

# Navigating the Waves: Analyzing Currency Exchange Rate Fluctuations

## Introduction

The **exchange rate** is a crucial economic indicator that reflects the value of one currency relative to another. It plays a vital role in international trade, investment decisions, and economic policy formulation. Analysis focuses on the **USD (US Dollar)** as the base currency and examines its exchange rates against currencies: **USD/EUR (Euro)**, **USD/GBP (British Pound)**, **USD/AUD (Australian Dollar)**, **USD/NZD (New Zealand Dollar)**, **USD/CAD (Canadian Dollar)**, **USD/SGD (Singapore Dollar)**, and **USD/CHF (Swiss Franc)**.

I will be analyzing the aforementioned currency pairs over the period from January 1, 2015, to November 17, 2024, to gain insights into their performance against the USD. This visual analysis will explore the following questions:

- What trends can be observed in the USD exchange rates against the various currencies during the period from January 2015 to November 2024?
- How do these currency pairs correlate with one another and with major events, such as COVID-19 and Brexit?
- What factors contributed to the decline in GBP prices during this period, and what were the underlying reasons for this downward trend?
- Assessing Overnight Currency Fluctuations.

```
In [1]: import yfinance as yf
import numpy as np
import pandas as pd
import seaborn as sns
import warnings
import matplotlib.pyplot as plt
import matplotlib as mlp
import datetime
#!pip install seaborn
warnings.filterwarnings("ignore")

#install the yahoo financial data API
#!pip install yfinance
```

```
In [2]: FILE_PATH = 'ccy_price.csv'
ccy_pair = ['USD/EUR', 'USD/GBP', 'USD/AUD', 'USD/NZD', 'USD/CAD', 'USD/SGD', 'USD/CHF']
```

```
In [10]: def download_data(ccy, start, end , cols = ['date' , 'ccy' , 'open' , 'close']):

    #get ric form ccy pair for query
    ric = ccy.split('/')[1] + "=X"

    # Fetch historical data
    data = yf.download(ric, start = start, end = end)

    #get date from index type timestamp
    data['date'] = data.index.date

    #remove the index
    data = data.reset_index(drop=True)

    # update column name
    data.columns = ['adj_close', 'close', 'high', 'low', 'open', 'volume' , 'date']

    # update ccy pair
    data['ccy'] = ccy

    return data[cols]

def load_and_store_data(start, end , file_path):

    #store the currency data to list
    price = []

    #download data for all the currency pair
    for ccy in ccy_pair:
        price.append(download_data(ccy, start, end))

    # concat the data for all the currency
    df = pd.concat(price)

    #store for analysis so we need not to connect API
    df.to_csv(file_path)

    return df.head()

def load_fx_data(file_path , col = ['date', 'ccy' , 'open' , 'close' , 'week' ]):

    #read csv file
    data = pd.read_csv(file_path)

    #populate date
    data['date'] = pd.to_datetime(data['date'])

    #sort data by date
    data = data.sort_values(by = ['date'])

    #update to quarter for future analysis
    data['week'] = data['date'].dt.to_period('W')

    return data[col]
```

```
In [11]: #API call to download Fx data
# Only need to run if we want to refresh the data
# Proxy environment variable setup will be need if outward connection is blocked. http_proxy = http://user:pass@host:port
```

```
#----- remove the comented line after proxy setup -----  
#load_and_store_data('2015-01-01', '2024-11-17', FILE_PATH)
```

