Final Project Appendix A

Olivia Yuengling

2024-12-16

Data Cleaning and Processing

```
# loads any necessary libraries
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```
library(dplyr)
library(ggplot2)
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.3.3
```

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

```
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.3.3
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(gplots)
```

```
## Warning: package 'gplots' was built under R version 4.3.3
```

```
##
## Attaching package: 'gplots'
##
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(tidyr)

# loads the dataset

# install.packages("tidytuesdayR")

tuesdata <- tidytuesdayR::tt_load('2024-10-22')
```

```
## ---- Compiling #TidyTuesday Information for 2024-10-22 ----
## --- There is 1 file available ---
##
##
## ── Downloading files
─────────────────────────────────────────────────────────────
##
##   1 of 1: "cia_factbook.csv"
```

```
## OR
tuesdata <- tidytuesdayR::tt_load(2024, week = 43)
```

```
## ---- Compiling #TidyTuesday Information for 2024-10-22 ----
## --- There is 1 file available ---
##
##
## ── Downloading files
─────────────────────────────────────────────────────────────
##
##   1 of 1: "cia_factbook.csv"
```

```
cia_factbook <- tuesdata$cia_factbook

# displays the rows and columns of the dataset
head(cia_factbook)
```

```
## # A tibble: 6 × 11
##   country        area birth_rate death_rate infant_mortality_rate
internet_users
```

```
##   <chr>         <dbl>      <dbl>      <dbl>                 <dbl>
<dbl>
## 1 Russia       1.71e7       11.9      13.8                   7.08
40853000
## 2 Canada       9.98e6       10.3       8.31                  4.71
26960000
## 3 United Stat… 9.83e6       13.4       8.15                  6.17
245000000
## 4 China        9.60e6       12.2       7.44                 14.8
389000000
## 5 Brazil       8.51e6       14.7       6.54                 19.2
75982000
## 6 Australia    7.74e6       12.2       7.07                  4.43
15810000
## # ℹ 5 more variables: life_exp_at_birth <dbl>, maternal_mortality_rate
<dbl>,
## #   net_migration_rate <dbl>, population <dbl>, population_growth_rate
<dbl>
```

```
str(cia_factbook)
```

```
## spc_tbl_ [259 × 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ country                : chr [1:259] "Russia" "Canada" "United States"
"China" ...
##  $ area                   : num [1:259] 17098242 9984670 9826675 9596960
8514877 ...
##  $ birth_rate             : num [1:259] 11.9 10.3 13.4 12.2 14.7 ...
##  $ death_rate             : num [1:259] 13.83 8.31 8.15 7.44 6.54 ...
##  $ infant_mortality_rate  : num [1:259] 7.08 4.71 6.17 14.79 19.21 ...
##  $ internet_users         : num [1:259] 4.09e+07 2.70e+07 2.45e+08
3.89e+08 7.60e+07 ...
##  $ life_exp_at_birth      : num [1:259] 70.2 81.7 79.6 75.2 73.3 ...
##  $ maternal_mortality_rate: num [1:259] 34 12 21 37 56 7 200 77 51 97 ...
##  $ net_migration_rate     : num [1:259] 1.69 5.66 2.45 -0.32 -0.15 5.74 -
0.05 0 0.42 -0.93 ...
##  $ population             : num [1:259] 1.42e+08 3.48e+07 3.19e+08
1.36e+09 2.03e+08 ...
##  $ population_growth_rate : num [1:259] -0.03 0.76 0.77 0.44 0.8 1.09 1.25
0.95 1.17 1.88 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   country = col_character(),
##   ..   area = col_double(),
##   ..   birth_rate = col_double(),
##   ..   death_rate = col_double(),
##   ..   infant_mortality_rate = col_double(),
##   ..   internet_users = col_double(),
##   ..   life_exp_at_birth = col_double(),
##   ..   maternal_mortality_rate = col_double(),
##   ..   net_migration_rate = col_double(),
##   ..   population = col_double(),
##   ..   population_growth_rate = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
summary(cia_factbook) # prints a 5 number summary of the dataset
```

```
##    country               area             birth_rate      death_rate
##  Length:259         Min.   :       0   Min.   : 6.72   Min.   : 1.530
##  Class :character   1st Qu.:     616   1st Qu.:11.84   1st Qu.: 5.930
##  Mode  :character   Median :   51197   Median :16.89   Median : 7.630
##                     Mean   :  530888   Mean   :19.66   Mean   : 7.907
##                     3rd Qu.:  338145   3rd Qu.:24.91   3rd Qu.: 9.450
##                     Max.   :17098242   Max.   :46.12   Max.   :17.490
##                     NA's   :2          NA's   :35      NA's   :34
##  infant_mortality_rate internet_users      life_exp_at_birth
##  Min.   :  1.810       Min.   :      464   Min.   :49.44
##  1st Qu.:  6.185       1st Qu.:    86400   1st Qu.:67.00
##  Median : 13.985       Median :   716400   Median :74.36
##  Mean   : 24.484       Mean   :  8311771   Mean   :71.83
##  3rd Qu.: 38.655       3rd Qu.:  4200000   3rd Qu.:78.29
##  Max.   :117.230       Max.   :389000000   Max.   :89.57
##  NA's   :35            NA's   :46          NA's   :35
##  maternal_mortality_rate net_migration_rate    population
##  Min.   :   2.0          Min.   :-113.5100   Min.   :4.800e+01
##  1st Qu.:  20.0          1st Qu.:  -2.0150   1st Qu.:3.266e+05
##  Median :  65.5          Median :  -0.0450   Median :5.220e+06
##  Mean   : 178.0          Mean   :  -0.1816   Mean   :3.229e+07
##  3rd Qu.: 240.0          3rd Qu.:   1.2575   3rd Qu.:1.826e+07
##  Max.   :2054.0          Max.   :  83.8200   Max.   :1.356e+09
##  NA's   :75              NA's   :37          NA's   :21
##  population_growth_rate
##  Min.   :-9.730
##  1st Qu.: 0.260
##  Median : 1.020
##  Mean   : 1.101
##  3rd Qu.: 1.920
##  Max.   : 9.370
##  NA's   :26
```

```
dim(cia_factbook) # prints dimensions of dataset
```

```
## [1] 259  11
```

Creating the Classification/Binary Variable

A stated in the beginning of the project, we want to observe if we can classify whether a country is third world or not. Unfortunately, the dataset does not come with a variable for this so we will have to code it ourselves.

We will create a vector list of all of the countries that have been classified as underdeveloped in 2014 according the the United Nations (The Least Developed Countries Report 2014 | Department of Economic and Social Affairs. (2014). Un.org. https://sdgs.un.org/publications/least-developed-countries-report-2014-17949). After that we will use the mutate function from the r package dplyr to create our target classification variable "third_world".

```
undeveloped_countries <- c(
  "Afghanistan", "Angola", "Bangladesh", "Benin", "Bhutan", "Burkina Faso",
"Burundi", "Cambodia",
  "Central African Republic", "Chad", "Comoros", "Democratic Republic of the
Congo", "Djibouti",
  "Equatorial Guinea", "Eritrea", "Ethiopia", "The Gambia", "Guinea",
"Guinea-Bissau", "Haiti",
  "Kiribati", "Lao People's Democratic Republic", "Lesotho", "Liberia",
"Madagascar", "Malawi",
  "Mali", "Mauritania", "Mozambique", "Myanmar", "Nepal", "Niger", "Rwanda",
"Sao Tome and Principe",
  "Senegal", "Sierra Leone", "Solomon Islands", "Somalia", "South Sudan",
"Sudan", "Timor-Leste",
  "Togo", "Tuvalu", "Uganda", "United Republic of Tanzania", "Vanuatu",
"Yemen", "Zambia"
) # creates a vector of a list of underdeveloped countries, derived from the
UN

cia_factbook <- cia_factbook %>% # loads mutate command into dataset
  mutate(third_world = ifelse(country %in% undeveloped_countries, # creates a
new binary variable of 3rd world status
                                     1,  # undeveloped country
                                     0)) # developed country
```

Handling Dataset NA's

```
# removes rows with more than 4 NA's
row_na_count <- rowSums(is.na(cia_factbook)) # counts the number of NA's in
each row
limit <- 4 # sets the threshold
cia_factbook <- cia_factbook[row_na_count <= limit, ] # removes rows with
more than 4 NA's
```

For the remaining NA's in the data we will use mean or median imputation depending if the data is skewed or normally distributed based on the shape from the histograms.The histograms with a normal distribution are net migration rate and population growth rate, so we will use mean imputation because the mean is not as influenced by outliers compared to the other values in the dataset.

```
cia_factbook$net_migration_rate[is.na(cia_factbook$net_migration_rate)] <-
  mean(cia_factbook$net_migration_rate, na.rm = TRUE)
```

```
cia_factbook$population_growth_rate[is.na(cia_factbook$population_growth_rate
)] <- mean(cia_factbook$net_migration_rate, na.rm = TRUE)

summary(cia_factbook) # prints summary to confirm there are no NA's in
normally distributed variables

##    country               area             birth_rate      death_rate    
##  Length:224         Min.   :       2   Min.   : 6.72   Min.   : 1.530  
##  Class :character   1st Qu.:    5836   1st Qu.:11.84   1st Qu.: 5.930  
##  Mode  :character   Median :   87971   Median :16.89   Median : 7.540  
##                     Mean   :  608449   Mean   :19.66   Mean   : 7.907  
##                     3rd Qu.:  448124   3rd Qu.:24.91   3rd Qu.: 9.457  
##                     Max.   :17098242   Max.   :46.12   Max.   :17.490  
##                                                                        
##  infant_mortality_rate internet_users      life_exp_at_birth
##  Min.   :  1.81        Min.   :      900   Min.   :49.44    
##  1st Qu.:  6.20        1st Qu.:    95000   1st Qu.:66.90    
##  Median : 14.00        Median :   746000   Median :74.29    
##  Mean   : 24.57        Mean   :  8470823   Mean   :71.76    
##  3rd Qu.: 38.70        3rd Qu.:  4393000   3rd Qu.:78.28    
##  Max.   :117.23        Max.   :389000000   Max.   :89.57    
##  NA's   :1             NA's   :15          NA's   :2        
##  maternal_mortality_rate net_migration_rate    population       
##  Min.   :   2.0          Min.   :-113.5100   Min.   :5.215e+03  
##  1st Qu.:  20.0          1st Qu.:  -2.0050   1st Qu.:5.843e+05  
##  Median :  65.5          Median :  -0.0700   Median :5.617e+06  
##  Mean   : 178.0          Mean   :  -0.1881   Mean   :3.202e+07  
##  3rd Qu.: 240.0          3rd Qu.:   1.2200   3rd Qu.:2.176e+07  
##  Max.   :2054.0          Max.   :  83.8200   Max.   :1.356e+09  
##  NA's   :40                                                     
##  population_growth_rate  third_world   
##  Min.   :-9.730         Min.   :0.000  
##  1st Qu.: 0.330         1st Qu.:0.000  
##  Median : 1.075         Median :0.000  
##  Mean   : 1.140         Mean   :0.192  
##  3rd Qu.: 1.923         3rd Qu.:0.000  
##  Max.   : 9.370         Max.   :1.000  
##
```

