Task A

Zixia Zeng

2024-11-04

```
# data wrangling
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   3.5.1     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
```

```
# data visualization
library(ggplot2)
```

(Q1)

```
# Load debt_data.csv
debt_df = read.csv("debt_data.csv")

# Load country_data.csv
country_df = read.csv("country_data.csv")

# Load indicator_data.csv
indicator_df = read.csv("indicator_data.csv")

# Use dim function to check the number of columns and rows of debt_df
dim(debt_df)
```

```
## [1] 13824    63
```

By using the dim() function, the number of rows is 13824, and the number of columns is 63. This means the debt_df data frame contains 63 variables and 13824 observations.

(Q2)

```
# Update debt_df by "DT.NFL.BLAT.CD" in descending order
debt_df = arrange(debt_df,desc(DT.NFL.BLAT.CD))

# Select the first 4 rows and specific columns
subset_df = debt_df[1:4, c("Country.Code", "Year", "NY.GNP.MKTP.CD", "DT.NFL.BLAT.CD")]

# Display the subset
print(subset_df)
```

```
##   Country.Code      Year NY.GNP.MKTP.CD DT.NFL.BLAT.CD
## 1          MEX year_1995    3.66827e+11     9398190731
## 2          EGY year_2013    2.81028e+11     7233642176
## 3          BRA year_2017    2.02494e+12     6506490468
## 4          PAK year_2018    3.50691e+11     6201281870
```

(Q3)

```
# Select the first two variables of debt_df, (Country.Code and Year)
debt_df_first2 = debt_df[, 1:2]
# Select rest variables and prepare to replace them
debt_df_rest = debt_df[,-c(1,2)]

# Match the Indicator_code in Indicator_df and replace them by the indicator_name
colnames(debt_df_rest) = indicator_df$INDICATOR_NAME[
  match(colnames(debt_df_rest),indicator_df$INDICATOR_CODE)
  ]

# Combine the first two variables
debt_df2 = cbind(debt_df_first2,debt_df_rest)

debt_df2 %>%
  select(c("Country.Code","Year","Net financial flows, others (NFL, current US$)")) %>%
  head(5)
```

```
##   Country.Code      Year Net financial flows, others (NFL, current US$)
## 1          MEX year_1995                                            NA
## 2          EGY year_2013                                      -14314777
## 3          BRA year_2017                                     -195705180
## 4          PAK year_2018                                      321846510
## 5          EGY year_2016                                     2141976215
```

(Q4)

```
# Use left_join to merge data in country_df to debt_df2
debt_df3 = left_join(debt_df2,country_df,by = "Country.Code")
```

```r
# Delete the "SpecialNotes"
debt_df3 = select(debt_df3,-SpecialNotes)

# Subset consisting of the first three rows and four columns
debt_df3 %>%
  select(c(Country.Name,IncomeGroup,Year,`Total reserves in months of imports`)) %>%
  head(5)
```

```
##       Country.Name         IncomeGroup      Year
## 1           Mexico Upper middle income year_1995
## 2 Egypt, Arab Rep. Lower middle income year_2013
## 3           Brazil Upper middle income year_2017
## 4         Pakistan Lower middle income year_2018
## 5 Egypt, Arab Rep. Lower middle income year_2016
##   Total reserves in months of imports
## 1                            2.825546
## 2                            2.730040
## 3                           14.861069
## 4                            1.905231
## 5                            3.885411
```

(Q5)

```r
#rename the five columns
debt_df3 = debt_df3 %>%
  rename(Total_reserves = `Total reserves in months of imports`) %>%
  rename(External_debt = `External debt stocks, total (DOD, current US$)`) %>%
  rename(Financial_flow = `Net financial flows, bilateral (NFL, current US$)`) %>%
  rename(Imports = `Imports of goods, services and primary income (BoP, current US$)`) %>%
  rename(IFC = `IFC, private nonguaranteed (NFL, US$)`)
# Display data frame after rename
debt_df3 %>%
  select(c(Total_reserves,External_debt,Financial_flow,Imports,IFC)) %>%
  head(5)
```

```
##   Total_reserves External_debt Financial_flow      Imports       IFC
## 1       2.825546  166734000000     9398190731  72391910000         0
## 2       2.730040   46534987115     7233642176  72685700000 -42864095
## 3      14.861069  543000000000     6506490468 301961000000 397855350
## 4       1.905231  100199000000     6201281870  74555877000  11389136
## 5       3.885411   69188517055     5714011601  73019900000  77244772
```

(Q6)

```r
# Summarize debt_df3 and create four new columns
debt_summary = debt_df3 %>%
  group_by(Region) %>%
  summarise(
    TR_mn = mean(Total_reserves, na.rm = TRUE),
```

