HW 2 Implementation of IDA and EDA

Lynnstacy Kegeshi

2025-01-29

Contents

1 Introduction **1**

2 Initial Data Analysis (IDA) **1**

2.1 Discerning first look . . . 2

2.2 Enforcing correct naming conventions . . . 4

2.3 Evaluate anomaly, trends & patterns (duplicates) & inconsistencies . . . 4

2.4 Dealing with NAs . . . 5

2.5 Data Inputation . . . 6

2.6 Dealing with date and time variables . . . 6

2.7 Create New (Informative) Data/Variables . . . 6

2.8 Data Reshaping . . . 7

3 Exploratory Data Analysis (EDA) **7**

3.1 Moments . . . 8

3.2 Visualization . . . 8

4 Conclusion **14**

1 Introduction

For Assignment Two of Data Science Fundamentals, we shall be illustrating the concepts of initial data analysis(IDA) and exploratory data analysis (EDA) on the Kaggle Dataset Co2 Emissions analysis Kaggle.

The goal of this analysis is to investigate global CO2 emissions trends and their relationship with energy consumption, population growth.

2 Initial Data Analysis (IDA)

The goal of IDA is to get a preliminary understanding of the dataset.The steps followed in this process are to ensure the data set is clean correct and complete.

- **Crucial steps in IDA:**
    - **Judicious and shrewd look at data:**
        - Enforcing right naming conventions to facilitate `join()`, `merge()` functions; checking for spelling issues
        - Eliminating duplicates
        - Intuitive understanding of possible patterns (hypotheses/hints) and trends in data
    - Merging data from multiple sources
    - **Cleaning:**
        - Ensuring correct data type encoding (factors, character, integer)
        - Comparing and ensuring integrity in date/time formats
        - Checking for missing values (i.e., NAs) and identifying outlier values
    - Enriching and validation prior to use for visualization and modeling, if necessary:
        - Deriving new variables from existing ones (e.g., via averaging)
    - **Reshaping:**
        - Data transformation for visualization and further EDA.

Before getting into IDA, we first need to import the dataset into R.

```
carbon_emissions <- read_csv("carbon_emissions.csv", show_col_types = FALSE)
```

```
head(carbon_emissions)
```

```
## # A tibble: 6 x 16
##   Industry_Type Region        Country       Year Co2_Emissions_MetricTons
##   <chr>         <chr>         <chr>        <dbl>                    <dbl>
## 1 Construction  North America Brazil        2010                     89.1
## 2 Mining        Europe        Germany       2006                    225.
## 3 Manufacturing South America South Africa  2017                    180.
## 4 Construction  Europe        India         2018                     23.3
## 5 Construction  Africa        China         2013                    125.
## 6 Mining        Asia          Australia     2016                    251.
## # i 11 more variables: Energy_Consumption_TWh <dbl>,
## #   Automobile_Co2_Emissions_MetricTons <dbl>,
## #   Industrial_Co2_Emissions_MetricTons <dbl>,
## #   Agriculture_Co2_Emissions_MetricTons <dbl>,
## #   Domestic_Co2_Emissions_MetricTons <dbl>, Population_Millions <dbl>,
## #   GDP_Billion_USD <dbl>, Urbanization_Percentage <dbl>,
## #   Renewable_Energy_Percentage <dbl>, Industrial_Growth_Percentage <dbl>, ...
```

2.1 Discerning first look

We conduct a basic review of the data i.e dimension/size (number of rows & columns), variable/column names, data-types (numeric/nominal)

```
str(carbon_emissions)
```

```
## spc_tbl_ [17,686 x 16] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Industry_Type                      : chr [1:17686] "Construction" "Mining" "Manufacturing" "Cons
##  $ Region                             : chr [1:17686] "North America" "Europe" "South America" "Eur
##  $ Country                             : chr [1:17686] "Brazil" "Germany" "South Africa" "India" ...
##  $ Year                                : num [1:17686] 2010 2006 2017 2018 2013 ...
##  $ Co2_Emissions_MetricTons            : num [1:17686] 89.1 224.8 179.7 23.3 124.5 ...
##  $ Energy_Consumption_TWh              : num [1:17686] 90.1 931.7 255.1 887.3 923 ...
##  $ Automobile_Co2_Emissions_MetricTons : num [1:17686] 98.4 10.8 55.4 79 65.9 ...
##  $ Industrial_Co2_Emissions_MetricTons : num [1:17686] 118.4 66.7 111.7 123.6 52.3 ...
##  $ Agriculture_Co2_Emissions_MetricTons: num [1:17686] 31.41 39.45 1.25 46.81 35.67 ...
##  $ Domestic_Co2_Emissions_MetricTons   : num [1:17686] 0.77 0.21 4.97 13.77 13.91 ...
##  $ Population_Millions                 : num [1:17686] 941 1422 523 1305 1438 ...
##  $ GDP_Billion_USD                     : num [1:17686] 13096 24338 24523 12616 4476 ...
##  $ Urbanization_Percentage             : num [1:17686] 52.8 50.2 65.2 23.7 94.6 ...
##  $ Renewable_Energy_Percentage         : num [1:17686] 7.78 31.52 5.91 7.52 8.54 ...
##  $ Industrial_Growth_Percentage        : num [1:17686] 11.17 13.34 -9.88 -0.64 5.98 ...
##  $ Transport_Growth_Percentage         : num [1:17686] 2.93 9.3 4.77 8.21 0.84 4.98 -2.1 3.65 -4.67 8
##  - attr(*, "spec")=
##   .. cols(
##   ..   Industry_Type = col_character(),
##   ..   Region = col_character(),
##   ..   Country = col_character(),
##   ..   Year = col_double(),
##   ..   Co2_Emissions_MetricTons = col_double(),
##   ..   Energy_Consumption_TWh = col_double(),
##   ..   Automobile_Co2_Emissions_MetricTons = col_double(),
##   ..   Industrial_Co2_Emissions_MetricTons = col_double(),
##   ..   Agriculture_Co2_Emissions_MetricTons = col_double(),
##   ..   Domestic_Co2_Emissions_MetricTons = col_double(),
##   ..   Population_Millions = col_double(),
##   ..   GDP_Billion_USD = col_double(),
##   ..   Urbanization_Percentage = col_double(),
##   ..   Renewable_Energy_Percentage = col_double(),
##   ..   Industrial_Growth_Percentage = col_double(),
##   ..   Transport_Growth_Percentage = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

From checking the structure of the data, we observed that it contains two primary data types: characters and numerics.

