**Big Data 2 - Mid Term**

Saraniyan Selvakumar - N01605920

**1.**

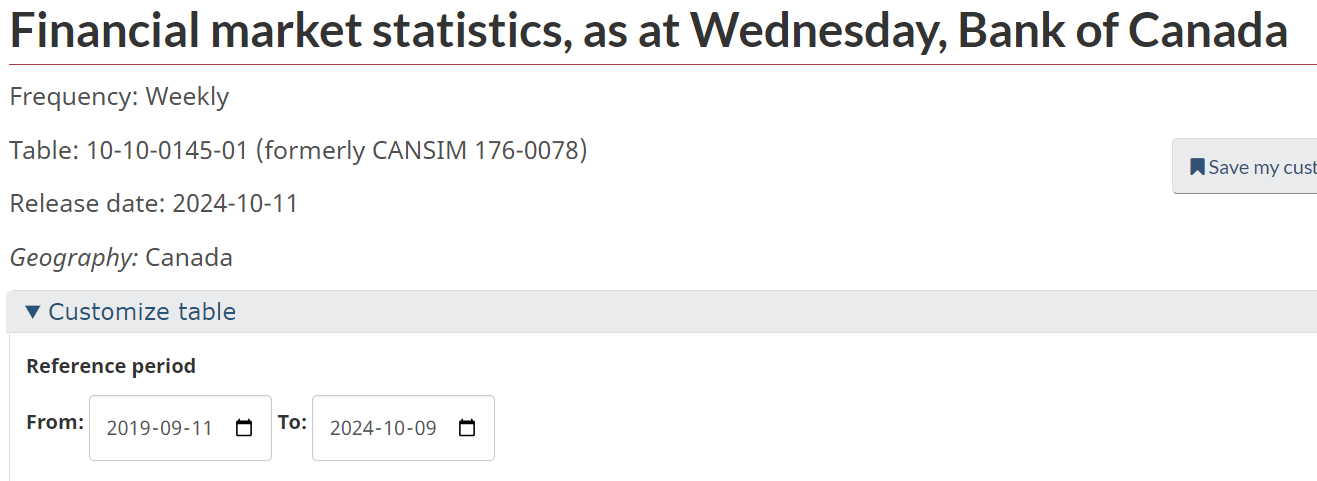
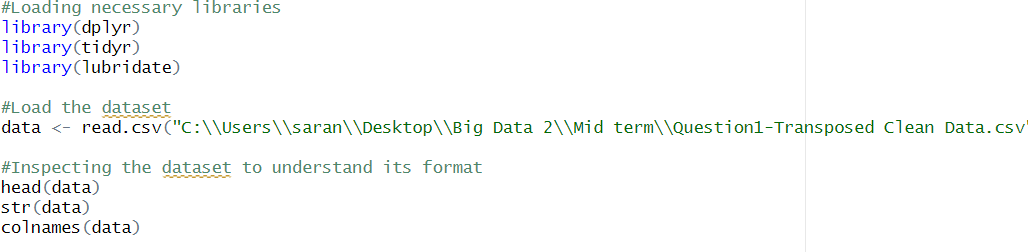


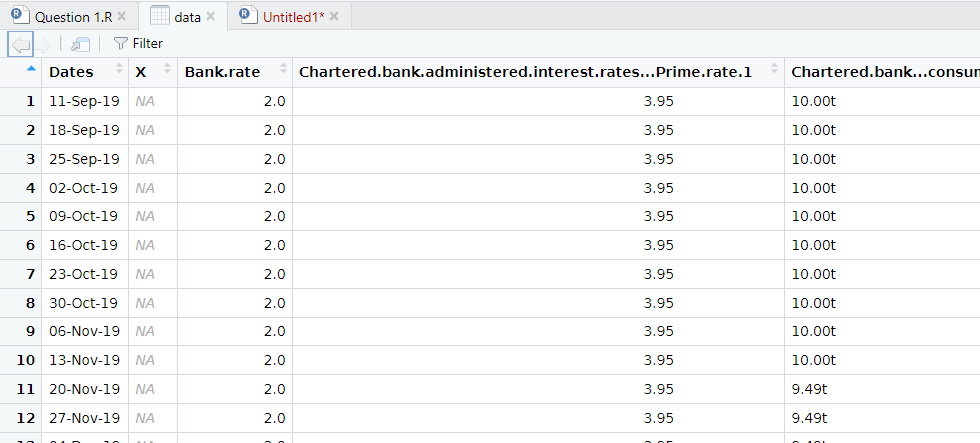
Table: 10-10-0145-01 (formerly CANSIM 176-0078)  
Release date: 2024-10-11

***Before loading the csv file into R, I transposed the necessary data from file a named "1010014501-eng" to a csv file named "Question1-Transposed Clean Data". Both file has been uploaded to Blackboard submission for your reference.***

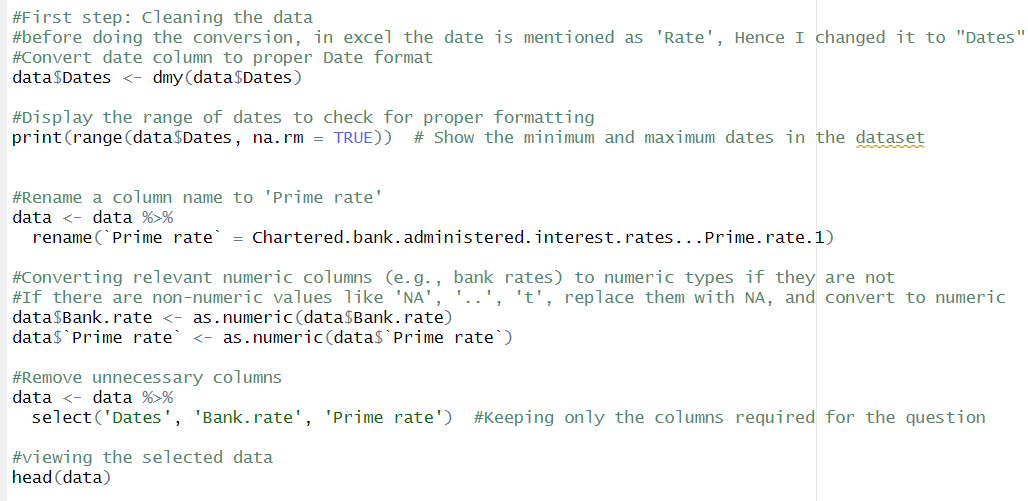
a. The average bank rate and prime rate (i.e Chartered bank administered interest rates - Prime rate) in each 6-months period over the past 5 years. Do not show other variables in the output

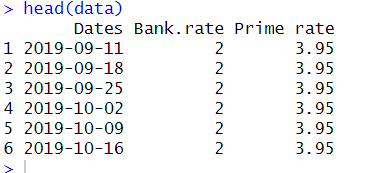
At first we load in the relevant library: here I’ve researched a library called lubridate which provides functions for working with dates and times.



