DOP: / /2023   DOS: / / 2023

Experiment No: 08

Aim: - Design and Implement IoT Data Processing using Pandas.

Theory:

What is Pandas?

Pandas is an open-source Python library used for data manipulation, analysis, and visualization. It provides data structures for efficiently storing and manipulating large datasets, and a wide range of functions for data processing and analysis.

The main data structures in pandas are the Series (one-dimensional labeled array) and Data Frame (two-dimensional labeled data structure with columns of potentially different types). Pandas also provides tools for working with time series data, handling missing data, and merging, grouping, and reshaping datasets.

Pandas is widely used in data science and analytics projects, as it offers a convenient and powerful way to work with tabular data. It can be used in conjunction with other Python libraries such as NumPy, Matplotlib, and Scikit-learn for more complex data analysis and modeling tasks.

```
[1] import pandas as pd

[3] from google.colab import files
    uploaded = files.upload()

    Choose Files   WeatherData.csv
    • WeatherData.csv(text/csv) - 462623 bytes, last modified: 2/28/2023 - 100% done
    Saving WeatherData.csv to WeatherData.csv

[4] data = pd.read_csv("WeatherData.csv")

[ ] data
```

	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

Head shows the first N rows in the data (by default, N=5).

```
[5] data.head()
```

	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

Shape shows the total no. of rows and no. of columns of the dataframe

```
[6] data.shape

    (8784, 8)
```

Index attribute provides the index of the dataframe

```
[ ] data.index
```

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

Columns shows the name of each column

```
data.columns
```

```
Index(['Date/Time', 'Temp_C', 'Dew Point Temp_C', 'Rel Hum_%',
       'Wind Speed_km/h', 'Visibility_km', 'Press_kPa', 'Weather'],
      dtype='object')
```

dtypes shows the data-type of each column

```
[ ] data.dtypes
```

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

unique() - In a column, it shows all the unique values. It can be applied on a single column only, not on the whole dataframe.

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

```
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)
```

nunique() - It shows the total no. of unique values in each column. It can be applied on a single column as well as on the whole dataframe.

```
data.nunique()
```

```
--NORMAL--
```

```
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
```

count - It shows the total no. of non-null values in each column. It can be applied on a single column as well as on the whole dataframe.

```
[ ] data.count()
```

```
Date/Time           8784
Temp_C              8784
Dew Point Temp_C    8784
Rel Hum_%           8784
Wind Speed_km/h     8784
Visibility_km       8784
Press_kPa           8784
```

value_counts - In a column, it shows all the unique values with their count. It can be applied on a single column only.

```
[ ]  data.value_counts
```

```
<bound method DataFrame.value_counts of              Date/Time  Temp_C  Dew Point Temp_C  Rel Hum_%  Wind Speed_km/h
0            1/1/2012 0:00    -1.8              -3.9         86                 4
1            1/1/2012 1:00    -1.8              -3.7         87                 4
2            1/1/2012 2:00    -1.8              -3.4         89                 7
3            1/1/2012 3:00    -1.5              -3.2         88                 6
4            1/1/2012 4:00    -1.5              -3.3         88                 7
...                    ...     ...               ...        ...               ...
8779     12/31/2012 19:00     0.1              -2.7         81                30
8780     12/31/2012 20:00     0.2              -2.4         83                24
8781     12/31/2012 21:00    -0.5              -1.5         93                28
8782     12/31/2012 22:00    -0.2              -1.8         89                28
8783     12/31/2012 23:00     0.0              -2.1         86                30

       Visibility_km  Press_kPa              Weather
0                8.0     101.24                  Fog
1                8.0     101.24                  Fog
2                4.0     101.26  Freezing Drizzle,Fog
3                4.0     101.27  Freezing Drizzle,Fog
4                4.8     101.23                  Fog
...              ...        ...                  ...
8779             9.7     100.13                 Snow
8780             9.7     100.03                 Snow
```

info() - Provides basic information about the dataframe.

```
    data.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
```

Q. 1) Find all the unique 'Wind Speed' values in the data.

```
[ ]  data.head(2)
```

	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

Double-click (or enter) to edit

```
    data.nunique()
```

```
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
```

```
[ ]  data['Wind Speed_km/h'].nunique()
```

```
34
```

```
[ ]  data['Wind Speed_km/h'].unique() #answer
```

```
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])
```

Q. 2) Find the number of times when the 'Weather is exactly Clear

```
data.groupby('Weather').get_group('Clear') #answer
```

	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

Q. 4) Find out all the Null Values in the data.

```
data.isnull()
```

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
8779	False	False	False	False	False	False	False	False
8780	False	False	False	False	False	False	False	False
8781	False	False	False	False	False	False	False	False
8782	False	False	False	False	False	False	False	False
8783	False	False	False	False	False	False	False	False

8784 rows × 8 columns

Q. 5) Rename the column name 'Weather' of the dataframe to 'Weather Condition'.

```
data.rename(columns={'Weather':'Weather Condition'})
```