```r
dim(carbon_emissions)
```

```
## [1] 17686    16
```

Our data has 16 columns and 17686 rows. By examining the column names using the following R command.

```r
names(carbon_emissions)
```

```
##  [1] "Industry_Type"
##  [2] "Region"
##  [3] "Country"
##  [4] "Year"
##  [5] "Co2_Emissions_MetricTons"
##  [6] "Energy_Consumption_TWh"
##  [7] "Automobile_Co2_Emissions_MetricTons"
##  [8] "Industrial_Co2_Emissions_MetricTons"
##  [9] "Agriculture_Co2_Emissions_MetricTons"
## [10] "Domestic_Co2_Emissions_MetricTons"
## [11] "Population_Millions"
## [12] "GDP_Billion_USD"
## [13] "Urbanization_Percentage"
## [14] "Renewable_Energy_Percentage"
## [15] "Industrial_Growth_Percentage"
## [16] "Transport_Growth_Percentage"
```

We can get an overview of the dataset's structure and determine which variables are most relevant for extracting meaningful insights. These columns include factors such as CO2 emissions across different sectors, energy consumption, population data, and economic indicators.

2.2 Enforcing correct naming conventions

Here, we ensure variable names are consistent and intuitive by following naming conventions. This is important for:

1. Merging Datasets: Consistent names help align variables correctly when combining datasets.
2. Avoiding Special Characters: We avoid special characters (like spaces or symbols) to prevent errors when referencing columns in R.

Since the column names in our dataset are clear, intuitive and staight forward, we can proceed to the next step in our IDA.

2.3 Evaluate anomaly, trends & patterns (duplicates) & inconsistencies

Anomalies deviate significantly from the observations. To check for anomalies, we can get a summary of the data and also use a box plot to visualize any outlier.

