



POLYTECHNIC UNIVERSITY OF THE PHILIPPINES  
STA. MESA MANILA

## **“Analyzing and Visualizing Carbon Emissions in Philippine Provinces Using R”**

Final Requirement for INTE-E1: IT Elective 1

By:

**Belonio, Christian Allen A.  
Empleo, Jericho B.  
Inday, Rey Adrian R.  
Quintana, Krizzia Ann C.**

February 27, 2023



**POLYTECHNIC UNIVERSITY OF THE PHILIPPINES**  
**STA. MESA MANILA**

## **Overview**

This R program analyzes and visualizes carbon emissions data by province in the Philippines. The program uses the `dplyr`, `tidyr`, `ggplot2`, and `writexl` libraries. First, the program generates a data frame with the names of the provinces, years from 2000 to 2020, and randomly generated carbon emissions data. It then pivots the data into a wide format, saves it as a CSV file, and creates a line graph showing the trend of carbon emissions by province over time. Next, the program calculates the total carbon emissions per province using the *dplyr* library, groups the data by province, and summarizes the total carbon emissions. Finally, the program creates a bar graph showing the total carbon emissions per province. The resulting chart provides a clear visualization of how carbon emissions have changed over time in different provinces. This program can be a starting tool that may allow policymakers and researchers to gain insights into carbon emissions trends and to identify areas for potential interventions to reduce carbon emissions.

## **Instructions on how to run the program:**

1. Open a code editor or integrated development environment (IDE) such as RStudio.
2. Create a new script file and paste the entire code into the editor.
3. Ensure that you have installed all the required packages (`dplyr`, `tidyr`, `ggplot2`, and `writexl`) by running the relevant installation commands or checking if they are already installed.
4. Highlight the entire code by clicking and dragging your mouse over it, or by pressing Ctrl + A to select all.
5. Press Ctrl + Enter (or Cmd + Enter on a Mac) to run the entire script at once.
6. The script will run and produce the output as specified, including the line graph and bar graph. You may need to adjust the plot window or zoom in/out to see the graphs clearly.



POLYTECHNIC UNIVERSITY OF THE PHILIPPINES  
STA. MESA MANILA

## Step 1: Essential Libraries

Install the needed package for the program. Just type in R studio “install.packages(“name of the package”) dplyr, tidyr, and ggplot2, and writexl. If the packages are already installed to your computer then load the library to your console.

The screenshot shows the RStudio environment. The script editor on the left contains the following code:

```
1 #Downloading Packages. If installed, skip this step.
2 install.packages("dplyr")
3 install.packages("tidyr")
4 install.packages("ggplot2")
5 install.packages("writexl")
6
7 # Loading Libraries
8 library(dplyr)
9 library(tidyr)
10 library(ggplot2)
11 library(writexl)
12
13 # Creating data frame
14 province <- c("Manila", "Rizal", "Laguna", "Bulacan", "Pampanga")
15 years <- 2002:2022
16 carbonEmissions <- data.frame(
17   province = rep(province, length(years)),
18   years = rep(years, each = length(province))
19 )
```

The console on the bottom left shows the execution of the library loading commands:

```
> # Loading Libraries
> library(dplyr)
> library(tidyr)
> library(ggplot2)
> library(writexl)
>
```

The right-hand pane shows the Environment, History, Connections, and Tutorial tabs, with the Environment tab currently active and empty.



POLYTECHNIC UNIVERSITY OF THE PHILIPPINES  
STA. MESA MANILA

## Step 2: Creating data frame

In this step you need to put the data in the variable “province” and what range of the year to compare the data. After that you need to add another variable by inserting the data.frame where the values are province and years.

```
8 library(dplyr)
9 library(tidyr)
10 library(ggplot2)
11 library(writext1)
12
13 # Creating data frame
14 province <- c("Manila", "Rizal", "Laguna", "Bulacan", "Pampanga")
15 years <- 2002:2022
16 carbonEmissions <- data.frame(
17   province = rep(province, length(years)),
18   year = rep(years, each = length(province))
19 )
20
21 # Generating values for the data of province
22 carbonEmissions$emissions <- runif(nrow(carbonEmissions), min = 0, max = 100)
23
24 # Pivot data | Summarizing the data
```

```
> # Loading Libraries
> library(dplyr)
> library(tidyr)
> library(ggplot2)
> library(writext1)
> # Creating data frame
> province <- c("Manila", "Rizal", "Laguna", "Bulacan", "Pampanga")
> years <- 2002:2022
> carbonEmissions <- data.frame(
+   province = rep(province, length(years)),
+   year = rep(years, each = length(province))
+ )
> |
```



