MANZI SHEMA CHRISTIAN

ID101382

October 13, 2025

Project I: World Population Trend and CO2 Emission

1. Instructions

- Assignment due date: Sunday, October 12, 2025, at 11:59 am CAT,
- Feel free to discuss, but **don't copy other student's work**,
- Late submission will result in a 10% reduction in total marks,
- This is an individual project. Everyone will submit a html knit markdown file named your__first__and__last__names.html and PDF report named last__names__CO2.pdf.

2.Task Context:

In this dataset project, we are interested in analyzing the relationship between population growth and CO2 emissions. We have data for world population and CO2 emissions. The population dataset contains different features such as county names,population density, population growth rate, population ranking, world population percentage, and so on.

3. Task Objectives

Implementation of data management skills. * Exploratory data analysis, * Prepare a dataset for analysis, * Use visualization tools for population trend, * Correlation analysis, * Creating new variable from existing ones.

4. Task description

4.1 Downloading data and load them to R studio

- Load both World Population Dataset and CO2 Emissions Around the World to your R studio.

A.Load or import World Population Dataset

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)
library(dplyr)
library(tinytex)
library(rmarkdown)
library(ggplot2)

install.packages("tinytex")
```

```
## Warning: package 'tinytex' is in use and will not be installed
```

```r
tinytex::install_tinytex()
```

```
## tlmgr --repository http://www.preining.info/tlgpg/ install tlgpg
```

```
## tlmgr option repository "https://us.mirrors.cicku.me/ctan/systems/texlive/tlnet"
```

```
## tlmgr update --list
```

```r
# loading world population data set
# Load required package

library(readr)

# Load the CSV file safely with proper encoding
file_path <- "C:/Users/pc/Desktop/REXAM/world_population.csv"

# Read CSV using base R with encoding Windows-1252
world_population <- read.csv(
  file_path,fileEncoding = "Windows-1252",
  check.names = FALSE,  # We'll sanitize names next
  stringsAsFactors = FALSE
)

# Convert all column names to UTF-8 and valid R names

names(world_population) <- make.names(iconv(names(world_population),
              from = "Windows-1252", to = "UTF-8"), unique = TRUE)

# Convert all character columns to UTF-8
world_population[] <- lapply(world_population, function(x) {
  if (is.character(x)) iconv(x, from = "Windows-1252",
                              to = "UTF-8") else x})

CO2_emission<-read_csv(
  "C:/Users/pc/Desktop/REXAM/CO2_emission.csv")
```

```
## New names:
## * `2019` -> `2019...34`
## * `2019` -> `2019...35`
```

```
## Rows: 215 Columns: 35
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): Country Name, country_code, Region, Indicator Name
## dbl (31): 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
  # Handles special characters


# Check the first few rows
head(world_population)
```

```
##   Rank CCA3 Country.Territory          Capital Continent X2022.Population
## 1   36  AFG       Afghanistan            Kabul      Asia         41128771
## 2  138  ALB           Albania           Tirana    Europe          2842321
## 3   34  DZA           Algeria          Algiers    Africa         44903225
## 4  213  ASM    American Samoa        Pago Pago   Oceania            44273
## 5  203  AND           Andorra Andorra la Vella    Europe            79824
## 6   42  AGO            Angola           Luanda    Africa         35588987
##   X2020.Population X2015.Population X2010.Population X2000.Population
## 1         38972230         33753499         28189672         19542982
## 2          2866849          2882481          2913399          3182021
## 3         43451666         39543154         35856344         30774621
## 4            46189            51368            54849            58230
## 5            77700            71746            71519            66097
## 6         33428485         28127721         23364185         16394062
##   X1990.Population X1980.Population X1970.Population Area..kmÂ..
## 1         10694796         12486631         10752971      652230
## 2          3295066          2941651          2324731       28748
## 3         25518074         18739378         13795915     2381741
## 4            47818            32886            27075         199
## 5            53569            35611            19860         468
## 6         11828638          8330047          6029700     1246700
##   Density..per.kmÂ.. Growth.Rate World.Population.Percentage
## 1            63.0587      1.0257                        0.52
## 2            98.8702      0.9957                        0.04
## 3            18.8531      1.0164                        0.56
## 4           222.4774      0.9831                        0.00
## 5           170.5641      1.0100                        0.00
## 6            28.5466      1.0315                        0.45
```

4.2 Exploratory data analysis

World Population Dataset display variable names

```r
names(world_population)
```

```
##  [1] "Rank"                        "CCA3"
##  [3] "Country.Territory"           "Capital"
##  [5] "Continent"                   "X2022.Population"
##  [7] "X2020.Population"            "X2015.Population"
##  [9] "X2010.Population"            "X2000.Population"
## [11] "X1990.Population"            "X1980.Population"
## [13] "X1970.Population"            "Area..kmÂ.."
## [15] "Density..per.kmÂ.."          "Growth.Rate"
## [17] "World.Population.Percentage"
```

the top 5 rows in population dataset,

```r
# the top 5 rows in population dataset
head(world_population,5)
```

```
##   Rank CCA3 Country.Territory          Capital Continent X2022.Population
## 1   36  AFG       Afghanistan            Kabul      Asia         41128771
## 2  138  ALB           Albania           Tirana    Europe          2842321
## 3   34  DZA           Algeria          Algiers    Africa         44903225
## 4  213  ASM    American Samoa        Pago Pago   Oceania            44273
## 5  203  AND           Andorra Andorra la Vella    Europe            79824
##   X2020.Population X2015.Population X2010.Population X2000.Population
## 1         38972230         33753499         28189672         19542982
## 2          2866849          2882481          2913399          3182021
## 3         43451666         39543154         35856344         30774621
## 4            46189            51368            54849            58230
## 5            77700            71746            71519            66097
##   X1990.Population X1980.Population X1970.Population Area..kmÂ..
## 1         10694796         12486631         10752971      652230
## 2          3295066          2941651          2324731       28748
## 3         25518074         18739378         13795915     2381741
## 4            47818            32886            27075         199
## 5            53569            35611            19860         468
##   Density..per.kmÂ.. Growth.Rate World.Population.Percentage
## 1            63.0587      1.0257                        0.52
## 2            98.8702      0.9957                        0.04
## 3            18.8531      1.0164                        0.56
## 4           222.4774      0.9831                        0.00
## 5           170.5641      1.0100                        0.00
```

```r
head(CO2_emission)
```

```
## # A tibble: 6 x 35
##   `Country Name`      country_code Region `Indicator Name` `1990` `1991`  `1992`
##   <chr>               <chr>        <chr>  <chr>             <dbl>  <dbl>   <dbl>
## 1 Aruba               ABW          Latin~ CO2 emissions (~ NA     NA     NA
## 2 Afghanistan         AFG          South~ CO2 emissions (~  0.192  0.168  0.0960
## 3 Angola              AGO          Sub-S~ CO2 emissions (~  0.554  0.545  0.544
## 4 Albania             ALB          Europ~ CO2 emissions (~  1.82   1.24   0.684
## 5 Andorra             AND          Europ~ CO2 emissions (~  7.52   7.24   6.96
```

```
## 6 United Arab Emirat~ ARE          Middl~ CO2 emissions (~ 30.2   31.8   29.1
## # i 28 more variables: '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...34' <dbl>, '2019...35' <dbl>
```

the top 10 rows in emission dataset,

```r
# the top 10 rows in emission dataset
tail(world_population,10)
```

```
##     Rank CCA3 Country.Territory      Capital     Continent X2022.Population
## 225   43  UZB        Uzbekistan     Tashkent          Asia         34627652
## 226  181  VUT           Vanuatu    Port-Vila       Oceania           326740
## 227  234  VAT      Vatican City Vatican City        Europe              510
## 228   51  VEN         Venezuela      Caracas South America         28301696
## 229   16  VNM           Vietnam        Hanoi          Asia         98186856
## 230  226  WLF Wallis and Futuna     Mata-Utu       Oceania            11572
## 231  172  ESH    Western Sahara    El AaiÃºn        Africa           575986
## 232   46  YEM             Yemen        Sanaa          Asia         33696614
## 233   63  ZMB            Zambia       Lusaka        Africa         20017675
## 234   74  ZWE          Zimbabwe       Harare        Africa         16320537
##     X2020.Population X2015.Population X2010.Population X2000.Population
## 225         33526656         30949417         28614227         24925554
## 226           311685           276438           245453           192074
## 227              520              564              596              651
## 228         28490453         30529716         28715022         24427729
## 229         96648685         92191398         87411012         79001142
## 230            11655            12182            13142            14723
## 231           556048           491824           413296           270375
## 232         32284046         28516545         24743946         18628700
## 233         18927715         16248230         13792086          9891136
## 234         15669666         14154937         12839771         11834676
##     X1990.Population X1980.Population X1970.Population Area..kmÂ..
## 225         20579100         15947129         12011361      447400
## 226           150882           118156            87019       12189
## 227              700              733              752           1
## 228         19750579         15210443         11355475      916445
## 229         66912613         52968270         41928849      331212
## 230            13454            11315             9377         142
## 231           178529           116775            76371      266000
## 232         13375121          9204938          6843607      527968
## 233          7686401          5720438          4281671      752612
## 234         10113893          7049926          5202918      390757
##     Density..per.kmÂ.. Growth.Rate World.Population.Percentage
## 225            77.3975      1.0160                        0.43
## 226            26.8061      1.0238                        0.00
## 227           510.0000      0.9980                        0.00
## 228            30.8820      1.0036                        0.35
## 229           296.4472      1.0074                        1.23
## 230            81.4930      0.9953                        0.00
```

```
## 231             2.1654      1.0184                          0.01
## 232            63.8232      1.0217                          0.42
## 233            26.5976      1.0280                          0.25
## 234            41.7665      1.0204                          0.20
```

```r
tail(CO2_emission)
```

```
## # A tibble: 6 x 35
##   `Country Name` country_code Region       `Indicator Name` `1990` `1991` `1992`
##   <chr>          <chr>        <chr>        <chr>             <dbl>  <dbl>  <dbl>
## 1 Vanuatu        VUT          East Asia &~ CO2 emissions (~  0.478  0.464  0.387
## 2 Samoa          WSM          East Asia &~ CO2 emissions (~  0.553  0.610  0.604
## 3 Yemen, Rep.    YEM          Middle East~ CO2 emissions (~  0.567  0.691  0.705
## 4 South Africa   ZAF          Sub-Saharan~ CO2 emissions (~  6.73   6.42   6.18
## 5 Zambia         ZMB          Sub-Saharan~ CO2 emissions (~  0.341  0.349  0.337
## 6 Zimbabwe       ZWE          Sub-Saharan~ CO2 emissions (~  1.59   1.71   1.69
## # i 28 more variables: `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...34` <dbl>, `2019...35` <dbl>
```

```r
# checking data type of my dataset (world population)
class(world_population)
```