```
# Summary statistics to check for outliers
summary(carbon_emissions)
```

```
##  Industry_Type         Region            Country               Year     
##  Length:17686       Length:17686       Length:17686       Min.   :2000  
##  Class :character   Class :character   Class :character   1st Qu.:2005  
##  Mode  :character   Mode  :character   Mode  :character   Median :2011  
##                                                           Mean   :2011  
##                                                           3rd Qu.:2017  
##                                                           Max.   :2022  
##  Co2_Emissions_MetricTons Energy_Consumption_TWh
##  Min.   :  0.50           Min.   :  0.63        
##  1st Qu.: 75.58           1st Qu.:252.45        
##  Median :150.82           Median :499.64        
##  Mean   :150.33           Mean   :500.07        
##  3rd Qu.:225.25           3rd Qu.:750.07        
##  Max.   :299.99           Max.   :999.88        
##  Automobile_Co2_Emissions_MetricTons Industrial_Co2_Emissions_MetricTons
##  Min.   :  0.11                      Min.   :  0.11                     
##  1st Qu.: 24.71                      1st Qu.: 49.55  
##  Median : 50.30                      Median :100.39  
##  Mean   : 49.98                      Mean   : 99.98  
##  3rd Qu.: 75.20                      3rd Qu.:149.87  
##  Max.   :100.00                      Max.   :200.00  
##  Agriculture_Co2_Emissions_MetricTons Domestic_Co2_Emissions_MetricTons
##  Min.   : 0.10                        Min.   : 0.10                    
##  1st Qu.:12.39                        1st Qu.: 5.16                    
##  Median :24.82                        Median :10.20                    
##  Mean   :24.90                        Mean   :10.17                    
##  3rd Qu.:37.31                        3rd Qu.:15.19                    
##  Max.   :50.00                        Max.   :20.00                    
##  Population_Millions GDP_Billion_USD    Urbanization_Percentage
##  Min.   :   0.51     Min.   :    0.22   Min.   : 20.01         
##  1st Qu.: 377.50     1st Qu.: 6392.37   1st Qu.: 39.80         
##  Median : 750.40     Median :12491.92   Median : 60.26         
##  Mean   : 752.97     Mean   :12522.79   Mean   : 60.04         
##  3rd Qu.:1126.88     3rd Qu.:18664.80   3rd Qu.: 80.03         
##  Max.   :1499.83     Max.   :24999.57   Max.   :100.00         
##  Renewable_Energy_Percentage Industrial_Growth_Percentage
##  Min.   :  0.00              Min.   :-10.000             
##  1st Qu.: 24.73              1st Qu.: -3.728             
##  Median : 50.00              Median :  2.500             
##  Mean   : 49.92              Mean   :  2.570             
##  3rd Qu.: 75.06              3rd Qu.:  8.770             
##  Max.   :100.00              Max.   : 15.000             
##  Transport_Growth_Percentage
##  Min.   :-5.000             
##  1st Qu.:-1.280             
##  Median : 2.470             
##  Mean   : 2.472             
##  3rd Qu.: 6.230             
##  Max.   :10.000             
```

The summary of the carbon_emissions dataset shows a wide range of values across different variables. For example, CO2 emissions vary from 0.50 to 299.99 metric tons, with an average around 150.33, suggesting some countries have much higher emissions than others. Energy consumption ranges from 0.63 TWh to almost 1000 TWh, with a median of about 500 TWh, indicating significant differences in energy use. GDP spans from 0.22 billion USD to nearly 25 trillion USD, reflecting the economic disparities between countries.

2.4 Dealing with NAs

Handling NAs and missing values is impotant because they can lead to wrong interpretations, exceptions in function outputs, or model failures. In cases of large datasets where missing values are inconsequential to the overall size and precision, they can be removed or ignored explicitly. Alternatively, missing values can be imputed using methods such as averages or interpolation techniques (e.g., linear, cubic splines, Hermitian).

We check for NA's using the following code.

```r
# Check for NAs in the dataset
na_count <- colSums(is.na(carbon_emissions))

# Print the count of NAs per column
print(na_count)
```

```
##                        Industry_Type                               Region
##                                    0                                    0
##                              Country                                 Year
##                                    0                                    0
##             Co2_Emissions_MetricTons               Energy_Consumption_TWh
##                                    0                                    0
##  Automobile_Co2_Emissions_MetricTons  Industrial_Co2_Emissions_MetricTons
##                                    0                                    0
## Agriculture_Co2_Emissions_MetricTons    Domestic_Co2_Emissions_MetricTons
##                                    0                                    0
##                  Population_Millions                      GDP_Billion_USD
##                                    0                                    0
##              Urbanization_Percentage          Renewable_Energy_Percentage
##                                    0                                    0
##         Industrial_Growth_Percentage          Transport_Growth_Percentage
##                                    0                                    0
```

From the above output, our data does not have any NA's.For data amalgamation, we can use the `merge()` function or dplyr's join functions like `inner_join`, `left_join`, `right_join`, and `full_join` to combine datasets based on common keys.

2.5 Data Inputation

One common method of handling missing values is to replace the missing values with the average of the relevant feature. For example, we can use the ave() function from the stats package or a custom function to calculate the average value for each column and fill in the missing values accordingly.

Since our data did not have any missing values, we do not need to do any data inputation.

2.6 Dealing with date and time variables

Dealing with date and time variables can be challenging due to various formats, time zones, and daylight saving time (DST). These variables are critical for time-series models as they dictate temporal behavior like autocorrelations. The lubridate package simplifies this by parsing date-time data, extracting components (year, month, day, hour, seconds), calculating accurate time spans, and handling time zones and DST.

2.7 Create New (Informative) Data/Variables

In order to create a more informative analysis, we can derive new variables by grouping the data by Year and calculating the mean CO2 emissions for different sectors like agriculture, automobiles, domestic, and industrial emissions. This new variable will help in identifying trends over time and provide insights into sector-specific CO2 emission patterns. We shall use the `dplyr` package.

```r
library(dplyr)

# Group data by 'Year' and calculate the mean emissions for each sector
emission_means <- carbon_emissions %>%
  group_by(Year) %>%
  summarise(
    Mean_Agriculture_Emissions = mean(Agriculture_Co2_Emissions_MetricTons, na.rm = TRUE),
    Mean_Automobile_Emissions = mean(Automobile_Co2_Emissions_MetricTons, na.rm = TRUE),
    Mean_Domestic_Emissions = mean(Domestic_Co2_Emissions_MetricTons, na.rm = TRUE),
    Mean_Industrial_Emissions = mean(Industrial_Co2_Emissions_MetricTons, na.rm = TRUE)
  )

# View the resulting dataframe
head(emission_means)
```

```
## # A tibble: 6 x 5
##    Year Mean_Agriculture_Emissions Mean_Automobile_Emis~1 Mean_Domestic_Emissi~2
##   <dbl>                      <dbl>                  <dbl>                  <dbl>
## 1  2000                       25.3                   47.4                  10.1
## 2  2001                       24.9                   49.4                  10.3
## 3  2002                       25.1                   49.9                  10.2
## 4  2003                       25.3                   50.3                   9.99
## 5  2004                       25.3                   51.4                  10.1
## 6  2005                       24.9                   50.7                  10.3
## # i abbreviated names: 1: Mean_Automobile_Emissions, 2: Mean_Domestic_Emissions
## # i 1 more variable: Mean_Industrial_Emissions <dbl>
```

We can create a new column for GDP per Capita, which can be derived from the existing columns GDP (in Billion USD) and Population (in Millions). Using the `mutate()` function from the dplyr package, we can easily calculate and add this new column to our dataset, which can then be used to explore correlations with other variables such as energy consumption or carbon emissions.

