


```
import pandas as pd
#Reading the csv file
df=pd.read_csv('1. Weather Data.csv')
```


```
df
```




	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
...
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

```
print(df.shape)
```

 (8784, 8)

```
#Q1: To check if there are null values and drop them
print(df.isnull().sum())
df.dropna() #drop any row with NaN
df.dropna(axis=1) #drop any column with NaN
```



```
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
```

	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
...
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

```
#Q2: To find unique instances of weather types
weather=df['Weather'].value_counts()
dfweather=pd.DataFrame(weather)
dfweather = dfweather.reset_index()
```

```
dfweather.columns = ['Weather', 'Frequency'] # change column names  
dfweather
```



	Weather	Frequency
0	Mainly Clear	2106
1	Mostly Cloudy	2069
2	Cloudy	1728
3	Clear	1326
4	Snow	390
5	Rain	306
6	Rain Showers	188
7	Fog	150
8	Rain,Fog	116
9	Drizzle,Fog	80
10	Snow Showers	60
11	Drizzle	41
12	Snow,Fog	37
13	Snow,Blowing Snow	19
14	Rain,Snow	18
15	Thunderstorms,Rain Showers	16
16	Haze	16
17	Drizzle,Snow,Fog	15
18	Freezing Rain	14
19	Freezing Drizzle,Snow	11
20	Freezing Drizzle	7
21	Snow,Ice Pellets	6
22	Freezing Drizzle,Fog	6
23	Snow,Haze	5
24	Freezing Fog	4
25	Snow Showers,Fog	4
26	Moderate Snow	4
27	Rain,Snow,Ice Pellets	4
28	Freezing Rain,Fog	4
29	Freezing Drizzle,Haze	3
30	Rain,Haze	3
31	Thunderstorms,Rain	3
32	Thunderstorms,Rain Showers,Fog	3
33	Freezing Rain,Haze	2
34	Drizzle,Snow	2
35	Rain Showers,Snow Showers	2
36	Thunderstorms	2
37	Moderate Snow,Blowing Snow	2
38	Rain Showers,Fog	1
39	Thunderstorms,Moderate Rain Showers,Fog	1
40	Snow Pellets	1
41	Rain,Snow,Fog	1
42	Moderate Rain,Fog	1
43	Freezing Rain,Ice Pellets,Fog	1
44	Drizzle,Ice Pellets,Fog	1
45	Thunderstorms.Rain.Fog	1

46	Rain,Ice Pellets	1
47	Rain,Snow Grains	1
48	Thunderstorms,Heavy Rain Showers	1
49	Freezing Rain,Snow Grains	1

#Q3: To rename column named 'Weather' to Weather Condition'

```
df.rename(columns = {'Weather' : 'Weather_Condition'}, inplace=False)
```

	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

#Q4: To find all records from data of when the weather was exactly clear

```
df[df['Weather_Condition'] == 'Clear']
```

	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
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

#Q5: To find the mean temperature, wind speed and visibility

```
print(df['Temp_C'].mean())
print(df['Wind Speed_km/h'].mean())
print(df['Visibility_km'].mean())
```

```
8.79814435336981
14.94546903460838
27.66444672131151
```

#Q6: To find the variance of pressure

```
print(df['Press_kPa'].var())
```

```
0.7123440111838408
```

```
#Q7: To find the days on which wind speed was less than or equal to 30 km/hr and temperature was greater than 0 C
windspeed=df[(df['Wind Speed_km/h'] <=30) & (df['Temp_C']>0)]
print(windspeed['Date/Time'])
print(windspeed.shape)
```

```
13      1/1/2012 13:00
14      1/1/2012 14:00
15      1/1/2012 15:00
16      1/1/2012 16:00
17      1/1/2012 17:00
...
8545    12/22/2012 1:00
8546    12/22/2012 2:00
8547    12/22/2012 3:00
8779    12/31/2012 19:00
8780    12/31/2012 20:00
Name: Date/Time, Length: 6302, dtype: object
```

```
#Q8: To find the date and temperatures for all instances when snow was recorded
snow=df[df['Weather_Condition'] == 'Snow']
snow.loc[:, ['Date/Time', 'Temp_C']]
```

```

Date/Time  Temp_C
55      1/3/2012 7:00   -14.0
84      1/4/2012 12:00   -13.7
86      1/4/2012 14:00   -11.3
87      1/4/2012 15:00   -10.2
88      1/4/2012 16:00    -9.4
...
8779    12/31/2012 19:00    0.1
8780    12/31/2012 20:00    0.2
8781    12/31/2012 21:00   -0.5
8782    12/31/2012 22:00   -0.2
8783    12/31/2012 23:00    0.0
```

390 rows × 2 columns

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
```

```
#Q9: To display a graph of variation of temperature with respect to time
graph=df[['Temp_C', 'Date/Time']]
graph.plot(x='Date/Time', y='Temp_C',figsize=(10,7) )
```