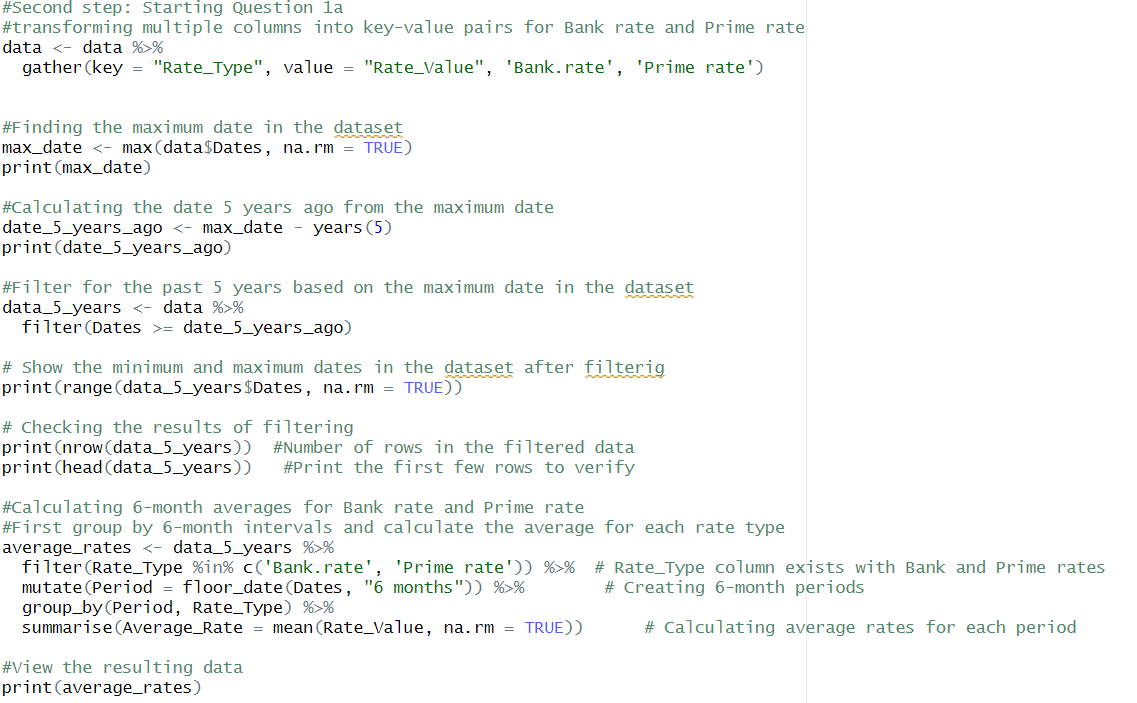


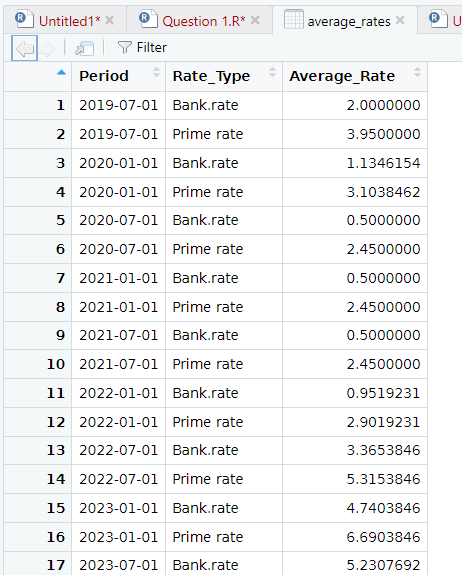
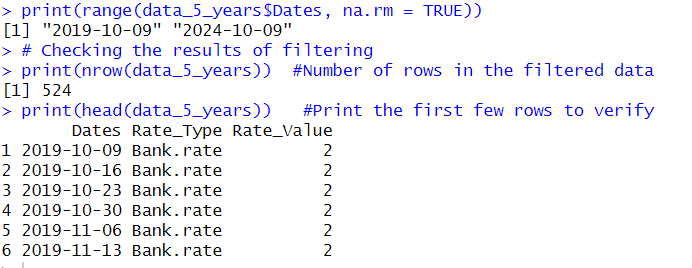
Here I’m loading the dataset by converting the csv file into a data frame, making it ready for analysis.





The first thing I did was clean the data before starting question 1a. The datasets contain irrelevant rows/columns and missing values that must be removed. Therefore, I utilized functions like select() to maintain only the relevant columns. I renamed “Chartered bank administered interest rates - Prime rate 1” to “Prime rate” and changed the date format to year-month-date.

