Assignment 3

Mark KUZNETCOV and Betsy LAINES

2025-12-10

FIRST PART

1. Scrape Tax Data from Wikipedia Use R to scrape the table containing top marginal income tax rates by country from the following Wikipedia page

We load required libraries

```
library(rvest)
library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'

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

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

```
library(stringr)
library(tidyr)
```

URL of the Wikipedia page

```
url <- "https://en.wikipedia.org/wiki/List_of_countries_by_tax_rates"
```

Read the HTML page

```
page <- read_html(url)
```

Extract all tables from the page

```
tables <- page %>% html_table(fill = TRUE)
```

The table with top marginal income tax rates is typically the first one So, let's inspect and get the correct table

```
tax_table <- tables[[2]]
```

using dplyr with column positions:

```
tax_data <- tax_table %>%
  select(1, 2, 4, 9)
```

2. Clean the Table by Changing the Column Names Rename the columns in the scraped table to standard names (e.g., Country, Top Marginal Income Tax Rate, etc.).

Rename with spaces

```
colnames(tax_data) <- c("Country",
                        "Corporate Tax Rate",
                        "Top Marginal Income Tax Rate",
                        "VAT Rate")
```

```
tax_data[, 2] <- sapply(tax_data[, 2], function(x) gsub("%.*", "%", as.character(x)))
tax_data[, 3] <- sapply(tax_data[, 3], function(x) gsub("%.*", "%", as.character(x)))
tax_data[, 4] <- sapply(tax_data[, 4], function(x) gsub("%.*", "%", as.character(x)))
```

Remove the first row after header (row 1)

```
tax_data <- tax_data[-1, ]
```

3. Clean the Table by Removing Non-Numeric Elements Clean the content of the cells by removing footnotes and non-numeric characters. Ensure the tax rates are numeric values.

Clean the country names - remove everything in brackets and after

```
tax_data[[1]] <- gsub("\\[.*?\\]", "", tax_data[[1]])
```

First, convert the entire data frame properly

```
tax_data <- as.data.frame(tax_data, stringsAsFactors = FALSE)
```

Now remove % and round

```
for(i in 2:4) {
  tax_data[, i] <- gsub("%", "", tax_data[, i])
  tax_data[, i] <- suppressWarnings(as.numeric(tax_data[, i]))
     tax_data[, i] <- round(tax_data[, i], 0)
}
```

lower number is taken into account in the range: ex 2-5% => 2

Verify they're numeric

```r
sapply(tax_data, class)
```

```
##                       Country           Corporate Tax Rate
##                   "character"                   "numeric"
## Top Marginal Income Tax Rate                    VAT Rate
##                     "numeric"                   "numeric"
```

Second Part 4. Add a Continent Column

Install and load the countrycode package install.packages("countrycode") library(countrycode)

Add a Continent column

install.packages("countrycode")

```r
library(countrycode)
```

```
## Warning: package 'countrycode' was built under R version 4.5.2
```

```r
tax_data$Continent <- countrycode(sourcevar = tax_data$Country,
                                  origin = "country.name",
                                  destination = "continent")
```

```
## Warning: Some values were not matched unambiguously: Azores, Kosovo, Madeira, Micronesia, Saint Marti
```

Check for any countries that didn't match (will be NA)

```r
tax_data[is.na(tax_data$Continent), ]
```

```
##          Country Corporate Tax Rate Top Marginal Income Tax Rate VAT Rate
## 15        Azores                 14                           36        4
## 69    Micronesia                 21                           NA       NA
## 109       Kosovo                 NA                           10        0
## 124      Madeira                  5                           45        5
## 175 Saint Martin                 20                           NA       NA
## 181         Sark                  0                            0        0
##     Continent
## 15       <NA>
## 69       <NA>
## 109      <NA>
## 124      <NA>
## 175      <NA>
## 181      <NA>
```

Manually assign continents to the unmatched countries/territories

```r
tax_data$Continent[tax_data$Country == "Azores"] <- "Europe"
tax_data$Continent[tax_data$Country == "Micronesia"] <- "Oceania"
tax_data$Continent[tax_data$Country == "Kosovo"] <- "Europe"
tax_data$Continent[tax_data$Country == "Madeira"] <- "Europe"
tax_data$Continent[tax_data$Country == "Saint Martin"] <- "Americas"
tax_data$Continent[tax_data$Country == "Sark"] <- "Europe"
```

Verify all are now assigned

```
tax_data[is.na(tax_data$Continent), ]
```

```
## [1] Country                      Corporate Tax Rate
## [3] Top Marginal Income Tax Rate VAT Rate
## [5] Continent
## <0 rows> (o 0- extensión row.names)
```

so there are 0 rows, then we continue

5. Analyze Average Rates by Continent

```
# Compute average top marginal income tax rate by continent
avg_income <- aggregate(`Top Marginal Income Tax Rate` ~ Continent,
                        data = tax_data,
                        FUN = mean,
                        na.rm = TRUE)
avg_income <- avg_income[order(-avg_income$`Top Marginal Income Tax Rate`), ]

