


```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
df=pd.read_csv('/content/file.csv')
```

```
df.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	Fo
1	1/1/2012 1:00	-1.8	-3.7	87	4	8.0	101.24	Fo

1/1/2012

Freezin


Next steps:

[Generate code with df](#)

 [View recommended plots](#)

[New interactive sheet](#)

```
df.info()
```

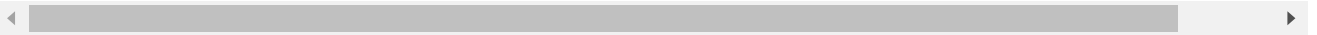


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8784 entries, 0 to 8783
Data columns (total 8 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                -
0   Date/Time                            8784 non-null   datetime64[ns]
1   Temp_C                              8784 non-null   float64
2   Dew Point Temp_C                    8784 non-null   float64
3   Rel Hum_%                           8784 non-null   float64
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 Condition                   8784 non-null   object
dtypes: datetime64[ns](1), float64(5), int64(1), object(1)
memory usage: 549.1+ KB
```

```
df.describe()
```



	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa
count	8784.000000	8784.000000	8784.000000	8784.000000	8784.000000	8784.000000
mean	8.798144	2.555294	67.431694	14.945469	27.664447	101.051623
std	11.687883	10.883072	16.918881	8.688696	12.622688	0.844005
min	-23.300000	-28.500000	18.000000	0.000000	0.200000	97.520000
25%	0.100000	-5.900000	56.000000	9.000000	24.100000	100.560000
50%	9.300000	3.300000	68.000000	13.000000	25.000000	101.070000
75%	18.800000	11.800000	81.000000	20.000000	25.000000	101.590000



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

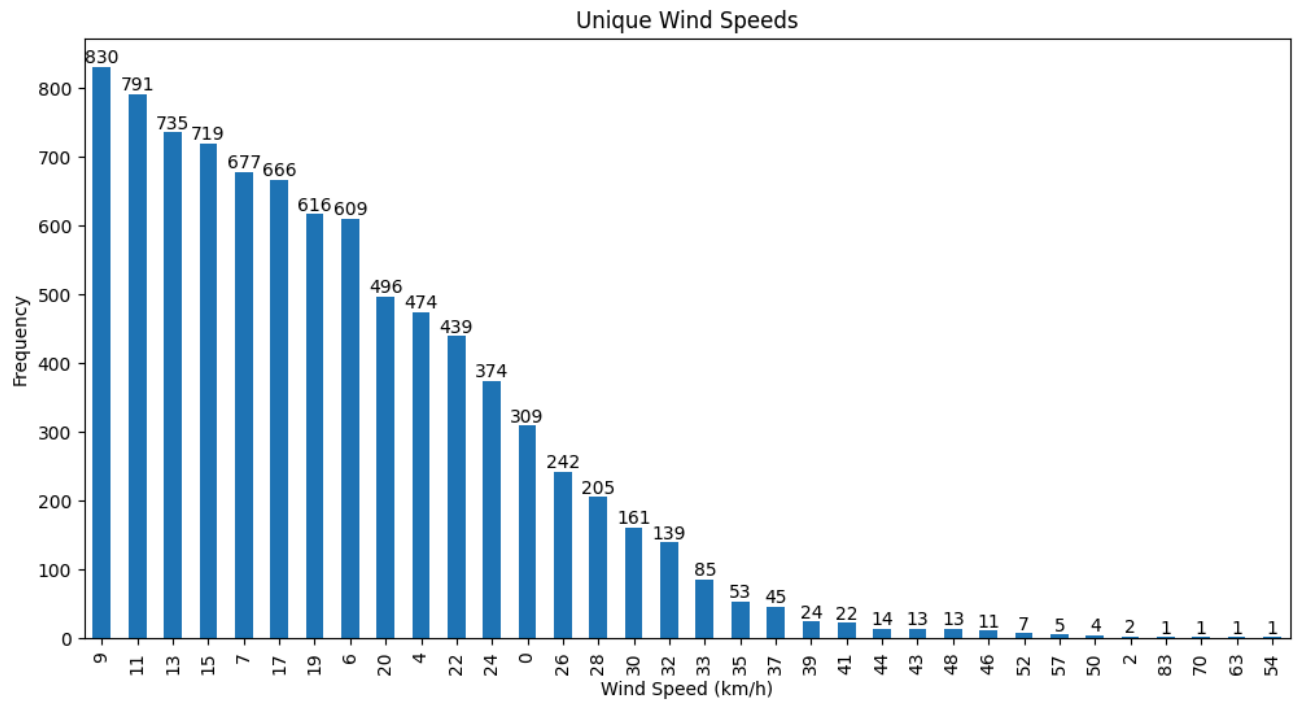
Q. 1) Find all the unique 'Wind Speed' values in the data. Q. 2) Find the number of times when the 'Weather is exactly Clear'. Q. 3) Find the number of times when the 'Wind Speed was exactly 4 km/h'. Q. 4) Find out all the Null Values in the data. Q. 5) Rename the column name 'Weather' of the dataframe to 'Weather Condition'. Q. 6) What is the mean 'Visibility' ? Q. 7) What is the Standard Deviation of 'Pressure' in this data? Q. 8) What is the Variance of 'Relative Humidity' in this data ? Q. 9) Find all instances when 'Snow' was recorded. Q. 10) Find all instances when 'Wind Speed is above 24' and 'Visibility is 25'. Q. 11) What is the Mean value of each column against each 'Weather Condition ? Q. 12) What is the Minimum & Maximum value of each column against each 'Weather Condition ? Q. 13) Show all the Records where Weather Condition is Fog. Q. 14) Find all instances when 'Weather is Clear' or 'Visibility is above 40'. Q. 15) Find all instances when : A. 'Weather is Clear' and 'Relative Humidity is greater than 50' or B. 'Visibility is above 40'

