Linear Regression Life Expectancy Group Project

2022-04-22

QF112 - Statistics in R for Quantitative Finance

Professor Diaco

I pledge my honor that I have abided by the Stevens Honor System.

Group Names: Andrew Finn, Sid Bhatia, Jacob Bussell, Rocco Mathews, Alex Woodward

Project Description

This following project uses a data set known as "LifeExp.csv." The goal of this project is to use the linear model (lm) function and only the comma-separated values (csv) data to predict the dependent variable LifeExpectancy.

In approaching this project, we wanted to leverage the tools and methodologies learned in our QF112 - Statistics in R for QF class in order to produce the best linear model. Specifically, we wanted to do the following:

- Use as minimal as predictors as possible.
- Avoid overfitting the linear model to training data.
- Use stratified random sampling to obtain a sample data that best represents the population.
- Experiment with different selection algorithms.
- Use analysis of variance (ANOVA) to identify significant categorical variables.
- Maximize the coefficient of determination ($R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$).
- Minimize the mean squared error (MSE) andresidual standard error (RSE) when applied to testing data.

Project Set-Up

In beginning the project, we needed to appropriately set up the data and project infrastructure.

Libraries

Firstly, we needed to identify the relevant libaries to use for the project. We chose the following:

- robustHD: "Robust methods for high-dimensional data, in particular linear model selection techniques based on least angle regression and sparse regression."
- leaps: "Regression subset selection, including exhaustive search."
- splitstackshape: "Stack and Reshape Datasets After Splitting Concatenated Values."
- caret: "Short for Classification And REgression Training) is a set of functions that attempt to streamline the process for creating predictive models."
- olsrr: "Tools for Building OLS Regression Models."

- dplyr: "A grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges."

```r
# Import relevant libraries
library(robustHD)
library(leaps)
library(splitstackshape)
library(caret)
library(olsrr)
library(dplyr)
```

Data Organization

After importing the relevant libraries, we had to organize the incoming data set and set the seed. The seed() function sets the starting the number used to generate a sequence of random numbers, allowing us to get the same result when we construct our random samples for testing and experimentation. We do this in the GD or "Gather Data" function.

Additionally, after the initial reading, we had to filter some aspects of the data and make sure to reclassify the categorical variables as "factors" in R. Although we use the "stringAsFactors = TRUE" argument, we found that the Country variable was not being identified as a factor. This is imperative as further in the project when we use analysis of variance or anova() in R, we have to make sure Country is a categorical variable.

Next, we wanted to create a stratified random sample for our Country categorical variable. Stratified random sampling (SRS) is a method of sampling that involves the division of a population into smaller sub-groups known as strata. The main advantage of SRS is that it captures key population characteristics in the sample, resulting in a better sample to use when constructing our linear model.

In this case, we wanted to stratify based on country since each country has different geographical and political influences that can heavily impact their life expectancy. For example, a country such as Afghanistan that has been engaged in war for countless years with lower socioeconomic status would have a lower life expectancy that a country that has not been engaged in wars for quite some time such as Australia. In addition, the differences in government and economic systems such as Democracy vs. Dictatorships and Capitalism vs. Socialism vs. Communism could be possible confounding variables within the countries themselves, meaning every country should be included in our random sample in order to result in a more accurate linear model.

After stratifying, we made sure to split our data set into training and testing data. We did this by using the subset function to retrieve training and testing data using the stratified random sampling using company as strata. In addition, we made sure to retrieve the dependent variable (LifeExpectancy) from the training and testing data respectively.

```r
# Create "Gather Data" function
GD <- function() {

  # Set seed as pi
  set.seed(3.1415)

  # Disable printing results in scientific notation
  options(scipen=999)

  # Read in LifeExp data and cast as global variable
```

```r
  Data <<- data.frame(read.csv("LifeExp.csv"),c(1:1005), stringsAsFactors = TRUE)

  # Retrieve the total number of data entries or overall data size
  n <- length(Data$LifeExpectancy)

  # Filter aspects of data
  Data <<- Data[-397,]

  # Convert categorical variables to factors
  Data$Country <<- factor(Data$Country)

  # Construct stratified random sample using "Country" as strata
  Train1 <- stratified(Data,"Country", .5)
  Test <<- subset(Data, !(Data$c.1.1005. %in% c(Train1$c.1.1005.)))
  Train2 <- Test[sample(1:526,24),]
  Test <<- subset(Test, !(Test$c.1.1005. %in% c(Train2$c.1.1005.)))
  Train <<- rbind(Train2,Train1)

  # Make separate dependent variables for Training and Testing Data
  TrainD <<- Train$LifeExpectancy
  TestD <<- Test$LifeExpectancy

  # Remove index & dependent variable
  Train <<- Train[,-c(3,22)]
  Test <<- Test[,-c(3,22)]
  Data <<- Data[,-c(22)]
}

# Invoke gather data function
GD()
```

Parameter Identification & Testing

After setting up and organizing the data, we get into the main course of the project: identifying and testing for pertinent parameters for our linear model.

