Project 2 Data Transformation - World Population Dataset

shri Tripathi

2024-10-11

World Population Dataset

Introduction

The World Population Dataset provides population data for countries and territories from 1970 to 2022. It includes key variables such as area, population density per square kilometer, growth rate, and the percentage share of the global population. Understanding how to work with population data is essential for making accurate predictions and building effective models.

Both *tidyr* and *dplyr* are part of the *tidyverse* and play crucial roles in data manipulation and preparation.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## 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 errors
```

```
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.3.3
```

```
## Linking to GEOS 3.11.2, GDAL 3.8.2, PROJ 9.3.1; sf_use_s2() is TRUE
```

```
library(downloader)
```

```
## Warning: package 'downloader' was built under R version 4.3.3
```

Load the untidy dataset

```
data <- read.csv(url("https://raw.githubusercontent.com/Shriyanshh/Project-2---Data-Transformation/refs/

# Get the number of rows and columns
dim(data)
```

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

```
# Display the structure
str(data)
```

```
## '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 4551
##  $ X2020.Population              : int  38972230 2866849 43451666 46189 77700 33428485 15585 92664 4503
##  $ X2015.Population              : int  33753499 2882481 39543154 51368 71746 28127721 14525 89941 4325
##  $ X2010.Population              : int  28189672 2913399 35856344 54849 71519 23364185 13172 85695 4110
##  $ X2000.Population              : int  19542982 3182021 30774621 58230 66097 16394062 11047 75055 3707
##  $ X1990.Population              : int  10694796 3295066 25518074 47818 53569 11828638 8316 63328 32637
##  $ X1980.Population              : int  12486631 2941651 18739378 32886 35611 8330047 6560 64888 280248
##  $ X1970.Population              : int  10752971 2324731 13795915 27075 19860 6029700 6283 64516 238428
##  $ Area..km..                    : int  652230 28748 2381741 199 468 1246700 91 442 2780400 29743 ...
##  $ Density..per.km..             : 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 ...
```

```
# Preview of the data frame
head(data)
```

```
##   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..km..
## 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.km.. 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
```

Tidying the dataset

The dataset was initially tidied by transforming it into a long format, which made it easier to handle and visualize. Additionally, the column names were standardized to enhance clarity and ensure consistency across the dataset.

```r
# The dataset contains regional population data from 1970 to 2022.
# To tidy the data, I will convert the year columns into a single column, transforming it into a long d
# This will allow for easier visualization of population trends over time, by year and country.

# First, I will rename the column headers to make them more descriptive and standardized.
data <- data %>%
  rename(
    "Country/Territory" = Country.Territory,        # Renaming the column for country or territory name
    "2022" = X2022.Population,                      # Renaming columns to show populations in respectiv
    "2020" = X2020.Population,
    "2015" = X2015.Population,
    "2010" = X2010.Population,
    "2000" = X2000.Population,
    "1990" = X1990.Population,
    "1980" = X1980.Population,
    "1970" = X1970.Population,
    "Area (km)" = Area..km..,                       # Renaming area column to indicate it's in square k
    "Density per km" = Density..per.km..,           # Clarifying density column to show it's population
    "Growth Rate" = Growth.Rate,                    # Keeping the growth rate column name as is
    "World Population Percentage" = World.Population.Percentage # Renaming to show percentage of world
  )

# Next, I will collapse the population columns from different years into a single "Year" column.
# This transforms the dataset into a long format, making it easier to analyze population changes over t
world_pop <- data %>%
  pivot_longer(`2022`:`1970`, names_to = "Year", values_to = "Population")

