Econ 409 Project Outline

Group Details

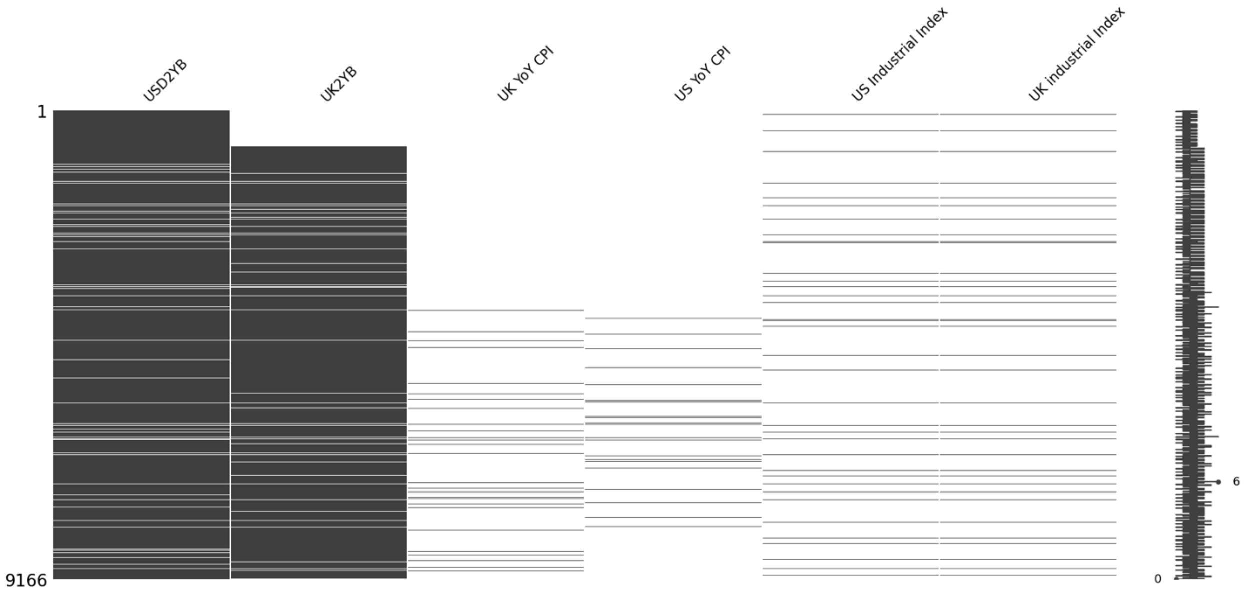
- **Group Name:** Data Mavericks
- **Group Members:**
  - Luis Jose Zapata Bobadilla
  - Oishika Kar
  - Shudan Deng
  - Sultan Ali Rashid Al Balushi

Project Overview

The project aims to develop and evaluate a trading strategy based on economic indicators and exchange rate predictions. Through rigorous data analysis, cleaning, transformation, and statistical modeling, we strive to implement a strategy that outperforms standard benchmarks.

1. Data Import and Preliminary Analysis

- **Tools and Libraries Used:** Utilization of libraries such as **yfinance**, **pandas**, **NumPy**, **datetime**, **io**, **statsmodels.formula.api**, and **missingno** for financial data analysis, manipulation, statistical modeling, and visualization.
- **Data Import:** A CSV file, **final_project_data.csv**, is imported into a Data frame with an emphasis on displaying the last 15 rows to understand recent data points.
- **Visualization of Missing Data:** The **missingno** library is employed to visually assess missing values within our dataset, enabling us to identify patterns of data incompleteness. We identified some missing values in columns containing monthly released data.



2. Data Cleaning and Preparation

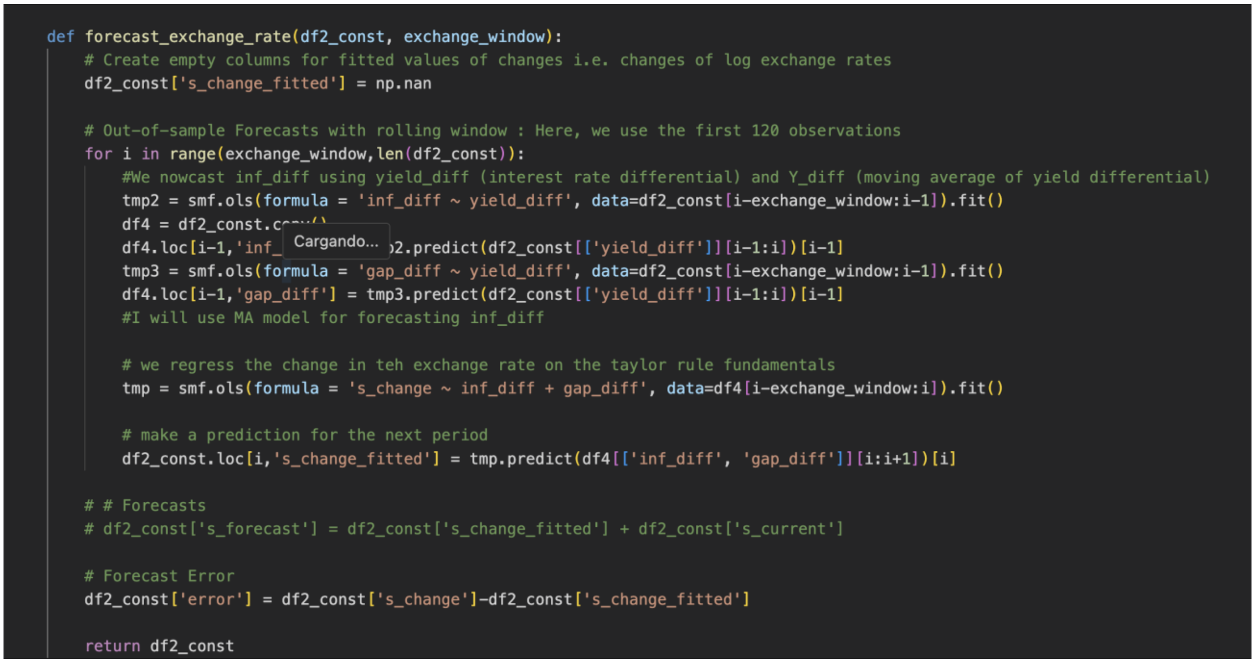
- **Initial Date Identification:** Through analysis of the 'UK YoY CPI' column, we established an initial date for our analysis, ensuring our dataset's temporal relevance to our research objectives.
- **Data interpretation:** We detected a look ahead possible problem: The release dates of CPI and Industrial Index are approximately two weeks after the end of the month. Additionally, we discovered that the 2 years yield government notes have a coupon that it executed every 6 months.
- **Historical Exchange Rates:** We acquired historical USD to GBP exchange rates from Yahoo Finance, starting from our established initial date, to support our currency exchange rate analysis.

3. Data Transformation and Feature Engineering

- **Dataframe Slicing:** To maintain data integrity, we excluded rows with missing values prior to our initial date, ensuring a clean dataset for analysis.
- **Financial Metrics Transformation:** Key financial metrics such as 'USD2YB' and 'UK2YB' are transformed into a daily frequency and subsequently into log returns to normalize their distribution for time series analysis. The CPI and Industrial Indexes are resampled maintaining their last value and shifting their values 1 month.

4. Statistical Analysis and Modeling

- **Exchange Rate Modeling:** Log returns of exchange rates are calculated, and a Taylor Rule model is utilized among other financial models to predict exchange rate changes based on economic indicators.
- **Taylor Rule Model:** We decided to do a Taylor Rule Model, where we use the difference of Inflation and GDP gap to forecast the exchange rate monthly variation. We nowcasted inflation and GDP Gap using the monthly interest rate differential of US and UK.



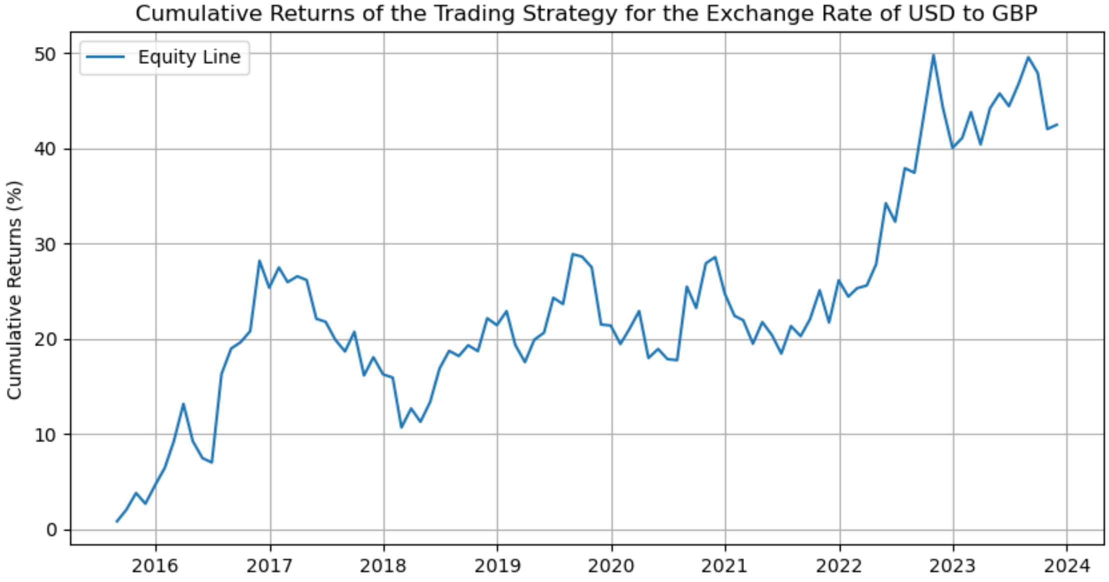
```
def forecast_exchange_rate(df2_const, exchange_window):
    # Create empty columns for fitted values of changes i.e. changes of log exchange rates
    df2_const['s_change_fitted'] = np.nan

    # Out-of-sample Forecasts with rolling window : Here, we use the first 120 observations
    for i in range(exchange_window,len(df2_const)):
        #We nowcast inf_diff using yield_diff (interest rate differential) and Y_diff (moving average of yield differential)
        tmp2 = smf.ols(formula = 'inf_diff ~ yield_diff', data=df2_const[i-exchange_window:i-1]).fit()
        df4 = df2_const.copy()
        df4.loc[i-1,'inf_ Cargando... 2.predict(df2_const[['yield_diff']][i-1:i])[i-1]
        tmp3 = smf.ols(formula = 'gap_diff ~ yield_diff', data=df2_const[i-exchange_window:i-1]).fit()
        df4.loc[i-1,'gap_diff'] = tmp3.predict(df2_const[['yield_diff']][i-1:i])[i-1]
        #I will use MA model for forecasting inf_diff

        # we regress the change in teh exchange rate on the taylor rule fundamentals
        tmp = smf.ols(formula = 's_change ~ inf_diff + gap_diff', data=df4[i-exchange_window:i]).fit()

        # make a prediction for the next period
        df2_const.loc[i,'s_change_fitted'] = tmp.predict(df4[['inf_diff', 'gap_diff']][i:i+1])[i]

    # # Forecasts
    # df2_const['s_forecast'] = df2_const['s_change_fitted'] + df2_const['s_current']

    # Forecast Error
    df2_const['error'] = df2_const['s_change']-df2_const['s_change_fitted']

    return df2_const
```

5. Strategy Implementation and Performance Evaluation

- **Trading Strategy Execution:** Our strategy involves taking positions based on the predicted direction of exchange rate changes, with cumulative returns calculated and visualized.
- **Performance Comparison:** The strategy's annualized return is benchmarked against a "random walk" model to quantify its effectiveness and potential for profit.

6. Visualization and Conclusion

- **Cumulative Returns Visualization:** Plots are generated to visually represent the cumulative returns of our trading strategy over time, offering insights into its performance trajectory.



- **Annualized Return Calculation:** A final computation of the strategy's annualized return provides a quantitative measure of its overall success, obtaining approximately 4 percent of annual return with 0.9 Sharpe Ratio.

Key Hyperparameters and Trading Strategy Details

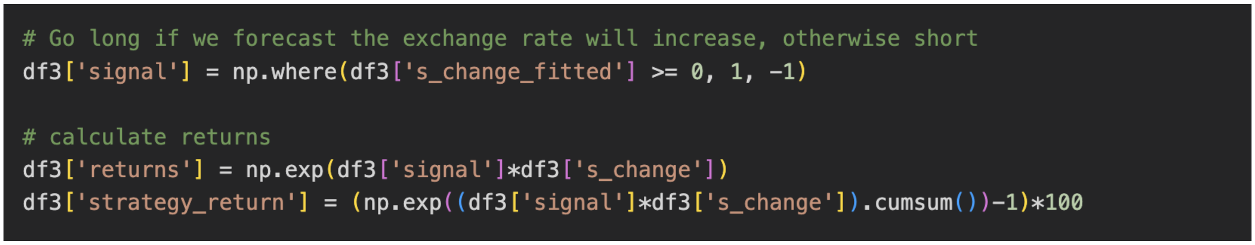
Hyperparameters

- **Rolling Window Size for Model Fitting:** A rolling window of size 120 is utilized for fitting our Taylor Rule-based model (as suggested by Papell paper 2003), balancing the incorporation of historical trends with responsiveness to recent data.
- **Initial Windows for GDP Output Gap:** We used a window of 26 months to derive the potential GDP and subsequently the Output Gap.

Trading Strategy

- **Signal Generation:** Our trading strategy generates signals based on the forecasted changes in exchange rates (`s_change_fitted`). If the forecast suggests an increase in exchange rates, we generate a long (buy) signal (assigned a value of 1), and if the forecast suggests a decrease, we generate a short (sell) signal (assigned a value of -1).