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

```
unique_wind_speeds = df['Wind Speed_km/h'].unique()
print(unique_wind_speeds)
plt.figure(figsize=(12, 6))
ax=df['Wind Speed_km/h'].value_counts().plot(kind='bar', title='Unique Wind Speeds')
for p in ax.patches:
    ax.text(p.get_x() + p.get_width() / 2, p.get_height(), int(p.get_height()),
            ha='center', va='bottom')

plt.xlabel('Wind Speed (km/h)')
plt.ylabel('Frequency')
plt.show()
```

➡ [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'.

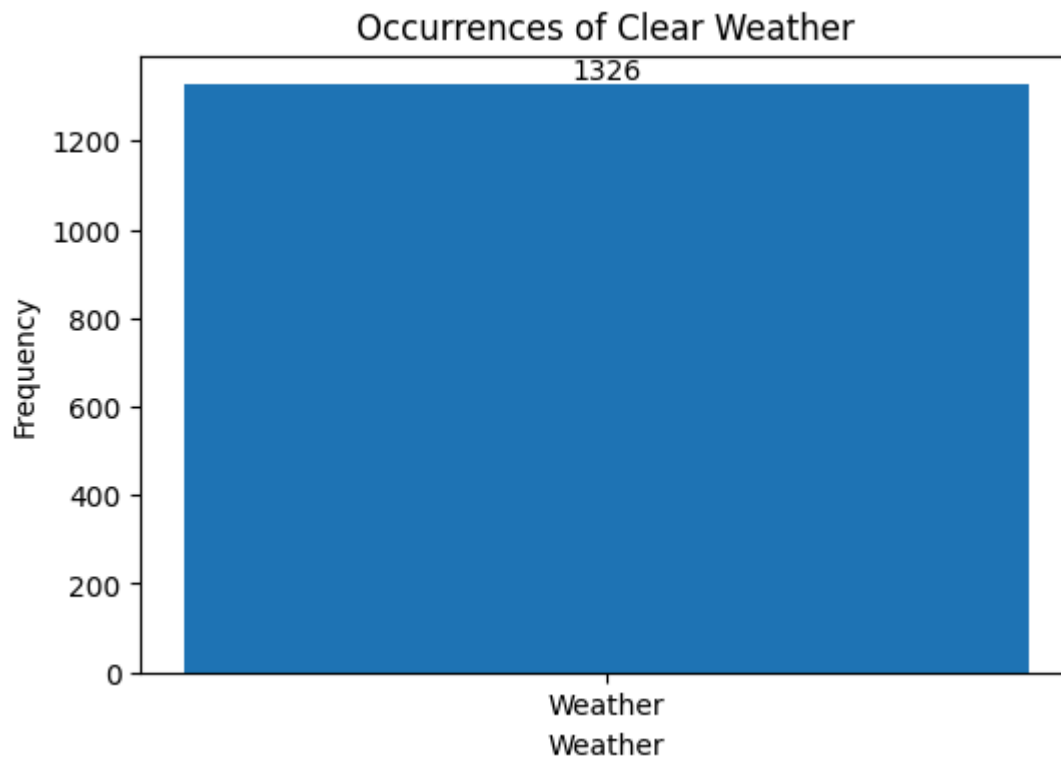
```
clear_weather_data = df[df['Weather'] == 'Clear']
print(clear_weather_data.value_counts().sum())

# Count the occurrences
count = clear_weather_data.shape[0]

# Create a bar chart
plt.figure(figsize=(6, 4))
plt.bar(['Weather'], [count])
plt.title('Occurrences of Clear Weather')
plt.xlabel('Weather')
plt.ylabel('Frequency')

# Add count on top of the bar
plt.text(0, count, str(count), ha='center', va='bottom')

plt.show()
```



