In [7]:

```
import pandas as pd
import numpy as np
import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
```

In [8]:

```
import pandas as pd
Weather_Data=pd.read_csv(r'C:\Users\your shop a\Downloads\Weather_Data.csv')
```

In [3]:

```
Weather_Data.head()
```

Out[3]:

	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

In [4]:

```
Weather_Data.info()
# this code print the entire information of the data such as no. of rows,no. pf columns,
```

```
<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
```

In [24]:

```
Weather_Data.describe()
# This code gives count,mean,standard deviation,minimum value,quartiles and maximum valu
```

Out[24]:

	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
max	33.000000	24.400000	100.000000	83.000000	48.300000	103.650000

In [25]:

```
!pip install scipy
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: scipy in c:\programdata\anaconda3\lib\site-packages (1.9.
1)
Requirement already satisfied: numpy<1.25.0,>=1.18.5 in c:\programdata\anaconda3\lib\sit
e-packages (from scipy) (1.21.5)
```

In [26]:

```
stats.hmean(Weather_Data['Rel Hum_%'])
```

Out[26]:

```
62.178866065947716
```

In [27]:

```
stats.normaltest(imdb['Rel Hum_%'])
```

Out[27]:

```
NormaltestResult(statistic=369.2998415057831, pvalue=6.420344518797528e-81)
```

In [29]:

```
stats.pearsonr(Weather_Data['Rel Hum_%'],Weather_Data['Visibility_km'])
```