POLYTECHNIC UNIVERSITY OF THE PHILIPPINES  
STA. MESA MANILA

### Step 3: Generating values

In this step, we are asking the R program to generate random values or data for the province that ranges between 0 - 100 outputs.

```
11 library(writexl)
12
13 # Creating data frame
14 province <- c("Manila", "Rizal", "Laguna", "Bulacan", "Pampanga")
15 years <- 2002:2022
16 carbonEmissions <- data.frame(
17   province = rep(province, length(years)),
18   year = rep(years, each = length(province))
19 )
20
21 # Generating values for the data of province
22 carbonEmissions$emissions <- runif(nrow(carbonEmissions), min = 0, max = 100)
23
24 # Pivot data | Summarizing the data
25 wideEmissions <- pivot_wider(carbonEmissions, names_from = province, values_from = emissions)
26
27 # Save the data to a CSV file
28 write_xlsx(wideEmissions, "carbonEmissions.xlsx")
29 (Top Level)
```

The screenshot shows the RStudio environment with the following components:

- Source Editor:** Contains the R script code for generating random values and pivoting the data.
- Environment:** Shows the loaded packages: tidyverse, ggplot2, and writexl.
- Console:** Displays the execution of the R script, showing the creation of the data frame and the generation of random values.
- Files:** Shows the project files, including the R script and the generated Excel file.

The Windows taskbar at the bottom shows the system clock as 4:08 pm on 27/02/2023, with a temperature of 30°C and a weather condition of Haze.



POLYTECHNIC UNIVERSITY OF THE PHILIPPINES  
STA. MESA MANILA

## Step 4: Pivot data

Next, Pivot the data to wide format for easier viewing. Also, summarize, sort, group and reorganize data, as well as execute other complex calculations on it.

```
Carbon-Emission-Group-Program.r
14 province <- c("Manila", "Rizal", "Laguna", "Bulacan", "Pampanga")
15 years <- 2002:2022
16 carbonEmissions <- data.frame(
17   province = rep(province, length(years)),
18   year = rep(years, each = length(province))
19 )
20
21 # Generating values for the data of province
22 carbonEmissions$emissions <- runif(nrow(carbonEmissions), min = 0, max = 100)
23
24 # Pivot data | Summarizing the data
25 wideEmissions <- pivot_wider(carbonEmissions, names_from = province, values_from = emissions)
26
27 # Save the data to a CSV file
28 write_xlsx(wideEmissions, "carbonEmissions.xlsx")
29
30 # Create line graph

R 4.2.2 . ~/
> library(ggplot2)
> library(writexl)
> # Creating data frame
> province <- c("Manila", "Rizal", "Laguna", "Bulacan", "Pampanga")
> years <- 2002:2022
> carbonEmissions <- data.frame(
+   province = rep(province, length(years)),
+   year = rep(years, each = length(province))
+ )
> # Generating values for the data of province
> carbonEmissions$emissions <- runif(nrow(carbonEmissions), min = 0, max = 100)
> # Pivot data | Summarizing the data
> wideEmissions <- pivot_wider(carbonEmissions, names_from = province, values_from = emissions)
> |
```



**POLYTECHNIC UNIVERSITY OF THE PHILIPPINES**  
**STA. MESA MANILA**

### Step 5: Save the data to an Excel file

By saving the data to a .xlsx file, excel will automatically open and display the data to a new workbook.

```
17 province = rep(province, length(years)),
18 year = rep(years, each = length(province))
19 )
20
21 # Generating values for the data of province
22 carbonEmissions$emissions <- runif(nrow(carbonEmissions), min = 0, max = 100)
23
24 # Pivot data | Summarizing the data
25 wideEmissions <- pivot_wider(carbonEmissions, names_from = province, values_from = emissions)
26
27 # Save the data to a CSV file
28 write_xlsx(wideEmissions, "carbonEmissions.xlsx")
29
30 # Create line graph
31 ggplot(wideEmissions, aes(x = year)) +
32   geom_line(aes(y = Manila, color = "Manila")) +
33   geom_line(aes(y = Rizal, color = "Rizal")) +
34
35 (Top Level)
```

```
R 4.2.2 - C:/Users/lericho/Downloads/
> library(ggplot2)
> library(writexl)
> # Creating data frame
> province <- c("Manila", "Rizal", "Laguna", "Bulacan", "Pampanga")
> years <- 2002:2022
> carbonEmissions <- data.frame(
+   province = rep(province, length(years)),
+   year = rep(years, each = length(province))
+ )
> # Generating values for the data of province
> carbonEmissions$emissions <- runif(nrow(carbonEmissions), min = 0, max = 100)
> # Save the data to a CSV file
> write_xlsx(wideEmissions, "carbonEmissions.xlsx")
>
```



