

In [1]:

```
import pandas as pd
import numpy as np
```

In [102]:

```
wt=pd.read_csv("Weather_Data.csv")
```

In [103]:

```
wt.head()
```

Out[103]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
0	1/1/2012 0:00	-1.8	-3.9	86	4	8.0	101.24	Fog
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fog
2	1/1/2012 2:00	-1.8	-3.4	89	7	4.0	101.26	Freezing Drizzle,Fog
3	1/1/2012 3:00	-1.5	-3.2	88	6	4.0	101.27	Freezing Drizzle,Fog
4	1/1/2012 4:00	-1.5	-3.3	88	7	4.8	101.23	Fog

In [104]:

```
wt.shape
```

Out[104]:

(8784, 8)

In [105]:

```
wt.index
```

Out[105]:

RangeIndex(start=0, stop=8784, step=1)

In [106]:

```
wt.dtypes
```

Out[106]:

```
Date/Time      object
Temp_C         float64
Dew Point Temp_C float64
Rel Hum_%      int64
Wind Speed_km/h int64
Visibility_km   float64
Press_kPa      float64
Weather        object
dtype: object
```

In [107]:

```
wt['Weather'].unique()
```

Out[107]:

```
array(['Fog', 'Freezing Drizzle,Fog', 'Mostly Cloudy', 'Cloudy', 'Rain',
      'Rain Showers', 'Mainly Clear', 'Snow Showers', 'Snow', 'Clear',
      'Freezing Rain,Fog', 'Freezing Rain', 'Freezing Drizzle',
      'Rain,Snow', 'Moderate Snow', 'Freezing Drizzle,Snow',
      'Freezing Rain,Snow Grains', 'Snow,Blowing Snow', 'Freezing Fog',
      'Haze', 'Rain,Fog', 'Drizzle,Fog', 'Drizzle',
      'Freezing Drizzle,Haze', 'Freezing Rain,Haze', 'Snow,Haze',
      'Snow,Fog', 'Snow,Ice Pellets', 'Rain,Haze', 'Thunderstorms,Rain',
      'Thunderstorms,Rain Showers', 'Thunderstorms,Heavy Rain Showers',
      'Thunderstorms,Rain Showers,Fog', 'Thunderstorms',
      'Thunderstorms,Rain,Fog',
      'Thunderstorms,Moderate Rain Showers,Fog', 'Rain Showers,Fog',
      'Rain Showers,Snow Showers', 'Snow Pellets', 'Rain,Snow,Fog',
      'Moderate Rain,Fog', 'Freezing Rain,Ice Pellets,Fog',
      'Drizzle,Ice Pellets,Fog', 'Drizzle,Snow', 'Rain,Ice Pellets',
      'Drizzle,Snow,Fog', 'Rain,Snow Grains', 'Rain,Snow,Ice Pellets',
      'Snow Showers,Fog', 'Moderate Snow,Blowing Snow'], dtype=object)
```

In [108]:

```
wt.nunique()
```

Out[108]:

```
Date/Time      8784
Temp_C         533
Dew Point Temp_C 489
Rel Hum_%      83
Wind Speed_km/h 34
Visibility_km   24
Press_kPa      518
Weather        50
dtype: int64
```

In [109]:

```
wt['Weather'].value_counts()
```

Out[109]:

Mainly Clear	2106
Mostly Cloudy	2069
Cloudy	1728
Clear	1326
Snow	390
Rain	306
Rain Showers	188
Fog	150
Rain,Fog	116
Drizzle,Fog	80
Snow Showers	60
Drizzle	41
Snow,Fog	37
Snow,Blowing Snow	19
Rain,Snow	18
Thunderstorms,Rain Showers	16
Haze	16
Drizzle,Snow,Fog	15
Freezing Rain	14
Freezing Drizzle,Snow	11
Freezing Drizzle	7
Snow,Ice Pellets	6
Freezing Drizzle,Fog	6
Snow,Haze	5
Freezing Fog	4
Snow Showers,Fog	4
Moderate Snow	4
Rain,Snow,Ice Pellets	4
Freezing Rain,Fog	4
Freezing Drizzle,Haze	3
Rain,Haze	3
Thunderstorms,Rain	3
Thunderstorms,Rain Showers,Fog	3
Freezing Rain,Haze	2
Drizzle,Snow	2
Rain Showers,Snow Showers	2
Thunderstorms	2
Moderate Snow,Blowing Snow	2
Rain Showers,Fog	1
Thunderstorms,Moderate Rain Showers,Fog	1
Snow Pellets	1
Rain,Snow,Fog	1
Moderate Rain,Fog	1
Freezing Rain,Ice Pellets,Fog	1
Drizzle,Ice Pellets,Fog	1
Thunderstorms,Rain,Fog	1
Rain,Ice Pellets	1
Rain,Snow Grains	1
Thunderstorms,Heavy Rain Showers	1
Freezing Rain,Snow Grains	1

Name: Weather, dtype: int64

In [110]:

wt.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8784 entries, 0 to 8783
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   Date/Time              8784 non-null   object
 1   Temp_C                 8784 non-null   float64
 2   Dew Point Temp_C       8784 non-null   float64
 3   Rel Hum_%              8784 non-null   int64
 4   Wind Speed_km/h        8784 non-null   int64
 5   Visibility_km           8784 non-null   float64
 6   Press_kPa              8784 non-null   float64
 7   Weather                8784 non-null   object
dtypes: float64(4), int64(2), object(2)
memory usage: 549.1+ KB
```

In [111]:

```
##Find out all the Null Values in the data.
wt.isnull().sum()
```

Out[111]:

```
Date/Time      0
Temp_C         0
Dew Point Temp_C  0
Rel Hum_%      0
Wind Speed_km/h 0
Visibility_km   0
Press_kPa      0
Weather        0
dtype: int64
```

In [112]:

```
## Find all the unique 'Wind Speed' values in the data.
wt['Wind Speed_km/h'].unique()
```

Out[112]:

```
array([ 4,  7,  6,  9, 15, 13, 20, 22, 19, 24, 30, 35, 39, 32, 33, 26, 44,
        43, 48, 37, 28, 17, 11,  0, 83, 70, 57, 46, 41, 52, 50, 63, 54,
        2],
      dtype=int64)
```

In [113]:

```
## Find the number of times when the 'Weather is exactly Clear'.  
  
wt.groupby('Weather').get_group('Clear')
```

Out[113]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
67	1/3/2012 19:00	-16.9	-24.8	50	24	25.0	101.74	Clear
114	1/5/2012 18:00	-7.1	-14.4	56	11	25.0	100.71	Clear
115	1/5/2012 19:00	-9.2	-15.4	61	7	25.0	100.80	Clear
116	1/5/2012 20:00	-9.8	-15.7	62	9	25.0	100.83	Clear
117	1/5/2012 21:00	-9.0	-14.8	63	13	25.0	100.83	Clear
...
8646	12/26/2012 6:00	-13.4	-14.8	89	4	25.0	102.47	Clear
8698	12/28/2012 10:00	-6.1	-8.6	82	19	24.1	101.27	Clear
8713	12/29/2012 1:00	-11.9	-13.6	87	11	25.0	101.31	Clear
8714	12/29/2012 2:00	-11.8	-13.1	90	13	25.0	101.33	Clear
8756	12/30/2012 20:00	-13.8	-16.5	80	24	25.0	101.52	Clear

1326 rows × 8 columns



In [114]:

```
##Find the number of times when the 'Wind Speed was exactly 4 km/h'.  
  
wt[wt['Wind Speed_km/h'] ==4]
```

Out[114]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
0	1/1/2012 0:00	-1.8	-3.9	86	4	8.0	101.24	Fog
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fog
96	1/5/2012 0:00	-8.8	-11.7	79	4	9.7	100.32	Snow
101	1/5/2012 5:00	-7.0	-9.5	82	4	4.0	100.19	Snow
146	1/7/2012 2:00	-8.1	-11.1	79	4	19.3	100.15	Cloudy
...
8768	12/31/2012 8:00	-8.6	-10.3	87	4	3.2	101.14	Snow Showers
8769	12/31/2012 9:00	-8.1	-9.6	89	4	2.4	101.09	Snow
8770	12/31/2012 10:00	-7.4	-8.9	89	4	6.4	101.05	Snow,Fog
8772	12/31/2012 12:00	-5.8	-7.5	88	4	12.9	100.78	Snow
8773	12/31/2012 13:00	-4.6	-6.6	86	4	12.9	100.63	Snow

474 rows × 8 columns



In [115]:

```
## Rename the column name 'Weather' of the dataframe to 'Weather Condition'.
wt.rename(columns= {'Weather': 'Weather Conduction'}, inplace =True)
wt
```

Out[115]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Condition
0	1/1/2012 0:00	-1.8	-3.9	86	4	8.0	101.24	Fc
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fc
2	1/1/2012 2:00	-1.8	-3.4	89	7	4.0	101.26	Freezir Drizzle,Fc
3	1/1/2012 3:00	-1.5	-3.2	88	6	4.0	101.27	Freezir Drizzle,Fc
4	1/1/2012 4:00	-1.5	-3.3	88	7	4.8	101.23	Fc
...
8779	12/31/2012 19:00	0.1	-2.7	81	30	9.7	100.13	Snc
8780	12/31/2012 20:00	0.2	-2.4	83	24	9.7	100.03	Snc
8781	12/31/2012 21:00	-0.5	-1.5	93	28	4.8	99.95	Snc
8782	12/31/2012 22:00	-0.2	-1.8	89	28	9.7	99.91	Snc
8783	12/31/2012 23:00	0.0	-2.1	86	30	11.3	99.89	Snc

8784 rows × 8 columns

In [116]:

```
## What is the mean 'Visibility' ?
wt['Visibility_km'].mean()
```

Out[116]:

27.66444672131151

In [117]:

```
##What is the Standard Deviation of 'Pressure' in this data?
wt.Press_kPa.std()
```

Out[117]:

0.8440047459486474

In [118]:

```
##Find all instances when 'Snow' was recorded.
```

```
wt.head(3)
```

Out[118]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Conduction
0	1/1/2012 0:00	-1.8	-3.9	86	4	8.0	101.24	Fog
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fog
2	1/1/2012 2:00	-1.8	-3.4	89	7	4.0	101.26	Freezing Drizzle,Fog

In [119]:

```
wt[wt['Weather Conduction']=='Snow']
```

Out[119]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weath Conductio
55	1/3/2012 7:00	-14.0	-19.5	63	19	25.0	100.95	Sno
84	1/4/2012 12:00	-13.7	-21.7	51	11	24.1	101.25	Sno
86	1/4/2012 14:00	-11.3	-19.0	53	7	19.3	100.97	Sno
87	1/4/2012 15:00	-10.2	-16.3	61	11	9.7	100.89	Sno
88	1/4/2012 16:00	-9.4	-15.5	61	13	19.3	100.79	Sno
...
8779	12/31/2012 19:00	0.1	-2.7	81	30	9.7	100.13	Sno
8780	12/31/2012 20:00	0.2	-2.4	83	24	9.7	100.03	Sno
8781	12/31/2012 21:00	-0.5	-1.5	93	28	4.8	99.95	Sno
8782	12/31/2012 22:00	-0.2	-1.8	89	28	9.7	99.91	Sno
8783	12/31/2012 23:00	0.0	-2.1	86	30	11.3	99.89	Sno

390 rows × 8 columns



In [120]:

```
## Find all instances when 'Wind Speed is above 24' and 'Visibility is 25'.
wt[(wt['Wind Speed_km/h'] > 24) & (wt['Visibility_km']==25)]
```

Out[120]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Conduction
23	1/1/2012 23:00	5.3	2.0	79	30	25.0	99.31	Cloud
24	1/2/2012 0:00	5.2	1.5	77	35	25.0	99.26	Ra Showe
25	1/2/2012 1:00	4.6	0.0	72	39	25.0	99.26	Cloud
26	1/2/2012 2:00	3.9	-0.9	71	32	25.0	99.26	Most Cloud
27	1/2/2012 3:00	3.7	-1.5	69	33	25.0	99.30	Most Cloud
...
8705	12/28/2012 17:00	-8.6	-12.0	76	26	25.0	101.34	Main Clea
8753	12/30/2012 17:00	-12.1	-15.8	74	28	25.0	101.26	Main Clea
8755	12/30/2012 19:00	-13.4	-16.5	77	26	25.0	101.47	Main Clea
8759	12/30/2012 23:00	-12.1	-15.1	78	28	25.0	101.52	Most Cloud
8760	12/31/2012 0:00	-11.1	-14.4	77	26	25.0	101.51	Cloud

308 rows × 8 columns

In [121]:

```
####Find all instances when :
#A. 'Weather is Clear' and 'Relative Humidity is greater than 50'
#or
#B. 'Visibility is above 40'####
wt.head(2)
```

Out[121]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Conduction
0	1/1/2012 0:00	-1.8	-3.9	86	4	8.0	101.24	Fog
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fog

In [122]:

```
wt[(wt['Weather Conduction']== 'Clear') & (wt['Rel Hum_%']> 50) | (wt['Visibility_km']>40)]
```

Out[122]:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Conduction
106	1/5/2012 10:00	-6.0	-10.0	73	17	48.3	100.45	Main Clear
107	1/5/2012 11:00	-5.6	-10.2	70	22	48.3	100.41	Main Clear
108	1/5/2012 12:00	-4.7	-9.6	69	20	48.3	100.38	Main Clear
109	1/5/2012 13:00	-4.4	-9.7	66	26	48.3	100.40	Main Clear
110	1/5/2012 14:00	-5.1	-10.7	65	22	48.3	100.46	Main Clear
...
8749	12/30/2012 13:00	-12.4	-16.2	73	37	48.3	100.92	Most Cloudy
8750	12/30/2012 14:00	-11.8	-16.1	70	37	48.3	100.96	Main Clear
8751	12/30/2012 15:00	-11.3	-15.6	70	32	48.3	101.05	Main Clear
8752	12/30/2012 16:00	-11.4	-15.5	72	26	48.3	101.15	Main Clear
8756	12/30/2012 20:00	-13.8	-16.5	80	24	25.0	101.52	Clear

2921 rows × 8 columns

In [124]:

```
bg=pd.read_csv('Bangalore_1990_2022_BangaloreCity.csv')
cn=pd.read_csv('Chennai_1990_2022_Madras.csv')
dl=pd.read_csv('Delhi_NCR_1990_2022_Safdarjung.csv')
lu=pd.read_csv('Lucknow_1990_2022.csv')
rg=pd.read_csv('Rajasthan_1990_2022_Jodhpur.csv')
mb=pd.read_csv('Mumbai_1990_2022_Santacruz.csv')
bw=pd.read_csv('weather_Bhubhneswar_1990_2022.csv')
ro=pd.read_csv('weather_Rourkela_2021_2022.csv')
```

In [125]:

```

bg['city-Name'] = 'Bangalore'
cn['city-Name'] = 'Chennai'
dl['city-Name'] = 'Delhi'
lu['city-Name'] = 'Lucknow'
mb['city-Name'] = 'Mumbai'
rg['city-Name'] = 'Rajasthan'
bw['city-Name'] = 'Bhubhneshwar'
ro['city-Name'] = 'Rourkela'

```

In [126]:

bg

Out[126]:

	time	tavg	tmin	tmax	prcp	city-Name
0	01-01-1990	22.9	19.1	28.4	NaN	Bangalore
1	02-01-1990	21.7	NaN	26.5	0.0	Bangalore
2	03-01-1990	21.0	16.4	26.5	0.0	Bangalore
3	04-01-1990	20.8	NaN	27.4	0.0	Bangalore
4	05-01-1990	20.4	14.2	26.1	0.0	Bangalore
...
11889	21-07-2022	23.7	20.5	30.8	82.5	Bangalore
11890	22-07-2022	23.2	21.1	27.9	0.0	Bangalore
11891	23-07-2022	23.1	20.9	26.7	0.0	Bangalore
11892	24-07-2022	22.8	20.0	26.7	0.3	Bangalore
11893	25-07-2022	24.1	20.2	28.5	0.5	Bangalore

11894 rows × 6 columns

In [127]:

```
cn
```

Out[127]:

	time	tavg	tmin	tmax	prcp	city-Name
0	01-01-1990	25.2	22.8	28.4	0.5	Chennai
1	02-01-1990	24.9	21.7	29.1	0.0	Chennai
2	03-01-1990	25.6	21.4	29.8	0.0	Chennai
3	04-01-1990	25.7	NaN	28.7	0.0	Chennai
4	05-01-1990	25.5	20.7	28.4	0.0	Chennai
...
11889	21-07-2022	28.4	24.5	32.8	21.1	Chennai
11890	22-07-2022	27.8	24.6	32.2	22.1	Chennai
11891	23-07-2022	27.4	24.7	32.6	18.6	Chennai
11892	24-07-2022	27.8	25.0	33.3	9.1	Chennai
11893	25-07-2022	28.1	25.4	32.6	2.9	Chennai

11894 rows × 6 columns

In [128]:

```
lu
```

Out[128]:

	time	tavg	tmin	tmax	prcp	city-Name
0	01-01-1990	7.2	NaN	18.1	0.0	Lucknow
1	02-01-1990	10.5	NaN	17.2	0.0	Lucknow
2	03-01-1990	10.2	1.8	18.6	NaN	Lucknow
3	04-01-1990	9.1	NaN	19.3	0.0	Lucknow
4	05-01-1990	13.5	NaN	23.8	0.0	Lucknow
...
11889	21-07-2022	27.4	25.1	33.1	27.3	Lucknow
11890	22-07-2022	28.1	26.1	31.1	16.0	Lucknow
11891	23-07-2022	30.3	26.2	34.7	11.9	Lucknow
11892	24-07-2022	30.0	28.1	34.7	2.0	Lucknow
11893	25-07-2022	27.1	24.1	34.3	0.5	Lucknow

11894 rows × 6 columns

In [129]:

```
d1
```

Out[129]:

	time	tavg	tmin	tmax	prcp	city-Name
0	01-01-1990	9.4	6.0	15.1	0.0	Delhi
1	02-01-1990	9.3	5.2	14.2	0.0	Delhi
2	03-01-1990	9.0	6.5	13.6	0.0	Delhi
3	04-01-1990	10.7	6.0	17.5	0.0	Delhi
4	05-01-1990	12.6	7.3	20.8	0.0	Delhi
...
11889	21-07-2022	28.6	26.8	30.6	21.2	Delhi
11890	22-07-2022	29.3	27.0	32.9	0.3	Delhi
11891	23-07-2022	30.1	25.5	34.9	8.9	Delhi
11892	24-07-2022	30.6	27.1	35.7	0.0	Delhi
11893	25-07-2022	30.7	26.8	35.7	0.0	Delhi

11894 rows × 6 columns

In [130]:

```
mb
```

Out[130]:

	time	tavg	tmin	tmax	prcp	city-Name
0	01-01-1990	23.2	17.0	NaN	0.0	Mumbai
1	02-01-1990	22.2	16.5	29.9	0.0	Mumbai
2	03-01-1990	21.8	16.3	30.7	0.0	Mumbai
3	04-01-1990	25.4	17.9	31.8	0.0	Mumbai
4	05-01-1990	26.5	19.3	33.7	0.0	Mumbai
...
11889	21-07-2022	27.6	25.6	30.5	10.9	Mumbai
11890	22-07-2022	28.3	26.0	30.5	3.0	Mumbai
11891	23-07-2022	28.2	25.8	31.3	5.1	Mumbai
11892	24-07-2022	28.1	25.6	30.4	7.1	Mumbai
11893	25-07-2022	28.3	25.1	30.2	7.1	Mumbai

11894 rows × 6 columns

In [131]:

```
ro
```

Out[131]:

	time	tavg	tmin	tmax	prcp	snow	wdir	wspd	wpgt	pres	tsun	city- Name
0	2021-07-06	29.3	26.2	32.6	NaN	NaN	197.0	6.8	NaN	1002.5	NaN	Rourkela
1	2021-07-07	29.7	27.3	33.4	11.1	NaN	199.0	6.9	NaN	1002.2	NaN	Rourkela
2	2021-07-08	27.4	25.8	29.7	66.9	NaN	186.0	6.3	NaN	1001.8	NaN	Rourkela
3	2021-07-09	28.5	26.1	32.1	11.4	NaN	173.0	3.9	NaN	1001.0	NaN	Rourkela
4	2021-07-10	29.0	26.2	32.6	2.7	NaN	121.0	4.6	NaN	1000.9	NaN	Rourkela
...
421	2022-08-31	29.0	26.6	33.5	2.0	NaN	187.0	8.6	NaN	1006.8	NaN	Rourkela
422	2022-09-01	29.1	25.7	33.2	11.5	NaN	205.0	6.7	NaN	1007.2	NaN	Rourkela
423	2022-09-02	29.4	26.4	33.7	1.5	NaN	189.0	7.0	NaN	1007.5	NaN	Rourkela
424	2022-09-03	28.7	26.6	32.6	8.0	NaN	203.0	8.0	NaN	1005.8	NaN	Rourkela
425	2022-09-04	28.2	25.9	31.8	17.7	NaN	211.0	6.8	NaN	1004.8	NaN	Rourkela

426 rows × 12 columns

In [132]:

```
ro.drop(['snow', 'wdir', 'wspd', 'wpgt', 'pres', 'tsun'],axis=1, inplace= True)
ro
```

Out[132]:

	time	tavg	tmin	tmax	prcp	city-Name
0	2021-07-06	29.3	26.2	32.6	NaN	Rourkela
1	2021-07-07	29.7	27.3	33.4	11.1	Rourkela
2	2021-07-08	27.4	25.8	29.7	66.9	Rourkela
3	2021-07-09	28.5	26.1	32.1	11.4	Rourkela
4	2021-07-10	29.0	26.2	32.6	2.7	Rourkela
...
421	2022-08-31	29.0	26.6	33.5	2.0	Rourkela
422	2022-09-01	29.1	25.7	33.2	11.5	Rourkela
423	2022-09-02	29.4	26.4	33.7	1.5	Rourkela
424	2022-09-03	28.7	26.6	32.6	8.0	Rourkela
425	2022-09-04	28.2	25.9	31.8	17.7	Rourkela

426 rows × 6 columns

In [133]:

bw

Out[133]:

	time	tavg	tmin	tmax	prcp	snow	wdir	wspd	wpgt	pres	tsun	city-Name
0	1990-01-01	20.1	NaN	28.0	0.0	NaN	NaN	NaN	NaN	NaN	NaN	Bhubhneshwar
1	1990-01-02	20.7	16.4	NaN	0.0	NaN	NaN	NaN	NaN	NaN	NaN	Bhubhneshwar
2	1990-01-03	20.7	16.0	27.4	0.0	NaN	NaN	NaN	NaN	NaN	NaN	Bhubhneshwar
3	1990-01-04	18.8	NaN	28.0	0.0	NaN	NaN	NaN	NaN	NaN	NaN	Bhubhneshwar
4	1990-01-05	19.8	11.0	28.2	0.0	NaN	NaN	NaN	NaN	NaN	NaN	Bhubhneshwar
...
11930	2022-08-31	29.2	26.3	33.0	9.0	NaN	186.0	8.2	NaN	1006.6	NaN	Bhubhneshwar
11931	2022-09-01	29.6	27.0	33.0	2.1	NaN	190.0	9.5	NaN	1006.8	NaN	Bhubhneshwar
11932	2022-09-02	29.7	26.3	33.0	3.3	NaN	198.0	9.5	NaN	1007.3	NaN	Bhubhneshwar
11933	2022-09-03	29.2	26.1	34.0	9.7	NaN	215.0	8.5	NaN	1005.5	NaN	Bhubhneshwar
11934	2022-09-04	27.6	25.9	31.6	12.8	NaN	214.0	8.6	NaN	1004.9	NaN	Bhubhneshwar

11935 rows × 12 columns



In [134]:

```
bw.drop(['snow', 'wdir', 'wspd', 'wpgt', 'pres', 'tsun'],axis=1, inplace= True)
bw
```

Out[134]:

	time	tavg	tmin	tmax	prcp	city-Name
0	1990-01-01	20.1	NaN	28.0	0.0	Bhubhneshwar
1	1990-01-02	20.7	16.4	NaN	0.0	Bhubhneshwar
2	1990-01-03	20.7	16.0	27.4	0.0	Bhubhneshwar
3	1990-01-04	18.8	NaN	28.0	0.0	Bhubhneshwar
4	1990-01-05	19.8	11.0	28.2	0.0	Bhubhneshwar
...
11930	2022-08-31	29.2	26.3	33.0	9.0	Bhubhneshwar
11931	2022-09-01	29.6	27.0	33.0	2.1	Bhubhneshwar
11932	2022-09-02	29.7	26.3	33.0	3.3	Bhubhneshwar
11933	2022-09-03	29.2	26.1	34.0	9.7	Bhubhneshwar
11934	2022-09-04	27.6	25.9	31.6	12.8	Bhubhneshwar

11935 rows × 6 columns

In [135]:

```
rg
```

Out[135]:

	time	tavg	tmin	tmax	prcp	city-Name
0	01-01-1990	22.9	19.1	28.4	NaN	Rajasthan
1	02-01-1990	21.7	NaN	26.5	0.0	Rajasthan
2	03-01-1990	21.0	16.4	26.5	0.0	Rajasthan
3	04-01-1990	20.8	NaN	27.4	0.0	Rajasthan
4	05-01-1990	20.4	14.2	26.1	0.0	Rajasthan
...
11889	21-07-2022	23.7	20.5	30.8	82.5	Rajasthan
11890	22-07-2022	23.2	21.1	27.9	0.0	Rajasthan
11891	23-07-2022	23.1	20.9	26.7	0.0	Rajasthan
11892	24-07-2022	22.8	20.0	26.7	0.3	Rajasthan
11893	25-07-2022	24.1	20.2	28.5	0.5	Rajasthan

11894 rows × 6 columns

In [136]:

```
def clean(df): ## ffill ffill: propagate last valid observation forward to next valid.
    df['prcp'].fillna(method="ffill",inplace=True)
    df['tmin'].fillna(method="ffill",inplace=True)
    df['tmax'].fillna(method="ffill",inplace=True)
    return df
```

In [137]:

```
bg = clean(bg)
cn = clean(cn)
dl = clean(dl)
lu= clean(lu)
mb= clean(mb)
rg= clean(rg)
bw= clean(bw)
ro= clean(ro) #/call the class of clean
```

In [138]:

```
cn['target']=cn.shift(-1)['tmax']
bg['target']=bg.shift(-1)['tmax']
dl['target']=dl.shift(-1)['tmax']
lu['target']=lu.shift(-1)['tmax']
mb['target']=mb.shift(-1)['tmax']
rg['target']=rg.shift(-1)['tmax']
bw['target']=bw.shift(-1)['tmax']
ro['target']=ro.shift(-1)['tmax']
```

In [139]:

```
cities = [cn,bg,dl,lu,mb,rg,bw,ro]
df = pd.concat(cities)
```

In [140]:

```
df.sample(10)
```

Out[140]:

	time	tavg	tmin	tmax	prcp	city-Name	target
9488	24-12-2015	13.2	4.3	21.9	0.0	Lucknow	21.9
4790	12-02-2003	24.9	18.3	32.2	0.0	Bangalore	32.3
10321	05-04-2018	30.4	20.7	38.8	0.0	Lucknow	36.8
5062	2003-11-11	24.0	21.1	31.0	0.0	Bhubhneshwar	29.7
4874	07-05-2003	33.0	28.6	40.7	0.0	Chennai	40.7
858	08-05-1992	25.3	22.6	32.2	0.0	Bangalore	34.2
7277	04-12-2009	26.9	20.0	31.6	0.0	Mumbai	31.6
10007	26-05-2017	26.2	20.9	33.3	0.0	Rajasthan	33.1
2238	17-02-1996	17.4	10.7	24.0	0.0	Delhi	25.6
1349	11-09-1993	24.7	19.7	29.6	0.0	Bangalore	28.8

In [141]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 83725 entries, 0 to 425
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
---  -
 0   time        83725 non-null  object  
 1   tavg        83235 non-null  float64
 2   tmin        83722 non-null  float64
 3   tmax        83724 non-null  float64
 4   prcp        83722 non-null  float64
 5   city-Name    83725 non-null  object  
 6   target      83717 non-null  float64
dtypes: float64(5), object(2)
memory usage: 5.1+ MB
```

In [142]:

```
df.isnull().sum()
```

Out[142]:

```
time          0
tavg          490
tmin           3
tmax           1
prcp           3
city-Name      0
target         8
dtype: int64
```

In [143]:

```
df.shape
```

Out[143]:

```
(83725, 7)
```

In [144]:

```
df.index
```

Out[144]:

```
Int64Index([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9,
            ...,
            416, 417, 418, 419, 420, 421, 422, 423, 424, 425],
            dtype='int64', length=83725)
```

In [145]:

```
df.isnull().sum()
```

Out[145]:

```
time          0
tavg          490
tmin           3
tmax           1
prcp           3
city-Name      0
target         8
dtype: int64
```

In [146]:

```
df.reset_index()
```

Out[146]:

	index	time	tavg	tmin	tmax	prcp	city-Name	target
0	0	01-01-1990	25.2	22.8	28.4	0.5	Chennai	29.1
1	1	02-01-1990	24.9	21.7	29.1	0.0	Chennai	29.8
2	2	03-01-1990	25.6	21.4	29.8	0.0	Chennai	28.7
3	3	04-01-1990	25.7	21.4	28.7	0.0	Chennai	28.4
4	4	05-01-1990	25.5	20.7	28.4	0.0	Chennai	26.1
...
83720	421	2022-08-31	29.0	26.6	33.5	2.0	Rourkela	33.2
83721	422	2022-09-01	29.1	25.7	33.2	11.5	Rourkela	33.7
83722	423	2022-09-02	29.4	26.4	33.7	1.5	Rourkela	32.6
83723	424	2022-09-03	28.7	26.6	32.6	8.0	Rourkela	31.8
83724	425	2022-09-04	28.2	25.9	31.8	17.7	Rourkela	NaN

