model could effectively define the correlation between covid-19 death tolls with different related variables. In the multuole linear regression model, variables such as the number of new test per 1 Million people, number of confimed cases per 1 million people etc. Since Linear regression model is a predictive model so that it would a perfect fit to analyze and predict the death rate. Predictors including: 1. Number of new covid 19 test conducted per 1 million people

2. Number of hospitals per 1 million people
3. Number of confimed cases per 1 million people
4. Number of super spreeders(Refers to someone who is capable of spreeding the virus to at least 25 people)
5. Number of average recovery days
6. Number of preventive masks sold

Question 8.2

```
# Set seed for reproducibility and import the library
set.seed(1)
library(GGally)
```

```
## Loading required package: ggplot2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(car)
```

```
## Loading required package: carData
```

```
library(scales)
library(Metrics)

#Import the data- uscrime.txt
uscrime<- read.table('uscrime.txt',header=T)
```

Firstly, we should take a quick look and get an idea about this dataset and to deceide if scaling is necessary and makes an impact on this dataset when inserting it into the lm function. (Scaling vs Non-scaling)

#step 1. Experiment lm function with Scaled vs unscaled dataset.

```
summary(uscrime)
```

```
##        CM              So               Ed             Pol
##  Min.   :11.90   Min.   :0.0000   Min.   : 8.70   Min.   : 4.50
##  1st Qu.:13.00   1st Qu.:0.0000   1st Qu.: 9.75   1st Qu.: 6.25
##  Median :13.60   Median :0.0000   Median :10.80   Median : 7.80
##  Mean   :13.86   Mean   :0.3404   Mean   :10.56   Mean   : 8.50
##  3rd Qu.:14.60   3rd Qu.:1.0000   3rd Qu.:11.45   3rd Qu.:10.45
##  Max.   :17.70   Max.   :1.0000   Max.   :12.20   Max.   :16.60
##       Po2               LF              M.F              Pop
##  Min.   : 4.100   Min.   :0.4800   Min.   : 93.40   Min.   :  3.00
##  1st Qu.: 5.850   1st Qu.:0.5305   1st Qu.: 96.45   1st Qu.: 10.00
##  Median : 7.300   Median :0.5600   Median : 97.70   Median : 25.00
##  Mean   : 8.023   Mean   :0.5612   Mean   : 98.30   Mean   : 36.62
##  3rd Qu.: 9.700   3rd Qu.:0.5930   3rd Qu.: 99.20   3rd Qu.: 41.50
##  Max.   :15.700   Max.   :0.6410   Max.   :107.10   Max.   :168.00
##        NW              U1                U2            Wealth
##  Min.   : 0.20   Min.   :0.07000   Min.   :2.000   Min.   :2880
##  1st Qu.: 2.40   1st Qu.:0.08050   1st Qu.:2.750   1st Qu.:4595
##  Median : 7.60   Median :0.09200   Median :3.400   Median :5370
##  Mean   :10.11   Mean   :0.09547   Mean   :3.398   Mean   :5254
##  3rd Qu.:13.25   3rd Qu.:0.10400   3rd Qu.:3.850   3rd Qu.:5915
##  Max.   :42.30   Max.   :0.14200   Max.   :5.800   Max.   :6890
##       Ineq            Prob              Time           Crime
##  Min.   :12.60   Min.   :0.00690   Min.   :12.20   Min.   : 342.0
##  1st Qu.:16.55   1st Qu.:0.03270   1st Qu.:21.60   1st Qu.: 658.5
##  Median :17.60   Median :0.04210   Median :25.80   Median : 831.0
##  Mean   :19.40   Mean   :0.04709   Mean   :26.60   Mean   : 905.1
##  3rd Qu.:22.75   3rd Qu.:0.05445   3rd Qu.:30.45   3rd Qu.:1057.5
##  Max.   :27.60   Max.   :0.11980   Max.   :44.00   Max.   :1993.0
```

```
head(uscrime)
```

```
##     CM So   Ed  Pol  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.015801
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.041399
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.034201
##      Time Crime
## 1 26.2011   791
## 2 25.2999  1635
## 3 24.3006   578
## 4 29.9012  1969
## 5 21.2998  1234
## 6 20.9995   682
```

```
#scaling the data
uscrime_sc <-scale(uscrime)
uscrime_sc <- as.data.frame(uscrime_sc)
head(uscrime_sc)
```

```
##           CM         So         Ed        Pol        Po2         LF         M.F
## 1  0.9886930  1.3770536 -1.3085099 -0.9085105 -0.8666988 -1.2667456 -1.12060499
## 2  0.3521372 -0.7107373  0.6580587  0.6056737  0.5280852  0.5396568  0.98341752
## 3  0.2725678  1.3770536 -1.4872888 -1.3459415 -1.2958632 -0.6976051 -0.47582390
## 4 -0.2048491 -0.7107373  1.3731746  2.1535064  2.1732150  0.3911854  0.37257228
## 5  0.1929983 -0.7107373  1.3731746  0.8075649  0.7426673  0.7376187  0.06714965
## 6 -1.3983912 -0.7107373  0.3898903  1.1104017  1.2433590 -0.3511718 -0.64550313
##           Pop           NW          U1         U2     Wealth       Ineq
## 1 -0.09500679  1.943738564  0.69510600  0.8313680 -1.3616094  1.6793638
## 2 -0.62033844  0.008483424  0.02950365  0.2393332  0.3276683  0.0000000
## 3 -0.48900552  1.146296747 -0.08143007 -0.1158877 -2.1492481  1.4036474
## 4  3.16204944 -0.205464381  0.36230482  0.5945541  1.5298536 -0.6767585
## 5 -0.48900552 -0.691709391 -0.24783066 -1.6551781  0.5453053 -0.5013026
## 6 -0.30513945 -0.555560788 -0.63609870 -0.5895155  1.6956723 -1.7044289
##         Prob        Time      Crime
## 1  1.6497631 -0.05599367 -0.2949744
## 2 -0.7693365 -0.18315796  1.8872422
## 3  1.5969416 -0.32416470 -0.8456997
## 4 -1.3761895  0.46611085  2.7508209
## 5 -0.2503580 -0.74759413  0.8504308
## 6 -0.5669349 -0.78996812 -0.5768010
```