# Compute average VAT rate by continent
avg_vat <- aggregate(`VAT Rate` ~ Continent,
                     data = tax_data,
                     FUN = mean,
                     na.rm = TRUE)
avg_vat <- avg_vat[order(-avg_vat$`VAT Rate`), ]

avg_income
```

```
##   Continent Top Marginal Income Tax Rate
## 4    Europe                     34.46154
## 1    Africa                     32.19355
## 2  Americas                     27.70732
## 3      Asia                     22.65957
## 5   Oceania                     19.20000
```

```
avg_vat
```

```
##   Continent  VAT Rate
## 3      Asia 18.214286
## 4    Europe 16.730769
## 1    Africa 15.096774
## 2  Americas 11.000000
## 5   Oceania  6.777778
```

So, we notice that:

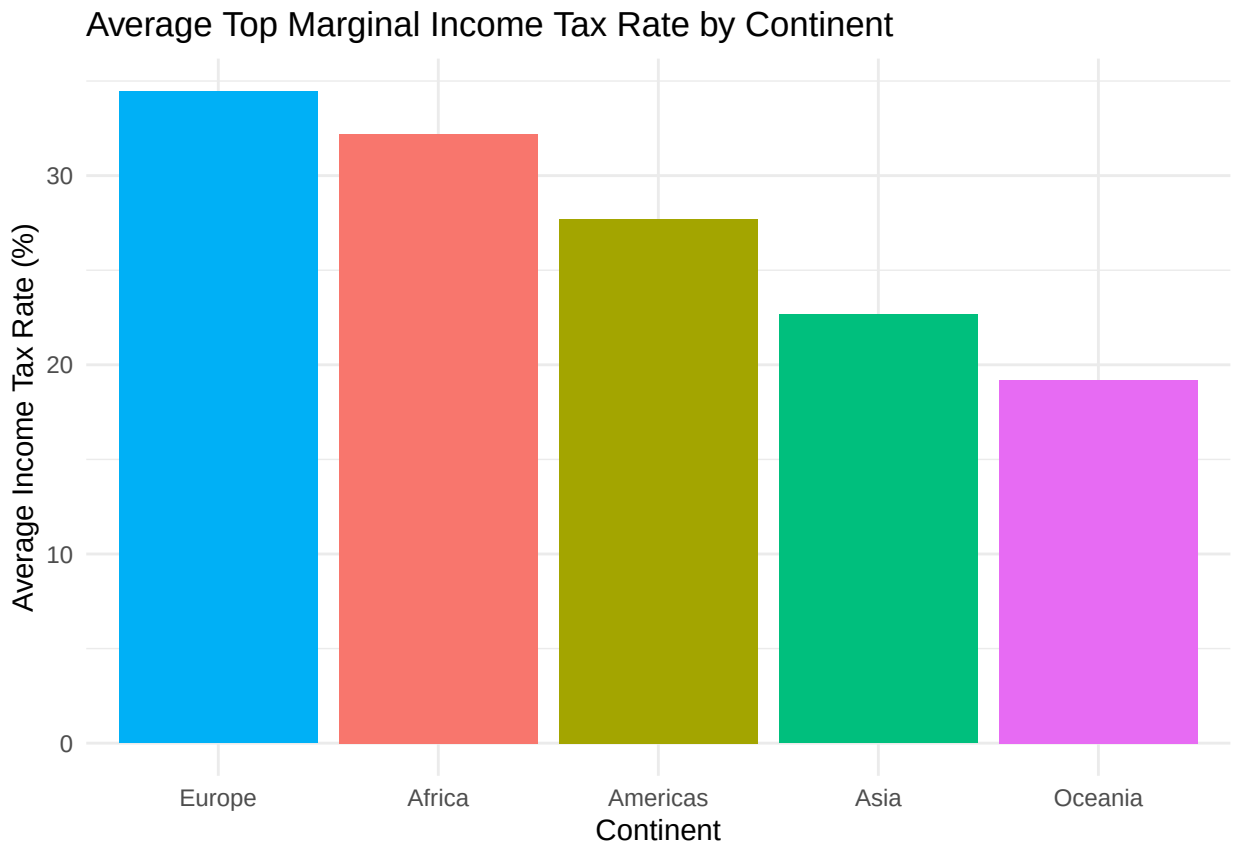
Europe has the highest average top marginal income tax rate, followed by Asia.
For VAT, Asia shows the highest average rate across all continents.

6. Graph: Average Top Income Tax Rate by Continent

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.5.2
```

```r
ggplot(avg_income,
       aes(x = reorder(Continent, -`Top Marginal Income Tax Rate`),
           y = `Top Marginal Income Tax Rate`,
           fill = Continent)) +
  geom_col() +
  labs(title = "Average Top Marginal Income Tax Rate by Continent",
       x = "Continent",
       y = "Average Income Tax Rate (%)") +
  theme_minimal() +
  theme(legend.position = "none")
```



Europe has the highest average rate, followed by Africa and the Americas. Asia and Oceania display significantly lower average income tax levels, indicating that these regions generally apply lighter top end taxation compared to Europe and Africa.