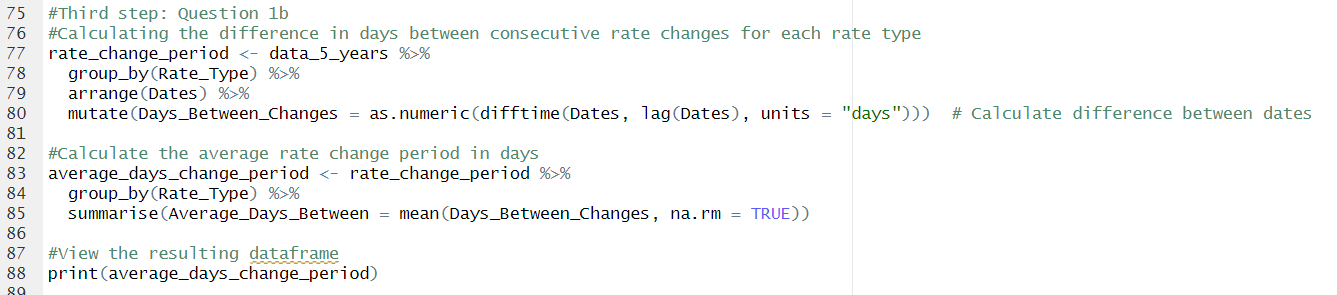


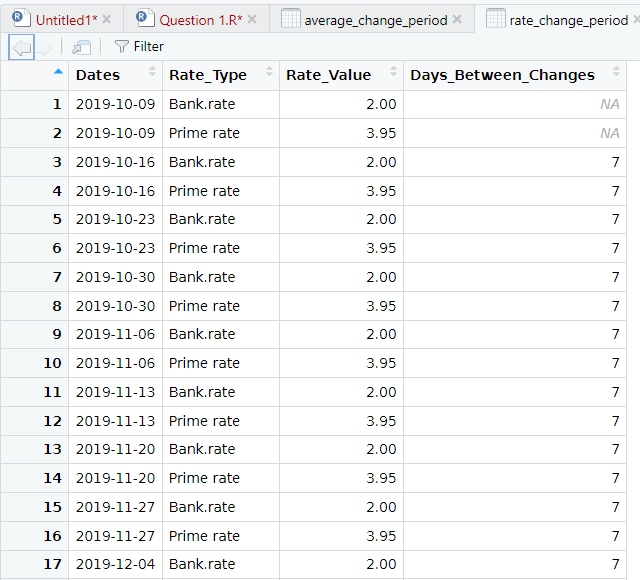


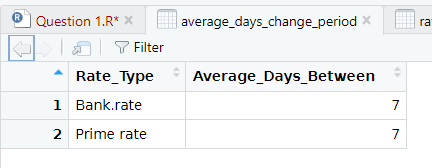
Here, I'm simply storing records from the last 5 years by comparing the Dates column to the date calculated (date\_5\_years\_ago) as 5 years before the dataset's most recent date. The floor\_date() function in Lubridate divides dates into 6-month intervals by rounding down each date to the nearest earlier 6-month boundary. For example, "2024-07-15" will be grouped with "2024-07-01."

After defining the 6-month intervals, I used group\_by() to group data by period (Period) and rate type (Rate\_Type - either "Bank rate" or "Prime rate"). The summarise() function determines the average Rate\_Value for each group, ignoring NA values.

b. The average rate change period in days (i.e how many days in average between consecutive rate changes)?





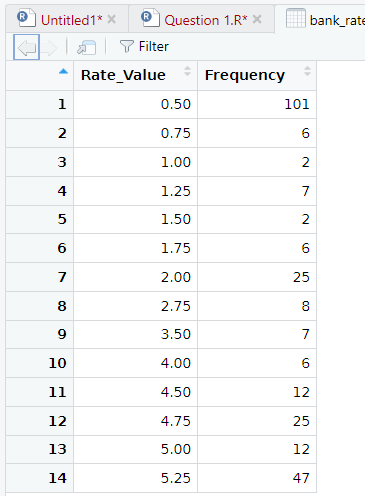
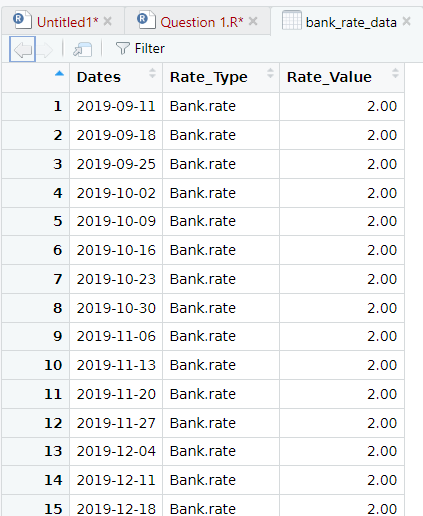
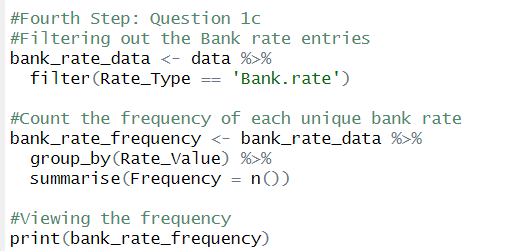


This section computes the difference in days between each rate change. The arrange() function sorts the data by date for each rate type. The lag() function adjusts the Dates column to determine the difference between the current and previous dates using difftime(). The difftime() functions calculates the time difference (in days) between consecutive rate changes.

After determining the number of days between consecutive rate changes, I calculated the average using mean(). The na.rm = TRUE option ensures that missing values (NAs) are ignored in the calculation.

The result displays the average number of days between rate adjustments for both the "bank rate" and the "prime rate."

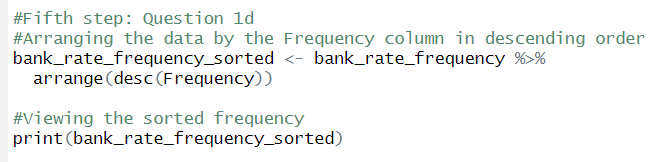
c. Unique (Distinct) bank rates and the frequency of each (i.e the repetitions: in how many months did each bank rate appear).

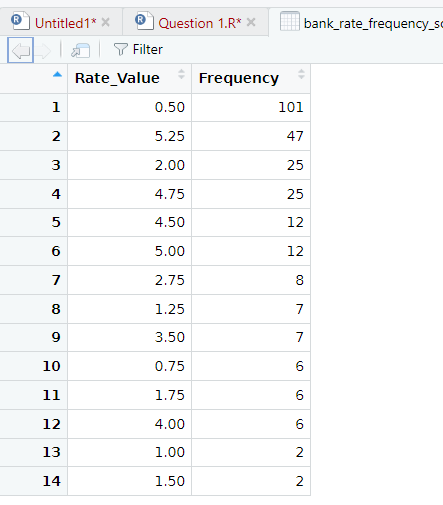


This step filters the data so that only rows with Rate\_Type="Bank rate" are maintained. It separates the bank rate from the prime rate data.