```
## [1] "data.frame"
```

```r
class(CO2_emission)
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

```r
# structure: Displays each variable's type (numeric, character, factor, etc.) and a sampleof the data.
str(world_population)
```

```
## 'data.frame':    234 obs. of  17 variables:
##  $ Rank                       : int  36 138 34 213 203 42 224 201 33 140 ...
##  $ CCA3                       : chr  "AFG" "ALB" "DZA" "ASM" ...
##  $ Country.Territory          : chr  "Afghanistan" "Albania" "Algeria" "American Samoa" ...
##  $ Capital                    : chr  "Kabul" "Tirana" "Algiers" "Pago Pago" ...
##  $ Continent                  : chr  "Asia" "Europe" "Africa" "Oceania" ...
##  $ X2022.Population           : int  41128771 2842321 44903225 44273 79824 35588987 15857 93763 45510
##  $ X2020.Population           : int  38972230 2866849 43451666 46189 77700 33428485 15585 92664 45030
##  $ X2015.Population           : int  33753499 2882481 39543154 51368 71746 28127721 14525 89941 43257
##  $ X2010.Population           : int  28189672 2913399 35856344 54849 71519 23364185 13172 85695 41100
##  $ X2000.Population           : int  19542982 3182021 30774621 58230 66097 16394062 11047 75055 37070
##  $ X1990.Population           : int  10694796 3295066 25518074 47818 53569 11828638 8316 63328 326370
##  $ X1980.Population           : int  12486631 2941651 18739378 32886 35611 8330047 6560 64888 2802480
##  $ X1970.Population           : int  10752971 2324731 13795915 27075 19860 6029700 6283 64516 2384280
##  $ Area..kmÂ..                 : int  652230 28748 2381741 199 468 1246700 91 442 2780400 29743 ...
##  $ Density..per.kmÂ..         : num  63.1 98.9 18.9 222.5 170.6 ...
##  $ Growth.Rate                : num  1.026 0.996 1.016 0.983 1.01 ...
##  $ World.Population.Percentage: num  0.52 0.04 0.56 0 0 0.45 0 0 0.57 0.03 ...
```

```r
str(CO2_emission)
```

```
## spc_tbl_ [215 x 35] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Country Name  : chr [1:215] "Aruba" "Afghanistan" "Angola" "Albania" ...
##  $ country_code  : chr [1:215] "ABW" "AFG" "AGO" "ALB" ...
##  $ Region        : chr [1:215] "Latin America & Caribbean" "South Asia" "Sub-Saharan Africa" "Europe
##  $ Indicator Name: chr [1:215] "CO2 emissions (metric tons per capita)" "CO2 emissions (metric tons p
##  $ 1990          : num [1:215] NA 0.192 0.554 1.82 7.522 ...
##  $ 1991          : num [1:215] NA 0.168 0.545 1.243 7.235 ...
##  $ 1992          : num [1:215] NA 0.096 0.544 0.684 6.963 ...
##  $ 1993          : num [1:215] NA 0.0847 0.709 0.6383 6.7242 ...
##  $ 1994          : num [1:215] NA 0.0755 0.8368 0.6454 6.5416 ...
##  $ 1995          : num [1:215] NA 0.0685 0.9121 0.6054 6.7335 ...
##  $ 1996          : num [1:215] NA 0.0626 1.0722 0.6124 6.9916 ...
##  $ 1997          : num [1:215] NA 0.0568 1.0866 0.4669 7.3074 ...
##  $ 1998          : num [1:215] NA 0.0527 1.0918 0.5722 7.6395 ...
##  $ 1999          : num [1:215] NA 0.0402 1.1099 0.9554 7.9232 ...
##  $ 2000          : num [1:215] NA 0.0366 0.9881 1.0262 7.9523 ...
##  $ 2001          : num [1:215] NA 0.0338 0.9418 1.0555 7.7215 ...
##  $ 2002          : num [1:215] NA 0.0456 0.8956 1.2324 7.5662 ...
##  $ 2003          : num [1:215] NA 0.0515 0.9249 1.339 7.2424 ...
##  $ 2004          : num [1:215] NA 0.0417 0.9303 1.4041 7.3443 ...
##  $ 2005          : num [1:215] NA 0.0604 0.8135 1.3382 7.3538 ...
##  $ 2006          : num [1:215] NA 0.0666 0.8218 1.34 6.7905 ...
##  $ 2007          : num [1:215] NA 0.0653 0.8118 1.3939 6.531 ...
##  $ 2008          : num [1:215] NA 0.128 0.889 1.384 6.439 ...
##  $ 2009          : num [1:215] NA 0.172 0.939 1.441 6.157 ...
##  $ 2010          : num [1:215] NA 0.244 0.976 1.528 6.157 ...
##  $ 2011          : num [1:215] NA 0.297 0.986 1.669 5.851 ...
##  $ 2012          : num [1:215] NA 0.259 0.951 1.503 5.945 ...
##  $ 2013          : num [1:215] NA 0.186 1.036 1.534 5.943 ...
##  $ 2014          : num [1:215] NA 0.146 1.1 1.668 5.807 ...
##  $ 2015          : num [1:215] NA 0.173 1.135 1.604 6.026 ...
##  $ 2016          : num [1:215] NA 0.15 1.03 1.56 6.08 ...
##  $ 2017          : num [1:215] NA 0.132 0.813 1.789 6.104 ...
##  $ 2018          : num [1:215] NA 0.163 0.778 1.783 6.363 ...
##  $ 2019...34     : num [1:215] NA 0.16 0.792 1.692 6.481 ...
##  $ 2019...35     : num [1:215] NA 0.16 0.792 1.692 6.481 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `Country Name` = col_character(),
##   ..   country_code = col_character(),
##   ..   Region = col_character(),
##   ..   `Indicator Name` = col_character(),
##   ..   `1990` = col_double(),
##   ..   `1991` = col_double(),
##   ..   `1992` = col_double(),
##   ..   `1993` = col_double(),
##   ..   `1994` = col_double(),
##   ..   `1995` = col_double(),
##   ..   `1996` = col_double(),
##   ..   `1997` = col_double(),
##   ..   `1998` = col_double(),
```

```
##   ..   `1999` = col_double(),
##   ..   `2000` = col_double(),
##   ..   `2001` = col_double(),
##   ..   `2002` = col_double(),
##   ..   `2003` = col_double(),
##   ..   `2004` = col_double(),
##   ..   `2005` = col_double(),
##   ..   `2006` = col_double(),
##   ..   `2007` = col_double(),
##   ..   `2008` = col_double(),
##   ..   `2009` = col_double(),
##   ..   `2010` = col_double(),
##   ..   `2011` = col_double(),
##   ..   `2012` = col_double(),
##   ..   `2013` = col_double(),
##   ..   `2014` = col_double(),
##   ..   `2015` = col_double(),
##   ..   `2016` = col_double(),
##   ..   `2017` = col_double(),
##   ..   `2018` = col_double(),
##   ..   `2019...34` = col_double(),
##   ..   `2019...35` = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

The shape of the dataset [Use dim(), nrow(), and ncol()]:

```r
# this code is used returns both rows and columns
dim(world_population)
```

```
## [1] 234  17
```

```r
dim(CO2_emission)
```

```
## [1] 215  35
```

```r
# checking duplicate in world population
sum(duplicated(world_population))
```

```
## [1] 0
```

```r
sum(duplicated(CO2_emission))
```

```
## [1] 0
```

```r
# checking missing values in each column
colSums(is.na(world_population))
```

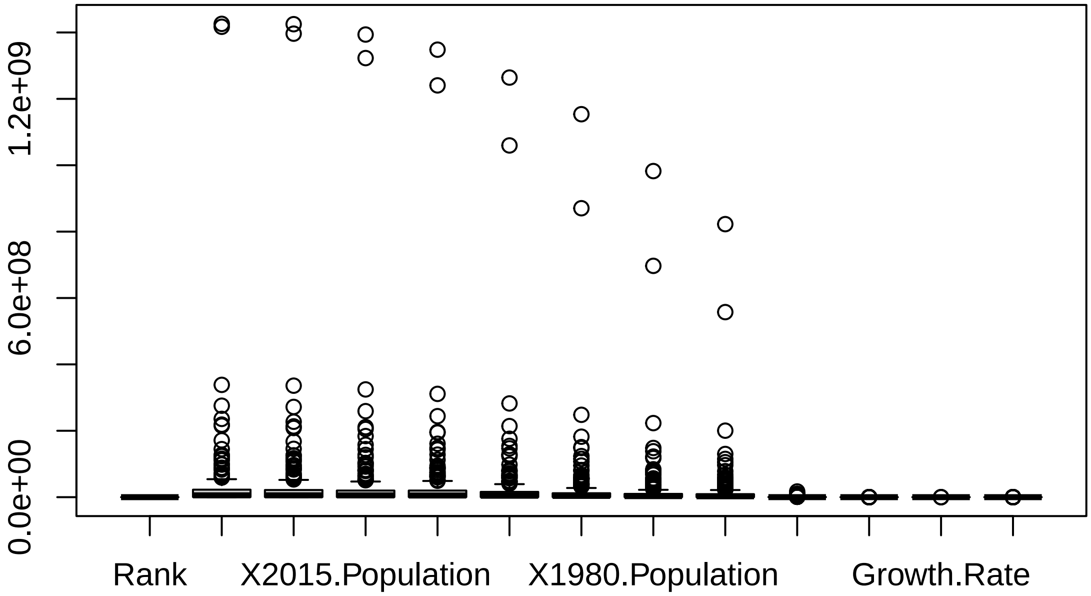
```
##                          Rank                          CCA3
##                             0                             0
```

```
##           Country.Territory                     Capital
##                           0                           0
##                   Continent             X2022.Population
##                           0                           0
##             X2020.Population             X2015.Population
##                           0                           0
##             X2010.Population             X2000.Population
##                           0                           0
##             X1990.Population             X1980.Population
##                           0                           0
##             X1970.Population                   Area..kmÂ..
##                           0                           0
##          Density..per.kmÂ..                 Growth.Rate
##                           0                           0
## World.Population.Percentage
##                           0
```

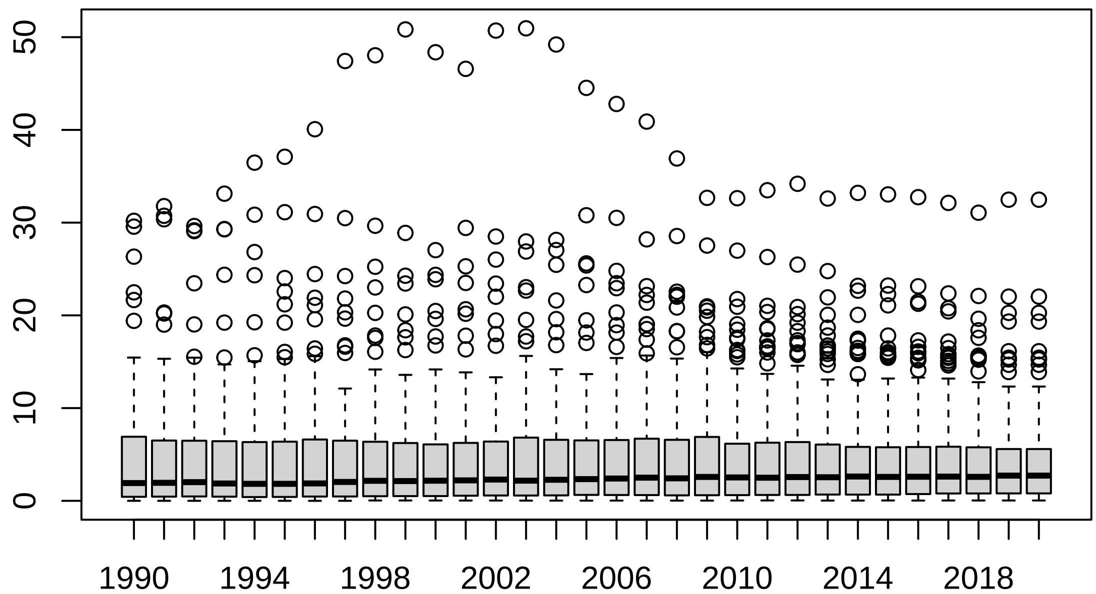
Use box plot to check if there are any outliers in the quantitative variables