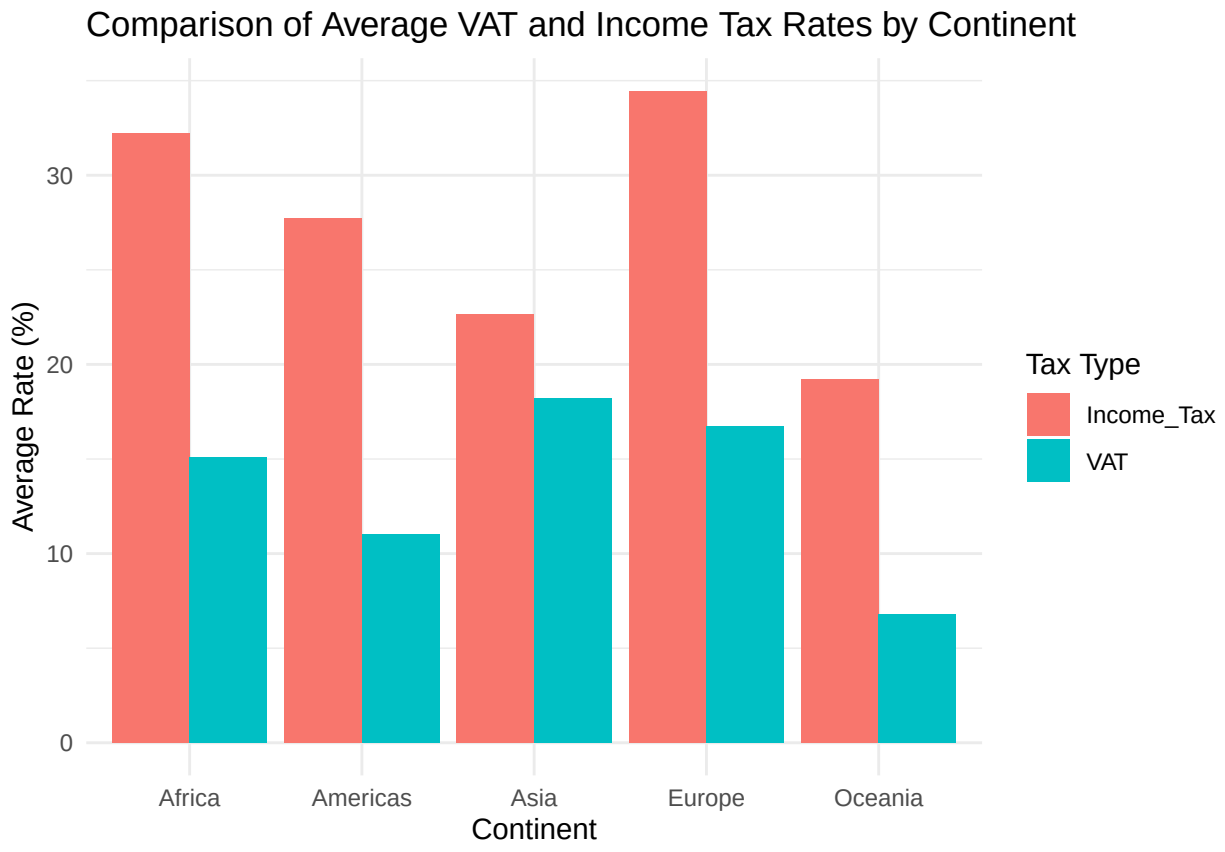
7. Graph: Compare VAT and Income Tax Rates Create a single grouped bar chart comparing the average VAT rates and top marginal income tax rates for each continent.

```r
compare_df <- merge(avg_income, avg_vat, by = "Continent")
```

```r
colnames(compare_df) <- c("Continent", "Income_Tax", "VAT")


library(tidyr)
compare_long <- pivot_longer(compare_df,
                             cols = c("Income_Tax", "VAT"),
                             names_to = "Tax_Type",
                             values_to = "Rate")


ggplot(compare_long,
       aes(x = Continent, y = Rate, fill = Tax_Type)) +
  geom_col(position = "dodge") +
  labs(title = "Comparison of Average VAT and Income Tax Rates by Continent",
       x = "Continent",
       y = "Average Rate (%)",
       fill = "Tax Type") +
  theme_minimal()
```



Explanation:

The grouped bar chart compares average VAT rates and top marginal income tax rates across continents. Europe shows the highest income tax levels, while Africa ranks second. VAT rates are systematically lower than income tax rates in all regions.

8. Map: Top Marginal Income Tax Rate by Country

Build a map visualizing the top marginal income tax rate for each country. (2 pts) Include: • A clear title and legend. • Gray shading for countries with missing data.

```
library(rworldmap)
```

```
## Warning: package 'rworldmap' was built under R version 4.5.2
```

```
## Cargando paquete requerido: sp
```

```
## Warning: package 'sp' was built under R version 4.5.2
```