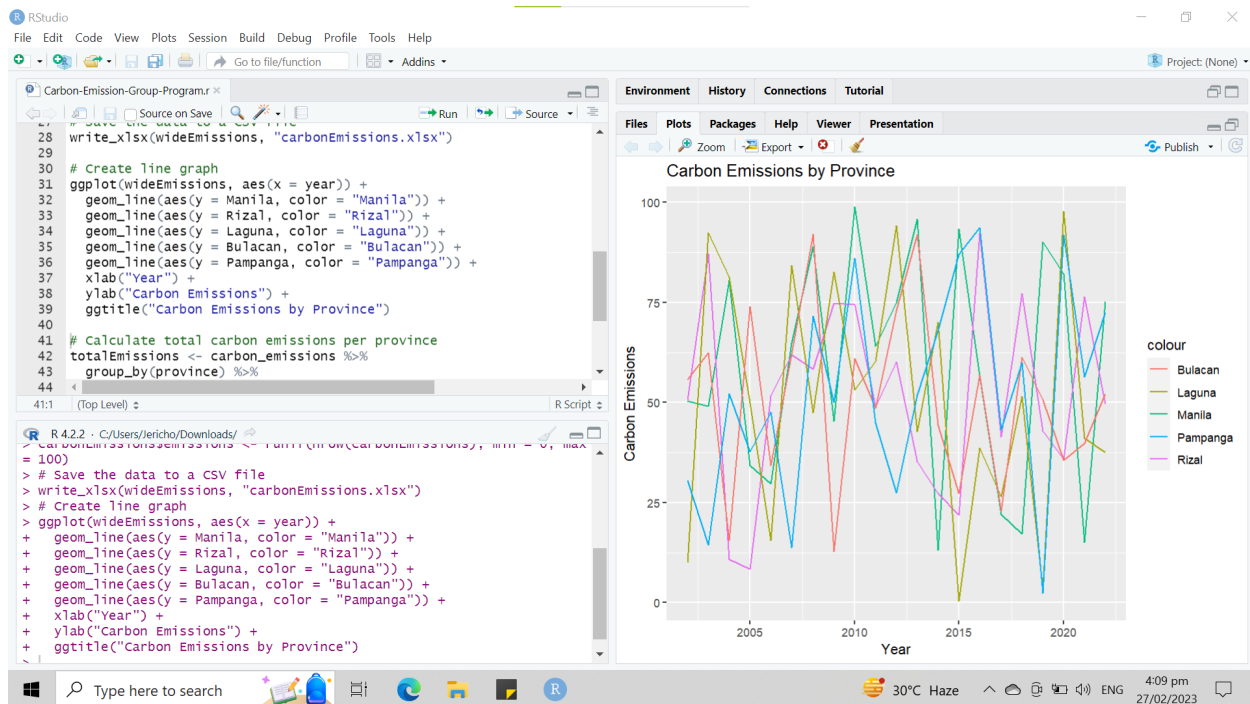




POLYTECHNIC UNIVERSITY OF THE PHILIPPINES  
STA. MESA MANILA

## Step 6. Plotting data using ggplot2

After reading the whole excel file, plot the data as x and y axis. Import first the required library for visualization and import the xlsx library for reading xlsx data.