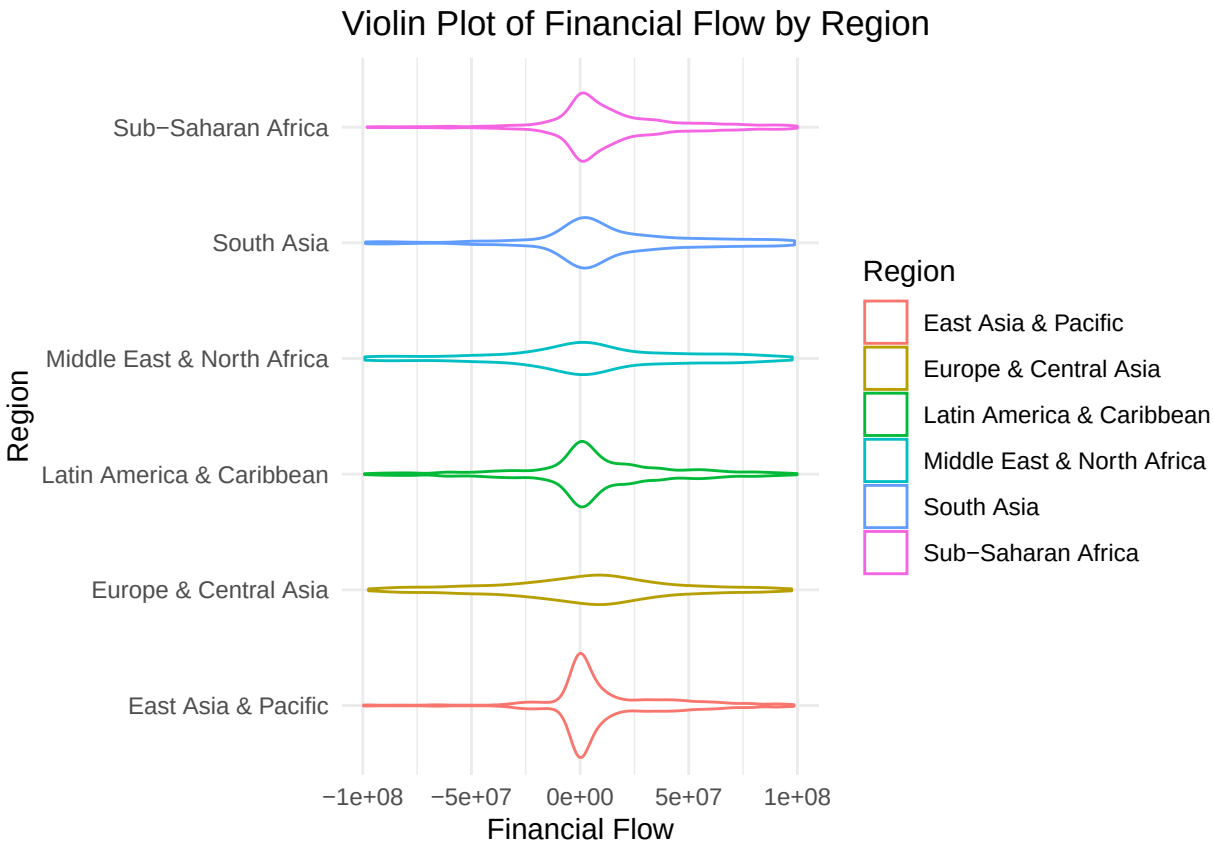
```
    ED_md = median(External_debt, na.rm = TRUE),
    FF_quantile = quantile(Financial_flow, 0.2, na.rm = TRUE),
    IFC_sd = sd(IFC, na.rm = TRUE)
    )
# Display summary of debt_df3
print(debt_summary)
```

```
## # A tibble: 7 x 5
##   Region                       TR_mn      ED_md FF_quantile    IFC_sd
##   <chr>                        <dbl>      <dbl>       <dbl>     <dbl>
## 1 East Asia & Pacific          5.19 2248479410   -2357020. 52498519.
## 2 Europe & Central Asia        3.58 8237728122  -53631246. 50820255.
## 3 Latin America & Caribbean    3.84 4159662669  -25144268. 62054545.
## 4 Middle East & North Africa   7.72 7481954468  -92269932. 21414719.
## 5 North America                1.99         NA         NA        NA
## 6 South Asia                   4.94 4940329805    -373253. 76630044.
## 7 Sub-Saharan Africa           3.32 1709094992   -1673594. 24748455.
```

(Q7)

```
# Filter data to remove missing values and out-of-range values in "Financial_flow"
debt_df3_filtered = debt_df3 %>%
  filter(!is.na(Financial_flow) & Financial_flow >= -1e8 & Financial_flow <= 1e8)

# Create the violin plot
ggplot(debt_df3_filtered, aes(x = Financial_flow, y = Region , color = Region)) +
  geom_violin(trim = TRUE) +
  labs(title = "Violin Plot of Financial Flow by Region",
       x = "Financial Flow",
       y = "Region") +
  theme_minimal()
```



(Q8)

```
# First check the data format of each variables
head(debt_df3$Year)
```

```
## [1] "year_1995" "year_2013" "year_2017" "year_2018" "year_2016" "year_1995"
```

Format of "Year" data is like "year_1996", it is not number, so "Year" has to be transformed to numeric data then it can be used to filter the data between 1960 to 2023

```
#Format of "Year" data is like "year_1996", it is not number.
debt_df3_filtered = debt_df3 %>%
  # To select data from 1960 to 2023, "Year" needs to delete the prefix and leave the number
  separate(Year, into = c("prefix", "Year"), sep = "_") %>%
  mutate(Year = as.numeric(Year)) %>%
  # Use filter function to select the data
  filter(`Country.Name` %in% c("Italy", "France", "United Kingdom", "Sudan", "Afghanistan", "Brazil") &
           Year >= 1960 & Year <= 2023 &
           # remove all the NA value in Total_reserves
           !is.na(Total_reserves))
```

Data in debt_df3 is already cleaned, then it can be used to create plot.

```r
# create plot
ggplot(debt_df3_filtered, aes(x = Year, y = Total_reserves, color = Country.Name)) +
  # Remove the NA value
  geom_line(na.rm = TRUE) +
  # Draw points at each data pair, and ignore NA value
  geom_point(na.rm = TRUE) +
  # divide the plot into different Income groups
  facet_wrap(~ IncomeGroup) +
  # Create Labels for the plot
  labs(title = "Total Reserves from 1960 to 2023",
       x = "Year",
       y = "Total Reserves",
       color = "Country")
```