# Display the first few rows of the transformed dataset to confirm changes
head(world_pop)
```

```
## # A tibble: 6 x 11
##    Rank CCA3  `Country/Territory` Capital Continent `Area (km)` `Density per km`
##   <int> <chr> <chr>               <chr>   <chr>           <int>            <dbl>
## 1    36 AFG   Afghanistan         Kabul   Asia           652230             63.1
## 2    36 AFG   Afghanistan         Kabul   Asia           652230             63.1
## 3    36 AFG   Afghanistan         Kabul   Asia           652230             63.1
## 4    36 AFG   Afghanistan         Kabul   Asia           652230             63.1
## 5    36 AFG   Afghanistan         Kabul   Asia           652230             63.1
## 6    36 AFG   Afghanistan         Kabul   Asia           652230             63.1
## # i 4 more variables: `Growth Rate` <dbl>, `World Population Percentage` <dbl>,
## #   Year <chr>, Population <int>
```

Analysis

Statistical summaries were generated to identify the countries with the highest and lowest growth rates. The dataset was then visualized to display population trends over time and across different continents. Specifically, graphs were created to showcase the top 10 and bottom 10 countries based on population growth rates, as well as population sizes for the year 2022.

```
# Calculating statistical summaries for growth rates and populations
# Summarizes the dataset by calculating the average, minimum, and maximum growth rates
# Also identifies the smallest and largest populations in the dataset
world_pop %>%
  summarize(
    average_growth_rate = mean(`Growth Rate`),  # Calculating the average growth rate
    min_growth_rate = min(`Growth Rate`),       # Finding the minimum growth rate
    max_growth_rate = max(`Growth Rate`),       # Finding the maximum growth rate
    smallest_population = min(Population),      # Finding the smallest population
    largest_population = max(Population)        # Finding the largest population
  )
```

```
## # A tibble: 1 x 5
##   average_growth_rate min_growth_rate max_growth_rate smallest_population
##                 <dbl>           <dbl>           <dbl>               <int>
## 1                1.01           0.912            1.07                 510
## # i 1 more variable: largest_population <int>
```

```
# Extracting the names of countries with the highest growth rates
# Sorting the dataset in descending order based on growth rates, and then pulling the country/territory
countries_with_highest_growth_rate <- world_pop %>%
  arrange(desc(`Growth Rate`)) %>%
  pull(`Country/Territory`)

# Removing duplicate results as the dataset contains 8 separate entries per country for each year
# Selecting every 8th value to represent a unique country/territory
countries_with_highest_growth_rate <- countries_with_highest_growth_rate[seq(1, 80, 8)]

# Extracting the highest growth rates from the dataset
# Sorting the dataset in descending order of growth rates and pulling the corresponding growth rate val
highest_growths <- world_pop %>%
  arrange(desc(`Growth Rate`)) %>%
  pull(`Growth Rate`)

# Similarly, removing duplicates by selecting every 8th value from the sorted list
highest_growths <- highest_growths[seq(1, 80, 8)]
```

Here, we begin by loading the `world_map` data frame using `st_read()`. Next, two groups of countries are identified: those with the highest and lowest population growth rates, which are stored in the `countries_with_highest_growth_rate` and `countries_with_lowest_growth_rate` variables, respectively. For the lowest growth rates, we select every 8th country from a sorted list, up to 80 entries.

The `top_and_bottom` data frame augments the `world_map` data by adding a `fill` column, which is used to color-code countries based on their growth rate classification. Finally, we generate a plot using `ggplot2`. The plot visually distinguishes countries with the highest and lowest growth rates using different colors and labels them by name. The plot is saved as a PNG file in the current working directory for future reference or reporting.

```r
# Get the current working directory
current_wd <- getwd()

# Download the ZIP file containing shapefiles to the current working directory
download.file("https://github.com/autistic96/project-2/archive/refs/heads/main.zip",
              paste0(current_wd, "/map_shapefiles.zip"), mode = "wb")

# Unzip the downloaded ZIP file to a new folder called "map_shapefiles_folder"
unzip("map_shapefiles.zip", exdir = "map_shapefiles_folder")

# Unzip the internal ZIP file (within the first unzip) containing the actual shapefiles
unzip("map_shapefiles_folder/project-2-main/map_shapefiles.zip", exdir = "map_shapefiles_folder")

# Define the path to the shapefile (the ".shp" file)
shp_path <- "map_shapefiles_folder/map_shapefiles"

# Read the shapefile into an sf (simple feature) object using st_read from the sf package
world_map = st_read(shp_path)
```