**Figure 1** Visualizing Exchange Rate Dynamics: A Line Plot of USD and Selected Currencies (Jan 2015 - Nov 2024)

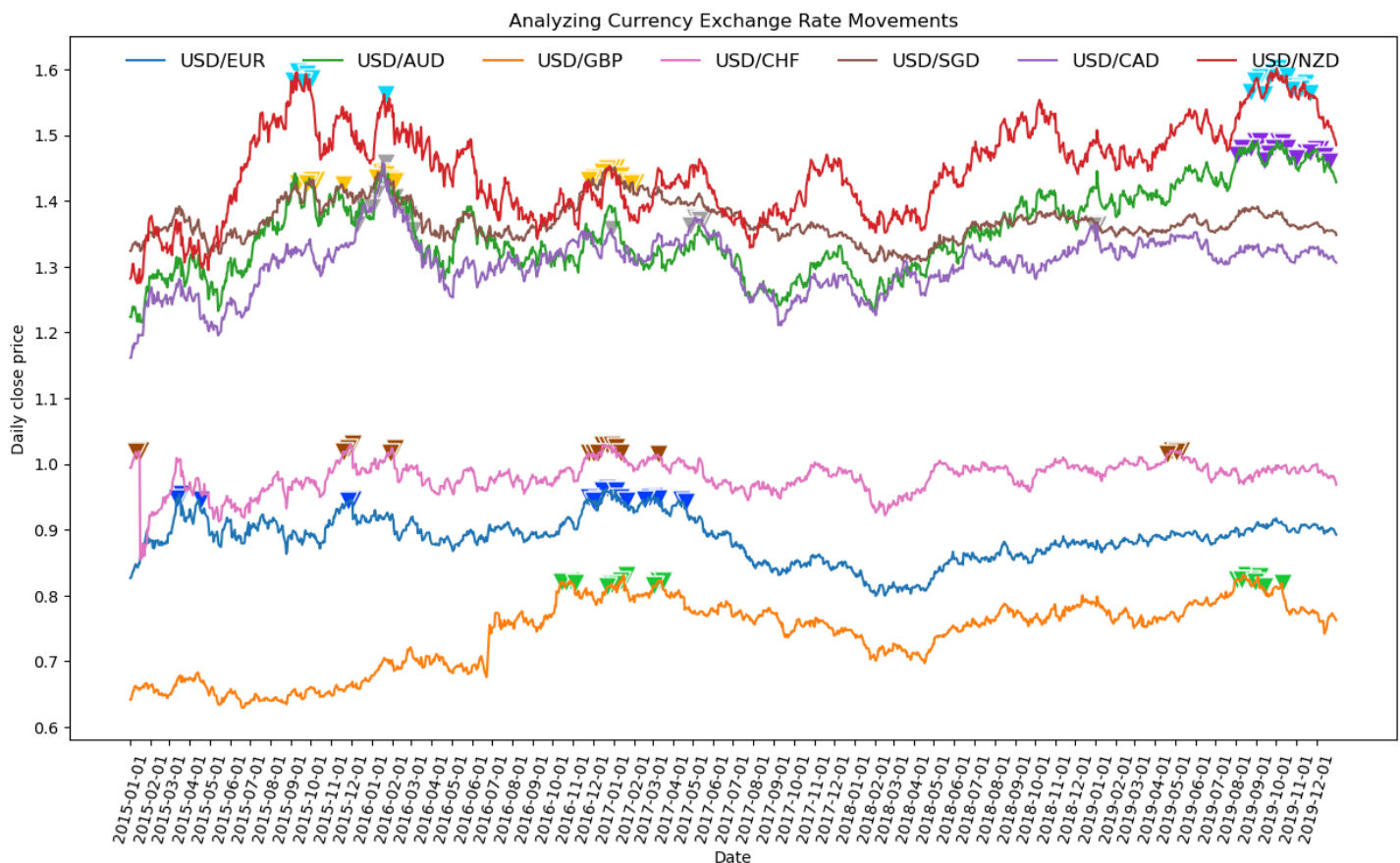
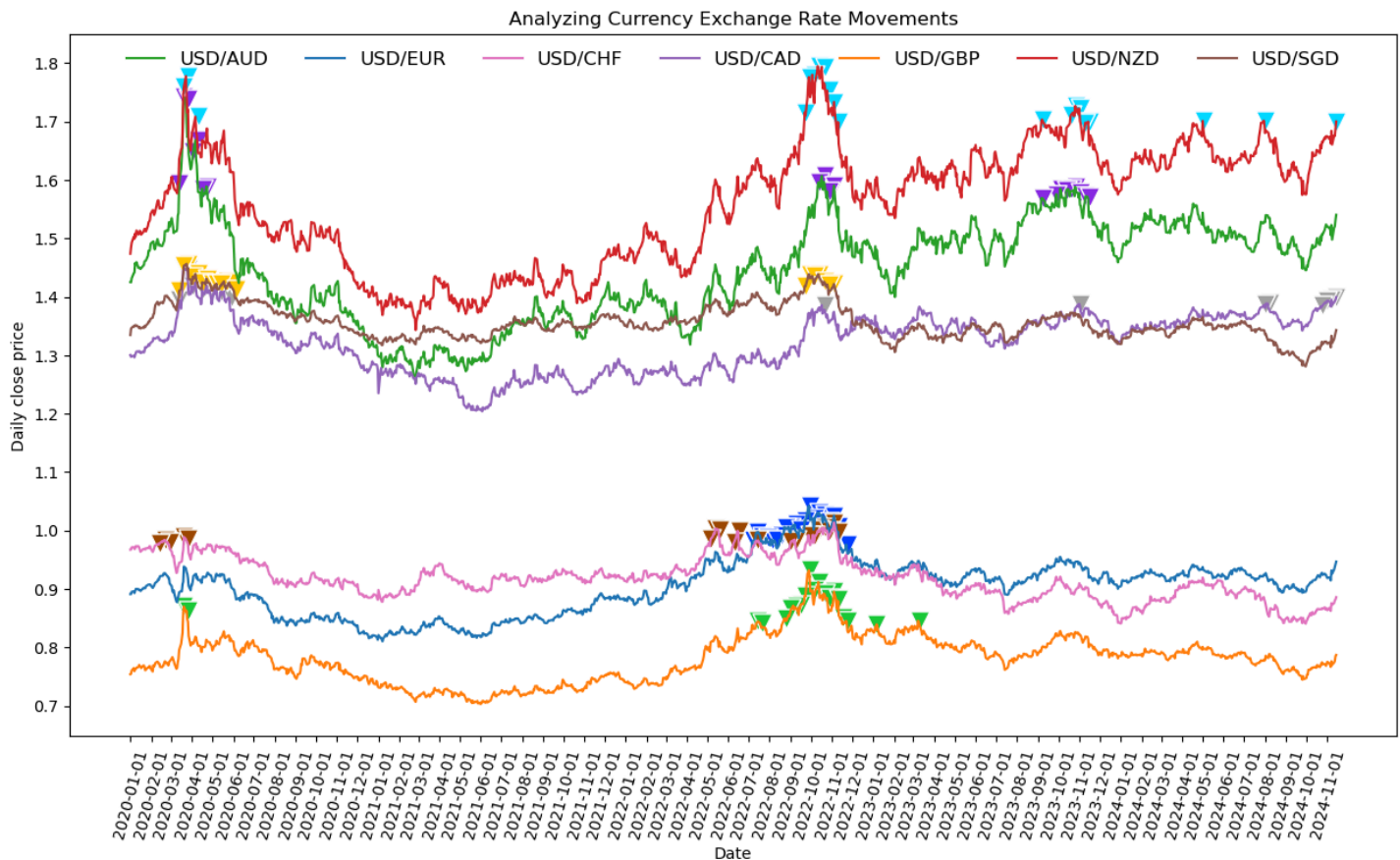
- Line plots provide a clear visual representation change over time. This allows for easy identification of trends, including upward or downward movements. In addition to the line plot, we will also employ scatter plots in conjunction with it to highlight the top 20 sharp dips in the exchange rate.
- The continuous nature of line graphs makes it simple to see periods of chage and stability in data over time.
- By plotting different currency pairs together, we can visually assess any correlations or relationships between them. If one currency is moving in tandem with the USD, it's easier to spot these relationships in a line graph.