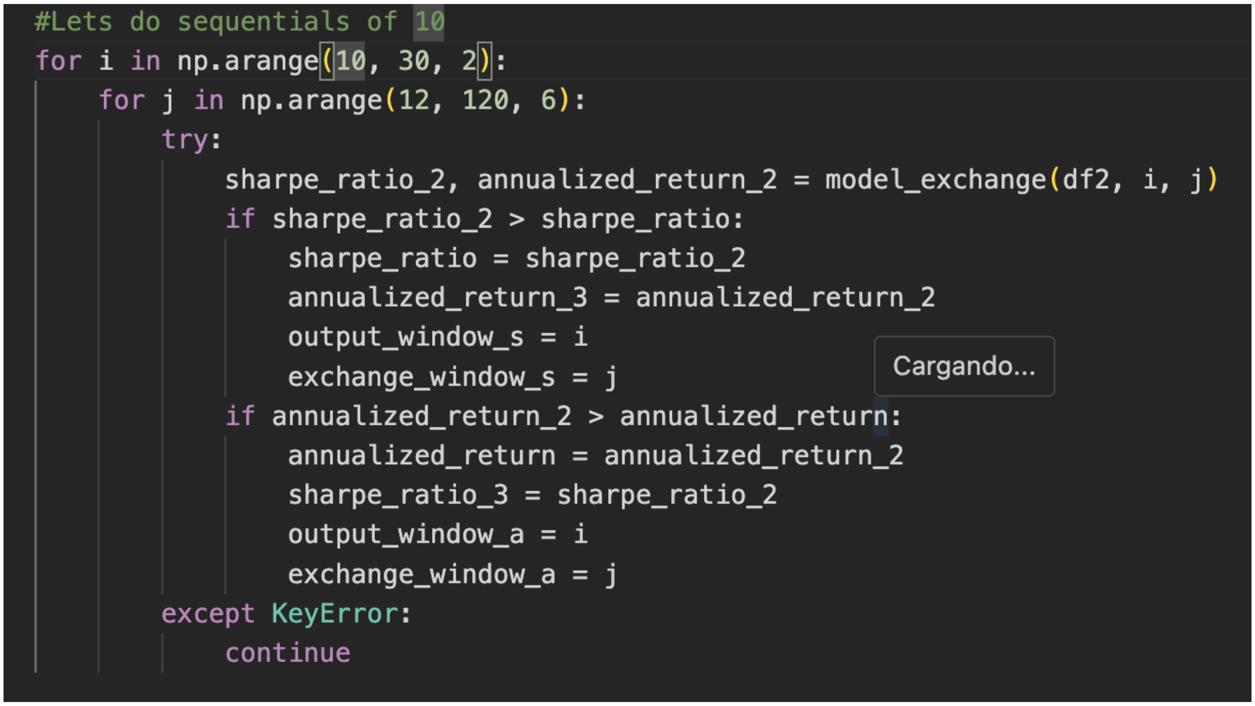
- - Long (Buy) Signal: We take a long position when our model forecasts a positive change in exchange rates (`s_change_fitted >= 0`).
- - Short (Sell) Signal: We take a short position when our model forecasts a negative change in exchange rates (`s_change_fitted < 0`).



```python
# Go long if we forecast the exchange rate will increase, otherwise short
df3['signal'] = np.where(df3['s_change_fitted'] >= 0, 1, -1)

# calculate returns
df3['returns'] = np.exp(df3['signal']*df3['s_change'])
df3['strategy_return'] = (np.exp((df3['signal']*df3['s_change']).cumsum())-1)*100
```

- **Position Entry/Exit:** Our strategy enters a long position (buys) when the forecasted change in exchange rates is positive and enters a short position (sells) when the forecasted change is negative. We exit our position (close the trade) at the end of each forecasting period.

- **Back-testing and Validation:** Our strategy undergoes rigorous back-testing against out-of-sample data to validate its effectiveness and adaptability. We try different values of the output windows and forecast windows to maximize annualized return and Sharpe Ratio. Our algorithm confirms that 26 as the output windows and 120 as the forecast windows are the best trade for this model.



```python
#Lets do sequentials of 10
for i in np.arange(10, 30, 2):
    for j in np.arange(12, 120, 6):
        try:
            sharpe_ratio_2, annualized_return_2 = model_exchange(df2, i, j)
            if sharpe_ratio_2 > sharpe_ratio:
                sharpe_ratio = sharpe_ratio_2
                annualized_return_3 = annualized_return_2
                output_window_s = i
                exchange_window_s = j
            if annualized_return_2 > annualized_return:
                annualized_return = annualized_return_2
                sharpe_ratio_3 = sharpe_ratio_2
                output_window_a = i
                exchange_window_a = j
        except KeyError:
            continue
```

This outline provides a structured overview of our project's methodology, from data import and preparation through to strategy implementation and evaluation. By adhering to this comprehensive approach, we aim to develop a robust trading strategy informed by detailed economic analysis and statistical modeling.