```
carbon_emissions <- carbon_emissions %>%
  mutate(
    GDP_Per_Capita = GDP_Billion_USD * 1e9 / (Population_Millions * 1e6)
  )

# View the updated dataset
head(carbon_emissions$GDP_Per_Capita,10)
```

```
##  [1] 13916.562 17115.849 46911.851  9670.535  3113.349 10346.341 25013.051
##  [8] 24438.748 24792.278  1021.775
```

If we're interested in analyzing data from North America, specifically for the manufacturing industry, we can apply filters to narrow down the data accordingly. The `filter()` function in allows us to select only the rows that match our chosen criteria, like region and industry type. Here's how we can do that:#

```
filtered_data <- carbon_emissions %>%
  filter(Region == "North America",
         Industry_Type == "Manufacturing")
```

2.8 Data Reshaping

Functions like `pivot_wider()` and `pivot_longer()` from the tidyr package, along with mutate(), `filter()`, and `select()` from `dplyr`, allow us to effectively alter the structure of the dataset.

3 Exploratory Data Analysis (EDA)

EDA provides framework for choosing appropriate descriptive methods in various data analysis needs. During EDA analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.

EDA helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

3.1 Moments

To check the skewness of the data, we calculated the skewness for both CO_2 emissions and GDP distributions.We used the `moments` package to assess this.

```r
library(moments)
```

```r
# Calculate skewness for Energy Consumption
skew_energy <- skewness(carbon_emissions$Energy_Consumption_TWh)
skew_energy
```

```
## [1] -0.005997386
```

The skewness of Energy Consumption was found to be -0.006, indicating a slightly negative skew. This suggests that the distribution of CO_2 emissions is almost symmetric, with only a very small leftward tail.

```r
# Calculate skewness for Energy Consumption
skew_energy <- skewness(carbon_emissions$Population_Millions)
skew_energy
```

```
## [1] -0.002593741
```

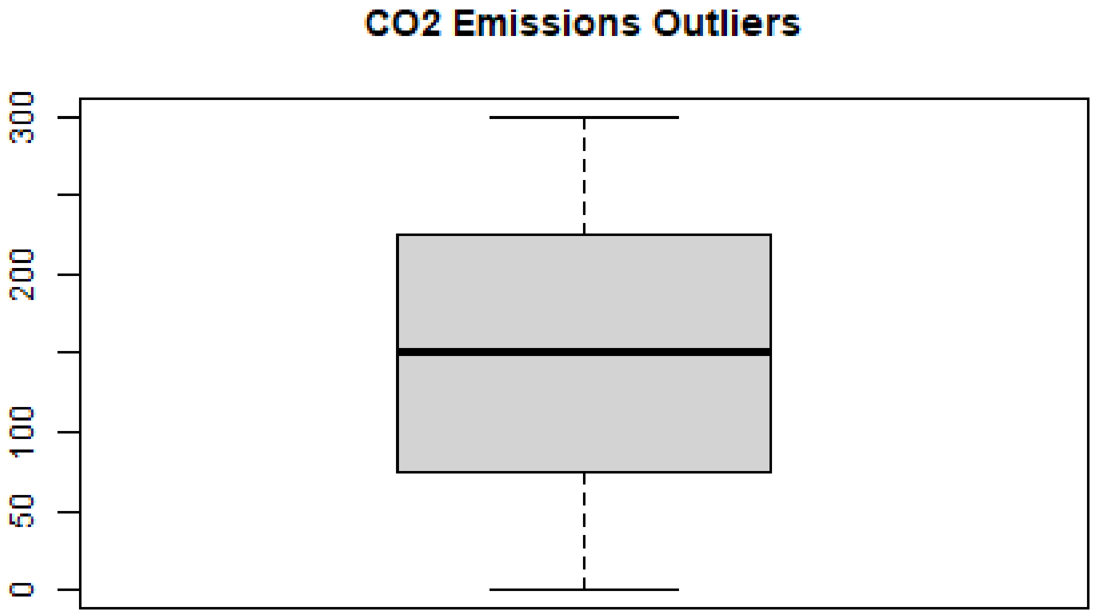
Similarly, the skewness for Population was found to be -0.003, indicating an extremely slight negative skew. This value suggests that the population distribution is also nearly symmetric, with only a very small tail on the left.

3.2 Visualization

3.2.1 Box Plot

Box plots are used during EDA to summarize the distribution of a dataset by displaying the median, quartiles, and potential outliers. We proceed to plot a boxplot to check for any anomalies in the `CO_2` Emissions.