Use lm function with the scaled data

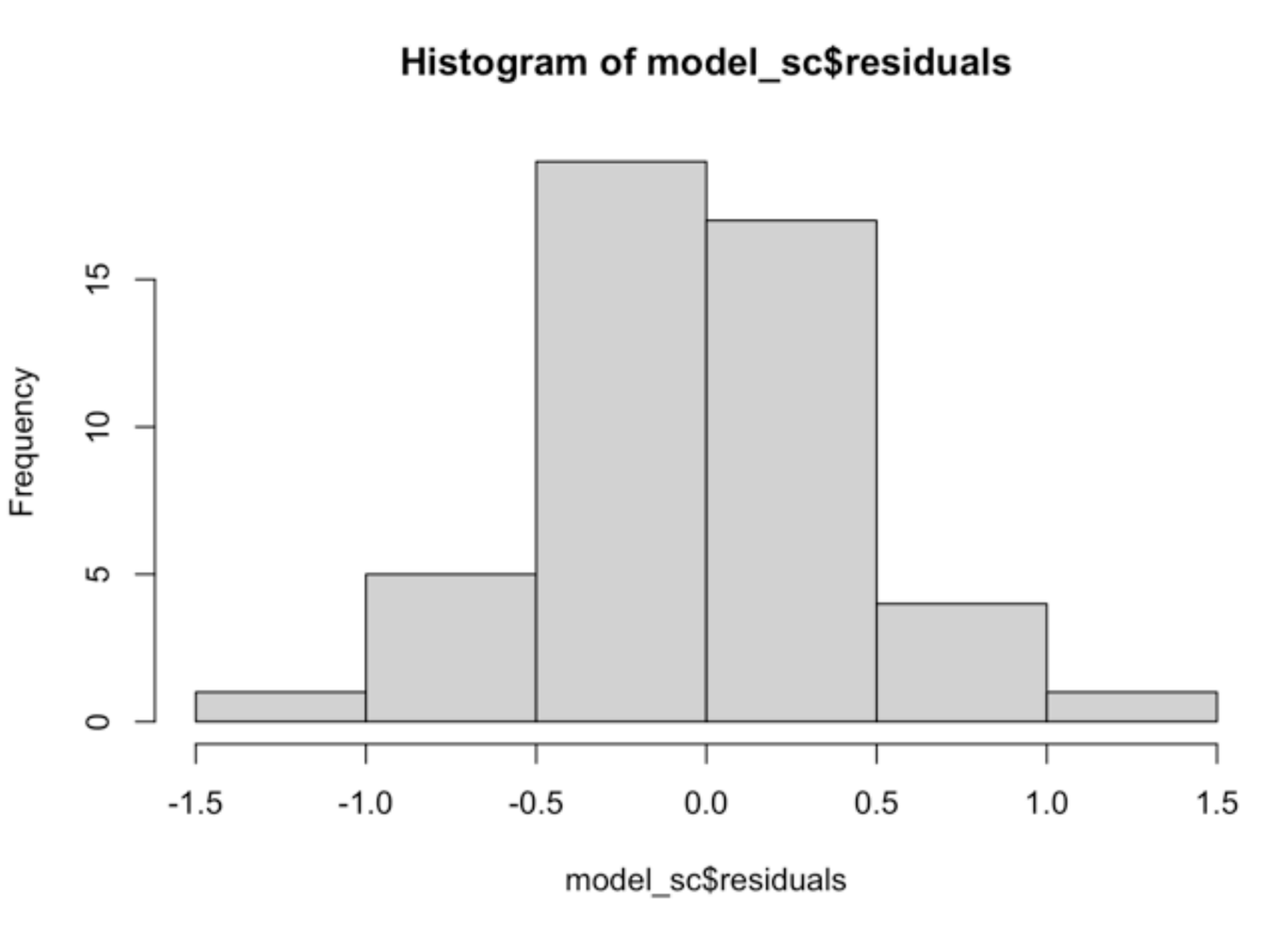
```
model_sc<-lm(Crime~.,uscrime_sc)
summary(model_sc)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = uscrime_sc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.02321 -0.25361 -0.01731  0.29214  1.32554
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.324e-16  7.885e-02   0.000  1.00000
## CM           2.854e-01  1.355e-01   2.106  0.04344 *
## So          -4.710e-03  1.842e-01  -0.026  0.97977
## Ed           5.447e-01  1.796e-01   3.033  0.00486 **
## Pol          1.482e+00  8.154e-01   1.817  0.07889 .
## Po2         -7.911e-01  8.493e-01  -0.931  0.35883
## LF          -6.936e-02  1.536e-01  -0.452  0.65465
## M.F          1.326e-01  1.551e-01   0.855  0.39900
## Pop         -7.215e-02  1.269e-01  -0.568  0.57385
## NW           1.118e-01  1.723e-01   0.649  0.52128
## U1          -2.716e-01  1.963e-01  -1.384  0.17624
## U2           3.664e-01  1.798e-01   2.038  0.05016 .
## Wealth       2.399e-01  2.586e-01   0.928  0.36075
## Ineq         7.290e-01  2.343e-01   3.111  0.00398 **
## Prob        -2.854e-01  1.336e-01  -2.137  0.04063 *
## Time        -6.375e-02  1.313e-01  -0.486  0.63071
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5405 on 31 degrees of freedom
## Multiple R-squared:  0.8031, Adjusted R-squared:  0.7078
## F-statistic: 8.429 on 15 and 31 DF,  p-value: 3.539e-07
```

From the output above, the linearesult comes up with p-value: 3.54e-07 ; Multiple R-squared: 0.803, Adjusted R-squared: 0.708.

```
#Check to see if the error(residual) is normally distributed.

hist(model_sc$residuals)
```




```
#From the graph, we can tell that the histogram is very close to normall distribute.so the scaled data is fulfill
ing the prerequisite of the linear regression that error needs to be normlly distributed
```

Then, we experiment with the unscaled data, as we took the orignal dataset.

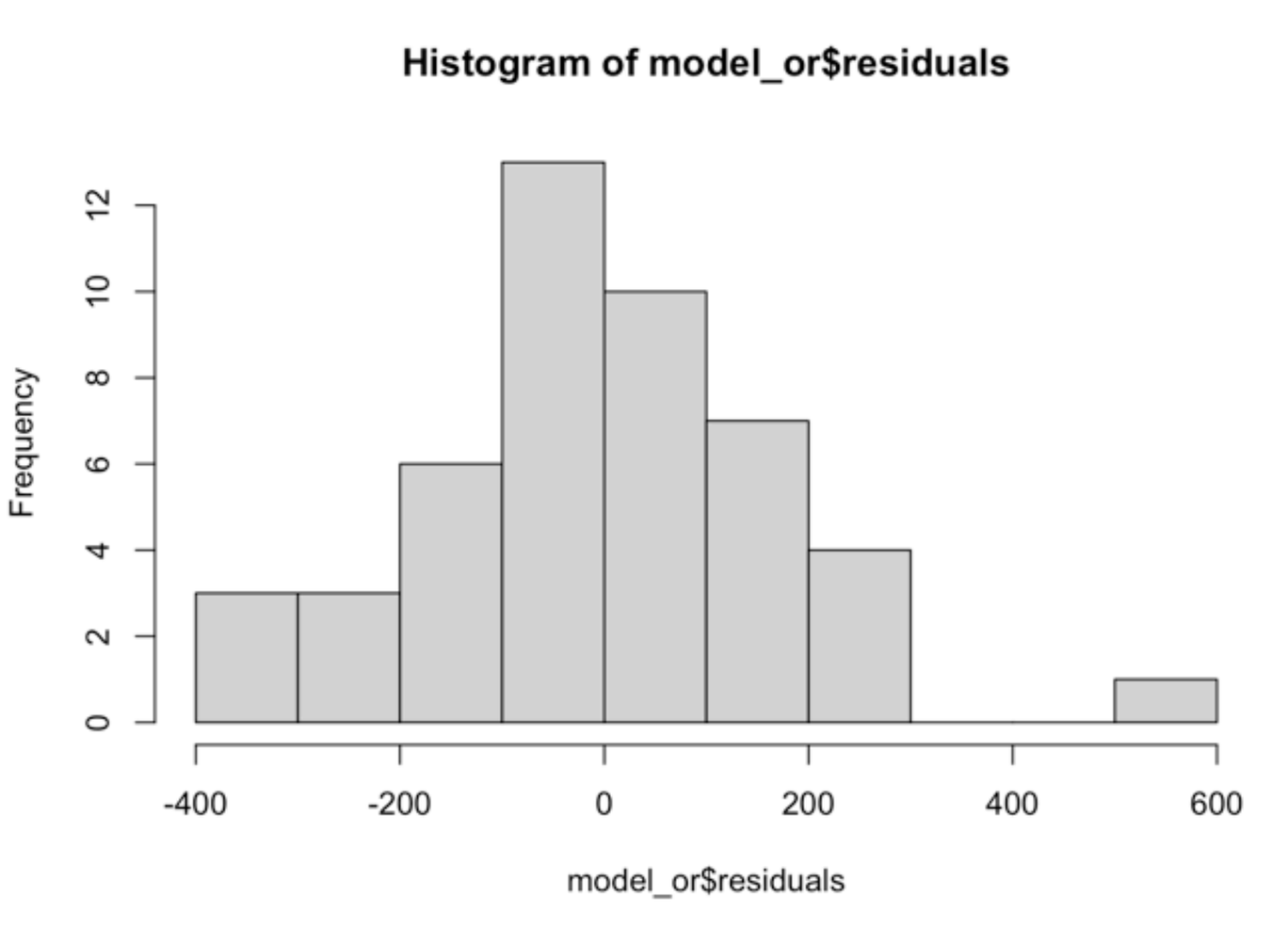
```
model_or<-lm(Crime~.,uscrime)
summary(model_or)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = uscrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -395.74  -98.09   -6.69  112.99  512.67
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
## CM           8.783e+01  4.171e+01   2.106 0.043443 *
## So          -3.803e+00  1.488e+02  -0.026 0.979765
## Ed           1.883e+02  6.209e+01   3.033 0.004861 **
## Pol          1.928e+02  1.061e+02   1.817 0.078892 .
## Po2         -1.094e+02  1.175e+02  -0.931 0.358830
## LF          -6.638e+02  1.470e+03  -0.452 0.654654
## M.F          1.741e+01  2.035e+01   0.855 0.398995
## Pop         -7.330e-01  1.290e+00  -0.568 0.573845
## NW           4.204e+00  6.481e+00   0.649 0.521279
## U1          -5.827e+03  4.210e+03  -1.384 0.176238
## U2           1.678e+02  8.234e+01   2.038 0.050161 .
## Wealth       9.617e-02  1.037e-01   0.928 0.360754
## Ineq         7.067e+01  2.272e+01   3.111 0.003983 **
## Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *
## Time        -3.479e+00  7.165e+00  -0.486 0.630708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 209.1 on 31 degrees of freedom
## Multiple R-squared:  0.8031, Adjusted R-squared:  0.7078
## F-statistic: 8.429 on 15 and 31 DF,  p-value: 3.539e-07
```

Several important outputs from unscaled dataset remain the same as the scaled dataset. Multiple R-squared: 0.803, Adjusted R-squared: 0.708 p-value: 3.54e-07

```
#Also,we take a look at the distribution of the error generated from unscaled dataset

hist(model_or$residuals)
```




```
# Although it's slightly skewed comparing with the histogram from scaled dataset, we still can consider this bein
g normally distributed.
```

After thorough comparison, i found there is not much differences between the models generated from scaled and unscaled dataset. Given that, we will stick with the original dataset in order to keep the nature of the dataset.

#step 2. Traing and testing dataset Then, the second thing we need to do with this data is to seperate it into training and testing dataset.As the rule of thumb, we split it by 80(training)/20(testing)

```
index<-sample(nrow(uscrime)*0.8)
index
```

```
##  [1]  4  1 34 23 14 18 27 19 36 21 28 10 22 33 26  7  9 15  5 31 24 30 16  2 35
## [26] 32 29 37  3  6 17 25 13 20  8 12 11
```

```
train_crime<-uscrime[index,]
test_crime<-uscrime[-index,]
```

#step 3. Investigate if the collinearity error exists in this lm model.

```
cor(uscrime) #examine the correlation coef of each variable
```

```
##                 CM          So          Ed         Pol         Po2         LF
## CM      1.00000000  0.58435534 -0.53023964 -0.50573690 -0.51317336 -0.1609488
## So      0.58435534  1.00000000 -0.70274132 -0.37263633 -0.37616753 -0.5054695
## Ed     -0.53023964 -0.70274132  1.00000000  0.48295213  0.49940958  0.5611780
## Pol    -0.50573690 -0.37263633  0.48295213  1.00000000  0.99358648  0.1214932
## Po2    -0.51317336 -0.37616753  0.49940958  0.99358648  1.00000000  0.1063496
## LF     -0.16094882 -0.50546948  0.56117795  0.12149320  0.10634960  1.0000000
## M.F    -0.02867993 -0.31473291  0.43691492  0.03376027  0.02284250  0.5135588
## Pop    -0.28063762 -0.04991832 -0.01722740  0.52628358  0.51378940 -0.1236722
## NW      0.59319826  0.76710262 -0.66488190 -0.21370878 -0.21876821 -0.3412144
## U1     -0.22438060 -0.17241931  0.01810345 -0.04369761 -0.05171199 -0.2293997
## U2     -0.24484339  0.07169289 -0.21568155  0.18509304  0.16922422 -0.4207625
## Wealth -0.67005506 -0.63694543  0.73599704  0.78722528  0.79426205  0.2946323
## Ineq    0.63921138  0.73718106 -0.76865789 -0.63050025 -0.64815183 -0.2698865
## Prob    0.36111641  0.53086199 -0.38992286 -0.47324704 -0.47302729 -0.2500861
## Time    0.11451072  0.06681283 -0.25397355  0.10335774  0.07562665 -0.1236404
## Crime  -0.08947240 -0.09063696  0.32283487  0.68760446  0.66671414  0.1888663
##                M.F         Pop          NW           U1          U2
## CM     -0.02867993 -0.28063762  0.59319826 -0.224380599 -0.24484339
## So     -0.31473291 -0.04991832  0.76710262 -0.172419305  0.07169289
## Ed      0.43691492 -0.01722740 -0.66488190  0.018103454 -0.21568155
## Pol     0.03376027  0.52628358 -0.21370878 -0.043697608  0.18509304
## Po2     0.02284250  0.51378940 -0.21876821 -0.051711989  0.16922422
## LF      0.51355879 -0.12367222 -0.34121444 -0.229399684 -0.42076249
## M.F     1.00000000 -0.41062750 -0.32730454  0.351891900 -0.01869169
## Pop    -0.41062750  1.00000000  0.09515301 -0.038119948  0.27042159
## NW     -0.32730454  0.09515301  1.00000000 -0.156450020  0.08090829
## U1      0.35189190 -0.03811995 -0.15645002  1.000000000  0.74592482
## U2     -0.01869169  0.27042159  0.08090829  0.745924815  1.00000000
## Wealth  0.17960864  0.30826271 -0.59010707  0.044857202  0.09207166
## Ineq   -0.16708869 -0.12629357  0.67731286 -0.063832178  0.01567818
## Prob   -0.05085826 -0.34728906  0.42805915 -0.007469032 -0.06159247
## Time   -0.42769738  0.46421046  0.23039841 -0.169852838  0.10135833
## Crime   0.21391426  0.33747406  0.03259884 -0.050477918  0.17732065
##               Wealth        Ineq         Prob          Time       Crime
## CM     -0.6700550558  0.63921138  0.361116408  0.1145107190 -0.08947240
## So     -0.6369454328  0.73718106  0.530861993  0.0668128312 -0.09063696
## Ed      0.7359970363 -0.76865789 -0.389922862 -0.2539735471  0.32283487
## Pol     0.7872252807 -0.63050025 -0.473247036  0.1033577449  0.68760446
## Po2     0.7942620503 -0.64815183 -0.473027293  0.0756266536  0.66671414
## LF      0.2946323090 -0.26988646 -0.250086098 -0.1236404364  0.18886635
## M.F     0.1796086363 -0.16708869 -0.050858258 -0.4276973791  0.21391426
## Pop     0.3082627091 -0.12629357 -0.347289063  0.4642104596  0.33747406
## NW     -0.5901070652  0.67731286  0.428059153  0.2303984071  0.03259884
## U1      0.0448572017 -0.06383218 -0.007469032 -0.1698528383 -0.05047792
## U2      0.0920716601  0.01567818 -0.061592474  0.1013583270  0.17732065
## Wealth  1.0000000000 -0.88399728 -0.555334708  0.0006485587  0.44131995
## Ineq   -0.8839972758  1.00000000  0.465321920  0.1018228182 -0.17902373
## Prob   -0.5553347075  0.46532192  1.000000000 -0.4362462614 -0.42742219
## Time    0.0006485587  0.10182282 -0.436246261  1.0000000000  0.14986606
## Crime   0.4413199490 -0.17902373 -0.427422188  0.1498660617  1.00000000
```

```
#As a rule of thumb, any variables with correlation coefficients above 0.7 should be considered as strongly corre
lated. From the below table, these variables surfaced under the this threshold.
```

