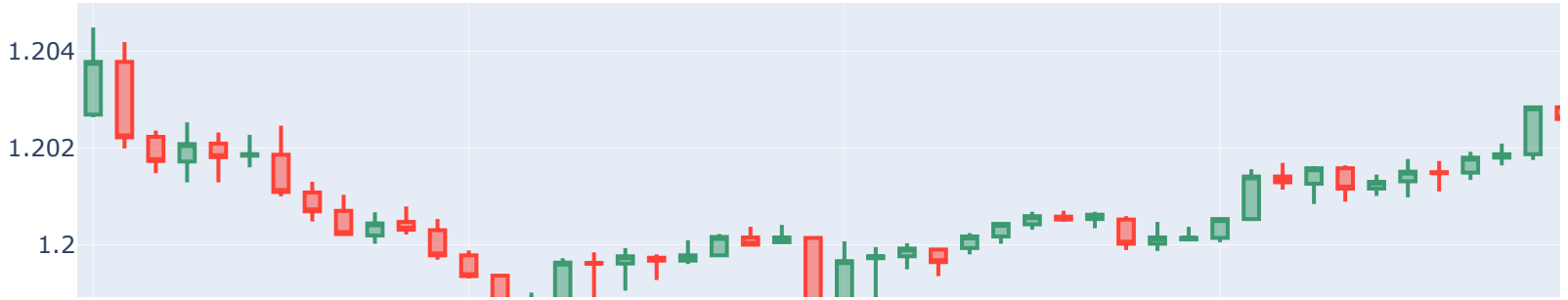
In [1]:
```
# Noah Manz
# 1-12-2023
# Script to connect to MetaTrader 5 and pull price history into a Pandas dataframe

#Import libraries
import MetaTrader5 as mt
print(f'MetaTrader5 version is: {mt.__version__}')
import numpy as np
print(f'Numpy version is {np.__version__}')
from datetime import datetime
import pandas as pd
import mplfinance as mpl
import plotly.graph_objects as go
import talib
import matplotlib.pyplot as plt
```

```
MetaTrader5 version is: 5.0.36
Numpy version is 1.22.3
```

In [2]:
```
# Connect to the trading platform w/ initialize()
if not mt.initialize():
    print('initialize() failed')
    mt.shutdown()

# Declare variables
symbol = 'GBPUSD'
timeframe = mt.TIMEFRAME_M15 # https://www.mql5.com/en/docs/integration/python_metatrader5/mt5copyratesfrom_py#timefram
num_bars = 100 # This is the number of historical bars you want back, counting backwards (inclusive) from the current b

# Collect data from "symbol" using "timeframe" and "num_bars"
ohlc_data = pd.DataFrame(mt.copy_rates_from_pos(symbol, timeframe, 0, num_bars))

# Terminate connection w/ trading platform
mt.shutdown()

if ohlc_data is None:
    print('Error occured. Data not downloaded successfuly')
else:
    print(f'{symbol} data downloaded successfuly')

# Reference https://www.mql5.com/en/docs/integration/python_metatrader5
```

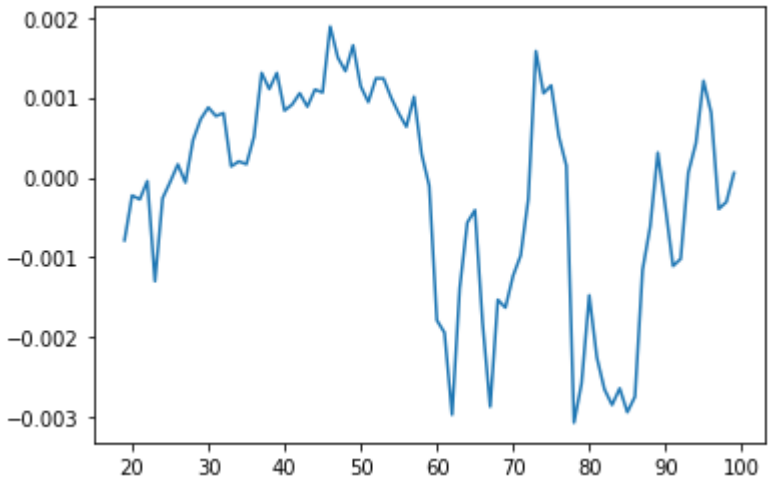
```
GBPUSD data downloaded successfuly
```

In [3]:
```
# Create new instance of the dataset so that the original data isn't affected by drop()
duplicate = ohlc_data

# Drop unnecessary columns (i.e tick_volume, spread & real_volume)
duplicate.drop(['tick_volume', 'spread', 'real_volume'], axis = 1, inplace = True)

# Convert time in seconds into the datetime format
duplicate['time'] = pd.to_datetime(duplicate['time'], unit = 's')

fig = go.Figure(data = [go.Candlestick(x = duplicate['time'],
        open = duplicate['open'],
        high = duplicate['high'],
        low = duplicate['low'],
        close = duplicate['close'])])

fig.show()
```



In [4]:
```
# Calculate bollinger bands
upperband, middleband, lowerband = talib.BBANDS(ohlc_data['close'].values, 20 , 1, 1)

# Plot mean reversion
plt.plot(ohlc_data['close'].values - middleband)
```

Out[4]:
```
[<matplotlib.lines.Line2D at 0x25874d9a770>]
```


0.002
0.001
0.000
−0.001
−0.002
−0.003
20
30
40
50
60
70
80
90
100