Looking at the summaries now, we can now see that there are no NA's for the normally distributed variables in the dataset. Now, let's do the remainder of the numerical variables but with their respective median value.

```
# median imputation

cia_factbook$net_migration_rate[is.na(cia_factbook$infant_mortality_rate)] <-
  median(cia_factbook$infant_mortality_rate, na.rm = TRUE)

cia_factbook$infant_mortality_rate[is.na(cia_factbook$infant_mortality_rate)]
<-
```

```
  median(cia_factbook$infant_mortality_rate, na.rm = TRUE)

cia_factbook$internet_users[is.na(cia_factbook$internet_users)] <-
  median(cia_factbook$internet_users, na.rm = TRUE)

cia_factbook$life_exp_at_birth[is.na(cia_factbook$life_exp_at_birth)] <-
  median(cia_factbook$life_exp_at_birth, na.rm = TRUE)

cia_factbook$maternal_mortality_rate[is.na(cia_factbook$maternal_mortality_ra
te)] <-
  median(cia_factbook$maternal_mortality_rate, na.rm = TRUE)

summary(cia_factbook) # prints summary to confirm there are no NA's

##    country               area            birth_rate      death_rate
##  Length:224         Min.   :       2   Min.   : 6.72   Min.   : 1.530
##  Class :character   1st Qu.:    5836   1st Qu.:11.84   1st Qu.: 5.930
##  Mode  :character   Median :   87971   Median :16.89   Median : 7.540
##                     Mean   :  608449   Mean   :19.66   Mean   : 7.907
##                     3rd Qu.:  448124   3rd Qu.:24.91   3rd Qu.: 9.457
##                     Max.   :17098242   Max.   :46.12   Max.   :17.490
##  infant_mortality_rate internet_users      life_exp_at_birth
##  Min.   :  1.810       Min.   :      900   Min.   :49.44
##  1st Qu.:  6.205       1st Qu.:   113150   1st Qu.:67.00
##  Median : 14.000       Median :   746000   Median :74.29
##  Mean   : 24.528       Mean   :  7953536   Mean   :71.78
##  3rd Qu.: 38.655       3rd Qu.:  4012750   3rd Qu.:78.25
##  Max.   :117.230       Max.   :389000000   Max.   :89.57
##  maternal_mortality_rate net_migration_rate    population
##  Min.   :   2.00         Min.   :-113.5100   Min.   :5.215e+03
##  1st Qu.:  26.75         1st Qu.:  -2.0050   1st Qu.:5.843e+05
##  Median :  65.50         Median :  -0.0550   Median :5.617e+06
##  Mean   : 157.89         Mean   :  -0.1248   Mean   :3.202e+07
##  3rd Qu.: 200.00         3rd Qu.:   1.2500   3rd Qu.:2.176e+07
##  Max.   :2054.00         Max.   :  83.8200   Max.   :1.356e+09
##  population_growth_rate  third_world
##  Min.   :-9.730         Min.   :0.000
##  1st Qu.: 0.330         1st Qu.:0.000
##  Median : 1.075         Median :0.000
##  Mean   : 1.140         Mean   :0.192
##  3rd Qu.: 1.923         3rd Qu.:0.000
##  Max.   : 9.370         Max.   :1.000
```

Now, there are no NA’s in the dataset. Let’s now analyze our target variable, net migration, in more detail.

Appendix A: Regression Modeling

Let’s start to create our regression model for net migration rate. We will use ggpairs and a summary of our full model to understand if there is a correlation between net migration and an variables in the dataset.

Correlation Matrix

```
cia_factbook_no_country <- cia_factbook %>%
  select(-country)
cormat <- round(cor(cia_factbook_no_country),2)
head(cormat)
```

```
##                         area birth_rate death_rate infant_mortality_rate
## area                    1.00      -0.04       0.09                 -0.01
## birth_rate             -0.04       1.00       0.15                  0.87
## death_rate              0.09       0.15       1.00                  0.36
## infant_mortality_rate -0.01       0.87       0.36                  1.00
## internet_users          0.57      -0.15       0.01                 -0.12
## life_exp_at_birth      -0.01      -0.83      -0.48                 -0.88
##                        internet_users life_exp_at_birth
maternal_mortality_rate
## area                             0.57             -0.01                   -
0.01
## birth_rate                      -0.15             -0.83
0.73
## death_rate                       0.01             -0.48
0.35
## infant_mortality_rate          -0.12             -0.88
0.78
## internet_users                   1.00              0.13                   -
0.11
## life_exp_at_birth                0.13              1.00                   -
0.67
##                        net_migration_rate population population_growth_rate
## area                                  0.03       0.46                  -0.02
## birth_rate                           -0.13      -0.02                   0.56
## death_rate                           -0.07       0.00                  -0.17
## infant_mortality_rate              -0.10       0.04                   0.45
## internet_users                        0.01       0.76                  -0.09
## life_exp_at_birth                     0.15      -0.02                  -0.35
##                        third_world
## area                        -0.06
## birth_rate                   0.70
## death_rate                   0.25
## infant_mortality_rate        0.72
## internet_users              -0.11
## life_exp_at_birth           -0.61
```

```
library(reshape2)
```

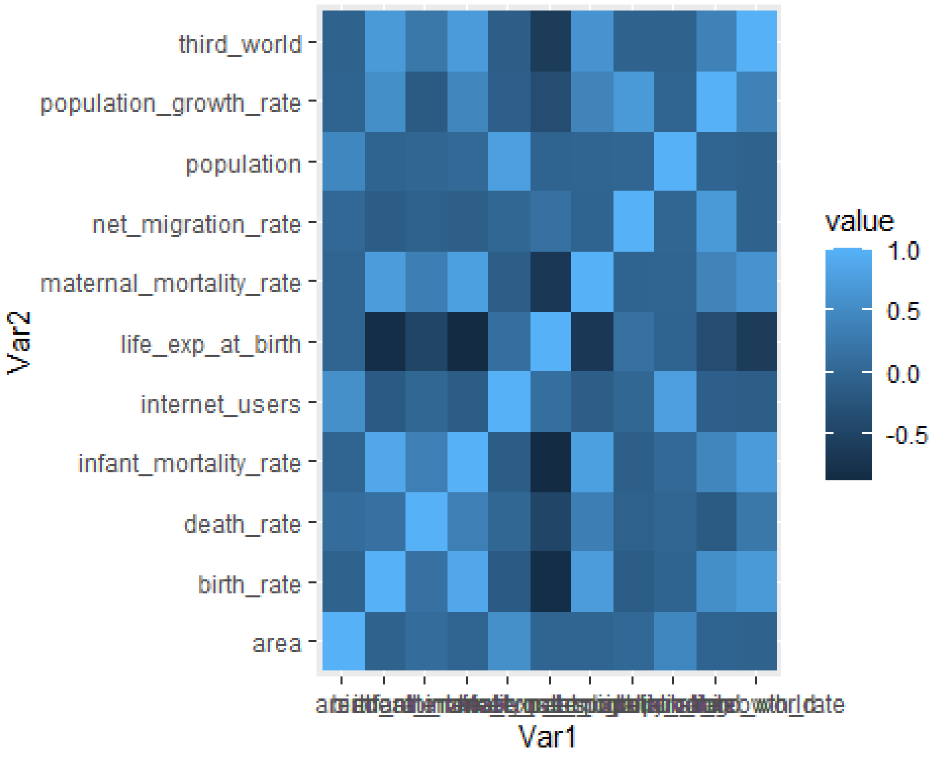
```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
melted_cormat <- melt(cormat)
head(melted_cormat)
```

```
##                     Var1 Var2 value
## 1                   area area  1.00
## 2             birth_rate area -0.04
## 3             death_rate area  0.09
## 4 infant_mortality_rate area -0.01
## 5         internet_users area  0.57
## 6     life_exp_at_birth area -0.01
```

```
library(ggplot2)
ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_raster()
```

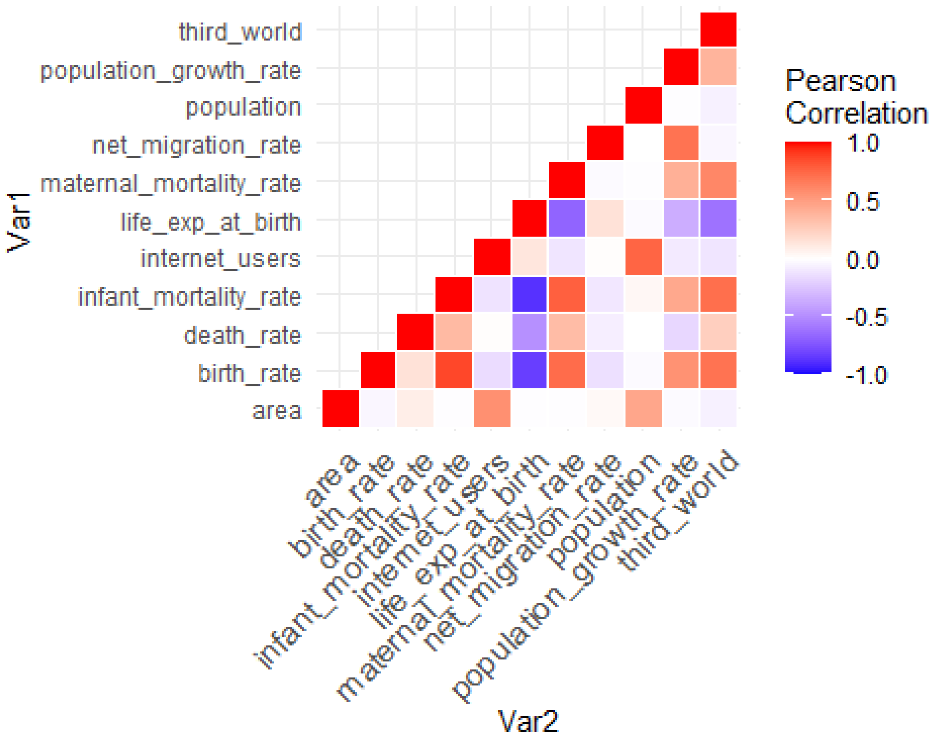


```
# Get lower triangle of the correlation matrix
  get_lower_tri<-function(cormat){
    cormat[upper.tri(cormat)] <- NA
    return(cormat)
  }
  # Get upper triangle of the correlation matrix
  get_upper_tri <- function(cormat){
    cormat[lower.tri(cormat)]<- NA
    return(cormat)
  }

  upper_tri <- get_upper_tri(cormat)
upper_tri
```

```
##                         area birth_rate death_rate infant_mortality_rate
## area                       1      -0.04       0.09                 -0.01
## birth_rate                NA       1.00       0.15                  0.87
## death_rate                NA         NA       1.00                  0.36
## infant_mortality_rate     NA         NA         NA                  1.00
## internet_users            NA         NA         NA                    NA
## life_exp_at_birth         NA         NA         NA                    NA
## maternal_mortality_rate   NA         NA         NA                    NA
## net_migration_rate        NA         NA         NA                    NA
## population                NA         NA         NA                    NA
## population_growth_rate    NA         NA         NA                    NA
## third_world               NA         NA         NA                    NA
##                         internet_users life_exp_at_birth
## area                              0.57             -0.01
## birth_rate                       -0.15             -0.83
## death_rate                        0.01             -0.48
## infant_mortality_rate            -0.12             -0.88
## internet_users                    1.00              0.13
## life_exp_at_birth                   NA              1.00
## maternal_mortality_rate             NA                NA
## net_migration_rate                  NA                NA
## population                          NA                NA
## population_growth_rate              NA                NA
## third_world                         NA                NA
##                         maternal_mortality_rate net_migration_rate
population
## area                                      -0.01               0.03
0.46
## birth_rate                                 0.73              -0.13      -
0.02
## death_rate                                 0.35              -0.07
0.00
## infant_mortality_rate                      0.78              -0.10
0.04
## internet_users                            -0.11               0.01
0.76
## life_exp_at_birth                         -0.67               0.15      -
0.02
## maternal_mortality_rate                    1.00              -0.02      -
0.01
## net_migration_rate                           NA               1.00
0.00
```

```
## population                                   NA                 NA
1.00
## population_growth_rate                       NA                 NA
NA
## third_world                                  NA                 NA
NA
##                         population_growth_rate third_world
## area                                     -0.02       -0.06
## birth_rate                                0.56        0.70
## death_rate                               -0.17        0.25
## infant_mortality_rate                     0.45        0.72
## internet_users                           -0.09       -0.11
## life_exp_at_birth                        -0.35       -0.61
## maternal_mortality_rate                   0.41        0.61
## net_migration_rate                        0.70       -0.04
## population                               -0.01       -0.06
## population_growth_rate                    1.00        0.39
## third_world                                 NA        1.00
```

```
# Melt the correlation matrix
library(reshape2)
melted_cormat <- melt(upper_tri, na.rm = TRUE)
# Heatmap
library(ggplot2)
ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
   name="Pearson\nCorrelation") +
  theme_minimal()+
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 12, hjust = 1))+
 coord_fixed()
```



We can observe there may be collinearlities with birth rate, death rate, infant mortality rate, life expectancy at birth, and maternal mortality. There may be a potential association between all of these variables. What is particularly interesting is that life expectancy at birth may have the strongest correlation out of all of the variables in the dataset as it has a single star next to it, which will be important when creating our regression model.

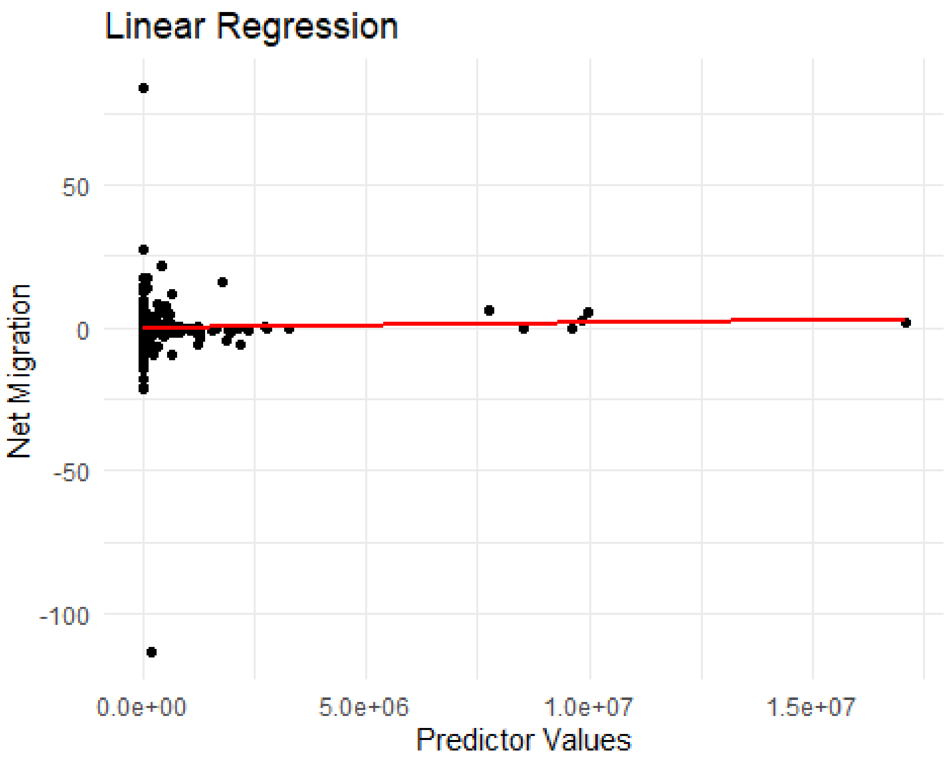
Now, let's move onto creating our full model to observe any statistically significant variables in our dataset.

Creating the Full Regression model

```
fullmodel <- lm(net_migration_rate ~ ., data = cia_factbook_no_country) #
creates full model
summary(fullmodel) # prints summary of the model

##
## Call:
## lm(formula = net_migration_rate ~ ., data = cia_factbook_no_country)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.322 -0.471 -0.222  0.028 34.272
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.421e+00  5.537e+00   0.257    0.798
## area                     -1.221e-08  1.288e-07  -0.095    0.925
## birth_rate              -9.966e-01  5.490e-02 -18.154   <2e-16 ***
## death_rate               9.303e-01  9.445e-02   9.849   <2e-16 ***
## infant_mortality_rate    1.615e-03  2.311e-02   0.070    0.944    
## internet_users          -2.219e-09  1.003e-08  -0.221    0.825    
## life_exp_at_birth       -3.200e-03  5.989e-02  -0.053    0.957    
## maternal_mortality_rate  6.532e-04  1.335e-03   0.489    0.625    
## population               2.611e-11  2.382e-09   0.011    0.991    
## population_growth_rate   9.437e+00  1.692e-01  55.763   <2e-16 ***
## third_world              2.136e-01  7.255e-01   0.294    0.769    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.817 on 213 degrees of freedom
## Multiple R-squared:  0.9389, Adjusted R-squared:  0.936 
## F-statistic:   327 on 10 and 213 DF,  p-value: < 2.2e-16
```

Looking at the statistics here, the full model is already excellent at predicting the values of the data judging by the high r-squared and extremely low p-value. Most of the variables do not pass the significance test, and according to the correlation matrix they don't necessarily correlate with the net migration rate.

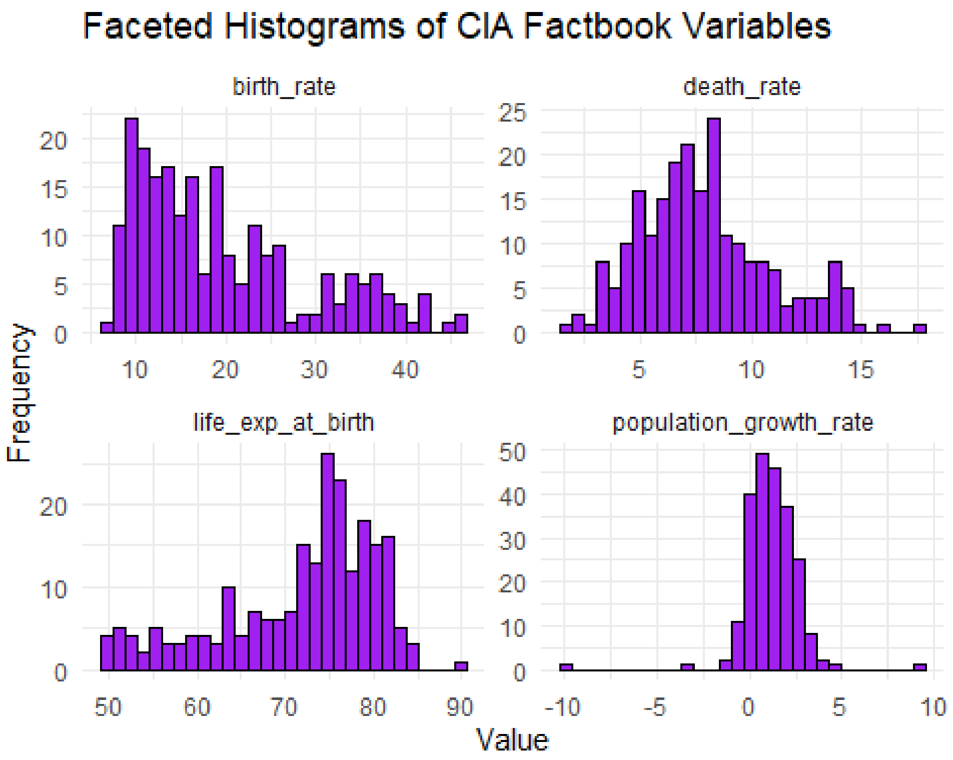
```
# creates linear regression plot

ggplot(cia_factbook_no_country, aes(x = area + birth_rate +
infant_mortality_rate, internet_users + life_exp_at_birth +
maternal_mortality_rate + population + population_rate + third_world, y =
net_migration_rate)) +
  geom_point(color = "black", size = 1.5) +
  geom_smooth(method = "lm", formula = y ~ x, color = "red", se = FALSE) +
  labs(title = "Linear Regression",
       x = "Predictor Values",
       y = "Net Migration") +
  theme_minimal()
```



Let us explore the possibility of using variables which correlate stronger or are more statistically significant produces a more robust model, with a high adjusted r-squared. We will use the variables of birth rate, death rate, life expectancy at birth, and population growth rate as they either have a strong correlation and/or are statistically significant.

First, let’s explore each variable individually.

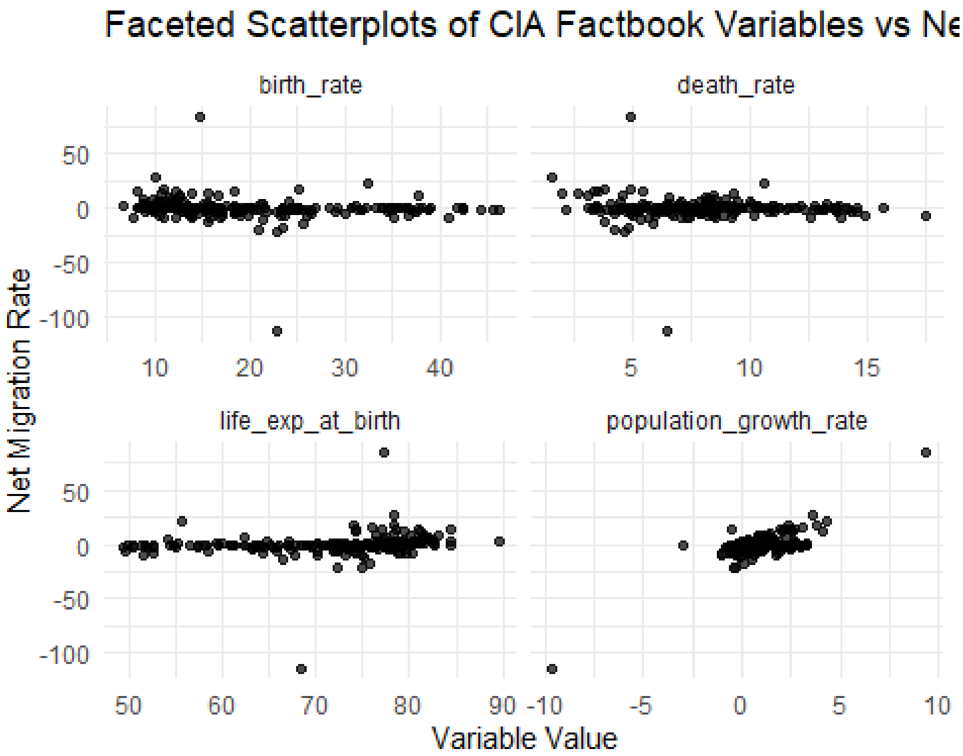
```
cia_factbook_long <- cia_factbook %>%
  pivot_longer(cols = c(birth_rate, death_rate, life_exp_at_birth,
                        population_growth_rate),
               names_to = "variable", values_to = "value")

# creates faceted histogram plot
ggplot(cia_factbook_long, aes(x = value)) +
  geom_histogram(bins = 30, fill = "purple", color = "black") +
  facet_wrap(~variable, scales = "free") + # facet by variable, allow
different scales
  theme_minimal() +
  labs(title = "Faceted Histograms of CIA Factbook Variables",
       x = "Value",
       y = "Frequency")
```



The majority of the variables we are looking at have a skewed spread. For birth rate, the data is skewed right which means that some countries have lots of babies being born but the vast majority is around 10-20%. The death rate is also skewed right and the majority of the data has a death rate of 5-10%, with some having death rates around 15%. Life expectancy at birth is skewed left, which makes sense considering that the numeric data is ordinal. The majority of countries have a life expectancy of 70-80 years old, which makes sense. The population growth rate is more bell shaped than the other variables but is still skewed right. The majority of the countries have a population growth rate around 1%.

Now let's see how each variable individually correlates with the net migration rate.

```r
cia_factbook_long <- cia_factbook %>%
  pivot_longer(cols = c(birth_rate, death_rate, life_exp_at_birth,
                        population_growth_rate),
             names_to = "variable", values_to = "x_value") # Renaming for
clarity

# Creating the faceted scatterplot
ggplot(cia_factbook_long, aes(x = x_value, y = net_migration_rate)) +
  geom_point(color = "black", alpha = 0.7) +
  facet_wrap(~variable, scales = "free_x") + # Facet by variable, allow free
x-scale
  theme_minimal() +
  labs(
    title = "Faceted Scatterplots of CIA Factbook Variables vs Net Migration
Rate",
    x = "Variable Value",
    y = "Net Migration Rate"
  )
```

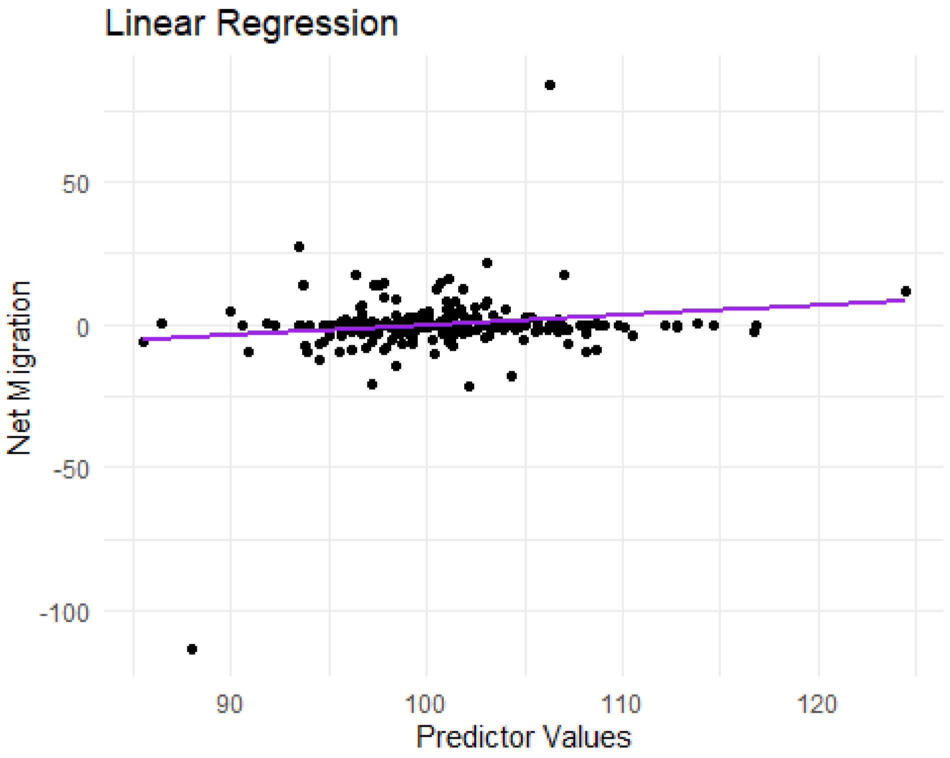


As shown above, there seems to be a very strong linear pattern between net migration and birth rate, life expectancy, and death rate. For population growth rate, there appears to be some bunchiness with the variables.

```
model1 <- lm(net_migration_rate ~ birth_rate + death_rate +
population_growth_rate + life_exp_at_birth, data = cia_factbook) # creates a
model with statistically significant values from the correlation matrix and
the full model summary
summary(model1) # provides a summary for model1

##
## Call:
## lm(formula = net_migration_rate ~ birth_rate + death_rate +
population_growth_rate +
##     life_exp_at_birth, data = cia_factbook)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.399 -0.453 -0.235  0.002 34.343
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.962298   4.868867   0.198    0.844
## birth_rate             -0.975848   0.044557 -21.901   <2e-16 ***
## death_rate              0.951361   0.085439  11.135   <2e-16 ***
## population_growth_rate  9.451968   0.165377  57.154   <2e-16 ***
## life_exp_at_birth      -0.002834   0.051298  -0.055    0.956    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.782 on 219 degrees of freedom
## Multiple R-squared:  0.9387, Adjusted R-squared:  0.9375 
## F-statistic: 837.7 on 4 and 219 DF,  p-value: < 2.2e-16
```

This version of the model is slightly more robust as the adjusted r-squared has increased by .0015. It is also just as statistically significant as the original model looking at the p-value of 2.2e-16.

```
ggplot(cia_factbook, aes(x = birth_rate + death_rate + population_growth_rate
+ life_exp_at_birth, y = net_migration_rate)) +
  geom_point(color = "black", size = 1.5) +
  geom_smooth(method = "lm", formula = y ~ x, color = "purple", se = FALSE) +
  labs(title = "Linear Regression",
       x = "Predictor Values",
       y = "Net Migration") +
  theme_minimal()
```



Is this model the best model we can get for our target variable? To confirm this we will use the step() function.

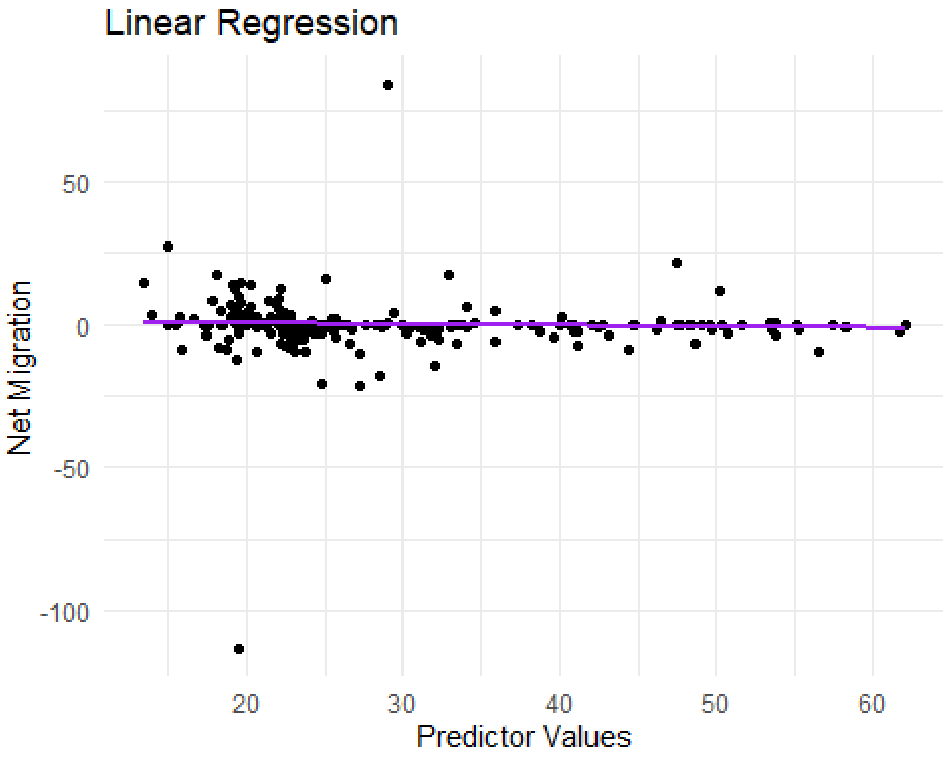
```r
model1 <- step(fullmodel) # uses stepwise function

## Start:  AIC=474.67
## net_migration_rate ~ area + birth_rate + death_rate + 
infant_mortality_rate + 
##     internet_users + life_exp_at_birth + maternal_mortality_rate + 
##     population + population_growth_rate + third_world
## 
##                           Df Sum of Sq     RSS     AIC
## - population               1       0.0  1690.0  472.67
## - life_exp_at_birth        1       0.0  1690.0  472.67
## - infant_mortality_rate    1       0.0  1690.0  472.67
## - area                     1       0.1  1690.1  472.68
## - internet_users           1       0.4  1690.4  472.72
## - third_world              1       0.7  1690.7  472.76
## - maternal_mortality_rate  1       1.9  1691.9  472.92
## <none>                                  1690.0  474.67
## - death_rate               1     769.6  2459.6  556.73
## - birth_rate               1    2614.9  4304.9  682.11
## - population_growth_rate   1   24671.3 26361.3 1088.03
## 
## Step:  AIC=472.67
## net_migration_rate ~ area + birth_rate + death_rate + 
infant_mortality_rate + 
##     internet_users + life_exp_at_birth + maternal_mortality_rate + 
##     population_growth_rate + third_world
## 
##                           Df Sum of Sq     RSS     AIC
## - life_exp_at_birth        1       0.0  1690.0  470.67
## - infant_mortality_rate    1       0.0  1690.0  470.67
## - area                     1       0.1  1690.1  470.68
## - third_world              1       0.7  1690.7  470.76
## - internet_users           1       0.7  1690.7  470.76
## - maternal_mortality_rate  1       1.9  1691.9  470.92
## <none>                                  1690.0  472.67
## - death_rate               1     786.5  2476.5  556.26
## - birth_rate               1    2648.7  4338.6  681.86
## - population_growth_rate   1   24671.3 26361.3 1086.03
## 
## Step:  AIC=470.67
## net_migration_rate ~ area + birth_rate + death_rate + 
infant_mortality_rate + 
##     internet_users + maternal_mortality_rate + population_growth_rate + 
##     third_world
## 
##                           Df Sum of Sq     RSS     AIC
## - area                     1       0.1  1690.1  468.68
## - infant_mortality_rate    1       0.1  1690.1  468.68
## - third_world              1       0.7  1690.7  468.76
## - internet_users           1       0.7  1690.7  468.77
## - maternal_mortality_rate  1       1.9  1691.9  468.92
## <none>                                 1690.0  470.67
## - death_rate               1    1117.7  2807.7  582.38
## - birth_rate               1    3675.1  5365.1  727.43
## - population_growth_rate   1   24793.5 26483.5 1085.07
##
## Step:  AIC=468.68
## net_migration_rate ~ birth_rate + death_rate + infant_mortality_rate +
##     internet_users + maternal_mortality_rate + population_growth_rate +
##     third_world
##
##                           Df Sum of Sq     RSS     AIC
## - infant_mortality_rate    1       0.1  1690.2  466.69
## - third_world              1       0.7  1690.8  466.77
## - internet_users           1       1.5  1691.6  466.88
## - maternal_mortality_rate  1       1.9  1692.0  466.93
## <none>                                 1690.1  468.68
## - death_rate               1    1129.0  2819.1  581.28
## - birth_rate               1    3681.1  5371.2  725.68
## - population_growth_rate   1   24850.8 26540.9 1083.55
##
## Step:  AIC=466.69
## net_migration_rate ~ birth_rate + death_rate + internet_users +
##     maternal_mortality_rate + population_growth_rate + third_world
##
##                           Df Sum of Sq     RSS     AIC
## - third_world              1       0.9  1691.1  464.81
## - internet_users           1       1.5  1691.7  464.89
## - maternal_mortality_rate  1       2.4  1692.6  465.01
## <none>                                 1690.2  466.69
## - death_rate               1    1292.8  2983.0  591.94
## - birth_rate               1    6441.5  8131.7  816.58
## - population_growth_rate   1   24854.0 26544.2 1081.58
##
## Step:  AIC=464.81
## net_migration_rate ~ birth_rate + death_rate + internet_users +
##     maternal_mortality_rate + population_growth_rate
##
##                           Df Sum of Sq     RSS     AIC
## - internet_users           1       1.5  1692.6  463.01
## - maternal_mortality_rate  1       2.9  1694.0  463.19
## <none>                                 1691.1  464.81
## - death_rate               1    1334.6  3025.7  593.13
## - birth_rate               1    7844.9  9536.0  850.26
## - population_growth_rate   1   24956.2 26647.2 1080.45
##
## Step:  AIC=463.01
## net_migration_rate ~ birth_rate + death_rate + maternal_mortality_rate +
##     population_growth_rate
##
```

```
##                           Df Sum of Sq     RSS     AIC
## - maternal_mortality_rate  1       3.0  1695.5  461.40
## <none>                                  1692.6  463.01
## - death_rate               1    1333.1  3025.7  591.13
## - birth_rate               1    7896.2  9588.8  849.50
## - population_growth_rate   1   24954.7 26647.3 1078.45
## 
## Step:  AIC=461.4
## net_migration_rate ~ birth_rate + death_rate + population_growth_rate
## 
##                          Df Sum of Sq     RSS     AIC
## <none>                                 1695.5  461.40
## - death_rate              1    1607.3  3302.8  608.76
## - birth_rate              1   12397.6 14093.1  933.76
## - population_growth_rate  1   25406.4 27102.0 1080.24
```

The most optimal model to predict net migration rate contains the variables birth rate, death rate, and population growth rate. All of the values pass the significance test and the R-squared is slightly improved from the previous model. This means that this is the best model to predict the net migration rate.

```
summary(model1)

## 
## Call:
## lm(formula = net_migration_rate ~ birth_rate + death_rate + 
population_growth_rate, 
##     data = cia_factbook_no_country)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -6.395 -0.447 -0.234  0.013 34.341 
## 
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             0.69542    0.61311   1.134    0.258    
## birth_rate             -0.97379    0.02428 -40.108   <2e-16 ***
## death_rate              0.95434    0.06608  14.441   <2e-16 ***
## population_growth_rate  9.45134    0.16461  57.416   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.776 on 220 degrees of freedom
## Multiple R-squared:  0.9387, Adjusted R-squared:  0.9378 
## F-statistic:  1122 on 3 and 220 DF,  p-value: < 2.2e-16
```

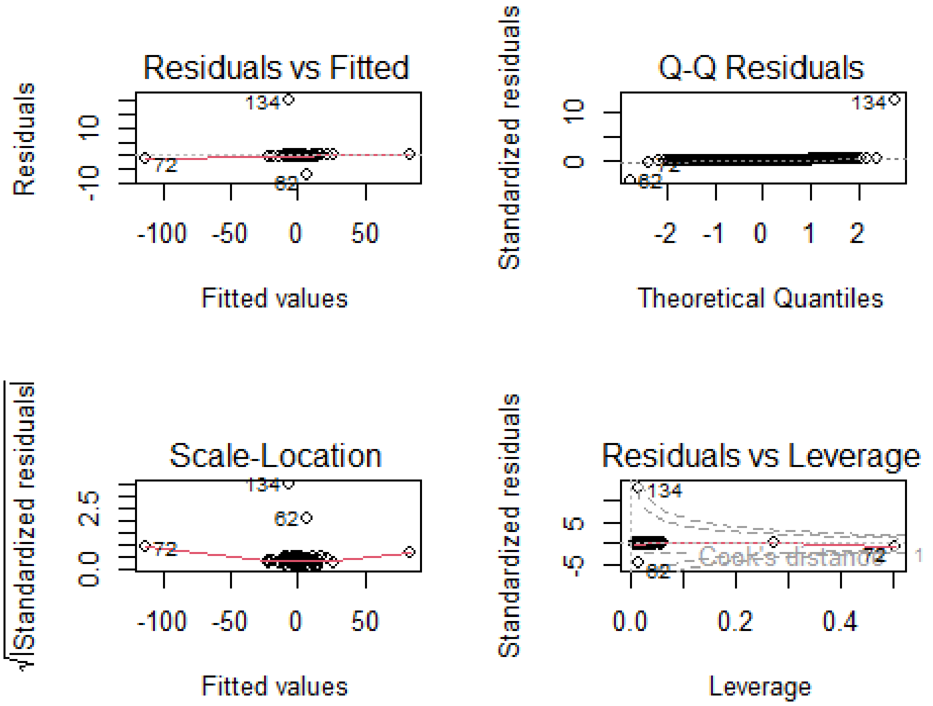
```
ggplot(cia_factbook, aes(x = birth_rate + death_rate + population_growth_rate
, y = net_migration_rate)) +
  geom_point(color = "black", size = 1.5) +
  geom_smooth(method = "lm", formula = y ~ x, color = "purple", se = FALSE) +
  labs(title = "Linear Regression",
       x = "Predictor Values",
       y = "Net Migration") +
  theme_minimal()
```



As shown, the values fit slightly better on the plot. Let’s move onto training our model.

```
#Creating a train/test partition using random sampling.

set.seed(1231)#use a seed for reproducibility

#use 80% of dataset as training set and 20% as test set
sample <- sample(c(TRUE, FALSE), size = nrow(cia_factbook_no_country),
replace = TRUE, prob = c(0.8, 0.2))
ciaTrain  <- cia_factbook_no_country[sample, ]
ciaTest   <- cia_factbook_no_country[!sample, ]

fullTrain = lm(net_migration_rate ~ birth_rate + death_rate +
population_growth_rate, data = ciaTrain) #Note selection of dataset
names(fullTrain) #reminder of all the elements contained in this object

##  [1] "coefficients" "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"

summary(fullTrain)#Print the model summary
```

```
##
## Call:
## lm(formula = net_migration_rate ~ birth_rate + death_rate +
population_growth_rate,
##     data = ciaTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7170 -0.1983 -0.0735  0.0474 19.8829
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.24254    0.40562   0.598    0.551
## birth_rate             -1.00936    0.01504 -67.114   <2e-16 ***
## death_rate              1.01357    0.04370  23.193   <2e-16 ***
## population_growth_rate  9.92339    0.09884 100.398   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.59 on 177 degrees of freedom
## Multiple R-squared:  0.9831, Adjusted R-squared:  0.9828
## F-statistic:  3424 on 3 and 177 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2)) #format the plot
plot(fullTrain) #print the plots of the main diagnostics for your model fit
to the training data
```



The diagnostic plots show some issues with the regression model. Outliers, especially observations 134, 62, and 72, stand out in multiple plots. The residuals may not have constant variance (heteroscedasticity), and there are slight deviations from normality. Observation 134 also has high leverage, indicating it might strongly influence the model. These findings suggest the need to address outliers and check the model's assumptions.

However, these outliers may originate from the Syria and Lebanon outliers from the original dataset so we should take that into account.

```
predFull<-predict(object = fullTrain,     # The regression model fit with
training data
        newdata = ciaTest)            #  creates a new dataframe with the test
data

summary(model1)$coefficients # prints coefficients

##                          Estimate Std. Error   t value      Pr(>|t|)
## (Intercept)             0.6954198 0.61310965   1.13425  2.579239e-01
## birth_rate             -0.9737851 0.02427918 -40.10783 3.892621e-103
## death_rate              0.9543434 0.06608360  14.44146  1.072226e-33
## population_growth_rate  9.4513397 0.16461203  57.41585 2.174566e-134

fullRMSE <- RMSE(predFull, ciaTest$net_migration_rate) ## calculates the RMSE
and R2
c(RMSE = fullRMSE, R2=summary(fullTrain)$r.squared)
```

```
##      RMSE        R2
## 5.5339733 0.9830616
```

The RMSE of the model is 5.5, which indicates there is a relatively large error meaning that the model does make incorrect predictions. Considering the migration rate's IQR is from around -2 to 1, the RMSE is high. However, the R-squared value of 0.983 means that 98.3% of the variation in the target variable is explained by the predictors of the model. This suggests an excellent fit!

Appendix A: Regression Models

We have already reviewed the performance of the first model and have used stepwise selection to optimize the baseline linear model. But what if there was a more efficient version of the model? Let's experiment with a polynomial regression to understand if it is a more efficient model.

```
polymodel <- lm(net_migration_rate ~ poly(birth_rate, 2) + poly(death_rate,
3) + poly(population_growth_rate, 4), data = cia_factbook_no_country)

summary(polymodel)
```

```
##
## Call:
## lm(formula = net_migration_rate ~ poly(birth_rate, 2) + poly(death_rate,
##     3) + poly(population_growth_rate, 4), data = cia_factbook_no_country)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7209 -0.5379  0.1748  0.4881 17.1008
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -1.248e-01  1.432e-01  -0.871   0.3845
## poly(birth_rate, 2)1             -1.356e+02  4.181e+00 -32.442  < 2e-16
***
## poly(birth_rate, 2)2             -5.218e+00  2.928e+00  -1.782   0.0762 .
## poly(death_rate, 3)1              3.598e+01  3.098e+00  11.612  < 2e-16
***
## poly(death_rate, 3)2             -3.837e+00  2.221e+00  -1.727   0.0855 .
## poly(death_rate, 3)3              7.659e-01  2.427e+00   0.316   0.7527
## poly(population_growth_rate, 4)1  1.986e+02  3.502e+00  56.722  < 2e-16
***
## poly(population_growth_rate, 4)2  7.282e-03  2.318e+00   0.003   0.9975
## poly(population_growth_rate, 4)3  1.409e+01  3.092e+00   4.557  8.7e-06
***
## poly(population_growth_rate, 4)4 -2.533e+01  2.460e+00 -10.296  < 2e-16
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.144 on 214 degrees of freedom
## Multiple R-squared:  0.9644, Adjusted R-squared:  0.9629
## F-statistic: 644.4 on 9 and 214 DF,  p-value: < 2.2e-16
```

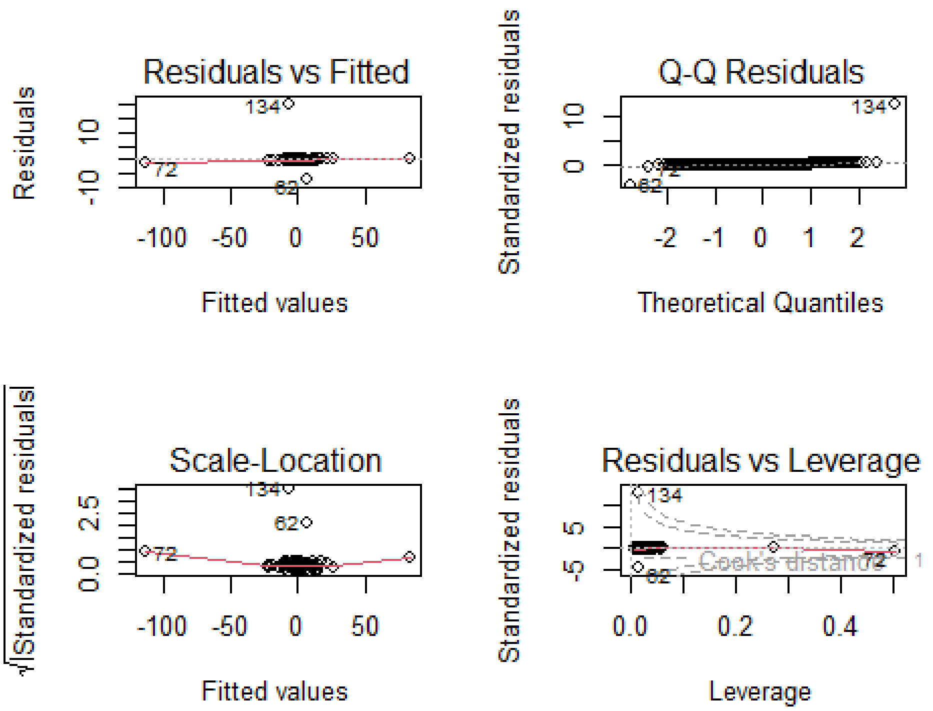
```
#Creating a train/test partition using random sampling.

set.seed(1234)#use a seed for reproducibility

#use 80% of dataset as training set and 20% as test set
sample <- sample(c(TRUE, FALSE), size = nrow(cia_factbook_no_country),
replace = TRUE, prob = c(0.8, 0.2))
ciaTrain1  <- cia_factbook_no_country[sample, ]
ciaTest1   <- cia_factbook_no_country[!sample, ]

fullTrain1 = lm(net_migration_rate ~ poly(birth_rate, 2) +
                     poly(death_rate, 3) +
                     poly(population_growth_rate, 4), data = ciaTrain)
names(fullTrain1) #reminder of all the elements contained in this object
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
summary(fullTrain1)#Print the model summary
```

```
##
## Call:
## lm(formula = net_migration_rate ~ poly(birth_rate, 2) + poly(death_rate,
##     3) + poly(population_growth_rate, 4), data = ciaTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5818 -0.2644  0.0219  0.1410 19.3587
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       -0.01121    0.11870  -0.094    0.925
## poly(birth_rate, 2)1            -127.43155    3.17491 -40.137   <2e-16
***
## poly(birth_rate, 2)2               0.82118    2.22661   0.369    0.713
## poly(death_rate, 3)1              36.05353    2.41180  14.949   <2e-16
***
## poly(death_rate, 3)2              -0.46970    1.65920  -0.283    0.777
## poly(death_rate, 3)3              -1.12060    1.84451  -0.608    0.544
## poly(population_growth_rate, 4)1  196.83233    2.63696  74.644   <2e-16
***
## poly(population_growth_rate, 4)2  -1.11153    1.73454  -0.641    0.522
## poly(population_growth_rate, 4)3   3.96686    2.49693   1.589    0.114
## poly(population_growth_rate, 4)4  -1.76286    1.92037  -0.918    0.360
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.597 on 171 degrees of freedom
## Multiple R-squared:  0.9835, Adjusted R-squared:  0.9826
## F-statistic:  1132 on 9 and 171 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2)) #format the plot
plot(fullTrain) #print the plots of the main diagnostics for the model fit to
the training data
```