Out[29]:

```
PearsonRResult(statistic=-0.6336831841537234, pvalue=0.0)
```

In [30]:

```
stats.iqr(Weather_Data['Rel Hum_%'])
```

Out[30]:

```
25.0
```

In [31]:

```
stats.skew(Weather_Data['Dew Point Temp_C'])
```

Out[31]:

```
-0.31837900621611687
```

In [32]:

```
stats.kurtosis(Weather_Data['Dew Point Temp_C'])
```

Out[32]:

```
-0.8160062908722221
```

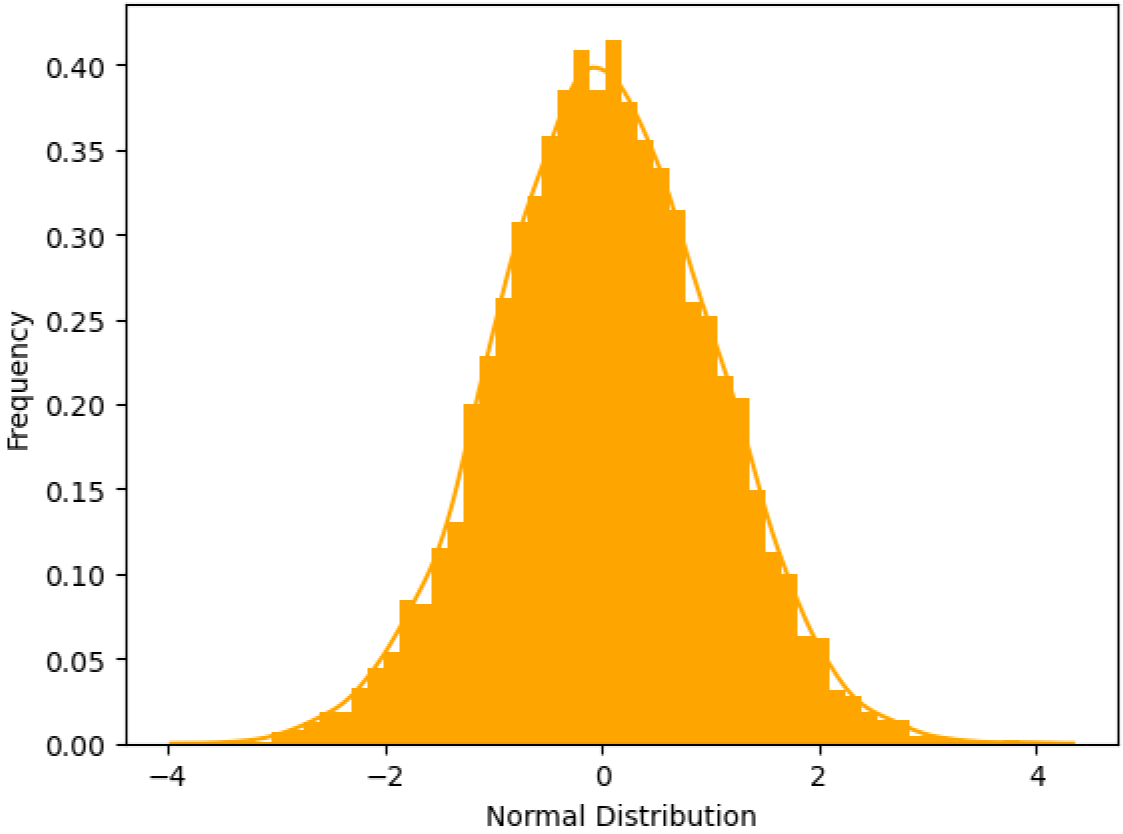
In [34]:

```
Weather_Data['Visibility_km'] =stats.norm(scale=1, loc=0).rvs(8784)
 #plotting a histogram
ax = sns.distplot(Weather_Data['Visibility_km'],
               bins=50,
               kde=True,
               color='orange',
               hist_kws={"linewidth":5,"alpha":1}) #kernel density estimator
ax.set(xlabel="Normal Distribution" , ylabel="Frequency")

plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

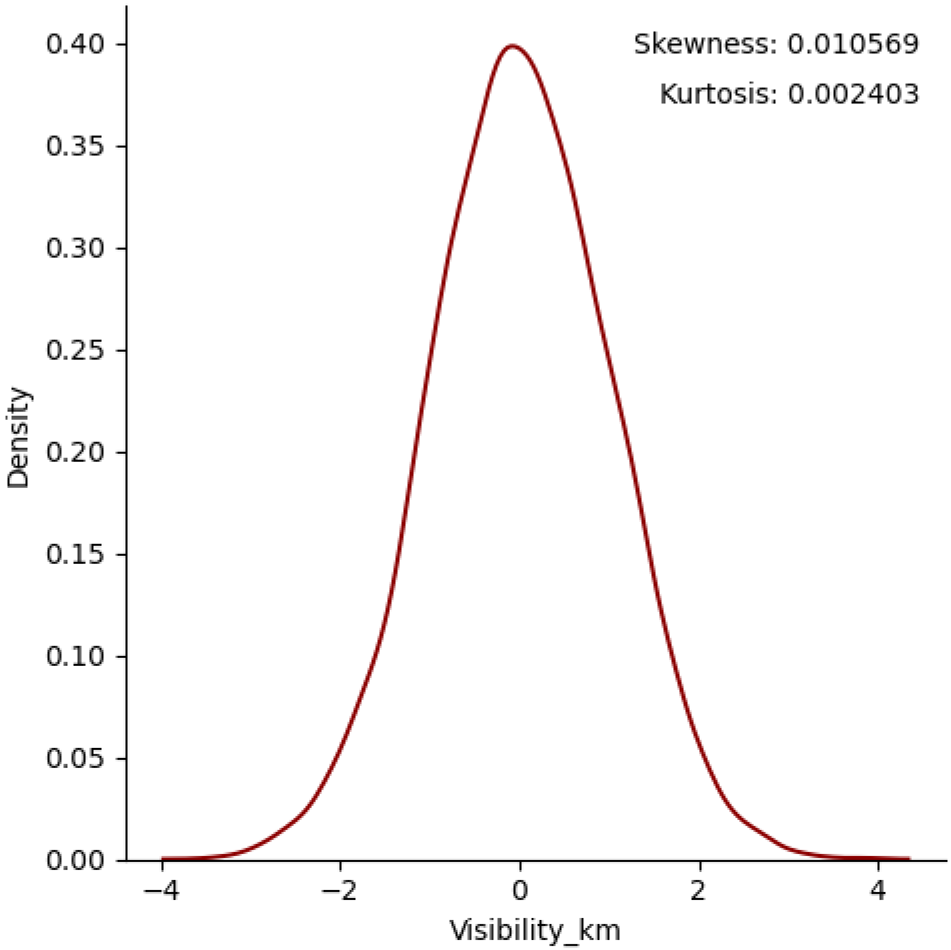
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [36]:

```
g = sns.displot(data=Weather_Data, x="Visibility_km",kind ='kde',color='darkred')
for ax in g.axes.ravel():
    ax.text(x=0.97, y=0.97, transform=ax.transAxes, s="Skewness: %f" % Weather_Data['Vis
        fontsize=10, verticalalignment='top', horizontalalignment='right')
    ax.text(x=0.97, y=0.91, transform=ax.transAxes, s="Kurtosis: %f" % Weather_Data['Vis
        fontsize=10, verticalalignment='top', horizontalalignment='right')
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



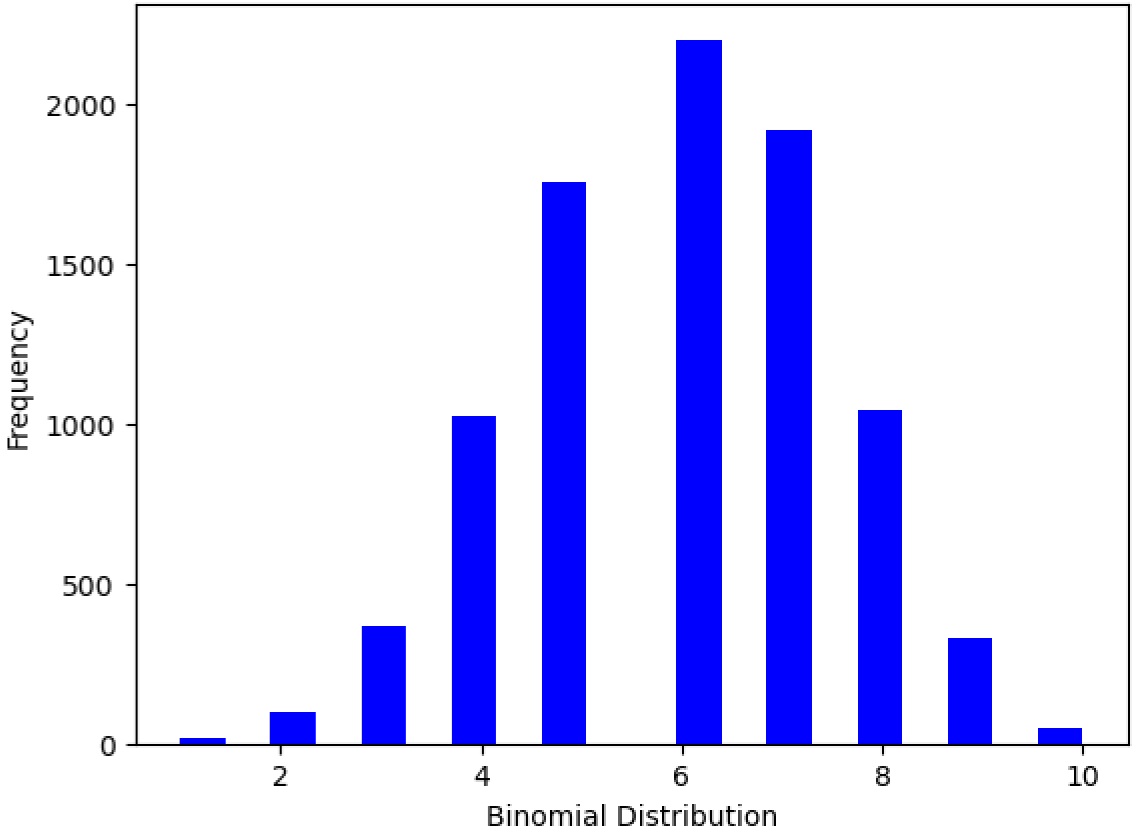
In [37]:

```
n, p = 10, .6
Weather_Data = np.random.binomial(n, p, 8784)
#plotting a histogram
ax = sns.distplot(Weather_Data,
                  bins=20,
                  kde=False,
                  color='Blue',
                  hist_kws={"linewidth": 15, 'alpha': 1})
ax.set(xlabel='Binomial Distribution', ylabel='Frequency')

plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



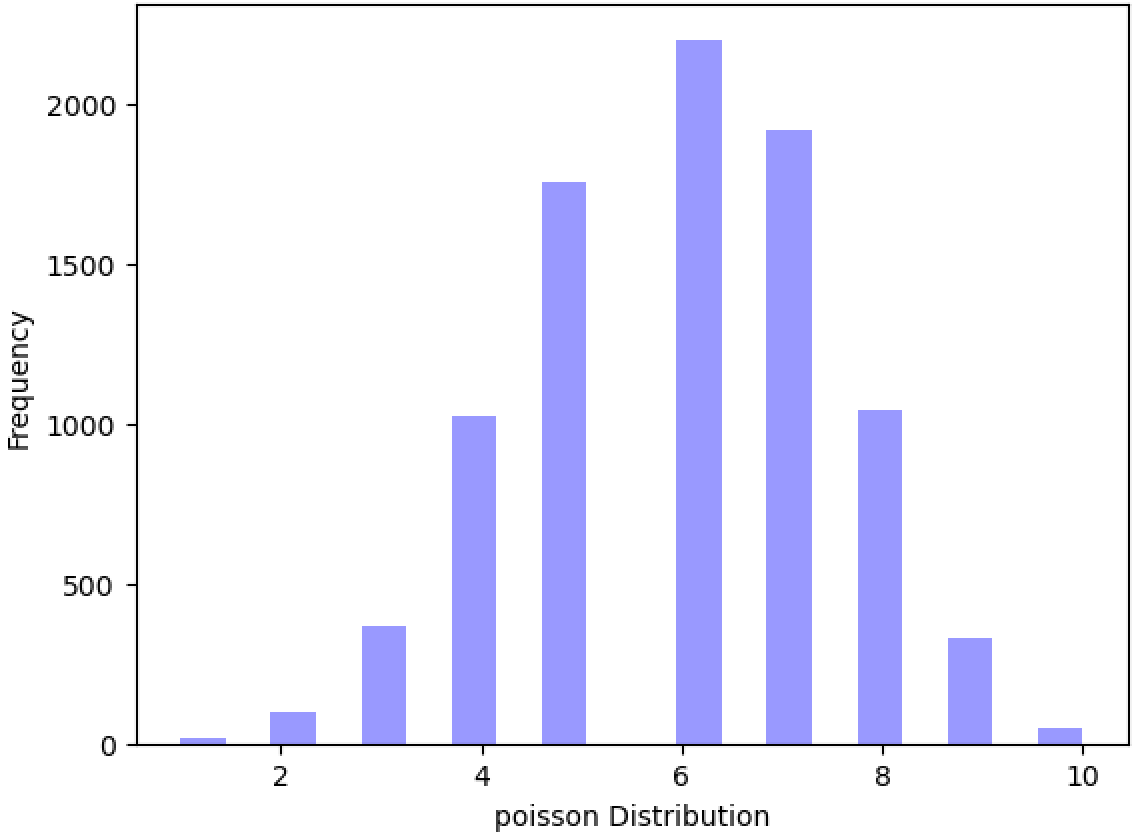
In [38]:

```python
#generate possion data
poisson_data = np.random.poisson(lam=5, size=1000)

#plotting a histogram
ax = sns.distplot(Weather_Data,
                  bins=20,
                  kde=False,
                  color='Blue',)
ax.set(xlabel='poisson Distribution', ylabel='Frequency')

plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [5]:

```
from scipy.stats import lognorm

numargs = lognorm.numargs
a, b = 4.32, 3.18
rv = lognorm(a, b)
print ("RV :\n", rv)
```

```
RV :
 <scipy.stats._distn_infrastructure.rv_continuous_frozen object at 0x0000018ADAD05E80>
```

In [40]:

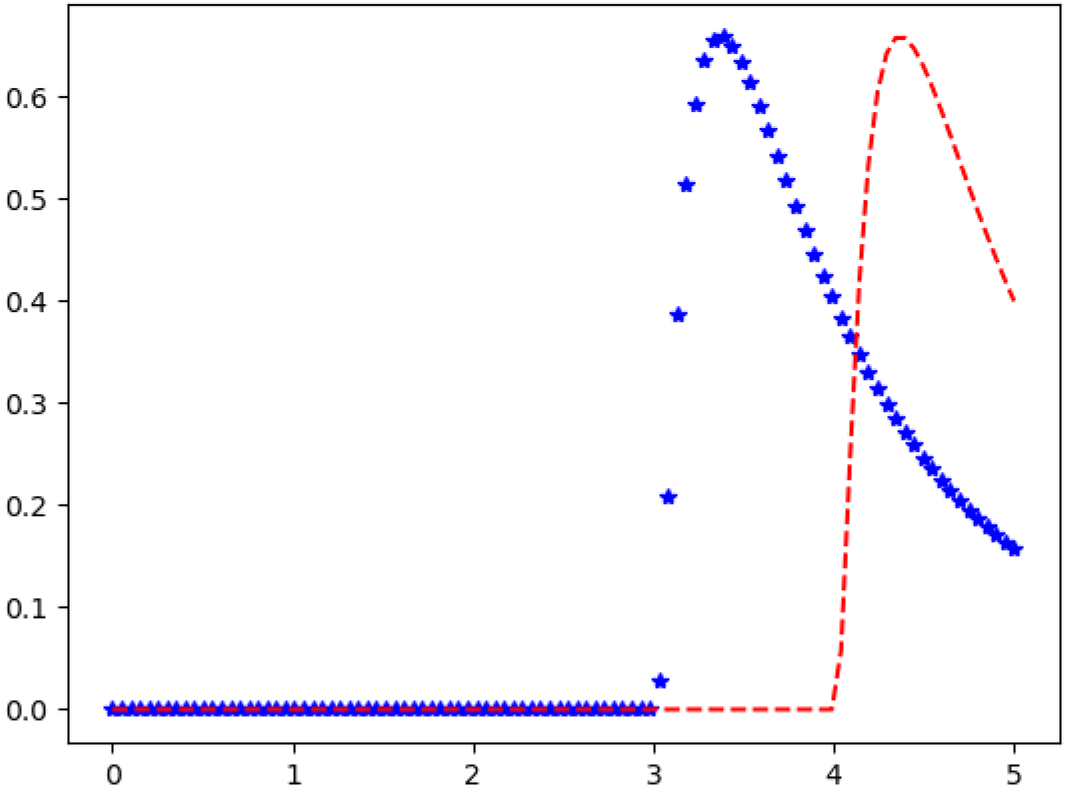
```
x = np.linspace(0, 5, 100)

# varying positional arguments
y1 = lognorm .pdf(x, 1, 3)
y2 = lognorm .pdf(x, 1, 4)
plt.plot(x, y1, "b*", x, y2, "r--")
```

Out[40]:

```
[<matplotlib.lines.Line2D at 0x225bf928af0>,
 <matplotlib.lines.Line2D at 0x225bf928be0>]
```

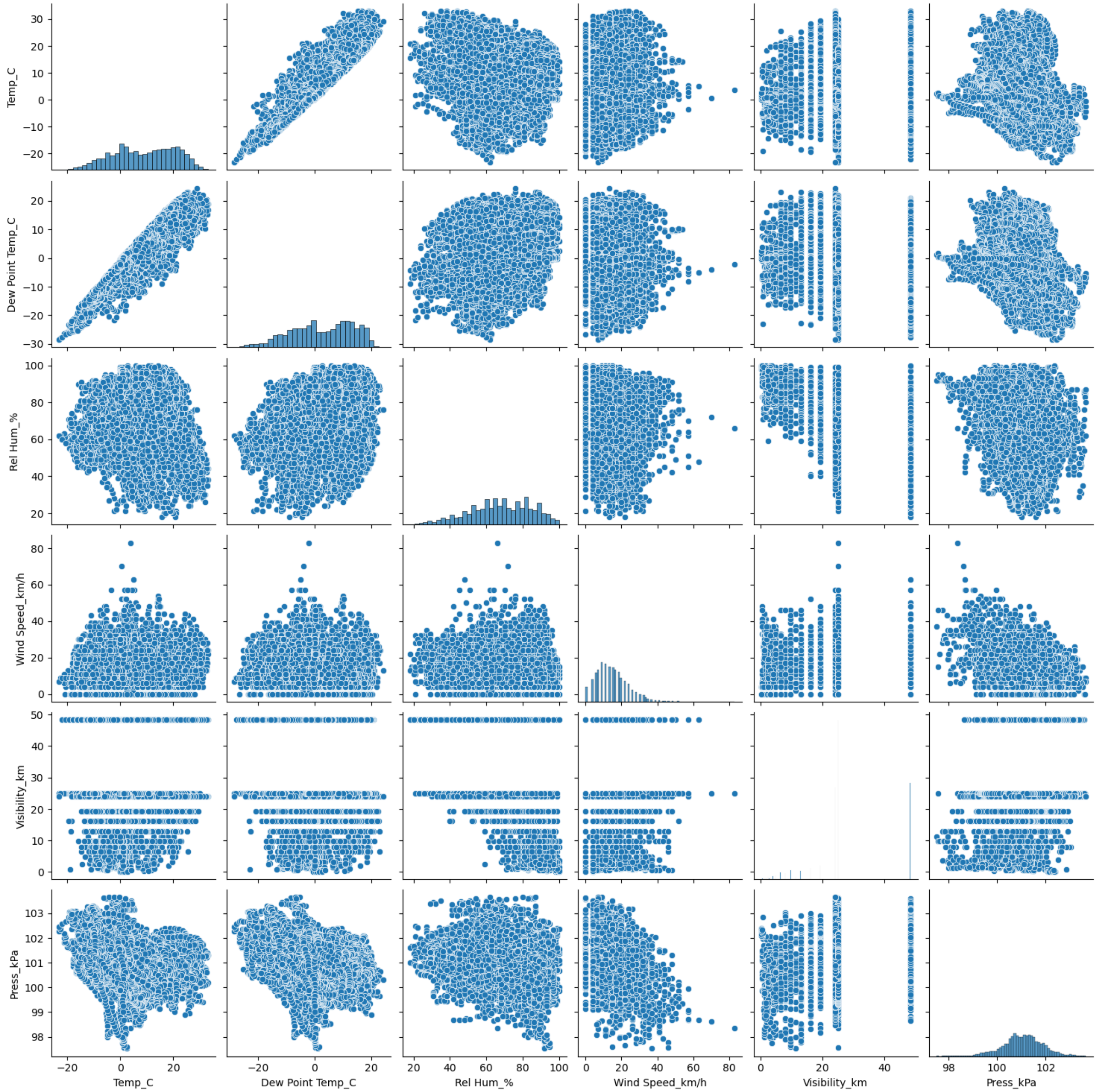
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [45]:
```
sns.pairplot(Weather_Data)
```

Out[45]:
```
<seaborn.axisgrid.PairGrid at 0x225bf95f100>
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



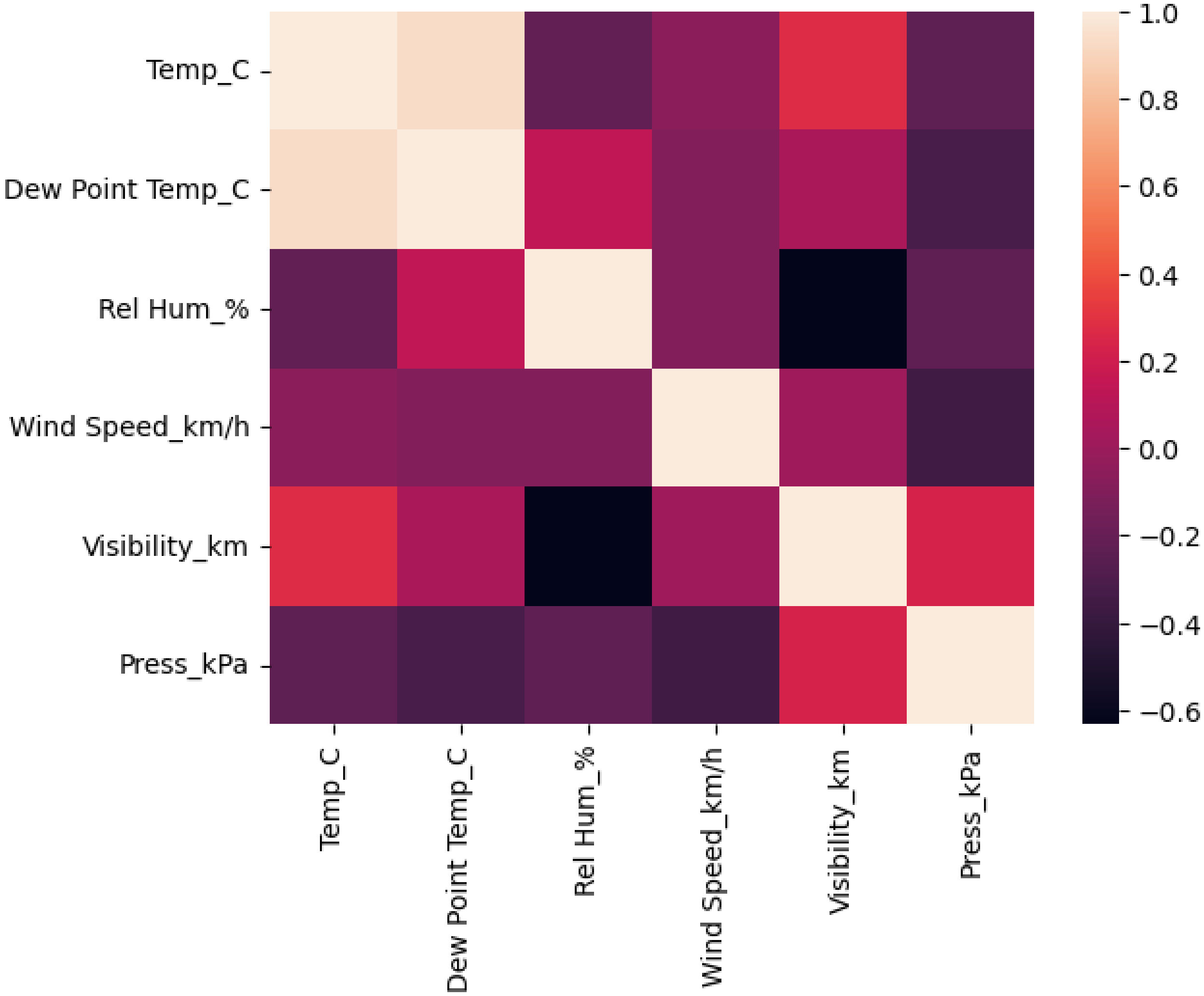
In [50]:

```
sns.heatmap(Weather_Data.corr())
```

Out[50]:

```
<AxesSubplot:>
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [12]:
```
print("1...correlation between Press_kPa  and Dew Point Temp_C
      variable:",stats.stats.spearmanr(Weather_Data['Press_kPa'],Weather_Data['Dew Point
```

```
1...correlation between Press_kPa  and Dew Point Temp_C variable: SpearmanrResult(correl
ation=-0.33614332519724605, pvalue=5.653441268885604e-231)
```

```
C:\Users\your shop a\AppData\Local\Temp\ipykernel_6916\2229521179.py:1: DeprecationWarni
ng: Please use `spearmanr` from the `scipy.stats` namespace, the `scipy.stats.stats` nam
espace is deprecated.
  print("1...correlation between Press_kPa  and Dew Point Temp_C variable:",stats.stats.
spearmanr(Weather_Data['Press_kPa'],Weather_Data['Dew Point Temp_C']))
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js