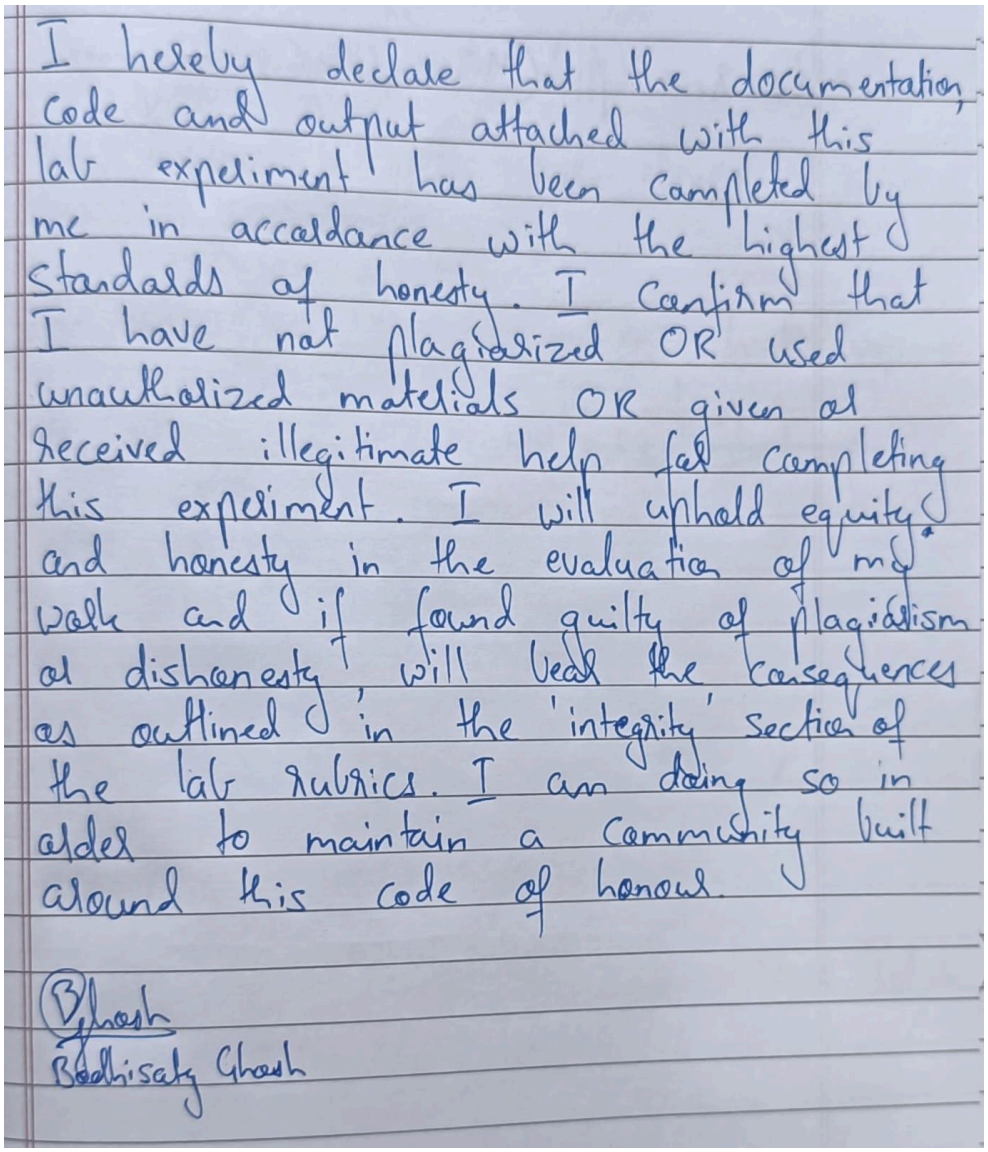
Name	**Bodhisatya Ghosh**
UID no.	**2021700026**

Experiment 4	
HONOUR PLEDGE	
PROBLEM STATEMENT :	**Dealing with Time Series Data** **● 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. For up sampling use FFill **● Shift a time series forward and backward in time** **Use naive shifts, and shift using frequency** **● Compute moving averages or rolling sums over a time series** **Apply 3 moving window functions to your dataset**
THEORY:	Time series data is an important form of structured data in many different fields, such as finance, economics, ecology, neuroscience, and physics. Anything that is recorded repeatedly at many points in time forms a time series. Many time series are *fixed frequency*, which is to say that data points occur at regular intervals according to some rule, such as every 15 seconds, every 5 minutes, or once per month. Time series can also be *irregular* without a fixed unit of time or offset between units. How you mark and refer to time series data depends on the application, and you may have one of the following: **Timestamps** Specific instants in time. **Fixed periods** Such as the whole month of January 2017, or the whole year 2020. **Intervals of time** Indicated by a start and end timestamp. Periods can be thought of as special cases of intervals. **Experiment or elapsed time** Each timestamp is a measure of time relative to a particular start time (e.g., the diameter of a cookie baking each second since being placed in the oven), starting from 0.

RESULT:

Resampling

Downsampling

```
df.resample('M').median()
```

[591] ✓ 0.0s Python

	astrophysicist: (India)	data analyst: (India)	chef: (India)	software developer: (India)	teacher: (India)
2019-03-31	18.0	19.0	49.0	26.0	41.0
2019-04-30	14.0	18.5	43.5	24.0	38.0
2019-05-31	12.0	20.0	44.5	28.5	38.0
2019-06-30	22.0	21.0	46.0	31.0	39.0
2019-07-31	20.0	23.5	52.5	32.0	42.5
...	...	...	...	...	...
2023-11-30	24.5	72.5	60.5	46.0	36.0
2023-12-31	35.0	79.0	59.0	49.0	39.0
2024-01-31	23.0	89.0	58.0	58.5	38.0
2024-02-29	21.5	94.5	57.0	61.0	40.5
2024-03-31	27.0	91.0	61.0	59.0	38.0

61 rows × 5 columns

Upsampling

```
df.resample('D').median().ffill()
```

[592] ✓ 0.0s Python

	astrophysicist: (India)	data analyst: (India)	chef: (India)	software developer: (India)	teacher: (India)
2019-03-17	26.0	18.0	51.0	26.0	41.0
2019-03-18	26.0	18.0	51.0	26.0	41.0
2019-03-19	26.0	18.0	51.0	26.0	41.0
2019-03-20	26.0	18.0	51.0	26.0	41.0
2019-03-21	26.0	18.0	51.0	26.0	41.0
...	...	...	...	...	...
2024-02-28	14.0	96.0	59.0	61.0	40.0
2024-02-29	14.0	96.0	59.0	61.0	40.0
2024-03-01	14.0	96.0	59.0	61.0	40.0
2024-03-02	14.0	96.0	59.0	61.0	40.0
2024-03-03	27.0	91.0	61.0	59.0	38.0

1814 rows × 5 columns

OHLC

OHLC

```
df.resample('M').ohlc()
```

[593] ✓ 0.0s Python

	astrophysicist: (India)				data analyst: (India)				chef: (India)				software developer: (India)				teacher: (India)			
	open	high	low	close	open	high	low	close	open	high	low	close	open	high	low	close	open	high	low	close
2019-03-31	26	26	13	18	18	19	18	19	51	51	47	47	26	27	26	27	41	43	41	41
2019-04-30	11	24	0	17	19	19	17	18	50	50	43	44	23	26	23	26	40	40	37	37
2019-05-31	12	12	12	12	19	24	19	24	44	50	44	50	32	32	27	28	37	39	37	38
2019-06-30	22	25	12	23	19	23	19	21	46	48	45	46	28	32	28	31	38	40	38	40
2019-07-31	22	34	17	34	23	24	23	24	49	55	49	55	31	34	31	34	41	43	41	43
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2023-11-30	22	42	22	42	68	85	67	77	59	66	58	66	44	59	43	59	35	36	35	36
2023-12-31	35	41	22	41	79	83	72	80	71	71	56	61	55	56	47	56	37	45	37	39
2024-01-31	25	25	14	25	89	94	89	94	61	61	56	58	65	65	53	62	38	40	37	40
2024-02-29	23	30	14	14	93	100	89	96	54	59	54	59	58	68	58	61	42	42	39	40
2024-03-31	27	27	27	27	91	91	91	91	61	61	61	61	59	59	59	59	38	38	38	38

61 rows × 20 columns

Shifting time series

Ln 5, Col 8 (7 selected) Spaces: 4 Spaces: 4 CRLF Cell 23 of 25

Shifting time series

Naive shifts

```
# Shifting forward
.shift(2)
```

[594] ✓ 0.0s Python

	astrophysicist: (India)	data analyst: (India)	chef: (India)	software developer: (India)	teacher: (India)
2019-03-17	NaN	NaN	NaN	NaN	NaN
2019-03-24	NaN	NaN	NaN	NaN	NaN
2019-03-31	26.0	18.0	51.0	26.0	41.0
2019-04-07	13.0	19.0	49.0	26.0	43.0
2019-04-14	18.0	19.0	47.0	27.0	41.0
...	...	...	...	...	...
2024-02-04	21.0	89.0	56.0	55.0	38.0
2024-02-11	25.0	94.0	58.0	62.0	40.0
2024-02-18	23.0	93.0	54.0	58.0	42.0
2024-02-25	20.0	89.0	58.0	68.0	39.0
2024-03-03	30.0	100.0	56.0	61.0	41.0