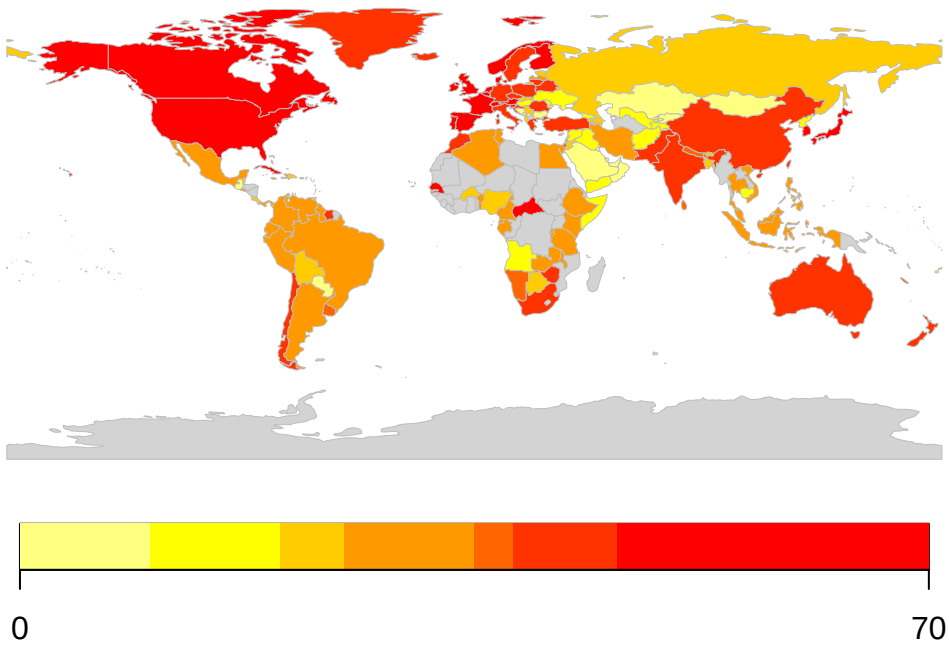
```
## ### Welcome to rworldmap ###
```

```
## For a short introduction type :   vignette('rworldmap')
```

```
world <- joinCountryData2Map(
  tax_data,
  joinCode = "NAME",
  nameJoinColumn = "Country",
  verbose = TRUE
)
```

```
## 216 codes from your data successfully matched countries in the map
## 16 codes from your data failed to match with a country code in the map
##       failedCodes failedCountries
##  [1,] NA          "Azores"
##  [2,] NA          "DR Congo"
##  [3,] NA          "Curaçao"
##  [4,] NA          "Eswatini (Swaziland)"
##  [5,] NA          "Falkland Islands"
##  [6,] NA          "Micronesia"
##  [7,] "GIB"       "Gibraltar"
##  [8,] NA          "Madeira"
##  [9,] NA          "North Macedonia"
## [10,] NA          "Saint Barthélemy"
## [11,] NA          "São Tomé and Príncipe"
## [12,] ""          "Sark"
## [13,] NA          "Sint Maarten"
## [14,] ""          "Svalbard"
## [15,] "TKL"       "Tokelau"
## [16,] NA          "U.S. Virgin Islands"
## 27 codes from the map weren't represented in your data
```

```
mapCountryData(world,
               nameColumnToPlot = "Top Marginal Income Tax Rate",
               mapTitle = "Top Marginal Income Tax Rate by Country",
               missingCountryCol = "lightgray")
```



Explanation:

This choropleth map displays the top marginal income tax rate for each country. Countries with higher tax rates appear in darker red, while lighter yellow tones correspond to lower rates. Countries shown in light gray are those for which no income tax data are available in the dataset.

9. Handle Missing Data: When the top marginal income tax rate is missing, replace it with the corporate income tax rate. Update your dataset accordingly.

```
tax_data$IncomeTax_Updated <- tax_data$`Top Marginal Income Tax Rate`

missing_idx <- is.na(tax_data$IncomeTax_Updated) &
               !is.na(tax_data$`Corporate Tax Rate`)

tax_data$IncomeTax_Updated[missing_idx] <-
  tax_data$`Corporate Tax Rate`[missing_idx]
```

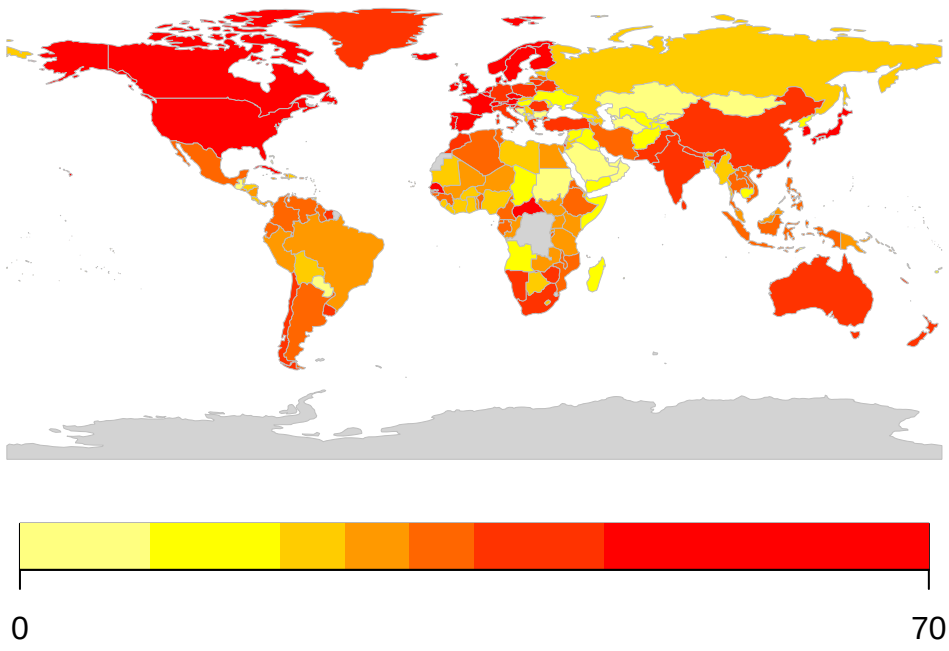
10. Graph: New Income Tax Rate Variable Create a graph to visualize the new variable that includes the substituted values for missing income tax rates. if (!requireNamespace("rworldmap", quietly = TRUE)) { install.packages("rworldmap") }

```
library(rworldmap)
library(ggplot2)
```

```r
world_updated <- joinCountryData2Map(
  tax_data,
  joinCode = "NAME",
  nameJoinColumn = "Country",
  verbose = TRUE
)
```

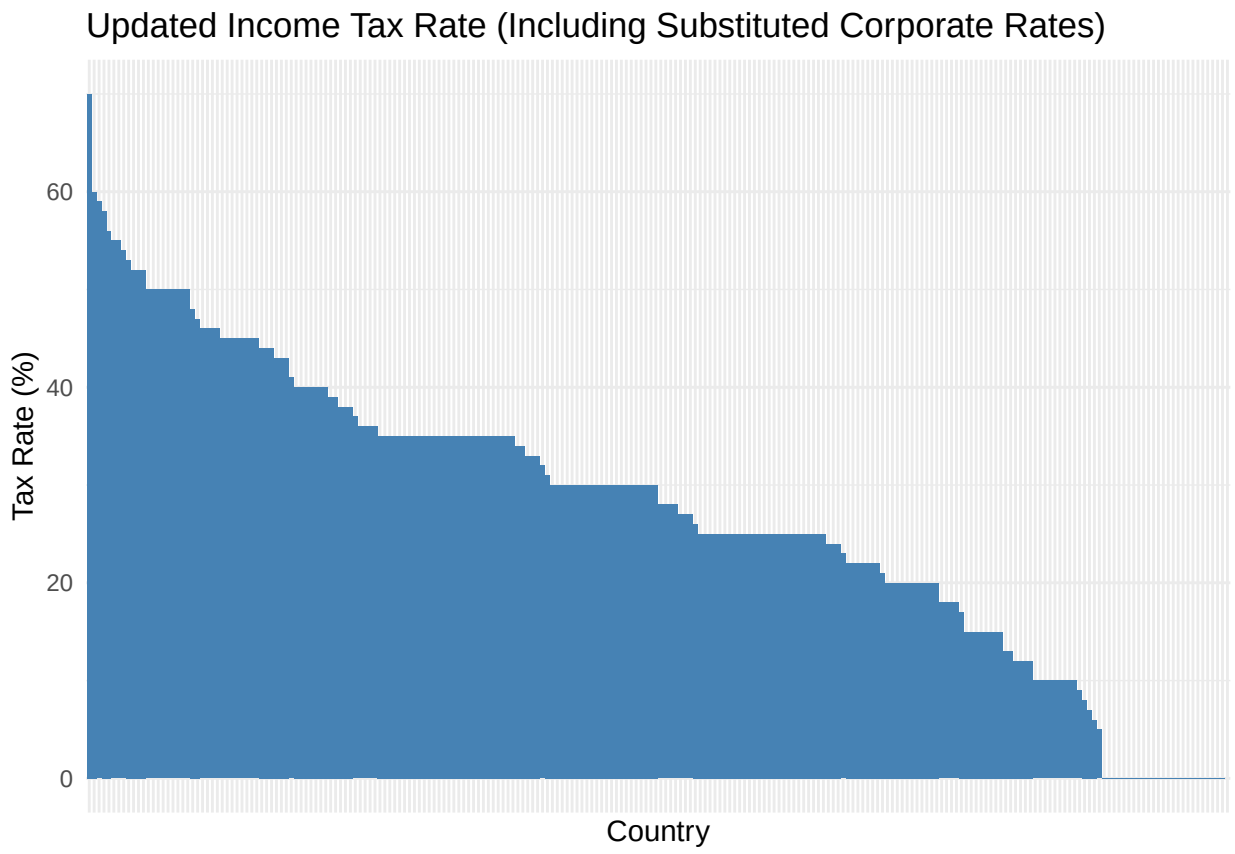
```
## 216 codes from your data successfully matched countries in the map
## 16 codes from your data failed to match with a country code in the map
##       failedCodes failedCountries
##  [1,] NA          "Azores"
##  [2,] NA          "DR Congo"
##  [3,] NA          "Curaçao"
##  [4,] NA          "Eswatini (Swaziland)"
##  [5,] NA          "Falkland Islands"
##  [6,] NA          "Micronesia"
##  [7,] "GIB"       "Gibraltar"
##  [8,] NA          "Madeira"
##  [9,] NA          "North Macedonia"
## [10,] NA          "Saint Barthélemy"
## [11,] NA          "São Tomé and Príncipe"
## [12,] ""          "Sark"
## [13,] NA          "Sint Maarten"
## [14,] ""          "Svalbard"
## [15,] "TKL"       "Tokelau"
## [16,] NA          "U.S. Virgin Islands"
## 27 codes from the map weren't represented in your data
```

```r
mapCountryData(
  world_updated,
  nameColumnToPlot = "IncomeTax_Updated",
  mapTitle = "Updated Income Tax Rate (Including Substitutions)",
  missingCountryCol = "lightgray"
)
```




```
ggplot(tax_data,
       aes(x = reorder(Country, -IncomeTax_Updated),
           y = IncomeTax_Updated)) +
  geom_col(fill = "steelblue") +
  labs(title = "Updated Income Tax Rate (Including Substituted Corporate Rates)",
       x = "Country",
       y = "Tax Rate (%)") +
  theme_minimal() +
  theme(axis.text.x = element_blank())
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_col()`).
```



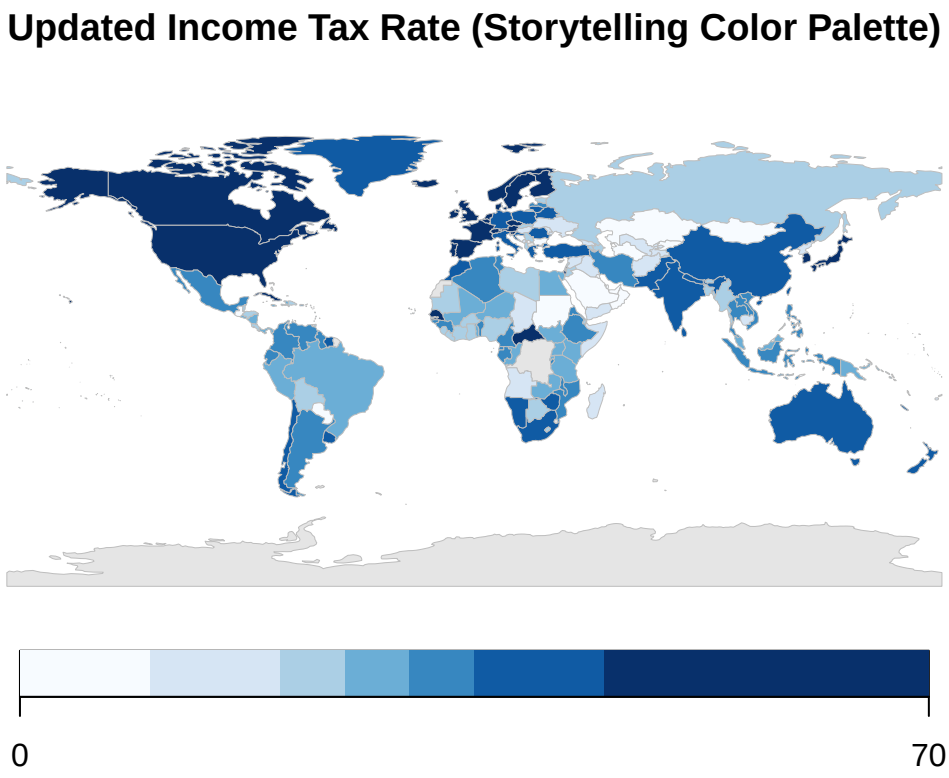
This ensures full country coverage and reveals wide differences in statutory tax levels across countries.

11. Enhanced Map: Custom Colors Use a scale function to change the colors on the map created in Question 10. Choose a color palette that enhances the storytelling of your visualization.

```
library(RColorBrewer)


mapCountryData(
  world_updated,
  nameColumnToPlot = "IncomeTax_Updated",
  mapTitle = "Updated Income Tax Rate (Storytelling Color Palette)",
  missingCountryCol = "gray90",
  colourPalette = brewer.pal(9, "Blues")
)
```

```
## Warning in rwmGetColours(colourPalette, numColours): 9 colours specified and 7
## required, using interpolation to calculate colours
```



THIRD PART

12. Free visualization From the Insee website (here) or another data producer (Banque de France, IMF, ILO etc.) pick a theme you are interested in (demography, wealth, labor market, poverty, firms, education etc.). Then, choose one / a combination of datasets and make a visualization. The visualization should tell us a story, and the visualization choices should be consistent with this story.

Comment the visualization

```
# Load libraries
library(dplyr)
library(lubridate)
```

```
##
## Adjuntando el paquete: 'lubridate'
```

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

```
library(ggplot2)
```

Import datasets

Monthly CPI / Inflation

```r
inflation <- read.csv("D:/AMSE Master in Economics/Data Visualization/december/inflation.csv",
                      stringsAsFactors = FALSE, sep = ";", skip = 5) %>%
  rename(Date = 1) %>%
  rename(Percentage = 2)
```

Quarterly Unemployment

```r
unemployment <- read.csv("D:/AMSE Master in Economics/Data Visualization/december/unemployment.csv",
                         stringsAsFactors = FALSE, sep = ";", skip = 5) %>%
  rename(Date = 1) %>%
  rename(Percentage = 2)
```

```r
View(inflation)
View(unemployment)
```

Convert character Percentage to numeric

```r
inflation$Percentage <- as.numeric(gsub(",", ".", inflation$Percentage))
```

```
## Warning: NAs introducidos por coerción
```

```r
unemployment$Percentage <- as.numeric(gsub(",", ".", unemployment$Percentage))
```

Filter unemployment from 1990 onwards

```r
unemployment <- unemployment %>%
  filter(Date >= as.Date("1990-01-01"))

str(unemployment)
```

```
## 'data.frame':    143 obs. of  2 variables:
##  $ Date      : chr  "2025-09-30" "2025-06-30" "2025-03-31" "2024-12-31" ...
##  $ Percentage: num  7.5 7.4 7.3 7.1 7.2 7.1 7.3 7.3 7.2 7 ...
```

```r
library(lubridate)
inflation$Date <- ymd(inflation$Date)
unemployment$Date <- ymd(unemployment$Date)
```

Plot

Add a Type column to each dataset

```r
inflation <- inflation %>% mutate(Type = "Inflation")
unemployment <- unemployment %>% mutate(Type = "Unemployment")
```

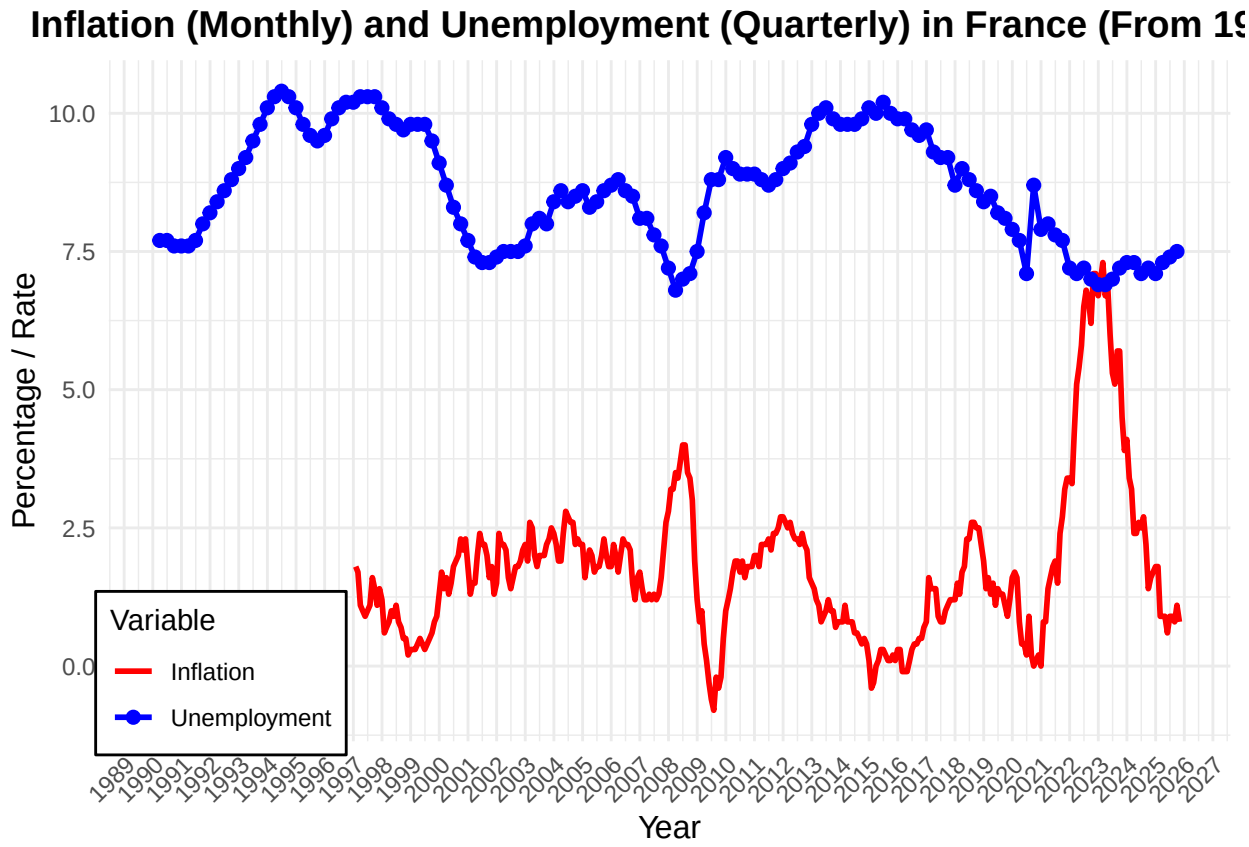
Combine datasets

```r
combined <- bind_rows(
  inflation %>% select(Date, Percentage, Type),
  unemployment %>% select(Date, Percentage, Type)
)
```

Plot with legend

```
library(scales)

ggplot(combined, aes(x = Date, y = Percentage, color = Type)) +
  geom_line(data = combined %>% filter(Type == "Inflation"), size = 1) +
  geom_line(data = combined %>% filter(Type == "Unemployment"), size = 1) +
  geom_point(data = combined %>% filter(Type == "Unemployment"), size = 2) +
  scale_color_manual(values = c("Inflation" = "red", "Unemployment" = "blue")) +
  scale_x_date(
    date_breaks = "1 year",      # Show ticks every year
    date_labels = "%Y"           # Format as 4-digit year
  ) +
  labs(
    title = "Inflation (Monthly) and Unemployment (Quarterly) in France (From 1990)",
    x = "Year",
    y = "Percentage / Rate",
    color = "Variable"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    axis.title = element_text(size = 12),
    legend.position = c(0.1, 0.1),
    legend.background = element_rect(fill = "white", color = "black"),
    axis.text.x = element_text(angle = 45, hjust = 1)  # rotate labels if crowded
  )
```