	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	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
...	...	...	...	...	...	...	...	...
8779	12/31/2012 19:00	0.1	-2.7	81	30	9.7	100.13	Snow
8780	12/31/2012 20:00	0.2	-2.4	83	24	9.7	100.03	Snow
8781	12/31/2012 21:00	-0.5	-1.5	93	28	4.8	99.95	Snow
8782	12/31/2012 22:00	-0.2	-1.8	89	28	9.7	99.91	Snow
8783	12/31/2012 23:00	0.0	-2.1	86	30	11.3	99.89	Snow

8784 rows × 8 columns

Q6 What is the mean 'Visibility' ?

```
[ ]  data.Visibility_km.mean()
```

```
27.664446721311478
```

mean function gives mean of the value

Q. 7) What is the Standard Deviation of 'Pressure' in this data?

.std--> Gives the standard Deviation of the column required

```
[ ]  data.Press_kPa.std()
```

```
0.8440047459486483
```

```
[ ]  data.head(2)
```

	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	Fog
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fog

Q. 8) What is the Variance of 'Relative Humidity' in this data ?

```
[ ]  data.head(2)
```

	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	Fog
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fog

```
[ ]  data['Rel Hum_%'].var()
```

```
286.24855019850196
```

.var gives the variance of the columns

Q. 9) Find all instances when 'Snow' was recorded.

```
[ ]  data[data['Weather Condition']=='Snow']
```

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

390 rows × 8 columns

Q. 10) Find all instances when 'Wind Speed is above 24' and 'Visibility is 25'.

```
data[(data['Wind Speed_km/h']>24)&(data['Visibility_km']==25)]
```

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

Q. 11) What is the Mean value of each column against each 'Weather Condition ?

```
data.groupby('Weather Condition').mean()
```

Weather Condition	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa
Clear	6.825716	0.089367	64.497738	10.557315	30.153243	101.587443
Cloudy	7.970544	2.375810	69.592593	16.127315	26.625752	100.911441
Drizzle	7.353659	5.504878	88.243902	16.097561	17.931707	100.435366
Drizzle,Fog	8.067500	7.033750	93.275000	11.862500	5.257500	100.786625
Drizzle,Ice Pellets,Fog	0.400000	-0.700000	92.000000	20.000000	4.000000	100.790000
Drizzle,Snow	1.050000	0.150000	93.500000	14.000000	10.500000	100.890000
Drizzle,Snow,Fog	0.693333	0.120000	95.866667	15.533333	5.513333	99.281333
Fog	4.303333	3.159333	92.286667	7.946667	6.248000	101.184067
Freezing Drizzle	-5.657143	-8.000000	83.571429	16.571429	9.200000	100.202857
Freezing Drizzle,Fog	-2.533333	-4.183333	88.500000	17.000000	5.266667	100.441667
Freezing Drizzle,Haze	-5.433333	-8.000000	82.000000	10.333333	2.666667	100.316667

Q. 12) What is the Minimum & Maximum value of each column against each 'Weather Condition ?

```
data.groupby('Weather Condition').min()
```

Haze	1/22/2012 12:00	-11.5	-16.0	68	0	4.0	100.55
Mainly Clear	1/10/2012 11:00	-22.8	-28.0	20	0	12.9	98.67
Moderate Rain,Fog	12/10/2012 8:00	1.7	0.8	94	17	6.4	99.98
Moderate Snow	1/12/2012 15:00	-6.3	-7.6	83	26	0.6	99.88
Moderate Snow,Blowing Snow	12/27/2012 10:00	-5.5	-6.6	92	39	0.6	100.50
Mostly Cloudy	1/1/2012 16:00	-23.2	-28.5	18	0	11.3	98.36
Rain	1/1/2012 18:00	0.3	-5.7	40	0	4.0	97.52
Rain Showers	1/1/2012 22:00	1.6	-7.2	37	0	6.4	98.51
Rain Showers,Fog	10/20/2012 3:00	12.8	12.1	96	13	6.4	99.83
Rain Showers,Snow Showers	11/4/2012 8:00	2.1	-1.8	75	17	19.3	101.09
Rain,Fog	1/23/2012 18:00	0.0	-1.2	83	0	2.0	98.61
Rain,Haze	3/13/2012 7:00	4.0	1.0	81	7	4.0	100.50
Rain,Ice Pellets	12/18/2012 5:00	0.6	-0.6	92	24	9.7	100.12
Rain,Snow	1/10/2012 5:00	0.6	-1.7	81	13	2.4	98.18
Rain,Snow Grains	12/21/2012 0:00	1.9	-2.1	75	26	25.0	100.60

```
data.groupby('Weather Condition').max()
```

Weather Condition	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa
Clear	9/9/2012 5:00	32.8	20.4	99	33	48.3	103.63
Cloudy	9/9/2012 23:00	30.5	22.6	99	54	48.3	103.65
Drizzle	9/30/2012 3:00	18.8	17.7	96	30	25.0	101.56
Drizzle,Fog	9/30/2012 2:00	19.9	19.1	100	28	9.7	102.07
Drizzle,Ice Pellets,Fog	12/17/2012 9:00	0.4	-0.7	92	20	4.0	100.79
Drizzle,Snow	12/19/2012 18:00	1.2	0.2	95	19	11.3	101.15
Drizzle,Snow,Fog	12/22/2012 3:00	1.1	0.6	98	32	9.7	100.15
Fog	9/22/2012 0:00	20.8	19.6	100	22	9.7	103.04
Freezing Drizzle	2/1/2012 5:00	-2.3	-3.3	93	26	12.9	101.02
Freezing Drizzle,Fog	12/10/2012 5:00	-0.3	-2.3	94	33	8.0	101.27
Freezing Drizzle,Haze	2/1/2012 13:00	-5.0	-7.7	83	11	4.0	100.36
Freezing Drizzle,Snow	3/2/2012 12:00	-3.3	-4.6	94	24	12.9	101.18
Freezing Fog	3/17/2012 6:00	-0.1	-0.3	99	9	0.8	102.85
Freezing Rain	2/1/2012 7:00	0.3	-1.7	92	28	16.1	101.00

Q. 13) Show all the Records where Weather Condition is Fog.

```
data[data['Weather Condition']=='Fog']
```

	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	Fog
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fog
4	1/1/2012 4:00	-1.5	-3.3	88	7	4.8	101.23	Fog
5	1/1/2012 5:00	-1.4	-3.3	87	9	6.4	101.27	Fog
6	1/1/2012 6:00	-1.5	-3.1	89	7	6.4	101.29	Fog
...	...	...	...	...	...	...	...	...
8716	12/29/2012 4:00	-16.0	-17.2	90	6	9.7	101.25	Fog
8717	12/29/2012 5:00	-14.8	-15.9	91	4	6.4	101.25	Fog
8718	12/29/2012 6:00	-13.8	-15.3	88	4	9.7	101.25	Fog
8719	12/29/2012 7:00	-14.8	-16.4	88	7	8.0	101.22	Fog
8722	12/29/2012 10:00	-12.0	-13.3	90	7	6.4	101.15	Fog

150 rows × 8 columns

Q. 14) Find all instances when 'Weather is Clear' or 'Visibility is above 40'.

```
data[(data['Weather Condition']=='Clear')&(data['Visibility_km']>40)]
```

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Condition
351	1/15/2012 15:00	-15.4	-22.8	53	24	48.3	102.71	Clear
352	1/15/2012 16:00	-15.1	-22.8	52	24	48.3	102.79	Clear
425	1/18/2012 17:00	-11.3	-18.8	54	26	48.3	101.54	Clear
440	1/19/2012 8:00	-13.7	-18.4	68	19	48.3	101.84	Clear
441	1/19/2012 9:00	-12.7	-17.2	69	17	48.3	101.73	Clear
...	...	...	...	...	...	...	...	...
8384	12/15/2012 8:00	-10.7	-15.6	67	13	48.3	102.69	Clear
8385	12/15/2012 9:00	-10.4	-15.9	64	19	48.3	102.74	Clear
8389	12/15/2012 13:00	-8.4	-14.7	60	19	48.3	102.64	Clear
8631	12/25/2012 15:00	-7.1	-13.7	59	17	48.3	101.98	Clear
8632	12/25/2012 16:00	-7.5	-13.9	60	11	48.3	102.03	Clear

313 rows × 8 columns

Q. 15) Find all instances when : A. 'Weather is Clear' and 'Relative Humidity is greater than 50' or B. 'Visibility is above 40'

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

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Condition
67	1/3/2012 19:00	-16.9	-24.8	50	24	25.0	101.74	Clear
106	1/5/2012 10:00	-6.0	-10.0	73	17	48.3	100.45	Mainly Clear
107	1/5/2012 11:00	-5.6	-10.2	70	22	48.3	100.41	Mainly Clear
108	1/5/2012 12:00	-4.7	-9.6	69	20	48.3	100.38	Mainly Clear
109	1/5/2012 13:00	-4.4	-9.7	66	26	48.3	100.40	Mainly Clear
...	...	...	...	...	...	...	...	...
8748	12/30/2012 12:00	-12.2	-15.7	75	26	48.3	100.91	Mostly Cloudy
8749	12/30/2012 13:00	-12.4	-16.2	73	37	48.3	100.92	Mostly Cloudy
8750	12/30/2012 14:00	-11.8	-16.1	70	37	48.3	100.96	Mainly Clear
8751	12/30/2012 15:00	-11.3	-15.6	70	32	48.3	101.05	Mainly Clear
8752	12/30/2012 16:00	-11.4	-15.5	72	26	48.3	101.15	Mainly Clear

Conclusion:

Pandas is a powerful and widely-used Python library for data manipulation, analysis, and visualization. Its data structures and functions allow for efficient handling of large datasets and provide a wide range of tools for data processing, manipulation, and analysis. Pandas is a must-have library for anyone working with tabular data in Python and is commonly used in data science and analytics projects.