In [1]:
```python
import numpy as np
import pandas as pd
df=pd.read_csv("weatherdata.csv")
print(df)
```

```
             Date/Time  Temp_C  Dew Point Temp_C  Rel Hum_%  Wind Speed_km/h  \
0     01-01-2012 00:00    -1.8              -3.9         86                4
1     01-01-2012 01:00    -1.8              -3.7         87                4
2     01-01-2012 02:00    -1.8              -3.4         89                7
3     01-01-2012 03:00    -1.5              -3.2         88                6
4     01-01-2012 04:00    -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
8781            4.8      99.95                  Snow
8782            9.7      99.91                  Snow
8783           11.3      99.89                  Snow

[8784 rows x 8 columns]
```

In [4]:
```python
import pandas as pd
df=pd.read_csv("weatherdata.csv")

# Q1Extract first 10 lines of thhe datsets
print("First 10 lines of thhe datsets\n",df.head(10))
```

```
First 10 lines of thhe datsets
           Date/Time  Temp_C  Dew Point Temp_C  Rel Hum_%  Wind Speed_km/h  \
0  01-01-2012 00:00    -1.8              -3.9         86                4
1  01-01-2012 01:00    -1.8              -3.7         87                4
2  01-01-2012 02:00    -1.8              -3.4         89                7
3  01-01-2012 03:00    -1.5              -3.2         88                6
4  01-01-2012 04:00    -1.5              -3.3         88                7
5  01-01-2012 05:00    -1.4              -3.3         87                9
6  01-01-2012 06:00    -1.5              -3.1         89                7
7  01-01-2012 07:00    -1.4              -3.6         85                7
8  01-01-2012 08:00    -1.4              -3.6         85                9
9  01-01-2012 09:00    -1.3              -3.1         88               15

   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
5            6.4     101.27                   Fog
6            6.4     101.29                   Fog
7            8.0     101.26                   Fog
8            8.0     101.23                   Fog
9            4.0     101.20                   Fog
```

In [18]:
```python
#Q1 Convert the 'Date/Time' column to datetime format
df['Date/Time'] = pd.to_datetime(df['Date/Time'])
print("The dataset 'Date/Time' column to datetime format is:\n ",df['Date/Time'])
```

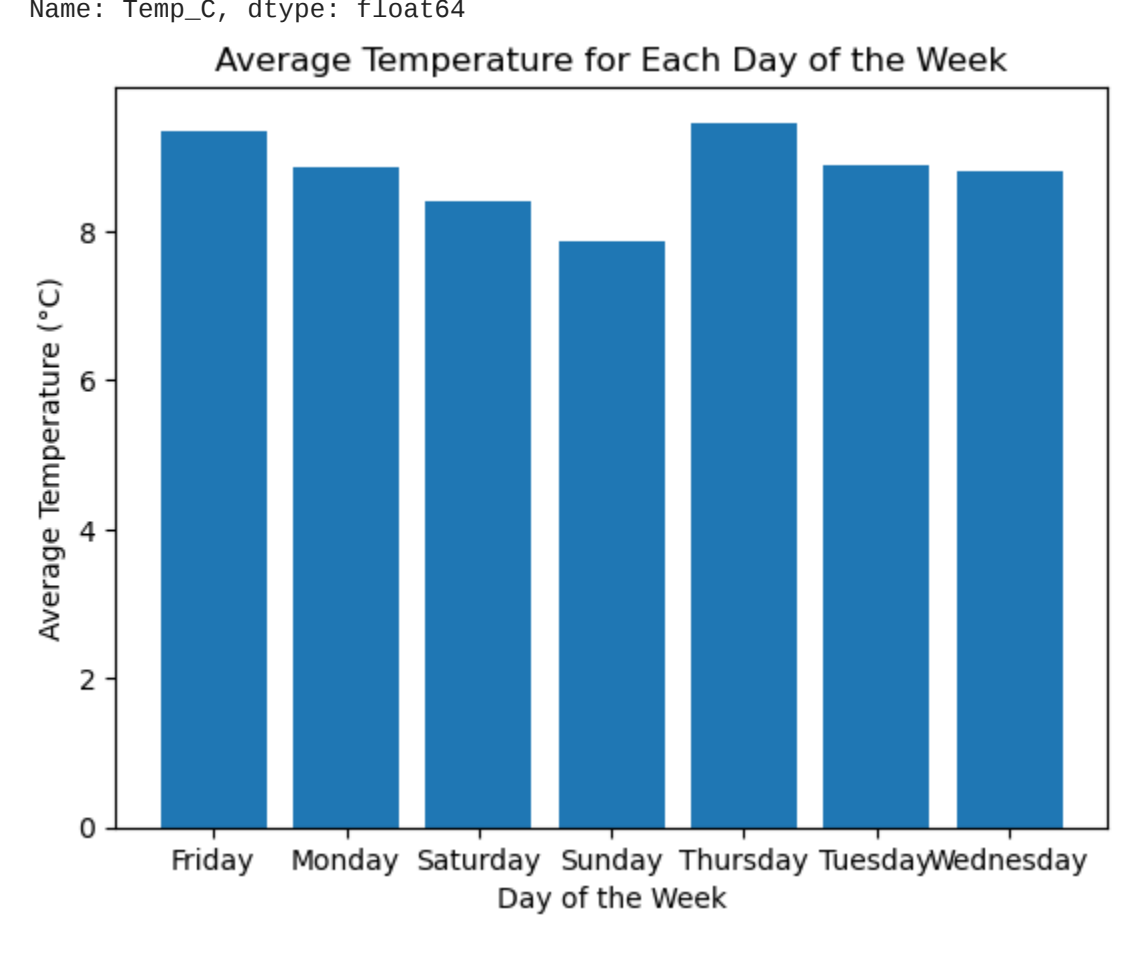
```
The dataset 'Date/Time' column to datetime format is:
  0      2012-01-01 00:00:00
1      2012-01-01 01:00:00
2      2012-01-01 02:00:00
3      2012-01-01 03:00:00
4      2012-01-01 04:00:00
               ...
8779   2012-12-31 19:00:00
8780   2012-12-31 20:00:00
8781   2012-12-31 21:00:00
8782   2012-12-31 22:00:00
8783   2012-12-31 23:00:00
Name: Date/Time, Length: 8784, dtype: datetime64[ns]
```