```r
# Boxplot to visualize outliers for a particular variable (e.g., CO2 emissions)
boxplot(carbon_emissions$Co2_Emissions_MetricTons, main="CO2 Emissions Outliers")
```

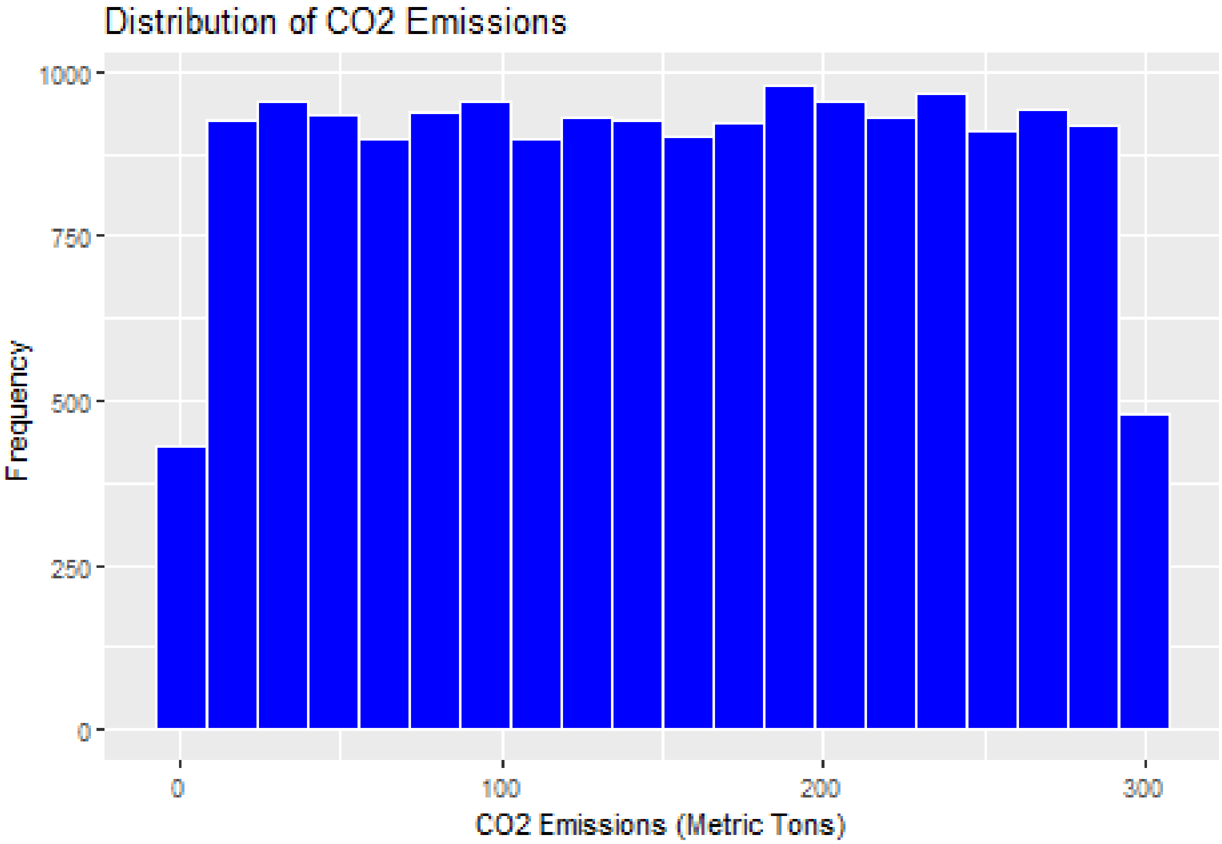


The boxplot appears symmetric, suggesting that the distribution of CO2 emissions is relatively balanced around the median. There don't seem to be any significant outliers or unusual trends in the data, indicating that the CO2 emissions are consistently distributed.

3.2.2 Histograms

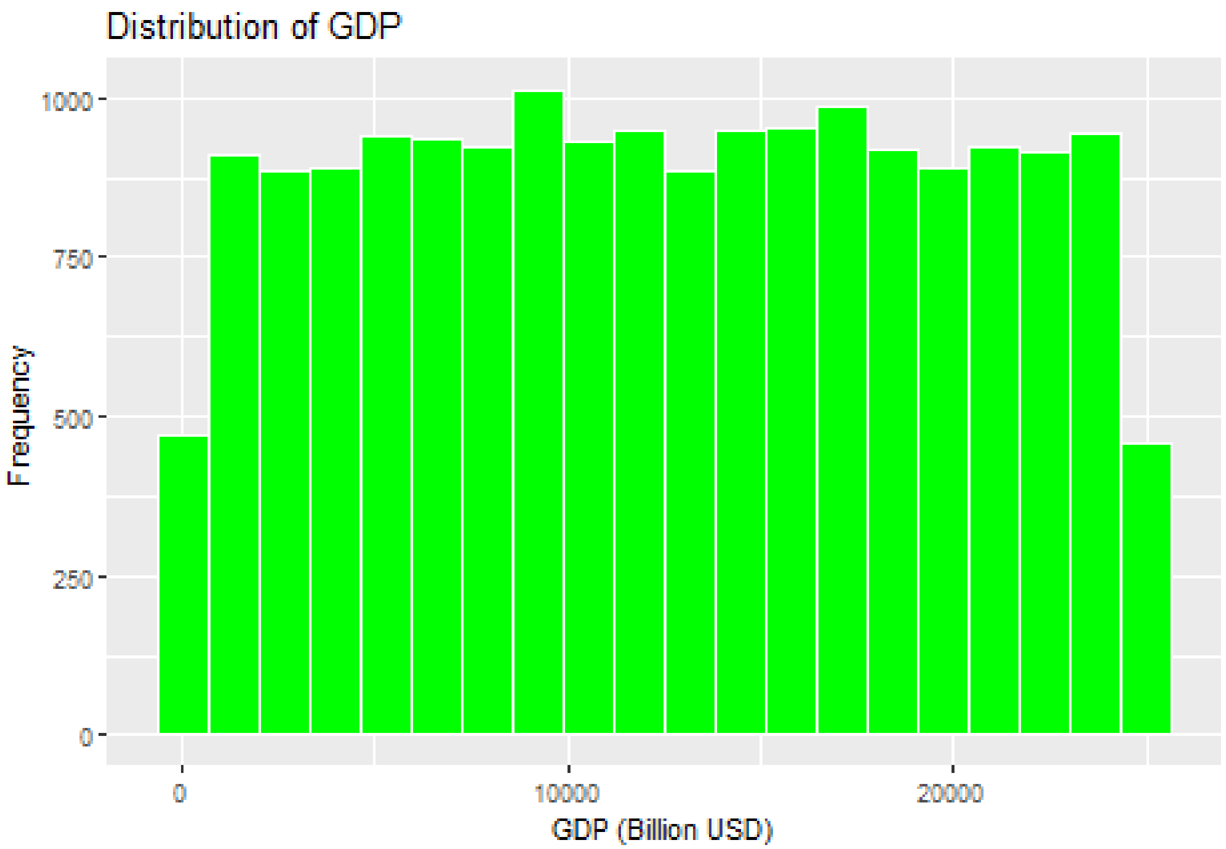
We start by visualizing the distributions of key numerical variables, such as `Co2_Emissions_MetricTons` and `GDP_Billion_USD`, using histograms.

