

Time series analysis

Importing necessary libraries

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

Importing the datasets

```
In [2]: df1=pd.read_csv("dataset_tk.csv")
df2=pd.read_csv("long_data_.csv")
```

Exploring the datasets

In [3]:

df1

	Unnamed: 0	Punjab	Haryana	Rajasthan	Delhi	UP	Uttarakhand	HP	J&K	Chandigarh	...	Odisha	West Bengal	Sikkim	Arunachal Pradesh	Assam	Manipur	Meghalaya	Mizoram	Nagaland
0	02/01/2019 00:00:00	119.9	130.3	234.1	85.8	313.9	40.7	30.0	52.5	5.0	...	70.2	108.2	2.0	2.1	21.7	2.7	6.1	1.9	2.2
1	03/01/2019 00:00:00	121.9	133.5	240.2	85.5	311.8	39.3	30.1	54.1	4.9	...	67.9	110.2	1.9	2.2	23.4	2.4	6.5	1.8	2.2
2	04/01/2019 00:00:00	118.8	128.2	239.8	83.5	320.7	38.1	30.1	53.2	4.8	...	66.3	106.8	1.7	2.2	21.7	2.4	6.3	1.7	2.2
3	05/01/2019 00:00:00	121.0	127.5	239.1	79.2	299.0	39.2	30.2	51.5	4.3	...	65.8	107.0	2.0	2.2	22.5	2.7	5.7	1.8	2.3
4	06/01/2019 00:00:00	121.4	132.6	240.4	76.6	286.8	39.2	31.0	53.2	4.3	...	62.9	106.4	2.0	2.2	21.7	2.7	6.2	1.9	2.3
...
498	01/12/2020 00:00:00	130.8	126.0	220.4	77.3	322.8	36.5	25.4	45.4	3.6	...	105.7	172.6	1.4	2.1	20.4	2.1	5.4	1.8	2.0
499	02/12/2020 00:00:00	129.4	127.3	218.4	81.3	331.4	37.0	25.8	45.2	3.7	...	112.3	174.4	1.5	2.1	23.3	2.4	5.4	1.6	2.1
500	03/12/2020 00:00:00	132.1	129.7	205.6	85.6	336.7	37.8	26.2	45.2	4.0	...	113.0	169.8	1.4	2.1	21.7	2.3	5.4	1.5	2.0
501	04/12/2020 00:00:00	132.1	128.4	207.0	83.9	334.6	38.2	27.0	43.0	4.0	...	111.6	145.5	1.2	2.0	24.2	2.5	5.7	1.6	2.0
502	05/12/2020 00:00:00	120.4	111.2	201.4	83.6	287.3	35.2	24.3	42.6	3.5	...	95.1	110.4	1.2	2.1	20.3	2.5	5.8	1.6	2.1

503 rows × 34 columns

In [4]:

df2

Out[4]:

	States	Regions	latitude	longitude	Dates	Usage
0	Punjab	ND	31.510074	75.680093	02/01/2019 00:00:00	119.9

```
In [4]: df2
```

	States	Regions	latitude	longitude	Dates	Usage
0	Punjab	NR	31.519974	75.980003	02/01/2019 00:00:00	119.9
1	Haryana	NR	28.450006	77.019991	02/01/2019 00:00:00	130.3
2	Rajasthan	NR	26.449999	74.639981	02/01/2019 00:00:00	234.1
3	Delhi	NR	28.669993	77.230004	02/01/2019 00:00:00	85.8
4	UP	NR	27.599981	78.050006	02/01/2019 00:00:00	313.9
...
16594	Manipur	NER	24.799971	93.950017	05/12/2020 00:00:00	2.5
16595	Meghalaya	NER	25.570492	91.880014	05/12/2020 00:00:00	5.8
16596	Mizoram	NER	23.710399	92.720015	05/12/2020 00:00:00	1.6
16597	Nagaland	NER	25.666998	94.116570	05/12/2020 00:00:00	2.1
16598	Tripura	NER	23.835404	91.279999	05/12/2020 00:00:00	3.3

16599 rows × 6 columns

```
In [6]: df1.shape
```

Out[6]: (503, 34)

```
In [7]: df2.shape
```

Out[7]: (16599, 6)

```
In [8]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 503 entries, 0 to 502
Data columns (total 34 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Unnamed: 0          503 non-null   object
1   Punjab              503 non-null   float64
2   Haryana             503 non-null   float64
3   Rajasthan           503 non-null   float64
4   Delhi               503 non-null   float64
5   UP                  503 non-null   float64
6   Uttarakhand         503 non-null   float64
7   HP                  503 non-null   float64
8   J&K                 503 non-null   float64
9   Chandigarh         503 non-null   float64
10  Chhattisgarh        503 non-null   float64
11  Gujarat             503 non-null   float64
12  MP                  503 non-null   float64
13  Maharashtra         503 non-null   float64
14  Goa                 503 non-null   float64
15  DNH                 503 non-null   float64
16  Andhra Pradesh      503 non-null   float64
17  Telangana           503 non-null   float64
18  Karnataka           503 non-null   float64
19  Kerala              503 non-null   float64
20  Tamil Nadu          503 non-null   float64
21  Pondy               503 non-null   float64
22  Bihar               503 non-null   float64
23  Jharkhand           503 non-null   float64
24  Odisha              503 non-null   float64
25  West Bengal         503 non-null   float64
26  Sikkim              503 non-null   float64
27  Arunachal Pradesh   503 non-null   float64
28  Assam               503 non-null   float64
29  Manipur             503 non-null   float64
30  Meghalaya           503 non-null   float64
31  Mizoram             503 non-null   float64
32  Nagaland            503 non-null   float64
33  Tripura             503 non-null   float64
dtypes: float64(33), object(1)
memory usage: 133.7+ KB
```

```
In [9]: df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16599 entries, 0 to 16598
Data columns (total 6 columns):
#   Column              Non-Null Count  Dtype
---  -
0   States              16599 non-null object
1   Regions             16599 non-null object
2   latitude            16599 non-null float64
3   longitude            16599 non-null float64
4   Dates               16599 non-null object
5   Usage               16599 non-null float64
dtypes: float64(3), object(3)
memory usage: 778.2+ KB
```

```
In [10]: df1.describe()
```

	Punjab	Haryana	Rajasthan	Delhi	UP	Uttarakhand	HP	J&K	Chandigarh	Chhattisgarh	...	Odisha	West Bengal	Sikkim	Arunachal Pradesh
count	503.000000	503.000000	503.000000	503.000000	503.000000	503.000000	503.000000	503.000000	503.000000	503.000000	...	503.000000	503.000000	503.000000	503.000000
mean	141.145527	138.333598	218.443340	83.380716	314.036382	36.157058	26.568191	44.264016	4.141551	83.877137	...	80.464612	138.708151	1.289463	2.109145
std	56.977361	38.106593	27.421615	25.915357	66.516960	6.705108	4.807040	4.769391	1.143422	10.130427	...	10.905131	27.849180	0.363565	0.231132
min	56.100000	64.800000	105.800000	41.800000	186.800000	16.800000	11.800000	17.800000	2.200000	37.200000	...	51.100000	51.500000	0.300000	1.100000
25%	104.000000	114.800000	205.800000	63.500000	263.650000	33.800000	25.600000	41.550000	3.300000	75.700000	...	71.650000	114.650000	1.100000	2.000000
50%	118.300000	126.800000	222.900000	72.700000	290.000000	37.000000	28.000000	44.100000	3.800000	82.600000	...	79.800000	131.200000	1.200000	2.200000
75%	162.500000	158.100000	237.600000	105.800000	370.550000	40.350000	29.700000	47.350000	4.900000	91.600000	...	88.400000	164.750000	1.400000	2.200000
max	300.000000	237.200000	278.000000	147.100000	471.800000	53.200000	34.000000	54.200000	7.400000	111.600000	...	113.000000	198.500000	2.800000	2.700000

8 rows × 33 columns

```
In [12]: df2.describe()
```

	latitude	longitude	Usage
count	16599.000000	16599.000000	16599.000000
mean	23.178220	81.794533	103.001862
std	6.146575	7.258429	116.044056
min	8.900373	71.192400	0.300000
25%	19.820430	76.569993	6.700000
50%	23.835404	78.570026	64.400000
75%	27.333330	88.329947	173.900000
max	33.450000	94.216667	522.100000

```
In [13]: df1.head()
```

Unnamed: 0	Punjab	Haryana	Rajasthan	Delhi	UP	Uttarakhand	HP	J&K	Chandigarh	...	Odisha	West Bengal	Sikkim	Arunachal Pradesh	Assam	Manipur	Meghalaya	Mizoram	Nagaland	T
0	02/01/2019 00:00:00	119.9	130.3	234.1	85.8	313.9	40.7	30.0	52.5	5.0	...	70.2	108.2	2.0	2.1	21.7	2.7	6.1	1.9	2.2
1	03/01/2019 00:00:00	121.9	133.5	240.2	85.5	311.8	39.3	30.1	54.1	4.9	...	67.9	110.2	1.9	2.2	23.4	2.4	6.5	1.8	2.2
2	04/01/2019 00:00:00	118.8	128.2	239.8	83.5	320.7	38.1	30.1	53.2	4.8	...	66.3	106.8	1.7	2.2	21.7	2.4	6.3	1.7	2.2
3	05/01/2019 00:00:00	121.0	127.5	239.1	79.2	299.0	39.2	30.2	51.5	4.3	...	65.8	107.0	2.0	2.2	22.5	2.7	5.7	1.8	2.3
4	06/01/2019 00:00:00	121.4	132.6	240.4	76.6	286.8	39.2	31.0	53.2	4.3	...	62.9	106.4	2.0	2.2	21.7	2.7	6.2	1.9	2.3

5 rows × 34 columns

```
In [14]: df2.head()
```

	States	Regions	latitude	longitude	Dates	Usage
0	Punjab	NR	31.519974	75.980003	02/01/2019 00:00:00	119.9
1	Haryana	NR	28.450006	77.019991	02/01/2019 00:00:00	130.3
2	Rajasthan	NR	26.449999	74.639981	02/01/2019 00:00:00	234.1
3	Delhi	NR	28.669993	77.230004	02/01/2019 00:00:00	85.8
4	UP	NR	27.599981	78.050006	02/01/2019 00:00:00	313.9

ANALYSIS OF 1ST DATASET

```
In [15]: DATE=df1.rename({'Unnamed: 0':'Date'}, axis=1, inplace=True)
```

```
In [16]: df1
```

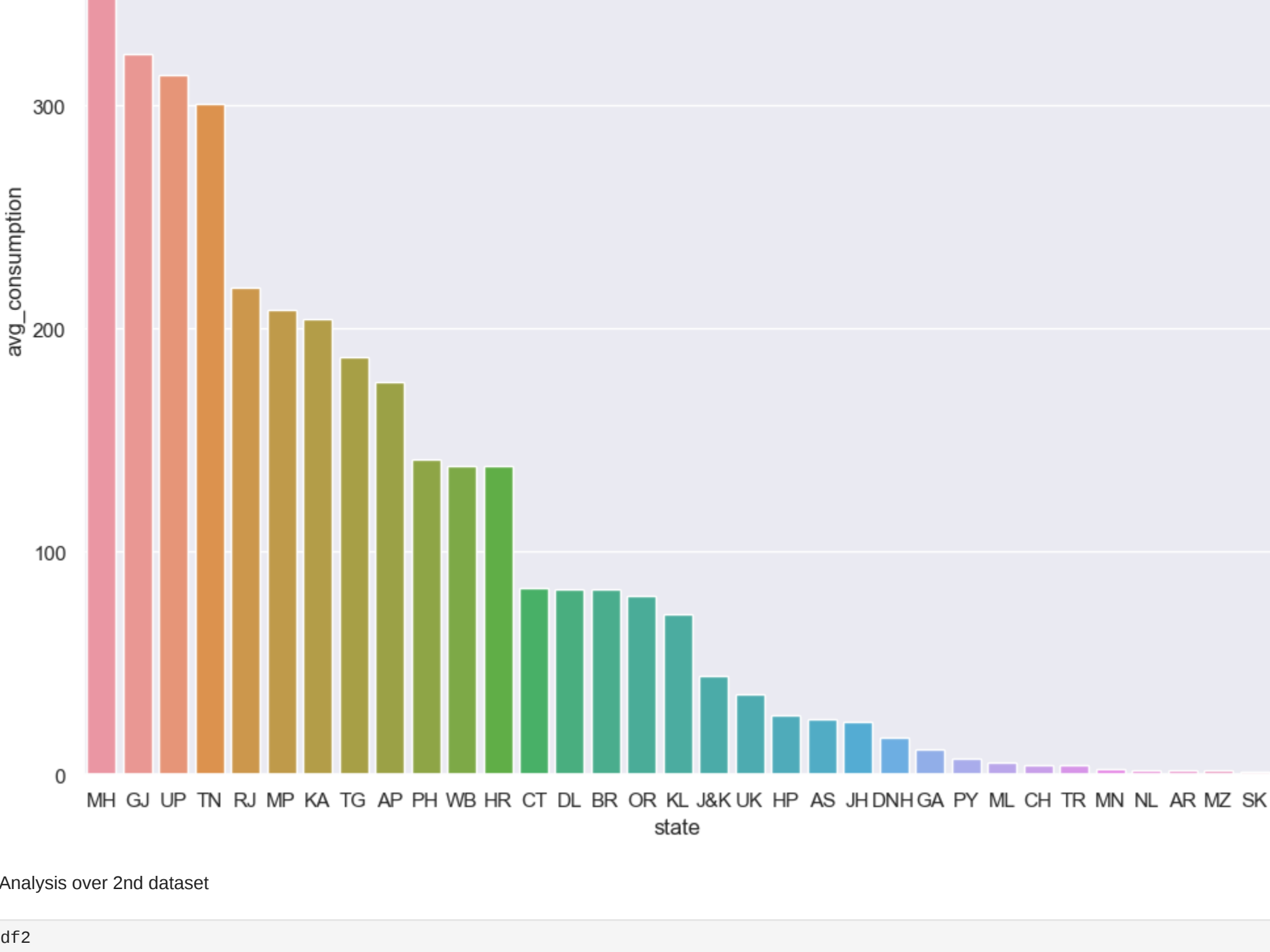
	Date	Punjab	Haryana	Rajasthan	Delhi	UP	Uttarakhand	HP	J&K	Chandigarh	...	Odisha	West Bengal	Sikkim	Arunachal Pradesh	Assam	Manipur	Meghalaya	Mizoram	Nagaland
0	02/01/2019 00:00:00	119.9	130.3	234.1	85.8	313.9	40.7	30.0	52.5	5.0	...	70.2	108.2	2.0	2.1	21.7	2.7	6.1	1.9	2.2
1	03/01/2019 00:00:00	121.9	133.5	240.2	85.5	311.8	39.3	30.1	54.1	4.9	...	67.9	110.2	1.9	2.2	23.4	2.4	6.5	1.8	2.2
2	04/01/2019 00:00:00	118.8	128.2	239.8	83.5	320.7	38.1	30.1	53.2	4.8	...	66.3	106.8	1.7	2.2	21.7	2.4	6.3	1.7	2.2
3	05/01/2019 00:00:00	121.0	127.5	239.1	79.2	299.0	39.2	30.2	51.5	4.3	...	65.8	107.0	2.0	2.2	22.5	2.7	5.7	1.8	2.3
4	06/01/2019 00:00:00	121.4	132.6	240.4	76.6	286.8	39.2	31.0	53.2	4.3	...	62.9	106.4	2.0	2.2	21.7	2.7	6.2	1.9	2.3
...
498	01/12/2020 00:00:00	130.8	126.0	220.4	77.3	322.8	36.5	25.4	45.4	3.6	...	105.7	172.6	1.4	2.1	20.4	2.1	5.4	1.8	2.0
499	02/12/2020 00:00:00	129.4	127.3	218.4	81.3	331.4	37.0	25.8	45.2	3.7	...	112.3	174.4	1.5	2.1	23.3	2.4	5.4	1.6	2.1
500	03/12/2020 00:00:00	132.1	129.7	205.6	85.6	336.7	37.8	26.2	45.2	4.0	...	113.0	169.8	1.4	2.1	21.7	2.3	5.4	1.5	2.0
501	04/12/2020 00:00:00	132.1	128.4	207.0	83.9	334.6	38.2	27.0	43.0	4.0	...	111.6	145.5	1.2	2.0	24.2	2.5	5.7	1.6	2.0
502	05/12/2020 00:00:00	120.4	111.2	201.4	83.6	287.3	35.2	24.3	42.6	3.5	...	95.1	110.4	1.2	2.1	20.3	2.5	5.8	1.6	2.1