```
# checking autlier using boxplot in the quantitative variable
world_population_numeric<-world_population[
  sapply(world_population,is.numeric)]

boxplot(world_population_numeric)
```

```
## Warning in x[floor(d)] + x[ceiling(d)]: NAs produced by integer overflow
## Warning in x[floor(d)] + x[ceiling(d)]: NAs produced by integer overflow
## Warning in x[floor(d)] + x[ceiling(d)]: NAs produced by integer overflow
## Warning in x[floor(d)] + x[ceiling(d)]: NAs produced by integer overflow
## Warning in x[floor(d)] + x[ceiling(d)]: NAs produced by integer overflow
## Warning in x[floor(d)] + x[ceiling(d)]: NAs produced by integer overflow
```




```r
CO2_emission_numeric<- CO2_emission[sapply(CO2_emission,is.numeric)]
boxplot(CO2_emission_numeric)
```



As the quantitative variables are not on the same scales, we have to normalize them into values between 0 and 1, to make make on the same scales.

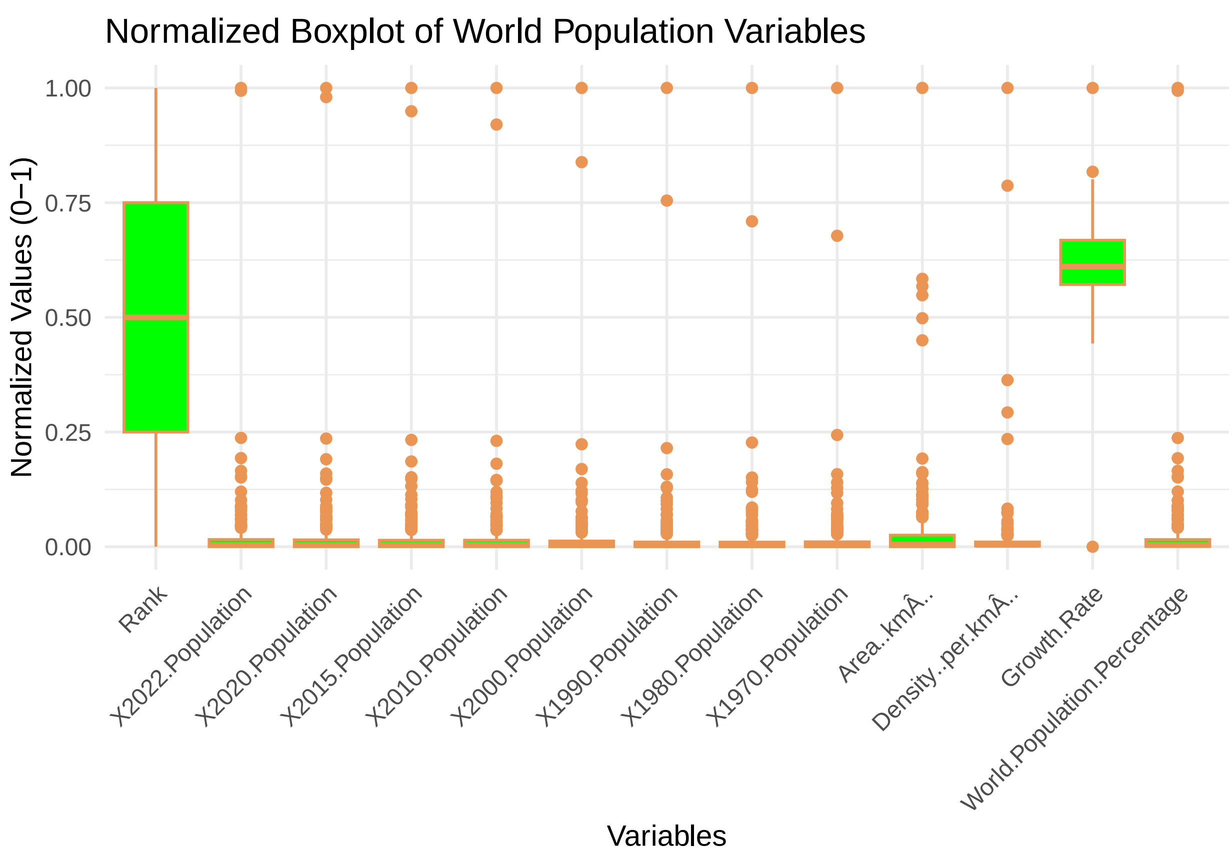
```
library(ggplot2)
library(reshape2)
```

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

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

```
# Select only numeric columns

numeric_info <- world_population[sapply(world_population, is.numeric)]

# Normalize numeric columns (0-1 scaling)

numeric_info_norm <- as.data.frame(lapply(numeric_info, function(x) {(x - min(x, na.rm = TRUE)) / (max(

# Convert to long format for ggplot
long_data <- melt(numeric_info_norm, variable.name = "Variable",
                  value.name = "Value")
```

```
## No id variables; using all as measure variables
```

```r
## No id variables; using all as measure variables
# Create the boxplot

ggplot(long_data, aes(x = Variable, y = Value)) +
geom_boxplot(fill = "green", color = "#eb9555") +
theme_minimal() +
labs(
title = "Normalized Boxplot of World Population Variables", x = "Variables", y = "Normalized Values (0-
) + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

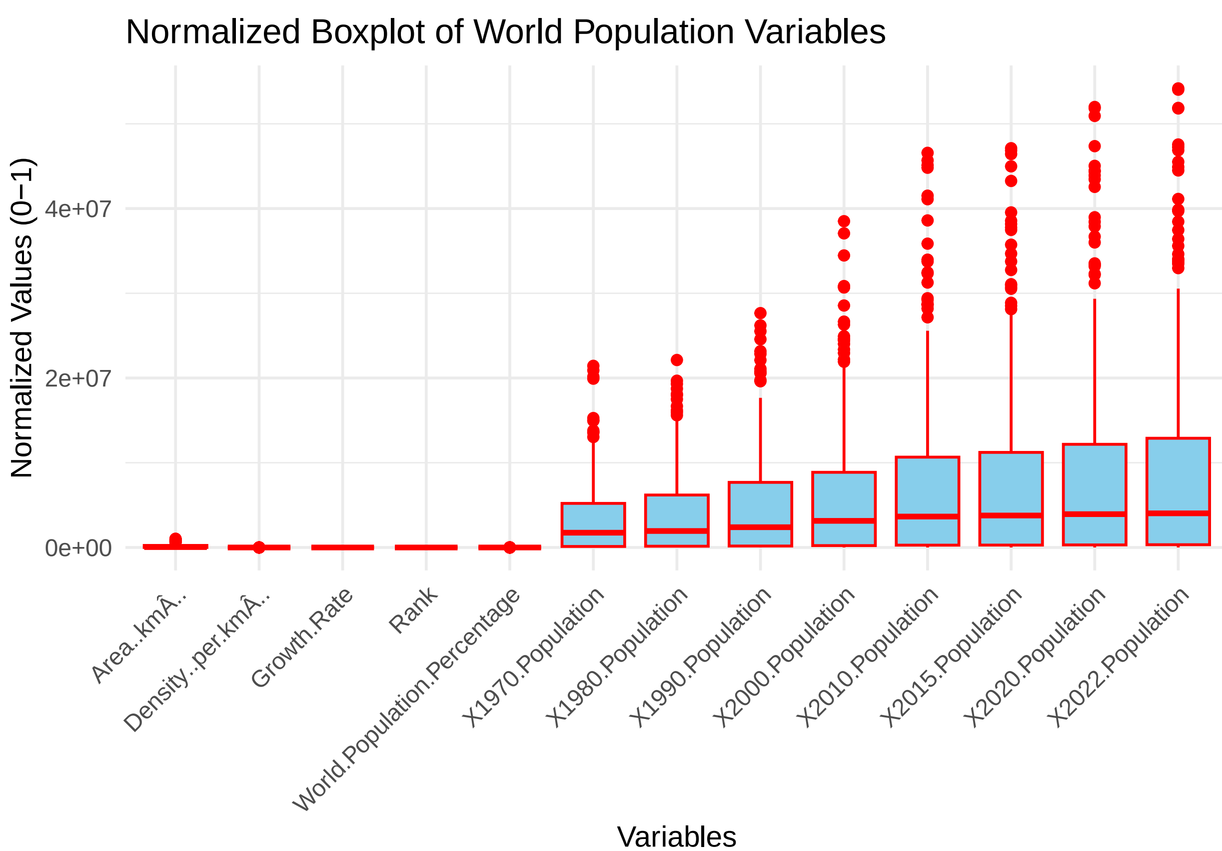



```r
## b. Removal of outliers
# Define a function to remove outliers using the Interquartile Range (IQR) method
World_Population_remove_outliers <- function(x) {
# Step 1: Calculate the first quartile (Q1) - the 25th percentile of the data
Q1 <- quantile(x, 0.25, na.rm = TRUE)
# Step 2: Calculate the third quartile (Q3) - the 75th percentile of the data
Q3 <- quantile(x, 0.75, na.rm = TRUE)
# Step 3: Compute the interquartile range (IQR = Q3 - Q1)
# This shows how spread out the middle 50% of values are
IQR <- Q3 - Q1
# Step 4: Keep only the values within the normal range:
# Values below (Q1 - 1.5 * IQR) or above (Q3 + 1.5 * IQR) are considered outliers
# These extreme values are removed from the result

x[x >= (Q1 - 1.5*IQR) & x <= (Q3 + 1.5*IQR)] }
```

```r
# Apply outlier removal to each column
World_Population_quant_without_outliers <- lapply(numeric_info, World_Population_remove_outliers)
```

```r
# Reshape data for ggplot
values <- unlist(World_Population_quant_without_outliers) # all numeric values
ind <- rep(names(World_Population_quant_without_outliers),# variable names repeated
sapply(World_Population_quant_without_outliers, length))
World_Population_quant_Final_reshaped <- data.frame(values, ind)# final long-format variable names repe
```

```r
# Create boxplot using ggplot2

ggplot(World_Population_quant_Final_reshaped, aes(x = ind, y = values)) +
geom_boxplot(fill = "skyblue", color = "red") + # boxplot style
theme_minimal() + # clean theme
labs(
title = "Normalized Boxplot of World Population Variables", x = "Variables", y = "Normalized Values (0-
) +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
names(world_population)
```

```
##  [1] "Rank"                        "CCA3"
##  [3] "Country.Territory"           "Capital"
##  [5] "Continent"                   "X2022.Population"
```

```
##  [7] "X2020.Population"            "X2015.Population"
##  [9] "X2010.Population"            "X2000.Population"
## [11] "X1990.Population"            "X1980.Population"
## [13] "X1970.Population"            "Area..kmÂ.."
## [15] "Density..per.kmÂ.."          "Growth.Rate"
## [17] "World.Population.Percentage"
```

```
time_var<-select(world_population,Country.Territory,X2022.Population,Growth.Rate)
head(time_var)
```

```
##   Country.Territory X2022.Population Growth.Rate
## 1       Afghanistan         41128771      1.0257
## 2           Albania          2842321      0.9957
## 3           Algeria         44903225      1.0164
## 4    American Samoa            44273      0.9831
## 5           Andorra            79824      1.0100
## 6            Angola         35588987      1.0315
```

