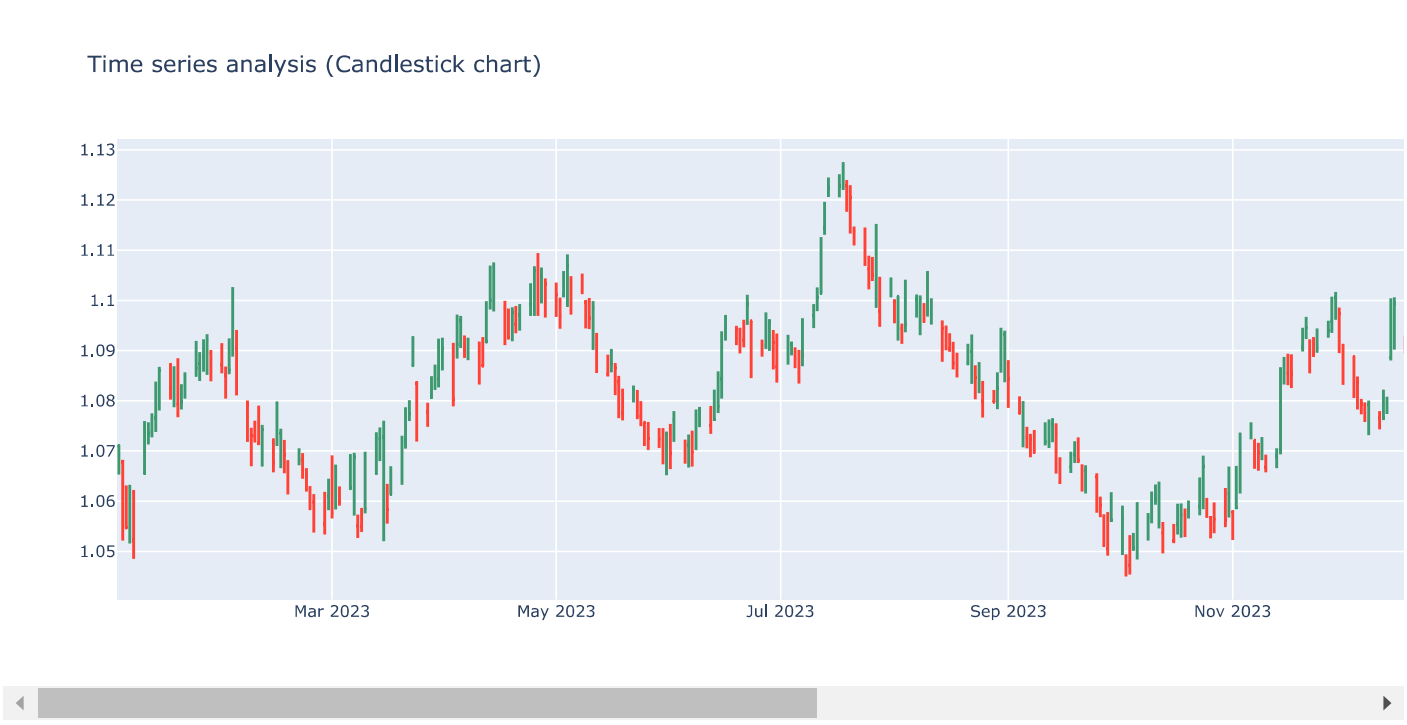
```
import pandas as pd
import yfinance as yf
import datetime
from datetime import date, timedelta
today = date.today()


d1 = today.strftime("%Y-%m-%d")
end_date = d1
d2 = date.today() - timedelta(days=720)
d2 = d2.strftime("%Y-%m-%d")
start_date = d2


start_date = "2023-01-01"
end_date = "2024-01-01"

data = yf.download('EURUSD=X', start=start_date, end=end_date, progress=False)


print(data.head())
```

```
                Open      High       Low     Close  Adj Close  Volume
Date
2023-01-02  1.070973  1.071237  1.065326  1.070973   1.070973       0
2023-01-03  1.067771  1.068262  1.052155  1.067771   1.067771       0
2023-01-04  1.054685  1.063151  1.054596  1.054685   1.054685       0
2023-01-05  1.060637  1.063264  1.051558  1.060637   1.060637       0
2023-01-06  1.052222  1.062225  1.048526  1.052222   1.052222       0
```

```
import plotly.express as px
figure = px.line(data, x = data.index,
                 y = "Close",  # Change 'close' to 'Close'
                 title = "Time series analysis (line plot)")
figure.show()
```

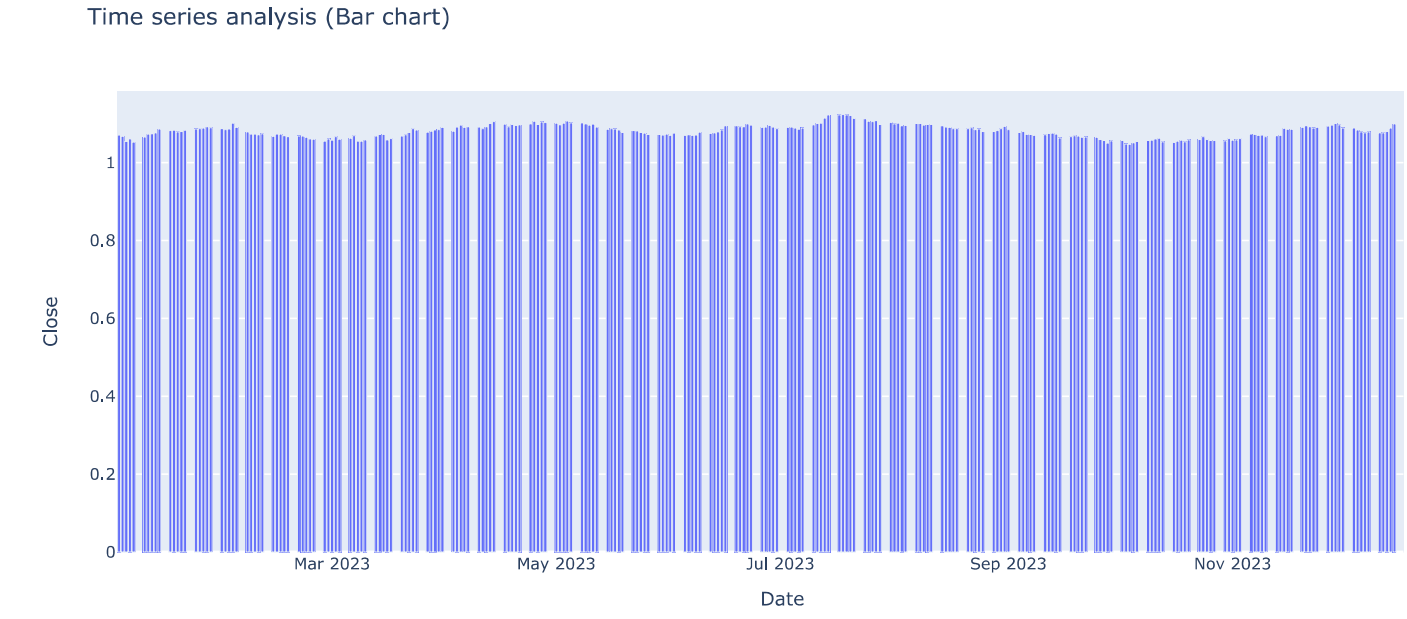
```
/usr/local/lib/python3.10/dist-packages/_plotly_utils/basevalidators.py:105: FutureWarning:

The behavior of DatetimeProperties.to_pydatetime is deprecated, in a future version this will return a Series containing python datetime
```

```
import plotly.graph_objects as go
figure = go.Figure(data=[go.Candlestick(x=data.index,
                                        open=data['Open'],
                                        high=data['High'],
                                        low=data['Low'],
                                        close=data['Close'])])
figure.update_layout(title = "Time series analysis (Candlestick chart)",
                     xaxis_rangeslider_visible=False)
figure.show()
```



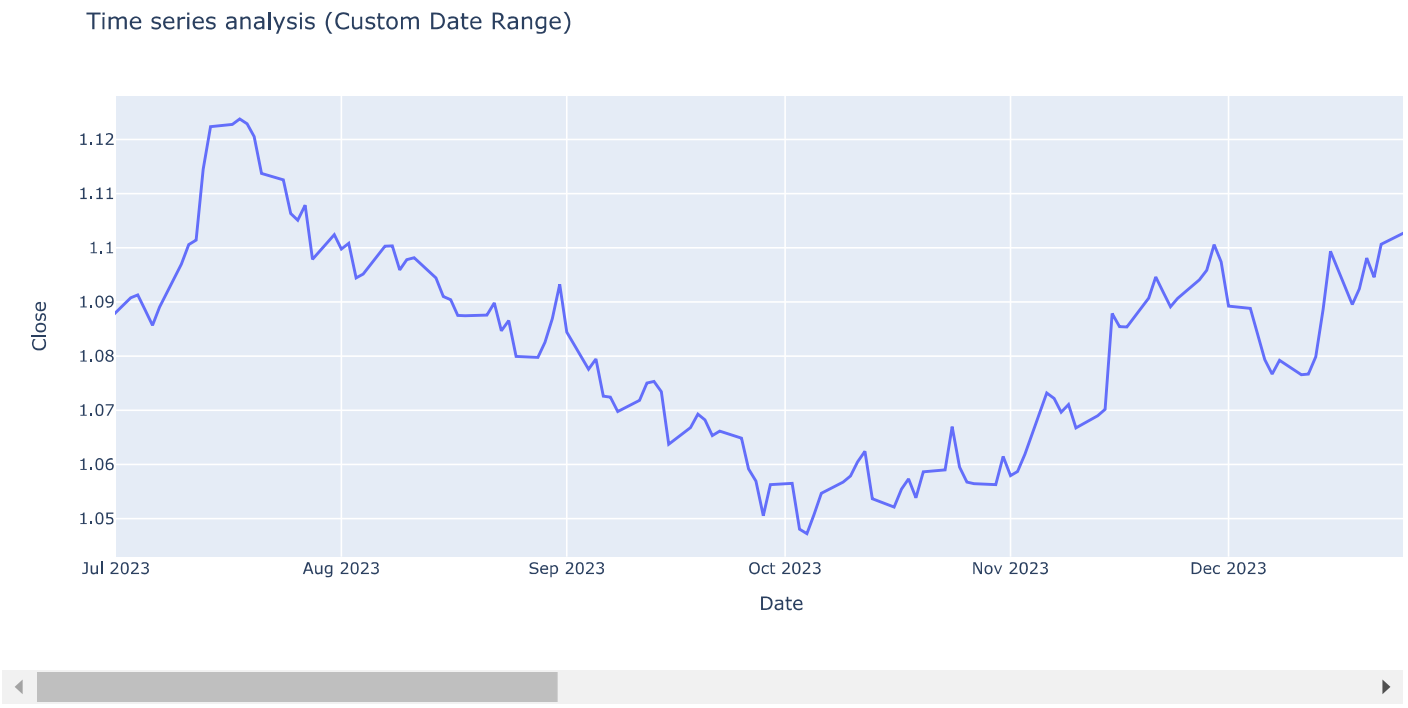

```
figure = px.bar(data, x = data.index,
                y = "Close",
                title = "Time series analysis (Bar chart)")
figure.show()
```

```
/usr/local/lib/python3.10/dist-packages/_plotly_utils/basevalidators.py:105: FutureWarning:

The behavior of DatetimeProperties.to_pydatetime is deprecated, in a future version this will return a Series containing python datetime
```



```
figure = px.line(data, x = data.index,
                 y = 'Close',
                 range_x = ['2023-07-01', '2023-12-31'],
                 title = "Time series analysis (Custom Date Range)")
figure.show()
```

```
/usr/local/lib/python3.10/dist-packages/_plotly_utils/basevalidators.py:105: FutureWarning:

The behavior of DatetimeProperties.to_pydatetime is deprecated, in a future version this will return a Series containing python datetime
```




```
figure = go.Figure(data = [go.Candlestick(x = data.index,
                                          open = data["Open"],
                                          high = data["High"],
                                          low = data["Low"],
                                          close = data["Close"])])
figure.update_layout(title = "Time Series Analysis (Candlestick Chart with Buttons and Slider)")

figure.update_xaxes(
    rangeslider_visible = True,
    rangeselector = dict(
        buttons = list([
            dict(count = 1, label = "1m", step = "month", stepmode = "backward"),
            dict(count = 6, label = "6m", step = "month", stepmode = "backward"),
            dict(count = 1, label = "YTD", step = "year", stepmode = "todate"),
            dict(count = 1, label = "1y", step = "year", stepmode = "backward"),
            dict(step = "all")
        ])
    )
)
figure.show()
```


Time Series Analysis (Candlestick Chart with Buttons and Slider)
1m
6m
YTD
1y
all
1.12
1.1
1.08
1.06
Aug 27
2023
Sep 10
Sep 24
Oct 8
Oct 22
Nov 5