Interest Rates Data Aggregation Tool

Requirement: To pull the data for the Reserve/Central Bank Interest Rates of the following countries for the past 12 months (May 2023 to April 2024) from public domains

Process : A python script is built to fetch the interest rate data for various countries (USA, Canada, Europe, Australia, England) from their respective central bank websites using libraries like `requests` and `BeautifulSoup` to parse HTML content. It then cleans and formats the data into a single DataFrame containing monthly interest rates for each country. The data is pulled from **publicly available** resources on the websites of the Federal Reserve, Bank of Canada, Reserve Bank of Australia, European Central Bank, and Bank of England

Respective URLS :

1. USA: https://www.federalreserve.gov/feeds/prates.html#3443
   From this the actual URL from which I have downloaded the CSV file
   with data from May-2023 to April-2024 is :

   https://www.federalreserve.gov/datadownload/Output.aspx?rel=H15&series=d7e27b7b09a3a7feae95b9c61781fcd8&lastobs=12&from=&to=&filetype=csv&label=include&layout=seriescolumn&type=package

2. United Kingdom/Bank of England:
   https://www.bankofengland.co.uk/boeapps/database/Bank-Rate.asp

   I have downloaded the CSV file which my python script is reading and the link for the same is : https://www.bankofengland.co.uk/boeapps/database/Bank-Rate.asp

3. Europe/European Central Bank: Marginal Lending Facility Rate
   For Europe I am using this link and parsing the table that has data directly from page : https://www.ecb.europa.eu/stats/policy_and_exchange_rates/key_ecb_interest_rates/html/index.en.html

4. Australia/Reserve Bank of Australia: Cash Rate Target
   For Australia, I am using this link and parsing the table that has interest rates directly from page: https://www.rba.gov.au/statistics/cash-rate/

5. Canada/Bank of Canada: Canadian Overnight Repo Rate Average
 For Canada, if you open this link :https://www.bankofcanada.ca/rates/interest-rates/corra/, you will find a page opening which has CSV file to download and I have downloaded it whose link is :https://www.bankofcanada.ca/valet/observations/group/CORRA/csv where I am parsing the csv file and extracting the required data. This data was starting from year 1997 and it is a on daily-level data. But as per our requirement of the business situation, I have considered the last available day for the interest rates for each month which I have done in Python Script.

Code Explaination:

Firstly we need to understand to do web scraping , what all packages are needed i.e. Requests and BeautifulSoup which are again needed to be PIP Installed in python terminal.

pip install requests beautifulsoup4 pandas lxml

I have built 5 different functions for the 5 countries as data was not in same format, some I had to download as CSV files , some I had table on page itself which I have parsed using html parser. There is one “ main” section of the code where I am calling the functions, creating a final dataframe and that I have exported as CSV file to built the line Chart in Google Sheet.

Importing Libraries:

- `requests`: This library is used to make HTTP requests to web servers in order to retrieve data.
- `pandas`: This library is used for data analysis and manipulation. It's particularly useful for working with DataFrames and time series data.
- `BeautifulSoup`: This library is used for parsing HTML content. It helps extract specific data from websites.
- `csv`: This built-in Python module is used for reading and writing CSV (comma-separated values) files.
- `re`: This built-in Python module provides regular expression functionalities for pattern matching in text.
- `json`: This built-in Python module is used for working with JSON (JavaScript Object Notation) data.
- `os`: This built-in Python module provides functions for interacting with the operating system, such as getting the path to the Downloads folder.

Dictionary of URLs:

- `urls`: This dictionary stores the URLs for fetching interest rate data for different countries: USA, Australia, Europe, Canada, and England.

Functions of Each Country

1.fetch_USA_csv_as_df(url, columns_to_import):

- This function takes the URL for the USA Federal Reserve data and a list of column names (`columns_to_import`) as input.
- It uses `requests` to download the CSV file.
- It then parses the CSV content using `csv` and `pandas`.
- It extracts the specified columns (`columns_to_import`) and creates a pandas DataFrame.
- It does some data cleaning, such as removing unnecessary rows and formatting the date column.
- Finally, it returns the cleaned DataFrame containing USA interest rates.

2.fetch_England_csv_as_df(url, columns_to_import):

- This function is similar to `fetch_USA_csv_as_df` but for England.
- It uses `requests` with specific headers to bypass cookie warnings and download the HTML content.
- It uses `BeautifulSoup` to parse the HTML table containing interest rate data.
- It extracts the relevant columns (`columns_to_import`) and creates a pandas DataFrame.
- It cleans and formats the date column (`Date Changed`) and renames it to `Date` for consistency.
- It fills missing values using forward fill (`ffill`) and merges the data with a generated date range DataFrame.
- Finally, it returns the cleaned DataFrame containing England interest rates.

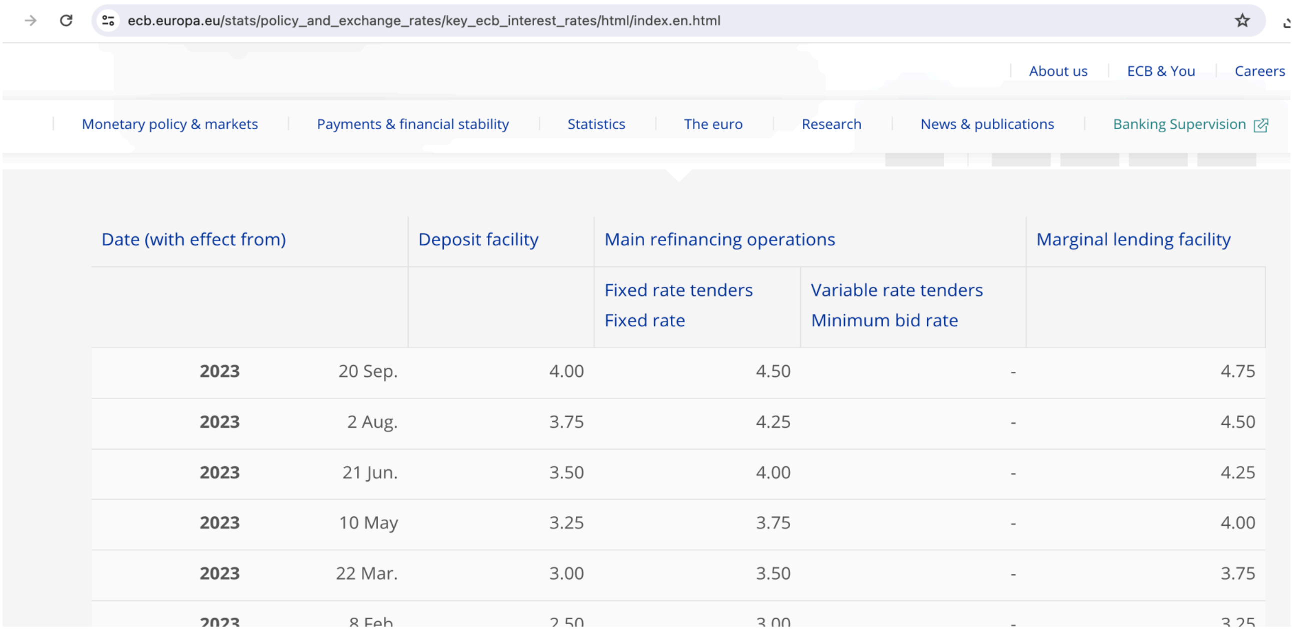
3. get_australia_data():

- This function fetches interest rate data for Australia.
- It follows a similar logic to `fetch_USA_csv_as_df`, using `requests` and `BeautifulSoup` to download and parse the HTML content.
- It extracts the relevant columns, cleans and formats the date (`Effective Date`), and renames it to `Date`.
- It filters data for the specified date range and merges it with a generated date range DataFrame.
- Finally, it returns the cleaned DataFrame containing Australia interest rates.

4. `get_europe_data()`:

- This function fetches interest rate data for Europe.
- It uses `requests` and `BeautifulSoup` to download and parse the HTML content.
- It combines multiple columns from the HTML table into a single column using list comprehension.
- It renames columns for better readability and extracts the `Marginal lending facility` column containing interest rates.
- It cleans and formats the date (`Period`), converting it to a datetime format and then back to "yyyy-mm" format.
- It filters data for the specified date range and merges it with a generated date range DataFrame.
- Finally, it returns the cleaned DataFrame containing Europe interest rates.

In parsing Europe data, one challenge I faced was with table as below snippet:



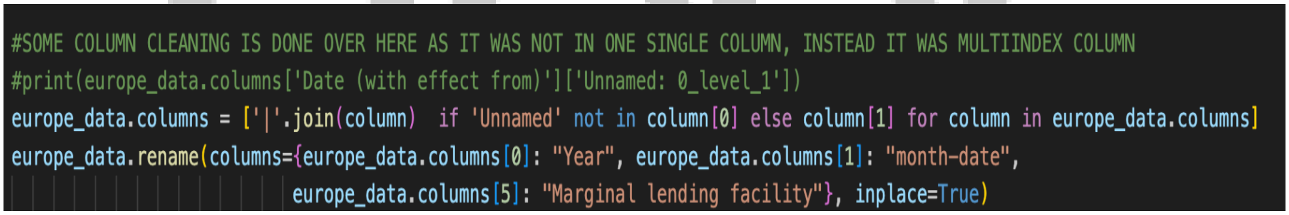
ecb.europa.eu/stats/policy_and_exchange_rates/key_ecb_interest_rates/html/index.en.html

About us | ECB & You | Careers

Monetary policy & markets | Payments & financial stability | Statistics | The euro | Research | News & publications | Banking Supervision

Date (with effect from)		Deposit facility	Main refinancing operations		Marginal lending facility
			Fixed rate tenders Fixed rate	Variable rate tenders Minimum bid rate	
2023	20 Sep.	4.00	4.50	-	4.75
2023	2 Aug.	3.75	4.25	-	4.50
2023	21 Jun.	3.50	4.00	-	4.25
2023	10 May	3.25	3.75	-	4.00
2023	22 Mar.	3.00	3.50	-	3.75
2023	8 Feb.	2.50	3.00	-	3.25

This was a multi-Indexing issue or we can say multi-level (hierarchical) column index :Date(with effect from), Main refinancing operations also has two levels of Indexing and so on. This was something different from other countries data. I have created some transformation to resolve this issue.



```
#SOME COLUMN CLEANING IS DONE OVER HERE AS IT WAS NOT IN ONE SINGLE COLUMN, INSTEAD IT WAS MULTIINDEX COLUMN
#print(europe_data.columns['Date (with effect from)']['Unnamed: 0_level_1'])
europe_data.columns = ['|'.join(column)  if 'Unnamed' not in column[0] else column[1] for column in europe_data.columns]
europe_data.rename(columns={europe_data.columns[0]: "Year", europe_data.columns[1]: "month-date",
                            europe_data.columns[5]: "Marginal lending facility"}, inplace=True)
```

The code uses a list comprehension to iterate over each `column` in `europe_data.columns`. Each `column` is a tuple representing the multi-level index.

Condition (`if 'Unnamed' not in column[0]`): Checks if the first level of the column index does not contain the string `'Unnamed'`.

- **`'|'.join(column)`**: If the condition is true, it joins the elements of the tuple `column` with a pipe (`|`) character. This creates a single string for each column by concatenating its multi-level index values, separated by `|`.
- **`else column[1]`**: If the condition is false (i.e., the first level of the column index contains `'Unnamed'`), it takes the second level of the column

index. This is typically used to handle cases where the DataFrame was read from a CSV with some unnamed levels in the index.

- The result is that `europe_data.columns` will be a flat list of strings, where each string is either the joined levels of the original multi-level index or just the second level if the first was `'Unnamed'.`
- **inplace=True**: This ensures the changes are made directly to the original DataFrame

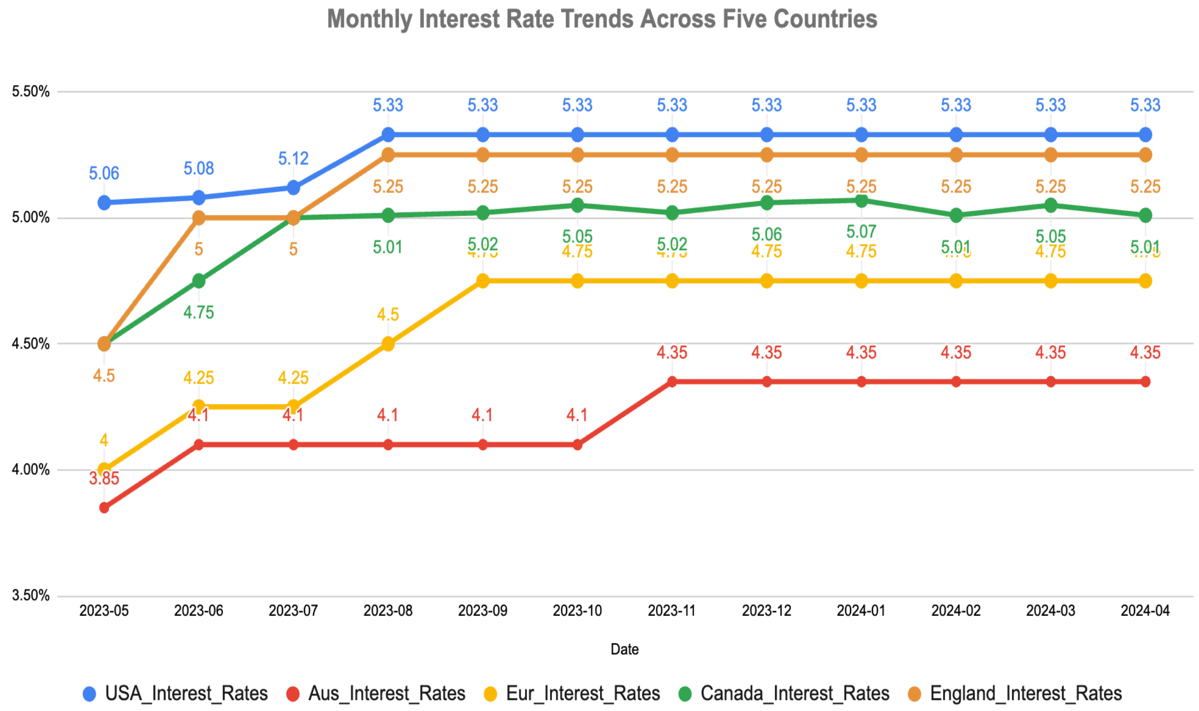
`5. fetch_Canada_csv_as_df(url, columns_to_import)`: This data was on daily level since 1997, I have taken the last available day and its corresponding Interest rate for our analysis.

- This function fetches interest rate data for Canada.
- It's similar to `fetch_USA_csv_as_df` but uses `requests` to download a CSV file directly.
- It parses the CSV content and finds the row containing the specified columns (`columns_to_import`).
- It extracts the relevant data, converts the date column to "yyyy-mm" format, and filters for the desired date range.
- It groups the data by month and keeps the last entry for each month to ensure unique values.
- Finally, it returns the cleaned DataFrame containing Canada interest rates.

Main Script

The main part of the python script generates a date range, fetches data from all sources, merges the data into a single DataFrame, and exports it as a CSV file to my local downloads folder.

Coming to our question-1: Line chart I have created in Googlesheet.



General Trends Analysis

- **Stability**: After initial fluctuations, most countries' interest rates stabilize around mid-2023 and remain constant through April 2024.
- **Relative Rates**: The USA maintains the highest interest rates throughout the period, while Australia has the lowest rates initially, which then become more stable and comparable to other countries by mid-2023.

Coming to our question-2 :The Industry Standard is Base Rate (Reserve/Central Bank Interest Rate) minus 0.5%.

Create a bar graph that shows the difference between Uber's Cash Pool Yields and the Industry Standard Rates and order them from greatest variance (negative) to least variance (positive). Some yields may be better than the Industry Standard rates.

Solution:

I have taken base rate from google using Central bank/Federal bank rates for respective countries.

Country/Continent	Currency	Uber's Cash Pool Yield	Base Rate	Industry standard	Variance
United Kingdom/GBP	GBP	3.50%	5.25%	4.75%	-1.25%
United States	USD	4%	5.38%	4.88%	-1%
Australia	AUD	3.50%	4.35%	3.85%	-0.35%
Canada	CAD	4.50%	5%	4.50%	0.00%
Europe	Euro	4.50%	4.50%	4.00%	0.50%

The industry standard serves as a reference point for evaluating the performance of a company's cash pool yield.

It represents the expected return on cash investments based on prevailing market conditions, typically set at 0.5% lower than the base rate set by the central bank of a country.

Comparison Points:

- A cash pool yield **higher** than the industry standard indicates efficient management of excess cash. The company is able to generate a return that surpasses the general market expectation for low-risk cash investments.
- Conversely, a cash pool yield **lower** than the industry standard suggests there might be room for improvement. The company might be able to find alternative investment options within the risk tolerance that offer a better return than the current strategy.

Analysis:

1. Across the countries, Uber's cash pool yields show varying degrees of performance compared to the industry standard.
2. Europe and Canada (4.50% yield) are performing at the industry standard (4.50% and 4.00% respectively).
3. The United Kingdom (3.50% yield), United States (4.00% yield), and Australia (3.50% yield) are all underperforming the industry standard by varying degrees.

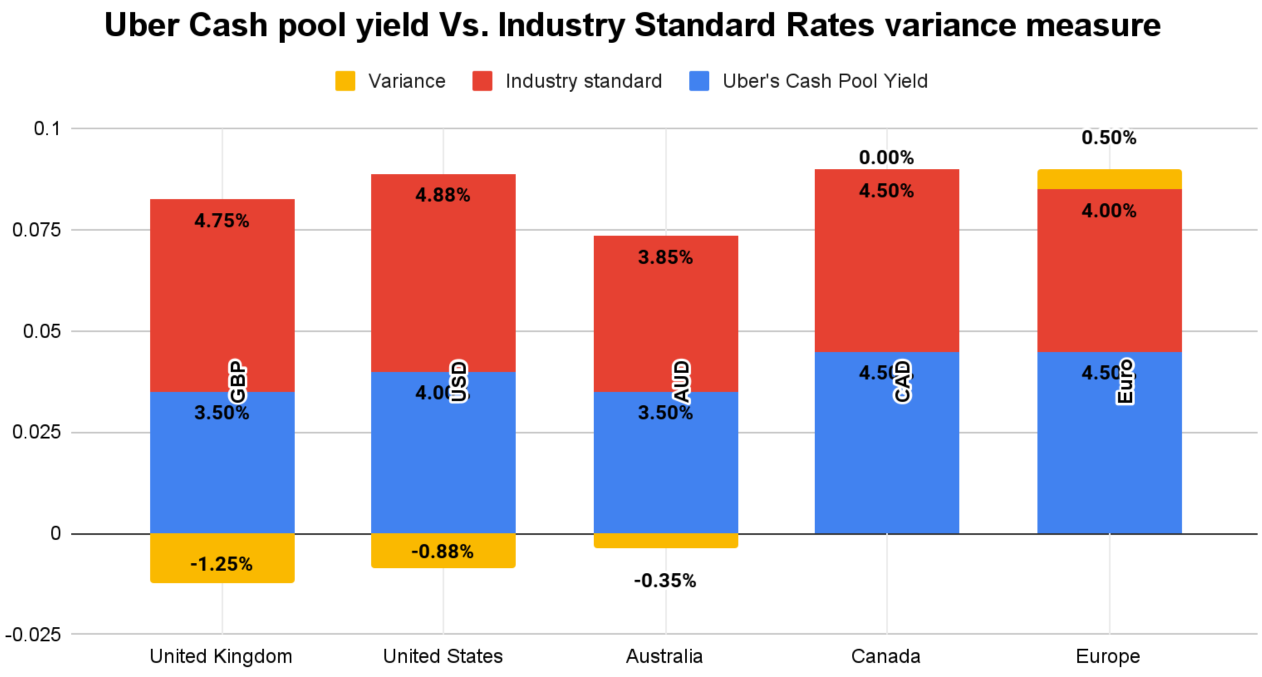
Possible Next Steps:

- The countries with yields below the standard (UK, US, Australia) might consider:

a)Reviewing their cash pool investment strategies to identify opportunities for higher returns.

b)Comparing their options with other companies in their industry for benchmarking purposes.

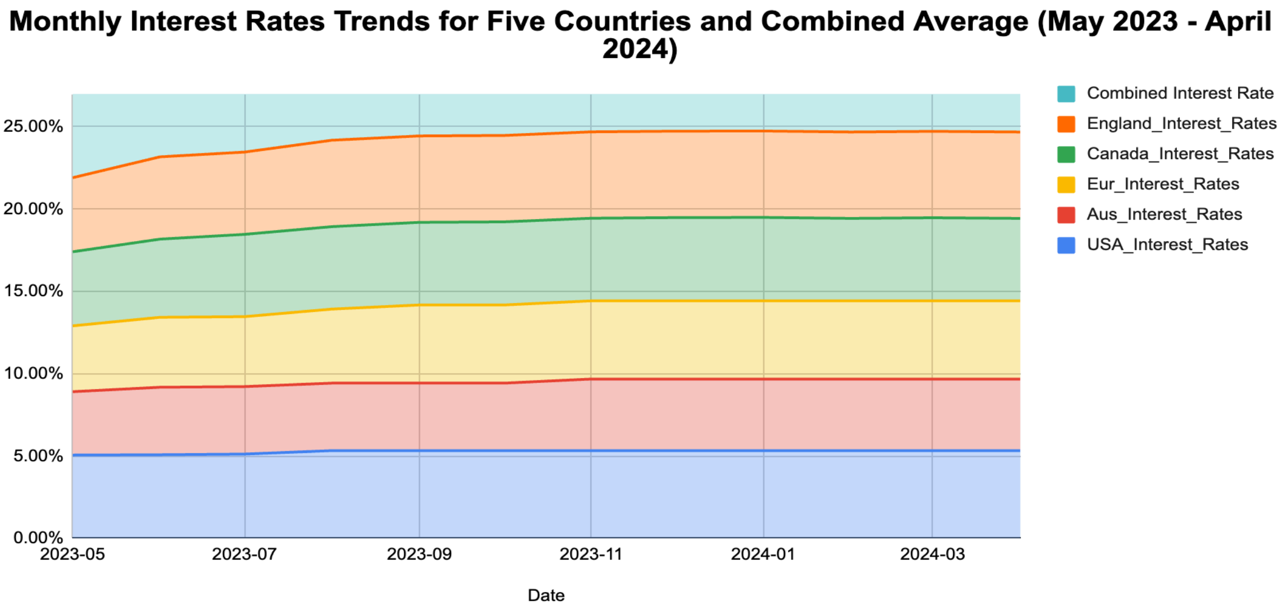
- Canada and Europe can continue monitoring their strategies and market conditions to ensure they maintain their current performance.



Q3:
Create a data visualization of your choice to represent a trend analysis of the prior 12 months (May 2023 to April 2024) that provide key insights.

For this part of the question, I have created a new column in data "Combined Interest rate",.The combined interest rate, calculated as the sum of individual countries' rates, provides a **broader perspective** on the overall borrowing environment. This would help us understand Uber's overall borrowing costs. However, a deeper analysis considering risk and individual loan terms is crucial for informed financial decision-making.

Below is the Stacked Area Chart:


Monthly Interest Rates Trends for Five Countries and Combined Average (May 2023 - April 2024)

Individual Trends Analysis:

- **United States (Blue)**: The USA interest rates remain relatively stable and lower compared to other countries throughout the period.
- **Australia (Red)**: Australia shows a slight increase in interest rates over time, positioned above the USA.
- **Europe (Yellow)**: European interest rates are slightly higher than those of Australia and show a gradual upward trend.
- **Canada (Green)**: Canada's interest rates are higher than Europe's and follow a similar upward trend.
- **England (Orange)**: England's interest rates are the highest among the five countries and contribute significantly to the combined interest rate.

The combined interest rate is significantly influenced by the higher rates from England and Canada.The overall trend indicates a steady rise in interest rates across all countries, suggesting tightening monetary policies during this period.

Further Improvements which can be done are as below:

1. Implement more advanced data cleaning techniques to handle missing values, outliers, and inconsistent data formats.

2. Dynamic Date Range Generation

   As of now in my code, I have hardcoded the dates from May-2023 to April-2024, but we can create a function to generate dates dynamically.

3. Extend the script to support more countries and their central banks.
4. Use a configuration file (e.g., JSON, YAML) to manage URLs and parsing rules for different data sources.
5. Database Integration: Save the aggregated data to a database (e.g., SQLite, PostgreSQL) for better data management and querying capabilities.Implement functions to retrieve historical data from the database for comparative analysis.