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

```
Wind_Speed= df[df['Wind Speed_km/h'] == 4]

print("The Number of Time the Weend Speed was Exactly 4Km/h: is" ,Wind_Speed.value_co

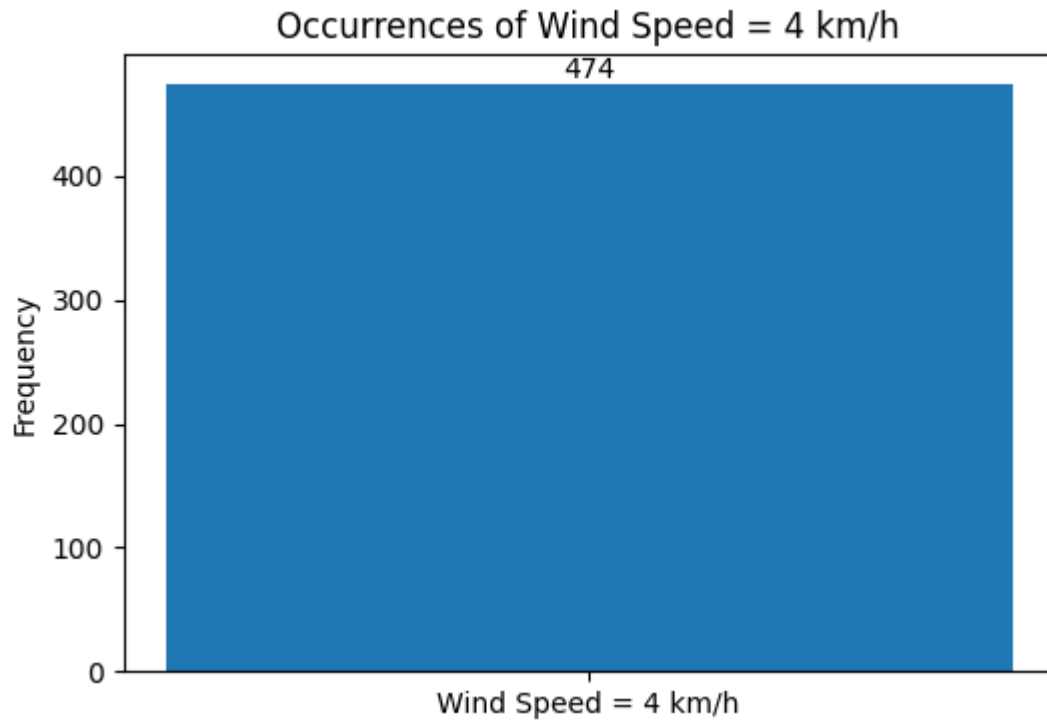
# Count the occurrences
count = Wind_Speed.shape[0]

# Create a bar chart
plt.figure(figsize=(6, 4))
plt.bar(['Wind Speed = 4 km/h'], [count])
plt.title('Occurrences of Wind Speed = 4 km/h')
plt.ylabel('Frequency')

# Add count on top of the bar
plt.text(0, count, str(count), ha='center', va='bottom')

plt.show()
```

⇒ The Number of Time the Weend Speed was Exactly 4Km/h: is 474 Times



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

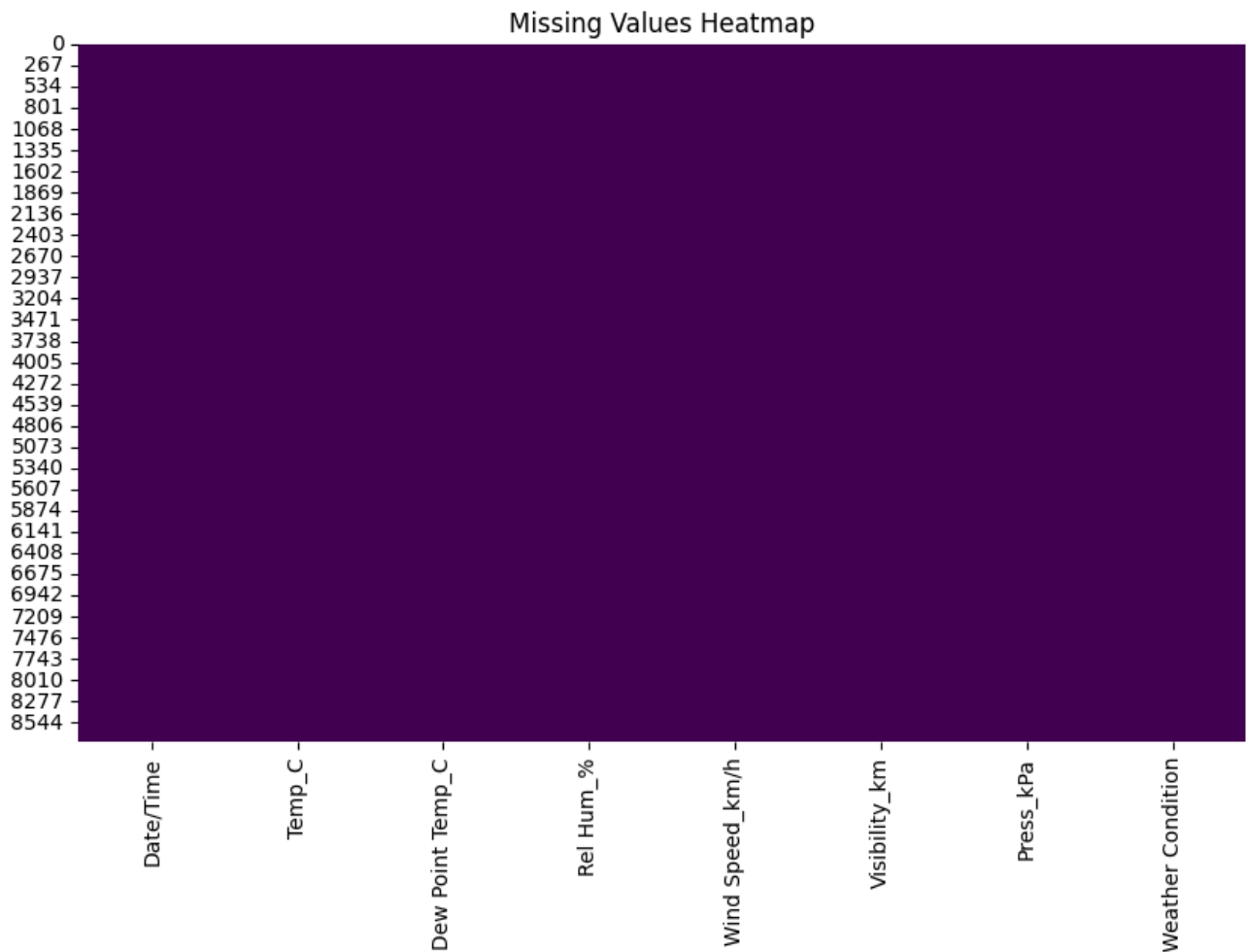
```
print(df.isnull().sum())

# Create a heatmap of missing values
plt.figure(figsize=(10, 6))
sns.heatmap(df.isnull(), cbar=False, cmap='viridis')
plt.title('Missing Values Heatmap')
plt.show()
```

```

→ 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 Condition 0
dtype: int64