I have divided the graph into two plots to enhance the clarity of visualization

- Plot 1 - Covers the period from 2020 to 2024
- Plot 2 - Covers the period from 2015 - 2020

```
In [12]: #seaborn color configuration  
#plot configs  
hex_color_code = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2']  
dark_color_code = [ '#023eff', '#1ac938', '#8b2be2', '#00d7ff', '#a3a3a3', '#ffc400', '#9f4800']  
  
#define palette so that same color are used accross graph  
palette = dict(zip(ccy_pair , hex_color_code))  
palette_mark = dict(zip(ccy_pair , dark_color_code))  
  
def prepare_plot(data):  
  
    # get tick size to plot  
    xticks = [str(i) for i in data['date'].dt.to_period('M').unique()]  
    yticks = np.linspace(.6, 1.9 , 14)  
  
    #set plot size  
    plt.subplots(figsize= (15,8))  
  
    sns.lineplot(data = data, x = 'date',  
                y= 'close', hue = 'ccy',  
                palette = palette ).set(xticks = xticks , yticks = yticks,  
                                       title = 'Analyzing Currency Exchange Rate Movements',  
                                       xlabel = 'Date' , ylabel = 'Daily close price' )  
  
    # add marker for currency Movement  
    for ccy in ccy_pair:  
        marker_data = data[data['ccy'] == ccy ].groupby('week' , as_index=False).agg('max').sort_values(by=['close'],  
                                                                                               ascending=False).iloc[0:20]  
        sns.scatterplot(data = marker_data , x = 'date' , y = 'close' ,marker = 'v' , s = 150 ,color = palette_mark[ccy] )  
  
    # set plot attribute  
    plt.xticks(rotation=75)  
    plt.legend( loc = "upper center",  
              fontsize='12', bbox_to_anchor=(.5, 1) ,  
              ncol = 7 , title = None , frameon = False )  
  
    return plt
```

```
In [13]: data = load_fx_data(FILE_PATH)  
  
#plot data form 2020 till yesterday  
plt_2020 = prepare_plot(data = data[data['date'].between('2020.01.01' , '2024.11.14')].reset_index(drop=True))  
plt_2020.show()  
  
#plot data form 2015 till 2002  
plt_2015 = prepare_plot(data = data[data['date'].between('2015.01.01' , '2019.12.31')].reset_index(drop=True))  
plt_2015.show()
```



### Visualization analysis

- The visual trend indicates that USD/NZD and USD/AUD exhibit the highest levels of volatility among the currency pairs.
- After 2020, USD/CHF (Swiss Franc) appears to be the most stable currency, showing the least impact from exchange rate fluctuations.
- During the COVID-19 pandemic, all currencies experienced fluctuations in exchange rates. However, following a dip, both the Euro and Swiss Franc recovered quickly, emerging as strong currencies.

**Examining the British Pound: A Historical Analysis of GBP Volatility** - The sharp dips in the GBP are closely correlated with economic policy, the political environment, and various public policies. Some of the major events affecting its valuation include`:

- **Brexit Vote (2016)** :[1] Following the announcement of the Brexit referendum result, the British Pound fell by 28% in valuation. The visual trend indicates that it has not fully recovered since this significant decline.
- **Boris Johnson's taking office as PM (2019)** :[2] The GBP experienced another decline shortly after Boris Johnson assumed office as Prime Minister.

- **COVID-19 Pandemic** : Although the pound lost value during the pandemic, it quickly regained its losses and exhibited a strong upward momentum thereafter.
- **UK Government's Tax Cut Announcement (2022)** :[3] The GBP lost valuation following the announcement of tax cuts by the UK government. However, it recovered its value within a few months.

## Figure 2 Heat map visualization

- A heatmap is an effective visualization tool for analyzing currency exchange rates because it provides a quick and intuitive way to identify trends, performance, and volatility across multiple currency pairs.
- Heatmaps will allow us to easily distinguish between strong and weak currencies at a glance, facilitating rapid decision-making.

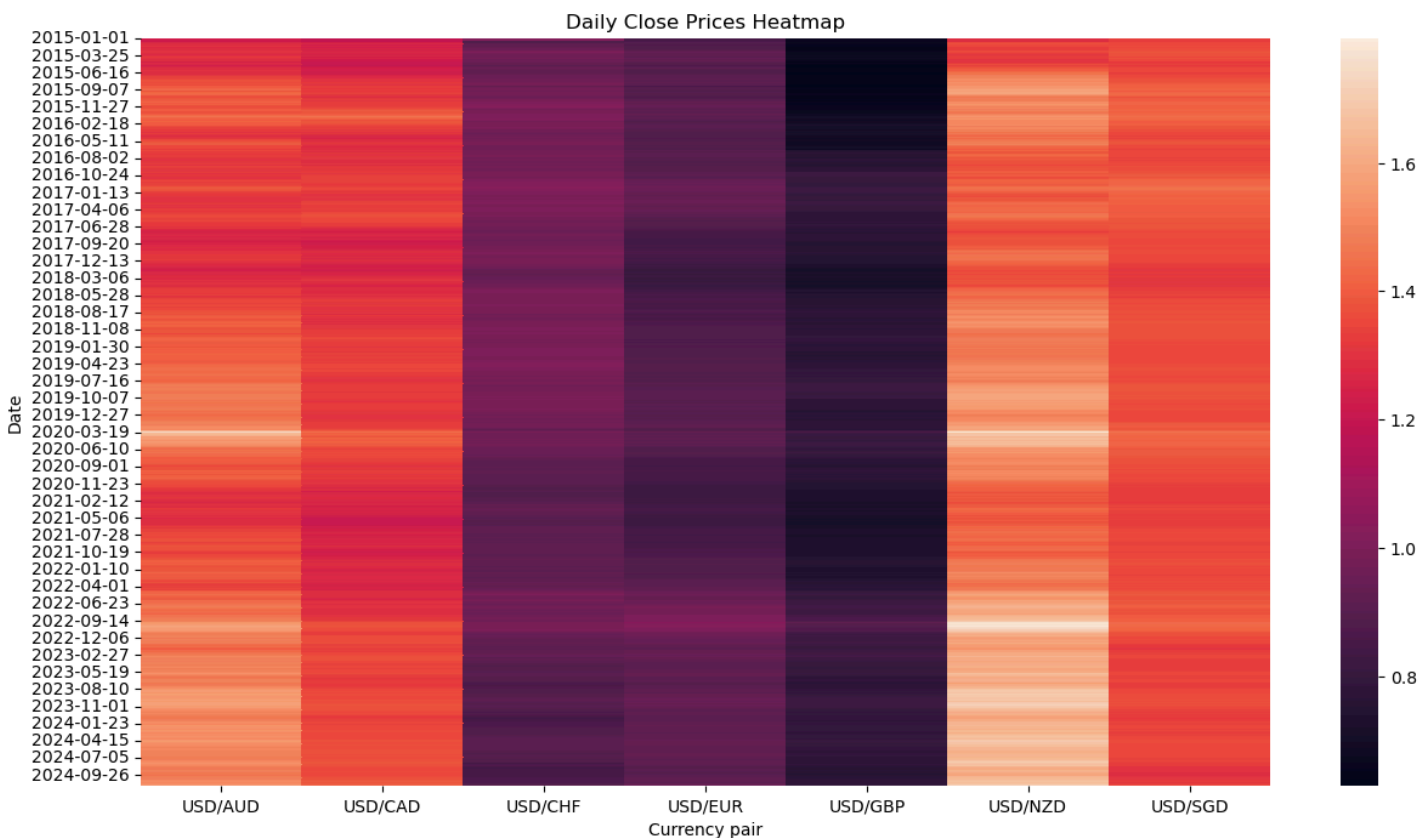
```
In [14]: def plot_head_map(data):

    data['date'] = data['date'].dt.date
    data = data.pivot(index = 'date' , columns = 'ccy' , values = 'close' )
    plt.subplots(figsize= (15,8))

    sns.heatmap(data).set(title = 'Daily Close Prices Heatmap' , xlabel = 'Currency pair' , ylabel = 'Date')

    plt.title("Daily Close Prices Heatmap")
    return plt

heat_map_data = load_fx_data(FILE_PATH)
plt_2015= plot_head_map(heat_map_data[heat_map_data['date'].between('2015.01.01' , '2024.11.17')])
plt_2015.show()
```