Strongly corelated predictors:

'So' with 'Ed','NW','Ineq' 'Ed' with 'Wealth','Ineq' 'Po1' with 'Po2', 'Wealth' 'Po2' with 'Wealth' 'U1' with 'U2' 'Wealth' with 'Ineq'

```
model_train<-lm(Crime~.,uscrime)
vif(model_train) # For given predictors, multicollinearity can assessed by computing a score called the variance
inflation factor (or VIF), which measures how much the variance of a regression coefficient is inflated due to mu
lticollinearity in the model.
```

```
##         CM         So         Ed        Pol        Po2         LF        M.F
##   2.892448   5.342783   5.077447 104.658667 113.559262   3.712690   3.785934
##        Pop         NW         U1         U2     Wealth       Ineq       Prob
##   2.536708   4.674088   6.063931   5.088880  10.530375   8.644528   2.809459
##       Time
##   2.713785
```

```
#The smallest possible value of VIF is one (absence of multicollinearity). As a rule of thumb, a VIF value that e
xceeds 5 or 10 indicates a problematic amount of collinearity
```

As, we see from above, Po2 has a excessively high score(10 times larger than our threshold) of VIF.

We first eliminate this factor for our model.

```
model_train1<- lm(Crime~.-Po2,uscrime)
vif(model_train1)
```

```
##        CM        So        Ed       Pol        LF       M.F       Pop        NW
##  2.884547  5.317813  4.893078  5.341037  3.421040  3.779003  2.532206  4.277326
##        U1        U2    Wealth      Ineq      Prob      Time
##  5.997380  5.086769 10.496921  8.564507  2.605654  2.376902
```

After removing Po2, the predictor of Po1 has dramastically decreased. Then we need to take a look at the model output to determine which variables to be eliminated next.

```
summary(model_train1)
```

```
##
## Call:
## lm(formula = Crime ~ . - Po2, data = uscrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -442.55 -116.46    8.86  118.26  473.49
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.379e+03  1.569e+03  -4.066 0.000291 ***
## CM           8.986e+01  4.157e+01   2.162 0.038232 *
## So           5.669e+00  1.481e+02   0.038 0.969705
## Ed           1.773e+02  6.082e+01   2.915 0.006445 **
## Pol          9.653e+01  2.392e+01   4.035 0.000317 ***
## LF          -2.801e+02  1.408e+03  -0.199 0.843538
## M.F          1.822e+01  2.029e+01   0.898 0.376026
## Pop         -7.836e-01  1.286e+00  -0.609 0.546523
## NW           2.446e+00  6.187e+00   0.395 0.695239
## U1          -5.416e+03  4.178e+03  -1.296 0.204164
## U2           1.694e+02  8.215e+01   2.062 0.047441 *
## Wealth       9.072e-02  1.033e-01   0.878 0.386292
## Ineq         7.271e+01  2.256e+01   3.222 0.002921 **
## Prob        -4.285e+03  2.184e+03  -1.962 0.058484 .
## Time        -1.128e+00  6.692e+00  -0.168 0.867251
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 208.6 on 32 degrees of freedom
## Multiple R-squared:  0.7976, Adjusted R-squared:  0.709
## F-statistic: 9.006 on 14 and 32 DF,  p-value: 1.673e-07
```

'Wealth' , "U1" and "So" should also be removed, beacuse of their high vif score and high p-value (>0.05) which indicates that they are not satistically significant

```
model_train2<-lm(Crime~.-Wealth-Po2-U1-So,uscrime)
vif(model_train2)
```

```
##       CM       Ed      Pol       LF      M.F      Pop       NW       U2
## 2.772989 4.587900 3.390368 2.393316 2.684577 2.434556 3.473027 1.955472
##     Ineq     Prob     Time
## 4.775186 2.502649 2.314582
```

Now that, from the updated vif score list, we can tell the collinearity error is greatly minimized after eliminating several factors.

#step 4. Find the optimal model with less attributes to avoid overfitting As we know, the number of parameters is crucial for the accuracy of the model since more than enough parameters being factored into the model always cause overfitting (better than expected R^2).

First, we set our p-value threshold as 0.1, any attributes with greater than 0.1 p values are considered statistically less significant.

Under this threshold, only these predictors stay: CM, Ed, Po1, U2, Ineq, Prob

```
model_train3<- lm(Crime~+CM+Ed+Po1+U2+Ineq+Prob,uscrime)
summary(model_train3)
```

```
##
## Call:
## lm(formula = Crime ~ +CM + Ed + Pol + U2 + Ineq + Prob, data = uscrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -470.68  -78.41  -19.68  133.12  556.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5040.50     899.84  -5.602 1.72e-06 ***
## CM            105.02      33.30   3.154  0.00305 **
## Ed            196.47      44.75   4.390 8.07e-05 ***
## Pol           115.02      13.75   8.363 2.56e-10 ***
## U2             89.37      40.91   2.185  0.03483 *
## Ineq           67.65      13.94   4.855 1.88e-05 ***
## Prob        -3801.84    1528.10  -2.488  0.01711 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 200.7 on 40 degrees of freedom
## Multiple R-squared:  0.7659, Adjusted R-squared:  0.7307
## F-statistic: 21.81 on 6 and 40 DF,  p-value: 3.418e-11
```

Once again, we narrow down our threshold to 0.05 on the initial output, so that only these predictors prevail : CM, Ed, Po1, U2, Ineq. Then we build a model with these 5 predictors and compare with the model_train3

```
model_train4<- lm(Crime~+CM+Ed+Po1+U2+Ineq,uscrime)
summary(model_train4)
```

```
##
## Call:
## lm(formula = Crime ~ +CM + Ed + Pol + U2 + Ineq, data = uscrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -453.44  -98.59  -18.07  106.03  629.64
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5243.74     951.16  -5.513 2.13e-06 ***
## CM            101.98      35.32   2.887 0.006175 **
## Ed            203.08      47.42   4.283 0.000109 ***
## Pol           123.31      14.16   8.706 7.26e-11 ***
## U2             91.36      43.41   2.105 0.041496 *
## Ineq           63.49      14.68   4.324 9.56e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 213 on 41 degrees of freedom
## Multiple R-squared:  0.7296, Adjusted R-squared:  0.6967
## F-statistic: 22.13 on 5 and 41 DF,  p-value: 1.105e-10
```

We found U2 is less statistically significant comparing with other factors, so i tried to eliminate U2 as well and build another model.

```
model_train5<- lm(Crime~+CM+Ed+Po1+Ineq,uscrime)
summary(model_train5)
```

```
##
## Call:
## lm(formula = Crime ~ +CM + Ed + Pol + Ineq, data = uscrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -530.93  -91.88    7.56  137.72  576.84
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4249.22     858.51  -4.950 1.25e-05 ***
## CM             76.02      34.42   2.209 0.032714 *
## Ed            166.05      45.80   3.626 0.000773 ***
## Pol           129.80      14.38   9.029 2.16e-11 ***
## Ineq           64.09      15.27   4.197 0.000137 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 221.5 on 42 degrees of freedom
## Multiple R-squared:  0.7004, Adjusted R-squared:  0.6719
## F-statistic: 24.55 on 4 and 42 DF,  p-value: 1.595e-10
```

#step 5: comapring model's quality #Comparing the models' quality by finding the lowest RMSE and AIC. (Candidate: model_train3, model_train4 and model_train5)

```
pred_mod3<- predict(model_train3,test_crime)
rmse(test_crime$Crime,pred_mod3)
```

```
## [1] 141.5162
```

```
pred_mod4<- predict(model_train4,test_crime)
rmse(test_crime$Crime,pred_mod4)
```

```
## [1] 144.668
```

```
pred_mod5<- predict(model_train5,test_crime)
rmse(test_crime$Crime,pred_mod5)
```

```
## [1] 157.2065
```

```
AIC(model_train3)
```

```
## [1] 640.1661
```

```
AIC(model_train4)
```

```
## [1] 644.9286
```

```
AIC(model_train5)
```

```
## [1] 647.7503
```

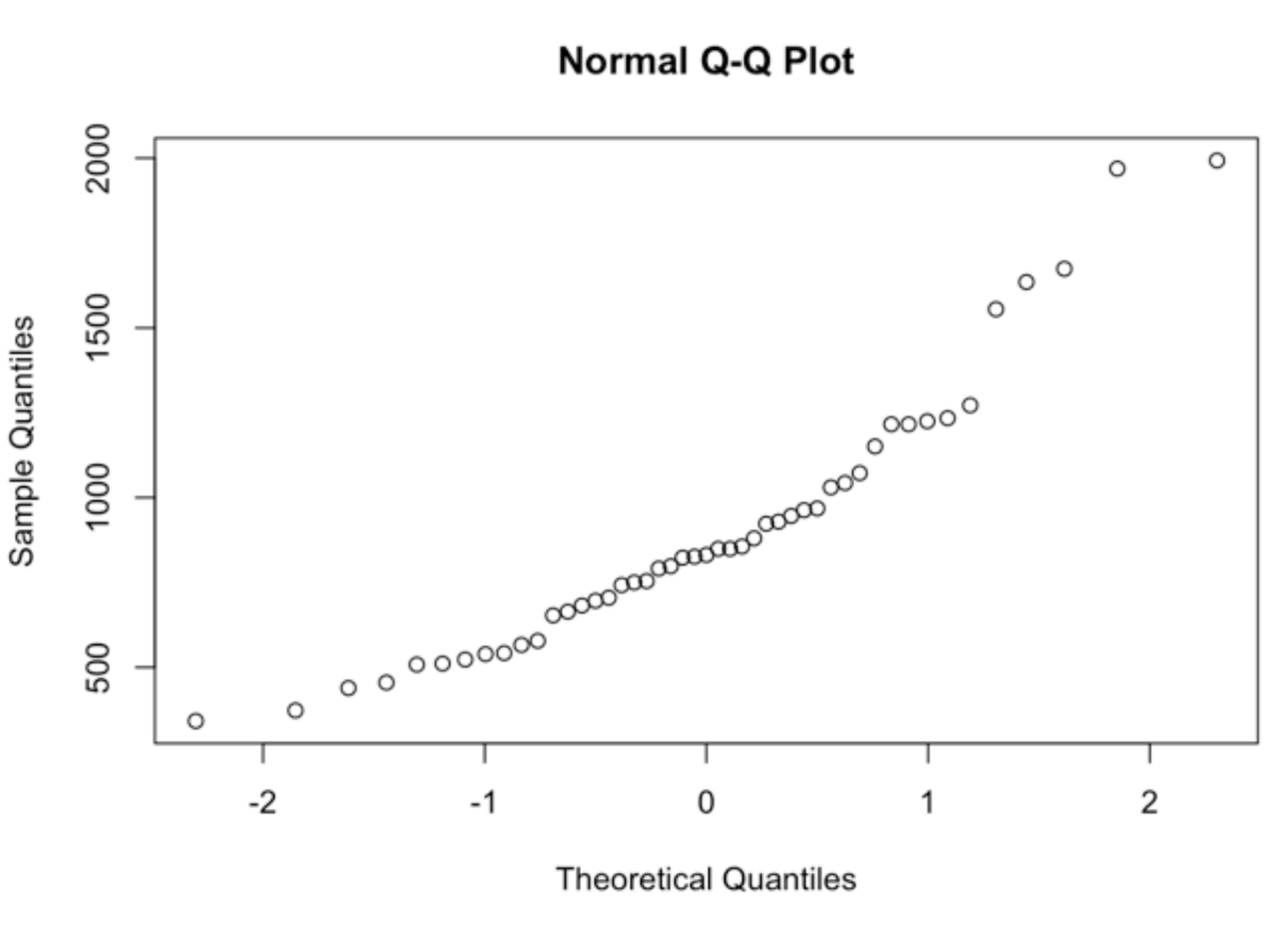
Based their performance on RMSE and AIC, model_train3 is the best model amongst these 3 finalists.(Lowest AIC & RMSE)

#final step: populate the given parameters from the questions and use the optimal model

```
test_data<-data.frame(CM=14.0,So=0,Ed=10.0,Po1=12.0,Po2=15.5,LF=0.64,M.F=94.0, Pop=150, NW=1.1, U1=0.12,U2=3.6,We
alth=3200,Ineq=20.1,Prob=0.04,Time=39.0)
pred_final<-predict(model_train3,test_data)
pred_final
```

```
##        1
## 1304.245
```

```
qqnorm(uscrime$Crime)
```



Conclusion: As we see above, the final prediction is 1304 which falls in line of the Normal QQ plot.