```



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

```
df.rename(columns={'Weather': 'Weather Condition'}, inplace=True)
```

```
df.columns
```

```

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

```

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

```
vis_mean=df['Visibility_km'].mean()

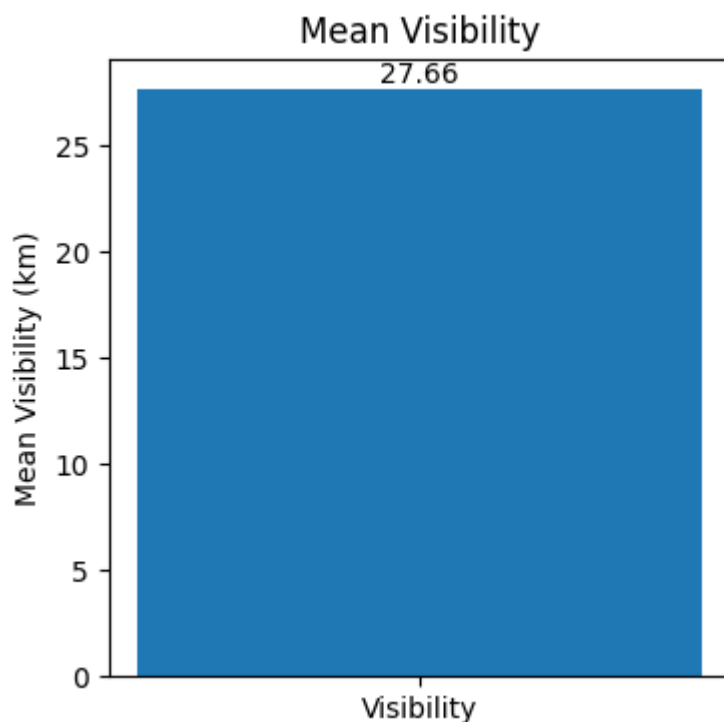
print(f"The Mean of Visibility is {vis_mean}")

plt.figure(figsize=(4, 4))
plt.bar(['Visibility'], [vis_mean])
plt.title('Mean Visibility')
plt.ylabel('Mean Visibility (km)')

plt.text(0, vis_mean, f'{vis_mean:.2f}', ha='center', va='bottom')

plt.show()
```

⇒ The Mean of Visibility is 27.664446721311478



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

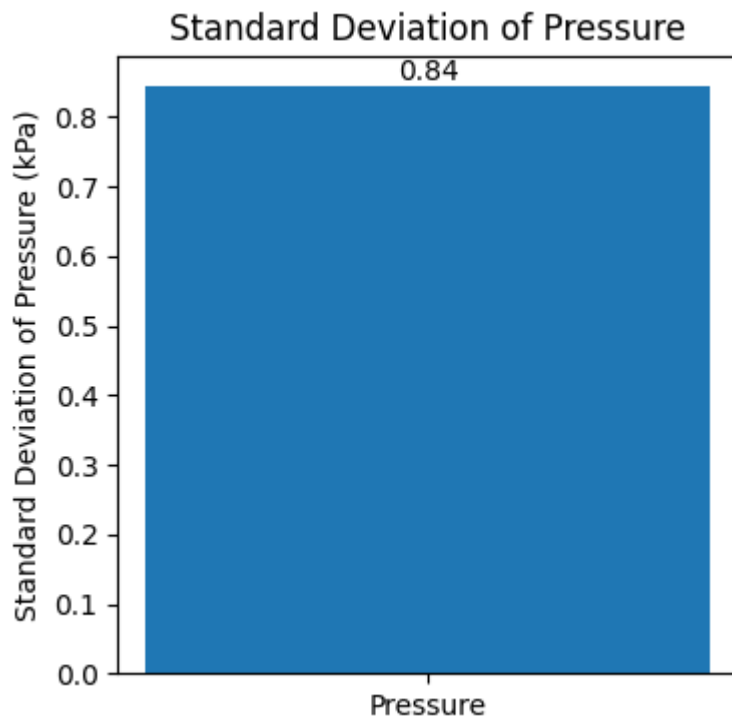
```
pressure_std=df['Press_kPa'].std()
print(f"The Standard Deviation of Pressure is {pressure_std}")

plt.figure(figsize=(4, 4))
plt.bar(['Pressure'], [pressure_std])
plt.title('Standard Deviation of Pressure')
plt.ylabel('Standard Deviation of Pressure (kPa)')
plt.show

plt.text(0, pressure_std, f'{pressure_std:.2f}', ha='center', va='bottom')
```

```
plt.show()
```