### Visual Analysis of Heatmap-

- The heatmap visually indicates that the British Pound (GBP) is the strongest currency among the analyzed pairs.
- The Swiss Franc (CHF) and Singapore Dollar (SGD) appear to be very stable currencies, showing consistent performance over the observed period.
- In contrast, the Australian Dollar (AUD) and New Zealand Dollar (NZD) exhibit high volatility, reflecting significant fluctuations in their exchange rates.

## Figure 3 Evaluating Overnight Movements

"Overnight Gain/Loss" is the change in the value of a currency exchange, calculated by subtracting day's closing price to the next day's opening price. It will help us understand how a currency performed outside of regular trading hours, influenced by events or market activities that occurred overnight.

```
In [15]: #plot overnight gain
def plot_gain(date):

    data = load_fx_data(FILE_PATH)
    data = data[data['date'] > date]

    xticks = [str(i) for i in data['date'].dt.to_period('M').unique()]

    for i in ccy_pair:

        #Filter the currency data
        ccy_data = data[data['ccy'] == i]

        #shift the close price to next day
        ccy_data['close_yes'] = ccy_data['close'].shift(1)

        #calculate the overnight again and filter out days where overnight gain id less then 2 cents
```

```

ccy_data = ccy_data[(ccy_data['close_yes'] - ccy_data['open']).abs() > 0.01]

#generate the y linspace
yticks = np.linspace(ccy_data['close_yes'].min(), ccy_data['close_yes'].max() , 40)

plt.subplots(figsize= (15,10))

#plot yesterday's close
sns.lineplot(data = ccy_data , x = 'date' , y='close_yes' , label = 'Yesterday close (T-1 Close price)' ).set(
    xticks = xticks , yticks = yticks , title = f'Overnight Gain/Loss data where movement is more than 2 cents {i}')

#plot open price
sns.lineplot(data = ccy_data , x = 'date' , y='open' , label = 'open').set(
    xticks = xticks , yticks = yticks )

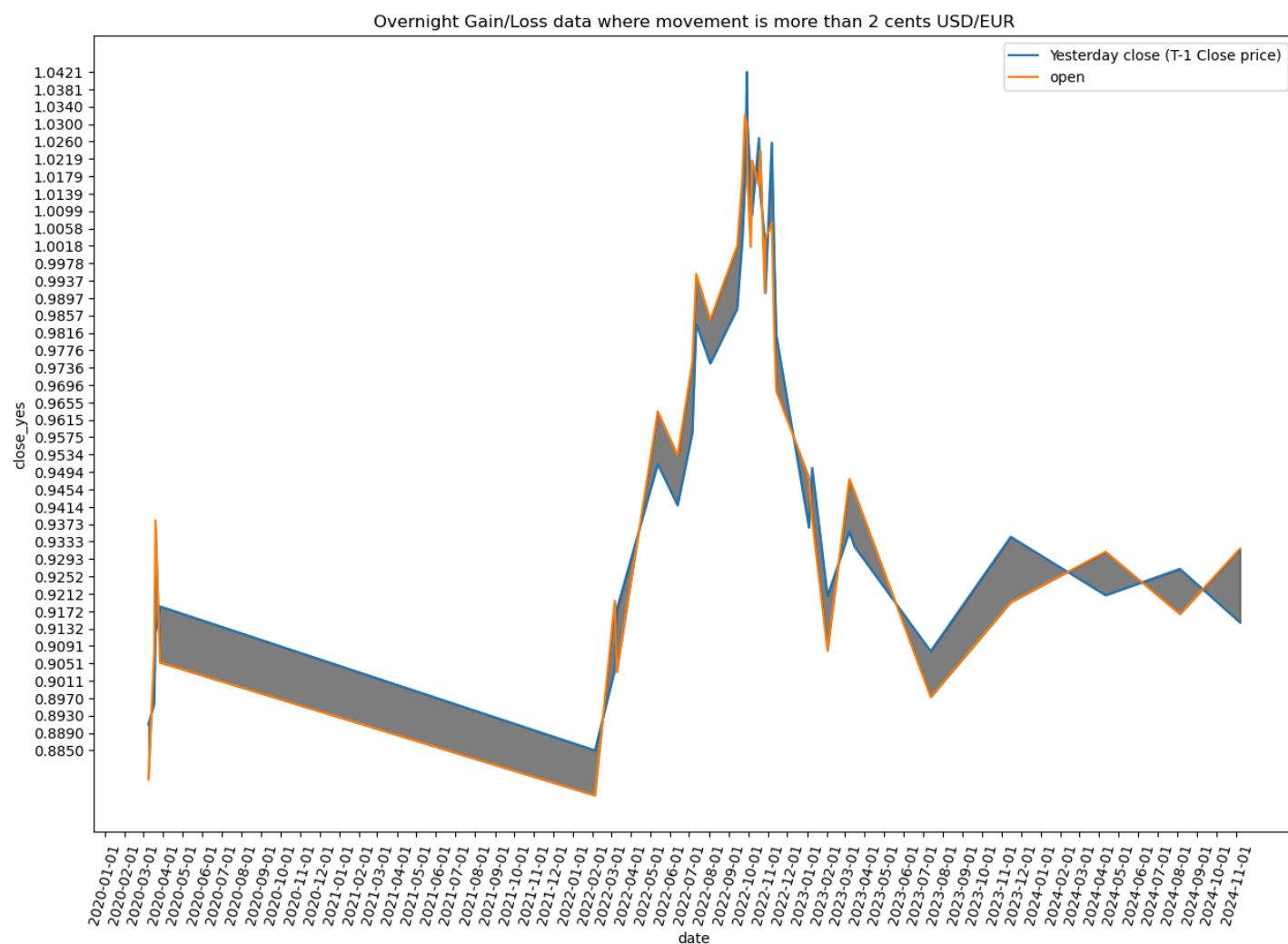
#update y
plt.xticks(rotation=75)
plt.legend()

# get the line details to fill the area between lines
line = plt.gca().lines
plt.fill_between(line[0].get_xdata(),line[0].get_ydata(), line[1].get_ydata(), color='black', alpha=.5)
plt.show()

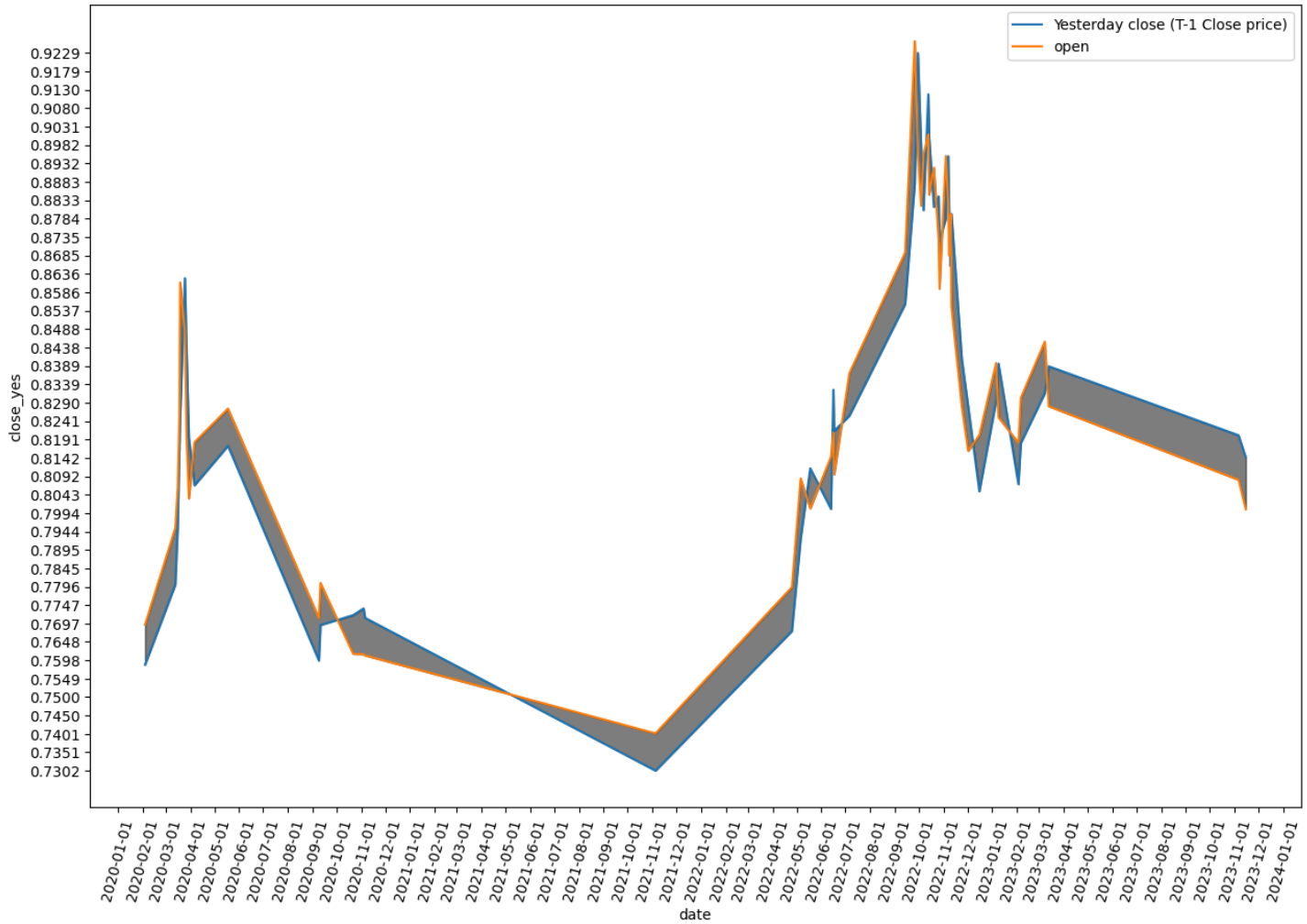
return None

plot_gain('2020-01-01')

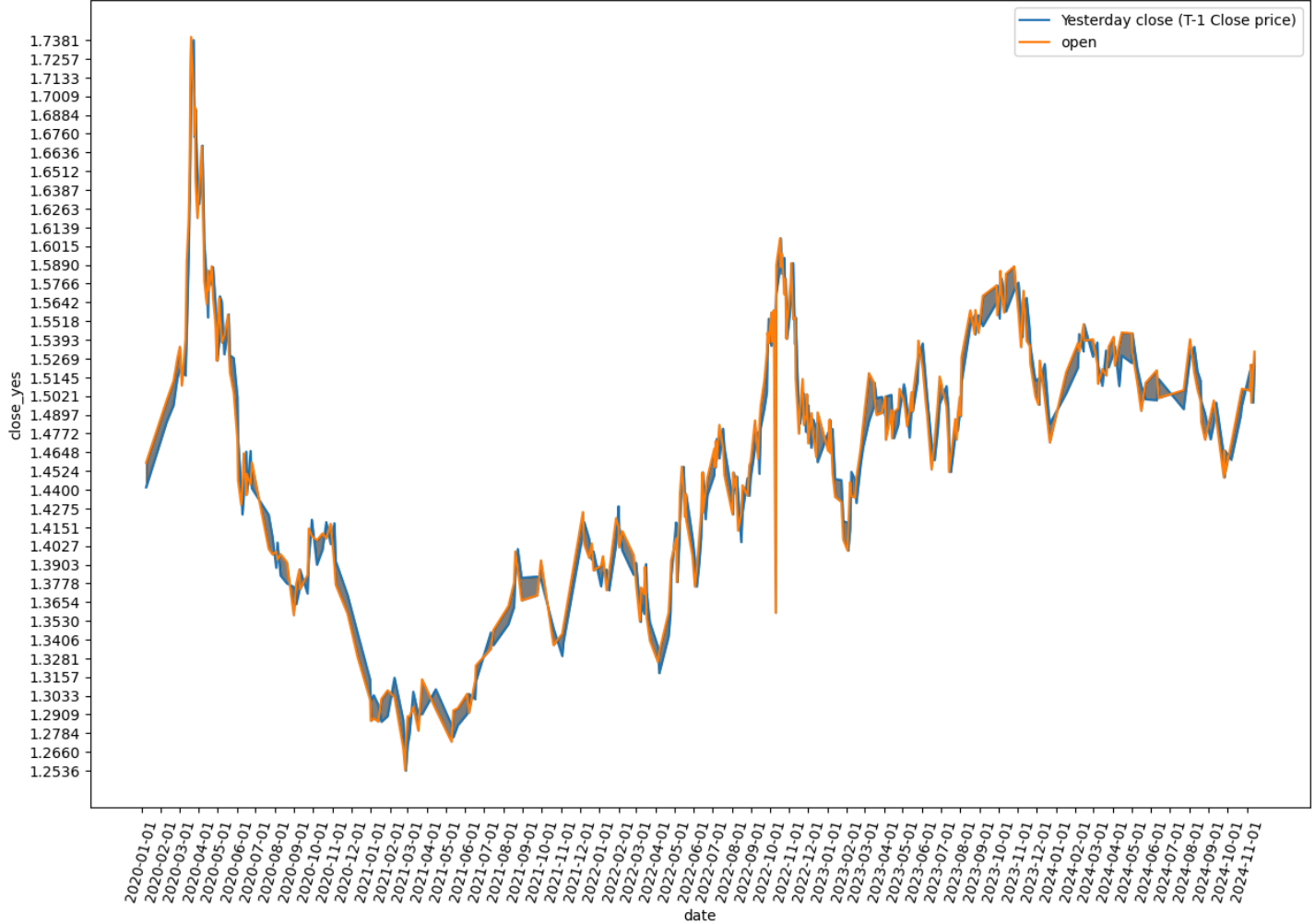
```



Overnight Gain/Loss data where movement is more than 2 cents USD/GBP

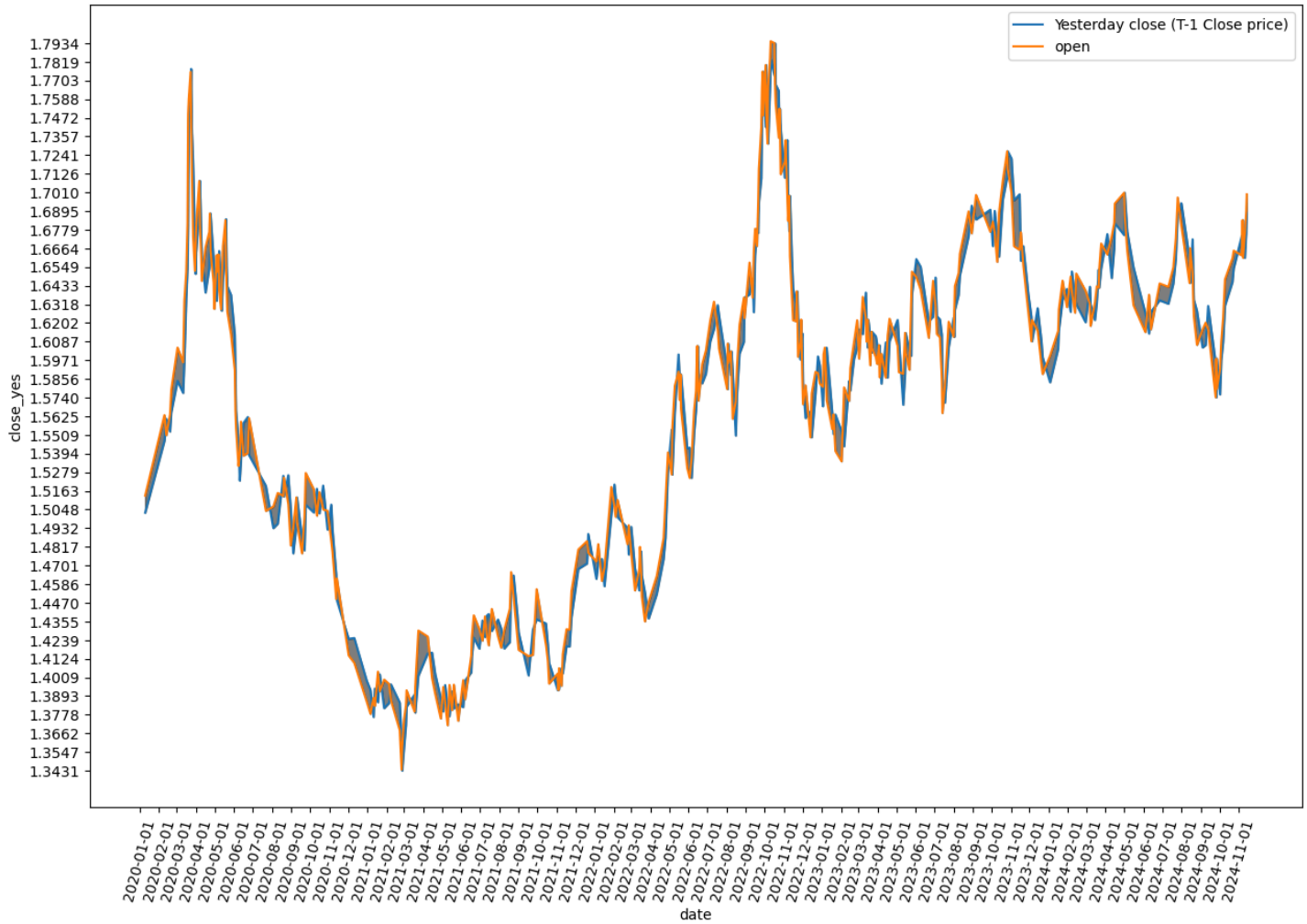


Overnight Gain/Loss data where movement is more than 2 cents USD/AUD

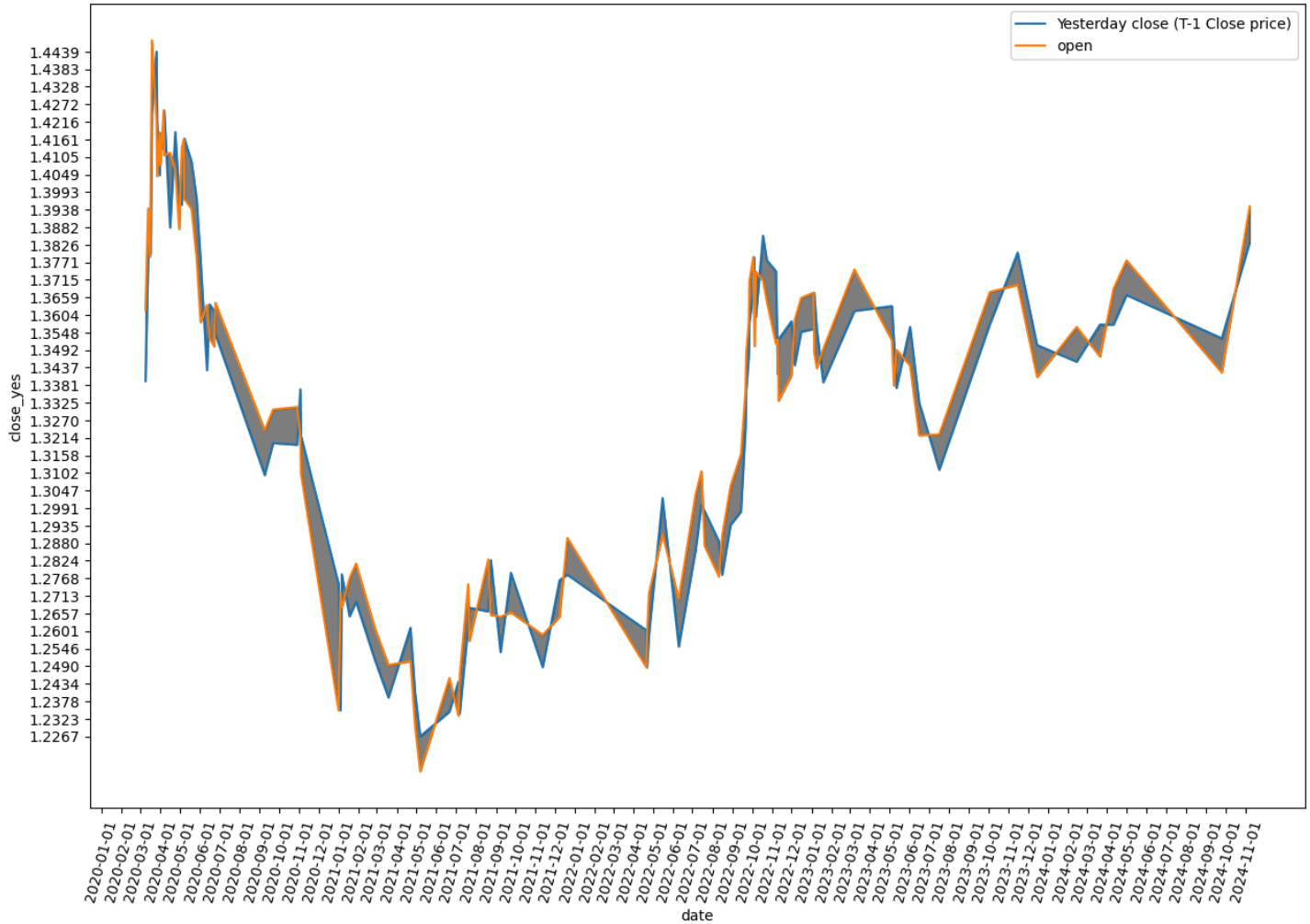




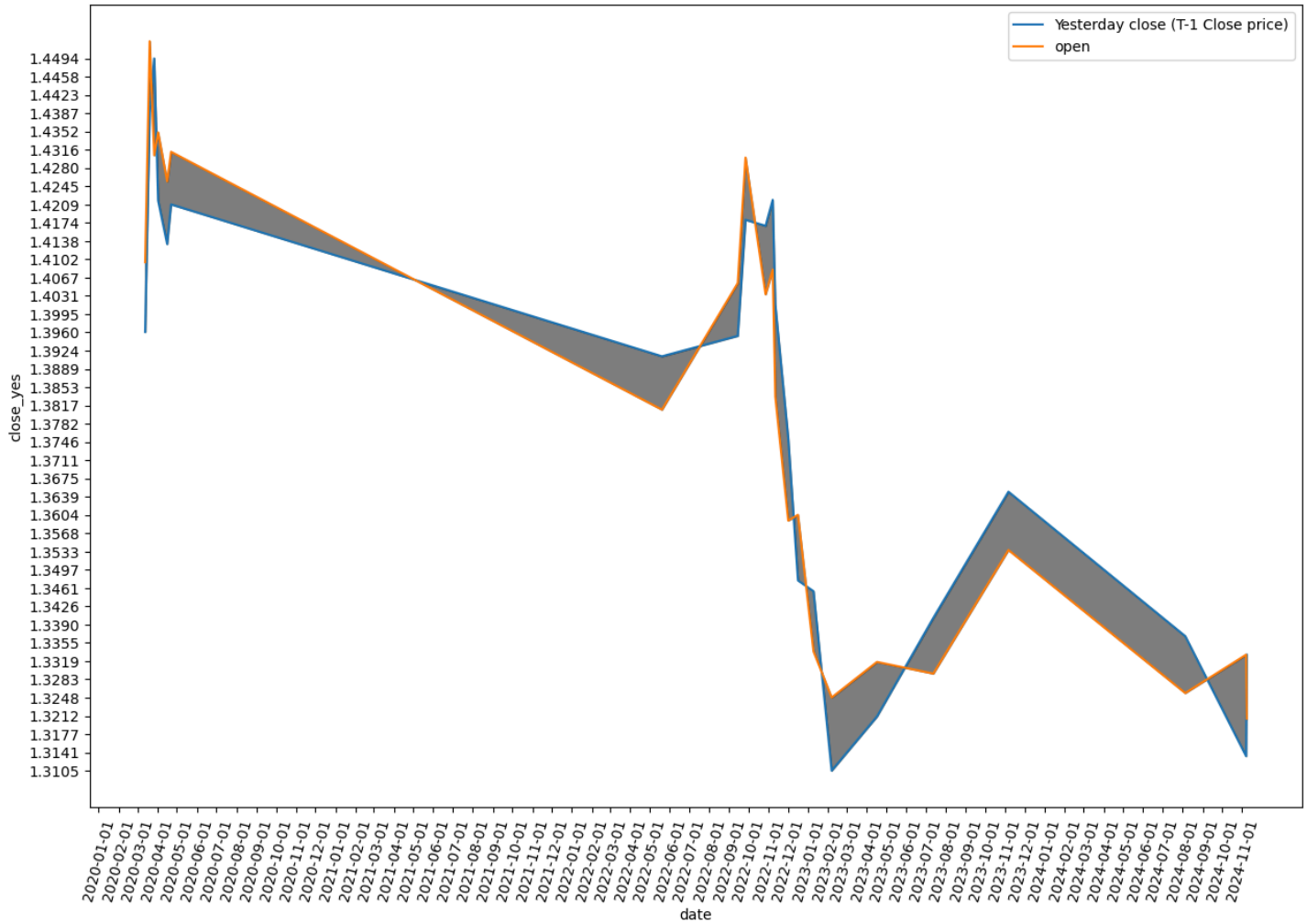
Overnight Gain/Loss data where movement is more than 2 cents USD/NZD