After isolating the "Bank rate" data, group\_by() sorts the data by the unique Rate\_Value, then summarise(n()) counts how many times each rate appears in the dataset. The result displays the number of times each unique bank rate appears throughout the time period selected.

d. Arrange the data frame above by frequencies (bank rate repetitions)





The arrange() function sorts the data frame's rows, and desc(Frequency) assures that the sorting is done in decreasing order of occurrences. This will display the most frequently used bank rates at the top of the table.

**Full code for question 1**:

#Loading necessary libraries

library(dplyr)

library(tidyr)

library(lubridate)

#Load the dataset

data <- read.csv("C:\\Users\\saran\\Desktop\\Big Data 2\\Mid term\\Question1-Transposed Clean Data.csv")

#Inspecting the dataset to understand its format

head(data)

str(data)

colnames(data)

#First step: Cleaning the data

#before doing the conversion, in excel the date is mentioned as 'Rate', Hence I changed it to "Dates"

#Convert date column to proper Date format

data$Dates <- dmy(data$Dates)

#Display the range of dates to check for proper formatting

print(range(data$Dates, na.rm = TRUE)) # Show the minimum and maximum dates in the dataset

#Rename a column name to 'Prime rate'

data <- data %>%

rename(`Prime rate` = Chartered.bank.administered.interest.rates...Prime.rate.1)

#Converting relevant numeric columns (e.g., bank rates) to numeric types if they are not

#If there are non-numeric values like 'NA', '..', 't', replace them with NA, and convert to numeric

data$Bank.rate <- as.numeric(data$Bank.rate)

data$`Prime rate` <- as.numeric(data$`Prime rate`)

#Remove unnecessary columns

data <- data %>%

select('Dates', 'Bank.rate', 'Prime rate') #Keeping only the columns required for the question

#viewing the selected data

head(data)

#Second step: Starting Question 1a

#transforming multiple columns into key-value pairs for Bank rate and Prime rate

data <- data %>%

gather(key = "Rate\_Type", value = "Rate\_Value", 'Bank.rate', 'Prime rate')

#Finding the maximum date in the dataset

max\_date <- max(data$Dates, na.rm = TRUE)

print(max\_date)

#Calculating the date 5 years ago from the maximum date

date\_5\_years\_ago <- max\_date - years(5)

print(date\_5\_years\_ago)

#Filter for the past 5 years based on the maximum date in the dataset

data\_5\_years <- data %>%

filter(Dates >= date\_5\_years\_ago)

# Show the minimum and maximum dates in the dataset after filterig

print(range(data\_5\_years$Dates, na.rm = TRUE))

# Checking the results of filtering

print(nrow(data\_5\_years)) #Number of rows in the filtered data

print(head(data\_5\_years)) #Print the first few rows to verify

#Calculating 6-month averages for Bank rate and Prime rate

#First group by 6-month intervals and calculate the average for each rate type

average\_rates <- data\_5\_years %>%

filter(Rate\_Type %in% c('Bank.rate', 'Prime rate')) %>% # Rate\_Type column exists with Bank and Prime rates

mutate(Period = floor\_date(Dates, "6 months")) %>% # Creating 6-month periods

group\_by(Period, Rate\_Type) %>%

summarise(Average\_Rate = mean(Rate\_Value, na.rm = TRUE)) # Calculating average rates for each period

#View the resulting data

print(average\_rates)

#Third step: Question 1b

#Calculating the difference in days between consecutive rate changes for each rate type

rate\_change\_period <- data\_5\_years %>%

group\_by(Rate\_Type) %>%

arrange(Dates) %>%

mutate(Days\_Between\_Changes = as.numeric(difftime(Dates, lag(Dates), units = "days"))) # Calculate difference between dates

#Calculate the average rate change period in days

average\_days\_change\_period <- rate\_change\_period %>%

group\_by(Rate\_Type) %>%

summarise(Average\_Days\_Between = mean(Days\_Between\_Changes, na.rm = TRUE))

#View the resulting dataframe

print(average\_days\_change\_period)

#Fourth Step: Question 1c

#Filtering out the Bank rate entries

bank\_rate\_data <- data %>%

filter(Rate\_Type == 'Bank.rate')

#Count the frequency of each unique bank rate

bank\_rate\_frequency <- bank\_rate\_data %>%

group\_by(Rate\_Value) %>%

summarise(Frequency = n())

#Viewing the frequency

print(bank\_rate\_frequency)

#Fifth step: Question 1d

#Arranging the data by the Frequency column in descending order

bank\_rate\_frequency\_sorted <- bank\_rate\_frequency %>%

arrange(desc(Frequency))