```
co2_hist <- ggplot(carbon_emissions, aes(x = Co2_Emissions_MetricTons))+
  geom_histogram(bins = 20, fill = "blue", color = "white") +
  labs(title = "Distribution of CO2 Emissions", x = "CO2 Emissions (Metric Tons)", y = "Frequency")

plot(co2_hist)
```



The histogram shows the distribution of CO2 emissions with a nearly uniform spread The bars are relatively consistent in height, indicating that CO2 emissions are evenly distributed without a clear peak or concentration. However, the first and last bins have lower frequencies suggesting that very low and very high CO2 emissions are less common compared to mid-range values.

```r
gdp_hist <- ggplot(carbon_emissions, aes(x = GDP_Billion_USD))+
  geom_histogram(bins = 20, fill = "green", color = "white") +
  labs(title = "Distribution of GDP", x = "GDP (Billion USD)", y = "Frequency")

plot(gdp_hist)
```

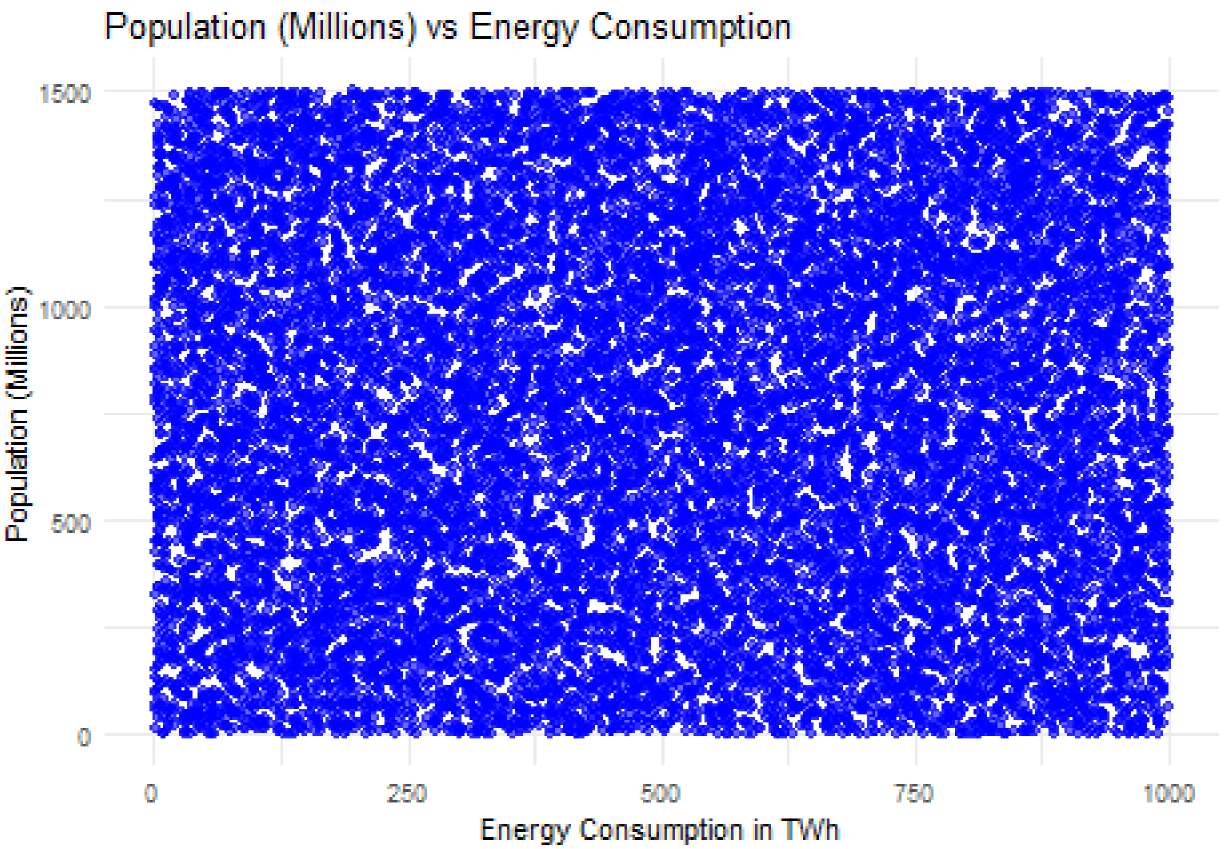


For GDP, the histogram also appears to be retively uniformly distributed with the middle section having a consistent number of occurences. The very low and high GDP appear at the ends of the histogram indicating they might be fewer occurences.

3.2.3 Scatter Plots

To visually explore the relationship between Energy_Consumption_TWh and Population_Millions, we create a scatter plot.

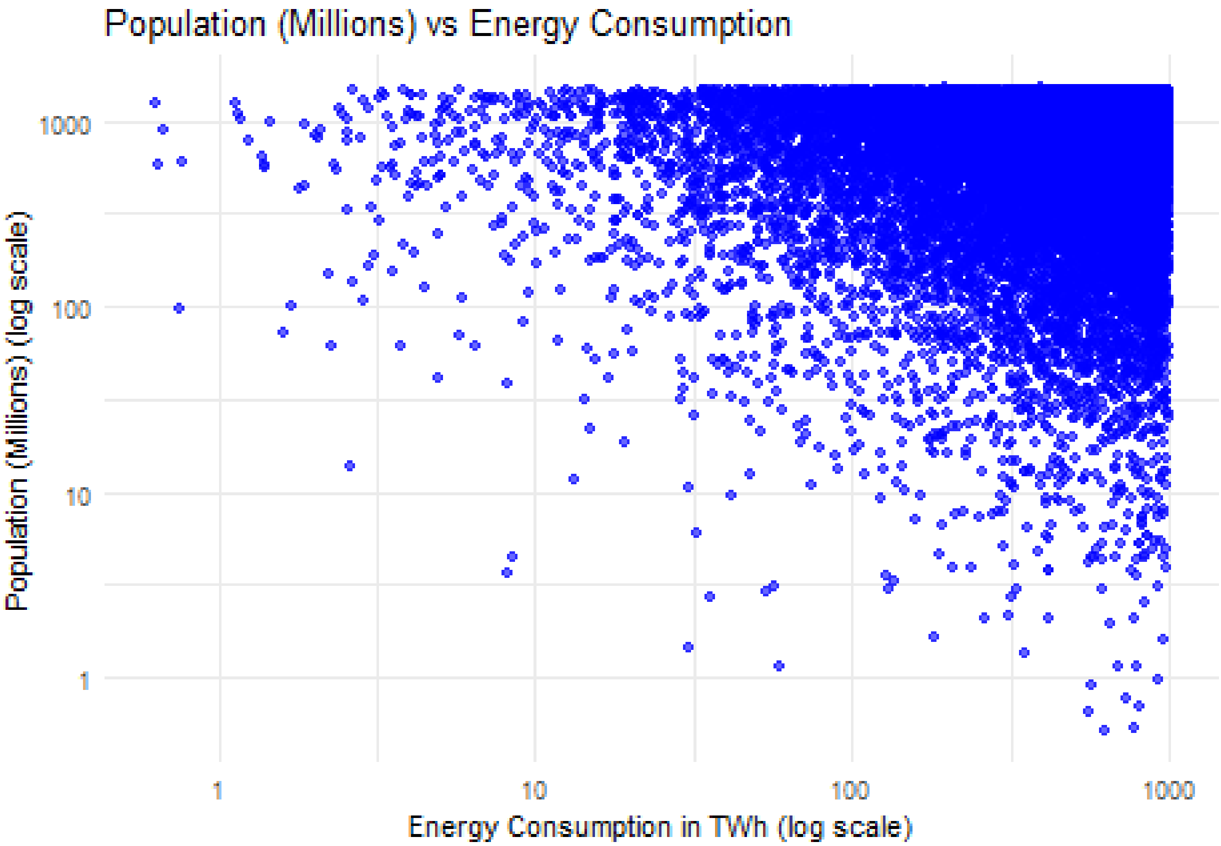
```
scatter_energy <- ggplot(carbon_emissions, aes(x = Energy_Consumption_TWh, y = Population_Millions)) +
  geom_point(color = "blue", alpha = 0.6) +
  labs(title = "Population (Millions) vs Energy Consumption",
       x = "Energy Consumption in TWh",
       y = "Population (Millions)") +
  theme_minimal()