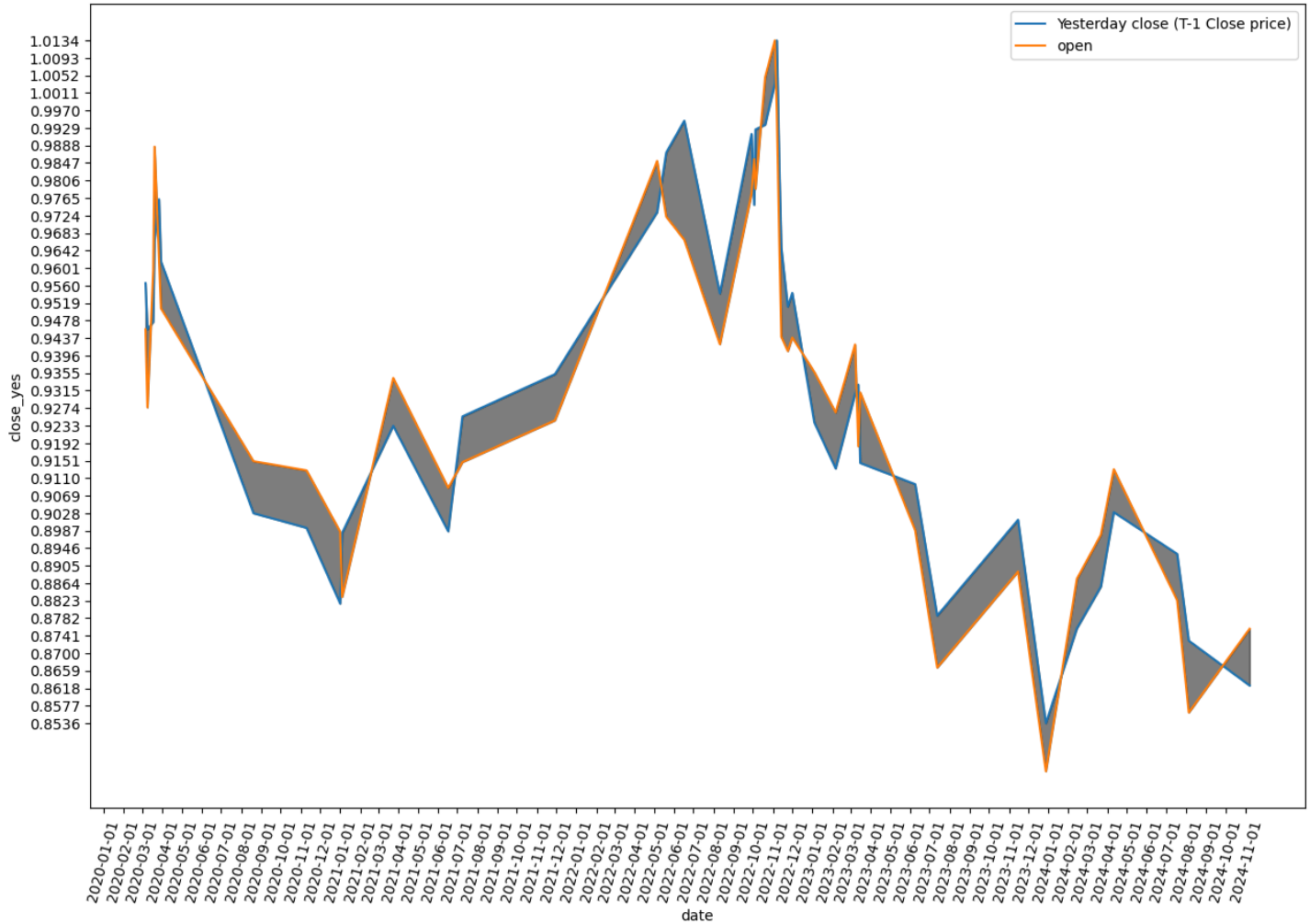
Overnight Gain/Loss data where movement is more than 2 cents USD/CAD



Overnight Gain/Loss data where movement is more than 2 cents USD/SGD



Overnight Gain/Loss data where movement is more than 2 cents USD/CHF





- The New Zealand Dollar (NZD) shows the least change overnight, indicating greater stability in its value.
- SGD, GBP, and EUR experience noticeable overnight gains and losses, reflecting more significant fluctuations in their exchange rates.

Insight -

Visual Insights

- The heatmap clearly indicated that the GBP is most valuable currency. However sharp dips in the GBP are closely correlated with economic policy, the political environment, and various public policie. Whereas the CHF and SGD maintained stability, exhibiting minimal fluctuations.

Overnight Gain/Loss

- The New Zealand Dollar (NZD) demonstrated minimal overnight changes, suggesting it is a stable option for investors.
- The USD, GBP, and EUR exhibited considerable overnight gains and losses, indicating increased volatility during this period.

Volatility

- High volatility of the Australian Dollar (AUD) and the New Zealand Dollar (NZD) indicates a higher risk, which could be attractive for traders looking to capitalize on short-term price movements.
- Swiss Franc (CHF) and Singapore Dollar (SGD) can be regarded as safe-haven currencies. Their stability suggests they may offer opportunities for risk-averse investors looking to hedge against currency volatility.

References

[1] - [Brexit Vote \(2016\)](#)

[2] - [Boris Johnson's taking office as PM \(2019\)](#)

[3] - [UK Government's Tax Cut Announcement \(2022\)](#)

In [ ]:

print