#Viewing the sorted frequency

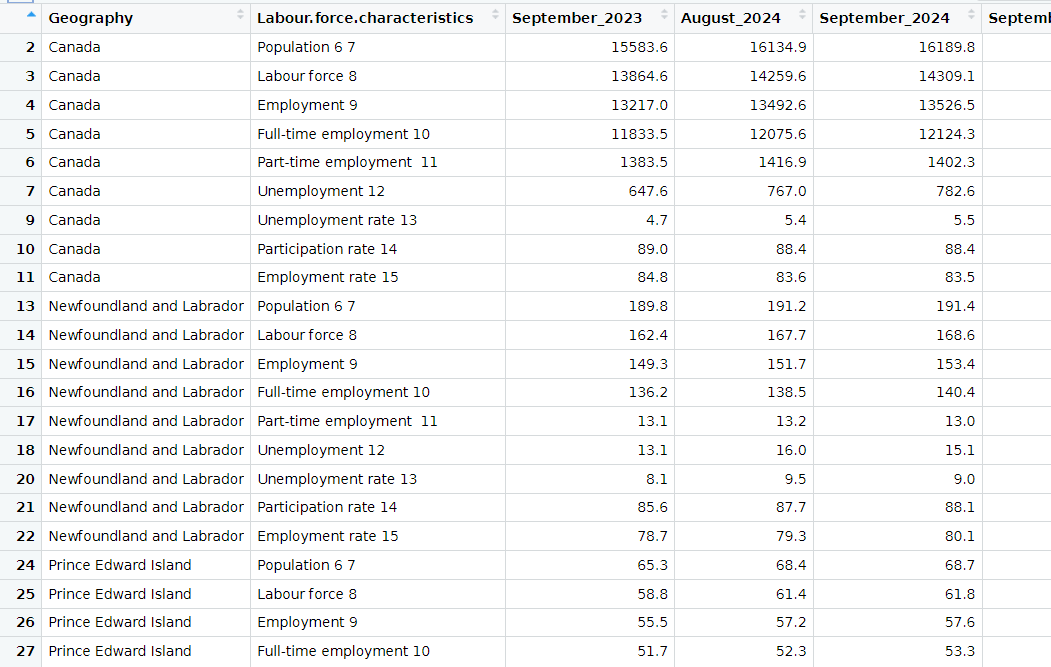
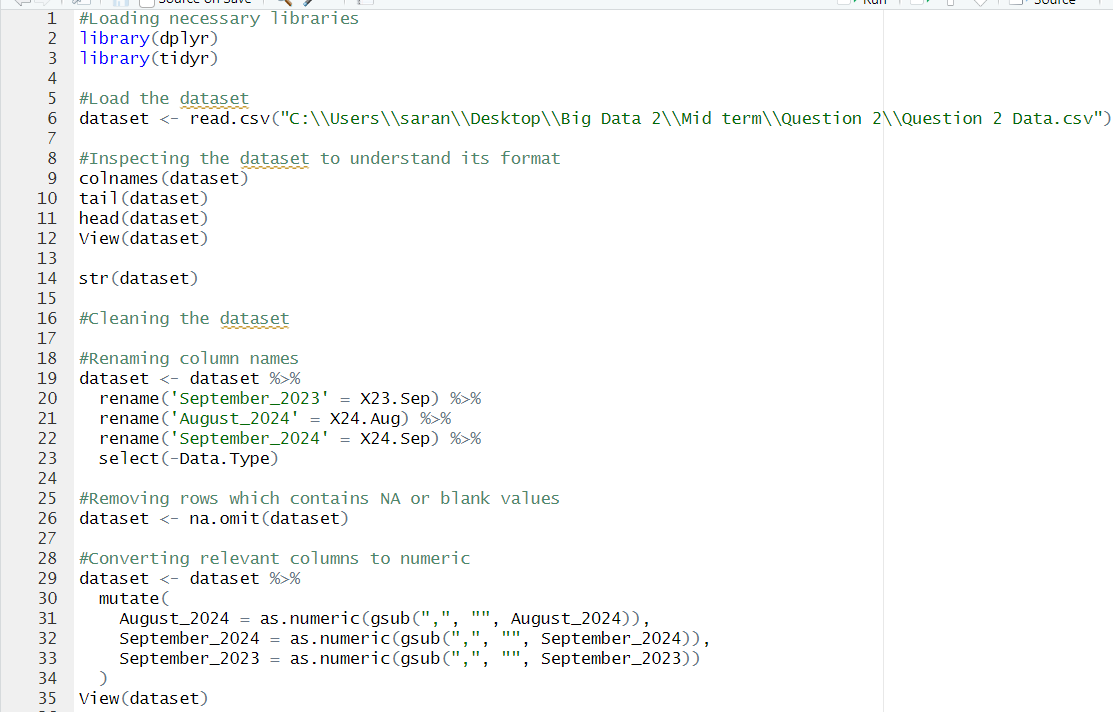
print(bank\_rate\_frequency\_sorted)

**2.**



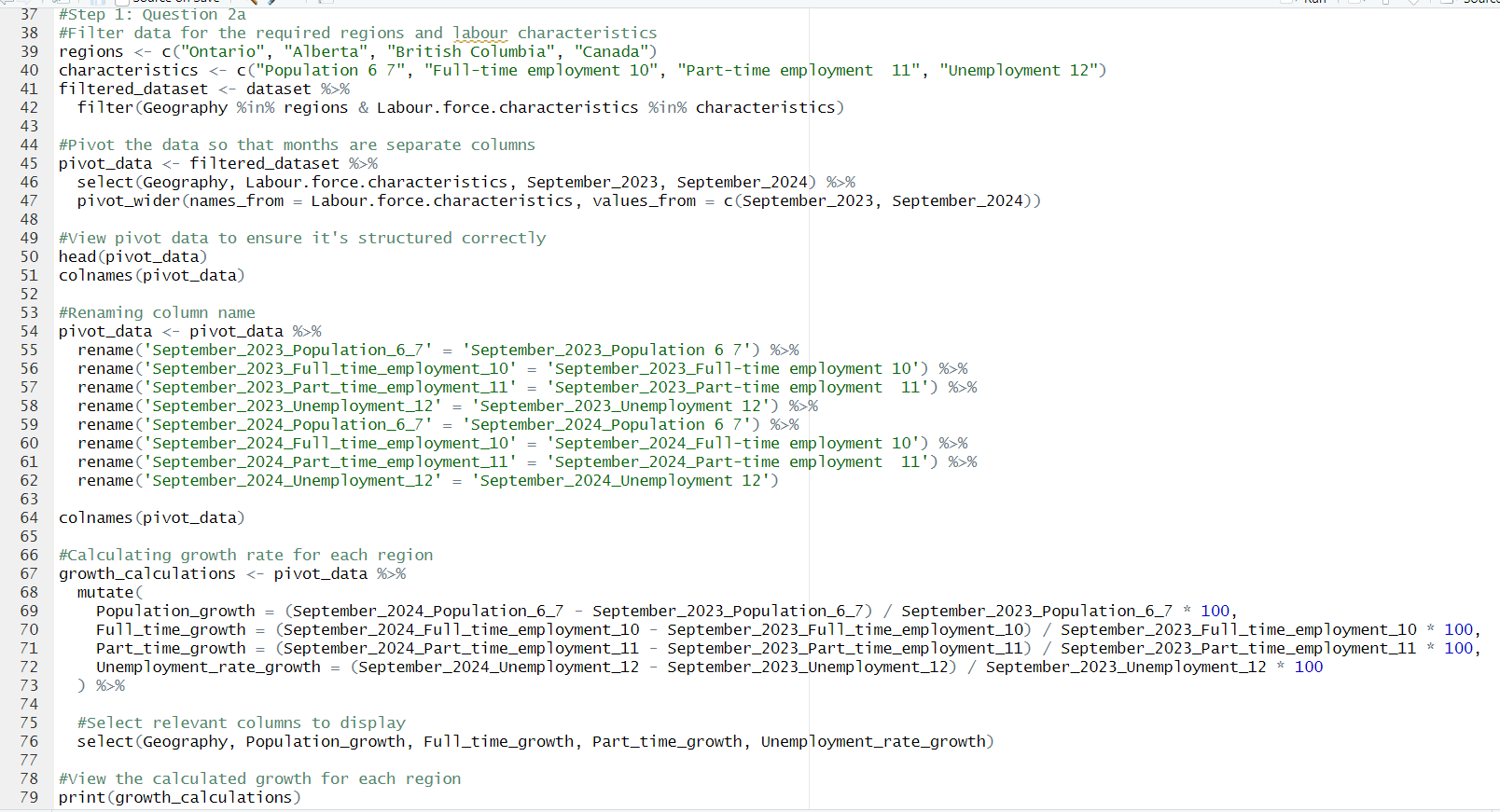
***Before loading the csv file into R, I cleaned some of the necessary data from a file named "1410028703-eng Sept 2024" to a csv file named "Question 2 Data". Both file has been uploaded to Blackboard submission for your reference.***

