

Report on weather

The data that we have collected should be understandable by the python to do analysis so that we have to load the data in the python IDE using the syntax,
`data=pd.read_csv(r"copy path")`

'data' is the variable name that we are given to the data we collected.

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

```
data = pd.read_csv(r"C:\Users\srava\OneDrive\Documents\weather data.csv")
```

```
data.head()
```

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather
0	01-01-2012 00:00	-1.3	-3.5	18	9	25.0	98.67	Clear
1	01-01-2012 01:00	7.4	2.8	20	24	24.1	99.37	Rain
2	01-01-2012 02:00	15.7	13.4	21	26	25.0	99.84	Cloudy
3	01-01-2012 03:00	4.9	-2.6	27	15	24.1	100.94	Mainly Clear
4	01-01-2012 04:00	-13.4	-19.7	30	4	25.0	102.32	Mostly Cloudy

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

- To know the unique wind speed values in the data we have to use the code : `data["Wind speed_km/h"].unique()`
- After running the above code it gives all the unique wind speed values in the wind speed column.

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

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

- The unique function gives the unique values in the data. So, the unique wind speed values are:
9, 24, 26, 15, 4, 0, 19, 17, 11, 22, 35, 13, 20, 6, 7, 30, 32, 41, 39, 28, 44, 33, 37, 52, 46, 2, 50, 48, 57, 63, 43, 83, 70, 54.

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

- To find the number of times when the data is clear , we have to use the function called 'loc'.
- It gives us the locations where the data is clear upon that we have to give length function, so that we'll get the number of times the weather is exactly clear.

- It gives us the locations where the data is clear upon that we have to give length function, so that we'll get the number of times the weather is exactly clear.

```
x = data['Weather']=='Clear'
len(data[x].value_counts())
```

1326

- The number of times the weather is clear is 1326.

Q. 3) Find the number of times when the 'Wind Speed was exactly 4 km/h'.

- To know the number of times that wind speed was exactly 4km/h is we have to access the wind speed column by applying assignment operator.
- (data["Wind Speed_km/h"]==4).value_counts()
- The above line will give the True and False counts where the True are the number of times the wind speed is exactly 4.

```
(data['Wind Speed_km/h']==4).value_counts()
```

False 8310

True 474

Name: Wind Speed_km/h, dtype: int64

- Here we got the total number of times that wind speed is exactly 4 is 474.

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

- To find the null values in the data we have to check the data using the code : data.isna()
- This will provides us a Boolean masking type table where in the table gives false if we do not have any null value and gives True if it consist of null values.

```
data.isna().value_counts()
```

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

dtype: int64

- By looking at the output we understand that the data do not have any null values because the value counts given false . so that we can understand that there are no null values in the data.

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

- To change the column name in a dataframe we have to use:
datac.rename({"Weather":"WeatherCondition"},axis=1,inplace=True)

- By the above code we can change the name of any column in the data

```
data.rename({'Weather': 'Weather Condition'}, axis=1, inplace=True)
data.head()
```

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Condition
0	01-01-2012 00:00	-1.3	-3.5	18	9	25.0	98.67	Clear
1	01-01-2012 01:00	7.4	2.8	20	24	24.1	99.37	Rain
2	01-01-2012 02:00	15.7	13.4	21	26	25.0	99.84	Cloudy
3	01-01-2012 03:00	4.9	-2.6	27	15	24.1	100.94	Mainly Clear
4	01-01-2012 04:00	-13.4	-19.7	30	4	25.0	102.32	Mostly Cloudy

- We can see that the Weather has changed to weather condition .

Q. 6) What is the mean 'Visibility' ?

- To find the mean for the column “visibility” we simply use the code : `data['Visibility_km'].mean()`
- It will gives the average (or) mean value of the required column as as central information of that column.

```
data['Visibility_km'].mean()
27.664446721311478
```

- For the data the variable (or) column name “Visibility_km” the mean is 27.6644

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

- We have to know the Standard deviation of ‘Pressure’ in the data.
- So, Now to find the standard deviation of the data we have a function called ‘.std()’.
- We have to use that function on the column.
- `data['Press_kPa'].std()`
- It will provide the information of standard deviation for the required column as

```
data['Press_kPa'].std()
0.8440047459486459
```