503 rows × 34 columns

```
In [17]: numeric_columns = df1.select_dtypes(include=[np.number]).columns
mean_temperature = df1[numeric_columns].mean().sort_values(ascending=False).reset_index().rename(columns={"index": "state", 0: "avg_consumption"})

state_code = ['MH', 'GJ', 'UP', 'TN', 'RJ', 'MP', 'KA', 'TG', 'AP', 'PH', 'WB', 'HR', 'CT', 'DL', 'BR', 'OR', 'KL', 'J&K', 'UK', 'HP', 'AS', 'JH',
if len(state_code) != len(mean_temperature):
    mean_temperature.state = state_code
else:
    print("Lengths of state_code and mean_temperature do not match.")
```

```
In [26]: sns.barplot(x="state", y = "avg_consumption", data = mean_temperature)
sns.set(rc={'figure.figsize':(11.7,6)})
plt.show()
```



Analysis over 2nd dataset

1	Haryana	NR	28.450006	77.019991	02/01/2019 00:00:00	130.3
2	Rajasthan	NR	26.449999	74.639981	02/01/2019 00:00:00	234.1
3	Delhi	NR	28.669993	77.230004	02/01/2019 00:00:00	85.8
4	UP	NR	27.599981	78.050006	02/01/2019 00:00:00	313.9
...
16594	Manipur	NER	24.799971	93.950017	05/12/2020 00:00:00	2.5
16595	Meghalaya	NER	25.570492	91.880014	05/12/2020 00:00:00	5.8
16596	Mizoram	NER	23.710399	92.720015	05/12/2020 00:00:00	1.6
16597	Nagaland	NER	25.666998	94.116570	05/12/2020 00:00:00	2.1
16598	Tripura	NER	23.835404	91.279999	05/12/2020 00:00:00	3.3

16599 rows x 6 columns

```
In [28]: df2['Dates'] = pd.to_datetime(df2.Dates, dayfirst=True)
```