➡ The Standard Deviation of Pressure is 0.8440047459486474



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

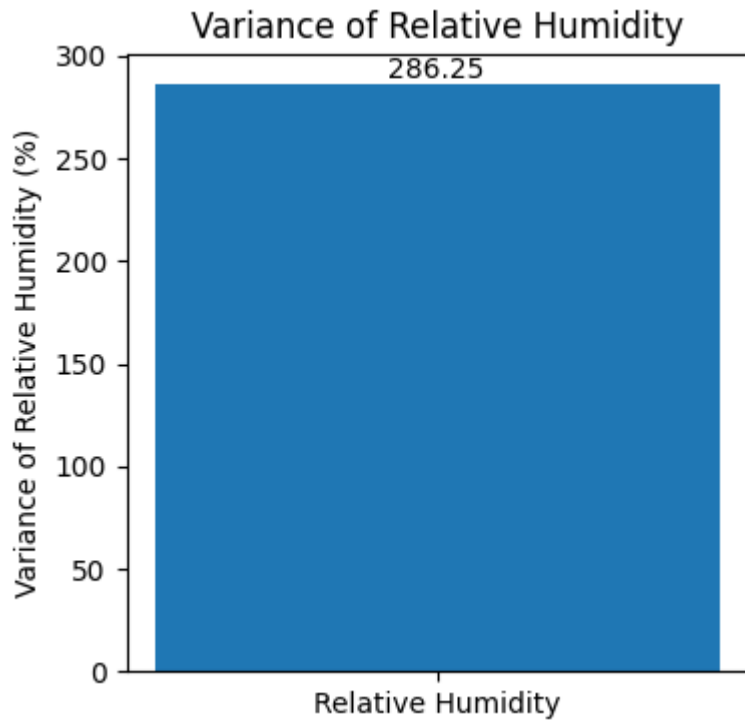
```
df['Rel Hum_%'] = df['Rel Hum_%'].astype(float)
humidity_variance = df['Rel Hum_%'].var()
print(f"The Variance of Relative Humidity is {humidity_variance}")

plt.figure(figsize=(4, 4))
plt.bar(['Relative Humidity'], [humidity_variance])
plt.title('Variance of Relative Humidity')
plt.ylabel('Variance of Relative Humidity (%)')
plt.show

plt.text(0, humidity_variance, f'{humidity_variance:.2f}', ha='center', va='bottom')

plt.show()
```


⇒ The Variance of Relative Humidity is 286.2485501984998



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

```
snow_data = df[df['Weather Condition'] == 'Snow']  
display(snow_data)
```



	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	63.0	19	25.0	100.95	
84	1/4/2012 12:00	-13.7	-21.7	51.0	11	24.1	101.25	
86	1/4/2012 14:00	-11.3	-19.0	53.0	7	19.3	100.97	
87	1/4/2012 15:00	-10.2	-16.3	61.0	11	9.7	100.89	
88	1/4/2012 16:00	-9.4	-15.5	61.0	13	19.3	100.79	
...	
8779	12/31/2012 19:00	0.1	-2.7	81.0	30	9.7	100.13	

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

```
wind_speed_visibility_data = df[(df['Wind Speed_km/h'] > 24) & (df['Visibility_km'] ==
display(wind_speed_visibility_data)
```



	Date/Time	Temp_C	Dew Point Temp_C	Rel Hum_%	Wind Speed_km/h	Visibility_km	Press_kPa	Wea Condi
23	1/1/2012 23:00	5.3	2.0	79.0	30	25.0	99.31	C
24	1/2/2012 0:00	5.2	1.5	77.0	35	25.0	99.26	Sh
25	1/2/2012 1:00	4.6	0.0	72.0	39	25.0	99.26	C
26	1/2/2012 2:00	3.9	-0.9	71.0	32	25.0	99.26	M C
27	1/2/2012 3:00	3.7	-1.5	69.0	33	25.0	99.30	M C
...	
8705	12/28/2012 17:00	-8.6	-12.0	76.0	26	25.0	101.34	M

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

```
weather_condition_means = df.groupby('Weather Condition')['Visibility_km'].mean()
print(weather_condition_means)
```



```
Weather Condition
Clear                30.153243
Cloudy              26.625752
Drizzle             17.931707
Drizzle,Fog         5.257500
Drizzle,Ice Pellets,Fog  4.000000
Drizzle,Snow       10.500000
Drizzle,Snow,Fog    5.513333
Fog                 6.248000
Freezing Drizzle    9.200000
Freezing Drizzle,Fog  5.266667
Freezing Drizzle,Haze  2.666667
Freezing Drizzle,Snow  5.872727
Freezing Fog        0.650000
Freezing Rain       8.242857
Freezing Rain,Fog   7.550000
Freezing Rain,Haze  2.400000
Freezing Rain,Ice Pellets,Fog  8.000000
Freezing Rain,Snow Grains  4.800000
Haze                7.831250
Mainly Clear        34.264862
Moderate Rain,Fog   6.400000
Moderate Snow       0.750000
Moderate Snow,Blowing Snow  0.600000
```

Mostly Cloudy	31.253842
Rain	18.856536
Rain Showers	22.816489
Rain Showers,Fog	6.400000
Rain Showers,Snow Showers	21.700000
Rain,Fog	6.873276
Rain,Haze	6.700000
Rain,Ice Pellets	9.700000
Rain,Snow	11.672222
Rain,Snow Grains	25.000000
Rain,Snow,Fog	6.400000
Rain,Snow,Ice Pellets	6.000000
Snow	11.171795
Snow Pellets	2.400000
Snow Showers	20.158333
Snow Showers,Fog	7.025000
Snow,Blowing Snow	4.105263
Snow,Fog	4.537838
Snow,Haze	4.640000
Snow,Ice Pellets	7.416667
Thunderstorms	24.550000
Thunderstorms,Heavy Rain Showers	2.400000
Thunderstorms,Moderate Rain Showers,Fog	3.200000
Thunderstorms,Rain	19.833333
Thunderstorms,Rain Showers	15.893750
Thunderstorms,Rain Showers,Fog	9.700000
Thunderstorms,Rain,Fog	4.800000

Name: Visibility_km, dtype: float64

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

```
# Group data by 'Weather Condition' and get the minimum and maximum values for each column
weather_condition_minmax = df.drop(columns=['Date/Time']).groupby('Weather Condition').minmax()

# Display the results
display(weather_condition_minmax)
```