- From the data the standard deviation for the column name “Press_kPa” is 0.84400

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

- To find the variance for the data in the column "Relative Humidity" is simply by using the function '.var()'
- It will gives the variance for the column name "'Relative Humidity'" for the data

```
data['Rel Hum_%'].var()
286.2485501985015
```

- The variance for the column name "Relative Humidity" is 286.24855

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

- To know the data whose weather condition is "snow" is by using the code `data.loc[data["Weather Condition"]=="Snow"]`
- Here the data gives the dataframe table where the data consist of weather condition as snow

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

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Condition
11	01-01-2012 11:00	-6.2	-9.6	37	35	4.8	101.56	Snow
70	03-01-2012 22:00	-4.0	-6.6	62	22	16.1	100.48	Snow
73	04-01-2012 01:00	2.3	-3.4	64	35	25.0	103.43	Snow
105	05-01-2012 09:00	-1.8	-4.2	73	15	6.4	101.28	Snow
112	05-01-2012 16:00	1.7	-0.3	75	6	9.7	101.47	Snow
...
8573	9/22/2012 13:00	-6.0	-10.2	82	19	16.1	101.71	Snow
8650	9/25/2012 18:00	-4.6	-6.6	52	4	12.9	100.48	Snow
8671	9/26/2012 15:00	-5.9	-10.5	60	13	16.1	101.01	Snow
8713	9/28/2012 1:00	-5.2	-7.8	72	33	4.0	101.33	Snow
8734	9/28/2012 8:00	0.7	-1.2	79	30	8.0	101.22	Snow

390 rows x 8 columns

- The above output shows the information about the dataframe of the data whose weather condition is snow.
- There are 390 data points whose weather condition is snow.

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

- We have to know the data whose wind speed is greater than 24 and visibility is equal to 25 we use the code as `data.loc[(data["Wind Speed_km/h"]>24) & (data["Visibility_km"]==25)]`

- After performing above code it gives the data of column name “wind speed” greater than 24 and “visibility” is equal to 25.

```
data.loc[(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
2	01-01-2012 02:00	15.7	13.4	21	26	25.0	99.84	Cloudy
73	04-01-2012 01:00	2.3	-3.4	64	35	25.0	103.43	Snow
126	06-01-2012 06:00	10.0	5.4	77	39	25.0	101.30	Cloudy
158	07-01-2012 14:00	1.9	-2.1	87	26	25.0	100.87	Rain,Snow Grains
184	08-01-2012 16:00	14.2	9.2	35	44	25.0	99.49	Mostly Cloudy
...
8707	9/27/2012 5:00	-1.0	-6.0	70	33	25.0	98.56	Mostly Cloudy
8714	9/28/2012 10:00	2.6	0.3	72	26	25.0	101.60	Rain
8738	9/29/2012 10:00	22.8	12.3	80	28	25.0	101.60	Mostly Cloudy
8745	9/29/2012 17:00	-10.3	-12.9	82	28	25.0	102.16	Cloudy
8776	9/30/2012 23:00	19.2	13.2	93	43	25.0	101.60	Mainly Clear

308 rows x 8 columns

- There are 308 data points which satisfies both the conditions.

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

- The mean for each column on the specified weather condition is, to find that we have to group all the weather conditions by using the function ‘groupby()’.
- The code is:
- x = data.groupby(“Weather Condition”)
- x.agg(“mean”)
- Where the groupby will specify the data into different categories in the particular column and also we can know different values like ‘mean’, ‘max’, ‘min’, etc., by using aggregate function with it.

```
x = data.groupby('Weather Condition')
x.agg('mean')
```

C:\Users\srava\AppData\Local\Temp\ipykernel_17368\550145491.py:1: FutureWarning: The default value of numeric_only in DataFrame GroupBy.mean is deprecated. In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.

```
x.agg('mean')
```

	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa
Weather Condition						
Clear	6.825716	0.089367	67.127451	10.557315	30.153243	101.084495
Cloudy	7.970544	2.375810	67.349537	16.127315	26.625752	101.056852
Drizzle	7.353659	5.504878	69.048780	16.097561	17.931707	101.099268
Drizzle,Fog	8.067500	7.033750	70.062500	11.862500	5.257500	100.820750
Drizzle,Ice Pellets,Fog	0.400000	-0.700000	52.000000	20.000000	4.000000	99.440000
Drizzle,Snow	1.050000	0.150000	44.000000	14.000000	10.500000	100.490000
Drizzle,Snow,Fog	0.693333	0.120000	69.800000	15.533333	5.513333	100.971333
Fog	4.303333	3.159333	66.466667	7.946667	6.248000	101.149400
Freezing Drizzle	-5.657143	-8.000000	68.857143	16.571429	9.200000	101.070000

Freezing Drizzle,Fog	-2.533333	-4.183333	64.000000	17.000000	5.266667	100.851667
Freezing Drizzle,Haze	-5.433333	-8.000000	63.333333	10.333333	2.666667	101.136667
Freezing Drizzle,Snow	-5.109091	-7.072727	62.454545	16.272727	5.872727	100.380909
Freezing Fog	-7.575000	-9.250000	68.000000	4.750000	0.650000	101.222500
Freezing Rain	-3.885714	-6.078571	60.785714	19.214286	8.242857	101.500714
Freezing Rain,Fog	-2.225000	-3.750000	52.750000	15.500000	7.550000	100.267500
Freezing Rain,Haze	-4.900000	-7.450000	63.000000	7.500000	2.400000	100.265000
Freezing Rain,Ice Pellets,Fog	-2.600000	-3.700000	65.000000	28.000000	8.000000	98.330000
Freezing Rain,Snow Grains	-5.000000	-7.300000	92.000000	32.000000	4.800000	102.520000
Haze	-0.200000	-2.975000	69.625000	10.437500	7.831250	100.805625
Mainly Clear	12.558927	4.581671	68.020893	14.144824	34.264862	101.040940
Moderate Rain,Fog	1.700000	0.800000	89.000000	17.000000	6.400000	100.450000
Moderate Snow	-5.525000	-7.250000	67.500000	33.750000	0.750000	100.760000
Moderate Snow,Blowing Snow	-5.450000	-6.500000	81.500000	40.000000	0.600000	102.215000
Mostly Cloudy	10.574287	3.131174	67.214113	15.813920	31.253842	101.051054
Rain	9.786275	7.042810	67.614379	19.254902	18.856536	101.051797
Rain Showers	13.722340	9.187766	68.335106	17.132979	22.816489	101.020106
Rain Showers,Fog	12.800000	12.100000	31.000000	13.000000	6.400000	99.800000
Rain Showers,Snow Showers	2.150000	-1.500000	68.500000	22.500000	21.700000	101.080000
Rain,Fog	8.273276	7.219828	66.818966	14.793103	6.873276	100.991983
Rain,Haze	4.633333	2.066667	57.666667	11.666667	6.700000	100.716667

Rain,Ice Pellets	0.600000	-0.600000	54.000000	24.000000	9.700000	101.880000
Rain,Snow	1.055556	-0.566667	66.944444	28.388889	11.672222	100.895000
Rain,Snow Grains	1.900000	-2.100000	87.000000	26.000000	25.000000	100.870000
Rain,Snow,Fog	0.800000	0.300000	61.000000	9.000000	6.400000	102.480000
Rain,Snow,Ice Pellets	1.100000	-0.175000	72.500000	23.250000	6.000000	101.170000
Snow	-4.524103	-7.623333	66.402564	20.038462	11.171795	101.077205
Snow Pellets	0.700000	-6.400000	66.000000	35.000000	2.400000	99.560000
Snow Showers	-3.506667	-7.866667	65.600000	19.233333	20.158333	100.999333
Snow Showers,Fog	-10.675000	-11.900000	63.750000	13.750000	7.025000	100.770000
Snow,Blowing Snow	-5.410526	-7.621053	72.631579	34.842105	4.105263	101.032105
Snow,Fog	-5.075676	-6.364865	70.459459	17.324324	4.537838	101.194865
Snow,Haze	-4.020000	-6.860000	66.000000	5.000000	4.640000	100.360000
Snow,Ice Pellets	-1.883333	-3.666667	74.000000	23.833333	7.416667	100.746667
Thunderstorms	24.150000	19.750000	56.500000	7.500000	24.550000	101.375000
Thunderstorms,Heavy Rain Showers	10.900000	9.000000	82.000000	9.000000	2.400000	101.400000
Thunderstorms,Moderate Rain Showers,Fog	19.600000	18.500000	58.000000	15.000000	3.200000	99.940000
Thunderstorms,Rain	20.433333	18.533333	71.666667	15.666667	19.833333	101.536667
Thunderstorms,Rain Showers	20.037500	17.618750	68.437500	18.312500	15.893750	100.976875
Thunderstorms,Rain Showers,Fog	21.600000	18.700000	58.666667	19.666667	9.700000	100.806667
Thunderstorms,Rain,Fog	20.600000	18.600000	42.000000	19.000000	4.800000	100.450000

- The above data is the mean of all the data with the specified weather condition.

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

- To know the minimum and maximum data for all the columns on the specified column we use the code as `x.agg(func=["min","max"])`
- Where the x means the data which is already grouped by the condition for the particular required column name weather condition
- Also we use the aggregate function for the data where we can use any type of function inside it

In [98]: `k.agg(func=["min","max"])`

Out[98]:

	Weather	Date/Time		Temp_C		Dew Point Temp_C		Rel Hum_%		Wind Speed_km/h		Visibility_km		Press_kPa	
		min	max	min	max	min	max	min	max	min	max	min	max	min	max
	Clear	2012-01-01 00:00:00	2012-12-31 22:00:00	-23.3	32.8	-28.5	20.4	18	100	0	33	11.3	48.3	97.75	103.83
	Cloudy	2012-01-01 02:00:00	2012-12-31 17:00:00	-21.4	30.5	-26.8	22.6	20	100	0	54	11.3	48.3	97.52	103.52
	Drizzle	2012-01-06 08:00:00	2012-12-27 15:00:00	1.1	18.8	-0.2	17.7	37	97	0	30	6.4	25.0	98.29	103.58
	Drizzle,Fog	2012-01-04 01:00:00	2012-12-31 13:00:00	0.0	19.9	-1.6	19.1	38	98	0	28	1.0	9.7	98.32	103.56
	Drizzle,Ice Pellets,Fog	2012-07-24 05:00:00	2012-07-24 05:00:00	0.4	0.4	-0.7	-0.7	52	52	20	20	4.0	4.0	99.44	99.44
	Drizzle,Snow	2012-03-17 01:00:00	2012-05-02 09:00:00	0.9	1.2	0.1	0.2	39	49	9	19	9.7	11.3	100.27	100.71
	Drizzle,Snow,Fog	2012-01-25 01:00:00	2012-12-18 10:00:00	0.3	1.1	-0.1	0.6	46	94	7	32	2.4	9.7	99.26	102.47
	Fog	2012-01-01 13:00:00	2012-12-31 03:00:00	-16.0	20.8	-17.2	19.6	21	99	0	22	0.2	9.7	97.97	103.22
	Freezing Drizzle	2012-02-29 03:00:00	2012-08-21 05:00:00	-9.0	-2.3	-12.2	-3.3	43	89	6	26	4.8	12.9	99.75	101.78
	Freezing Drizzle,Fog	2012-02-21 21:00:00	2012-12-27 01:00:00	-6.4	-0.3	-9.0	-2.3	31	80	6	33	3.6	8.0	98.81	103.01
	Freezing Drizzle,Haze	2012-01-08 00:00:00	2012-10-15 21:00:00	-5.8	-5.0	-8.3	-7.7	32	81	9	11	2.0	4.0	100.55	101.83
	Freezing Drizzle,Snow	2012-01-19 15:00:00	2012-11-20 11:00:00	-8.3	-3.3	-10.4	-4.6	37	90	6	24	2.4	12.9	99.74	101.15
	Freezing Fog	2012-03-29 16:00:00	2012-12-31 02:00:00	-19.0	-0.1	-22.9	-0.3	34	86	0	9	0.2	0.8	100.66	101.64
	Freezing Rain	2012-01-03 13:00:00	2012-12-25 03:00:00	-6.5	0.3	-9.0	-1.7	40	100	7	28	2.8	16.1	100.92	102.45
	Freezing Rain,Fog	2012-03-25 00:00:00	2012-12-06 08:00:00	-6.1	0.1	-8.7	-0.9	35	77	7	26	2.8	9.7	99.45	101.21
	Freezing Rain,Haze	2012-03-25 23:00:00	2012-11-23 07:00:00	-4.9	-4.9	-7.5	-7.4	57	69	6	9	2.0	2.8	100.23	100.30
	Freezing Rain,Ice Pellets,Fog	2012-08-16 23:00:00	2012-08-16 23:00:00	-2.6	-2.6	-3.7	-3.7	65	65	28	28	8.0	8.0	98.33	98.33
	Freezing Rain,Snow Grains	2012-04-02 07:00:00	2012-04-02 07:00:00	-5.0	-5.0	-7.3	-7.3	92	92	32	32	4.8	4.8	102.52	102.52
	Haze	2012-01-02 17:00:00	2012-12-08 19:00:00	-11.5	14.1	-16.0	11.1	37	98	0	17	4.8	9.7	99.27	103.29
	Mainly Clear	2012-01-01 03:00:00	2012-12-31 18:00:00	-22.8	33.0	-28.0	21.2	20	100	0	63	12.9	48.3	97.84	103.65
	Moderate Rain,Fog	2012-08-20 16:00:00	2012-08-20 16:00:00	1.7	1.7	0.8	0.8	89	89	17	17	6.4	6.4	100.45	100.45
	Moderate Snow	2012-04-18 00:00:00	2012-12-05 21:00:00	-6.3	-4.9	-7.6	-6.7	29	85	26	39	0.6	0.8	99.93	101.96
	Moderate Snow,Blowing Snow	2012-06-07 00:00:00	2012-06-26 00:00:00	-5.5	-5.4	-6.6	-6.4	67	96	39	41	0.6	0.6	101.97	102.46

	00:00:00	02:00:00														
Mostly Cloudy	2012-01-01 04:00:00	2012-12-31 20:00:00	-23.2	32.4	-28.5	24.4	18	100	0	83	11.3	48.3	97.56	103.63		
Rain	2012-01-01 01:00:00	2012-12-30 18:00:00	0.3	22.8	-5.7	20.4	20	97	0	52	4.0	48.3	98.06	103.59		
Rain Showers	2012-01-03 11:00:00	2012-12-31 21:00:00	1.6	26.4	-7.2	23.0	24	99	0	41	6.4	48.3	97.93	103.65		
Rain Showers,Fog	2012-12-17 16:00:00	2012-12-17 16:00:00	12.8	12.8	12.1	12.1	31	31	13	13	6.4	6.4	99.80	99.80		
Rain Showers,Snow Showers	2012-08-31 07:00:00	2012-08-31 11:00:00	2.1	2.2	-1.8	-1.2	67	70	17	28	19.3	24.1	100.54	101.62		
Rain,Fog	2012-01-02 12:00:00	2012-12-30 07:00:00	0.0	21.7	-1.2	19.5	23	93	0	46	2.0	9.7	98.70	102.71		
Rain,Haze	2012-01-02 19:00:00	2012-12-10 11:00:00	4.0	5.5	1.0	2.9	40	75	7	17	4.0	9.7	99.89	101.52		
Rain,Ice Pellets	2012-10-03 01:00:00	2012-10-03 01:00:00	0.6	0.6	-0.6	-0.6	54	54	24	24	9.7	9.7	101.88	101.88		
Rain,Ice Pellets	2012-10-03 01:00:00	2012-10-03 01:00:00	0.6	0.6	-0.6	-0.6	54	54	24	24	9.7	9.7	101.88	101.88		
Rain,Snow	2012-03-26 20:00:00	2012-12-03 20:00:00	0.6	1.7	-1.7	0.5	31	93	13	52	2.4	25.0	100.03	102.21		
Rain,Snow Grains	2012-07-01 14:00:00	2012-07-01 14:00:00	1.9	1.9	-2.1	-2.1	87	87	26	26	25.0	25.0	100.87	100.87		
Rain,Snow,Fog	2012-05-10 03:00:00	2012-05-10 03:00:00	0.8	0.8	0.3	0.3	61	61	9	9	6.4	6.4	102.48	102.48		
Rain,Snow,Ice Pellets	2012-02-09 11:00:00	2012-07-19 22:00:00	0.9	1.3	-0.7	0.1	53	86	17	28	4.8	6.4	100.30	101.90		
Snow	2012-01-01 11:00:00	2012-12-31 23:00:00	-16.7	3.7	-24.6	0.3	20	100	0	57	1.0	25.0	97.99	103.65		
Snow Pellets	2012-07-19 02:00:00	2012-07-19 02:00:00	0.7	0.7	-6.4	-6.4	66	66	35	35	2.4	2.4	99.56	99.56		
Snow Showers	2012-01-03 05:00:00	2012-12-29 20:00:00	-13.3	2.9	-19.3	-0.7	31	95	0	37	2.4	48.3	99.09	102.45		
Snow Showers,Fog	2012-01-20 03:00:00	2012-11-29 10:00:00	-11.3	-10.0	-12.7	-11.1	56	76	7	22	4.0	9.7	100.33	101.48		
Snow,Blowing Snow	2012-01-09 10:00:00	2012-12-12 10:00:00	-12.0	-1.4	-16.2	-2.9	44	97	24	48	0.6	9.7	99.23	103.59		
Snow,Fog	2012-01-02 09:00:00	2012-12-21 01:00:00	-10.1	1.1	-12.0	0.8	38	99	4	35	1.2	9.7	99.60	103.51		
Snow,Haze	2012-01-06 17:00:00	2012-12-13 14:00:00	-4.3	-3.6	-7.2	-6.4	48	83	0	15	4.0	6.4	98.58	101.90		
Snow,Ice Pellets	2012-01-03 17:00:00	2012-11-07 12:00:00	-4.3	0.8	-5.9	-1.7	50	92	19	33	2.8	11.3	100.13	101.73		
Thunderstorms	2012-09-25 05:00:00	2012-11-29 16:00:00	21.6	26.7	19.4	20.1	56	57	0	15	24.1	25.0	100.86	101.89		
Thunderstorms,Heavy Rain Showers	2012-11-05 12:00:00	2012-11-05 12:00:00	10.9	10.9	9.0	9.0	82	82	9	9	2.4	2.4	101.40	101.40		
Thunderstorms,Moderate Rain Showers,Fog	2012-10-01 14:00:00	2012-10-01 14:00:00	19.6	19.6	18.5	18.5	58	58	15	15	3.2	3.2	99.94	99.94		
Thunderstorms,Rain	2012-09-19 14:00:00	2012-12-01 22:00:00	19.4	21.3	18.2	19.1	64	80	4	30	16.1	24.1	100.56	102.82		
Thunderstorms,Rain	2012-09-19 14:00:00	2012-12-01 22:00:00	19.4	21.3	18.2	19.1	64	80	4	30	16.1	24.1	100.56	102.82		
Thunderstorms,Rain Showers	2012-01-11 11:00:00	2012-10-04 07:00:00	11.0	25.5	7.0	23.1	44	95	7	32	6.4	25.0	99.40	102.55		
Thunderstorms,Rain Showers,Fog	2012-04-17 03:00:00	2012-08-15 09:00:00	19.5	22.9	16.1	21.3	34	82	7	35	9.7	9.7	99.33	101.77		
Thunderstorms,Rain,Fog	2012-11-04 16:00:00	2012-11-04 16:00:00	20.6	20.6	18.6	18.6	42	42	19	19	4.8	4.8	100.45	100.45		

- Here above is the data for the different weather conditions minimum and maximum values of the other parameters (or) columns in the data likely the data is in the above is given the maximum and minimum values of all the data on the specified weather condition

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

- To know the data records whose weather condition is Fog we use the code as `datac.loc[datac["Weather Condition"]=="Fog"]`

- It gives the entire data for the weather condition is equal to fog

```
In [40]: datac.loc[datac["Weather Condition"]=="Fog"]
```

```
Out[40]:
```

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Condition
13	01-01-2012 13:00	9.5	7.8	40	13	6.4	100.90	Fog
53	03-01-2012 05:00	-3.6	-4.3	57	7	9.7	101.32	Fog
136	06-01-2012 16:00	14.8	13.5	80	19	9.7	100.86	Fog
197	09-01-2012 05:00	2.1	0.7	43	11	8.0	101.44	Fog
278	12-01-2012 14:00	1.2	0.6	70	13	6.4	103.22	Fog
...
8475	9/18/2012 11:00	6.2	5.4	56	7	4.8	102.03	Fog
8511	9/19/2012 22:00	15.7	15.4	66	7	8.0	101.93	Fog
8518	9/19/2012 8:00	-2.9	-4.5	68	6	6.4	100.41	Fog
8537	9/20/2012 3:00	-0.5	-2.1	74	7	4.0	100.81	Fog
8771	9/30/2012 19:00	12.8	12.2	91	19	4.8	100.60	Fog

150 rows x 8 columns

- Here is the data whose weather condition is fog
- We have 150 different data points in the data whose weather condition is fog

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

- To get the data for the conditions either weather condition is clear or visibility is 40
- We use the code as

```
datac.loc[(datac["Weather Condition"]=="Clear") |
(datac["Visibility_km"]>40)]
```

- We get the data which satisfies the condition that weather condition is clear or visibility is greater than 40

```
In [41]: datac.loc[(datac["Weather Condition"]=="Clear") | (datac["Visibility_km"]>40)]
```

```
Out[41]:
```

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather Condition
0	01-01-2012 00:00	-1.3	-3.5	18	9	25.0	98.67	Clear
9	01-01-2012 09:00	20.0	3.8	35	17	48.3	100.11	Clear
16	01-01-2012 16:00	23.8	17.6	42	9	25.0	100.52	Clear
17	01-01-2012 17:00	-6.8	-9.8	42	20	48.3	100.76	Mainly Clear
18	01-01-2012 18:00	2.3	-2.4	42	6	48.3	101.05	Cloudy
...
8774	9/30/2012 21:00	23.0	14.7	62	13	48.3	101.93	Mostly Cloudy
8777	9/30/2012 3:00	9.3	5.8	95	9	48.3	101.25	Mainly Clear
8779	9/30/2012 5:00	1.4	-3.7	97	22	48.3	100.16	Cloudy
8780	9/30/2012 6:00	-4.6	-9.5	98	11	48.3	101.46	Mostly Cloudy
8781	9/30/2012 7:00	1.5	-6.3	99	30	24.1	101.48	Clear

3027 rows x 8 columns

- In the condition we should use the bitwise or (|) operator to satisfy the condition
- And the above data is the which satisfied the conditions
- We have 3027 number of data points in the data which satisfying the condition for weather condition is clear or visibility is greater than 40

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

- To get the data for the conditions either weather condition is clear and Relative humidity greater than 50 or visibility is greater than 40
- We use the code as

```
datac.loc[((datac["Weather Condition"]=="Clear") & (datac["Rel Hum_%"]>50))|(datac["Visibility_km"]>40)]
```

- This condition satisfies the requirement for the weather condition is clear and relative humidity is greater than 50 or else the visibility is greater than 40

```
In [42]: datac.loc[((datac["Weather Condition"]=="Clear") & (datac["Rel Hum_%"]>50))|(datac["Visibility_km"]>40)]
```

```
Out[42]:
```

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_kmh	Visibility_km	Press_kPa	Weather Condition
9	01-01-2012 09:00	20.0	3.8	35	17	48.3	100.11	Clear
17	01-01-2012 17:00	-6.8	-9.8	42	20	48.3	100.76	Mainly Clear
18	01-01-2012 18:00	2.3	-2.4	42	6	48.3	101.05	Cloudy
19	01-01-2012 19:00	-12.7	-17.2	43	17	48.3	101.16	Clear
23	01-01-2012 23:00	29.5	16.8	45	4	48.3	101.07	Mainly Clear
...
8774	9/30/2012 21:00	23.0	14.7	92	13	48.3	101.93	Mostly Cloudy
8777	9/30/2012 3:00	9.3	5.8	95	9	48.3	101.25	Mainly Clear
8779	9/30/2012 5:00	1.4	-3.7	97	22	48.3	100.16	Cloudy
8780	9/30/2012 6:00	-4.6	-9.5	98	11	48.3	101.46	Mostly Cloudy
8781	9/30/2012 7:00	1.5	-6.3	99	30	24.1	101.48	Clear

2864 rows x 8 columns

- In the condition we should use the bitwise and (&) and bitwise or (|) operators to satisfy the conditions
- And the above data is the which satisfied the conditions
- We have the total count of 2864 datapoints from the data which satisfies the condition for the weather condition should be clear and relative humidity greater than 50 or else the visibility is greater than 40