The diagnostic plots for the regression model revealed a few issues which are worth addressing. The Residuals vs Fitted plot exhibits slight heteroscedasticity, especially at higher fitted values, where the spread of residuals increases. The Q-Q plot highlights deviations from normality, with heavy tails at both ends and a handful of clear outliers, such as points 206 and 166. The Scale-Location plot reinforces that the variance is not consistent with the data, as the red line dips and then rises again. Finally, the Residuals vs Leverage plot highlights points 206, 166, and 89 as influential, with high leverage and residuals, and suggests that they may be interfering with the model too much. While the model performs well overall, we may need to look into the outliers and understand how to solve for these issues.

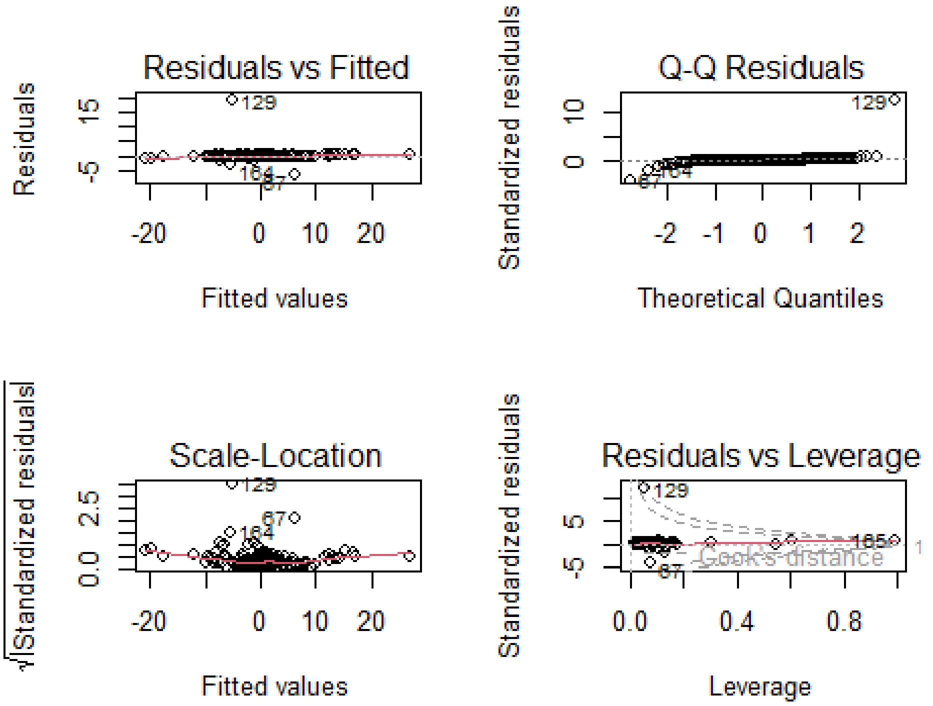
These outliers may have originated from the Lebanon and Syria situation, so we will test the model using data that doesn't contain the outliers. We will create a version of the data without the

```
cia_factbook_noutlier <- subset(cia_factbook, !(country %in% c("Lebanon",
"Syria"))) # removes lebanon and syria

cia_factbook_noutlier <- cia_factbook_noutlier %>% # removes categorical
variable
  select(-country)

summary(cia_factbook_noutlier)

##       area            birth_rate      death_rate     infant_mortality_rate
##  Min.   :       2   Min.   : 6.72   Min.   : 1.530   Min.   :  1.810
##  1st Qu.:    5789   1st Qu.:11.78   1st Qu.: 5.933   1st Qu.:  6.195
##  Median :   87971   Median :16.89   Median : 7.635   Median : 14.000
##  Mean   :  613050   Mean   :19.67   Mean   : 7.926   Mean   : 24.641
##  3rd Qu.:  449571   3rd Qu.:24.96   3rd Qu.: 9.473   3rd Qu.: 38.745
##  Max.   :17098242   Max.   :46.12   Max.   :17.490   Max.   :117.230
##  internet_users      life_exp_at_birth maternal_mortality_rate
##  Min.   :      900   Min.   :49.44     Min.   :   2.0
##  1st Qu.:   111050   1st Qu.:66.90     1st Qu.:  27.0
##  Median :   746000   Median :74.29     Median :  65.5
##  Mean   :  8000554   Mean   :71.78     Mean   : 158.9
##  3rd Qu.:  3872000   3rd Qu.:78.28     3rd Qu.: 200.0
##  Max.   :389000000   Max.   :89.57     Max.   :2054.0
##  net_migration_rate    population          population_growth_rate
##  Min.   :-21.64000   Min.   :5.215e+03   Min.   :-3.000
##  1st Qu.: -1.98250   1st Qu.:5.770e+05   1st Qu.: 0.330
##  Median : -0.05500   Median :5.587e+06   Median : 1.075
##  Mean   :  0.00781   Mean   :3.220e+07   Mean   : 1.152
##  3rd Qu.:  1.22000   3rd Qu.:2.183e+07   3rd Qu.: 1.920
##  Max.   : 27.35000   Max.   :1.356e+09   Max.   : 4.360
##   third_world
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.1937
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

Note the change of the interquartile range for the net migration rate.

Now let’s try and train the polynomial data based on the cia_factbook dataset without the outliers. We will also test our baseline model with the new data for a more accurate prediction between the two models.

```
#Creating a train/test partition using random sampling.

set.seed(1234)#use a seed for reproducibility

#use 80% of dataset as training set and 20% as test set
sample <- sample(c(TRUE, FALSE), size = nrow(cia_factbook_noutlier), replace
= TRUE, prob = c(0.8, 0.2))
ciaTrain1  <- cia_factbook_noutlier[sample, ]
ciaTest1   <- cia_factbook_noutlier[!sample, ]

fullTrain1 = lm(net_migration_rate ~ poly(birth_rate, 2) + poly(death_rate,
3) + poly(population_growth_rate, 4), data = ciaTrain1)
names(fullTrain1) #reminder of all the elements contained in this object

##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"

summary(fullTrain1)#Print the model summary

##
## Call:
## lm(formula = net_migration_rate ~ poly(birth_rate, 2) + poly(death_rate,
##     3) + poly(population_growth_rate, 4), data = ciaTrain1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3246 -0.0978 -0.0090  0.0636 19.0591
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         0.1236     0.1189   1.039    0.300
## poly(birth_rate, 2)1             -128.4249     3.2394 -39.644  < 2e-16 ***
## poly(birth_rate, 2)2                1.5785     2.3664   0.667    0.506
## poly(death_rate, 3)1               37.6356     2.4021  15.668  < 2e-16 ***
## poly(death_rate, 3)2               -1.0804     1.7288  -0.625    0.533
## poly(death_rate, 3)3               -1.4247     1.8030  -0.790    0.431
## poly(population_growth_rate, 4)1  131.6824     3.1980  41.177  < 2e-16 ***
## poly(population_growth_rate, 4)2   24.5450     1.8467  13.292  < 2e-16 ***
## poly(population_growth_rate, 4)3  -21.4135     1.7870 -11.983  < 2e-16 ***
## poly(population_growth_rate, 4)4    8.4223     1.7938   4.695 5.47e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.596 on 170 degrees of freedom
## Multiple R-squared:  0.9325, Adjusted R-squared:  0.9289
## F-statistic: 260.8 on 9 and 170 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2)) #format the plot
plot(fullTrain1) #print the plots of the main diagnostics for the model fit
to the training data

