2025

Weather Data Analysis

DATA ANALYSIS PROJECT

UTKARSH SHARMA

sharma.utkarsh2402@gmail.com

Project Overview

This project embarks on an in-depth analysis of weather patterns using a dataset encompassing key meteorological variables: date, time, temperature, dew point temperature, relative humidity percentage, wind speed in km/h, visibility in kilometres, atmospheric pressure in kilopascals, and general weather conditions. By dissecting this data, we aim to uncover hidden patterns, correlations, and trends that not only enhance our understanding of atmospheric behaviours but also contribute to more accurate forecasting models. The goal is to translate raw weather data into actionable insights that can benefit various sectors like agriculture, aviation, and urban planning.

Project Objectives

1. **Analyse Temporal Weather Patterns:**
    - Examine how weather variables fluctuate over different times of the day and throughout the year.
    - Identify peak periods for certain weather conditions, like high winds or low visibility.
2. **Evaluate speed patterns:**
    - Examine the maximum and minimum speed of the wind
3. **Access Weather Impact on Visibility:**
    - Investigate factors contributing to changes in visibility, such as humidity and precipitation.
    - Determine patterns that lead to foggy conditions or clear skies.
4. **Investigate Extreme Weather Events:**
    - Pinpoint dates and times with weather variables significantly deviating from the norm. Analyse conditions leading up to extreme events like heatwaves, cold snaps, or storms.
5. **Frequency Analysis:**
    - Determine how often extreme weather events occur within the dataset.
    - Assess any increasing or decreasing trends in extreme weather frequencies.

Cleaning of the data

We need to clean the dataset to analyse the data.

There are some steps we need to follow:

Step 1: We need to first import all the library required for the data analysis.

Output:

```
[3]
# Importing of the libraries
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

Step 2: Now we need to read the database by using panda's library

Output:

```
[2]
# Load/Read the dataset
df = pd.read_csv('Weather_Dataset.csv')
✓ 0.0s
```

Step 3: Now we need to see the top rows from the table

Output:

```
[3]
df.head(5)
✓ 0.0s
```

	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

Step 4: Then we need to see the last 5 rows.

Output:

```
df.tail(5)
```

✓ 0.0s

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

Step 5: Now we will see how many rows and columns there in the dataset are.

Output:

```
# To check all the rows and columns
df.shape
```

[5] ✓ 0.0s

```
(8784, 8)
```

Step 6: Now we need to check the all the information about the dataset like null values and datatypes.

Output:

```
df.info()
```

[6] ✓ 0.0s

```
<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
```

Step 7: After that we need to check the total of all the null values.

Output:

```
# To check the null values
df.isna().sum()
```

[7] ✓ 0.0s

```
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
```

Step 8: After checking all the null values then we need to drop all the duplicate values in the dataset

Output:

```
# Drop all the duplicate values. (if any)
df.drop_duplicates(inplace=True)
```

[8] ✓ 0.0s

Step 9: After that we need to convert the datatype of the column 'Date/Time' into the DateTime.

Output:

```
# Conversion into Date/Time
df['Date/Time'] = pd.to_datetime(df['Date/Time'])
```

[9] ✓ 0.0s

Step 10: After changing the data type, we need to sperate the date and time column from the column DateTime.

Output:

```
# Seperate columns
df['Date'] = df['Date/Time'].dt.date
df['Time'] = df['Date/Time'].dt.time
```

[10] ✓ 0.0s

Step 11: Now we need to see the 5 pointer theory which can be calculated with the help of describe method.

Output:

```
# 5 pointer theory
df.describe()
```

[11] ✓ 0.0s

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa
count	8784	8784.000000	8784.000000	8784.000000	8784.000000	8784.000000	8784.000000
mean	2012-07-01 23:30:00	8.798144	2.555294	67.431694	14.945469	27.664447	101.051623
min	2012-01-01 00:00:00	-23.300000	-28.500000	18.000000	0.000000	0.200000	97.520000
25%	2012-04-01 11:45:00	0.100000	-5.900000	56.000000	9.000000	24.100000	100.560000
50%	2012-07-01 23:30:00	9.300000	3.300000	68.000000	13.000000	25.000000	101.070000
75%	2012-10-01 11:15:00	18.800000	11.800000	81.000000	20.000000	25.000000	101.590000
max	2012-12-31 23:00:00	33.000000	24.400000	100.000000	83.000000	48.300000	103.650000
std	NaN	11.687883	10.883072	16.918881	8.688696	12.622688	0.844005

Interpretation of the data

1. **Maximum and minimum speed of the wind.**

   Code:

   ```
   print(f"Max Speed is: {df['Wind Speed_km/h'].max()} km\h")
   print(f"Max Speed is: {df['Wind Speed_km/h'].min()} km\h")
   ```

   Output:

   ```
   Max Speed is: 83 km\h
   Max Speed is: 0 km\h
   ```

2. **No. of unique wind values**

   Code:

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

   Output:

   ```
   34
   ```

3. **All the unique wind values in the dataset**

   Code:

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

   Output:

   ```
   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])
   ```

4. **All the data where weather is “Clear”**

   Code:

   ```
   df[df['Weather']=='Clear']
   ```

   Output:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather	Date	Time
67	2012-01-03 19:00:00	-16.9	-24.8	50	24	25.0	101.74	Clear	2012-01-03	19:00:00
114	2012-01-05 18:00:00	-7.1	-14.4	56	11	25.0	100.71	Clear	2012-01-05	18:00:00
115	2012-01-05 19:00:00	-9.2	-15.4	61	7	25.0	100.80	Clear	2012-01-05	19:00:00
116	2012-01-05 20:00:00	-9.8	-15.7	62	9	25.0	100.83	Clear	2012-01-05	20:00:00
117	2012-01-05 21:00:00	-9.0	-14.8	63	13	25.0	100.83	Clear	2012-01-05	21:00:00
...	...	...	...	...	...	...	...	...	...	...
8646	2012-12-26 06:00:00	-13.4	-14.8	89	4	25.0	102.47	Clear	2012-12-26	06:00:00
8698	2012-12-28 10:00:00	-6.1	-8.6	82	19	24.1	101.27	Clear	2012-12-28	10:00:00
8713	2012-12-29 01:00:00	-11.9	-13.6	87	11	25.0	101.31	Clear	2012-12-29	01:00:00
8714	2012-12-29 02:00:00	-11.8	-13.1	90	13	25.0	101.33	Clear	2012-12-29	02:00:00
8756	2012-12-30 20:00:00	-13.8	-16.5	80	24	25.0	101.52	Clear	2012-12-30	20:00:00

1326 rows × 10 columns

5. **Average humidity according to the weather conditions.**

Code:

df.groupby(df['Weather'])['Rel Hum_%'].mean().round()

Output:

Weather	
Clear	64.0
Cloudy	70.0
Drizzle	88.0
Drizzle,Fog	93.0
Drizzle,Ice Pellets,Fog	92.0
Drizzle,Snow	94.0
Drizzle,Snow,Fog	96.0
Fog	92.0
Freezing Drizzle	84.0
Freezing Drizzle,Fog	88.0
Freezing Drizzle,Haze	82.0
Freezing Drizzle,Snow	86.0
Freezing Fog	88.0
Freezing Rain	85.0
Freezing Rain,Fog	90.0
Freezing Rain,Haze	82.0
Freezing Rain,Ice Pellets,Fog	92.0
Freezing Rain,Snow Grains	84.0

6. **Count of different weather conditions in the dataset**

Code:

df['Weather'].value_counts()

Output:

Weather	
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
Freezing Drizzle,Fog	6
Snow,Ice Pellets	6
Snow,Haze	5
Freezing Rain,Fog	4
Moderate Snow	4
Rain,Snow,Ice Pellets	4
Freezing Fog	4
Snow Showers,Fog	4
Thunderstorms,Rain Showers,Fog	3

7. Average Visibility according to the weather in the dataset.

Code:

```
df.groupby(df['Weather'],as_index=False)['Visibility_km'].mean().sort_values(
by='Visibility_km',ascending=False)
```

Output:

	Weather	Visibility_km
19	Mainly Clear	34.264862
23	Mostly Cloudy	31.253842
0	Clear	30.153243
1	Cloudy	26.625752
32	Rain,Snow Grains	25.000000
43	Thunderstorms	24.550000
25	Rain Showers	22.816489
27	Rain Showers,Snow Showers	21.700000
37	Snow Showers	20.158333
46	Thunderstorms,Rain	19.833333
24	Rain	18.856536
2	Drizzle	17.931707
47	Thunderstorms,Rain Showers	15.893750
31	Rain,Snow	11.672222
35	Snow	11.171795
5	Drizzle,Snow	10.500000
30	Rain,Ice Pellets	9.700000
48	Thunderstorms,Rain Showers,Fog	9.700000
8	Freezing Drizzle	9.200000
13	Freezing Rain	8.242857
16	Freezing Rain,Ice Pellets,Fog	8.000000
18	Haze	7.831250
14	Freezing Rain,Fog	7.550000
42	Snow,Ice Pellets	7.416667

8. Weather that contain “Snow”

Code:

```
df[df['Weather'].str.contains('Snow')]
```

Output:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather	Date	Time
41	2012-01-02 17:00:00	-2.1	-9.5	57	22	25.0	99.66	Snow Showers	2012-01-02	17:00:00
44	2012-01-02 20:00:00	-5.6	-13.4	54	24	25.0	100.07	Snow Showers	2012-01-02	20:00:00
45	2012-01-02 21:00:00	-5.8	-12.8	58	26	25.0	100.15	Snow Showers	2012-01-02	21:00:00
47	2012-01-02 23:00:00	-7.4	-14.1	59	17	19.3	100.27	Snow Showers	2012-01-02	23:00:00
48	2012-01-03 00:00:00	-9.0	-16.0	57	28	25.0	100.35	Snow Showers	2012-01-03	00:00:00
...	...	...	...	...	...	...	...	...	...	...
8779	2012-12-31 19:00:00	0.1	-2.7	81	30	9.7	100.13	Snow	2012-12-31	19:00:00
8780	2012-12-31 20:00:00	0.2	-2.4	83	24	9.7	100.03	Snow	2012-12-31	20:00:00
8781	2012-12-31 21:00:00	-0.5	-1.5	93	28	4.8	99.95	Snow	2012-12-31	21:00:00
8782	2012-12-31 22:00:00	-0.2	-1.8	89	28	9.7	99.91	Snow	2012-12-31	22:00:00
8783	2012-12-31 23:00:00	0.0	-2.1	86	30	11.3	99.89	Snow	2012-12-31	23:00:00

583 rows × 10 columns

9. Weather that contain “Fog”

Code:

```
df[df['Weather'].str.contains('Fog')]
```

Output:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather	Date	Time
0	2012-01-01 00:00:00	-1.8	-3.9	86	4	8.0	101.24	Fog	2012-01-01	00:00:00
1	2012-01-01 01:00:00	-1.8	-3.7	87	4	8.0	101.24	Fog	2012-01-01	01:00:00
2	2012-01-01 02:00:00	-1.8	-3.4	89	7	4.0	101.26	Freezing Drizzle,Fog	2012-01-01	02:00:00
3	2012-01-01 03:00:00	-1.5	-3.2	88	6	4.0	101.27	Freezing Drizzle,Fog	2012-01-01	03:00:00
4	2012-01-01 04:00:00	-1.5	-3.3	88	7	4.8	101.23	Fog	2012-01-01	04:00:00
...	...	...	...	...	...	...	...	...	...	...
8739	2012-12-30 03:00:00	-9.1	-10.4	90	11	3.6	100.30	Snow,Fog	2012-12-30	03:00:00
8740	2012-12-30 04:00:00	-9.3	-10.6	90	13	9.7	100.28	Snow,Fog	2012-12-30	04:00:00
8741	2012-12-30 05:00:00	-9.1	-10.4	90	11	4.0	100.32	Snow,Fog	2012-12-30	05:00:00
8742	2012-12-30 06:00:00	-9.3	-10.8	89	17	8.0	100.39	Snow,Fog	2012-12-30	06:00:00
8770	2012-12-31 10:00:00	-7.4	-8.9	89	4	6.4	101.05	Snow,Fog	2012-12-31	10:00:00

426 rows × 10 columns

10. Data where the wind speed>25 and visibility =25

Code:

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

Output:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather	Date	Time
23	2012-01-01 23:00:00	5.3	2.0	79	30	25.0	99.31	Cloudy	2012-01-01	23:00:00
24	2012-01-02 00:00:00	5.2	1.5	77	35	25.0	99.26	Rain Showers	2012-01-02	00:00:00
25	2012-01-02 01:00:00	4.6	0.0	72	39	25.0	99.26	Cloudy	2012-01-02	01:00:00
26	2012-01-02 02:00:00	3.9	-0.9	71	32	25.0	99.26	Mostly Cloudy	2012-01-02	02:00:00
27	2012-01-02 03:00:00	3.7	-1.5	69	33	25.0	99.30	Mostly Cloudy	2012-01-02	03:00:00
...	...	...	...	...	...	...	...	...	...	...
8705	2012-12-28 17:00:00	-8.6	-12.0	76	26	25.0	101.34	Mainly Clear	2012-12-28	17:00:00
8753	2012-12-30 17:00:00	-12.1	-15.8	74	28	25.0	101.26	Mainly Clear	2012-12-30	17:00:00
8755	2012-12-30 19:00:00	-13.4	-16.5	77	26	25.0	101.47	Mainly Clear	2012-12-30	19:00:00
8759	2012-12-30 23:00:00	-12.1	-15.1	78	28	25.0	101.52	Mostly Cloudy	2012-12-30	23:00:00
8760	2012-12-31 00:00:00	-11.1	-14.4	77	26	25.0	101.51	Cloudy	2012-12-31	00:00:00

308 rows × 10 columns

11. Either Weather is 'Clear' or Visibility is greater than 20

Code:

```
df[(df['Weather']=='Clear') | (df['Visibility_km']>20)]
```

Output:

	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Weather	Date	Time
20	2012-01-01 20:00:00	3.2	1.3	87	19	25.0	99.50	Cloudy	2012-01-01	20:00:00
21	2012-01-01 21:00:00	4.0	1.7	85	20	25.0	99.39	Cloudy	2012-01-01	21:00:00
23	2012-01-01 23:00:00	5.3	2.0	79	30	25.0	99.31	Cloudy	2012-01-01	23:00:00
24	2012-01-02 00:00:00	5.2	1.5	77	35	25.0	99.26	Rain Showers	2012-01-02	00:00:00
25	2012-01-02 01:00:00	4.6	0.0	72	39	25.0	99.26	Cloudy	2012-01-02	01:00:00
...	...	...	...	...	...	...	...	...	...	...
8762	2012-12-31 02:00:00	-10.1	-13.4	77	9	25.0	101.45	Cloudy	2012-12-31	02:00:00
8763	2012-12-31 03:00:00	-11.8	-14.4	81	6	25.0	101.42	Mostly Cloudy	2012-12-31	03:00:00
8764	2012-12-31 04:00:00	-10.5	-12.8	83	11	25.0	101.34	Cloudy	2012-12-31	04:00:00
8765	2012-12-31 05:00:00	-10.2	-12.4	84	6	25.0	101.28	Cloudy	2012-12-31	05:00:00
8766	2012-12-31 06:00:00	-9.7	-11.7	85	4	25.0	101.23	Cloudy	2012-12-31	06:00:00

7302 rows × 10 columns

12. All the means values in respect to weather column

Code:

```
numerical_cols =['Temp_C','Dew Point Temp_C','Rel Hum_%','Wind
Speed_km/h','Visibility_km','Press_kPa']
df.groupby(df['Weather'])[numerical_cols].mean()
```

Output:

Weather	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
Freezing Drizzle,Snow	-5.109091	-7.072727	86.090909	16.272727	5.872727	100.520909
Freezing Fog	-7.575000	-9.250000	87.750000	4.750000	0.650000	102.320000
Freezing Rain	-3.885714	-6.078571	84.642857	19.214286	8.242857	99.647143
Freezing Rain,Fog	-2.225000	-3.750000	89.500000	15.500000	7.550000	99.945000
Freezing Rain,Haze	-4.900000	-7.450000	82.500000	7.500000	2.400000	100.375000
Freezing Rain,Ice Pellets,Fog	-2.600000	-3.700000	92.000000	28.000000	8.000000	100.950000
Freezing Rain,Snow Grains	-5.000000	-7.300000	84.000000	32.000000	4.800000	98.560000
Haze	-0.200000	-2.975000	81.625000	10.437500	7.831250	101.482500
Mainly Clear	12.558927	4.581671	60.667142	14.144824	34.264862	101.248832
Moderate Rain,Fog	1.700000	0.800000	94.000000	17.000000	6.400000	99.980000
Moderate Snow	-5.525000	-7.250000	87.750000	33.750000	0.750000	100.275000
Moderate Snow,Blowing Snow	-5.450000	-6.500000	92.500000	40.000000	0.600000	100.570000

13. All the max values in respect to Weather column

Code:

```
numerical_cols =['Temp_C','Dew Point Temp_C','Rel Hum_%','Wind
Speed_km/h','Visibility_km','Press_kPa']
df.groupby(df['Weather'])[numerical_cols].max()
```

Output:

Weather	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa
Clear	32.8	20.4	99	33	48.3	103.63
Cloudy	30.5	22.6	99	54	48.3	103.65
Drizzle	18.8	17.7	96	30	25.0	101.56
Drizzle,Fog	19.9	19.1	100	28	9.7	102.07
Drizzle,Ice Pellets,Fog	0.4	-0.7	92	20	4.0	100.79
Drizzle,Snow	1.2	0.2	95	19	11.3	101.15
Drizzle,Snow,Fog	1.1	0.6	98	32	9.7	100.15
Fog	20.8	19.6	100	22	9.7	103.04
Freezing Drizzle	-2.3	-3.3	93	26	12.9	101.02
Freezing Drizzle,Fog	-0.3	-2.3	94	33	8.0	101.27
Freezing Drizzle,Haze	-5.0	-7.7	83	11	4.0	100.36
Freezing Drizzle,Snow	-3.3	-4.6	94	24	12.9	101.18
Freezing Fog	-0.1	-0.3	99	9	0.8	102.85
Freezing Rain	0.3	-1.7	92	28	16.1	101.00
Freezing Rain,Fog	0.1	-0.9	93	26	9.7	101.01
Freezing Rain,Haze	-4.9	-7.4	83	9	2.8	100.41
Freezing Rain,Ice Pellets,Fog	-2.6	-3.7	92	28	8.0	100.95
Freezing Rain,Snow Grains	-5.0	-7.3	84	32	4.8	98.56
Haze	14.1	11.1	86	17	9.7	102.97
Mainly Clear	33.0	21.2	99	63	48.3	103.59

14. All the min values in respect to Weather column

Code:

```
numerical_cols =['Temp_C','Dew Point Temp_C','Rel Hum_%','Wind
Speed_km/h','Visibility_km','Press_kPa']
df.groupby(df['Weather'])[numerical_cols].min()
```

Output:

Weather	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa
Clear	-23.3	-28.5	20	0	11.3	99.52
Cloudy	-21.4	-26.8	18	0	11.3	98.39
Drizzle	1.1	-0.2	74	0	6.4	97.84
Drizzle,Fog	0.0	-1.6	85	0	1.0	98.65
Drizzle,Ice Pellets,Fog	0.4	-0.7	92	20	4.0	100.79
Drizzle,Snow	0.9	0.1	92	9	9.7	100.63
Drizzle,Snow,Fog	0.3	-0.1	92	7	2.4	97.79
Fog	-16.0	-17.2	80	0	0.2	98.31
Freezing Drizzle	-9.0	-12.2	78	6	4.8	98.44
Freezing Drizzle,Fog	-6.4	-9.0	82	6	3.6	98.74
Freezing Drizzle,Haze	-5.8	-8.3	81	9	2.0	100.28
Freezing Drizzle,Snow	-8.3	-10.4	79	6	2.4	99.19
Freezing Fog	-19.0	-22.9	71	0	0.2	101.97
Freezing Rain	-6.5	-9.0	81	7	2.8	98.22
Freezing Rain,Fog	-6.1	-8.7	82	7	2.8	98.32
Freezing Rain,Haze	-4.9	-7.5	82	6	2.0	100.34
Freezing Rain,Ice Pellets,Fog	-2.6	-3.7	92	28	8.0	100.95
Freezing Rain,Snow Grains	-5.0	-7.3	84	32	4.8	98.56
Haze	-11.5	-16.0	68	0	4.8	100.35
Mainly Clear	-22.8	-28.0	20	0	12.9	98.67
Moderate Rain,Fog	1.7	0.8	94	17	6.4	99.98
Moderate Snow	-6.3	-7.6	83	26	0.6	99.88
Moderate Snow,Blowing Snow	-5.5	-6.6	92	39	0.6	100.50

15. Average visibility each hours

Code:

```
df.groupby(df['Time'])[['Visibility_km']].mean()
```

Output:

Time	Visibility_km
00:00:00	22.650546
01:00:00	22.693989
02:00:00	22.496175
03:00:00	22.331694
04:00:00	22.727322
05:00:00	24.734153
06:00:00	27.744536
07:00:00	29.095082
08:00:00	29.938798
09:00:00	31.267760
10:00:00	31.668033
11:00:00	32.222678
12:00:00	32.881967
13:00:00	33.054645
14:00:00	33.141803
15:00:00	32.882514
16:00:00	32.446721
17:00:00	31.324863
18:00:00	29.067213
19:00:00	26.506284
20:00:00	24.245628
21:00:00	22.944536
22:00:00	23.046448

Conclusion

The analysis of wind data, incorporating factors such as wind speed, weather conditions, average humidity, and visibility, provides a comprehensive understanding of the wind patterns and their impact on various weather-related phenomena.

Key findings include:

1. **Wind Speed:** The data revealed the mean, maximum, and minimum wind speeds, offering insights into the variability and extremes of wind behavior over the observed period.
2. **Weather Conditions:** Different weather conditions were analyzed to understand their influence on wind speed and direction. It was observed that certain weather conditions, like thunderstorms or high-pressure systems, significantly impacted wind characteristics.
3. **Average Humidity:** The average humidity levels were correlated with wind speed, showing how moisture in the air can influence wind patterns.
4. **Visibility:** Average visibility was assessed and compared with wind speed. The analysis indicated that higher wind speeds often corresponded with reduced visibility due to factors like dust or precipitation being carried by the wind.
5. **Comparative Analysis:** The comparative study of visibility and wind speed highlighted the interplay between these two factors, providing valuable information for weather forecasting and planning.
6. **Statistical Insights:** The mean, maximum, and minimum wind speeds offered a statistical perspective on the data, helping to identify trends and patterns.

In conclusion, the wind data analysis provides a detailed picture of wind behavior and its correlation with various meteorological factors. These insights can aid in improving weather prediction models, enhancing safety measures, and optimizing various wind-dependent activities and industries.