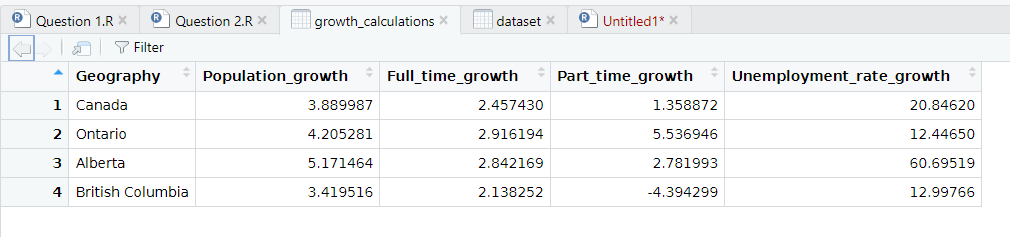
At first we load in the relevant library and inspect the structure and details of the file before cleaning the rest of the data. Below I have shown the clean data frame and here onwards started the questions.



a. Growth of population, Full-time employment, Part-time employment and Unemployment rates between first and last months, in Ontario, Alberta, BC and overall Canada. Do the calculation for these 4 regions only

***Note: In this dataset I have considered last month as September 2024 and first month as September 2023.***



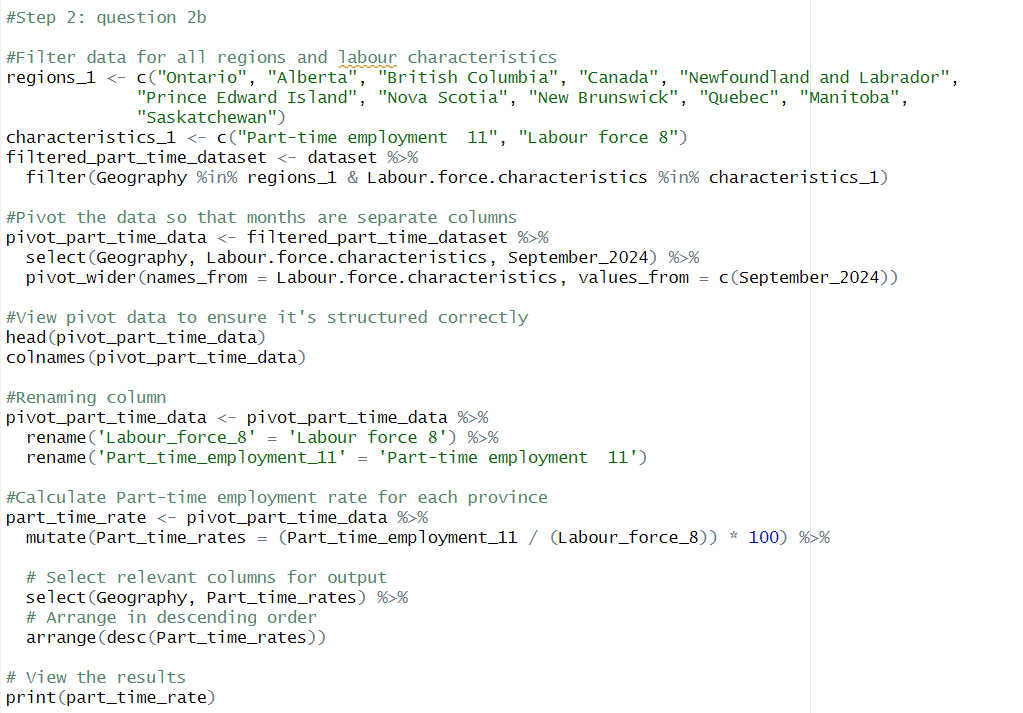


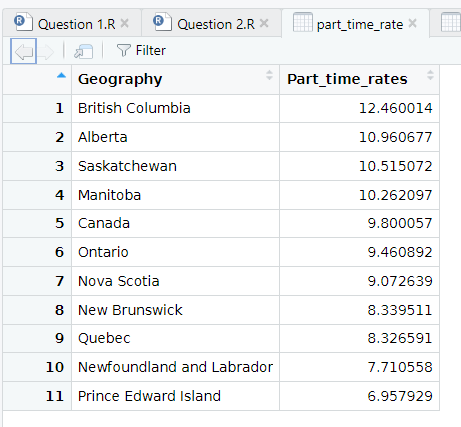
I'm limiting the original dataset to only include data from four regions: Ontario, Alberta, British Columbia, and Canada. In addition, there is data on labor characteristics such as population, full-time employment, part-time employment, and unemployment. (Pivot table process) The next stage is to separate each labor attribute (such as population and full-time employment) into its own column. Instead of having these features in one column, they are now separated into two columns containing data for September 2023 and September 2024, respectively. After pivoting, I renamed the columns to improve readability and consistency.

Here, I'm calculating the percentage growth for each region between September 2023 and September 2024. using the formula: Growth = (newValue – oldValue)/oldValue\*100 %

Finally, I select and display only the relevant columns, which include the Geography and the calculated growth percentages for population, full-time employment, part-time employment, and unemployment rates.

b. Part-time employment rate (expressed as a percentage of the labour force) of each province, in descending Part-time employment rate order. Use last month data only, for all of the regions.