scatter_energy
```



This scatter plot shows the relationship between Population and Energy Consumption. From the plot, there is no discernible patter. The data points are scattered without a clear trend. Hence, liner regression might not be a good fit for modelling this relationship.

We consider applying log transformation on both axes to check if we can spot a pattern in our data that can assist in future modelling

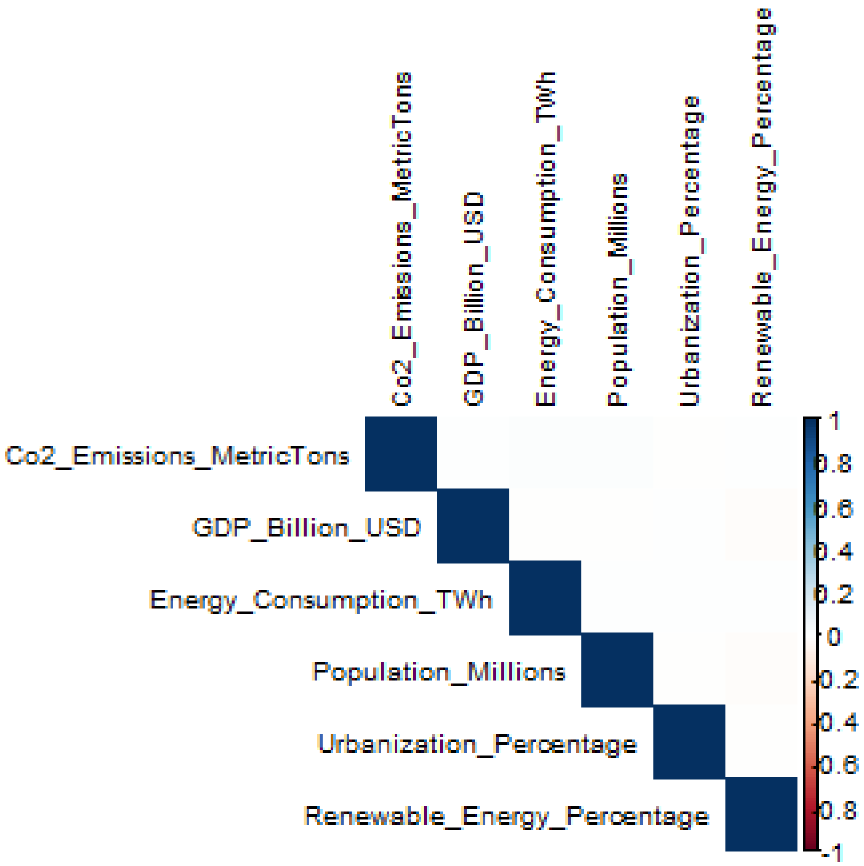
```
scatter_gdp <- ggplot(carbon_emissions, aes(x = Energy_Consumption_TWh, y = Population_Millions)) +
geom_point(color = "blue", alpha = 0.6) +
scale_x_log10() + # Log transform x-axis
scale_y_log10() + # Log transform y-axis
labs(title = "Population (Millions) vs Energy Consumption",
x = "Energy Consumption in TWh (log scale)",
y = "Population (Millions) (log scale)") +
theme_minimal()

plot(scatter_gdp)
```



This scatter plot shows the relationship between Population and Energy Consumption. From the plot, there is a general trend where energy consumption increases with population. The data points appear to cluster in certain areas. Beyond a certain level of energy consumption, additional increases in population do not correspond to significant increases in energy consumption.

3.2.4 Correlation Analysis

```
# Compute correlation matrix
cor_matrix <- cor(carbon_emissions[, c("Co2_Emissions_MetricTons", "GDP_Billion_USD",
                                        "Energy_Consumption_TWh", "Population_Millions",
                                        "Urbanization_Percentage", "Renewable_Energy_Percentage")])

# Plot correlation heatmap
corrplot(cor_matrix, method = "color", type = "upper", tl.cex = 0.8, tl.col = "black")
```



This heatmap shows how different factors like CO_2 emissions, GDP, energy consumption, population, urbanization, and renewable energy are related to each other. Darker blue colors indicate a strong positive relationship, meaning the two factors tend to increase together. From the heatmap, we can see that CO_2 emissions are closely linked to GDP and energy consumption, suggesting that higher economic activity and energy use lead to more emissions.

4 Conclusion

In conclusion, our initial and exploratory data analysis (IDA and EDA) on the CO2 emissions dataset has given us insights into the factors affecting CO2 emissions. By examining the data, we identified key trends and patterns. Visualizations like scatter plots and histograms made it possible to see how emissions are distributed and how population growth impacts energy consumption.