TaskA

zerofrom

2024-11-17

Q1. Data Wrangling

```
# Set working directory and read data from CSV files into data frames
setwd(getwd())
debt_df <- read.csv("debt_data.csv", header = TRUE, sep = ",")
country_df <- read.csv("country_data.csv", header = TRUE, sep = ",")
indicator_df <- read.csv("indicator_data.csv", header = TRUE, sep = ",")
#head(debt_df)
#summary(debt_df)
numCol = ncol(debt_df)
numRow = nrow(debt_df)
print(paste("debt_df: The number of columns: ",numCol))
```

```
## [1] "debt_df: The number of columns:  63"
```

```
print(paste("debt_df: The number of rows: ",numRow))
```

```
## [1] "debt_df: The number of rows:  13824"
```

Q2.Update data frame by reordering

```
# reordering by the column 'DT.NFL.BLAT.CD'
debt_df <- debt_df %>%
  arrange(desc(DT.NFL.BLAT.CD))
# select the specifed columns and display the first 4 rows
debt_df_sub <- debt_df %>%
  select("Country.Code", "Year", "NY.GNP.MKTP.CD", "DT.NFL.BLAT.CD") %>%
  head(4)
debt_df_sub
```

```
##   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. Create debt_df2 and replace indicator column names in debt_df by referring to indicator_df

```r
# Reorder debt_df
column_mapping <-setNames(indicator_df$INDICATOR_NAME, indicator_df$INDICATOR_CODE)
column_mapping[c("Country.Code", "Year")] <- c("Country.Code", "Year")
debt_df2 <- debt_df %>%
  rename_with(~column_mapping[.x],.cols = names(debt_df))

# Select specified columns
debt_df2_sub <- debt_df2 %>%
  select("Country.Code","Year","Net financial flows, others (NFL, current US$)") %>%
  head(5)
debt_df2_sub
```

```
##   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. Combine two data frames: debt_df and country_df

```r
# Select specified columns
country_df_select <- country_df %>%
  select("Country.Code","Region","IncomeGroup","Country.Name")
debt_df3 <- debt_df2 %>%
  left_join(country_df_select, by = "Country.Code")
# Check the number of columns
# ncol2=ncol(debt_df2)
# ncol3=ncol(debt_df3)
# print(paste("debt_df2:The number of columns:", ncol2))
# print(paste("debt_df3:The number of columns:", ncol3))

# Preview the merged data with selected columns
debt_df3_sub <- debt_df3 %>%
  select("Country.Name","IncomeGroup", "Year", "Total reserves in months of imports") %>%
  head(3)
debt_df3_sub
```

```
##       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
##   Total reserves in months of imports
## 1                            2.825546
## 2                            2.730040
## 3                           14.861069
```

Q5. Rename 5 columns from their original names to the new names

```r
# Rename specified columns in debt_df3
debt_df3 <- debt_df3 %>%
  rename(
```

```
    Total_reserves = 'Total reserves in months of imports',
    External_debt = 'External debt stocks, total (DOD, current US$)',
    Financial_flow = 'Net financial flows, bilateral (NFL, current US$)',
    Imports = 'Imports of goods, services and primary income (BoP, current US$)',
    IFC = 'IFC, private nonguaranteed (NFL, US$)'
  )

# Select and preview the renamed columns
debt_df3_sub2 <- debt_df3 %>%
  select('Country.Code','Year','Total_reserves','External_debt','Financial_flow','Imports','IFC') %>%
  head(5)
debt_df3_sub2
```

```
##   Country.Code      Year Total_reserves External_debt Financial_flow
## 1          MEX year_1995       2.825546  166734000000     9398190731
## 2          EGY year_2013       2.730040   46534987115     7233642176
## 3          BRA year_2017      14.861069  543000000000     6506490468
## 4          PAK year_2018       1.905231  100199000000     6201281870
## 5          EGY year_2016       3.885411   69188517055     5714011601
##        Imports       IFC
## 1  72391910000         0
## 2  72685700000 -42864095
## 3 301961000000 397855350
## 4  74555877000  11389136
## 5  73019900000  77244772
```

Q6. Create debt_summary

```
# Group data by Region and calculate summary statistics
debt_summary <- debt_df3 %>%
  group_by(Region) %>%
  summarize(
    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)
  )
print(debt_summary)
```