4.3 Generating new Variable by using World Population Dataset

```
# Generating new Variable by using World Population Dataset
exponential_growth <- function(P, r, t) {
Population_2030 <- P * exp(r * t)
return(Population_2030)
}
# adding new variable by using mutate function
population_data <- time_var %>%
mutate(Population_2030 = X2022.Population * exp((Growth.Rate/100) * 8))
head(population_data)
```

```
##   Country.Territory X2022.Population Growth.Rate Population_2030
## 1       Afghanistan         41128771      1.0257     44645963.54
## 2           Albania          2842321      0.9957      3077990.57
## 3           Algeria         44903225      1.0164     48706944.55
## 4    American Samoa            44273      0.9831        47895.57
## 5           Andorra            79824      1.0100        86541.51
## 6            Angola         35588987      1.0315     38650365.67
```

4.4 Value extraction and plot

Based on 2022 population, extract top ten countries with high population number

```
attach(time_var)
detach(time_var)
top10_countries<-time_var%>%
arrange(desc(X2022.Population))
head(top10_countries,10)
```

```
##    Country.Territory X2022.Population Growth.Rate
```

```
## 1              China       1425887337      1.0000
## 2              India       1417173173      1.0068
## 3      United States        338289857      1.0038
## 4          Indonesia        275501339      1.0064
## 5           Pakistan        235824862      1.0191
## 6            Nigeria        218541212      1.0241
## 7             Brazil        215313498      1.0046
## 8         Bangladesh        171186372      1.0108
## 9             Russia        144713314      0.9973
## 10            Mexico        127504125      1.0063
```

```r
top10 <-head(top10_countries,10)
top10
```

```
##    Country.Territory X2022.Population Growth.Rate
## 1              China       1425887337      1.0000
## 2              India       1417173173      1.0068
## 3      United States        338289857      1.0038
## 4          Indonesia        275501339      1.0064
## 5           Pakistan        235824862      1.0191
## 6            Nigeria        218541212      1.0241
## 7             Brazil        215313498      1.0046
## 8         Bangladesh        171186372      1.0108
## 9             Russia        144713314      0.9973
## 10            Mexico        127504125      1.0063
```

Use an appropriate graph to present top 10 most populous counties and their population number during 2022.

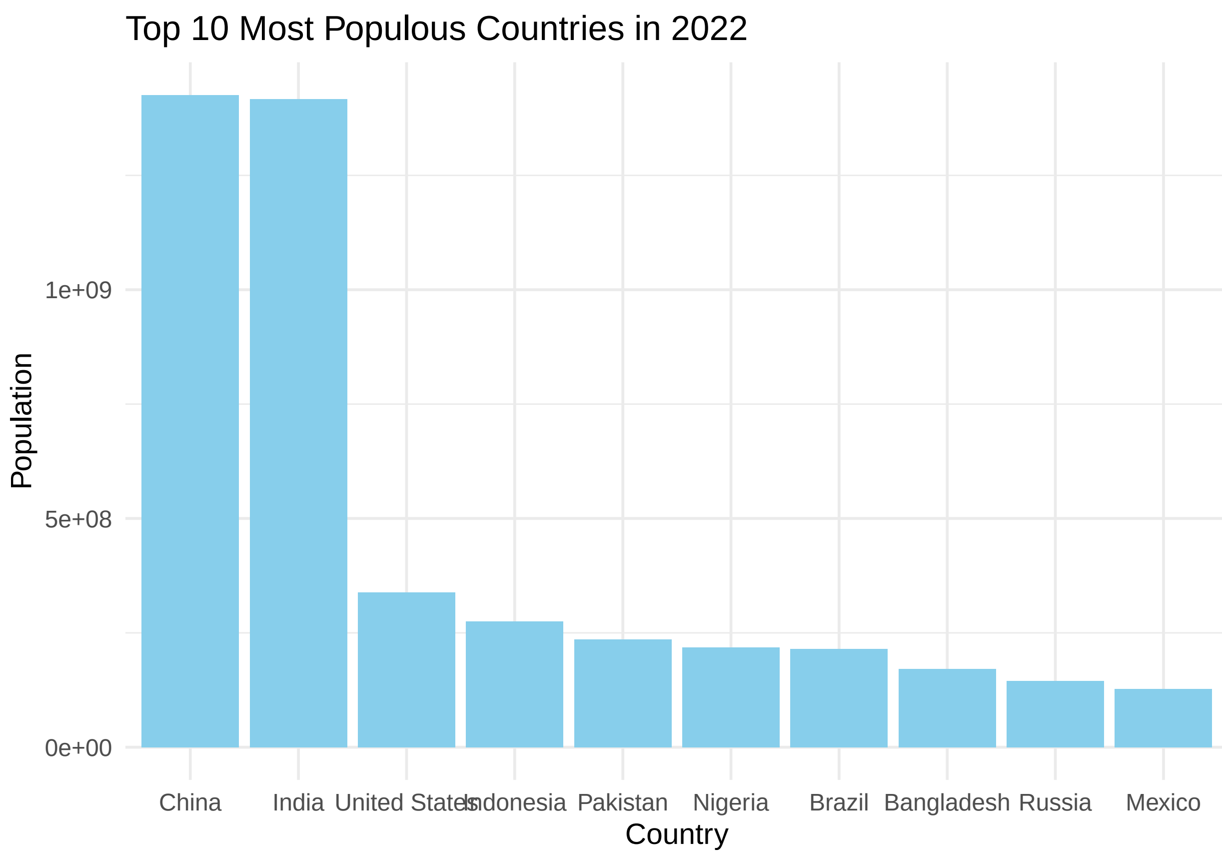
```r
# Reorder Country.Territory by descending population
top10$Country.Territory <- factor(
top10$Country.Territory, levels = top10$Country.Territory[order(top10$X2022.Population, decreasing = FA
)
```

2. Bar Plot: Top 10 Most Populous Countries in 2022

```r
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
ggplot(top10, aes(x = reorder(Country.Territory,-X2022.Population),
                  y =X2022.Population)) +
  geom_col(fill = "skyblue") +
theme_minimal() +

labs(title = "Top 10 Most Populous Countries in 2022", x= "Country", y= "Population")
```



Show the trend in their population number since 1990-2022 by using appropriate graph.

```
names(world_population)
```

```
##  [1] "Rank"                          "CCA3"
##  [3] "Country.Territory"             "Capital"
##  [5] "Continent"                     "X2022.Population"
##  [7] "X2020.Population"              "X2015.Population"
##  [9] "X2010.Population"              "X2000.Population"
## [11] "X1990.Population"              "X1980.Population"
## [13] "X1970.Population"              "Area..kmÂ.."
## [15] "Density..per.kmÂ.."            "Growth.Rate"
## [17] "World.Population.Percentage"
```

```
# Show the trend in their population number since 1990-2022 by using appro priate graph.
world_pop1990_2022<-world_population[,c(
"Country.Territory",
"X1990.Population",
"X2000.Population",
"X2010.Population",
"X2015.Population",
"X2020.Population",
"X2022.Population")]

head(world_pop1990_2022)
```

```
##   Country.Territory X1990.Population X2000.Population X2010.Population
## 1       Afghanistan         10694796         19542982         28189672
## 2           Albania          3295066          3182021          2913399
## 3           Algeria         25518074         30774621         35856344
## 4    American Samoa            47818            58230            54849
## 5           Andorra            53569            66097            71519
## 6            Angola         11828638         16394062         23364185
##   X2015.Population X2020.Population X2022.Population
## 1         33753499         38972230         41128771
## 2          2882481          2866849          2842321
## 3         39543154         43451666         44903225
## 4            51368            46189            44273
## 5            71746            77700            79824
## 6         28127721         33428485         35588987
```

```
top10_pop1990_2022 <- world_pop1990_2022[world_pop1990_2022$Country %in% c
("Mexico", "Russia", "Bangladesh", "Brazil", "Nigeria","Pakistan","China","India","Unit
ed States","Indonesia" ), ]
top10_pop1990_2022
```

```
##     Country.Territory X1990.Population X2000.Population X2010.Population
## 17         Bangladesh        107147651        129193327        148391139
## 28             Brazil        150706446        175873720        196353492
## 42              China       1153704252       1264099069       1348191368
## 93              India        870452165       1059633675       1240613620
## 94          Indonesia        182159874        214072421        244016173
## 132            Mexico         81720428         97873442        112532401
## 150           Nigeria         95214257        122851984        160952853
## 157          Pakistan        115414069        154369924        194454498
## 172            Russia        148005704        146844839        143242599
##     X2015.Population X2020.Population X2022.Population
## 17         157830000        167420951        171186372
## 28         205188205        213196304        215313498
## 42        1393715448       1424929781       1425887337
## 93        1322866505       1396387127       1417173173
## 94         259091970        271857970        275501339
## 132        120149897        125998302        127504125
## 150        183995785        208327405        218541212
## 157        210969298        227196741        235824862
## 172        144668389        145617329        144713314
```

2. Show the trend in their population number since 1990-2022 by using appropriate graph.

```
# Load library
library(tidyverse)
```