Analysis of Variance (ANOVA)

The main method for identifying and testing for parameters we used was analysis of variance (ANOVA). ANOVA is a statistical method used to compare variances across the means (or average) of different groups. More specifically, ANOVA separates observed variance data into different components to use for additional tests where $F = \frac{\text{MST}}{\text{MSE}}$ and the F-statistic can be used to determine whether a categorical variable is statistically significant.

We decided to use ANOVA since Country is an incredibly significant categorical variable for our model and one of the primary factors we used for our final model. After further experimentation and testing, as you'll see in the next few sections, we deemed ANOVA and specifically the Country parameter to be incredibly significant in comparison to other methodologies, rendering $F = 103$ and the $p < 0.0000000000000002$.

```r
# Create ANOVA function
anva <- function() {
```

```
  # Conduct analysis of variance test
  ANOVA<-aov(LifeExpectancy~Country, data = Data)
  # Display summary of ANOVA
  summary(ANOVA)
}

# Invoke analysis of variance function
anva()
```

```
##              Df Sum Sq Mean Sq F value              Pr(>F)
## Country     130  73336   564.1     103 <0.0000000000000002 ***
## Residuals   873   4783     5.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Principal Component Analysis (PCA)

Another methodology we experimented with was principal component analysis (PCA). PCA is a technique for reducing the dimensionality of such datasets, increasing interpretability but at the same time minimizing information loss. It does so by creating new uncorrelated variables that successively maximize variance. Finding such new variables, the principal components, reduces to solving an eigenvalue/eigenvector problem. PCA is especially useful in identifying variables that are collinear or a model that has multicollinearity.

Firstly, we decided to choose a forward subset selection algorithm to choosing the statistically significant parameters. The forward selection starts with no predictors in the model, iteratively adds the most contributive predictors, and stops when the improvement is no longer statistically significant. Next, we isolate the most statistically significant variables given the input argument p. If we only want five parameters, we would invoke Pslct1(5).

However, after we experimented with different parameters (3, 5, 7), we ended up with MSE equal to $16.48, 14.73, 14.03$ respectively. Although these mean squared errors were not bad, we realized that with further experimentation, we could create an even better model with lower error. As such, this linear model was not used as our final model.

```
# First Subset Selection and PCA model (not including country due to PCA)
Pslct1 <- function(p) {

  # Conduct forward subset selection
  SubSel <<- regsubsets(TestD~., data=Test[,-1], method = "forward")

  # Isolate "best" or most statistically significant independent variables
  coefs <- coef(SubSel, p)
  name <<- c(names(coefs[-1]))
  Train <<- Train[name]
  Test <<- Test[name]

  # Use parameters to construct PCA model
  modelPCA <<- train(
    TrainD~ .,
    data = cbind(Train,TrainD),
    method = 'lm',
    preProcess = c("center", "scale", "pca")
```

```
  )
  Pred <<- predict(modelPCA,Test)
  MSE <<- mean((TestD-Pred)^2)
}
```

Double Subset Selection

For our final model, we decided to use double subset selection as well include Country. The reason for this since we already identified Country to be an incredibly significant categorical variable for our linear model. As such, we ran one subset selection including country to identify quantitative variables that would be significant.

After isolating the top five independent variables excluding countries, we ran another forward subset model that selects the best subset of predictors that minimize the MSE using ols step best subset.

After retrieving the best parameters from subset selection, we choose three different parameters to include in our linear model, which entailed HIVAIDS, Schooling, and Country.

Finally, we used our parameters to create our final linear model, make predictions, and compute the MSE. This mean squared error was the lowest out of all models at 3.76.

```
# Second Double Subset Selection and Normal Linear Model (including Country)
Pslct2 <- function(){

  # Subset Selection (excluding Countries)
  SubSel1<<-regsubsets(TestD~.,data=Test[,-1],method = "forward")

  # Isolate Best Independent Variables (excluding Countries)
  coefs <- coef(SubSel1, 5)
  name <<- c(names(coefs[-1]),"Country")
  Train <<- Train[name]
  Test <<- Test[name]

  # Subset Selection (Including Countries)
  model <- lm(TestD~.,data=Test)
  SubSel2 <<- ols_step_best_subset(model,method = "forward")

  # Isolate Best Independent Variables (Including Countries)
  name <<- c(unlist(strsplit(SubSel2$predictors[3],split = " ")))
  Train <<- Train[name]
  Test <<- Test[name]

  # Use Parameters to Make Model and Prediction
  model2 <<- lm(TrainD~.,data = Train)

  # Conduct predictions using test set
  Pred <<- predict(model2,Test)

  # Compute MSE
  MSE <<- mean((TestD-Pred)^2)
}

Pslct2()
MSE
```

```
## [1] 3.765423
```

Model Analysis

Summary of Model

As seen by the model summary, our final linear model ended up having an RSE = 1.824 with an adjusted $R^2 = 0.9569$, indicating a strong model for predicting life expectancy. One thing to note is that we do have many factors for Country. We tried to address this by possibly removing the insignificant factors, as seen in the following section.