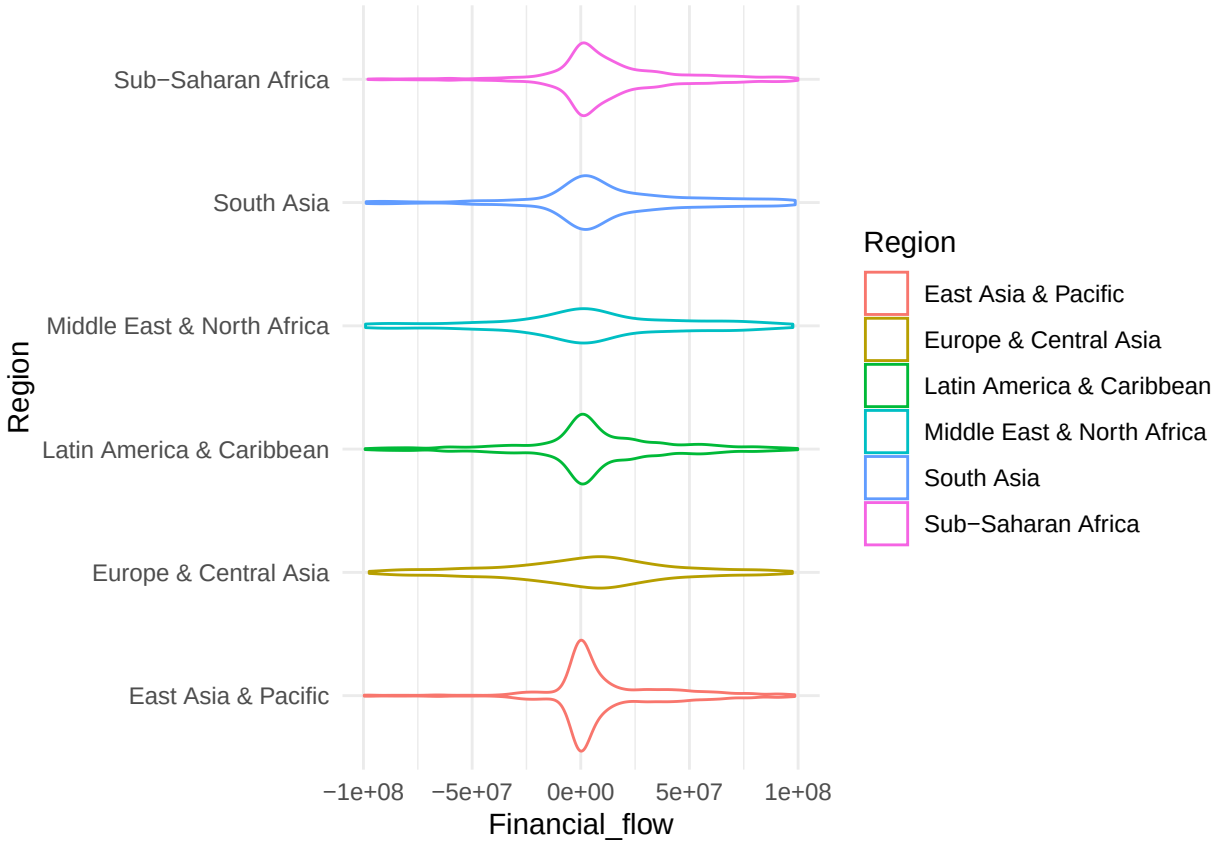
```
## # A tibble: 7 x 5
##   Region                     TR_mn      ED_md FF_quantile    IFC_sd
##   <chr>                      <dbl>      <dbl>       <dbl>     <dbl>
## 1 East Asia & Pacific         5.19 2248479410   -2357020. 52498519.
## 2 Europe & Central Asia       3.58 8237728122  -53631246. 50820255.
## 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. Create a violin plot of "Financial_flow" for each of the regions.

```
# Filter out missing and extreme values
debt_filter_df <- debt_df3 %>%
  filter(!is.na(Financial_flow)) %>%
  filter(Financial_flow >-10^8 & Financial_flow < 10^8)

#Check columns' names
#names(debt_filter_df)

#Create a violin polt
ggplot(debt_filter_df, aes( x = Financial_flow, y = Region, color = Region)) +
  geom_violin() +
  labs( x = "Financial_flow", y = "Region") +
  theme_minimal()
```



Q8. Create a plot which displays the "Total_reserves" as a function of the years (from 1960 to 2023)

```
# Select specified Country and Years
debt_filter_df2 <- debt_df3 %>%
  filter(Country.Name %in% c("Italy", "France", "United Kingdom", "Sudan", "Afghanistan", "Brazil")) %>%
  mutate(Year = as.numeric(stringr::str_extract(Year, "\\d{4}"))) %>%
  filter(Year >= '1960' & Year <= '2023') %>%
  filter(!is.na(Total_reserves))
```

```r
# Create a plot
ggplot(debt_filter_df2, aes(x = Year, y = Total_reserves, color = Country.Name)) +
  geom_line(linewidth = 0.5) +
  geom_point(size = 1 # Add points
             ) +
  labs(x = "Year", y = "Total Reserves", color = "Country Name") +
  facet_wrap(~ IncomeGroup) +
  theme(
    panel.border = element_rect(color = "black", fill = NA, linewidth = 0.5)  # Add black border to eac
  ) +
  scale_x_continuous(breaks=seq(1960,2023,by=10)) # Set X axis ticks every 10 years
```