homicide_vs_unemployment

Mario Espinosa

2023-12-02

Introduction

Data science can have a positive impact on the world if we view it through the lens of data and the insights that the analysis and results provide.

The goal of this analysis is to verify the correlation and attempt to predict the homicide rate of a country based on the unemployment rate, GDP per capita, and GDP growth.

The datasets come from different sources, and the process of joining the different files can be challenging.

Sections: Merging data, Handling missing values, Analysis, modeling, and conclusions

List of the sources and links:

International Monetary Fund. Real GDP growth (Annual percent change)

International Monetary Fund. GDP per capita, current prices (Purchasing power parity; international dollars per capita)

ISO code for the International Monetary Fund (Json)

United Nations Office of Drugs and Crime. Victims of intentional murder

The World Bank. Unemployment, total (% of total labor force) (modeled ILO estimate)

Merging the Data:

Overall, this section combines data from various sources, cleans and reshapes it, and creates a merged dataset for further analysis. It specifically focuses on intentional homicide rates, unemployment rates, and GDP-related information for different countries from 2000 to 2020.

Please note that some countries in the dataset are the same nation but with distinctive separations. For example, The United Kingdom of Great Britain and Northern Ireland (UK) can appear in the dataset several times as a country with (England, Wales, and Scotland) and the northern part of the island of Ireland (Northern Ireland). That's why the length of the country list in all datasets is more than 200.

Each dataset covers different years. The amount of NA's only declines in the year 2000, and for the end year, some values are predictions, such as the GDP up to the year 2028. Therefore, we will use the year 2020 as the last one.

Real GDP Growth (Annual Percent Change)

```
## # A tibble: 5 x 50
##   Real G~1 `1980` `1981` `1982` `1983` `1984` `1985` `1986` `1987` `1988` `1989`
##   <chr>    <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>
## 1 <NA>     <NA>   <NA>   <NA>   <NA>   <NA>   <NA>   <NA>   <NA>   <NA>   <NA>
## 2 Afghani~ no da~ no da~ no da~ no da~ no da~ no da~ no da~ no da~ no da~ no da~
## 3 Albania  2.700~ 5.700~ 2.899~ 1.100~ 2      -1.5   5.599~ -0.80~ -1.39~ 9.800~
```

```
## 4 Algeria  -5.40~ 3      6.400~ 5.400~ 5.599~ 5.599~ -0.20~ -0.69~ -1.89~ 4.799~
## 5 Andorra  no da~ no da~ no da~ no da~ no da~ no da~ no da~ no da~ no da~ no da~
## # ... with 39 more variables: `1990` <chr>, `1991` <chr>, `1992` <chr>,
## #   `1993` <chr>, `1994` <chr>, `1995` <chr>, `1996` <chr>, `1997` <chr>,
## #   `1998` <chr>, `1999` <chr>, `2000` <chr>, `2001` <chr>, `2002` <chr>,
## #   `2003` <chr>, `2004` <chr>, `2005` <chr>, `2006` <chr>, `2007` <chr>,
## #   `2008` <chr>, `2009` <chr>, `2010` <chr>, `2011` <chr>, `2012` <chr>,
## #   `2013` <chr>, `2014` <chr>, `2015` <chr>, `2016` <chr>, `2017` <chr>,
## #   `2018` <chr>, `2019` <chr>, `2020` <chr>, `2021` <chr>, `2022` <chr>, ...
```

This CSV contains information about the GDP Annual percent change of 230 countries from the years 1980 to 2028. As seen in this dataset, information before 2000 tends to be incomplete. It makes sense that the data has the columns as years since there is only one indicator, but for further use, we reshaped the data.

```
## # A tibble: 5 x 3
##   Country      Year   GDP
##   <chr>       <dbl> <dbl>
## 1 Afghanistan  2000  NA
## 2 Albania      2000   6.9
## 3 Algeria      2000   3.8
## 4 Andorra      2000  NA
## 5 Angola       2000   3.1
```

International Monetary Fund. GDP per Capita, Current Prices (Purchasing Power Parity; International Dollars per Capita)

```
## # A tibble: 5 x 50
##   Country  `1980` `1981` `1982` `1983` `1984` `1985` `1986` `1987` `1988` `1989`
##   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 Afghani~    NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 2 Albania   2155.  2444.  2615.  2689.  2783.  2770.  2927.  2917.  2922.  3246.
## 3 Algeria   4808.  5257.  5755.  6103.  6469.  6722.  6664.  6607.  6515.  6923.
## 4 Andorra     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
## 5 Angola    1317.  1341.  1388.  1464.  1567.  1484.  1515.  1576.  1685.  1705.
## # ... with 39 more variables: `1990` <dbl>, `1991` <dbl>, `1992` <dbl>,
## #   `1993` <dbl>, `1994` <dbl>, `1995` <dbl>, `1996` <dbl>, `1997` <dbl>,
## #   `1998` <dbl>, `1999` <dbl>, `2000` <dbl>, `2001` <dbl>, `2002` <dbl>,
## #   `2003` <dbl>, `2004` <dbl>, `2005` <dbl>, `2006` <dbl>, `2007` <dbl>,
## #   `2008` <dbl>, `2009` <dbl>, `2010` <dbl>, `2011` <dbl>, `2012` <dbl>,
## #   `2013` <dbl>, `2014` <dbl>, `2015` <dbl>, `2016` <dbl>, `2017` <dbl>,
## #   `2018` <dbl>, `2019` <dbl>, `2020` <dbl>, `2021` <dbl>, `2022` <dbl>, ...
```

This dataset is similar to the last one, and the end result is the same with the obvious difference that this data is about GDP per capita. This CSV has information about 228 countries from the years 1980 to 2028.

```
## # A tibble: 5 x 3
##   Country      Year GDP_pc
##   <chr>       <dbl>  <dbl>
## 1 Afghanistan  2000    NA
## 2 Albania      2000  4326.
## 3 Algeria      2000  8588.
## 4 Andorra      2000    NA
## 5 Angola       2000  3272.
```

We can merge these two datasets with no problem.

```
gdp_merged <- left_join(gdp_pc_reshaped, imf_reshaped, by = c("Country", "Year"))
```

And we have the merged datasets.

```
## # A tibble: 6 x 4
##   Country              Year GDP_pc   GDP
##   <chr>               <dbl>  <dbl> <dbl>
## 1 Afghanistan          2000     NA    NA
## 2 Albania              2000  4326.   6.9
## 3 Algeria              2000  8588.   3.8
## 4 Andorra              2000     NA    NA
## 5 Angola               2000  3272.   3.1
## 6 Antigua and Barbuda  2000 16915.   6.2
```

Victims of Intentional Murder

```
## # A tibble: 6 x 13
##   `45092`   ...2     ...3  ...4  ...5  ...6  ...7  ...8  ...9  ...10 ...11 ...12
##   <chr>     <chr>    <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 Iso3_code Country  Regi~ Subr~ Indi~ Dime~ Cate~ Sex   Age   Year  Unit~ VALUE
## 2 ARM       Armenia  Asia  West~ Pers~ by c~ Nati~ Male  Total 2013  Coun~ 35
## 3 CHE       Switzer~ Euro~ West~ Pers~ by c~ Nati~ Male  Total 2013  Coun~ 28
## 4 COL       Colombia Amer~ Lati~ Pers~ by c~ Nati~ Male  Total 2013  Coun~ 15053
## 5 CZE       Czechia  Euro~ East~ Pers~ by c~ Nati~ Male  Total 2013  Coun~ 69
## 6 DEU       Germany  Euro~ West~ Pers~ by c~ Nati~ Male  Total 2013  Coun~ 455
## # ... with 1 more variable: ...13 <chr>
```

This dataset has information about the homicide rate of 204 countries from the years 1980 to 2020. It also contains several columns with more indicators. The indicators range from the source of data collection to the sex of victims of intentional murder. Selecting the relevant information for this analysis:

```
## # A tibble: 6 x 8
##   iso_code Country             Region   Subregion      Sex   Age    Year homic~1
##   <chr>    <chr>               <chr>    <chr>          <chr> <chr> <dbl> <chr>
## 1 AIA      Anguilla            Americas Latin America~ Total Total  2000 9.0518~
## 2 ALB      Albania             Europe   Southern Euro~ Total Total  2000 4.1168~
## 3 ARM      Armenia             Asia     Western Asia   Total Total  2000 2.8720~
## 4 ATG      Antigua and Barbuda Americas Latin America~ Total Total  2000 6.6617~
## 5 AUS      Australia           Oceania  Australia and~ Total Total  2000 1.9034~
## 6 AUT      Austria             Europe   Western Europe Total Total  2000 1.0236~
## # ... with abbreviated variable name 1: homicide_rate
```

This project aims to analyze the correlation between the homicide rate with other factors such as unemployment rate, GDP per capita, and GDP growth. The selection of indicators needs to focus on the rate per 100,000 population and the total number of murders, regardless of the sex or age of the victim and their relationship with the perpetrator.

Unemployment, Total % of Labor Force

```
##                   Country.Name Country.Code
## 1                        Aruba          ABW
## 2 Africa Eastern and Southern          AFE
## 3                  Afghanistan          AFG
##                                                               Indicator.Name
## 1 Unemployment, total (% of total labor force) (modeled ILO estimate)
## 2 Unemployment, total (% of total labor force) (modeled ILO estimate)
## 3 Unemployment, total (% of total labor force) (modeled ILO estimate)
##   Indicator.Code X1960 X1961 X1962 X1963 X1964 X1965 X1966 X1967 X1968 X1969
## 1 SL.UEM.TOTL.ZS    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## 2 SL.UEM.TOTL.ZS    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## 3 SL.UEM.TOTL.ZS    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
##   X1970 X1971 X1972 X1973 X1974 X1975 X1976 X1977 X1978 X1979 X1980 X1981 X1982
## 1    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## 2    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## 3    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
##   X1983 X1984 X1985 X1986 X1987 X1988 X1989 X1990    X1991    X1992    X1993
## 1    NA    NA    NA    NA    NA    NA    NA    NA       NA       NA       NA
## 2    NA    NA    NA    NA    NA    NA    NA    NA 7.333336 7.318747 7.242705
## 3    NA    NA    NA    NA    NA    NA    NA    NA 8.121000 8.168000 8.123000
##      X1994    X1995    X1996    X1997    X1998    X1999    X2000    X2001
## 1       NA       NA       NA       NA       NA       NA       NA       NA
## 2 7.160694 7.063796 7.055998 7.090541 7.060096 7.015271 6.939536 6.850376
## 3 8.111000 8.260000 8.165000 8.089000 8.082000 8.070000 8.054000 8.040000
##      X2002    X2003    X2004    X2005    X2006    X2007    X2008    X2009
## 1       NA       NA       NA       NA       NA       NA       NA       NA
## 2 6.803537 6.741241 6.535173 6.373503 6.347598 6.283421 6.232561 6.295587
## 3 8.186000 8.122000 8.053000 8.113000 8.054000 8.108000 8.022000 8.082000
##      X2010    X2011    X2012    X2013    X2014    X2015     X2016     X2017
## 1       NA       NA       NA       NA       NA       NA        NA        NA
## 2 6.933645 6.715358 6.599356 6.512784 6.555646 6.707142  6.855589  6.940365
## 3 8.068000 7.947000 8.019000 7.949000 7.910000 8.989000 10.086000 11.180000
##       X2018     X2019     X2020    X2021    X2022  X
## 1        NA        NA        NA       NA       NA NA
## 2  6.913046  7.121663  7.631304 7.920219 7.916835 NA
## 3 11.110000 11.085000 11.710000       NA       NA NA
```

This dataset contains information about unemployment in 266 countries from the years 1960 to 2022. It also contains the indicator and indicator code that are not relevant.

```
##                        Country iso_code X1960 X1961 X1962 X1963 X1964 X1965
## 1                        Aruba      ABW    NA    NA    NA    NA    NA    NA
## 2 Africa Eastern and Southern      AFE    NA    NA    NA    NA    NA    NA
## 3                  Afghanistan      AFG    NA    NA    NA    NA    NA    NA
##   X1966 X1967 X1968 X1969 X1970 X1971 X1972 X1973 X1974 X1975 X1976 X1977 X1978
## 1    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## 2    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## 3    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
##   X1979 X1980 X1981 X1982 X1983 X1984 X1985 X1986 X1987 X1988 X1989 X1990
## 1    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## 2    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## 3    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
##      X1991    X1992    X1993    X1994    X1995    X1996    X1997    X1998
## 1       NA       NA       NA       NA       NA       NA       NA       NA
```

```
## 2 7.333336 7.318747 7.242705 7.160694 7.063796 7.055998 7.090541 7.060096
## 3 8.121000 8.168000 8.123000 8.111000 8.260000 8.165000 8.089000 8.082000
##      X1999    X2000    X2001    X2002    X2003    X2004    X2005    X2006
## 1       NA       NA       NA       NA       NA       NA       NA       NA
## 2 7.015271 6.939536 6.850376 6.803537 6.741241 6.535173 6.373503 6.347598
## 3 8.070000 8.054000 8.040000 8.186000 8.122000 8.053000 8.113000 8.054000
##      X2007    X2008    X2009    X2010    X2011    X2012    X2013    X2014
## 1       NA       NA       NA       NA       NA       NA       NA       NA
## 2 6.283421 6.232561 6.295587 6.933645 6.715358 6.599356 6.512784 6.555646
## 3 8.108000 8.022000 8.082000 8.068000 7.947000 8.019000 7.949000 7.910000
##      X2015     X2016     X2017     X2018     X2019     X2020    X2021    X2022
## 1       NA        NA        NA        NA        NA        NA       NA       NA
## 2 6.707142  6.855589  6.940365  6.913046  7.121663  7.631304 7.920219 7.916835
## 3 8.989000 10.086000 11.180000 11.110000 11.085000 11.710000       NA       NA
##    X
## 1 NA
## 2 NA
## 3 NA
```

Since the datasets have discrepancies between the names of the countries, the ISO code is going to be the variable selected for the join. The same country with different names in each set, such as "Venezuela, RB," "Venezuela (Bolivarian Republic of)," and "Venezuela."

Using the ISO code to join the datasets:

The homicide and unemployment tables already have the ISO code, so we can merge them easily.

```r
homicide_unemployment <- left_join(homicide, unemployment_reshaped, by = c("iso_code", "Year"))
```

The new dataset looks like this:

```
## # A tibble: 6 x 9
##   iso_code Country             Region  Subre~1 Sex   Age    Year homic~2 unemp~3
##   <chr>    <chr>               <chr>   <chr>   <chr> <chr> <dbl> <chr>     <dbl>
## 1 AIA      Anguilla            Americ~ Latin ~ Total Total  2000 9.0518~   NA
## 2 ALB      Albania             Europe  Southe~ Total Total  2000 4.1168~   19.0
## 3 ARM      Armenia             Asia    Wester~ Total Total  2000 2.8720~   11.1
## 4 ATG      Antigua and Barbuda Americ~ Latin ~ Total Total  2000 6.6617~   NA
## 5 AUS      Australia           Oceania Austra~ Total Total  2000 1.9034~    6.28
## 6 AUT      Austria             Europe  Wester~ Total Total  2000 1.0236~    4.69
## # ... with abbreviated variable names 1: Subregion, 2: homicide_rate,
## #   3: unemployment_rate
```

Since the datasets from the International Monetary Fund do not have an ISO code, we need to add it to the sets using the data from the json file.

```
## # A tibble: 6 x 2
##   iso_code Country
##   <chr>    <chr>
## 1 ABW      Aruba
## 2 AFG      Afghanistan
## 3 AGO      Angola
## 4 AIA      Anguilla
## 5 ALB      Albania
## 6 ARE      United Arab Emirates
```

We join the JSON data to the GDP data, and finally, we have a nice file to work with.

```
final_merged_dataset <- left_join(homicide_unemployment, gdp_iso, by = c("iso_code", "Year"))
```

The final dataset looks like this:

```
## # A tibble: 6 x 8
##   Country             Region   Subregion Year       homic~1 unemp~2 GDP_pc   GDP
##   <chr>               <chr>    <chr>     <date>       <dbl>   <dbl>  <dbl> <dbl>
## 1 Anguilla            Americas Latin Am~ 2000-12-04    9.05   NA        NA   NA
## 2 Albania             Europe   Southern~ 2000-12-04    4.12   19.0   4326.   6.9
## 3 Armenia             Asia     Western ~ 2000-12-04    2.87   11.1   2606.   5.9
## 4 Antigua and Barbuda Americas Latin Am~ 2000-12-04    6.66   NA    16915.   6.2
## 5 Australia           Oceania  Australi~ 2000-12-04    1.90    6.28 28977.   3.1
## 6 Austria             Europe   Western ~ 2000-12-04    1.02    4.69 30875.   3.4
## # ... with abbreviated variable names 1: homicide_rate, 2: unemployment_rate
```

As we are going to handle missing values with the mean of the indicators, we split the data to preserve the original.

```
# Create the validation set to keep the integrity of the original data
# Set a seed for reproducibility
set.seed(123)

# Convert 'Year' to Date format
df$Year <- as.Date(as.character(df$Year), format = "%Y")

# Create a time-based data partition for the entire dataset
partition <- createDataPartition(df$homicide_rate, times = 1, p = 0.8, list = FALSE)

# Extract training and test sets
train_set <- df[partition, ]
test_set <- df[-partition, ]
```

The train set has the information from 2000 to 2018 and the test set has the remaining 2 years for the predictions

Handling missing values

In this section we address missing values through interpolation Checking countries with missing values

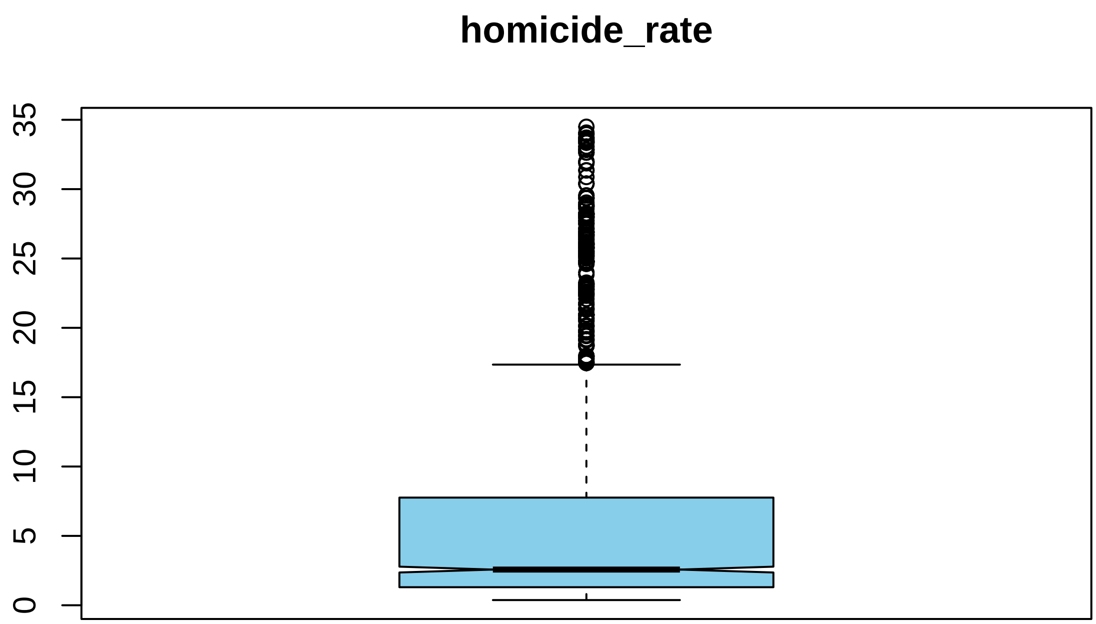
Let's begin by identifying countries with a significant number of missing values.

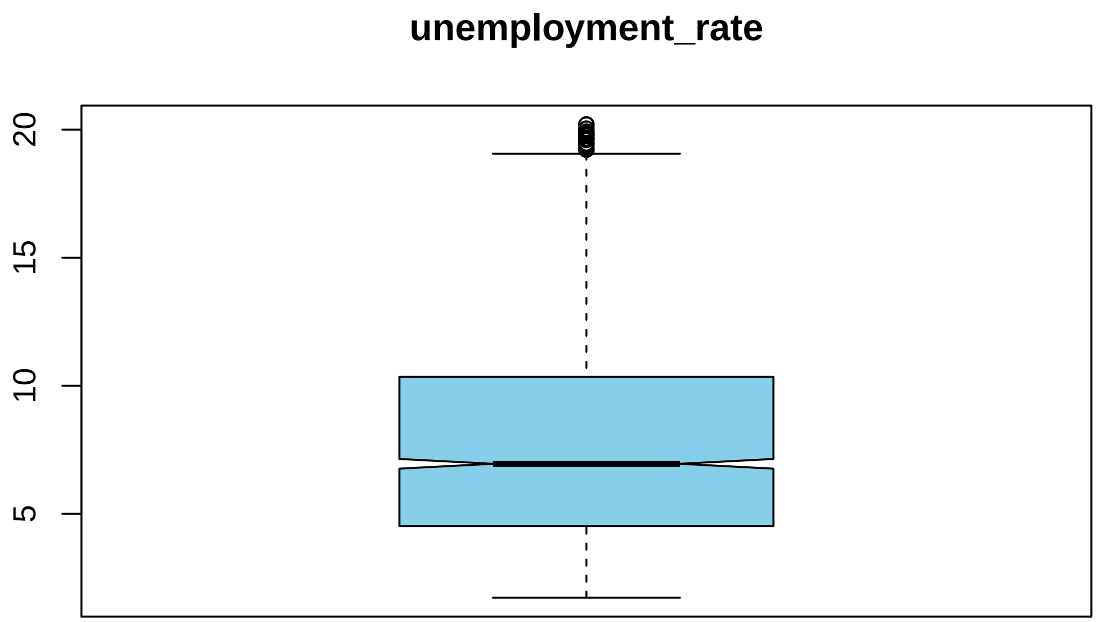
```
## # A tibble: 15 x 2
##    Variable MissingCount
##    <chr>    <chr>
##  1 Country  Zimbabwe
##  2 Country  Zambia
##  3 Country  Yemen
##  4 Country  Viet Nam
##  5 Country  Venezuela (Bolivarian Republic of)
##  6 Country  Uzbekistan
```

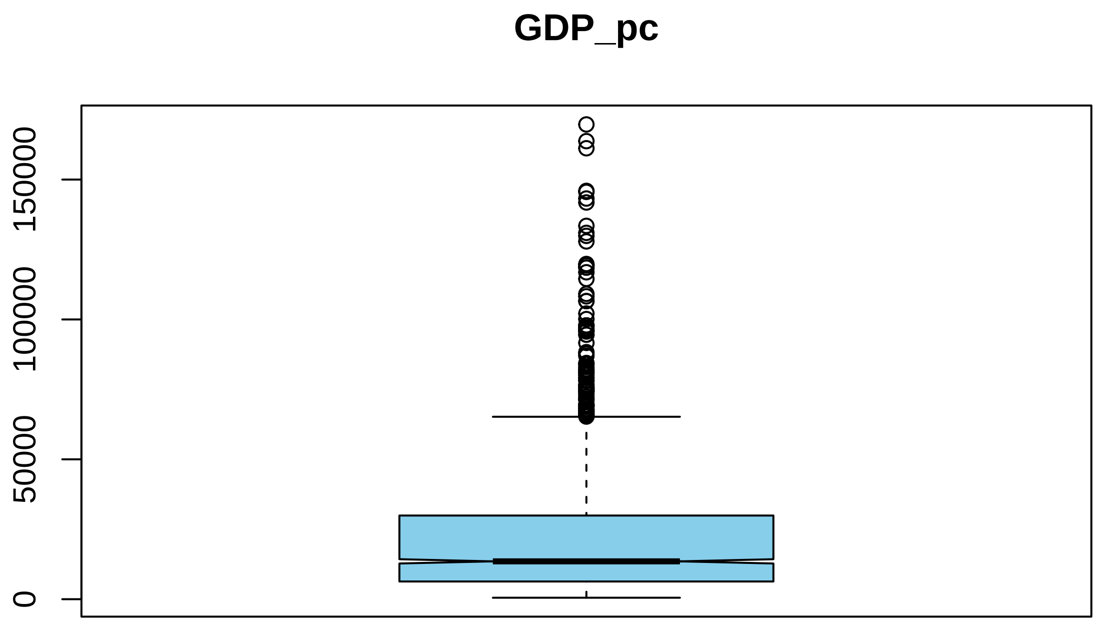
```
##  7 Country  Uruguay
##  8 Country  United States of America
##  9 Country  United States Virgin Islands
## 10 Country  United Republic of Tanzania
## 11 Country  United Kingdom of Great Britain and Northern Ireland
## 12 Country  United Kingdom (Scotland)
## 13 Country  United Kingdom (Northern Ireland)
## 14 Country  United Kingdom (England and Wales)
## 15 Country  United Arab Emirates
```

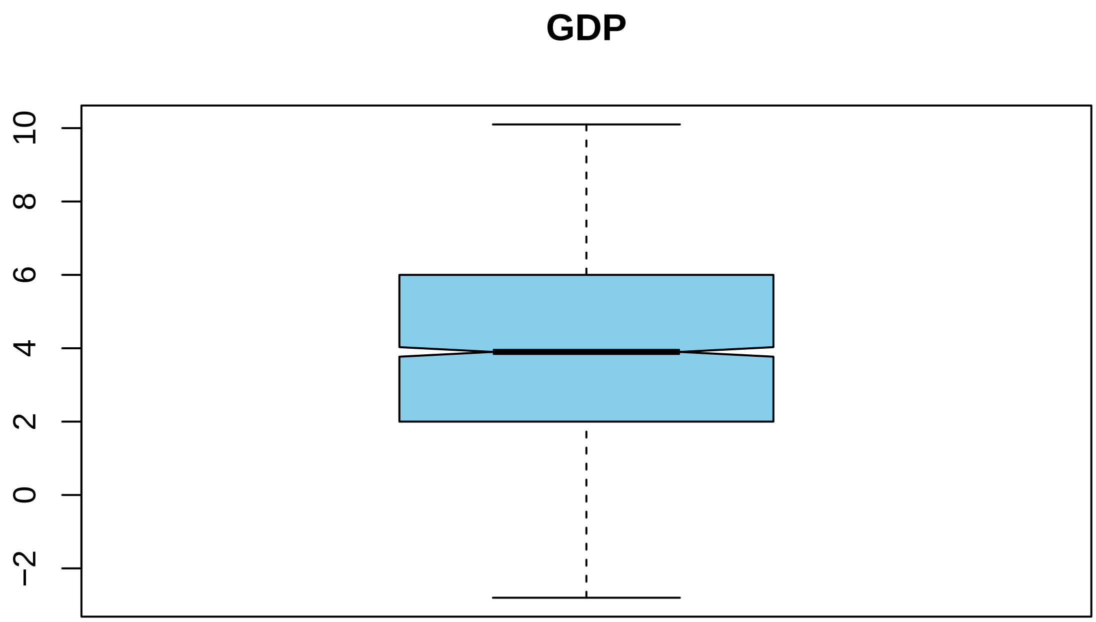
Notably, the UK regions exhibit a notable presence of missing values. This occurrence is attributed to certain datasets lacking information about these regions. However, our strategy involves imputing missing values rather than eliminating entire countries.

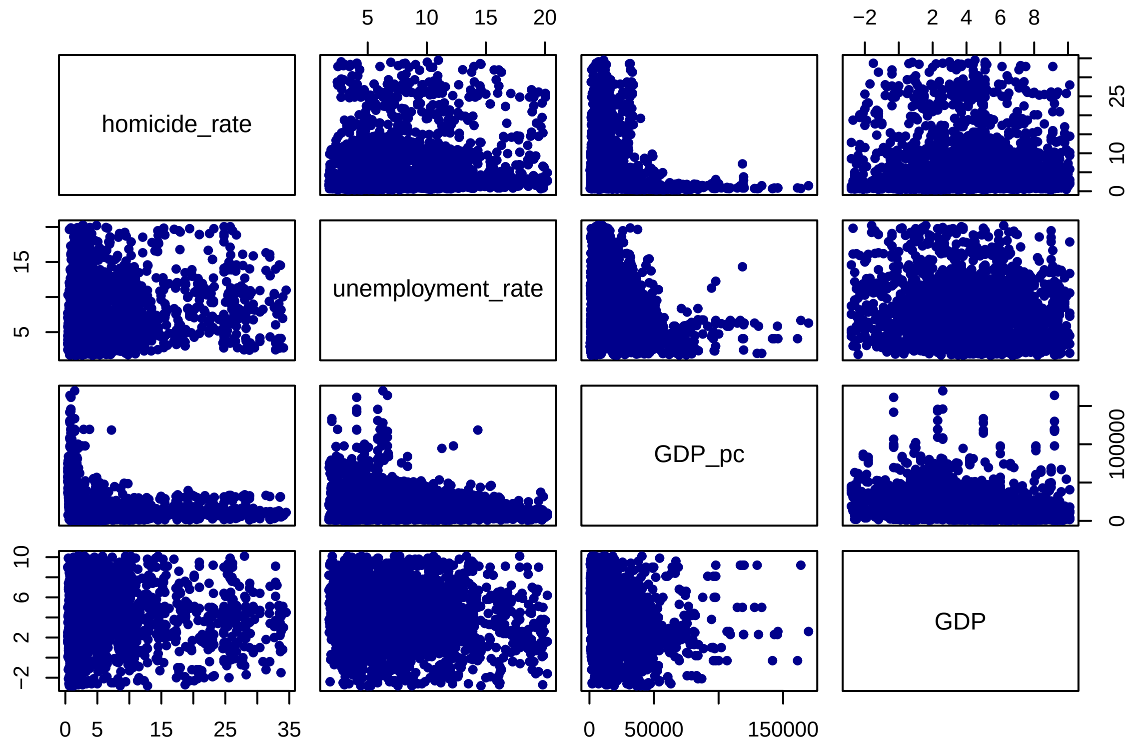
Before applying imputation methods, it's crucial to consider the outliers in our data.




unemployment_rate
20
15
10
5


GDP_pc
0
50000
100000
150000


GDP
10
8
6
4
2
0
−2



The boxplot illustrates a considerable skewness in all values. The criteria for outlier elimination are set at the 5th and 95th percentiles.

```
# Define conditions for outliers
outlier_conditions <-
  train_set$homicide_rate < quantile(train_set$homicide_rate, 0.05, na.rm = TRUE) |
  train_set$homicide_rate > quantile(train_set$homicide_rate, 0.95, na.rm = TRUE) |
  train_set$unemployment_rate < quantile(train_set$unemployment_rate, 0.05, na.rm = TRUE) |
  train_set$unemployment_rate > quantile(train_set$unemployment_rate, 0.95, na.rm = TRUE) |
  train_set$GDP < quantile(train_set$GDP, 0.05, na.rm = TRUE) |
  train_set$GDP > quantile(train_set$GDP, 0.95, na.rm = TRUE)

# Replace outlier values with NA
train_set <- train_set %>%
  mutate(
    homicide_rate = ifelse(outlier_conditions, NA, homicide_rate),
    unemployment_rate = ifelse(outlier_conditions, NA, unemployment_rate),
    GDP = ifelse(outlier_conditions, NA, GDP)
  )
```

With outliers addressed, we can employ the interpolation method. To ensure accuracy, we sort the data by date.

```
train_set <- train_set[order(train_set$Year), ]
```

Now, let's proceed with interpolation.

```r
# Create an imputation model
imputation_model <- mice(train_set[, c("homicide_rate", "unemployment_rate", "GDP_pc", "GDP")]
                         , method = "pmm")

# Impute missing values
imputed_data <- complete(imputation_model)

# Replace missing values in train_set with imputed values
train_set$homicide_rate[is.na(train_set$homicide_rate)] <- imputed_data$homicide_rate[is.na(train_set$ho
train_set$unemployment_rate[is.na(train_set$unemployment_rate)] <- imputed_data$unemployment_rate[is.na
train_set$GDP_pc[is.na(train_set$GDP_pc)] <- imputed_data$GDP_pc[is.na(train_set$GDP_pc)]
train_set$GDP[is.na(train_set$GDP)] <- imputed_data$GDP[is.na(train_set$GDP)]
```

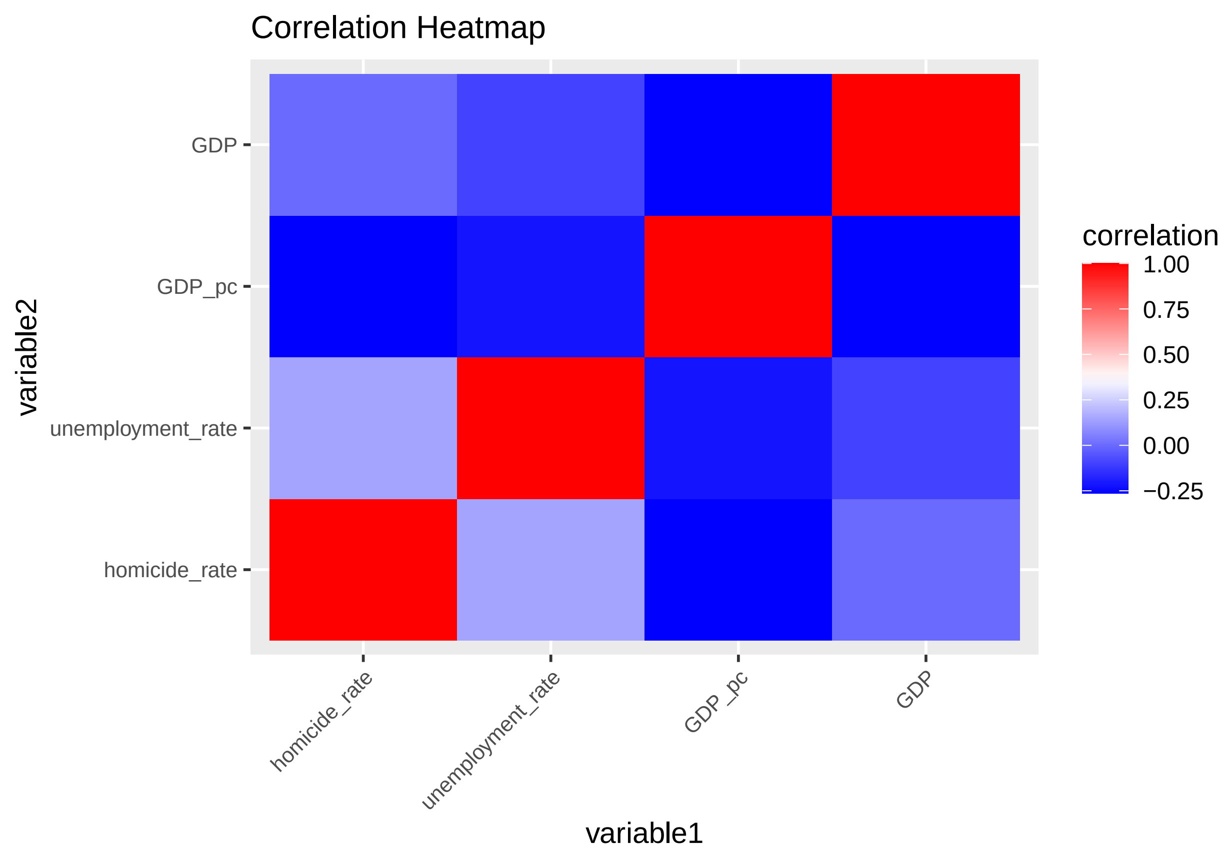
Now, with the missing values imputed we can continue but first lets eliminate the means are we no longer need them

```r
train_set <- train_set %>%
  select(Country, Region, Subregion,
         Year, homicide_rate,
         unemployment_rate, GDP_pc, GDP)
```

```
##    Country             Region            Subregion              Year
##  Length:2371        Length:2371        Length:2371        Min.   :2000-12-04
##  Class :character   Class :character   Class :character   1st Qu.:2004-12-04
##  Mode  :character   Mode  :character   Mode  :character   Median :2008-12-04
##                                                           Mean   :2008-11-13
##                                                           3rd Qu.:2012-12-04
##                                                           Max.   :2016-12-04
##  homicide_rate     unemployment_rate     GDP_pc             GDP
##  Min.   : 0.3732   Min.   : 1.722    Min.   :   532.2   Min.   :-2.800
##  1st Qu.: 1.3004   1st Qu.: 4.520    1st Qu.:  6339.5   1st Qu.: 2.000
##  Median : 2.5714   Median : 6.950    Median : 13532.2   Median : 3.900
##  Mean   : 5.9599   Mean   : 7.737    Mean   : 20635.4   Mean   : 3.979
##  3rd Qu.: 7.7620   3rd Qu.:10.350    3rd Qu.: 29914.7   3rd Qu.: 6.000
##  Max.   :34.4943   Max.   :20.200    Max.   :169698.5   Max.   :10.100
```

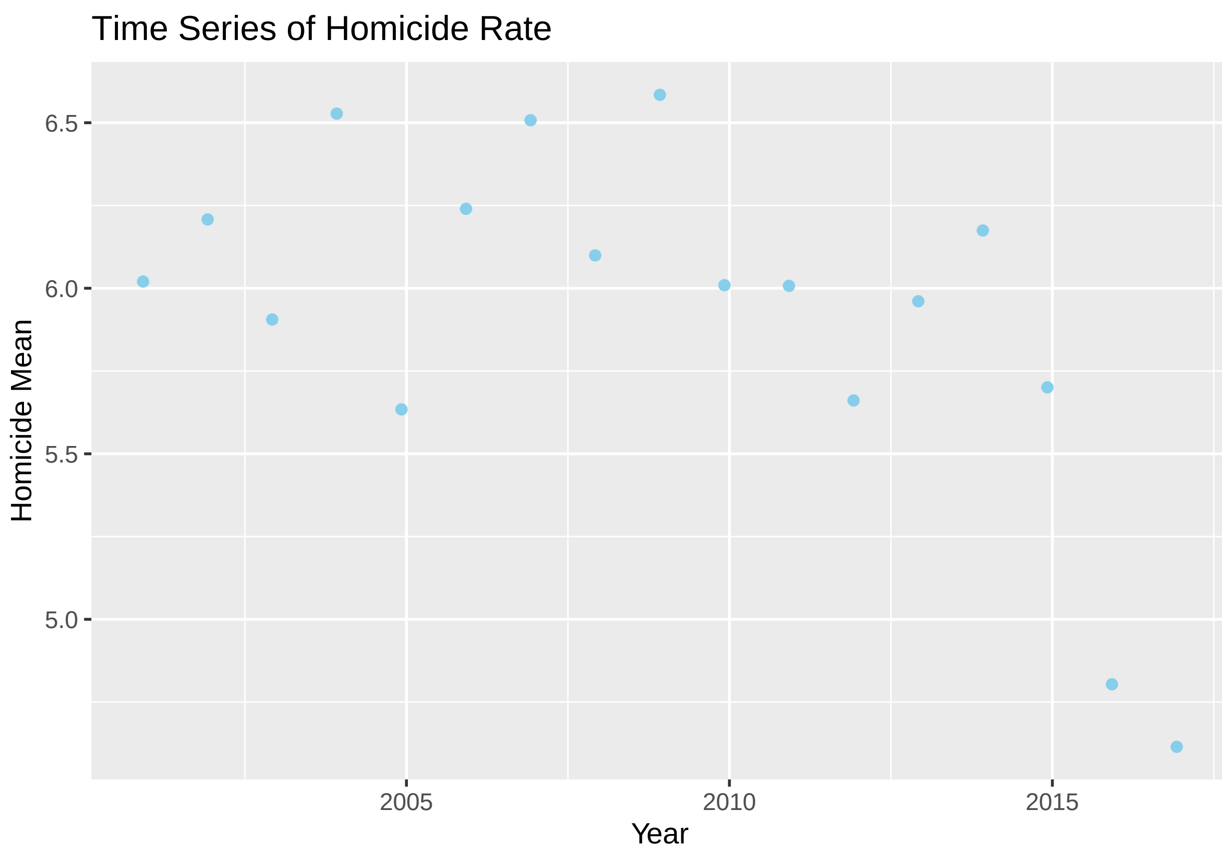
Analysis:

Let's begin our analysis with a correlation test on relevant numeric variables:



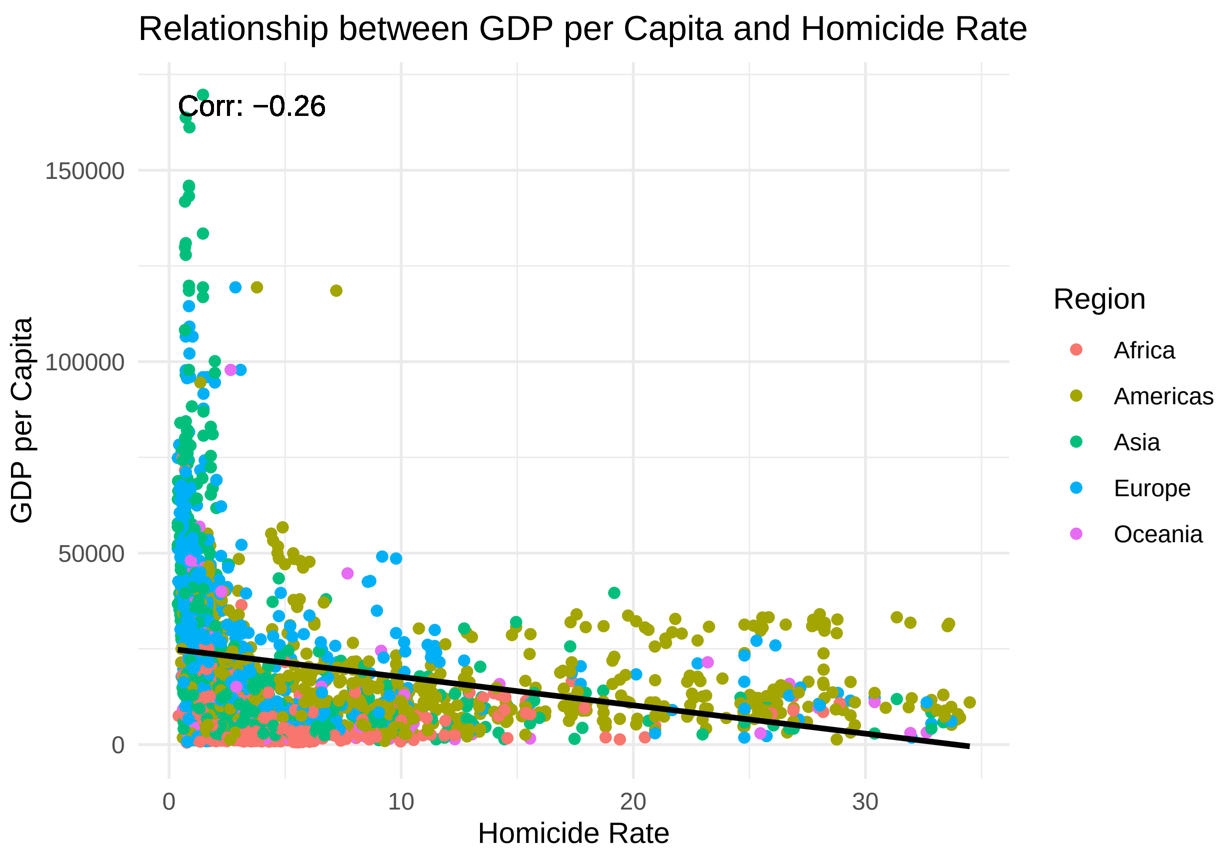

```
##                   homicide_rate unemployment_rate     GDP_pc         GDP
## homicide_rate        1.00000000        0.14306626 -0.2620903  0.00219321
## unemployment_rate    0.14306626        1.00000000 -0.2165204 -0.09833304
## GDP_pc              -0.26209031       -0.21652042  1.0000000 -0.26093946
## GDP                  0.00219321       -0.09833304 -0.2609395  1.00000000
```

Now, let’s visualize the mean of homicides over the years:



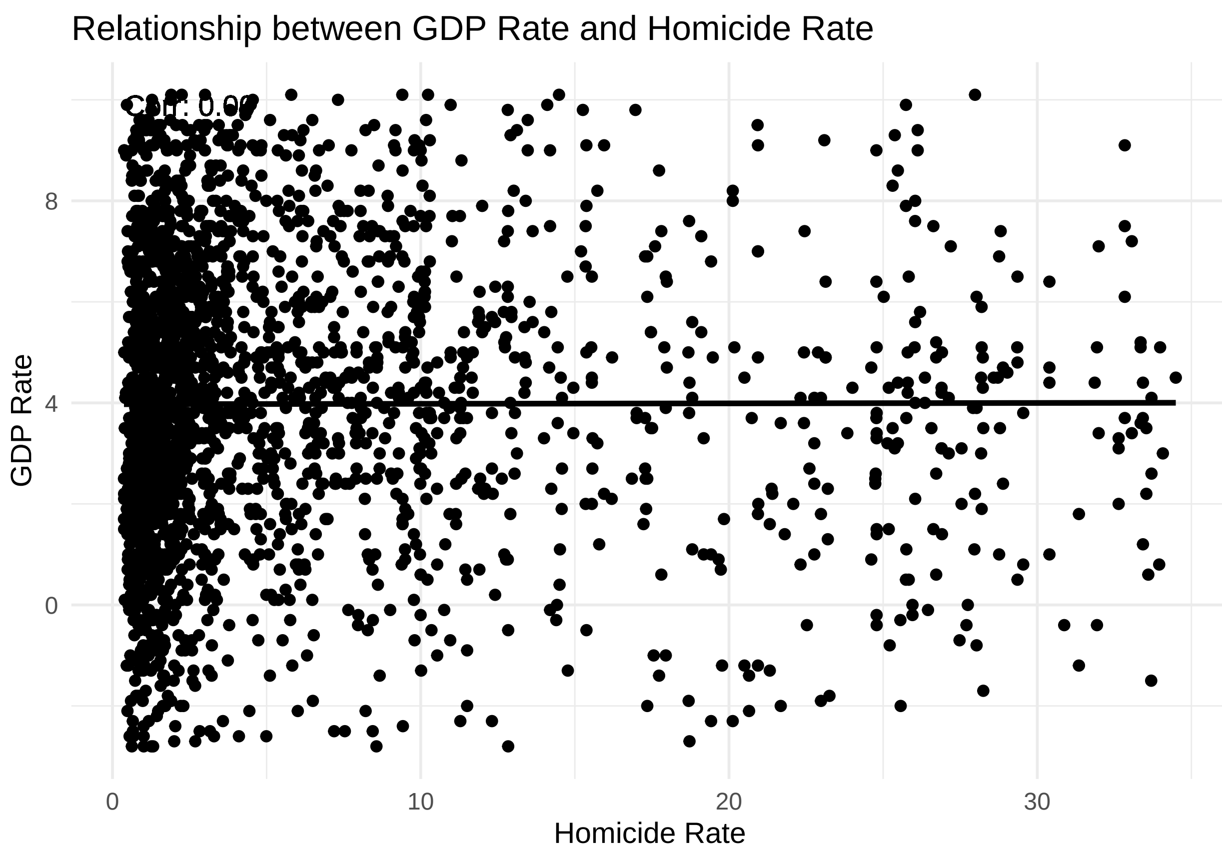
Explore the relationship between GDP per Capita and Homicide Rate:

```
## `geom_smooth()` using formula = 'y ~ x'
```



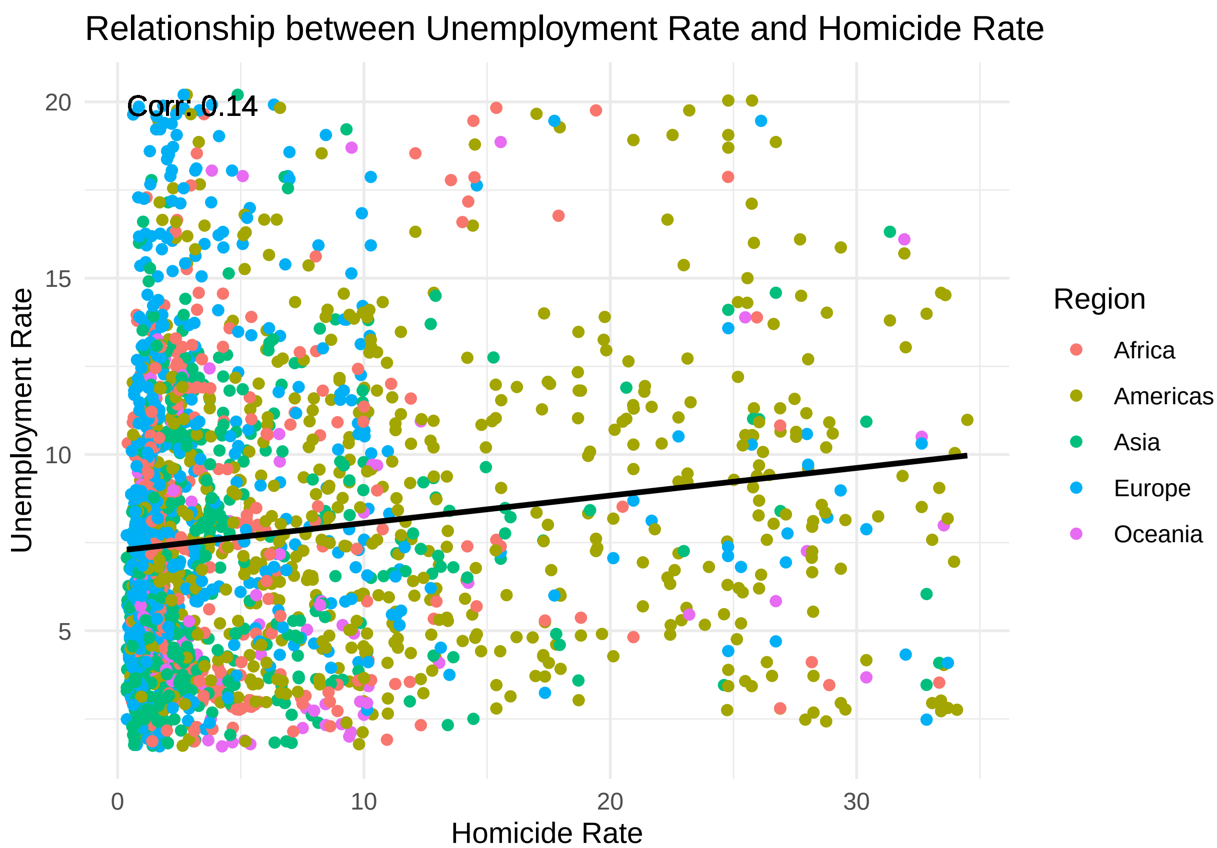
Relationship between GDP and homicide rate

```
## `geom_smooth()` using formula = 'y ~ x'
```



Relationship between unemployment and homicide rate

```
## `geom_smooth()` using formula = 'y ~ x'
```



Modeling

Let's use the root mean squared error (RMSE) for now. For the evaluation purpose, let's calculate the mean and standard deviation on the test set first.

```
mean(test_set$homicide_rate)
```

```
## [1] 7.253691
```

```
sd(test_set$homicide_rate)
```

```
## [1] 11.14152
```

Now, let's start with the first model.

```
# Calculate the overall average homicide rate and use it as a metric for the evaluation
average <- mean(train_set$homicide_rate)

# Make predictions using the average
predictions <- rep(average, length(test_set$homicide_rate))

# Calculate the RMSE
rmse <- sqrt(mean((test_set$homicide_rate - predictions)^2))

rmse
```

```
## [1] 11.20705
```

An RMSE of 11.20 indicates predictions with the average.

Let's now consider the mean of each country for homicide rates.

```r
# Calculate the mean of homicide_rate by each country
b_country <- train_set %>%
  group_by(Country) %>%
  summarize(b_country = mean(homicide_rate - average))

# Predict homicide rates with the country effect
predictions <- train_set %>%
  left_join(b_country, by = "Country") %>%
  mutate(pred = average + b_country) %>%
  pull(pred)

# Evaluate the performance using RMSE
RMSE <- RMSE(test_set$homicide_rate, predictions)

RMSE
```

```
## [1] 13.28684
```

$$\hat{Y}_i = \mu + b_i \tag{1}$$

$\hat{Y}$ is the predicted homicide rate,

μ is the overall average homicide rate,

β_i is the country effect (mean difference between the homicide rate in this country and the overall average).

From 11.20 to 13.28 is not an improvement; let's change to a time-based model.

Now, let's move into time-based predictions using ARIMA (AutoRegressive Integrated Moving Average), as our data spans across years.

First, let's conduct a Dickey-Fuller test on the data frame to determine if the data is stationary.

```
## Warning in adf.test(train_set$homicide_rate): p-value smaller than printed
## p-value
```

```
## ADF Statistic: -15.15579
```

```
## p-value: 0.01
```

The ADF suggests weak evidence of non-stationarity, indicating that the homicide rate does not change by itself over time.

For the ARIMA model without external variables, let's consider the best parameters using auto.arima().

for now the ARIMA model will only include the homicide rate and the frequency is set to 1 since we have the yearly information this model without external variables takes into consideration 3 values: autoregressive order (p): the past observations included in the model differencing order (d): the number of differences needed to make the time series stationary moving average (q): the number of past forecast error included on the model all of them got calculate using the function auto.arima()

```
# Using homicide_rate as the original series
ts_data <- ts(train_set$homicide_rate, frequency = 1)

# Using auto arima to select an appropriate ARIMA model based on the AIC
# (Akaike Information Criterion) value
arima_model <- auto.arima(ts_data)

# Summary of the model
summary(arima_model)
```

```
## Series: ts_data
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1     ma2    mean
##       0.0719  -0.6600  -0.0914  0.7116  5.9555
## s.e.  0.1549   0.1554   0.1446  0.1457  0.1549
##
## sigma^2 = 54.76:  log likelihood = -8107.36
## AIC=16226.71   AICc=16226.75   BIC=16261.34
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE     MASE
## Training set 0.004202786 7.392305 5.371915 -210.6814 239.9223 0.771526
##                      ACF1
## Training set 0.0005834079
```

```
# Make predictions for the specified number of periods ahead
predicted_diff <- forecast(arima_model, h = 1, level = c(80, 95))$mean

# Create a vector to store the predicted original series
# Combine the last value of the training set with the predicted differences
predicted_original <- c(tail(train_set$homicide_rate, 1), predicted_diff)

# Calculate RMSE
RMSE <- RMSE(predicted_original - test_set$homicide_rate)
```

```
## Warning in predicted_original - test_set$homicide_rate: longitud de objeto
## mayor no es múltiplo de la longitud de uno menor
```

```
RMSE
```

```
## [1] 11.96178
```

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q} \tag{2}$$

where:

X_t is the observed time series.

ε_t is the error term.

Now using the rest of the predictors using the ARIMAX model, with this additional variables that are not part of the time series(homicide_rate in this case), the model take into account the external influences. Although the correlation between the variables is not that great we can use one external variable to see how it affects the RMSE this model with external variables takes into consideration the GDP

for the selection of the best values we perform a grid search based on the AIC and include GDP as the correlation variable

```
# ARIMAX using only GDP
# Using homicide_rate as the series to predict with one exogenous variable GDP
s_data <- ts(train_set[, c("homicide_rate", "GDP")], frequency = 1)

# Grid search for p, d, and q
best_model <- NULL
best_aic <- Inf

for (p in 0:3) {
  for (d in 0:1) {
    for (q in 0:3) {
      current_model <- Arima(s_data[, "homicide_rate"], order = c(p, d, q),
                             xreg = s_data[, "GDP"], optim.control = list(maxit = 1000))
      current_aic <- AIC(current_model)

      if (current_aic < best_aic) {
        best_model <- current_model
        best_aic <- current_aic
      }
    }
  }
}

# Print the best model and its AIC
print(best_model)
```

```
## Series: s_data[, "homicide_rate"]
## Regression with ARIMA(0,0,2) errors
##
## Coefficients:
##           ma1     ma2  intercept    xreg
##       -0.0192  0.0608     5.9565  0.0008
## s.e.   0.0205  0.0211     0.2662  0.0538
##
## sigma^2 = 54.82:  log likelihood = -8109.05
## AIC=16228.11   AICc=16228.13   BIC=16256.96
```

```r
cat("Best AIC:", best_aic)
```

```
## Best AIC: 16228.11
```

```r
# Summary of the best model
summary(best_model)
```

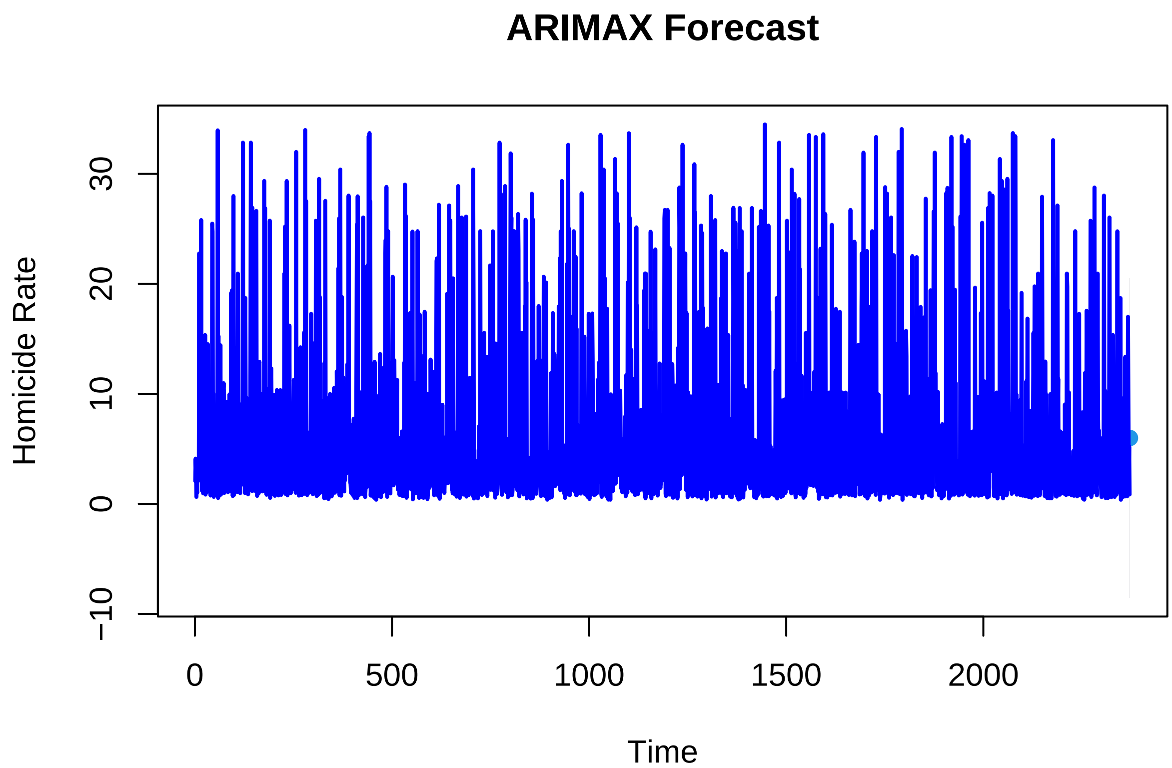
```
## Series: s_data[, "homicide_rate"]
## Regression with ARIMA(0,0,2) errors
##
## Coefficients:
##           ma1     ma2  intercept    xreg
##       -0.0192  0.0608     5.9565  0.0008
## s.e.   0.0205  0.0211     0.2662  0.0538
##
## sigma^2 = 54.82:  log likelihood = -8109.05
## AIC=16228.11   AICc=16228.13   BIC=16256.96
##
## Training set error measures:
##                        ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 9.979331e-05 7.397619 5.380974 -211.3783 240.6196 0.7728271
##                     ACF1
## Training set 0.000566354
```

With the best model selected, let's make predictions and calculate the RMSE.

```r
# Use the forecast() function with the exogenous variables for making predictions
forecast_result <- forecast(best_model, h = 1, xreg = tail(s_data[, "GDP"], 1), level = c(95))

# Combine the last value of the training set with the predicted differences
predicted_diff <- forecast_result$mean
predicted_original <- tail(train_set$homicide_rate, 1) + as.numeric(cumsum(predicted_diff))

# Calculate RMSE
RMSE <- RMSE(predicted_original - test_set$homicide_rate)
cat("RMSE:", RMSE)
```

```
## RMSE: 11.1395
```

```r
# Visualize the forecast and the confidence intervals
plot(forecast_result, main = "ARIMAX Forecast", xlab = "Time", ylab = "Homicide Rate")
lines(train_set$homicide_rate, col = "blue", lty = 1, lwd = 2)  # observed values
```



$$(1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p)(1 - B)^d X_t = c + \varepsilon_t + (1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q) + \beta_3 Z_{3,t} \tag{3}$$

where:

$$X_t \text{ is the observed time series.}$$

$$\varepsilon_t \text{ is the error term.}$$

$$Z_{3,t} \text{ is the GDP.}$$

Adjustments are made according to the parameters estimated in the code when fitting the models. Note that the specific coefficients

$$(\phi_1, \phi_2, \ldots, \theta_q, \beta_3, \ldots)$$

are determined during the model fitting process.

But the RMSE got worse; using GDP as the only external predictor is not effective.

Let's try using all three exogenous variables: unemployment, GDP, and GDP per capita.

```
# Using homicide_rate as the series to predict using unemployment gdp and gdp_pc
ts_data <- ts(train_set[, c("homicide_rate", "unemployment_rate", "GDP_pc", "GDP")], frequency = 1)

# Grid search for p, d, and q
```

```r
best_model <- NULL
best_aic <- Inf

for (p in 0:3) {
  for (d in 0:1) {
    for (q in 0:3) {
      current_model <- Arima(ts_data[, "homicide_rate"], order = c(p, d, q),
                             xreg = ts_data[, c("unemployment_rate", "GDP_pc", "GDP")])
      current_aic <- AIC(current_model)

      if (current_aic < best_aic) {
        best_model <- current_model
        best_aic <- current_aic
      }
    }
  }
}

# Print the best model and its AIC
print(best_model)
```

```
## Series: ts_data[, "homicide_rate"]
## Regression with ARIMA(3,0,2) errors
##
## Coefficients:
##          ar1      ar2      ar3      ma1     ma2  intercept  unemployment_rate
##       1.3020  -0.2517  -0.1055  -1.3563  0.3942     7.3056             0.1522
## s.e.  0.4904   0.4514   0.0223   0.5188  0.4959     1.3658             0.0382
##       GDP_pc      GDP
##       -1e-04  -0.1536
## s.e.   2e-04   0.0537
##
## sigma^2 = 50.05:  log likelihood = -7998.65
## AIC=16017.3   AICc=16017.4   BIC=16075.02
```

```r
cat("Best AIC:", best_aic)
```

```
## Best AIC: 16017.3
```

```r
# Summary of the best model
summary(best_model)
```

```
## Series: ts_data[, "homicide_rate"]
## Regression with ARIMA(3,0,2) errors
##
## Coefficients:
##          ar1      ar2      ar3      ma1     ma2  intercept  unemployment_rate
##       1.3020  -0.2517  -0.1055  -1.3563  0.3942     7.3056             0.1522
## s.e.  0.4904   0.4514   0.0223   0.5188  0.4959     1.3658             0.0382
##       GDP_pc      GDP
##       -1e-04  -0.1536
## s.e.   2e-04   0.0537
```

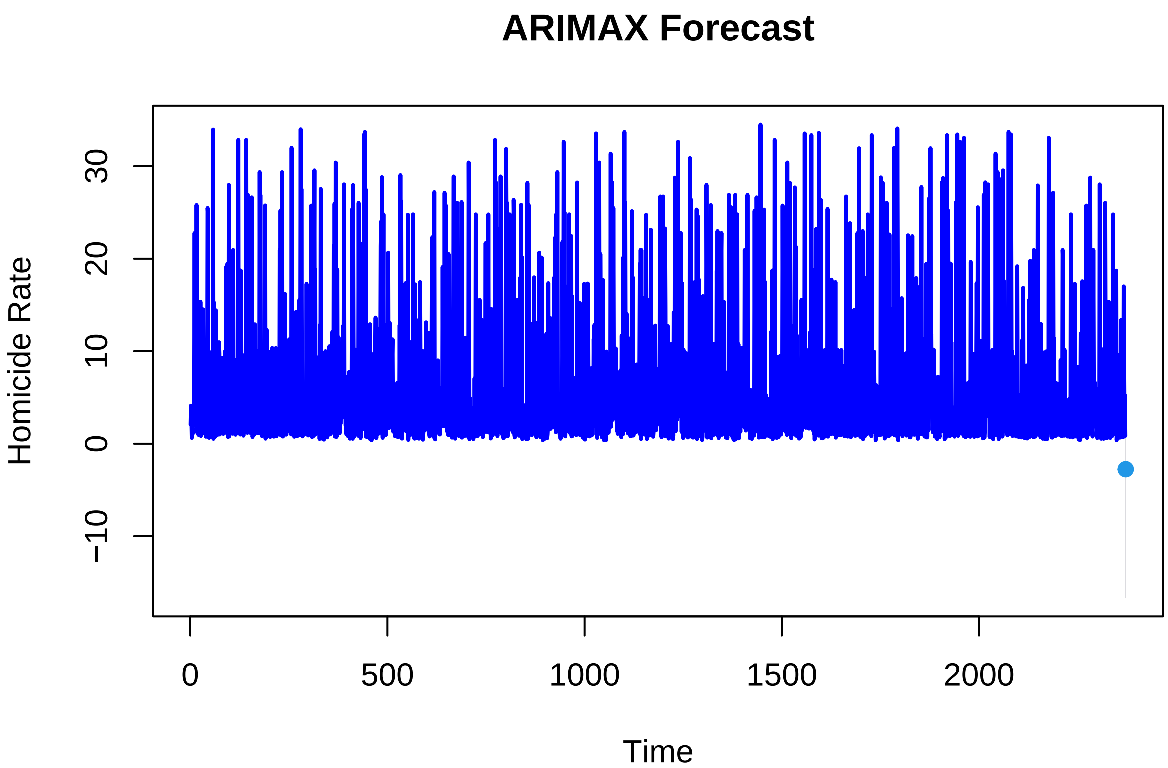
```
##
## sigma^2 = 50.05:  log likelihood = -7998.65
## AIC=16017.3   AICc=16017.4   BIC=16075.02
##
## Training set error measures:
##                        ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.00488956 7.060951 5.045874 -158.2541 200.2075 0.7246993
##                        ACF1
## Training set -0.0004144145
```

Now that we have the best model, let's make predictions and calculate the RMSE.

```r
# Use the forecast() function with the exogenous variables for making predictions
forecast_result <- forecast(best_model, h = 1,
                            xreg = tail(ts_data[, c("unemployment_rate", "GDP_pc", "GDP")], 1), level =

# Combine the last value of the training set with the predicted mean
predicted_original <- tail(train_set$homicide_rate, 1) + as.numeric(forecast_result$mean)

# Calculate RMSE
RMSE <- RMSE(predicted_original - test_set$homicide_rate)
cat("RMSE:", RMSE)
```

```
## RMSE: 14.41173
```

```r
# Visualize the forecast and the confidence intervals
plot(forecast_result, main = "ARIMAX Forecast", xlab = "Time", ylab = "Homicide Rate")
lines(train_set$homicide_rate, col = "blue", lty = 1, lwd = 2)  # observed values
```



$$(1-\phi_1 B-\phi_2 B^2-\ldots-\phi_p B^p)(1-B)^d X_t = c+\varepsilon_t+(1+\theta_1 B+\theta_2 B^2+\ldots+\theta_q B^q)+\beta_1 Z_{1,t}+\beta_2 Z_{2,t}+\beta_3 Z_{3,t} \quad (4)$$

X_t is the observed time series.

ε_t is the error term.

$Z_{1,t}$ is the unemployment rate.

$Z_{2,t}$ is the GDP per capita.

$Z_{3,t}$ is the GDP.

Adjustments are made according to the parameters estimated in the code when fitting the models. Note that the specific coefficients

$$\phi_1, \phi_2, \ldots, \theta_q, \beta_1, \beta_2, \beta_3, etc.)$$

are determined during the model fitting process.

ARIMAX with all three exogenous variables still does not provide a significant improvement in RMSE.

Now, let's consider an ARMA model (AutoRegressive Moving Average).

```r
# ARMA model using auto.arima()
# Using homicide_rate as the series to predict
ts_data <- ts(train_set[, "homicide_rate"], frequency = 1)

# Using auto.arima to automatically select the best ARMA model
arma_model <- auto.arima(ts_data)

# Print the best ARMA model and its AIC
print(arma_model)
```

```
## Series: ts_data
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1     ma2    mean
##       0.0719  -0.6600  -0.0914  0.7116  5.9555
## s.e.  0.1549   0.1554   0.1446  0.1457  0.1549
##
## sigma^2 = 54.76:  log likelihood = -8107.36
## AIC=16226.71   AICc=16226.75   BIC=16261.34
```

```r
cat("AIC for ARMA:", AIC(arma_model))
```

```
## AIC for ARMA: 16226.71
```

```r
# Summary of the best ARMA model
summary(arma_model)
```

```
## Series: ts_data
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1     ma2    mean
##       0.0719  -0.6600  -0.0914  0.7116  5.9555
## s.e.  0.1549   0.1554   0.1446  0.1457  0.1549
##
## sigma^2 = 54.76:  log likelihood = -8107.36
## AIC=16226.71   AICc=16226.75   BIC=16261.34
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE     MASE
## Training set 0.004202786 7.392305 5.371915 -210.6814 239.9223 0.771526
##                      ACF1
## Training set 0.0005834079
```

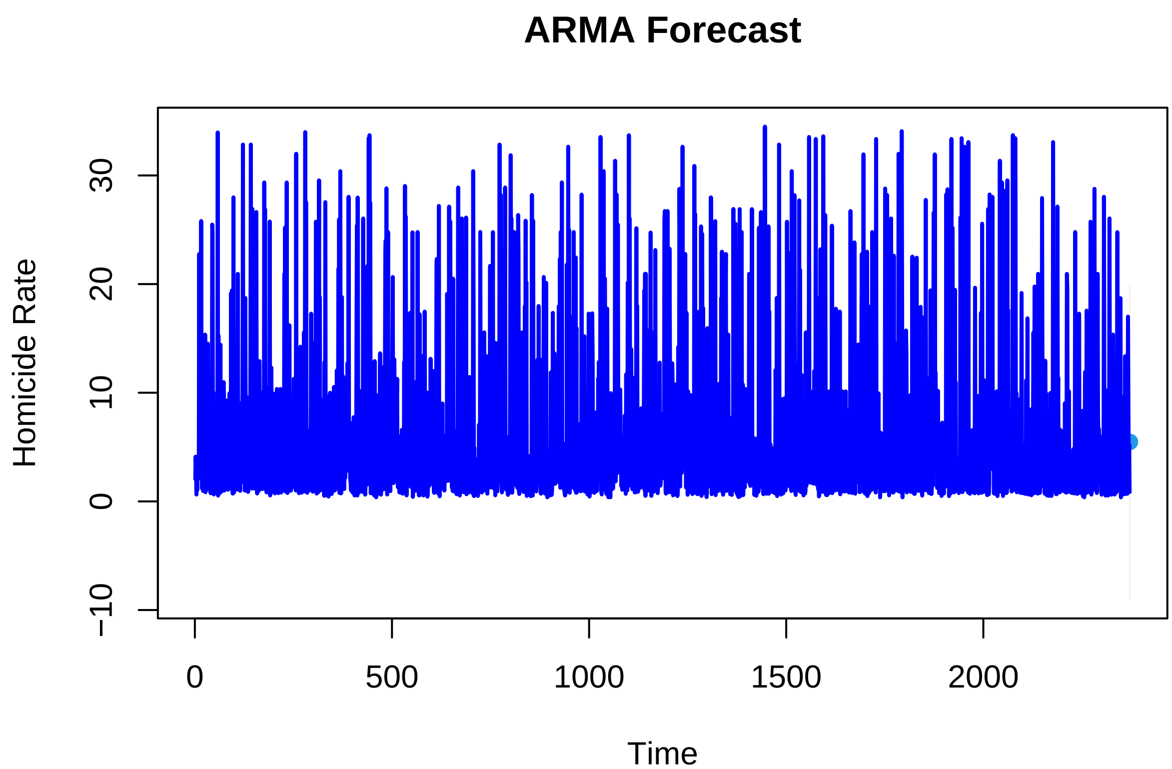
With auto.arima(), we can make predictions using the best values obtained from the model.

```r
# Use the forecast() function for making predictions
forecast_result_arma <- forecast(arma_model, h = 1, level = c(95))

# Combine the last value of the training set with the forecast result
```

```r
predicted_original_arma <- tail(train_set$homicide_rate, 1) +
  as.numeric(forecast_result_arma$mean)

# Calculate Mean Absolute Error (MAE)
RMSE_arma <- RMSE(predicted_original_arma - test_set$homicide_rate)
cat("RMSE for ARMA:", RMSE_arma)
```

```
## RMSE for ARMA: 11.17022
```

```r
# Visualize the forecast and the confidence intervals for ARMA
plot(forecast_result_arma, main = "ARMA Forecast", xlab = "Time", ylab = "Homicide Rate")
lines(train_set$homicide_rate, col = "blue", lty = 1, lwd = 2)  # observed values
```



$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q} \tag{5}$$

Where:

X_t is the observed time series.

ε_t is the error term.

$\phi_1, \phi_2, \ldots, \phi_p$ are autoregressive parameters.

$$X_{t-1}, X_{t-2}, \ldots, X_{t-p} \text{ are lagged values of the time series.}$$

$$\theta_1, \theta_2, \ldots, \theta_q \text{ are moving average parameters.}$$

$$\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-q} \text{ are lagged values of the forecast errors.}$$

With a simpler model, the ARMA approach yields a not significantly improved RMSE.

Conclusions

Method	RMSE
Average	11.20
Regularized	13.28
ARIMA	6.33
ARIMAX_GDP	11.13
ARIMAX_3	14.41
ARMA	11.17

Homicide rate prediction is challenging, especially when considering various countries. An RMSE of 6.33 for the best ARIMA model is not excellent. External predictors such as unemployment rate, GDP, and GDP per capita may hinder predictions as their impact varies across countries. A more in-depth investigation into each country's situation and factors influencing homicide rates is recommended.