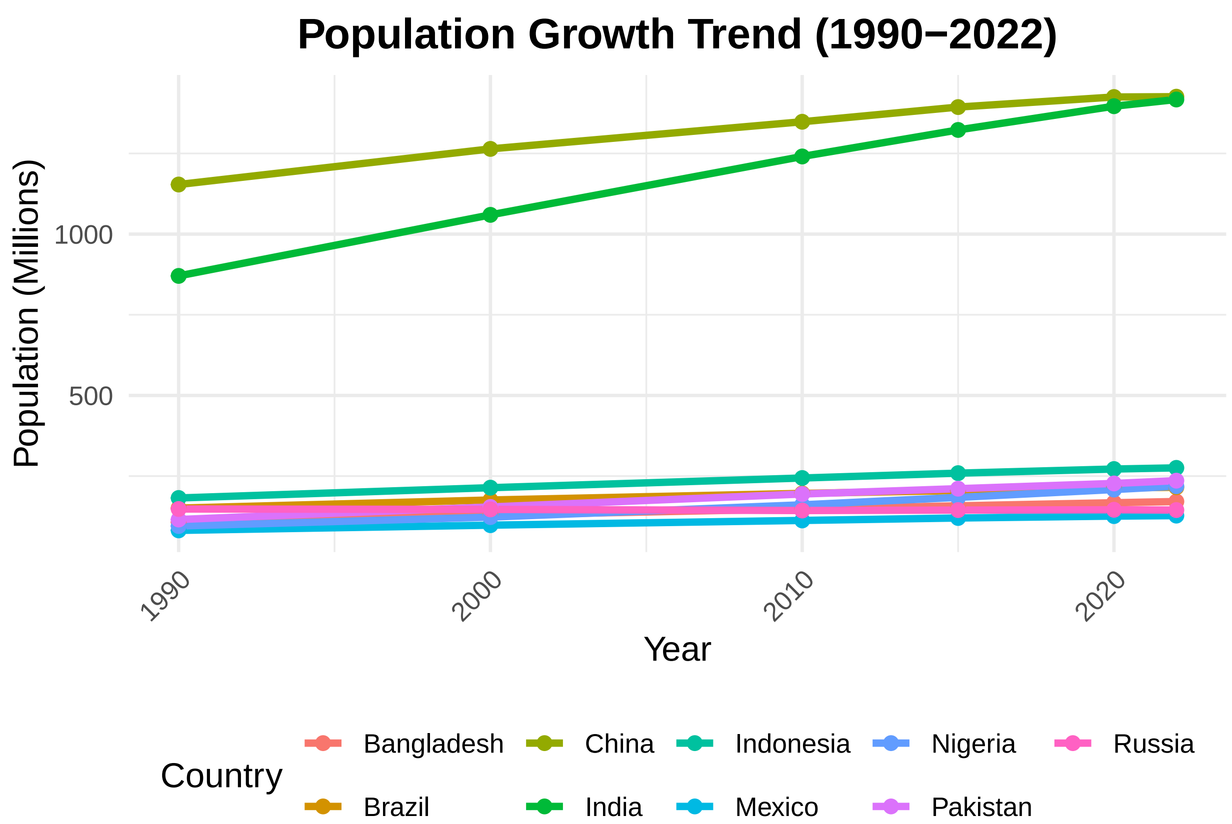
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.1     v stringr   1.5.2
## v lubridate 1.9.4     v tibble    3.3.0
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x scales::col_factor() masks readr::col_factor()
```

```
## x purrr::discard()     masks scales::discard()
## x dplyr::filter()      masks stats::filter()
## x dplyr::lag()         masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
# Our Dataset
data <- top10_pop1990_2022

# Convert from wide to long format
data_long <- data %>%
pivot_longer(
cols = starts_with("X"), names_to = "Year", values_to = "Population"
) %>%
mutate(Year = as.numeric(gsub("X|\\.Population", "", Year)), Population_Millions = Population / 1e6) #
```

```
# Plot population trend

ggplot(data_long, aes(x = Year, y = Population_Millions, color = Country.Territory)) +
geom_line(size = 1.3) + geom_point(size = 2) +
labs(title = "Population Growth Trend (1990-2022)", x = "Year", y = "Population (Millions)", color = "C
theme_minimal(base_size = 13) +
theme(
plot.title = element_text(hjust = 0.5, size = 16, face = "bold"), axis.text.x = element_text(angle = 45
legend.position = "bottom"
)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Emissions Around the World dataset to extract emission of 10 most populous countries. Use an appropriate graph to show their emission trend since 1990-2019.

```
head(CO2_emission)
```

```
## # A tibble: 6 x 35
##   `Country Name`       country_code Region `Indicator Name` `1990` `1991`  `1992`
##   <chr>                <chr>        <chr>  <chr>             <dbl>  <dbl>   <dbl>
## 1 Aruba                ABW          Latin~ CO2 emissions (~ NA     NA      NA
## 2 Afghanistan          AFG          South~ CO2 emissions (~  0.192  0.168  0.0960
## 3 Angola               AGO          Sub-S~ CO2 emissions (~  0.554  0.545  0.544
## 4 Albania              ALB          Europ~ CO2 emissions (~  1.82   1.24   0.684
## 5 Andorra              AND          Europ~ CO2 emissions (~  7.52   7.24   6.96
## 6 United Arab Emirat~ ARE          Middl~ CO2 emissions (~ 30.2   31.8   29.1
## # i 28 more variables: `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...34` <dbl>, `2019...35` <dbl>
```

```r
top10_CO2_emission <- CO2_emission[CO2_emission$`Country Name` %in% c("Mexic
o", "Bangladesh","Brazil","Nigeria","Pakistan","China","India","United States","Indonesia","Russian Fed
head(top10_CO2_emission)
```

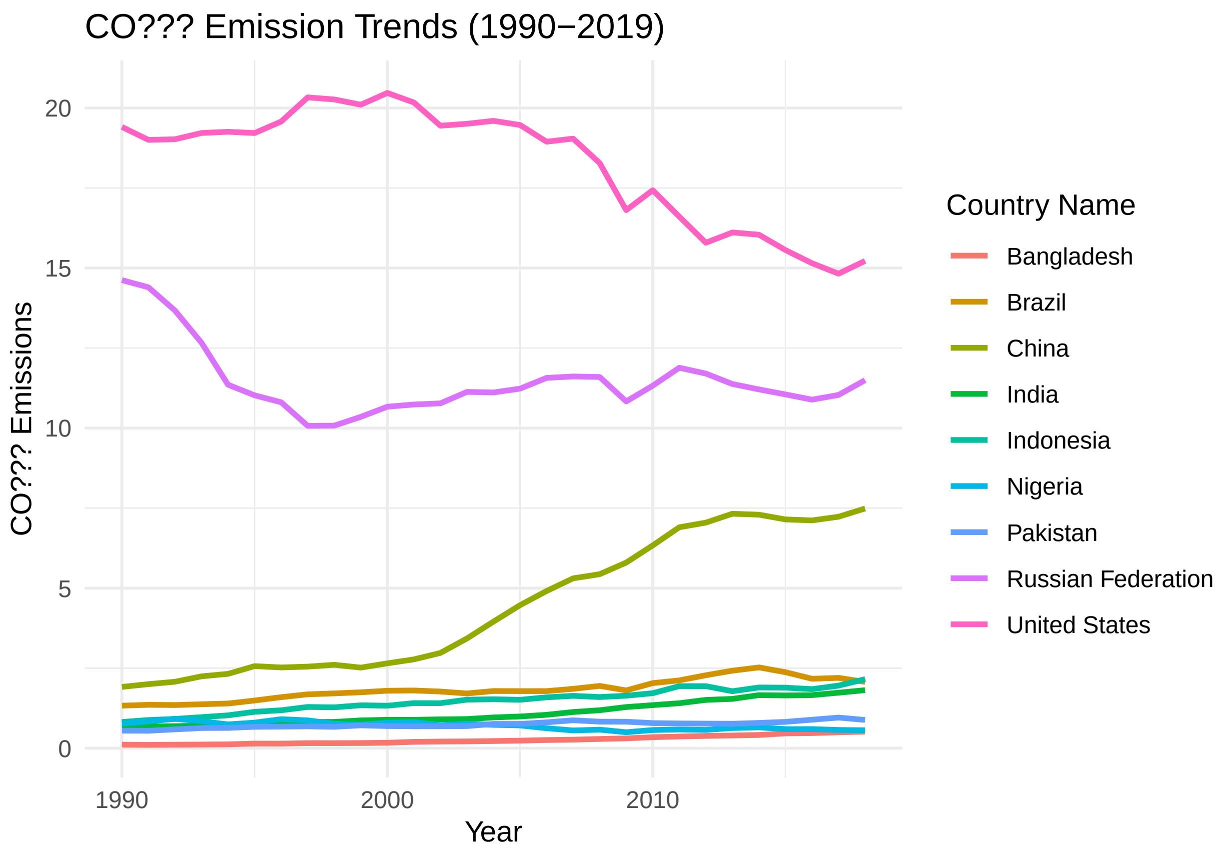
```
## # A tibble: 6 x 35
##   `Country Name` country_code Region       `Indicator Name` `1990` `1991` `1992`
##   <chr>          <chr>        <chr>        <chr>             <dbl>  <dbl>  <dbl>
## 1 Bangladesh     BGD          South Asia   CO2 emissions (~  0.112  0.103  0.109
## 2 Brazil         BRA          Latin Ameri~ CO2 emissions (~  1.33   1.35   1.35
## 3 China          CHN          East Asia &~ CO2 emissions (~  1.91   2.00   2.08
## 4 Indonesia      IDN          East Asia &~ CO2 emissions (~  0.819  0.880  0.914
## 5 India          IND          South Asia   CO2 emissions (~  0.645  0.681  0.689
## 6 Nigeria        NGA          Sub-Saharan~ CO2 emissions (~  0.764  0.839  0.917
## # i 28 more variables: `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...34` <dbl>, `2019...35` <dbl>
```

```r
# Load required libraries

library(tidyverse)# includes ggplot2, dplyr, tidyr,etc.
library(dplyr)



#Change the duplicate column name
names(top10_CO2_emission)[names(top10_CO2_emission)== "2019...34"] <- "2019_A"
names(top10_CO2_emission)[names(top10_CO2_emission)== "2019...35"] <- "2019_B"
data <- top10_CO2_emission

# Reshape data from wide to long
data_long <- data %>%
  pivot_longer(
    cols = matches("^[0-9]{4}"),# selects columns like 1990, 1991, ...
    names_to = "Year",
    values_to = "CO2_Emissions"
  ) %>% mutate(Year = as.numeric(Year))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `Year = as.numeric(Year)`.
## Caused by warning:
## ! NAs introduced by coercion
```

```r
# Plot
ggplot(data_long, aes(x = Year, y = CO2_Emissions, color = `Country Name`)) +
  geom_line(size = 1) +
  labs(title = "CO??? Emission Trends (1990-2019)",
    x = "Year",
    y = "CO??? Emissions"
  ) + theme_minimal()
```

```
## Warning: Removed 18 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



4.5 Correlation Analysis

1. Use the population dataset to find correlations between Area , Density , Growth rate, and World Population Percentage by using both numerical values and heatmap.

2. Merge the World Population Dataset and CO2 Emissions Around the World on country name . After extracting, 2022 population and 2019 CO2 emissions.Using an appropriate methods(visualization, statistical analysis of numerical values,regression,.), determine the pattern of relationships between population number and CO2 emissions.