```
summary(model2)
```

```
##
## Call:
## lm(formula = TrainD ~ ., data = Train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2007 -0.6329 -0.0624  0.5037  6.1331
##
## Coefficients:
##                                Estimate Std. Error t value
## (Intercept)                    47.58994    1.44235  32.995
## HIVAIDS                        -0.35495    0.02971 -11.948
## Schooling                       1.21895    0.12979   9.392
## CountryAlbania                 12.87745    1.18981  10.823
## CountryAlgeria                 10.51003    1.36102   7.722
## CountryAngola                  -7.90207    1.53102  -5.161
## CountryArgentina                7.54224    1.57504   4.789
## CountryArmenia                 11.48211    1.18256   9.710
## CountryAustralia                9.51188    1.88926   5.035
## CountryAustria                 14.31682    1.43774   9.958
## CountryAzerbaijan               9.49719    1.26162   7.528
## CountryBangladesh              10.94929    1.22473   8.940
## CountryBelarus                  4.26372    1.43097   2.980
## CountryBelgium                 12.90796    1.48308   8.703
## CountryBelize                   6.60766    1.30217   5.074
## CountryBenin                   -0.27124    1.22516  -0.221
## CountryBhutan                   5.95086    1.15680   5.144
## CountryBosnia and Herzegovina  12.65856    1.33648   9.472
## CountryBotswana                -1.01458    1.44895  -0.700
## CountryBrazil                   9.09402    1.32118   6.883
## CountryBulgaria                 8.78814    1.36675   6.430
## CountryBurkina Faso             2.52765    1.37624   1.837
## CountryBurundi                 -0.50860    1.24114  -0.410
## CountryCabo Verde               9.71401    1.29159   7.521
## CountryCambodia                 6.08583    1.34505   4.525
## CountryCameroon                -2.28509    1.34618  -1.697
## CountryCanada                  15.56977    1.50235  10.364
## CountryCentral African Republic -1.75789    1.36590  -1.287
## CountryChad                    -3.13465    1.54524  -2.029
```

```
## CountryChile                  13.41026    1.54794   8.663
## CountryChina                  12.93003    1.25702  10.286
## CountryColombia               10.75177    1.28086   8.394
## CountryComoros                 2.85606    1.23176   2.319
## CountryCosta Rica             15.26877    1.32254  11.545
## CountryCroatia                11.78798    1.68437   6.998
## CountryCyprus                 14.93903    1.33518  11.189
## CountryDjibouti                8.98138    1.59042   5.647
## CountryDominican Republic      9.26520    1.29766   7.140
## CountryEcuador                11.37687    1.24895   9.109
## CountryEl Salvador             8.92468    1.24036   7.195
## CountryEritrea                 9.36472    1.43677   6.518
## CountryEstonia                 7.76814    1.52915   5.080
## CountryEthiopia                5.41843    1.53073   3.540
## CountryFiji                    3.88088    1.32499   2.929
## CountryFrance                 14.05645    1.51564   9.274
## CountryGabon                   2.11120    1.43643   1.470
## CountryGeorgia                10.59466    1.29607   8.174
## CountryGermany                13.07745    1.55282   8.422
## CountryGhana                   3.18519    1.22615   2.598
## CountryGreece                 14.46740    1.51182   9.570
## CountryGuatemala              15.22608    1.33879  11.373
## CountryGuinea                  0.76010    1.53154   0.496
## CountryGuinea-Bissau           0.78885    1.53324   0.514
## CountryGuyana                  5.20972    1.17580   4.431
## CountryHonduras               12.44098    1.17199  10.615
## CountryIndia                   5.66006    1.23159   4.596
## CountryIndonesia               5.58890    1.26834   4.406
## CountryIraq                   12.13493    1.23301   9.842
## CountryIreland                12.83409    1.95313   6.571
## CountryIsrael                 14.35024    1.50047   9.564
## CountryItaly                  15.72998    1.48193  10.615
## CountryJamaica                12.06631    1.28758   9.371
## CountryJordan                  9.08045    1.33132   6.821
## CountryKazakhstan              2.55987    1.40956   1.816
## CountryKenya                   0.50375    1.27141   0.396
## CountryKiribati                3.32293    1.26904   2.618
## CountryLatvia                  6.62687    1.47465   4.494
## CountryLebanon                 9.50219    1.36965   6.938
## CountryLesotho                -4.18601    1.42854  -2.930
## CountryLiberia                 0.93555    1.52978   0.612
## CountryLithuania               5.46169    1.49291   3.658
## CountryLuxembourg             18.24908    1.36387  13.380
## CountryMadagascar              4.39659    1.22490   3.589
## CountryMalawi                 -3.10444    1.29468  -2.398
## CountryMalaysia               10.90635    1.30373   8.365
## CountryMaldives               13.94556    1.27343  10.951
## CountryMali                   -0.20924    1.27602  -0.164
## CountryMalta                  15.31698    1.38605  11.051
## CountryMauritania              7.16510    1.34824   5.314
## CountryMauritius               8.80480    1.27492   6.906
## CountryMexico                 13.17914    1.27616  10.327
## CountryMongolia                3.28624    1.26321   2.601
## CountryMontenegro              9.75524    1.48798   6.556
```

```
## CountryMorocco                  11.96303    1.23467   9.689
## CountryMozambique               -0.51450    1.20928  -0.425
## CountryMyanmar                   7.08921    1.22683   5.778
## CountryNamibia                   4.03761    1.56877   2.574
## CountryNepal                     6.65512    1.17323   5.672
## CountryNetherlands              12.18114    1.88245   6.471
## CountryNicaragua                12.55991    1.24759  10.067
## CountryNiger                     7.44308    1.62651   4.576
## CountryNigeria                  -5.23517    1.34115  -3.903
## CountryPakistan                  8.76606    1.18916   7.372
## CountryPanama                   13.64614    1.31647  10.366
## CountryPapua New Guinea          5.42984    1.23591   4.393
## CountryParaguay                 10.58535    1.28377   8.246
## CountryPeru                     10.50120    1.22776   8.553
## CountryPhilippines               5.66672    1.26241   4.489
## CountryPoland                    8.89507    1.38905   6.404
## CountryPortugal                 13.21430    1.50424   8.785
## CountryRomania                  10.14998    1.33260   7.617
## CountryRussian Federation        2.49981    1.35035   1.851
## CountryRwanda                    2.39242    1.23204   1.942
## CountrySamoa                    12.19777    1.30040   9.380
## CountrySao Tome and Principe     5.51355    1.23280   4.472
## CountrySenegal                   7.88602    1.16929   6.744
## CountrySerbia                    9.97572    1.34443   7.420
## CountrySeychelles                9.23993    1.30712   7.069
## CountrySierra Leone             -8.18403    1.33316  -6.139
## CountrySolomon Islands           9.56808    1.22606   7.804
## CountrySouth Africa              1.50969    1.35605   1.113
## CountrySpain                    14.93116    1.45259  10.279
## CountrySri Lanka                 9.74214    1.34043   7.268
## CountrySuriname                  8.73906    1.37961   6.334
## CountrySwaziland                 2.09536    1.54299   1.358
## CountrySweden                   14.76430    1.75916   8.393
## CountrySyrian Arab Republic     12.79040    1.53339   8.341
## CountryTajikistan                7.08387    1.24392   5.695
## CountryThailand                 10.51787    1.29041   8.151
## CountryTimor-Leste               4.31966    1.58484   2.726
## CountryTogo                     -2.27365    1.56570  -1.452
## CountryTonga                     7.70834    1.34058   5.750
## CountryTrinidad and Tobago       9.44020    1.29882   7.268
## CountryTunisia                   9.66698    1.38605   6.974
## CountryTurkey                   12.64956    1.25777  10.057
## CountryTurkmenistan              4.58566    1.23693   3.707
## CountryUganda                   -2.24887    1.20533  -1.866
## CountryUkraine                   4.45762    1.40440   3.174
## CountryUruguay                   9.78235    1.47284   6.642
## CountryUzbekistan                6.40576    1.27628   5.019
## CountryVanuatu                  10.84735    1.24045   8.745
## CountryZambia                   -1.42533    1.40642  -1.013
## CountryZimbabwe                 -1.25666    1.48372  -0.847
##                                            Pr(>|t|)
## (Intercept)                     < 0.0000000000000002 ***
## HIVAIDS                         < 0.0000000000000002 ***
## Schooling                       < 0.0000000000000002 ***
```

```
## CountryAlbania                  < 0.0000000000000002 ***
## CountryAlgeria                  0.000000000000108691 ***
## CountryAngola                   0.000000401574390071 ***
## CountryArgentina                0.000002435713989606 ***
## CountryArmenia                  < 0.0000000000000002 ***
## CountryAustralia                0.000000749863029156 ***
## CountryAustria                  < 0.0000000000000002 ***
## CountryAzerbaijan               0.000000000000399515 ***
## CountryBangladesh               < 0.0000000000000002 ***
## CountryBelarus                              0.003077 **
## CountryBelgium                  < 0.0000000000000002 ***
## CountryBelize                   0.000000617521484485 ***
## CountryBenin                                0.824909
## CountryBhutan                   0.000000437183651426 ***
## CountryBosnia and Herzegovina   < 0.0000000000000002 ***
## CountryBotswana                             0.484231
## CountryBrazil                   0.000000000025234624 ***
## CountryBulgaria                 0.000000000395441740 ***
## CountryBurkina Faso                         0.067067 .
## CountryBurundi                              0.682200
## CountryCabo Verde               0.000000000000418041 ***
## CountryCambodia                 0.000008170713251604 ***
## CountryCameroon                             0.090453 .
## CountryCanada                   < 0.0000000000000002 ***
## CountryCentral African Republic             0.198908
## CountryChad                                 0.043219 *
## CountryChile                    < 0.0000000000000002 ***
## CountryChina                    < 0.0000000000000002 ***
## CountryColombia                 0.000000000000001016 ***
## CountryComoros                              0.020958 *
## CountryCosta Rica               < 0.0000000000000002 ***
## CountryCroatia                  0.000000000012269840 ***
## CountryCyprus                   < 0.0000000000000002 ***
## CountryDjibouti                 0.000000032604506557 ***
## CountryDominican Republic       0.000000000004999778 ***
## CountryEcuador                  < 0.0000000000000002 ***
## CountryEl Salvador              0.000000000003507892 ***
## CountryEritrea                  0.000000000234432230 ***
## CountryEstonia                  0.000000600498130126 ***
## CountryEthiopia                             0.000452 ***
## CountryFiji                                 0.003612 **
## CountryFrance                   < 0.0000000000000002 ***
## CountryGabon                                0.142481
## CountryGeorgia                  0.000000000000004819 ***
## CountryGermany                  0.000000000000000834 ***
## CountryGhana                                0.009760 **
## CountryGreece                   < 0.0000000000000002 ***
## CountryGuatemala                < 0.0000000000000002 ***
## CountryGuinea                               0.619980
## CountryGuinea-Bissau                        0.607211
## CountryGuyana                   0.000012393225380048 ***
## CountryHonduras                 < 0.0000000000000002 ***
## CountryIndia                    0.000005929841160947 ***
## CountryIndonesia                0.000013790399163956 ***
```

```
## CountryIraq                   < 0.0000000000000002 ***
## CountryIreland                  0.000000000170482862 ***
## CountryIsrael                 < 0.0000000000000002 ***
## CountryItaly                  < 0.0000000000000002 ***
## CountryJamaica                < 0.0000000000000002 ***
## CountryJordan                   0.000000000037212381 ***
## CountryKazakhstan                          0.070170 .
## CountryKenya                               0.692177
## CountryKiribati                            0.009197 **
## CountryLatvia                   0.000009373251345614 ***
## CountryLebanon                  0.000000000017969746 ***
## CountryLesotho                             0.003597 **
## CountryLiberia                             0.541208
## CountryLithuania                           0.000291 ***
## CountryLuxembourg             < 0.0000000000000002 ***
## CountryMadagascar                          0.000376 ***
## CountryMalawi                              0.016988 *
## CountryMalaysia                 0.000000000000001247 ***
## CountryMaldives               < 0.0000000000000002 ***
## CountryMali                                0.869836
## CountryMalta                  < 0.0000000000000002 ***
## CountryMauritania               0.000000185596330157 ***
## CountryMauritius                0.000000000021877931 ***
## CountryMexico                 < 0.0000000000000002 ***
## CountryMongolia                            0.009655 **
## CountryMontenegro               0.000000000186541614 ***
## CountryMorocco                < 0.0000000000000002 ***
## CountryMozambique                          0.670751
## CountryMyanmar                  0.000000016045816300 ***
## CountryNamibia                             0.010450 *
## CountryNepal                    0.000000028472028332 ***
## CountryNetherlands              0.000000000310243916 ***
## CountryNicaragua              < 0.0000000000000002 ***
## CountryNiger                    0.000006481223003541 ***
## CountryNigeria                             0.000113 ***
## CountryPakistan                 0.000000000001117467 ***
## CountryPanama                 < 0.0000000000000002 ***
## CountryPapua New Guinea         0.000014603796456989 ***
## CountryParaguay                 0.000000000000002922 ***
## CountryPeru                     0.000000000000000324 ***
## CountryPhilippines              0.000009586901754105 ***
## CountryPoland                   0.000000000461786272 ***
## CountryPortugal               < 0.0000000000000002 ***
## CountryRomania                  0.000000000000220945 ***
## CountryRussian Federation                  0.064935 .
## CountryRwanda                              0.052918 .
## CountrySamoa                  < 0.0000000000000002 ***
## CountrySao Tome and Principe    0.000010312350053518 ***
## CountrySenegal                  0.000000000059545839 ***
## CountrySerbia                   0.000000000000813812 ***
## CountrySeychelles               0.000000000007857906 ***
## CountrySierra Leone             0.000000002148150552 ***
## CountrySolomon Islands          0.000000000000062445 ***
## CountrySouth Africa                        0.266307
```

```
## CountrySpain                     < 0.0000000000000002 ***
## CountrySri Lanka                 0.000000000002194420 ***
## CountrySuriname                  0.000000000693471045 ***
## CountrySwaziland                             0.175295
## CountrySweden                    0.000000000000001026 ***
## CountrySyrian Arab Republic      0.000000000000001482 ***
## CountryTajikistan                0.000000025247698454 ***
## CountryThailand                  0.000000000000005688 ***
## CountryTimor-Leste                           0.006724 **
## CountryTogo                                  0.147306
## CountryTonga                     0.000000018733491018 ***
## CountryTrinidad and Tobago       0.000000000002189533 ***
## CountryTunisia                   0.000000000014265663 ***
## CountryTurkey                    < 0.0000000000000002 ***
## CountryTurkmenistan                          0.000242 ***
## CountryUganda                                0.062867 .
## CountryUkraine                               0.001630 **
## CountryUruguay                   0.000000000111195107 ***
## CountryUzbekistan                0.000000809291403140 ***
## CountryVanuatu                   < 0.0000000000000002 ***
## CountryZambia                                0.311513
## CountryZimbabwe                              0.397564
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.824 on 369 degrees of freedom
## Multiple R-squared:  0.9683, Adjusted R-squared:  0.9569
## F-statistic: 85.29 on 132 and 369 DF,  p-value: < 0.00000000000000022
```

Level Adjustment

As aforementioned, we tried to combine some insignificant levels to try to reduce the total number of parameters for our model. However, we realized after experimentation and testing, this did not have a significant impact on our MSE and RSE. As such, we did not implement it in our final model.

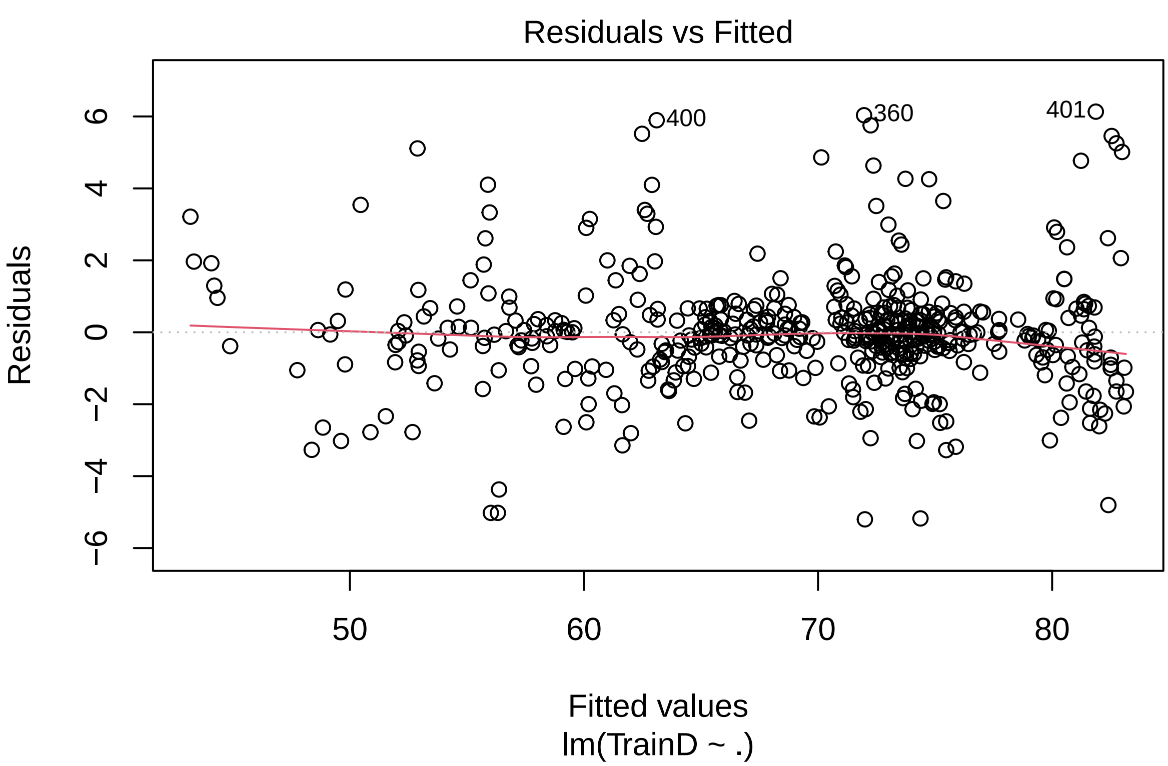
```r
# Combines insignificant levels for Pslct2()
GetLev <- function(i){
  sum<-summary(model2)

  if(is.na(sum$coefficients[,4][i])==TRUE){
    print(Clevs)

    #Assigns the list of insignificant levels to the same arbitrary insignificant level.
    levels(Test$Country)[as.numeric(Clevs)]<<-c(14)
    levels(Train$Country)[as.numeric(Clevs)]<<-c(14)

    #Makes a new model with consolidated levels

    model2<<-lm(TrainD~.,data = Train)

    #New MSE, very slightly higher. Don't know if this really makes much of an improvement.
    Pred<<-predict(model2,Test)
```

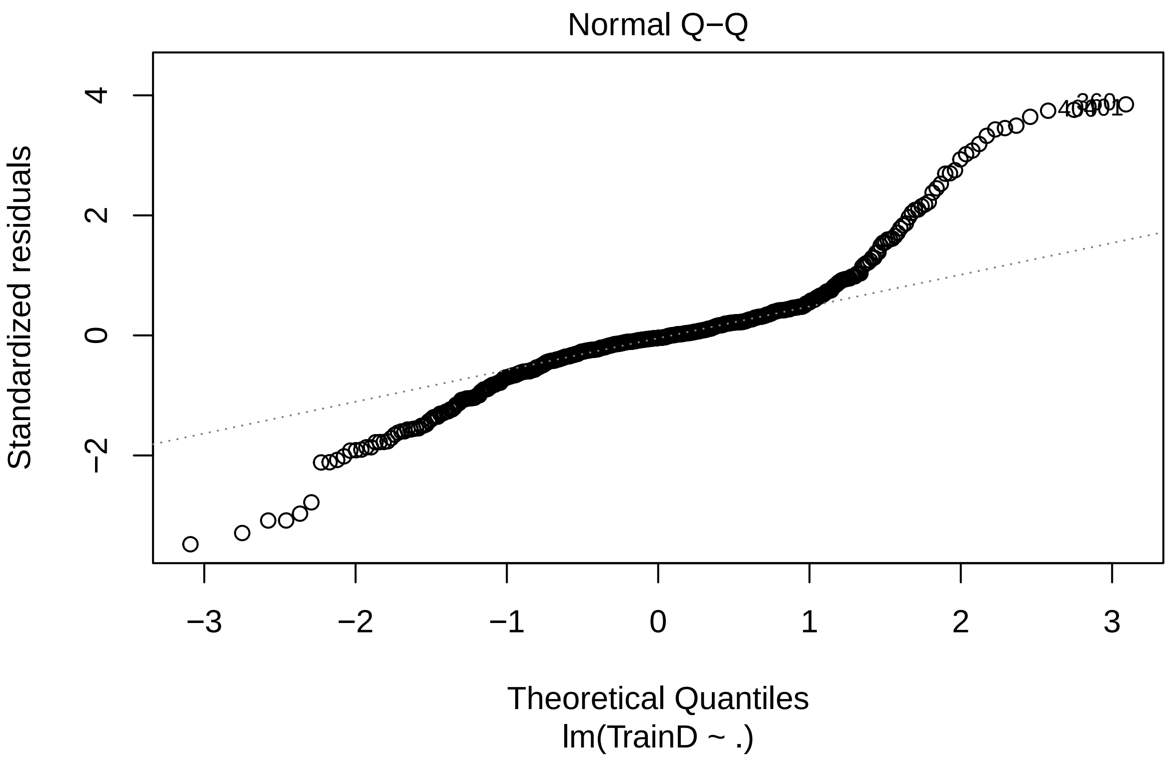
```
    MSE<<-mean((TestD-Pred)^2)
  }

  else{
    #If the p-value is insignificant at alpha==.05 then that level is stored to be used in the first if
    if(sum$coefficients[,4][i]>.05){
    level<-names(sum$coefficients[,4][i])
    Clevs<<-c(Clevs,substr(level,8,nchar(level)))
    GetLev(i+1)

    }
    #Recursively repeats if significant
    else{
      GetLev(i+1)
    }
  }
}
```

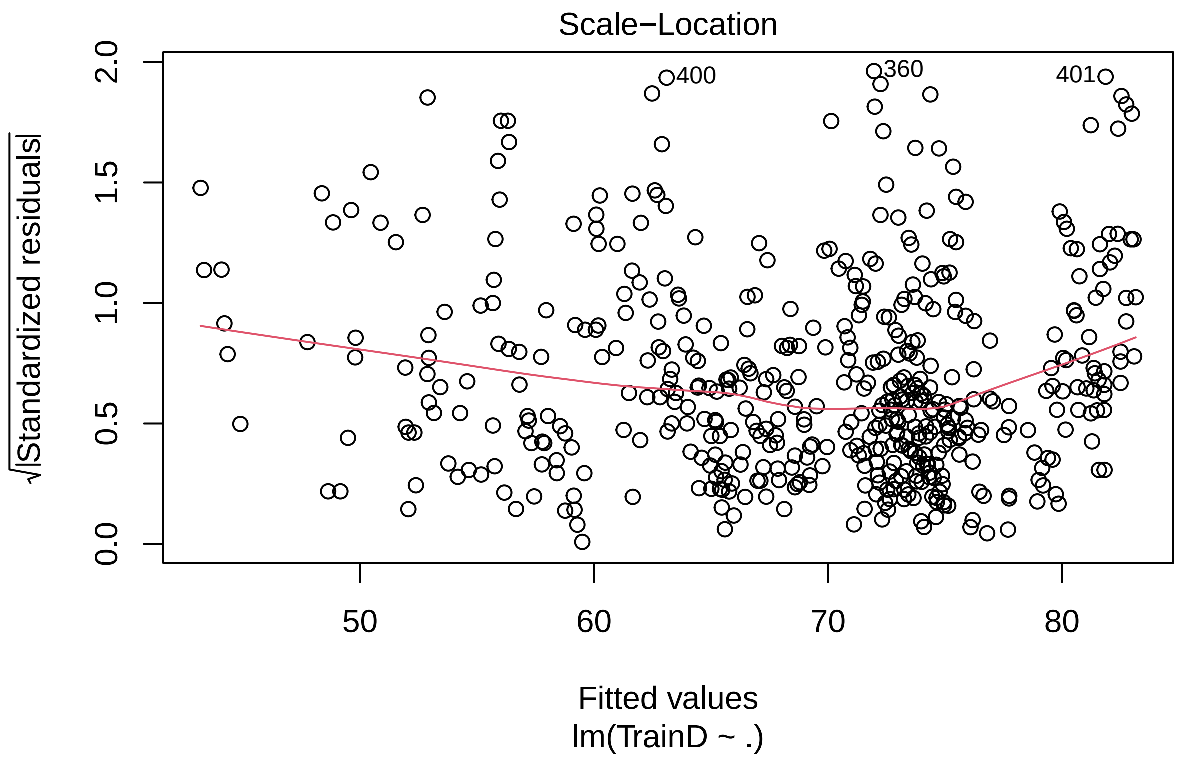
Graphical Analysis

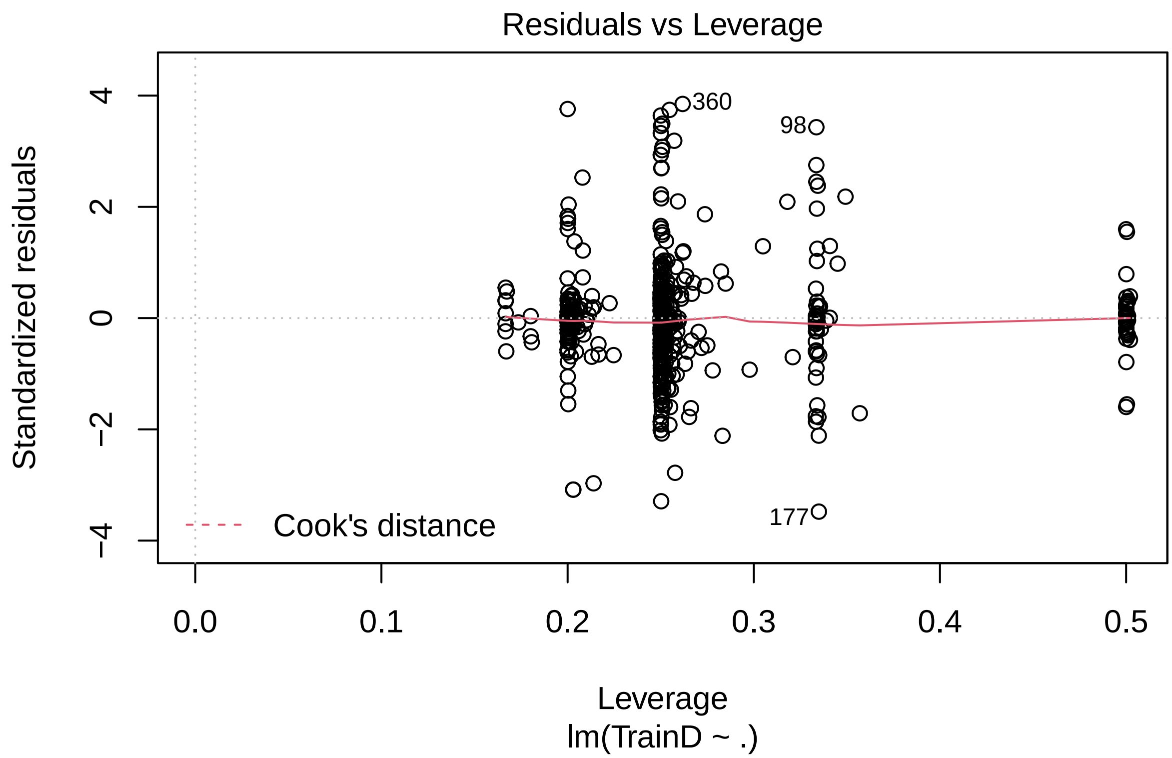
Finally, for our linear model we decided to display some prominent graphs. As seen in the Residuals vs. Fitted graph, the residuals are distributed relatively randomly above and below the red line centered about zero, indicating a decent sign. The Normal Q-Q plot is slightly curved around the center, meaning that the model may have some bias, but it still has a linear, upward trend.

```
plot(model2)
```




Normal Q−Q
Standardized residuals
4
2
0
−2
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(TrainD ~ .)
360
40401


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
400
360
401
50
60
70
80
Fitted values
lm(TrainD ~ .)



Saving Model

The following code saves the final linear model for easy access and testing:

```
save(model2, file = 'LMModel.Rdata')
```