260 rows × 5 columns

```
# Shifting backwards
df.shift(-2)
```

[595] ✓ 0.0s Python

	astrophysicist: (India)	data analyst: (India)	chef: (India)	software developer: (India)	teacher: (India)
2019-03-17	18.0	19.0	47.0	27.0	41.0
2019-03-24	11.0	19.0	50.0	23.0	40.0
2019-03-31	0.0	17.0	43.0	23.0	38.0
2019-04-07	24.0	19.0	43.0	25.0	38.0
2019-04-14	17.0	18.0	44.0	26.0	37.0
...	...	...	...	...	...
2024-02-04	30.0	100.0	56.0	61.0	41.0
2024-02-11	14.0	96.0	59.0	61.0	40.0
2024-02-18	27.0	91.0	61.0	59.0	38.0
2024-02-25	NaN	NaN	NaN	NaN	NaN
2024-03-03	NaN	NaN	NaN	NaN	NaN

260 rows × 5 columns

Frequency shifting

```
# Shifting forward
df.shift(2,freq='W')
```

[596] ✓ 0.0s Python

	astrophysicist: (India)	data analyst: (India)	chef: (India)	software developer: (India)	teacher: (India)
2019-03-31	26	18	51	26	41
2019-04-07	13	19	49	26	43
2019-04-14	18	19	47	27	41
2019-04-21	11	19	50	23	40
2019-04-28	0	17	43	23	38
...	...	...	...	...	...
2024-02-18	23	93	54	58	42
2024-02-25	20	89	58	68	39
2024-03-03	30	100	56	61	41
2024-03-10	14	96	59	61	40
2024-03-17	27	91	61	59	38

260 rows × 5 columns

```
# Shifting backwards
df.shift(-2,freq='W')
```

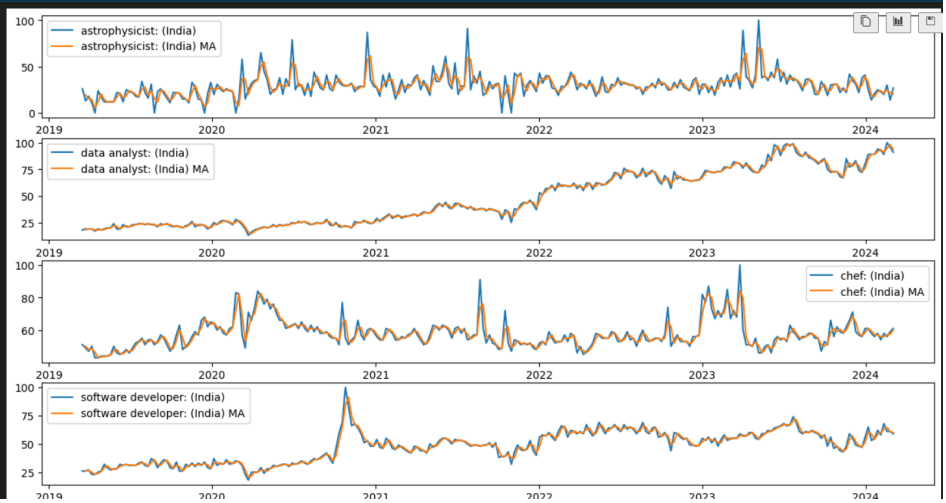
[597] ✓ 0.0s

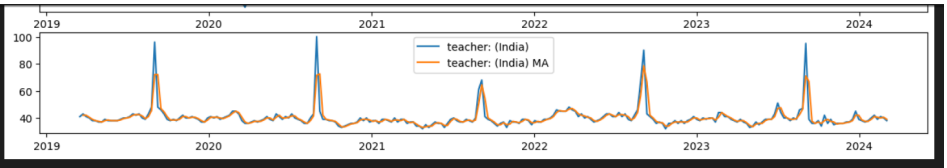
	astrophysicist: (India)	data analyst: (India)	chef: (India)	software developer: (India)	teacher: (India)
2019-03-03	26	18	51	26	41
2019-03-10	13	19	49	26	43
2019-03-17	18	19	47	27	41
2019-03-24	11	19	50	23	40
2019-03-31	0	17	43	23	38
...	...	...	...	...	...
2024-01-21	23	93	54	58	42
2024-01-28	20	89	58	68	39
2024-02-04	30	100	56	61	41
2024-02-11	14	96	59	61	40
2024-02-18	27	91	61	59	38

260 rows × 5 columns

```
getMaPlots(df,2)
```

[600] ✓ 0.8s






```
getMaPlots(df,4)
```

[601] ✓ 0.9s

REFERENCES:	pandas.merge — pandas 2.2.1 documentation (pydata.org)pandas.DataFrame.shift — pandas 2.2.1 documentation (pydata.org)pandas.DataFrame.rolling — pandas 2.2.1 documentation (pydata.org)pandas.DataFrame.resample — pandas 3.0.0.dev0+514.gd6c258691d documentation (pydata.org)

CONCLUSION: In this experiment, I have learnt how to handle, resample, shift and calculate the moving average for time series data