```r
#Rename the column of both dataset
names(CO2_emission)[names(CO2_emission) == "Country Name"] <- "Country"
names(world_population)[names(world_population) == "Country.Territory"] <- "Country"
names(CO2_emission)[names(CO2_emission)== "2019...34"] <- "2019_A"
names(CO2_emission)[names(CO2_emission)== "2019...35"] <- "2019_B"

# Merge datasets on country name
merged_data <- world_population %>%
inner_join(CO2_emission, by = c("Country" = "Country"))

head(merged_data)
```

```
##    Rank CCA3         Country           Capital Continent X2022.Population
```

```
## 1   36  AFG    Afghanistan           Kabul      Asia         41128771
## 2  138  ALB        Albania          Tirana    Europe          2842321
## 3   34  DZA        Algeria         Algiers    Africa         44903225
## 4  213  ASM American Samoa       Pago Pago   Oceania            44273
## 5  203  AND        Andorra Andorra la Vella    Europe            79824
## 6   42  AGO         Angola          Luanda    Africa         35588987
##   X2020.Population X2015.Population X2010.Population X2000.Population
## 1         38972230         33753499         28189672         19542982
## 2          2866849          2882481          2913399          3182021
## 3         43451666         39543154         35856344         30774621
## 4            46189            51368            54849            58230
## 5            77700            71746            71519            66097
## 6         33428485         28127721         23364185         16394062
##   X1990.Population X1980.Population X1970.Population Area..kmÂ..
## 1         10694796         12486631         10752971      652230
## 2          3295066          2941651          2324731       28748
## 3         25518074         18739378         13795915     2381741
## 4            47818            32886            27075         199
## 5            53569            35611            19860         468
## 6         11828638          8330047          6029700     1246700
##   Density..per.kmÂ.. Growth.Rate World.Population.Percentage country_code
## 1            63.0587      1.0257                        0.52          AFG
## 2            98.8702      0.9957                        0.04          ALB
## 3            18.8531      1.0164                        0.56          DZA
## 4           222.4774      0.9831                        0.00          ASM
## 5           170.5641      1.0100                        0.00          AND
## 6            28.5466      1.0315                        0.45          AGO
##                       Region                           Indicator Name      1990
## 1                 South Asia CO2 emissions (metric tons per capita) 0.1917451
## 2      Europe & Central Asia CO2 emissions (metric tons per capita) 1.8195416
## 3 Middle East & North Africa CO2 emissions (metric tons per capita) 2.4434300
## 4        East Asia & Pacific CO2 emissions (metric tons per capita)        NA
## 5      Europe & Central Asia CO2 emissions (metric tons per capita) 7.5218317
## 6         Sub-Saharan Africa CO2 emissions (metric tons per capita) 0.5536620
##        1991       1992       1993       1994       1995       1996       1997
## 1 0.1676816 0.09595774 0.08472111 0.07554583 0.06846796 0.06258803 0.05682662
## 2 1.2428102 0.68369983 0.63830704 0.64535519 0.60543625 0.61236736 0.46692147
## 3 2.5162433 2.47296078 2.61330374 2.60900907 2.65806257 2.60093353 2.50243923
## 4        NA         NA         NA         NA         NA         NA         NA
## 5 7.2353792 6.96307870 6.72417752 6.54157891 6.73347949 6.99159455 7.30744115
## 6 0.5445386 0.54355722 0.70898423 0.83680440 0.91214149 1.07216847 1.08663697
##         1998       1999      2000       2001       2002       2003       2004
## 1 0.05269086 0.04015697 0.0365737 0.03378536 0.04557366 0.05151838 0.04165539
## 2 0.57215370 0.95535931 1.0262131 1.05549588 1.23237878 1.33898498 1.40405869
## 3 2.47244786 2.53107052 2.5787445 2.50067461 2.58671220 2.73337366 2.73735406
## 4         NA         NA        NA         NA         NA         NA         NA
## 5 7.63953851 7.92319165 7.9522863 7.72154906 7.56623988 7.24241557 7.34426233
## 6 1.09182531 1.10985966 0.9880774 0.94182891 0.89557767 0.92486944 0.93026295
##         2005       2006       2007      2008      2009      2010      2011
## 1 0.06041878 0.06658329 0.06531235 0.1284166 0.1718624 0.2436140 0.2965062
## 2 1.33820940 1.33999574 1.39393137 1.3843112 1.4414936 1.5276237 1.6694232
## 3 2.84135137 2.96691468 3.00728985 3.1024511 3.1745733 3.1736545 3.2947426
## 4         NA         NA         NA        NA        NA        NA        NA
## 5 7.35378001 6.79054277 6.53104692 6.4393039 6.1566875 6.1571978 5.8508861
```

```
## 6 0.81353929 0.82184008 0.81175351 0.8886580 0.9394040 0.9761842 0.9855223
##        2012      2013      2014      2015      2016      2017      2018
## 1 0.2592953 0.1856237 0.1462356 0.1728967 0.1497893 0.1316946 0.1632953
## 2 1.5032405 1.5336300 1.6683374 1.6037751 1.5576644 1.7887861 1.7827389
## 3 3.6093077 3.6449793 3.7956323 3.9334959 3.8200903 3.8256380 3.9201091
## 4        NA        NA        NA        NA        NA        NA        NA
## 5 5.9446542 5.9428004 5.8071277 6.0261818 6.0806003 6.1041339 6.3629754
## 6 0.9506959 1.0362939 1.0997791 1.1350441 1.0318113 0.8133007 0.7776749
##      2019_A    2019_B
## 1 0.1598244 0.1598244
## 2 1.6922483 1.6922483
## 3 3.9776505 3.9776505
## 4        NA        NA
## 5 6.4812174 6.4812174
## 6 0.7921371 0.7921371
```

```r
names(merged_data)
```

```
##  [1] "Rank"                        "CCA3"
##  [3] "Country"                     "Capital"
##  [5] "Continent"                   "X2022.Population"
##  [7] "X2020.Population"            "X2015.Population"
##  [9] "X2010.Population"            "X2000.Population"
## [11] "X1990.Population"            "X1980.Population"
## [13] "X1970.Population"            "Area..kmÂ.."
## [15] "Density..per.kmÂ.."          "Growth.Rate"
## [17] "World.Population.Percentage" "country_code"
## [19] "Region"                      "Indicator Name"
## [21] "1990"                        "1991"
## [23] "1992"                        "1993"
## [25] "1994"                        "1995"
## [27] "1996"                        "1997"
## [29] "1998"                        "1999"
## [31] "2000"                        "2001"
## [33] "2002"                        "2003"
## [35] "2004"                        "2005"
## [37] "2006"                        "2007"
## [39] "2008"                        "2009"
## [41] "2010"                        "2011"
## [43] "2012"                        "2013"
## [45] "2014"                        "2015"
## [47] "2016"                        "2017"
## [49] "2018"                        "2019_A"
## [51] "2019_B"
```

```r
# extract 2022 population and 2019 CO2 emissions
library(dplyr)

extract_data1 <- merged_data %>%
  select(`Country`,`X2022.Population`,`2019_B`)

names(extract_data1)
```

```
## [1] "Country"          "X2022.Population" "2019_B"
```

```r
colSums(is.na(extract_data1))
```

```
##          Country X2022.Population           2019_B
##                0                0               18
```

```r
extract_clean <- na.omit(extract_data1)
head(extract_clean)
```

```
##               Country X2022.Population    2019_B
## 1         Afghanistan         41128771 0.1598244
## 2             Albania          2842321 1.6922483
## 3             Algeria         44903225 3.9776505
## 5             Andorra            79824 6.4812174
## 6              Angola         35588987 0.7921371
## 7 Antigua and Barbuda            93763 5.3544765
```

```r
# Statistical correlation
correlation_value <- cor( extract_clean[["X2022.Population"]], extract_clean[["2019_B"]], method ="pear

print(paste("Correlation between 2022 Population and 2019 CO2 emissions:", round(correlation_value, 3))
```

```
## [1] "Correlation between 2022 Population and 2019 CO2 emissions: 0.008"
```

The Result show no clear relationship between pop and CO2 emmision

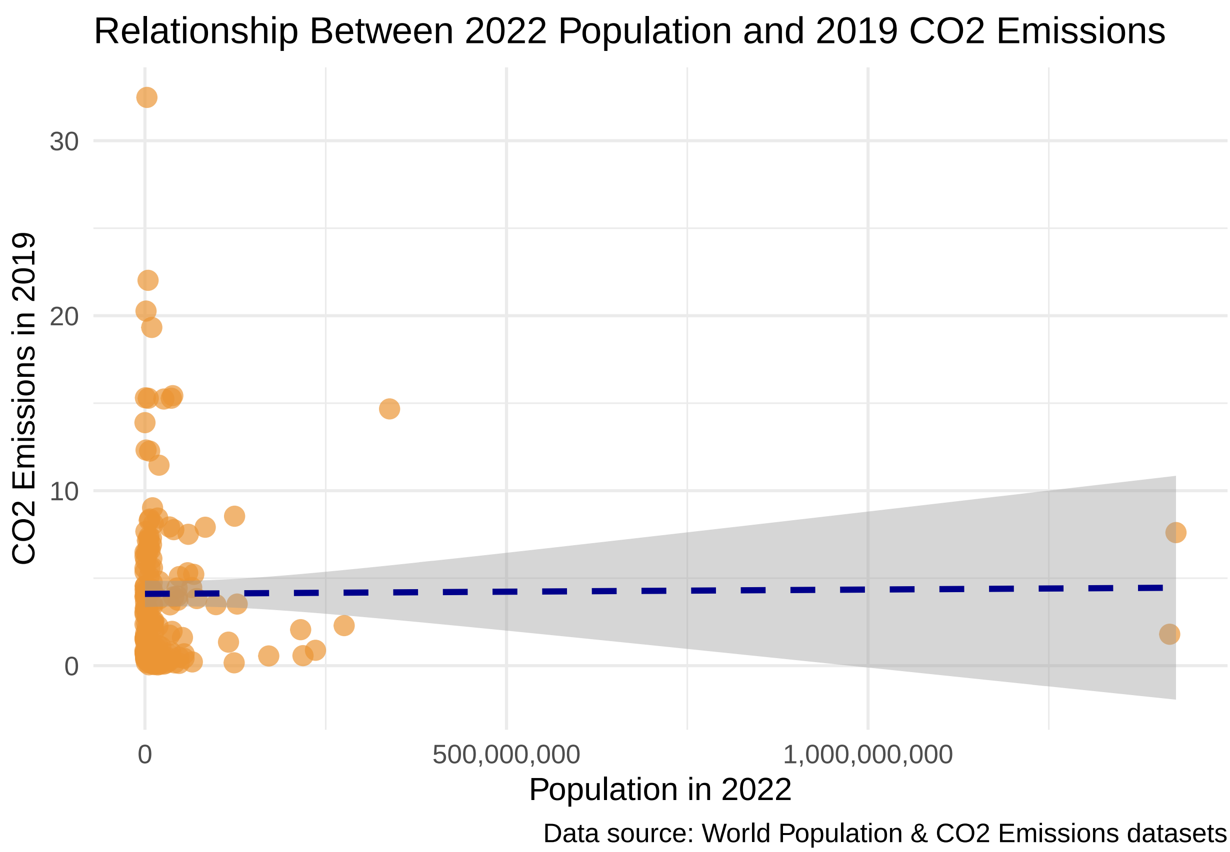
```r
# Simple regression model
model<-lm(`2019_B` ~ X2022.Population, data = extract_clean)
summary(model)
```