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced


## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```

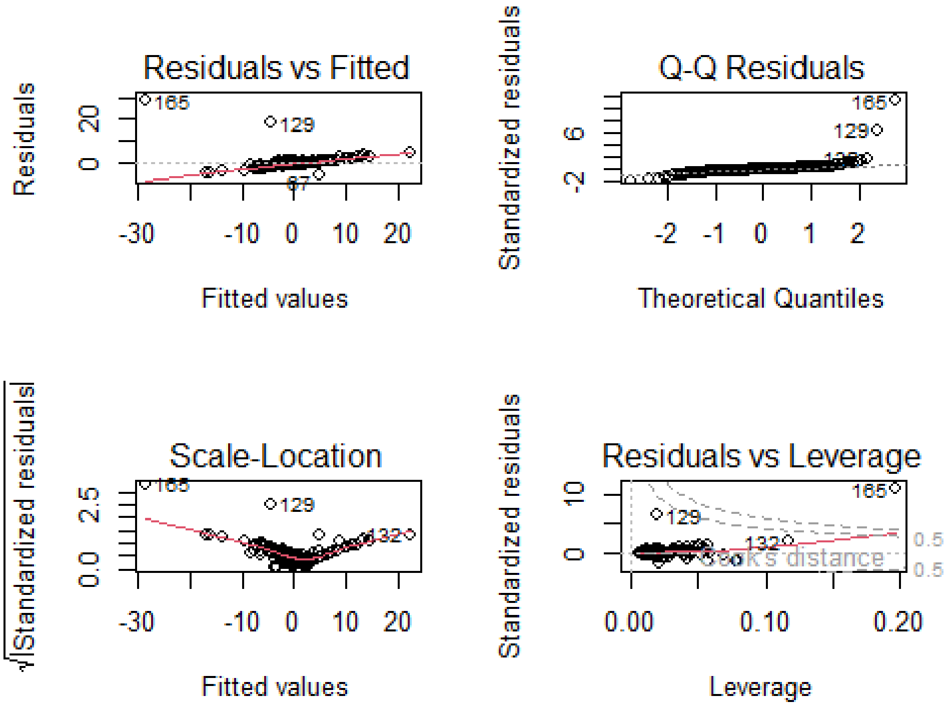


After removing Lebanon and Syria, the model looks way cleaner. The residuals are more evenly spread around the fitted values, which shows the predictions are more consistent. The Q-Q plot looks better too, with the points mostly following the normal line, except for a couple of slight deviations at the ends. The scale-location plot shows the variance is more stable, and the leverage plot doesn't have as many high-influence points anymore. Overall, the model feels more solid and reliable now.. Judging by the residuals vs fitted plot, we can observe that serveral datapoints on both dies of zero hae high residuals which indicate that the model may be way off with its predicitons. Let's confirm this by finding the RMSE and R-squared.

```
predFull1<-predict(object = fullTrain1,      # The regression model fit with
training data
        newdata = ciaTest1)           #  creates a new dataframe with the
test data

fullRMSE <- RMSE(predFull1, ciaTest1$net_migration_rate) ## calculates the
RMSE and R2
c(RMSE = fullRMSE, R2=summary(fullTrain1)$r.squared)

##      RMSE        R2
## 0.5604154 0.9324586
```

The RMSE of 0.56 is very small compared to the range and variability of the net migration rate. Additionally, the r-squared is excellent because it means that the model represents approximately 93 percent of the data which is a good sign that the model is accurate with its predictions. How does the linear regression model compare with the dataset without the outlieers though? Let’s take a look.

```
fullTrain2 = lm(net_migration_rate ~ birth_rate + death_rate +
population_growth_rate, data = ciaTrain1)
names(fullTrain2) #reminder of all the elements contained in this object

##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"

summary(fullTrain2)#Print the model summary

##
## Call:
## lm(formula = net_migration_rate ~ birth_rate + death_rate +
population_growth_rate,
##     data = ciaTrain1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1473 -0.7070 -0.3299  0.3824 28.3843
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.41480    0.70708   2.001   0.0469 *
## birth_rate             -0.83712    0.03751 -22.318   <2e-16 ***
## death_rate              0.76350    0.08456   9.029    3e-16 ***
## population_growth_rate  7.87897    0.33466  23.543   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.866 on 176 degrees of freedom
## Multiple R-squared:  0.7745, Adjusted R-squared:  0.7706
## F-statistic: 201.5 on 3 and 176 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2)) #format the plot
plot(fullTrain2) #print the plots of the main diagnostics for the model fit
to the training data
```



The diagnostic plots reveal key insights about the regression model. The Residuals vs Fitted plot shows some patterns, suggesting potential non-linearity in the data, which indicates the model might not fully capture the relationship between predictors and the response. The Q-Q Residuals plot shows slight deviations from the straight line, particularly at the tails, meaning the residuals are not perfectly normally distributed. The Scale-Location plot highlights heteroscedasticity, as the spread of residuals is not constant across fitted values. Lastly, the Residuals vs Leverage plot identifies influential points, such as 165 and 129, which may disproportionately affect the model. These issues suggest the model may benefit from further refinements, such as transformations or outlier handling.

```
predFull2<-predict(object = fullTrain2,     # The linear regression model fit with training data
        newdata = ciaTest1)

print("Linear Regression Model")

## [1] "Linear Regression Model"

fullRMSE <- RMSE(predFull2, ciaTest1$net_migration_rate) ## calculates the RMSE and R2
c(RMSE = fullRMSE, R2=summary(fullTrain)$r.squared)

##      RMSE        R2
## 1.3434074 0.9830616
```

The linear model's metrics exhibit an RMSE of 1.34, indicating that the average prediction error is relatively low. The R-Squared value suggests that the model can explain 93.3

percent of the variance in the response variable, meaning that the model is a very strong fit for the data. However, the impressive R-squared is impressive but the residual plots point to potential issues with non-linearity and influential points which may need to be addressed to improve the model and its performance. That is why we have the polynomial model which attempts to take these non-linear relationships into account.

Now let’s compare the linear model to the polynomial model.

Comparing the Models

```
summary(model1) # prints summary statistics of the linear model

## 
## Call:
## lm(formula = net_migration_rate ~ birth_rate + death_rate + 
population_growth_rate, 
##     data = cia_factbook_no_country)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -6.395 -0.447 -0.234  0.013 34.341 
## 
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)    
## (Intercept)              0.69542    0.61311   1.134    0.258    
## birth_rate              -0.97379    0.02428 -40.108   <2e-16 ***
## death_rate               0.95434    0.06608  14.441   <2e-16 ***
## population_growth_rate   9.45134    0.16461  57.416   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.776 on 220 degrees of freedom
## Multiple R-squared:  0.9387, Adjusted R-squared:  0.9378 
## F-statistic:  1122 on 3 and 220 DF,  p-value: < 2.2e-16

summary(polymodel) # prints summary statistics of the polynomial model

## 
## Call:
## lm(formula = net_migration_rate ~ poly(birth_rate, 2) + poly(death_rate, 
##     3) + poly(population_growth_rate, 4), data = cia_factbook_no_country)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -7.7209 -0.5379  0.1748  0.4881 17.1008 
## 
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                       -1.248e-01  1.432e-01  -0.871   0.3845    
## poly(birth_rate, 2)1              -1.356e+02  4.181e+00 -32.442  < 2e-16
```

```
***
## poly(birth_rate, 2)2             -5.218e+00  2.928e+00  -1.782   0.0762 .
## poly(death_rate, 3)1              3.598e+01  3.098e+00  11.612  < 2e-16
***
## poly(death_rate, 3)2             -3.837e+00  2.221e+00  -1.727   0.0855 .
## poly(death_rate, 3)3              7.659e-01  2.427e+00   0.316   0.7527
## poly(population_growth_rate, 4)1  1.986e+02  3.502e+00  56.722  < 2e-16
***
## poly(population_growth_rate, 4)2  7.282e-03  2.318e+00   0.003   0.9975
## poly(population_growth_rate, 4)3  1.409e+01  3.092e+00   4.557  8.7e-06
***
## poly(population_growth_rate, 4)4 -2.533e+01  2.460e+00 -10.296  < 2e-16
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.144 on 214 degrees of freedom
## Multiple R-squared:  0.9644, Adjusted R-squared:  0.9629
## F-statistic: 644.4 on 9 and 214 DF,  p-value: < 2.2e-16
```

The comparison between the two models demonstrates that the polynomial regression model fits the data better than the linear model. With an adjusted R-squared of 0.9629 and a residual standard error of 2.144, the polynomial model explains 96.44% of the variance in the net migration rate, while the linear model has an adjusted R-squared of 0.9378 and a residual standard error of 2.776. This indicates that the polynomial model more effectively captures the complexities of the relationships among birth rate, death rate, and population growth rate. Although the polynomial model offers greater accuracy, it is also more complex and may be harder to interpret. Thus, while the polynomial model is statistically superior, the decision between the two should consider whether predictive accuracy or model simplicity is more crucial for the analysis.

We will also compare the RMSE for both of the models additionally to compare which model has more accurate predicitons.

```
print("Linear Regression Model")

## [1] "Linear Regression Model"

fullRMSE <- RMSE(predFull2, ciaTest1$net_migration_rate) ## calculates the
RMSE and R2
c(RMSE = fullRMSE, R2=summary(fullTrain)$r.squared)

##      RMSE        R2
## 1.3434074 0.9830616

print("Polynomial Regression Model")

## [1] "Polynomial Regression Model"
```

```
fullRMSE <- RMSE(predFull1, ciaTest1$net_migration_rate) ## calculates the
RMSE and R2
c(RMSE = fullRMSE, R2=summary(fullTrain1)$r.squared)
```

```
##      RMSE        R2
## 0.5604154 0.9324586
```

The polynomial model seems to be superior to the linear regression model in terms of prediction accuracy with a lower RMSE, although the linear model has a slightly higher R-squared. This would mean that although the linear model fits the data well, it does not necessarily generalize as well as the polynomial model, which appears to capture underlying relationships between varaibles with fewer prediction errors. Therefore, for more accurate predicitions the polynomial model is the best model.