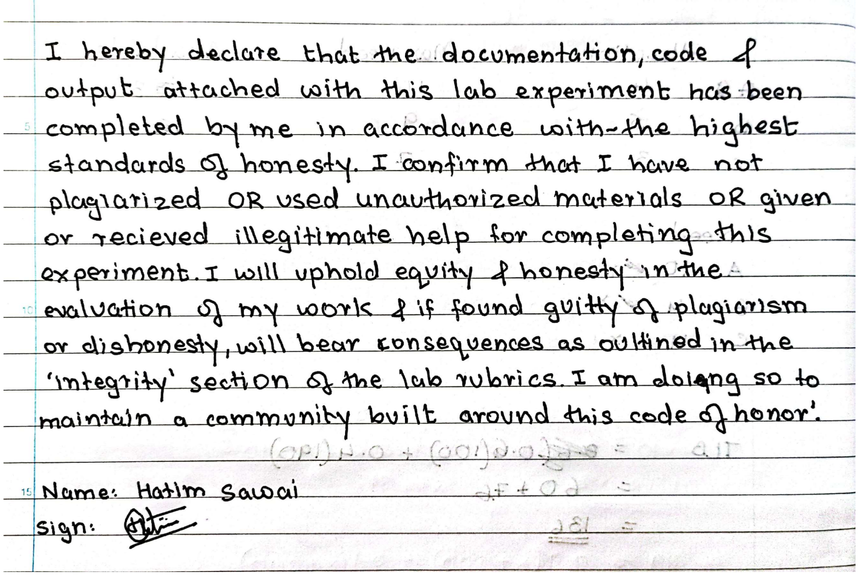
Name	Hatim Sawai
UID No.	2021300108

Experiment 4

HONOR PLEDGE	 I hereby declare that the documentation, code & output attached with this lab experiment has been completed by me in accordance with the highest standards of honesty. I confirm that I have not plagiarized OR used unauthorized materials OR given or recieved illegitimate help for completing this experiment. I will uphold equity & honesty in the evaluation of my work & if found guilty of plagiarism or dishonesty, will bear consequences as outlined in the 'integrity' section of the lab rubrics. I am doing so to maintain a community built around this code of honor'. Name: Hatim Sawai Sign:
PROBLEM STATEMENT	**Dealing with Time Series Data :** 1. Resample a time series to a different time frequency (eg. Daily, monthly). One up sampling and one down sampling and OHLC sampling required to be done 2. Shift a time series forward and backward in time. Use naive shifts, and shift using frequency 3. Compute moving averages or rolling sums over a time series. Apply 3 moving window functions to your dataset
THEORY	**I. Resampling Time Series Data:** **Aim:** Change the frequency of data points in a time series. **Types:** * **Upsampling:** Increase frequency (e.g., daily to hourly). * **Downsampling:** Decrease frequency (e.g., hourly to monthly). **Methods:** **Upsampling:** * **Forward fill (FFill):** Fill missing values with the last available value. * **Linear interpolation:** Estimate missing values by interpolating between existing points. **Downsampling:** * **Mean:** Average values within the chosen period. * **Median:** Choose the middle value within the period. * **OHLC (Open-High-Low-Close):** Summarize the period

with opening, highest, lowest, and closing values.
Choice of method: Depends on data characteristics and research question.
For your experiment: Using `interpolate()` for upsampling might be better than FFill because it captures potential trends between data points, offering a more nuanced representation.

2. Shifting Time Series Data:

Aim: Move data points forward or backward in time.
Types:
* **Naive shift:** Shift all data points by the same fixed amount (e.g., 1 day forward).
* **Shift using frequency:** Shift data points according to the chosen frequency (e.g., shift daily data by 3 days).
Application: Align time series for comparison, model future values, simulate delays.

3. Moving Averages/Rolling Sums:

Aim: Smooth out fluctuations and capture underlying trends in time series data.
Types:
* **Simple moving average (SMA):** Average values within a specified window.
* **Exponential moving average (EMA):** Weights recent values more heavily, giving more importance to recent trends.
* **Weighted moving average (WMA):** Assigns custom weights to values within the window.
Choice of method: Depends on desired level of smoothing and sensitivity to recent trends.

1. Importing Libraries & Dataset

In [ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# read csv file
df = pd.read_csv('../Datasets/trends_musicians.csv')
df.head()
```

Out[ ]:

	Week	Taylor Swift	Drake	Justin Bieber	Eminem	Selena Gomez
0	2010-01-10	13	8	13	10	8
1	2010-01-17	15	9	14	6	7
2	2010-01-24	15	9	14	7	7
3	2010-01-31	30	13	17	11	7
4	2010-02-07	16	9	17	8	8

2. Resampling Time Series Data

In [ ]:

```
# Convert the date index to datetime
df['Week'] = pd.to_datetime(df['Week'])
df.set_index('Week', inplace=True)
df.head()
```

Out[ ]:

	Taylor Swift	Drake	Justin Bieber	Eminem	Selena Gomez
Week					
2010-01-10	13	8	13	10	8
2010-01-17	15	9	14	6	7
2010-01-24	15	9	14	7	7
2010-01-31	30	13	17	11	7
2010-02-07	16	9	17	8	8

In [ ]:

```
# Downsample to monthly frequency using mean
df_monthly = df.resample('M').mean()
df_monthly.head()
```

Out[ ]:

	Taylor Swift	Drake	Justin Bieber	Eminem	Selena Gomez
Week					
2010-01-31	18.25	9.75	14.50	8.50	7.25
2010-02-28	13.25	9.50	21.75	7.50	7.50
2010-03-31	10.00	13.50	38.00	7.25	7.25
2010-04-30	9.50	13.75	42.75	10.25	6.75
2010-05-31	9.20	19.20	40.40	18.00	6.40

In [ ]:

```
# Upsample to daily frequency using forward fill (ffill)
df_daily = df.resample('D').ffill()
df_daily.head()
```

Out[ ]:

	Taylor Swift	Drake	Justin Bieber	Eminem	Selena Gomez
Week					
2010-01-10	13	8	13	10	8
2010-01-11	13	8	13	10	8
2010-01-12	13	8	13	10	8
2010-01-13	13	8	13	10	8
2010-01-14	13	8	13	10	8

In [ ]:

```
# Upsample to daily using interpolation
df_interpolated = df.resample('D').interpolate()
df_interpolated.head()
```

Out[ ]:

	Taylor Swift	Drake	Justin Bieber	Eminem	Selena Gomez
Week					
2010-01-10	13.000000	8.000000	13.000000	10.000000	8.000000
2010-01-11	13.285714	8.142857	13.142857	9.428571	7.857143
2010-01-12	13.571429	8.285714	13.285714	8.857143	7.714286
2010-01-13	13.857143	8.428571	13.428571	8.285714	7.571429
2010-01-14	14.142857	8.571429	13.571429	7.714286	7.428571

In [ ]:

```
# use ohlc method on monthly data
df_ohlc = df.resample('M').ohlc()
df_ohlc.head()
```

Out[ ]:

	Taylor Swift				Drake				Justin Bieber					
	open	high	low	close	open	high	low	close	open	high	low	close	open	hi
Week														
2010-01-31	13	30	13	30	8	13	8	13	13	17	13	17	10	
2010-02-28	16	16	11	12	9	11	8	11	17	26	17	26	8	
2010-03-31	11	11	9	10	13	15	13	13	26	51	26	51	7	
2010-04-30	8	10	8	10	12	15	12	14	48	48	38	39	7	
2010-05-31	9	10	9	9	15	24	15	24	39	45	35	35	20	

3. Shifting Time Series Data

In [ ]:

```
# shift the data using naive method
df_shifted = df.shift(2)
df_shifted.head()
```

Out[ ]:

	Taylor Swift	Drake	Justin Bieber	Eminem	Selena Gomez
Week					
2010-01-10	NaN	NaN	NaN	NaN	NaN
2010-01-17	NaN	NaN	NaN	NaN	NaN
2010-01-24	13.0	8.0	13.0	10.0	8.0
2010-01-31	15.0	9.0	14.0	6.0	7.0
2010-02-07	15.0	9.0	14.0	7.0	7.0

In [ ]:

```
# shift the data foward using freq
df_shifted = df.shift(2, freq='D')
df_shifted.head()
```

Out[ ]:

	Taylor Swift	Drake	Justin Bieber	Eminem	Selena Gomez
Week					
2010-01-12	13	8	13	10	8
2010-01-19	15	9	14	6	7
2010-01-26	15	9	14	7	7
2010-02-02	30	13	17	11	7
2010-02-09	16	9	17	8	8

In [ ]:

```
# shift the data backward using freq
df_shifted = df.shift(-2, freq='W')
df_shifted.head()
```

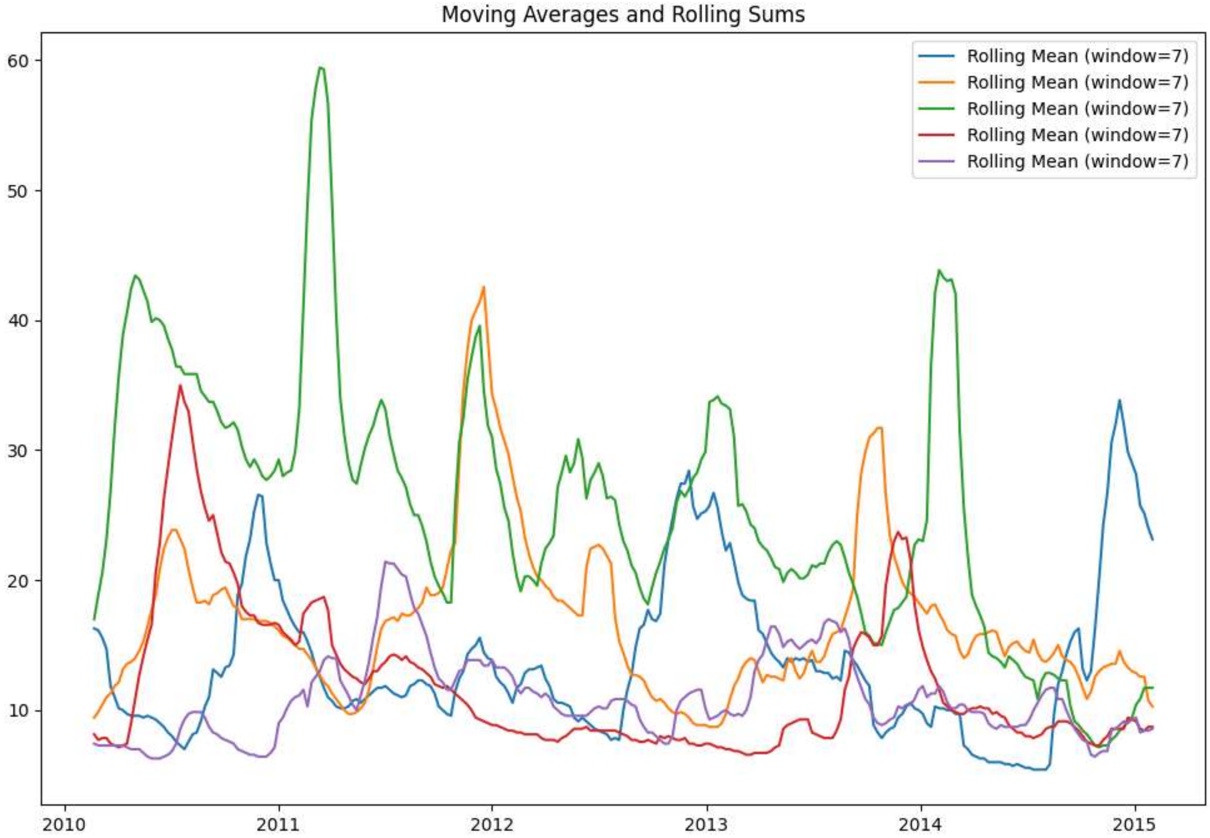
Out[ ]:

	Taylor Swift	Drake	Justin Bieber	Eminem	Selena Gomez
Week					
2009-12-27	13	8	13	10	8
2010-01-03	15	9	14	6	7
2010-01-10	15	9	14	7	7
2010-01-17	30	13	17	11	7
2010-01-24	16	9	17	8	8

4. Moving Averages

```
In [ ]: # Apply 3 moving window functions to your dataset
        rolling_mean_7 = df.rolling(window=7).mean()
        rolling_sum_14 = df.rolling(window=14).sum()
        rolling_expanding_mean = df.expanding().mean()
```

```
In [ ]: plt.figure(figsize=(12, 8))
        plt.plot(rolling_mean_7, label='Rolling Mean (window=7)')
        # plt.plot(rolling_sum_14, label='Rolling Sum (window=14)')
        # plt.plot(rolling_expanding_mean, label='Expanding Mean')
        plt.legend()
        plt.title('Moving Averages and Rolling Sums')
        plt.show()
```



CONCLUSION	In this experiment we learned about resampling, shifting and moving averages of time series data. We learned how to change the frequency of data points in a time series, move data points forward or backward in time and smooth out fluctuations and capture underlying trends in time series data. We also learned about the different methods of resampling, shifting and moving averages and their applications. We also implemented these methods on a dataset and observed the changes in the dataset after applying these methods.