Drizzle	1.1	18.8	-0.2	17.7	74.0	96.0	0	30	6.4
Drizzle,Fog	0.0	19.9	-1.6	19.1	85.0	100.0	0	28	1.0
Drizzle,Ice Pellets,Fog	0.4	0.4	-0.7	-0.7	92.0	92.0	20	20	4.0
Drizzle,Snow	0.9	1.2	0.1	0.2	92.0	95.0	9	19	9.7
Drizzle,Snow,Fog	0.3	1.1	-0.1	0.6	92.0	98.0	7	32	2.4
Fog	-16.0	20.8	-17.2	19.6	80.0	100.0	0	22	0.2
Freezing Drizzle	-9.0	-2.3	-12.2	-3.3	78.0	93.0	6	26	4.8
Freezing Drizzle,Fog	-6.4	-0.3	-9.0	-2.3	82.0	94.0	6	33	3.6
Freezing Drizzle,Haze	-5.8	-5.0	-8.3	-7.7	81.0	83.0	9	11	2.0
Freezing Drizzle,Snow	-8.3	-3.3	-10.4	-4.6	79.0	94.0	6	24	2.4
Freezing Fog	-19.0	-0.1	-22.9	-0.3	71.0	99.0	0	9	0.2
Freezing Rain	-6.5	0.3	-9.0	-1.7	81.0	92.0	7	28	2.8
Freezing Rain,Fog	-6.1	0.1	-8.7	-0.9	82.0	93.0	7	26	2.8
Freezing Rain,Haze	-4.9	-4.9	-7.5	-7.4	82.0	83.0	6	9	2.0
Freezing Rain,Ice Pellets,Fog	-2.6	-2.6	-3.7	-3.7	92.0	92.0	28	28	8.0
Freezing Rain,Snow Grains	-5.0	-5.0	-7.3	-7.3	84.0	84.0	32	32	4.8
Haze	-11.5	14.1	-16.0	11.1	68.0	86.0	0	17	4.8
Mainly Clear	-22.8	33.0	-28.0	21.2	20.0	99.0	0	63	12.9
Moderate Rain,Fog	1.7	1.7	0.8	0.8	94.0	94.0	17	17	6.4
Moderate Snow	-6.3	-4.9	-7.6	-6.7	83.0	93.0	26	39	0.6
Moderate Snow,Blowing Snow	-5.5	-5.4	-6.6	-6.4	92.0	93.0	39	41	0.6
Mostly Cloudy	-23.2	32.4	-28.5	24.4	18.0	100.0	0	83	11.3
Rain	0.3	22.8	-5.7	20.4	40.0	99.0	0	52	4.0
Rain Showers	1.6	26.4	-7.2	23.0	37.0	97.0	0	41	6.4
Rain Showers,Fog	12.8	12.8	12.1	12.1	96.0	96.0	13	13	6.4
Rain Showers,Snow	2.1	2.2	-1.8	-1.2	75.0	78.0	17	28	19.3

Showers										
Rain,Fog	0.0	21.7	-1.2	19.5	83.0	100.0	0	46	2.0	
Rain,Haze	4.0	5.5	1.0	2.9	81.0	86.0	7	17	4.0	
Rain,Ice Pellets	0.6	0.6	-0.6	-0.6	92.0	92.0	24	24	9.7	
Rain,Snow	0.6	1.7	-1.7	0.5	81.0	94.0	13	52	2.4	
Rain,Snow Grains	1.9	1.9	-2.1	-2.1	75.0	75.0	26	26	25.0	
Rain,Snow,Fog	0.8	0.8	0.3	0.3	96.0	96.0	9	9	6.4	
Rain,Snow,Ice Pellets	0.9	1.3	-0.7	0.1	88.0	94.0	17	28	4.8	
Snow	-16.7	3.7	-24.6	0.3	41.0	96.0	0	57	1.0	
Snow Pellets	0.7	0.7	-6.4	-6.4	59.0	59.0	35	35	2.4	
Snow Showers	-13.3	2.9	-19.3	-0.7	52.0	94.0	0	37	2.4	
Snow Showers,Fog	-11.3	-10.0	-12.7	-11.1	89.0	92.0	7	22	4.0	
Snow,Blowing Snow	-12.0	-1.4	-16.2	-2.9	70.0	91.0	24	48	0.6	
Snow,Fog	-10.1	1.1	-12.0	0.8	77.0	99.0	4	35	1.2	
Snow,Haze	-4.3	-3.6	-7.2	-6.4	80.0	81.0	0	15	4.0	
Snow,Ice Pellets	-4.3	0.8	-5.9	-1.7	76.0	92.0	19	33	2.8	
Thunderstorms	21.6	26.7	19.4	20.1	67.0	87.0	0	15	24.1	
Thunderstorms,Heavy Rain Showers	10.9	10.9	9.0	9.0	88.0	88.0	9	9	2.4	
Thunderstorms,Moderate Rain Showers,Fog	19.6	19.6	18.5	18.5	93.0	93.0	15	15	3.2	
Thunderstorms,Rain	19.4	21.3	18.2	19.1	83.0	93.0	4	30	16.1	
Thunderstorms,Rain Showers	11.0	25.5	7.0	23.1	68.0	98.0	7	32	6.4	
Thunderstorms,Rain Showers,Fog	19.5	22.9	16.1	21.3	80.0	91.0	7	35	9.7	
Thunderstorms,Rain,Fog	20.6	20.6	18.6	18.6	88.0	88.0	19	19	4.8	