In [17]:
```python
#Q2 Extract month from the 'Date/Time' column
df['Month'] = df['Date/Time'].dt.month
print("Only months from 'Date/Time' column are :\n",df['Month'])
```

```
Only months from 'Date/Time' column are :
 0       1
1       1
2       1
3       1
4       1
        ..
8779    12
8780    12
8781    12
8782    12
8783    12
Name: Month, Length: 8784, dtype: int64
```

In [3]:
```python
#Q3 Find the average temperature (Temp_C) for each day of the week and plot it in a bar chart.
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("weatherdata.csv")
df['Date/Time']=pd.to_datetime(df['Date/Time'])
df['dayofweek']=df['Date/Time'].dt.day_name()
print(df['dayofweek'])

avg_tem_day=df.groupby('dayofweek')['Temp_C'].mean()
print(avg_tem_day)


df['Date/Time'] = pd.to_datetime(df['Date/Time'], format='%m-%d-%Y %H:%M')

df['Day_of_Week'] = df['Date/Time'].dt.day_name()

average_temp = df.groupby('Day_of_Week')['Temp_C'].mean()

plt.bar(average_temp.index, average_temp)
plt.xlabel('Day of the Week')
plt.ylabel('Average Temperature (°C)')
plt.title('Average Temperature for Each Day of the Week')
plt.show()
```

```
0       Sunday
1       Sunday
2       Sunday
3       Sunday
4       Sunday
         ...
8779    Monday
8780    Monday
8781    Monday
8782    Monday
8783    Monday
Name: dayofweek, Length: 8784, dtype: object
dayofweek
Friday       9.353766
Monday       8.853931
Saturday     8.388622
Sunday       7.854560
Thursday     9.456731
Tuesday      8.889583
Wednesday    8.806891
Name: Temp_C, dtype: float64
```



In [ ]:
```python
#Q4
```

In [49]:
```python
#Days when the weather is cloudy
w_cloud=df[df['Weather']=='Cloudy']['Date/Time']
print("Time when the weather is cloudy :\n",w_cloud)
```

```
Time when the weather is cloudy :
 17     2012-01-01 17:00:00
20     2012-01-01 20:00:00
21     2012-01-01 21:00:00
23     2012-01-01 23:00:00
25     2012-01-02 01:00:00
               ...
8761   2012-12-31 01:00:00
8762   2012-12-31 02:00:00
8764   2012-12-31 04:00:00
8765   2012-12-31 05:00:00
8766   2012-12-31 06:00:00
Name: Date/Time, Length: 1728, dtype: datetime64[ns]
```

In [36]:
```python
print(df.head(168))
```

```
              Date/Time  Temp_C  Dew Point Temp_C  Rel Hum_%  Wind Speed_km/h  \
0   2012-01-01 00:00:00    -1.8              -3.9         86                4
1   2012-01-01 01:00:00    -1.8              -3.7         87                4
2   2012-01-01 02:00:00    -1.8              -3.4         89                7
3   2012-01-01 03:00:00    -1.5              -3.2         88                6
4   2012-01-01 04:00:00    -1.5              -3.3         88                7
..                  ...     ...               ...        ...              ...
163 2012-01-07 19:00:00    -3.0              -5.0         86                6
164 2012-01-07 20:00:00    -1.8              -3.7         87                4
165 2012-01-07 21:00:00    -1.3              -3.1         88                0
166 2012-01-07 22:00:00    -1.5              -3.0         89                4
167 2012-01-07 23:00:00     0.4              -1.5         87               22

     Visibility_km  Press_kPa               Weather  Month dayofweek
0              8.0     101.24                   Fog      1    Sunday
1              8.0     101.24                   Fog      1    Sunday
2              4.0     101.26  Freezing Drizzle,Fog      1    Sunday
3              4.0     101.27  Freezing Drizzle,Fog      1    Sunday
4              4.8     101.23                   Fog      1    Sunday
..             ...        ...                   ...    ...       ...
163            8.0     100.41                  Snow      1  Saturday
164            9.7     100.48                  Snow      1  Saturday
165            6.4     100.50                  Snow      1  Saturday
166            4.0     100.54                   Fog      1  Saturday
167            9.7     100.62                  Snow      1  Saturday

[168 rows x 10 columns]
```

In [61]:
```python
#At what time was the wind speed is above 20
w_speed_grt=df[df['Wind Speed_km/h']>45]['Date/Time']
print("The timings when the speed is greater than 45 are :\n",w_speed_grt.to_string(index=False))
```

```
The timings when the speed is greater than 45 are :
 2012-01-02 12:00:00
2012-01-18 01:00:00
2012-01-18 02:00:00
2012-01-18 03:00:00
2012-01-18 04:00:00
2012-01-18 05:00:00
2012-01-28 23:00:00
2012-01-29 00:00:00
2012-01-29 01:00:00
2012-01-29 02:00:00
2012-03-03 13:00:00
2012-03-03 14:00:00
2012-03-03 15:00:00
2012-03-03 16:00:00
2012-03-03 17:00:00
2012-03-03 18:00:00
2012-03-03 20:00:00
2012-03-03 21:00:00
2012-03-03 22:00:00
2012-03-08 18:00:00
2012-03-08 22:00:00
2012-04-17 07:00:00
2012-04-23 00:00:00
2012-04-23 03:00:00
2012-04-23 05:00:00
2012-04-23 06:00:00
2012-04-23 07:00:00
2012-04-23 08:00:00
2012-04-23 10:00:00
2012-04-23 11:00:00
2012-04-23 12:00:00
2012-10-29 16:00:00
2012-10-29 17:00:00
2012-10-29 19:00:00
2012-10-29 20:00:00
2012-10-29 21:00:00
2012-10-29 22:00:00
2012-10-29 23:00:00
2012-10-30 01:00:00
2012-11-13 00:00:00
2012-12-21 17:00:00
2012-12-21 18:00:00
2012-12-27 13:00:00
2012-12-27 14:00:00
```

In [65]:
```python
# when visibility is below 10
w_speed_grt=df[df['Visibility_km']<.5]['Date/Time']
print("The timings when the visibility is below .5 km  are :\n",w_speed_grt.to_string(index=False))
```

```
The timings when the visibility is below .5 km  are :
 2012-03-17 02:00:00
2012-03-17 03:00:00
2012-03-17 04:00:00
2012-03-17 06:00:00
2012-03-17 07:00:00
2012-03-17 08:00:00
2012-03-17 09:00:00
2012-03-17 10:00:00
2012-05-02 02:00:00
2012-05-02 03:00:00
2012-10-20 06:00:00
```

In [71]:
```python
#When the temperature is below 0 & weather  is Freezing Drizzle,Fog
temp_bel_free=df[( df['Temp_C']<0) & (df['Weather']=='Freezing Drizzle,Fog')]['Date/Time']
print("The timings when the temperature is below 0 & weather  is Freezing Drizzle,Fog are:")
print(temp_bel_free.to_string(index=False))
```

```
The timings when the temperature is below 0 & weather  is Freezing Drizzle,Fog are:
2012-01-01 02:00:00
2012-01-01 03:00:00
2012-01-07 08:00:00
2012-01-27 16:00:00
2012-12-10 04:00:00
2012-12-10 05:00:00
```

In [73]:
```python
#Maximum temperature recorded in the dataset
print("Maximumm temperature recorded in the dataset is:",df['Temp_C'].max(),"°C")
```

```
Maximumm temperature recorded in the dataset is: 33.0 °C
```

In [75]:
```python
#What is the avg visibility during 'Fog' weather conndition

avg_vis_fog=df[df['Weather']=='Fog']['Visibility_km'].mean()

print("Avg visibility during 'Fog' weather condition is:",avg_vis_fog)
```

```
Avg visibility during 'Fog' weather condition is: 6.248
```

In [77]:
```python
# How many hours experienced Freezing Drizzle Weather
fdw=df[df['Weather']=='Freezing Drizzle']['Date/Time'].value_counts().sum()
print("The hours in which experienced Freezing Drizzle Weather are ",fdw,"Hours")
```

```
The hours in which experienced Freezing Drizzle Weather are  7 Hours
```

In [9]:
```python
# Count the occurrences of each weather condition for a specific month.
import pandas as pd
df=pd.read_csv('weatherdata.csv')
# Convert the 'Date/Time' column to datetime format
df['Date/Time'] = pd.to_datetime(df['Date/Time'])

# Extract the month from the 'Date/Time' column
df['Month'] = df['Date/Time'].dt.month

# Specify the month for analysis
target_month = 1

# Filter the dataset for the specified month
target_data = df[df['Month'] == target_month]

# Count the occurrences of each weather condition
weather_counts = target_data['Weather'].value_counts()

print("For the first month of 2012 Weather is follwing way:\n ",weather_counts)
```

```
For the first month of 2012 Weather is follwing way:
  Mostly Cloudy               164
Snow                        148
Cloudy                      145
Mainly Clear                117
Clear                        77
Fog                          18
Snow Showers                 16
Rain                         10
Freezing Rain                 9
Drizzle,Fog                   6
Freezing Drizzle,Snow         5
Snow,Blowing Snow             5
Freezing Drizzle,Fog          4
Freezing Drizzle              4
Rain,Fog                      3
Rain Showers                  2
Freezing Rain,Fog             2
Rain,Snow                     2
Moderate Snow                 2
Haze                          2
Freezing Rain,Snow Grains     1
Freezing Fog                  1
Drizzle                       1
Name: Weather, dtype: int64
```

In [12]:
```python
#Find the date and time with the lowest visibility.
# Find the row with the lowest visibility
low_vis_row = df.loc[df['Visibility_km'].idxmin()]

print("The date & timing when there is lowest visibility is: ",low_vis_row['Date/Time'])
```

```
The date & timing when there is lowest visibility is:  2012-03-17 02:00:00
```

In [16]:
```python
#Determine the most common weather condition during nighttime (between 6 PM and 6 AM).
# Convert the 'Date/Time' column to datetime format
df['Date/Time'] = pd.to_datetime(df['Date/Time'])

# Extract the hour from the 'Date/Time' column
df['Hour'] = df['Date/Time'].dt.hour

# Filter the dataset for nighttime hours
night_data = df[(df['Hour'] >= 18) | (df['Hour'] < 6)]

# Determine the most common weather condition during nighttime
most_common_weather_nighttime = night_data['Weather'].value_counts().idxmax()

print("The most commonn weather condition  during nightime (between 6 PM and 6 AM) is: ",most_common_weather_nighttime)
```

```
The most commonn weather condition  during nightime (between 6 PM and 6 AM) is:  Clear
```

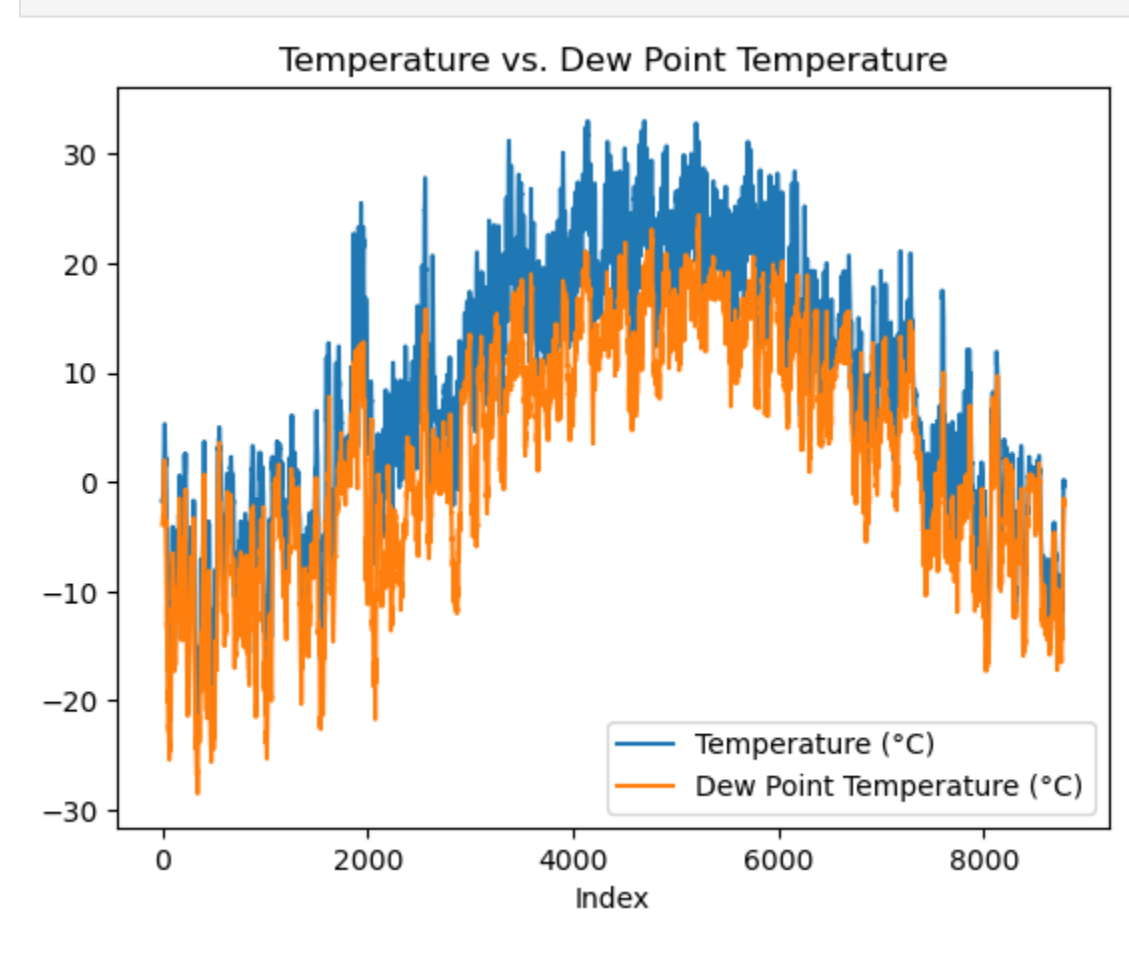
In [20]:
```python
#What is the most common wind speed value in the dataset?

# Find the most common wind speed value
most_co = df['Wind Speed_km/h'].mode().iloc[0]

print("Most common wind speed  is:", most_co, "km/h")
```

```
Most common wind speed  is: 9 km/h
```

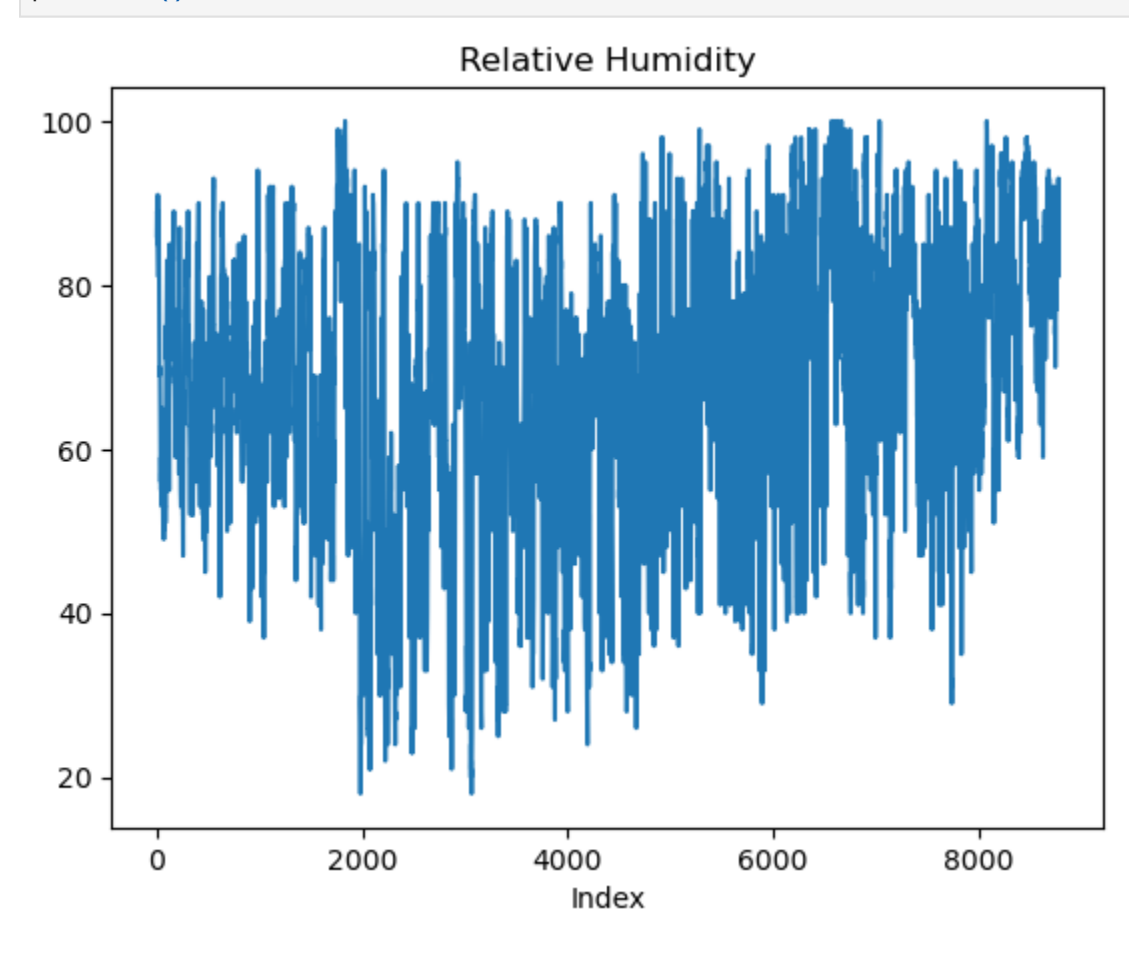
In [21]:
```python
#How many unique weather conditions are there in the dataset?
# Count the number of unique weather conditions
num_weather = df['Weather'].nunique()

print("Number of unique weather conditions:", num_weather)
```

```
Number of unique weather conditions: 50
```

In [3]:
```python
# Plot the graph for comparing Temperature with Dew Point Temperature
import matplotlib.pyplot as plt
import pandas as pd
df=pd.read_csv("weatherdata.csv")

plt.plot(df['Temp_C'], label='Temperature (°C)')
plt.plot(df['Dew Point Temp_C'], label='Dew Point Temperature (°C)')
plt.legend()
plt.title('Temperature vs. Dew Point Temperature')
plt.xlabel('Index')
plt.show()
```

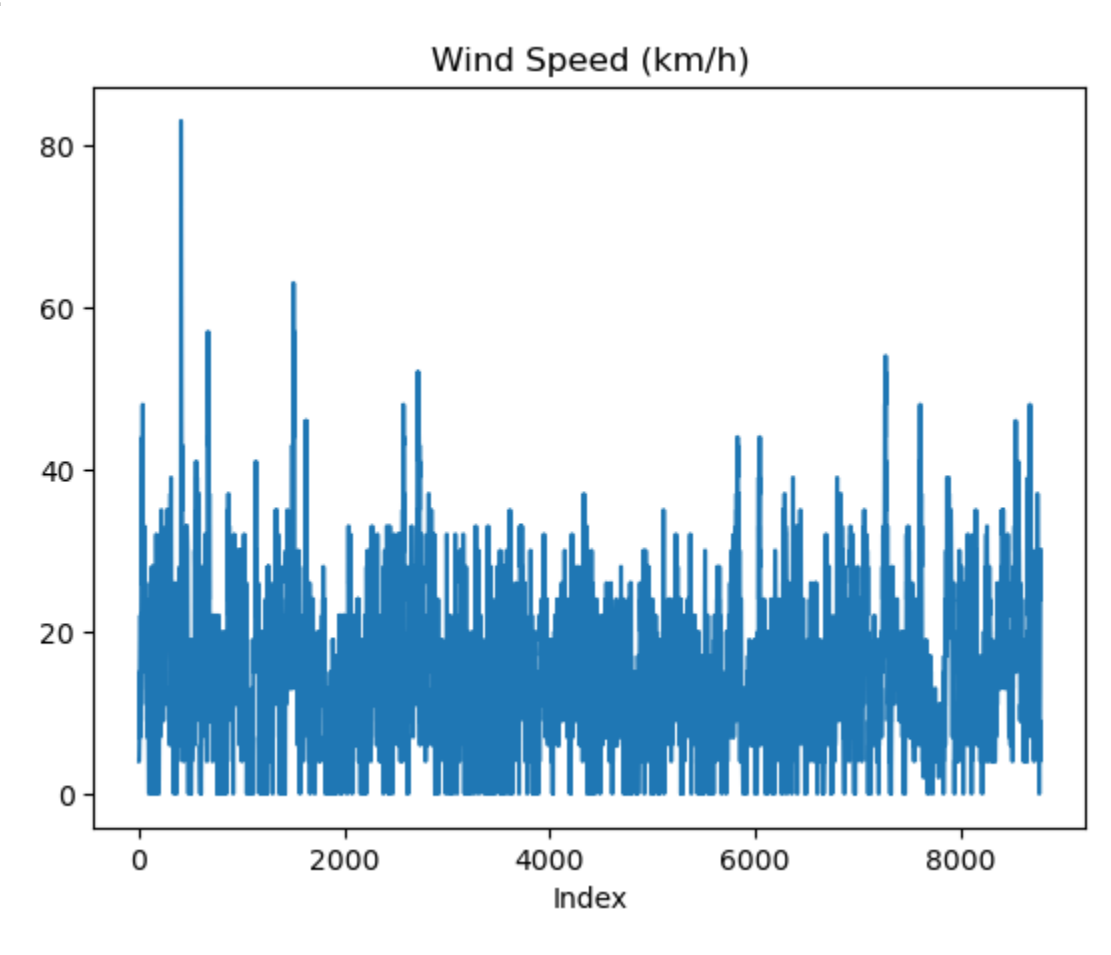


In [5]:
```python
# Plot the graph showing the change  in Relative Humidity
plt.plot(df['Rel Hum_%'])
plt.title('Relative Humidity')
plt.xlabel('Index')
plt.show()
```

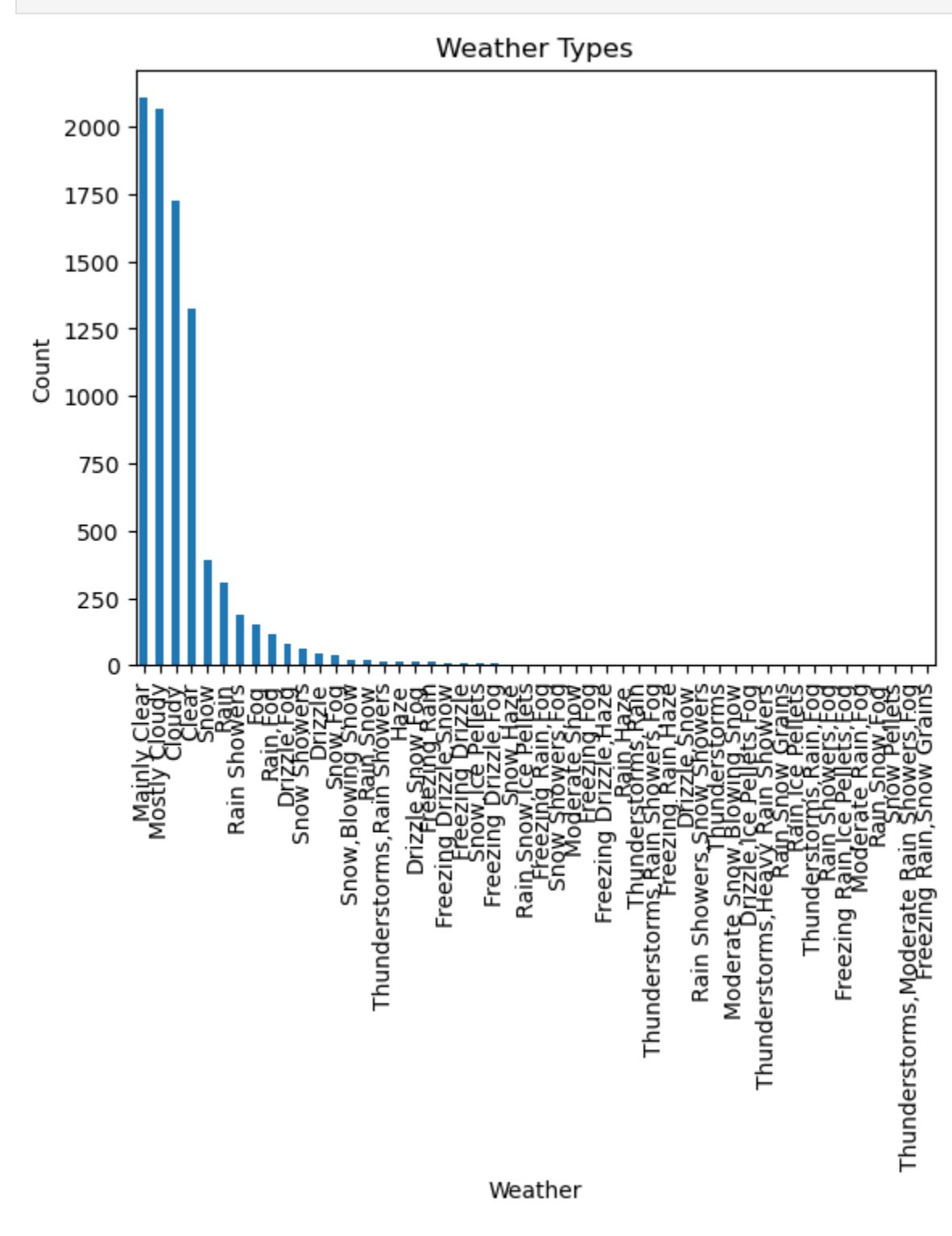


In [7]:
```python
# Plot the graph  wind speed of all the time dataset
plt.plot(df['Wind Speed_km/h'])
plt.title('Wind Speed (km/h)')
plt.xlabel('Index')
```

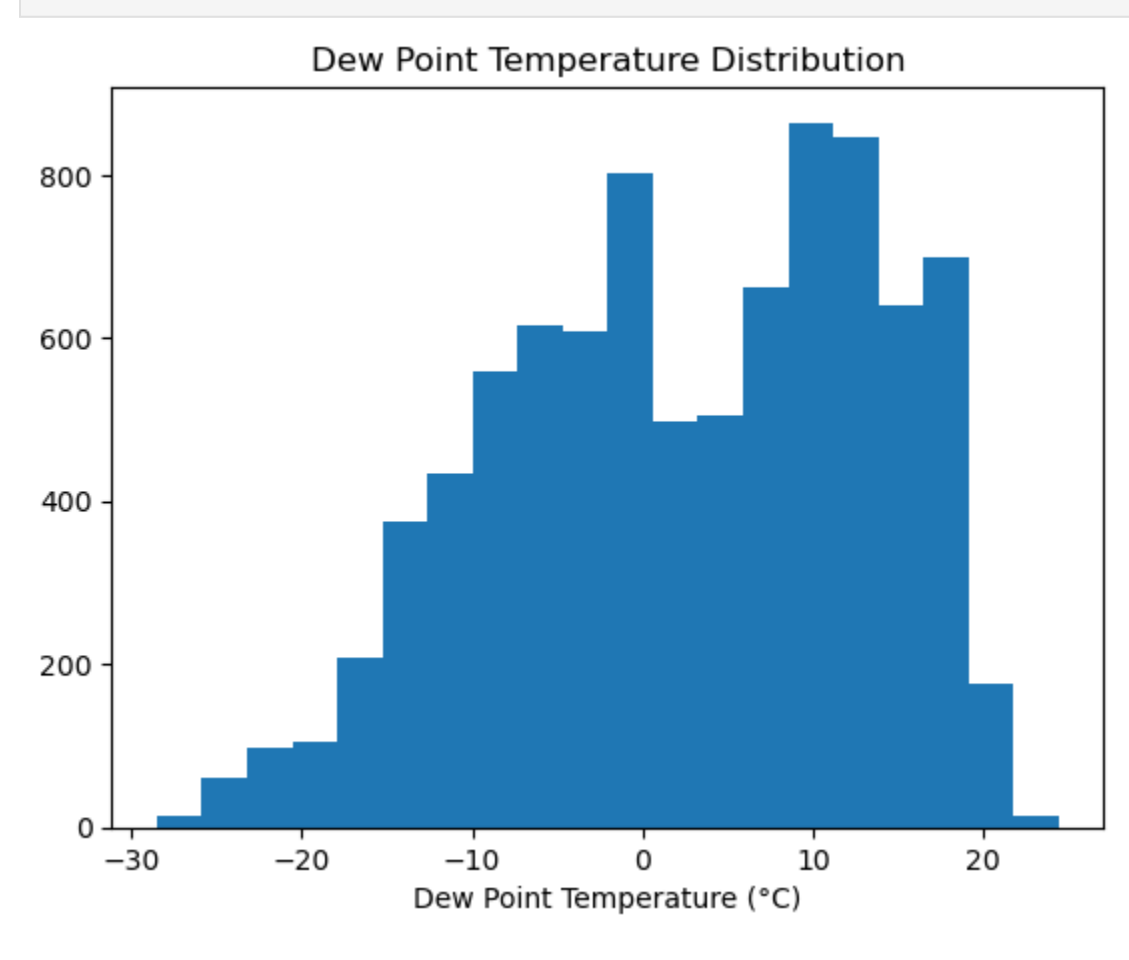
Out[7]:
```
Text(0.5, 0, 'Index')
```



In [8]:
```python
# Plot the graph as per count of each weather data type
weather_counts = df['Weather'].value_counts().sort_values(ascending=False)
weather_counts.plot(kind='bar')
plt.title('Weather Types')
plt.xlabel('Weather')
plt.ylabel('Count')
plt.show()
```



In [12]:
```python
# Show graphically Dew Point Temperature Distribution

plt.hist(df['Dew Point Temp_C'], bins=20)
plt.title('Dew Point Temperature Distribution')
plt.xlabel('Dew Point Temperature (°C)')
plt.show()
```

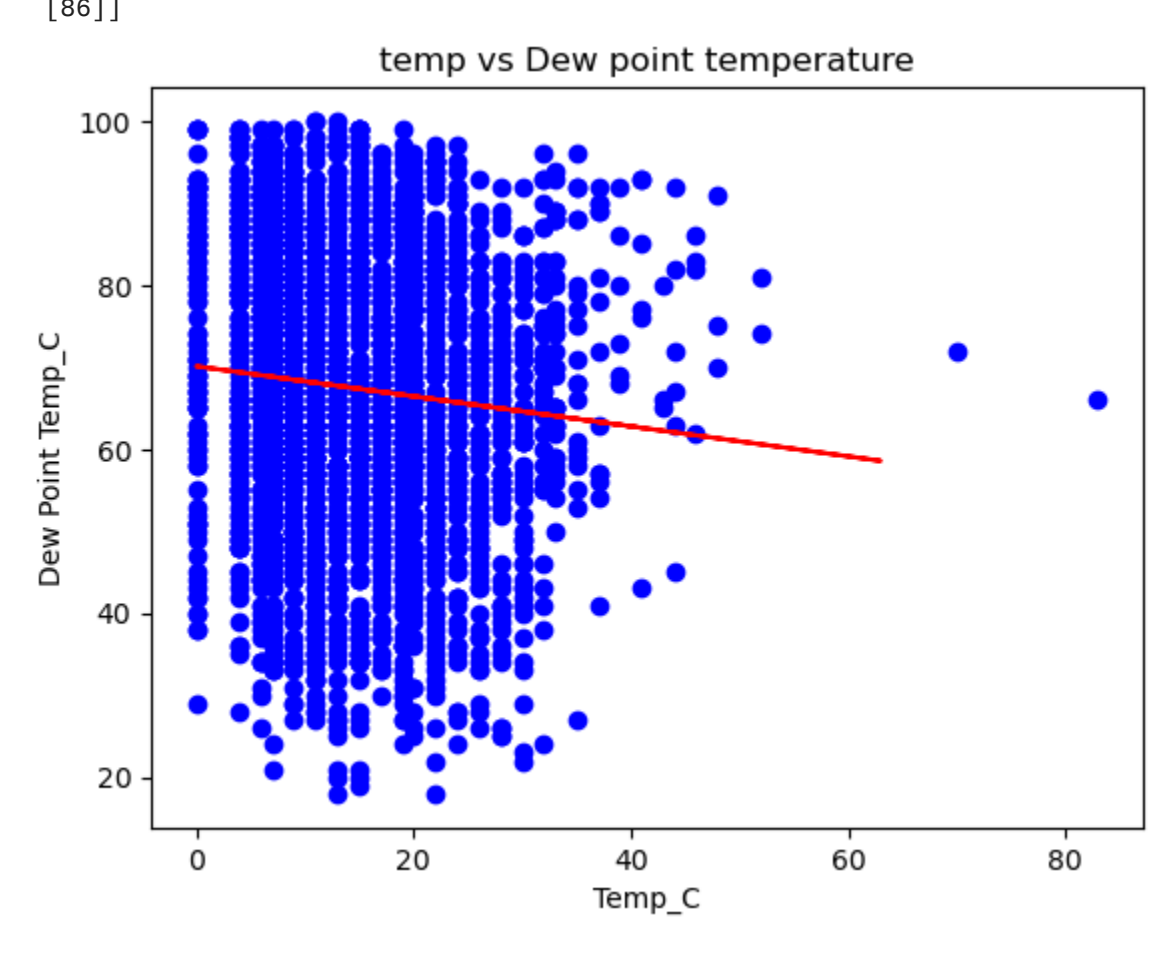


In [1]:
```python
x= df['Weather'].iloc[:, :-1].values
y= data_set.iloc[:, 1].values
Fitting the Simple Linear Regression model to the training dataset
from sklearn.linear_model import LinearRegression
regressor= LinearRegression()
regressor.fit(x_train, y_train)
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 1/3, random_state=0)
```

```
  Cell In[1], line 3
    Fitting the Simple Linear Regression model to the training dataset
            ^
SyntaxError: invalid syntax
```

In [31]:
```python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
df=pd.read_csv("weatherdata.csv")
from sklearn.linear_model import LinearRegression

x= np.array(df['Wind Speed_km/h']).reshape(-1,1)
y= np.array(df['Rel Hum_%']).reshape(-1,1)
print(x)
print(y)
#print(y)

# Splitting the dataset into training and test set.
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 1/3, random_state=0)

#x_train = x_train.reshape(-1, 1)
#y_train = y_train.reshape(1, 1)

#Fitting the Simple Linear Regression model to the training dataset
from sklearn.linear_model import LinearRegression
regressor= LinearRegression()
regressor.fit(x_train, y_train)

#Prediction of Test and Training set result
y_pred= regressor.predict(x_test)
x_pred= regressor.predict(x_train)

#visualizing the Test set results
plt.scatter(x_test, y_test, color="blue")
plt.plot(x_train, x_pred, color="red")
plt.title("temp vs Dew point temperature")
plt.xlabel("Temp_C")
plt.ylabel("Dew Point Temp_C")
plt.show()
```

```
[[ 4]
 [ 4]
 [ 7]
 ...
 [28]
 [28]
 [30]]
[[86]
 [87]
 [89]
 ...
 [93]
 [89]
 [86]]
```



In [ ]:

In [ ]: