DATA534: Project Proposal

Karthik Chandran

20th Jan

My Teammates and I analysed the problem statement and came up with the Objective of the project. We delegated tasks between ourself to find existing API's and technologies to use for our project.

Some of the API's we experimented on are as follows

- NOAA Climate Data Online

- WeatherAPI

- World Bank Data API

- OECD Data API

- NewsAPI

The API's we finalized on are listed below

- Open-Meteo API

- FRED API

- GDELT API

22th Jan

Worked on our PPT for Presentation, showcasing a sample final time series dataframe and some sample plots to help the audience better understand our objective.

26th Jan

Extended on my Teammate ilakiya's work on Merging the all API data by Data Column. Tested and Error handled the following functions -

- get_daily_weather()

- get_daily_economic_data()

- gdelt_timeline_daily()

- get_macroeconomic_data()

31st Jan

Created and pushed two additional function to aurora.R file called **revenue_merge()** and **plotter**.

The `revenue_merge()` function gets the dataframe returned by get_macroeconomic_data and merge it with revenue data that the user want to compare with. The required parameters are input_data and csv_path. The merge_revenue function supports the .csv file format to get revenue data and returns a dataframe that has been merged by the date column.

The `plotter()` function is used to plot the macro date and the revenue data to create timeseries plot using ggplot2 to help visualize the data and analyze correlations. The plotter function has the following parameters - **df** (The Dataframe returned by revenue_merge()) - **cols** (The Macro Columns you want to explicitly plot) - **date_col** (The name of the Date column in the dataframe) - **revenue_col** (The name of the Revenue column in the dataframe) - **ncol** (The no of the column plots to fit in the final grid) - **save_path** (The path to store the plot in) - **width** (Plot width) - **height** (Plot Height) - **dpi** (Plot DPI) - **scale_method** (Scaling Method used to plot Macro and Revenue - `zscore`, `minmax`) - **drop_na** (if True drops all the NA values from the Dateframe)

1st Jan

- Tested and Error Handled for the functions **revenue_merge()** and **plotter** (R\aurora.R\).
- Performed Unit Testing and Integration Testing in R for all 6 functions (tests\testthat\).
- Updated tester.R to call the newly created functions (Scripts\).
- Created github secret tokens to initialize API Keys as Environmental Variables.
- Created .yaml files for Continuous Integration through Github Actions. (.github\workflows\).

2nd Jan

- Resolved Github Conflicts and Errors in Continuous Integration.
- Wrapped up final changes and updates for package deployment.

Commit Links

- Project Proposal [Link]

- Added Revenue and Plotter Functions [Link]

- Unit and Integration Testing [Link]

- Continuous Integration [Link]

- Testing Github Actions [Link]