83725 rows × 8 columns

In [147]:

```
df['city-Name']=df['city-Name'].map({'Chennai': '0', 'Bangalore': '1', 'Delhi': '2', 'Luckno
```

In [148]:

```
df
```

Out[148]:

	time	tavg	tmin	tmax	prcp	city-Name	target
0	01-01-1990	25.2	22.8	28.4	0.5	0	29.1
1	02-01-1990	24.9	21.7	29.1	0.0	0	29.8
2	03-01-1990	25.6	21.4	29.8	0.0	0	28.7
3	04-01-1990	25.7	21.4	28.7	0.0	0	28.4
4	05-01-1990	25.5	20.7	28.4	0.0	0	26.1
...
421	2022-08-31	29.0	26.6	33.5	2.0	7	33.2
422	2022-09-01	29.1	25.7	33.2	11.5	7	33.7
423	2022-09-02	29.4	26.4	33.7	1.5	7	32.6
424	2022-09-03	28.7	26.6	32.6	8.0	7	31.8
425	2022-09-04	28.2	25.9	31.8	17.7	7	NaN

83725 rows × 7 columns

In [149]:

```
df.dtypes
```

Out[149]:

```
time          object
tavg          float64
tmin          float64
tmax          float64
prcp          float64
city-Name     object
target        float64
dtype: object
```

In [150]:

```
df.set_index('time',inplace=True)
```

In [151]:

```
df.sort_values(by='time')
```

Out[151]:

	tavg	tmin	tmax	prcp	city-Name	target
time						
01-01-1990	25.2	22.8	28.4	0.5	0	29.1
01-01-1990	7.2	NaN	18.1	0.0	3	17.2
01-01-1990	23.2	17.0	NaN	0.0	4	29.9
01-01-1990	22.9	19.1	28.4	NaN	1	26.5
01-01-1990	22.9	19.1	28.4	NaN	5	26.5
...
31-12-2021	20.3	17.6	26.5	0.0	5	24.6
31-12-2021	13.9	12.1	20.5	0.0	3	20.7
31-12-2021	10.9	4.6	18.2	0.0	2	20.0
31-12-2021	26.1	22.0	31.4	0.0	4	32.0
31-12-2021	20.3	17.6	26.5	0.0	1	24.6

83725 rows × 6 columns

In [152]:

```
df.index
```

Out[152]:

```
Index(['01-01-1990', '02-01-1990', '03-01-1990', '04-01-1990', '05-01-1990',
      '06-01-1990', '07-01-1990', '08-01-1990', '09-01-1990', '10-01-1990',
      ...,
      '2022-08-26', '2022-08-27', '2022-08-28', '2022-08-29', '2022-08-30',
      '2022-08-31', '2022-09-01', '2022-09-02', '2022-09-03', '2022-09-04'],
      dtype='object', name='time', length=83725)
```

In [153]:

```
df.index= pd.to_datetime(df.index)
df.dtypes
Specify infer_datetime_format=True for consistent parsing.
df.index= pd.to_datetime(df.index)
C:\Users\sRakesh\AppData\Local\Temp\ipykernel_13600\2246822679.py:1: Use
rWarning: Parsing '26-01-1991' in DD/MM/YYYY format. Provide format or s
pecify infer_datetime_format=True for consistent parsing.
df.index= pd.to_datetime(df.index)
C:\Users\sRakesh\AppData\Local\Temp\ipykernel_13600\2246822679.py:1: Use
rWarning: Parsing '27-01-1991' in DD/MM/YYYY format. Provide format or s
pecify infer_datetime_format=True for consistent parsing.
df.index= pd.to_datetime(df.index)
C:\Users\sRakesh\AppData\Local\Temp\ipykernel_13600\2246822679.py:1: Use
rWarning: Parsing '28-01-1991' in DD/MM/YYYY format. Provide format or s
pecify infer_datetime_format=True for consistent parsing.
df.index= pd.to_datetime(df.index)
C:\Users\sRakesh\AppData\Local\Temp\ipykernel_13600\2246822679.py:1: Use
rWarning: Parsing '29-01-1991' in DD/MM/YYYY format. Provide format or s
pecify infer_datetime_format=True for consistent parsing.
df.index= pd.to_datetime(df.index)
C:\Users\sRakesh\AppData\Local\Temp\ipykernel_13600\2246822679.py:1: Use
rWarning: Parsing '30-01-1991' in DD/MM/YYYY format. Provide format or s
pecify infer_datetime_format=True for consistent parsing.
```

In [154]:

```
df['city-Name']=df['city-Name'].astype(str).astype(int)
```

In [58]:

```
df.index.year
```

Out[58]:

```
Int64Index([1990, 1990, 1990, 1990, 1990, 1990, 1990, 1990, 1990, 1990,
...
2022, 2022, 2022, 2022, 2022, 2022, 2022, 2022, 2022, 2022],
dtype='int64', name='time', length=83725)
```