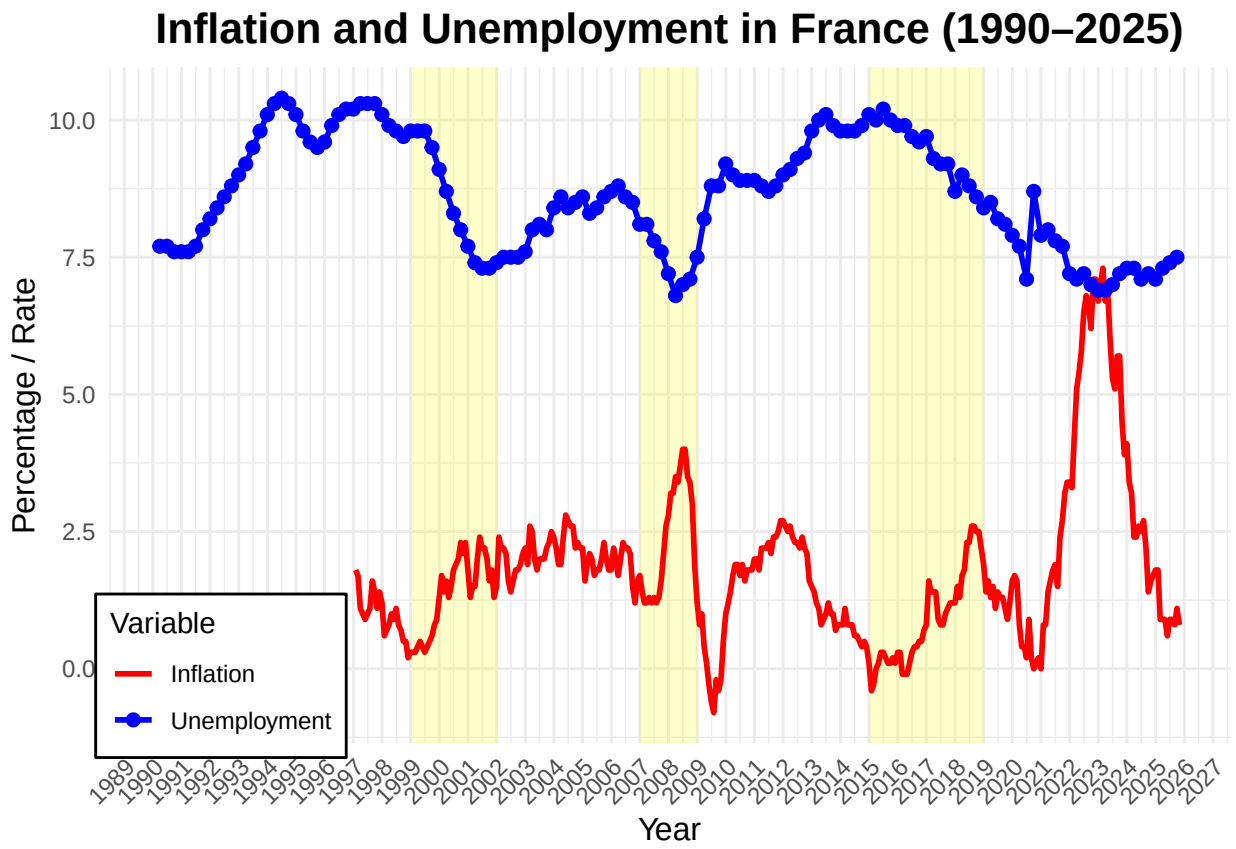
Define periods to highlight

```
highlight_periods <- data.frame(
  start = ymd(c("1999-01-01", "2007-01-01", "2015-01-01")),
  end   = ymd(c("2001-12-31", "2008-12-31", "2018-12-31"))
)

ggplot(combined, aes(x = Date, y = Percentage, color = Type)) +
  # Highlight periods
  geom_rect(data = highlight_periods,
            aes(xmin = start, xmax = end, ymin = -Inf, ymax = Inf),
            inherit.aes = FALSE,
            fill = "yellow", alpha = 0.2) +
  # Inflation line
  geom_line(data = combined %>% filter(Type == "Inflation"), size = 1) +
  # Unemployment line + points
  geom_line(data = combined %>% filter(Type == "Unemployment"), size = 1) +
  geom_point(data = combined %>% filter(Type == "Unemployment"), size = 2) +
  # Colors
  scale_color_manual(values = c("Inflation" = "red", "Unemployment" = "blue")) +
  # X-axis with yearly breaks
  scale_x_date(
    date_breaks = "1 year",
    date_labels = "%Y"
  ) +
  labs(
```

```
    title = "Inflation and Unemployment in France (1990-2025)",
    x = "Year",
    y = "Percentage / Rate",
    color = "Variable"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
    axis.title = element_text(size = 12),
    legend.position = c(0.1, 0.1),
    legend.background = element_rect(fill = "white", color = "black"),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

```
## Warning: Removed 72 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



The highlighted periods on the graph illustrate moments where inflation increases while unemployment falls, which is consistent with the logic of the Phillips Curve. For example, during 1999–2001, 2007–2008, and 2015–2018, France experienced rising price levels alongside a declining unemployment rate. These episodes visually show the short-run inverse relationship between the two variables: when economic activity strengthens and inflation picks up, the labor market typically tightens, reducing unemployment. While this relationship is not perfectly stable over time, these highlighted periods show that the Phillips Curve mechanism does appear at several points in the French economy during the past decades.