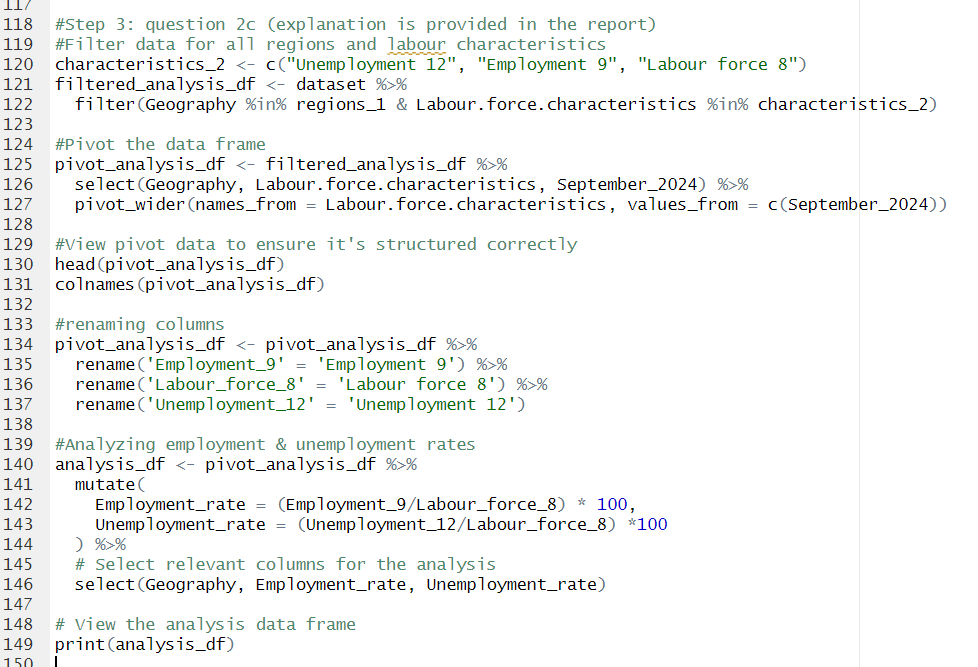


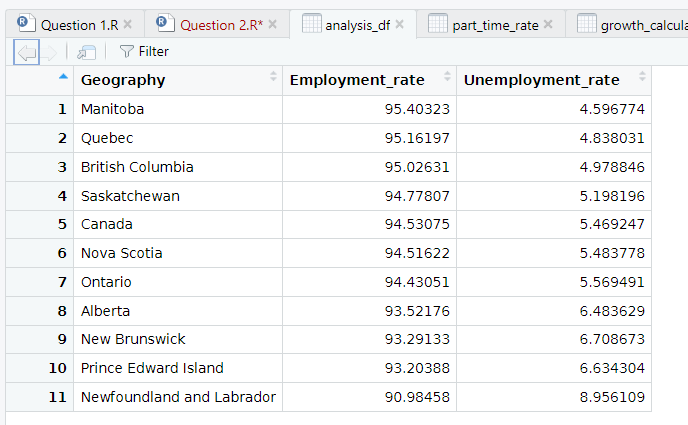


At first I'm filtering the dataset to only include all regions (geography) and two labor characteristics: Part-time employment 11 and Labour force 8. After filtering, I'm pivoting the dataset so that each labor feature (such as Part-time employment 11 and Labour force 8) has its own column, with values from September 2024.

Next I'm calculating the part-time employment rate for each province. The part-time employment rate is calculated by dividing the number of part-time workers (Part\_time\_employment\_11) by the total labor force (Labour\_force\_8) and then multiplying by 100 to get a percentage. Finally I select only the Geography and the calculated part-time rates for output and sort the provinces in descending order based on the part-time employment rate, so that the region with the highest part-time rate comes first.

c. Generate a data frame to analyse the employment/unemployment rate in each province. Use last month data only, for all of the regions. What can you conclude (based on this data) in regard to provincial employment opportunities?





Here I'm filtering the dataset to include regions\_1 (created in Step 2: question 2b) and three key labor characteristics:

Unemployment 12: The number of unemployed people.

Employment 9: The number of employed people.

Labour force 8: The total number of people in the labor force (both employed and unemployed).

After filtering the dataset, I'm pivoting the data so that the labor characteristics (Unemployment 12, Employment 9, Labour force 8) become separate columns, with the data from September 2024 as values.

Next step is the calculation.

Employment rate: The percentage of people in the labor force who are employed, which is calculated by dividing the number of employed people (Employment\_9) by the total labor force (Labour\_force\_8) and multiplying by 100.

Unemployment rate: The percentage of people in the labor force who are unemployed, which is calculated by dividing the number of unemployed people (Unemployment\_12) by the total labor force (Labour\_force\_8) and multiplying by 100.

Finally I select only the relevant columns to display: Geography, calculated Employment rate, and the Unemployment rate for each region.

Based on the data analysis, below are a few major conclusions about provincial employment opportunities:

High employment rates in the region:

All provinces have high employment rates, ranging from 90% to 95%, indicating that the majority of the population is employed. This points to solid labor markets across the country.

Lowest unemployment rates:

Manitoba (4.6%), Quebec (4.8%), and British Columbia (5.0%) have the lowest unemployment rates, reflecting good economic circumstances and more work prospects in these provinces than others.

Increased Unemployment in Some Provinces:

Newfoundland and Labrador has the highest unemployment rate at 9%, indicating less work prospects or worse economic issues in this province than the rest of Canada.

Prince Edward Island and New Brunswick also have slightly higher unemployment rates, at roughly 6.6% and 6.7%, respectively, which remain relatively high in comparison to other provinces.

In conclusion, Manitoba, Quebec, and British Columbia appear to offer the best employment possibilities with low unemployment rates, but Newfoundland and Labrador has the most tough labor market with a higher unemployment rate. When looking for work, this information might assist in picking places with the best job opportunities.

**Full code on question 2:**

#Loading necessary libraries

library(dplyr)

library(tidyr)

#Load the dataset

dataset <- read.csv("C:\\Users\\saran\\Desktop\\Big Data 2\\Mid term\\Question 2\\Question 2 Data.csv")

#Inspecting the dataset to understand its format

colnames(dataset)

tail(dataset)

head(dataset)

View(dataset)

str(dataset)

#Cleaning the dataset

#Renaming and dropping columns

dataset <- dataset %>%

rename('September\_2023' = X23.Sep) %>%

rename('August\_2024' = X24.Aug) %>%

rename('September\_2024' = X24.Sep) %>%

select(-Data.Type)

#Removing rows which contains NA or blank values

dataset <- na.omit(dataset)

#Converting relevant columns to numeric

dataset <- dataset %>%

mutate(

August\_2024 = as.numeric(gsub(",", "", August\_2024)),

September\_2024 = as.numeric(gsub(",", "", September\_2024)),

September\_2023 = as.numeric(gsub(",", "", September\_2023))

)

View(dataset)

#Step 1: Question 2a

#Filter data for the required regions and labour characteristics

regions <- c("Ontario", "Alberta", "British Columbia", "Canada")

characteristics <- c("Population 6 7", "Full-time employment 10", "Part-time employment 11", "Unemployment 12")

filtered\_dataset <- dataset %>%

filter(Geography %in% regions & Labour.force.characteristics %in% characteristics)

#Pivot the data so that months are separate columns

pivot\_data <- filtered\_dataset %>%

select(Geography, Labour.force.characteristics, September\_2023, September\_2024) %>%

pivot\_wider(names\_from = Labour.force.characteristics, values\_from = c(September\_2023, September\_2024))

#View pivot data to ensure it's structured correctly

head(pivot\_data)

colnames(pivot\_data)

#Renaming column name

pivot\_data <- pivot\_data %>%

rename('September\_2023\_Population\_6\_7' = 'September\_2023\_Population 6 7') %>%

rename('September\_2023\_Full\_time\_employment\_10' = 'September\_2023\_Full-time employment 10') %>%

rename('September\_2023\_Part\_time\_employment\_11' = 'September\_2023\_Part-time employment 11') %>%

rename('September\_2023\_Unemployment\_12' = 'September\_2023\_Unemployment 12') %>%

rename('September\_2024\_Population\_6\_7' = 'September\_2024\_Population 6 7') %>%

rename('September\_2024\_Full\_time\_employment\_10' = 'September\_2024\_Full-time employment 10') %>%

rename('September\_2024\_Part\_time\_employment\_11' = 'September\_2024\_Part-time employment 11') %>%

rename('September\_2024\_Unemployment\_12' = 'September\_2024\_Unemployment 12')

colnames(pivot\_data)

#Calculating growth rate for each region

growth\_calculations <- pivot\_data %>%

mutate(

Population\_growth = (September\_2024\_Population\_6\_7 - September\_2023\_Population\_6\_7) / September\_2023\_Population\_6\_7 \* 100,

Full\_time\_growth = (September\_2024\_Full\_time\_employment\_10 - September\_2023\_Full\_time\_employment\_10) / September\_2023\_Full\_time\_employment\_10 \* 100,

Part\_time\_growth = (September\_2024\_Part\_time\_employment\_11 - September\_2023\_Part\_time\_employment\_11) / September\_2023\_Part\_time\_employment\_11 \* 100,

Unemployment\_rate\_growth = (September\_2024\_Unemployment\_12 - September\_2023\_Unemployment\_12) / September\_2023\_Unemployment\_12 \* 100

) %>%

#Select relevant columns to display

select(Geography, Population\_growth, Full\_time\_growth, Part\_time\_growth, Unemployment\_rate\_growth)

#View the calculated growth for each region

print(growth\_calculations)

#Step 2: question 2b

#Filter data for all regions and labour characteristics

regions\_1 <- c("Ontario", "Alberta", "British Columbia", "Canada", "Newfoundland and Labrador",

"Prince Edward Island", "Nova Scotia", "New Brunswick", "Quebec", "Manitoba",

"Saskatchewan")

characteristics\_1 <- c("Part-time employment 11", "Labour force 8")

filtered\_part\_time\_dataset <- dataset %>%

filter(Geography %in% regions\_1 & Labour.force.characteristics %in% characteristics\_1)

#Pivot the data so that months are separate columns

pivot\_part\_time\_data <- filtered\_part\_time\_dataset %>%

select(Geography, Labour.force.characteristics, September\_2024) %>%

pivot\_wider(names\_from = Labour.force.characteristics, values\_from = c(September\_2024))

#View pivot data to ensure it's structured correctly

head(pivot\_part\_time\_data)

colnames(pivot\_part\_time\_data)

#Renaming column

pivot\_part\_time\_data <- pivot\_part\_time\_data %>%

rename('Labour\_force\_8' = 'Labour force 8') %>%

rename('Part\_time\_employment\_11' = 'Part-time employment 11')

#Calculate Part-time employment rate for each province

part\_time\_rate <- pivot\_part\_time\_data %>%

mutate(Part\_time\_rates = (Part\_time\_employment\_11 / (Labour\_force\_8)) \* 100) %>%

# Select relevant columns for output

select(Geography, Part\_time\_rates) %>%

# Arrange in descending order

arrange(desc(Part\_time\_rates))

# View the results

print(part\_time\_rate)

#Step 3: question 2c (explanation is provided in the report)

#Filter data for all regions and labour characteristics

characteristics\_2 <- c("Unemployment 12", "Employment 9", "Labour force 8")

filtered\_analysis\_df <- dataset %>%

filter(Geography %in% regions\_1 & Labour.force.characteristics %in% characteristics\_2)

#Pivot the data frame

pivot\_analysis\_df <- filtered\_analysis\_df %>%

select(Geography, Labour.force.characteristics, September\_2024) %>%

pivot\_wider(names\_from = Labour.force.characteristics, values\_from = c(September\_2024))

#View pivot data to ensure it's structured correctly

head(pivot\_analysis\_df)

colnames(pivot\_analysis\_df)

#renaming columns

pivot\_analysis\_df <- pivot\_analysis\_df %>%

rename('Employment\_9' = 'Employment 9') %>%

rename('Labour\_force\_8' = 'Labour force 8') %>%

rename('Unemployment\_12' = 'Unemployment 12')

#Analyzing employment & unemployment rates

analysis\_df <- pivot\_analysis\_df %>%

mutate(

Employment\_rate = (Employment\_9/Labour\_force\_8) \* 100,

Unemployment\_rate = (Unemployment\_12/Labour\_force\_8) \*100

) %>%

# Select relevant columns for the analysis

select(Geography, Employment\_rate, Unemployment\_rate) %>%

#For proper analysis

arrange(desc(Employment\_rate))

# View the analysis data frame

print(analysis\_df)