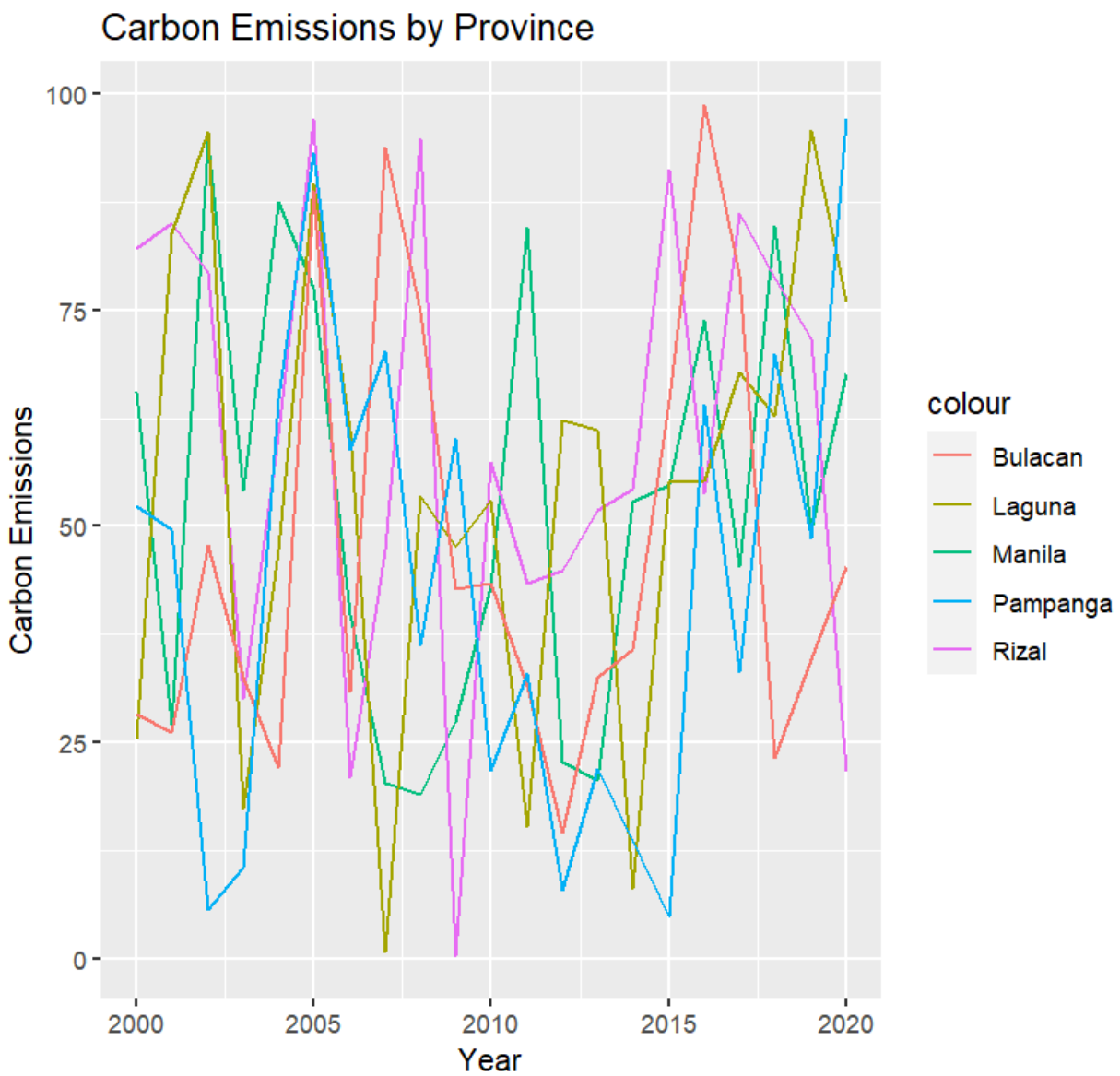




POLYTECHNIC UNIVERSITY OF THE PHILIPPINES  
STA. MESA MANILA

**Step 7: Line Graph**

Here is the Line graph of Carbon Emissions by Province:





POLYTECHNIC UNIVERSITY OF THE PHILIPPINES  
STA. MESA MANILA

## Step 8: Calculate total carbon emissions per province for Bar Graph

Next step is to collect all the data and calculate the total Carbon Emissions per Province and generate a bar graph of the output.

```
35 geom_line(aes(y = Bulacan, color = "Bulacan")) +  
36 geom_line(aes(y = Pampanga, color = "Pampanga")) +  
37 xlab("Year") +  
38 ylab("Carbon Emissions") +  
39 ggtitle("Carbon Emissions by Province")  
40  
41 # Calculate total carbon emissions per province  
42 totalEmissions <- carbon_emissions %>%  
43 group_by(province) %>%  
44 summarize(totalEmissions = sum(emissions))  
45  
46 # Create bar graph  
47 ggplot(totalEmissions, aes(x = province, y = totalEmissions, fill = province)) +  
48 geom_bar(stat = "identity") +  
49 xlab("Province") +  
50 ylab("Total Carbon Emissions") +  
51 ggtitle("Total Carbon Emissions by Province")  
52  
46:1 (Top Level) ±
```

```
R 4.2.2 - C:/Users/ericho/Downloads/  
+ ggplot(carbonEmissions, aes(x = year)) +  
+ geom_line(aes(y = Manila, color = "Manila")) +  
+ geom_line(aes(y = Rizal, color = "Rizal")) +  
+ geom_line(aes(y = Laguna, color = "Laguna")) +  
+ geom_line(aes(y = Bulacan, color = "Bulacan")) +  
+ geom_line(aes(y = Pampanga, color = "Pampanga")) +  
+ xlab("Year") +  
+ ylab("Carbon Emissions") +  
+ ggtitle("Carbon Emissions by Province")  
> # Calculate total carbon emissions per province  
> totalEmissions <- carbon_emissions %>%  
+ group_by(province) %>%  
+ summarize(totalEmissions = sum(emissions))  
>
```

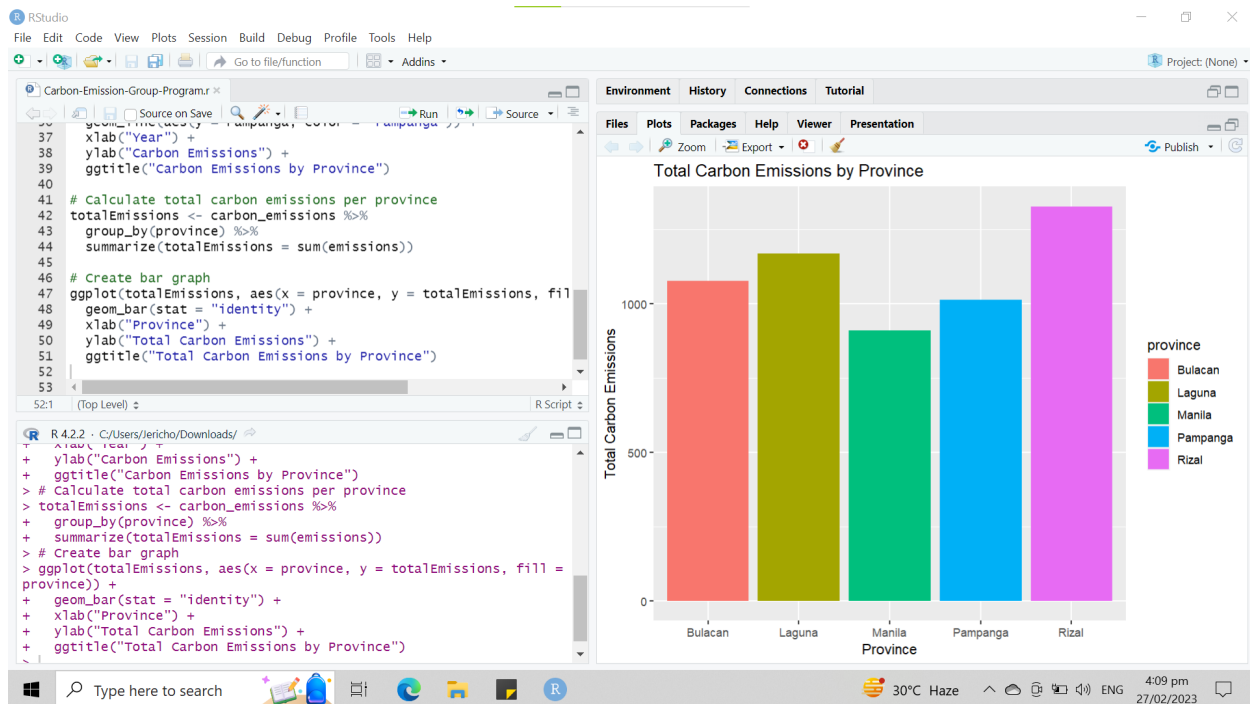
The screenshot shows the RStudio environment with a script editor containing R code for data visualization. The code includes two main parts: a line graph and a bar graph. The line graph plots carbon emissions over time for five provinces: Manila, Rizal, Laguna, Bulacan, and Pampanga. The bar graph calculates the total carbon emissions for each province. The RStudio interface includes a menu bar, a toolbar, a script editor, a console, and an environment pane.



POLYTECHNIC UNIVERSITY OF THE PHILIPPINES  
STA. MESA MANILA

## Step 9: Creating bar graph

Finally, create a bar graph for this output to visualize the total Carbon Emissions by Province.





POLYTECHNIC UNIVERSITY OF THE PHILIPPINES  
STA. MESA MANILA

Step 10: Bar Graph