```
## Reading layer ‘ne_10m_admin_0_countries’ from data source
##   ‘C:\Users\16462\Desktop\data607\project2\3\map_shapefiles_folder\map_shapefiles’
##   using driver ‘ESRI Shapefile’
## Simple feature collection with 258 features and 168 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -90 xmax: 180 ymax: 83.6341
## Geodetic CRS:  WGS 84
```

```r
# Assign the highest growth rates to their corresponding countries/territories
# Names are set to the country/territory names
names(highest_growths) = countries_with_highest_growth_rate
highest_growths
```

```
##  Moldova   Poland    Niger    Syria Slovakia DR Congo  Mayotte     Chad
##   1.0691   1.0404   1.0378   1.0376   1.0359   1.0325   1.0319   1.0316
##   Angola     Mali
##   1.0315   1.0314
```

```r
# Extract the countries with the lowest growth rates by sorting the data and pulling the relevant names
countries_with_lowest_growth_rate <- world_pop %>%
  arrange(`Growth Rate`) %>%
  pull(`Country/Territory`)

# Select every 8th country to avoid duplicates (since data spans multiple years)
countries_with_lowest_growth_rate <- countries_with_lowest_growth_rate[seq(1, 80, 8)]

# Pull the lowest growth rates and match them to the countries with the lowest rates
lowest_growths <- world_pop %>%
  arrange(`Growth Rate`) %>%
  pull(`Growth Rate`)

# Select every 8th value to remove duplicates (similar to above)
```

```r
lowest_growths <- lowest_growths[seq(1, 80, 8)]

# Assign names to the lowest growth rates (country/territory names)
names(lowest_growths) <- countries_with_lowest_growth_rate
lowest_growths
```

```
##                Ukraine                Lebanon         American Samoa
##                 0.9120                 0.9816                 0.9831
##               Bulgaria              Lithuania                 Latvia
##                 0.9849                 0.9869                 0.9876
## Bosnia and Herzegovina       Marshall Islands                 Serbia
##                 0.9886                 0.9886                 0.9897
##                Croatia
##                 0.9927
```

```r
# Verify the country lists for highest and lowest growth rates
countries_with_lowest_growth_rate
```

```
##  [1] "Ukraine"                "Lebanon"                "American Samoa"
##  [4] "Bulgaria"               "Lithuania"              "Latvia"
##  [7] "Bosnia and Herzegovina" "Marshall Islands"       "Serbia"
## [10] "Croatia"
```

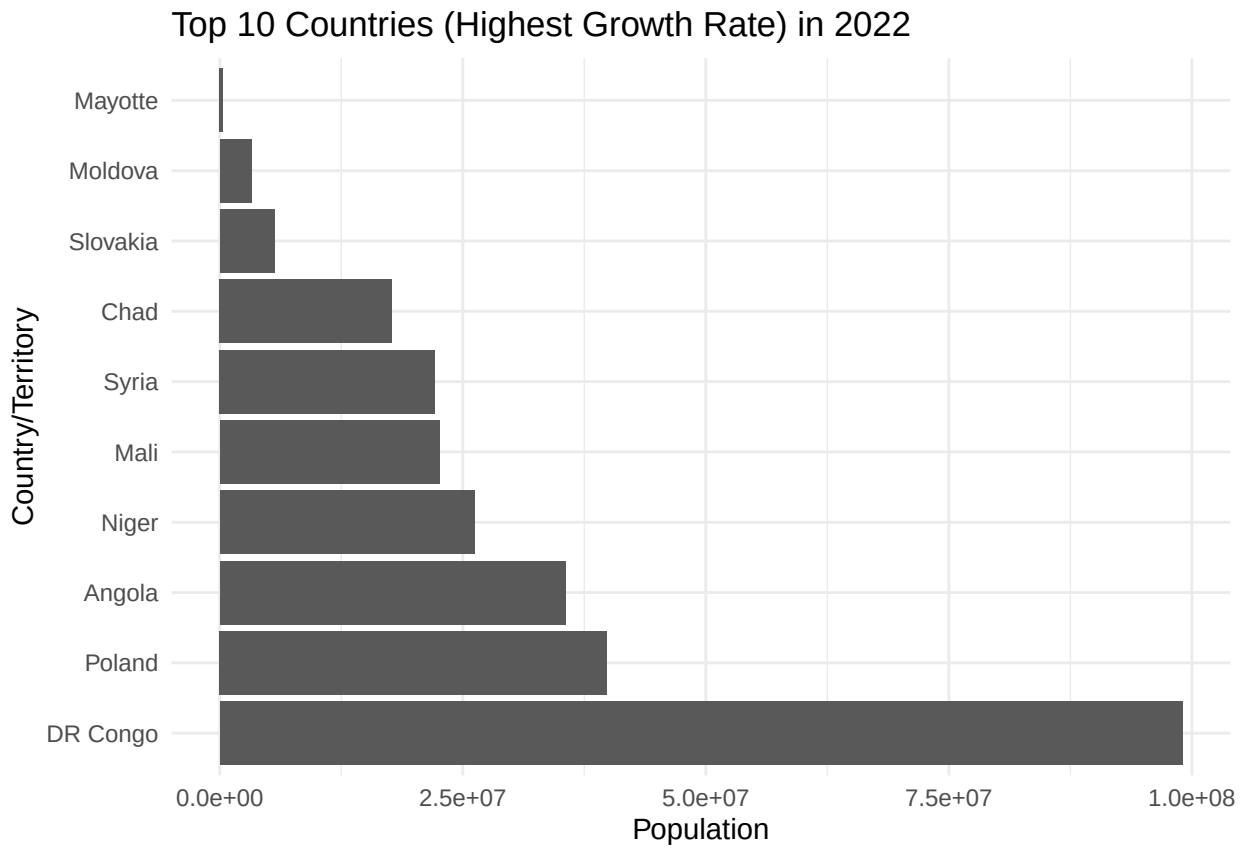
```r
countries_with_highest_growth_rate
```

```
##  [1] "Moldova"  "Poland"   "Niger"    "Syria"    "Slovakia" "DR Congo"
##  [7] "Mayotte"  "Chad"     "Angola"   "Mali"
```

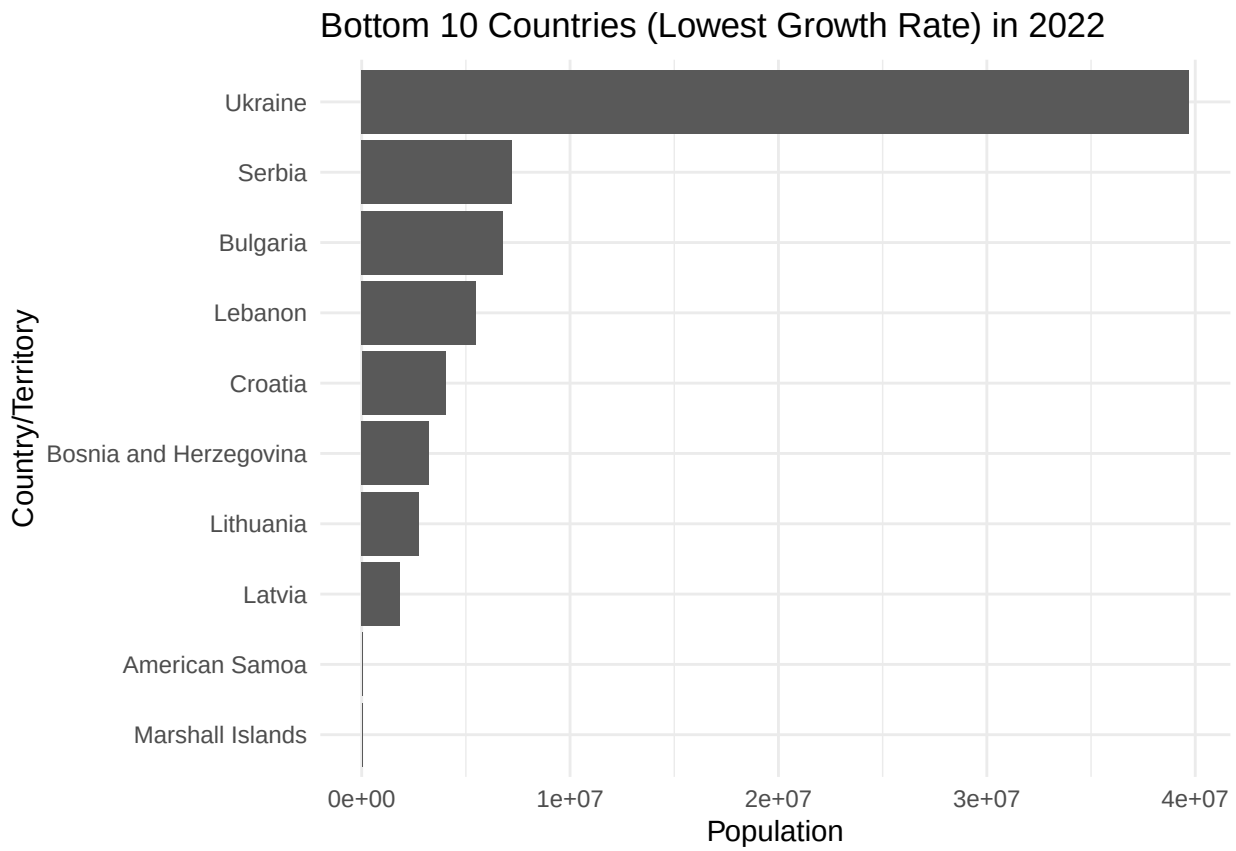
```r
# Add a 'fill' column to the world_map data to color-code countries
# Countries with the highest growth rates are colored blue, lowest in red, others in white
top_and_bottom <- world_map %>%
  mutate(fill = case_when(
    `NAME` %in% countries_with_highest_growth_rate ~ "blue",  # High growth rates colored blue
    `NAME` %in% countries_with_lowest_growth_rate ~ "red",    # Low growth rates colored red
    TRUE ~ "white"                                            # All other countries colored white
  ))

# Generate the plot using ggplot2
# geom_sf is used for drawing the map, and geom_sf_text adds country labels with check_overlap to avoid
p <- ggplot(data = top_and_bottom) +
  geom_sf(aes(fill = fill)) +                          # Fill the map based on the 'fill' column
  geom_sf_text(aes(label = NAME), check_overlap = TRUE) +  # Add country names as labels, avoiding over
  ggtitle("Map of World") +                            # Add a title to the plot
  scale_fill_identity()                                # Use the specified colors without any addition

# Save the plot as a PNG file in the current working directory
ggsave("top_and_bottom_10_with_labels.png", plot = p, width = 44, height = 40)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data
```

```r
# Plot of the top 10 countries/territories with the highest population growth rate
# Filter the dataset for 2022 and only include countries with the highest growth rates
# Create a bar plot of population for the top 10 countries with the highest growth rates in 2022
world_pop %>%
  filter(Year == "2022" & `Country/Territory` %in% countries_with_highest_growth_rate) %>%
  ggplot(aes(x = reorder(`Country/Territory`, -Population), y = Population)) +
  geom_bar(stat="identity") +                          # Use geom_bar to create a bar plot
  ggtitle("Top 10 Countries (Highest Growth Rate) in 2022") +  # Add plot title
  xlab("Country/Territory") +                          # Label for the x-axis
  ylab("Population") +                                 # Label for the y-axis
  theme_minimal() +                                    # Apply a minimal theme for better aesthetics
  coord_flip()                                         # Flip coordinates to make bars horizontal
```




```r
# Plot of bottom 10 countries/territories with the lowest population growth rate
# Similar process to the highest growth rate plot, but for the lowest growth rate countries
world_pop %>%
  filter(Year == "2022" & `Country/Territory` %in% countries_with_lowest_growth_rate) %>%
  ggplot(aes(x = reorder(`Country/Territory`, Population), y = Population)) +
  geom_bar(stat="identity") +                          # Use geom_bar for a bar plot
  ggtitle("Bottom 10 Countries (Lowest Growth Rate) in 2022") +  # Add plot title
  xlab("Country/Territory") +                          # Label for the x-axis
  ylab("Population") +                                 # Label for the y-axis
  theme_minimal() +                                    # Apply a minimal theme
  coord_flip()                                         # Flip coordinates for horizontal bars
```




```r
# Filter the dataset for the most recent year (2022) and arrange by population in descending order
recent_pop_data <- world_pop %>%
  filter(Year == 2022) %>%
  arrange(desc(Population))

# Display the top 10 countries/territories with the largest populations in 2022
head(recent_pop_data, n = 10)
```

```
## # A tibble: 10 x 11
##     Rank CCA3  `Country/Territory` Capital          Continent     `Area (km)`
##    <int> <chr> <chr>               <chr>            <chr>               <int>
##  1     1 CHN   China               Beijing          Asia              9706961
##  2     2 IND   India               New Delhi        Asia              3287590
##  3     3 USA   United States       Washington, D.C. North America     9372610
##  4     4 IDN   Indonesia           Jakarta          Asia              1904569
##  5     5 PAK   Pakistan            Islamabad        Asia               881912
##  6     6 NGA   Nigeria             Abuja            Africa             923768
##  7     7 BRA   Brazil              Brasilia         South America     8515767
##  8     8 BGD   Bangladesh          Dhaka            Asia               147570
##  9     9 RUS   Russia              Moscow           Europe           17098242
## 10    10 MEX   Mexico              Mexico City      North America     1964375
## # i 5 more variables: `Density per km` <dbl>, `Growth Rate` <dbl>,
## #   `World Population Percentage` <dbl>, Year <chr>, Population <int>
```

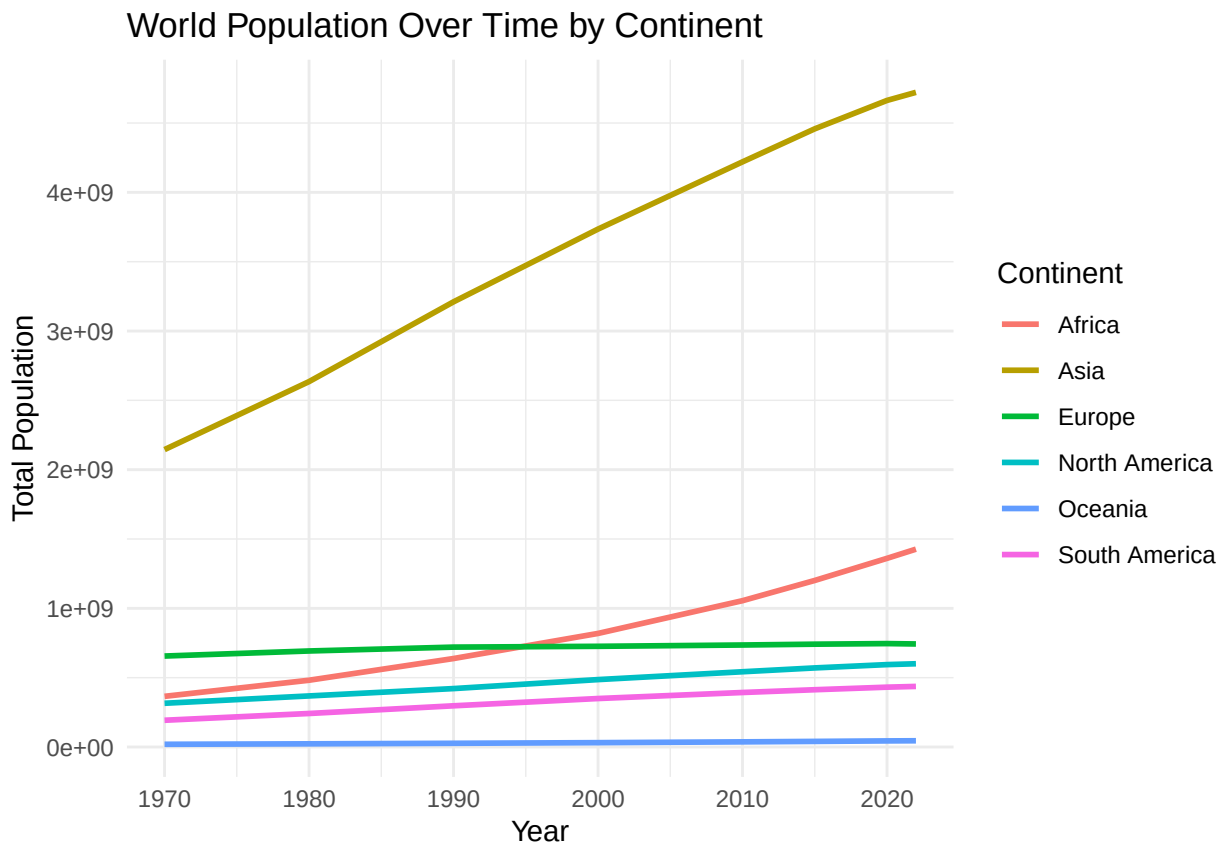
```r
# Display the bottom 10 countries/territories with the smallest populations in 2022
tail(recent_pop_data, n = 10)
```

```
## # A tibble: 10 x 11
##     Rank CCA3  `Country/Territory`       Capital      Continent     `Area (km)`
##    <int> <chr> <chr>                     <chr>        <chr>               <int>
##  1   225 NRU   Nauru                     Yaren        Oceania                21
##  2   226 WLF   Wallis and Futuna         Mata-Utu     Oceania               142
##  3   227 TUV   Tuvalu                    Funafuti     Oceania                26
##  4   228 BLM   Saint Barthelemy          Gustavia     North America          21
##  5   229 SPM   Saint Pierre and Miquelon Saint-Pierre North America         242
##  6   230 MSR   Montserrat                Brades       North America         102
##  7   231 FLK   Falkland Islands          Stanley      South America       12173
##  8   232 NIU   Niue                      Alofi        Oceania               260
##  9   233 TKL   Tokelau                   Nukunonu     Oceania                12
## 10   234 VAT   Vatican City              Vatican City Europe                  1
## # i 5 more variables: `Density per km` <dbl>, `Growth Rate` <dbl>,
## #   `World Population Percentage` <dbl>, Year <chr>, Population <int>
```

```r
# Plot of population growth over the years for all countries/territories
# Asia shows the biggest increase in population over time

# Convert the Year column from character to numeric for plotting
world_pop$Year <- as.numeric(world_pop$Year)

# Group the data by Year and Continent, then sum the population for each group
world_pop_summary <- world_pop %>%
  group_by(Year, Continent) %>%
  summarise(Total_Population = sum(Population))
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```r
# Create a line plot showing total population over time by continent
ggplot(data = world_pop_summary, aes(x = Year, y = Total_Population, color = Continent)) +
  geom_line(linewidth = 1) +                          # Use geom_line to create a line graph with specif
  ggtitle("World Population Over Time by Continent") +  # Add plot title
  xlab("Year") +                                      # Label for the x-axis
  ylab("Total Population") +                          # Label for the y-axis
  theme_minimal()                                     # Apply a minimal theme for clean presentation
```



Conclusion

After tidying and analyzing the World Population Dataset, several key insights became clear. We identified countries with notably high population growth rates, as well as others experiencing low or even negative growth. This information can be invaluable for policymakers in these regions as they plan for future demographic challenges. Additionally, visualizations of the 10 countries with the highest and lowest growth rates, along with their population sizes for 2022, provided a snapshot of global population dynamics, highlighting the disparities between nations.

We also examined population trends over time by continent. The line graph revealed that Asia has seen the most substantial population growth over the years. This trend could have significant socio-economic impacts, such as increased demand for resources and potential pressure on public services in densely populated areas.