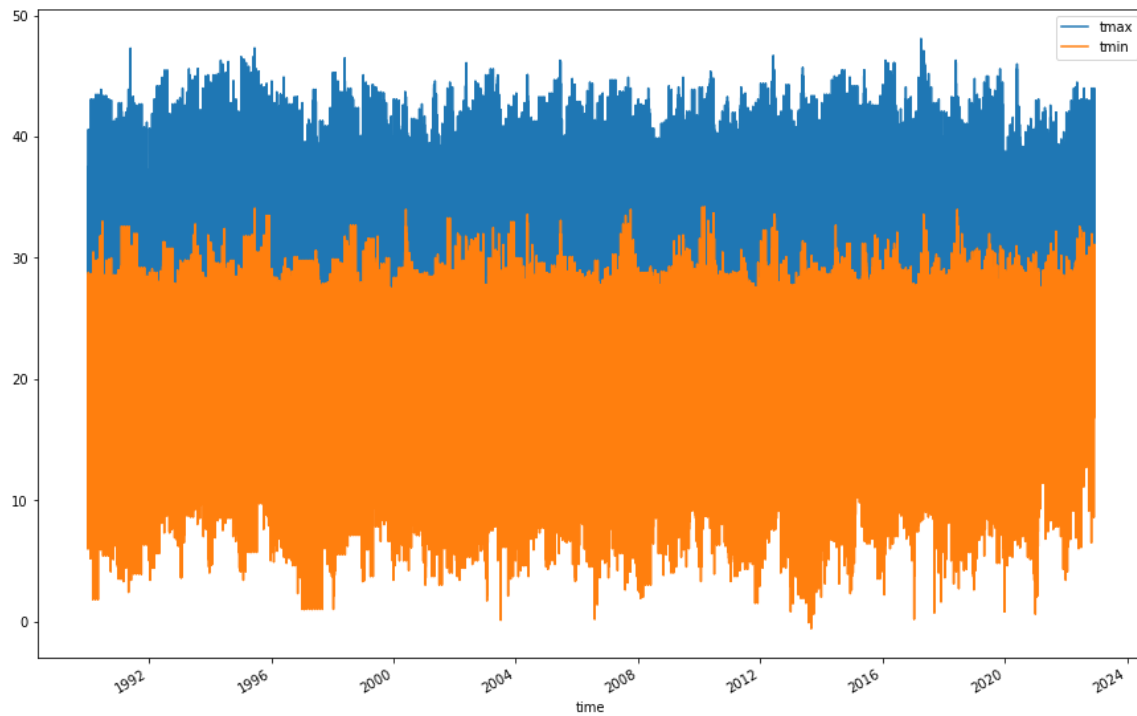
In [59]:

```
import matplotlib as mlt
import seaborn as scn

df[['tmax', 'tmin']].plot(figsize=(15,10))
```

Out[59]:

<AxesSubplot:xlabel='time'>

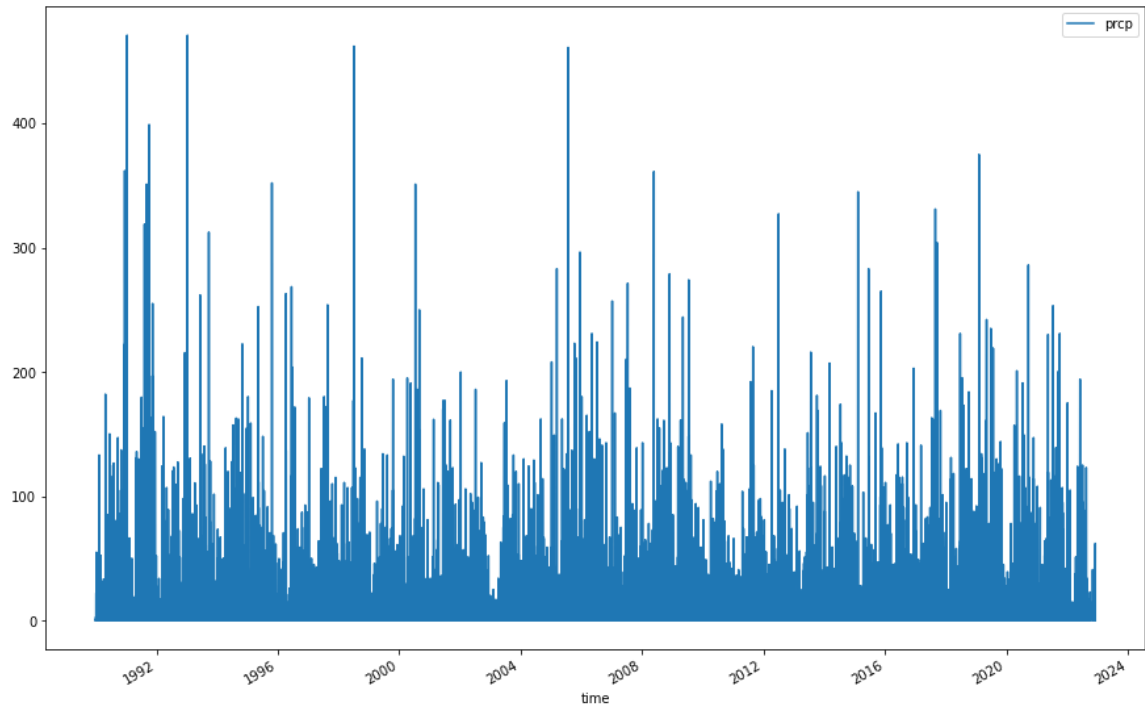


In [61]:

```
df[['prcp']].plot(figsize=(15,10))
```

Out[61]:

<AxesSubplot:xlabel='time'>



In [62]:

```
df
```

Out[62]:

	tavg	tmin	tmax	prcp	city-Name	target
time						
1990-01-01	25.2	22.8	28.4	0.5	0	29.1
1990-02-01	24.9	21.7	29.1	0.0	0	29.8
1990-03-01	25.6	21.4	29.8	0.0	0	28.7
1990-04-01	25.7	21.4	28.7	0.0	0	28.4
1990-05-01	25.5	20.7	28.4	0.0	0	26.1
...
2022-08-31	29.0	26.6	33.5	2.0	7	33.2
2022-09-01	29.1	25.7	33.2	11.5	7	33.7
2022-09-02	29.4	26.4	33.7	1.5	7	32.6
2022-09-03	28.7	26.6	32.6	8.0	7	31.8
2022-09-04	28.2	25.9	31.8	17.7	7	NaN