```
##
## Call:
## lm(formula = '2019_B' ~ X2022.Population, data = extract_clean)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.065 -3.326 -1.209  1.474 28.369
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.105e+00  3.835e-01  10.703   <2e-16 ***
## X2022.Population 2.449e-10  2.327e-09   0.105    0.916
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.801 on 166 degrees of freedom
## Multiple R-squared:  6.675e-05, Adjusted R-squared:  -0.005957
## F-statistic: 0.01108 on 1 and 166 DF,  p-value: 0.9163
```

Since p-value: 0.9163 the relationship is not statistically significant

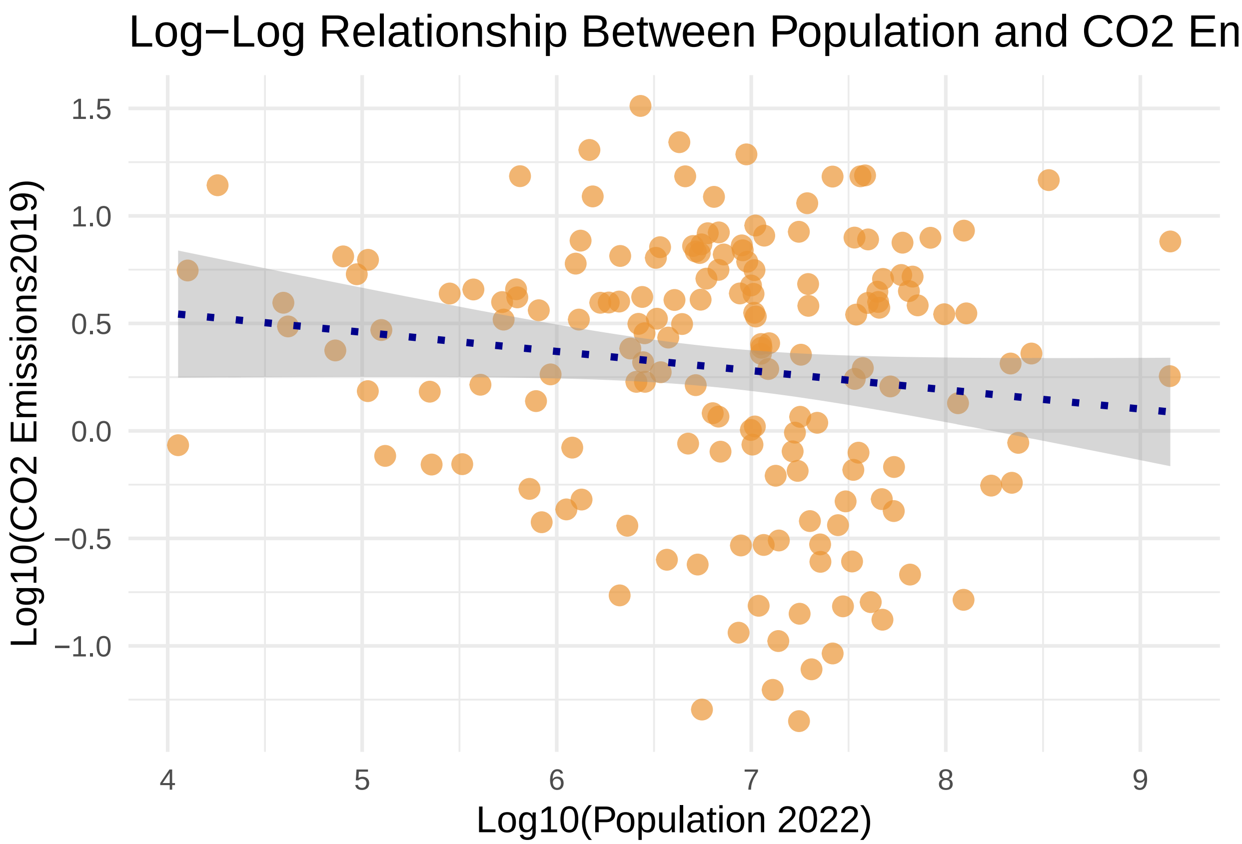
```r
# Step 1: Visualization - Scatter plot with regression line

ggplot(extract_clean, aes(x = X2022.Population, y = `2019_B`)) +
geom_point(color = "#eb9534", size = 3, alpha = 0.7) +
geom_smooth(method = "lm", se = TRUE, color = "darkblue", linetype = "dashed") +
theme_minimal(base_size = 12) +
labs(title = "Relationship Between 2022 Population and 2019 CO2 Emissions",
     x = "Population in 2022", y = "CO2 Emissions in 2019",
     caption = "Data source: World Population & CO2 Emissions datasets"
) +
scale_x_continuous(labels = scales::comma) +
scale_y_continuous(labels = scales::comma)
```

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




```r
ggplot(extract_clean, aes(x = log10(X2022.Population), y = log10(`2019_B`))) +

geom_point(color = "#eb9534", size = 3, alpha = 0.7) +
geom_smooth(method = "lm", se = TRUE, color = "darkblue", linetype = "dotted") +
theme_minimal(base_size = 14) +
labs(
title = "Log-Log Relationship Between Population and CO2 Emissions", x = "Log10(Population 2022)", y =
```

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



#4.6 Comparing CO2 Emissions in continents

```r
head(merged_data)
```

```
##   Rank CCA3        Country          Capital Continent X2022.Population
## 1   36  AFG    Afghanistan            Kabul      Asia         41128771
## 2  138  ALB        Albania           Tirana    Europe          2842321
## 3   34  DZA        Algeria          Algiers    Africa         44903225
## 4  213  ASM American Samoa        Pago Pago   Oceania            44273
## 5  203  AND        Andorra Andorra la Vella    Europe            79824
## 6   42  AGO         Angola           Luanda    Africa         35588987
##   X2020.Population X2015.Population X2010.Population X2000.Population
## 1         38972230         33753499         28189672         19542982
## 2          2866849          2882481          2913399          3182021
## 3         43451666         39543154         35856344         30774621
## 4            46189            51368            54849            58230
## 5            77700            71746            71519            66097
## 6         33428485         28127721         23364185         16394062
##   X1990.Population X1980.Population X1970.Population Area..kmÂ..
## 1         10694796         12486631         10752971      652230
## 2          3295066          2941651          2324731       28748
## 3         25518074         18739378         13795915     2381741
## 4            47818            32886            27075         199
## 5            53569            35611            19860         468
## 6         11828638          8330047          6029700     1246700
##   Density..per.kmÂ.. Growth.Rate World.Population.Percentage country_code
```

```
## 1            63.0587      1.0257                        0.52          AFG
## 2            98.8702      0.9957                        0.04          ALB
## 3            18.8531      1.0164                        0.56          DZA
## 4           222.4774      0.9831                        0.00          ASM
## 5           170.5641      1.0100                        0.00          AND
## 6            28.5466      1.0315                        0.45          AGO
##                      Region                      Indicator Name      1990
## 1                South Asia CO2 emissions (metric tons per capita) 0.1917451
## 2      Europe & Central Asia CO2 emissions (metric tons per capita) 1.8195416
## 3 Middle East & North Africa CO2 emissions (metric tons per capita) 2.4434300
## 4         East Asia & Pacific CO2 emissions (metric tons per capita)        NA
## 5      Europe & Central Asia CO2 emissions (metric tons per capita) 7.5218317
## 6          Sub-Saharan Africa CO2 emissions (metric tons per capita) 0.5536620
##        1991       1992       1993       1994       1995       1996       1997
## 1 0.1676816 0.09595774 0.08472111 0.07554583 0.06846796 0.06258803 0.05682662
## 2 1.2428102 0.68369983 0.63830704 0.64535519 0.60543625 0.61236736 0.46692147
## 3 2.5162433 2.47296078 2.61330374 2.60900907 2.65806257 2.60093353 2.50243923
## 4        NA         NA         NA         NA         NA         NA         NA
## 5 7.2353792 6.96307870 6.72417752 6.54157891 6.73347949 6.99159455 7.30744115
## 6 0.5445386 0.54355722 0.70898423 0.83680440 0.91214149 1.07216847 1.08663697
##         1998       1999      2000       2001       2002       2003       2004
## 1 0.05269086 0.04015697 0.0365737 0.03378536 0.04557366 0.05151838 0.04165539
## 2 0.57215370 0.95535931 1.0262131 1.05549588 1.23237878 1.33898498 1.40405869
## 3 2.47244786 2.53107052 2.5787445 2.50067461 2.58671220 2.73337366 2.73735406
## 4         NA         NA        NA         NA         NA         NA         NA
## 5 7.63953851 7.92319165 7.9522863 7.72154906 7.56623988 7.24241557 7.34426233
## 6 1.09182531 1.10985966 0.9880774 0.94182891 0.89557767 0.92486944 0.93026295
##         2005       2006       2007      2008      2009      2010      2011
## 1 0.06041878 0.06658329 0.06531235 0.1284166 0.1718624 0.2436140 0.2965062
## 2 1.33820940 1.33999574 1.39393137 1.3843112 1.4414936 1.5276237 1.6694232
## 3 2.84135137 2.96691468 3.00728985 3.1024511 3.1745733 3.1736545 3.2947426
## 4         NA         NA         NA        NA        NA        NA        NA
## 5 7.35378001 6.79054277 6.53104692 6.4393039 6.1566875 6.1571978 5.8508861
## 6 0.81353929 0.82184008 0.81175351 0.8886580 0.9394040 0.9761842 0.9855223
##        2012      2013      2014      2015      2016      2017      2018
## 1 0.2592953 0.1856237 0.1462356 0.1728967 0.1497893 0.1316946 0.1632953
## 2 1.5032405 1.5336300 1.6683374 1.6037751 1.5576644 1.7887861 1.7827389
## 3 3.6093077 3.6449793 3.7956323 3.9334959 3.8200903 3.8256380 3.9201091
## 4        NA        NA        NA        NA        NA        NA        NA
## 5 5.9446542 5.9428004 5.8071277 6.0261818 6.0806003 6.1041339 6.3629754
## 6 0.9506959 1.0362939 1.0997791 1.1350441 1.0318113 0.8133007 0.7776749
##      2019_A    2019_B
## 1 0.1598244 0.1598244
## 2 1.6922483 1.6922483
## 3 3.9776505 3.9776505
## 4        NA        NA
## 5 6.4812174 6.4812174
## 6 0.7921371 0.7921371
```

```r
# Load required packages
library(dplyr)
library(ggplot2)
library(readr)
```

```r
# Step 1: Read the dataset

newdata <- merged_data
names(merged_data)
```

```
##  [1] "Rank"                        "CCA3"
##  [3] "Country"                     "Capital"
##  [5] "Continent"                   "X2022.Population"
##  [7] "X2020.Population"            "X2015.Population"
##  [9] "X2010.Population"            "X2000.Population"
## [11] "X1990.Population"            "X1980.Population"
## [13] "X1970.Population"            "Area..kmÂ.."
## [15] "Density..per.kmÂ.."          "Growth.Rate"
## [17] "World.Population.Percentage" "country_code"
## [19] "Region"                      "Indicator Name"
## [21] "1990"                        "1991"
## [23] "1992"                        "1993"
## [25] "1994"                        "1995"
## [27] "1996"                        "1997"
## [29] "1998"                        "1999"
## [31] "2000"                        "2001"
## [33] "2002"                        "2003"
## [35] "2004"                        "2005"
## [37] "2006"                        "2007"
## [39] "2008"                        "2009"
## [41] "2010"                        "2011"
## [43] "2012"                        "2013"
## [45] "2014"                        "2015"
## [47] "2016"                        "2017"
## [49] "2018"                        "2019_A"
## [51] "2019_B"
```

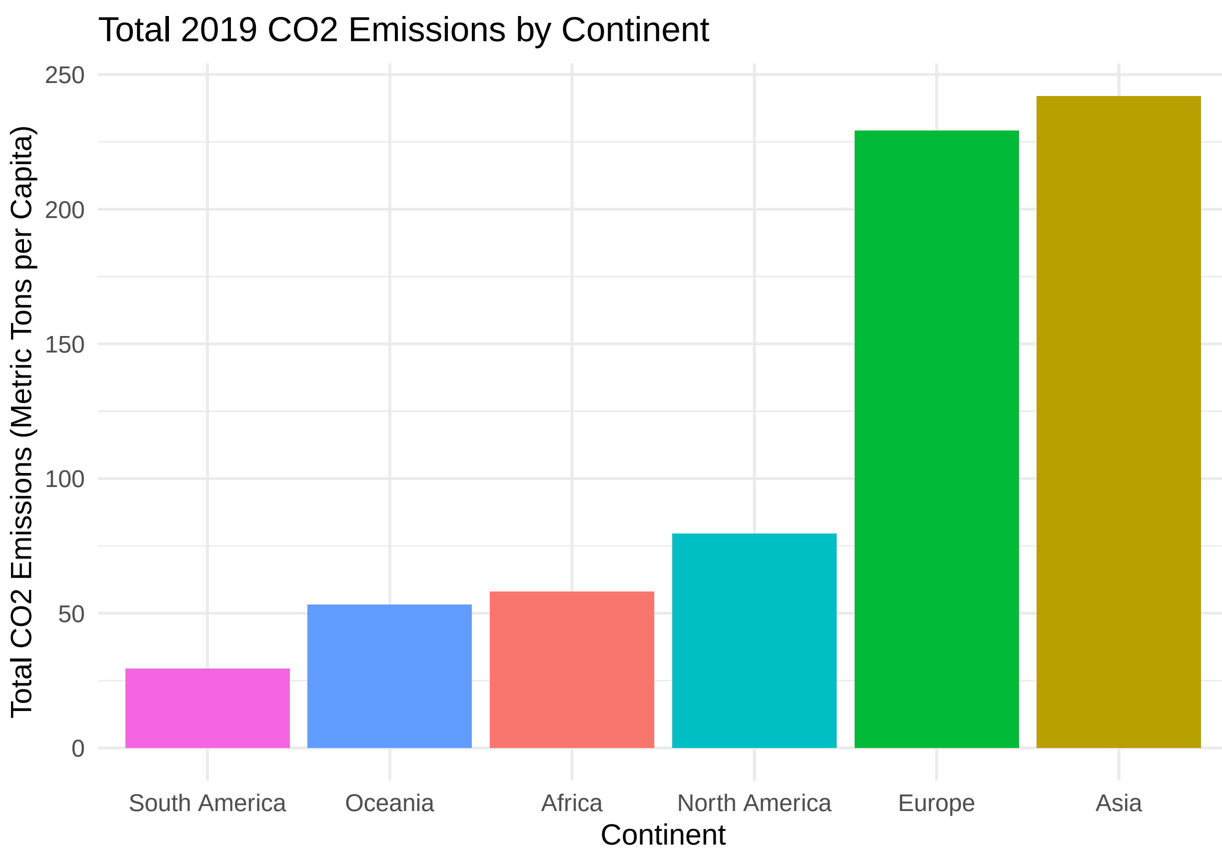
```r
# Step 2: Clean and rename relevant columns
renamed_data <- newdata %>%
rename(Country ="Country", Continent = "Continent", Population_2022 ="X2022.Population", CO2_2019 = "20
)

# Step 3: Remove missing values if any
cleaned_data <- na.omit(newdata)

# Step 4: Summarize total CO2 emissions per continent
continent_summary <- renamed_data %>%
group_by(Continent) %>%
summarise(Total_CO2_2019 = sum(CO2_2019, na.rm = TRUE),
Avg_CO2_per_capita = mean(CO2_2019, na.rm = TRUE),
Total_Population = sum(Population_2022, na.rm = TRUE)
) %>%
  arrange(desc(Total_CO2_2019))
```

```r
# Step 5: Display the summary table
print(continent_summary)
```

```
## # A tibble: 6 x 4
```

```
##   Continent     Total_CO2_2019 Avg_CO2_per_capita Total_Population
##   <chr>                  <dbl>              <dbl>            <dbl>
## 1 Asia                   242.                6.54       4361847775
## 2 Europe                 229.                5.73        592616182
## 3 North America           79.6               4.19        598578924
## 4 Africa                  58.0               1.21       1177424371
## 5 Oceania                 53.2               4.09         44892002
## 6 South America           29.4               2.67        408206575
```

```r
# Step 6: Create a bar chart comparing continents

ggplot(continent_summary, aes(x = reorder(Continent,Total_CO2_2019),
                              y = Total_CO2_2019, fill = Continent)) +
geom_bar(stat = "identity") +
labs(title = "Total 2019 CO2 Emissions by Continent", x = "Continent", y = "Total CO2 Emissions (Metric
theme_minimal() +
theme(legend.position = "none")
```



1. What is the first continent with high 2019 CO2 emission = Europe 2. What is the third continent in terms of emitting CO2? What is their total emission(metric tons per capita) = North America, total emission 3. What is the last continent in terms of emitting CO2? What their total emission(metric tons per capita)? = South America, total emission

```r
colnames(data)
```

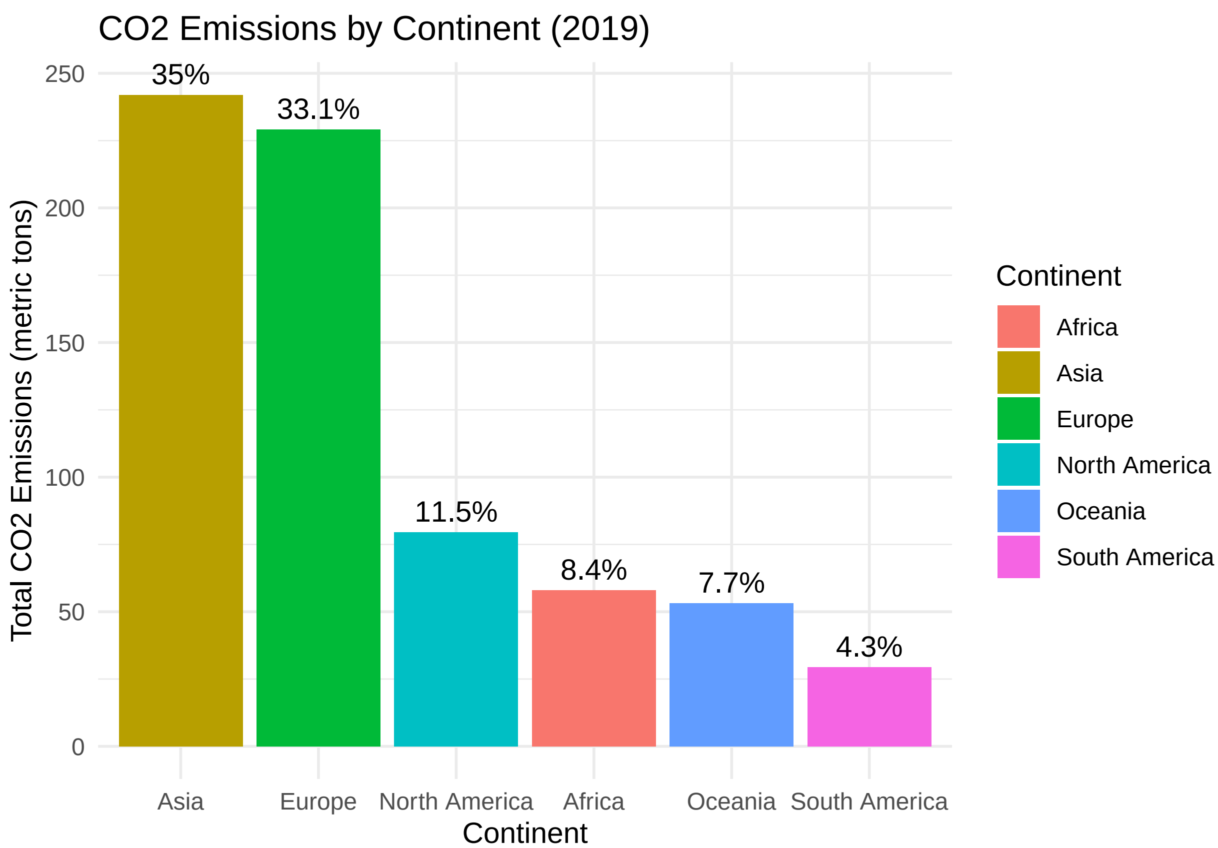
```
##  [1] "Country Name"   "country_code"   "Region"         "Indicator Name"
```

```
##  [5] "1990"           "1991"           "1992"           "1993"
##  [9] "1994"           "1995"           "1996"           "1997"
## [13] "1998"           "1999"           "2000"           "2001"
## [17] "2002"           "2003"           "2004"           "2005"
## [21] "2006"           "2007"           "2008"           "2009"
## [25] "2010"           "2011"           "2012"           "2013"
## [29] "2014"           "2015"           "2016"           "2017"
## [33] "2018"           "2019_A"         "2019_B"
```

```
continent_summary <- renamed_data %>%
group_by(Continent) %>%
summarise(
Total_CO2_2019 = sum(CO2_2019, na.rm = TRUE), Avg_CO2_per_capita = mean(CO2_2019, na.rm = TRUE)
) %>%
arrange(desc(Total_CO2_2019))
print(continent_summary)
```

```
## # A tibble: 6 x 3
##   Continent     Total_CO2_2019 Avg_CO2_per_capita
##   <chr>                  <dbl>              <dbl>
## 1 Asia                   242.                6.54
## 2 Europe                 229.                5.73
## 3 North America           79.6               4.19
## 4 Africa                  58.0               1.21
## 5 Oceania                 53.2               4.09
## 6 South America           29.4               2.67
```

```
library(dplyr)
library(ggplot2)
# Summarize total CO??? emissions by continent
continent_summary <- renamed_data %>%
group_by(Continent) %>%
summarise(Total_CO2_2019 = sum(CO2_2019, na.rm = TRUE)) %>%
mutate(Percent_CO2 = 100 * Total_CO2_2019 / sum(Total_CO2_2019))
ggplot(continent_summary, aes(x = reorder(Continent, -Total_CO2_2019), y = Total_CO2_2019, fill = Conti
geom_bar(stat = "identity") +
geom_text(aes(label = paste0(round(Percent_CO2, 1), "%")), vjust = -0.5, size = 4) +
labs(
title = "CO2 Emissions by Continent (2019)", x = "Continent", y = "Total CO2 Emissions (metric tons)",
) +
theme_minimal()
```


CO2 Emissions by Continent (2019)
Total CO2 Emissions (metric tons)
250
200
150
100
50
0
35%
33.1%
11.5%
8.4%
7.7%
4.3%
Asia
Europe
North America
Africa
Oceania
South America
Continent
Continent
Africa
Asia
Europe
North America
Oceania
South America