Implementation of IDA and EDA

Lynnstacy Kegeshi

2025-01-24

Contents

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, urbanization, and economic indicators. By exploring patterns and relationships across sectors, regions, and countries, we aim to identify key drivers of emissions and provide actionable insights for reducing them.

As part of this analysis, we will develop a statistical model to understand how factors like energy consumption, population size, and economic activity influence CO2 emissions.

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 - facilitate join(), merge() functions; spelling - Eliminating duplicates - Intuitive understanding of possible patterns (hypotheses/hints) andtrends 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; & adjudging outlier values *Enriching and validation prior to use* *for visualization and modelling,if necessary: Deriving new variables from existing ones, say viaaveraging* 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
##  - 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

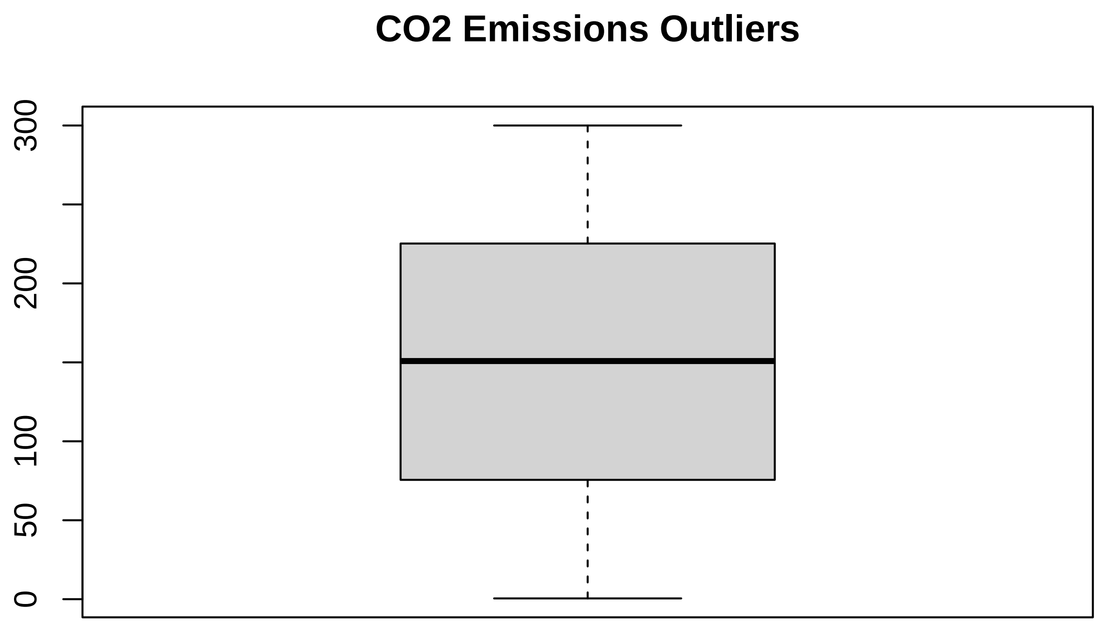
Anomalies deviate significantly from the observations. To check for anomalies, we can get a summary of the data and also use a box plot to visualize any outlier.

```r
# 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 Co2_Emissions_MetricTons column has a minimum value of 0.50, and the maximum is 299.99. We proceed to plot a boxplot for the C02 emissions to check if there might be any anomalies since the mean is 150.33 and median 150.82.

```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.

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 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.

2.6 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>
```