83725 rows × 6 columns

In [63]:

```
df=df.iloc[:-1,:].copy()
df
```

Out[63]:

	tavg	tmin	tmax	prcp	city-Name	target
time						
1990-01-01	25.2	22.8	28.4	0.5	0	29.1
1990-02-01	24.9	21.7	29.1	0.0	0	29.8
1990-03-01	25.6	21.4	29.8	0.0	0	28.7
1990-04-01	25.7	21.4	28.7	0.0	0	28.4
1990-05-01	25.5	20.7	28.4	0.0	0	26.1
...
2022-08-30	29.8	26.4	34.3	0.0	7	33.5
2022-08-31	29.0	26.6	33.5	2.0	7	33.2
2022-09-01	29.1	25.7	33.2	11.5	7	33.7
2022-09-02	29.4	26.4	33.7	1.5	7	32.6
2022-09-03	28.7	26.6	32.6	8.0	7	31.8

83724 rows × 6 columns

In [64]:

```
df.drop(['tavg'],axis=1,inplace=True)
```

In [65]:

```
df.dropna(inplace=True)
```

In [67]:

```
from sklearn.linear_model import Ridge

reg = Ridge(alpha=.1)
predictors = ["prcp", "tmax", "tmin","city-Name"]
train=df.loc[:'2020-01-01']
test=df.loc['2020-01-01':]
```

In [68]:

```
train[predictors]
```

Out[68]:

	prcp	tmax	tmin	city-Name
time				
1990-01-01	0.5	28.4	22.8	0
1990-02-01	0.0	29.1	21.7	0
1990-03-01	0.0	29.8	21.4	0
1990-04-01	0.0	28.7	21.4	0
1990-05-01	0.0	28.4	20.7	0
...
2019-12-28	0.0	24.0	14.0	6
2019-12-29	0.0	24.6	10.2	6
2019-12-30	0.0	24.0	10.6	6
2019-12-31	0.0	25.8	9.8	6
2020-01-01	0.0	26.6	12.6	6

76700 rows × 4 columns

In [69]:

```
reg.fit(train[predictors], train["target"])
```

Out[69]:

Ridge(alpha=0.1)

In [70]:

```
predictions = reg.predict(test[predictors])
```

In [71]:

```
from sklearn.metrics import mean_squared_error
mean_squared_error(test["target"], predictions)
```

Out[71]:

2.187583661270361

In [72]:

```
combined = pd.concat([test["target"], pd.Series(predictions, index=test.index)], axis=1)
combined.columns = ["actual", "predictions"]
combined
```

Out[72]:

	actual	predictions
time		
2020-01-01	31.0	30.585713
2020-02-01	31.2	31.271966
2020-03-01	31.2	31.400053
2020-04-01	31.6	31.421375
2020-05-01	31.0	31.814453
...
2022-08-30	33.5	34.295601
2022-08-31	33.2	33.582470
2022-09-01	33.7	33.294445
2022-09-02	32.6	33.754356
2022-09-03	31.8	32.779593

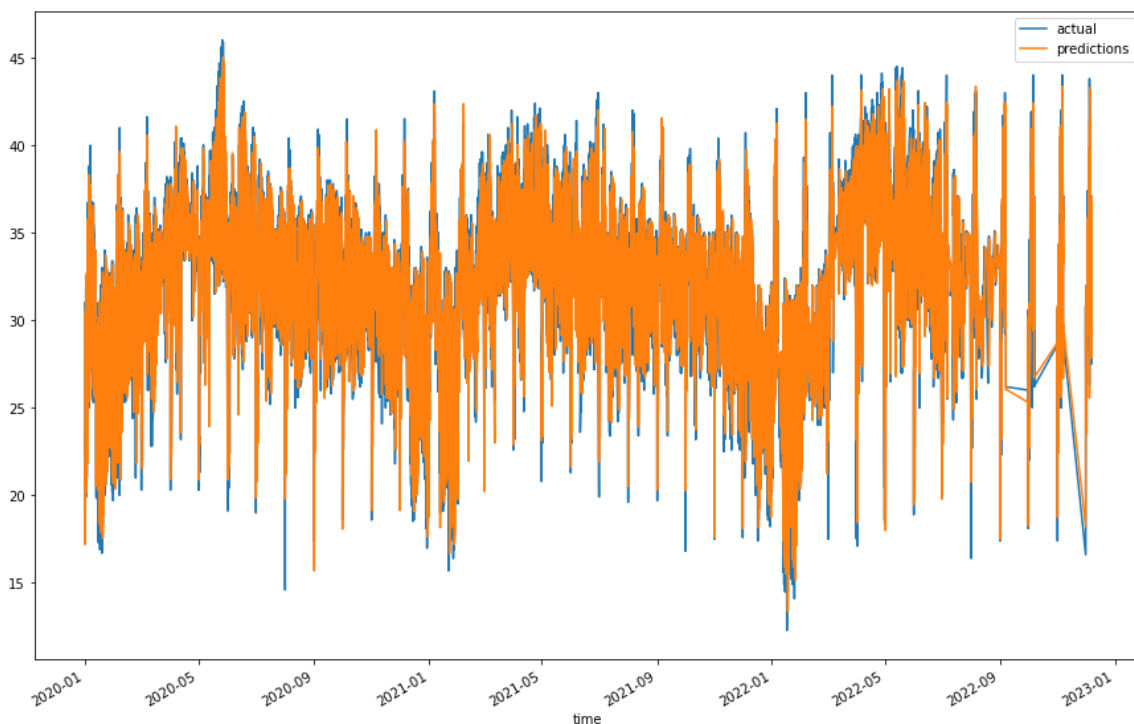
7017 rows × 2 columns

In [73]:

```
combined.plot(figsize=(15,10))
```

Out[73]:

<AxesSubplot:xlabel='time'>



In [159]:

```
Requirement already satisfied: pandoc in c:\users\sakesh\anaconda3\lib\site-packages (2.3)
Requirement already satisfied: ply in c:\users\sakesh\anaconda3\lib\site-packages (from pandoc) (3.11)
Requirement already satisfied: plumbum in c:\users\sakesh\anaconda3\lib\site-packages (from pandoc) (1.8.2)
Requirement already satisfied: pywin32 in c:\users\sakesh\anaconda3\lib\site-packages (from plumbum->pandoc) (302)
Note: you may need to restart the kernel to use updated packages.
```

In []: