Weather Data Visualization using Python

By Sarthak Sarode

In [2]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
sns.set(color_codes=True)
```

In [3]:

```
weather = pd.read_csv('Test.csv')
weather
```

Out[3]:

_pollution_index	humidity	wind_speed	wind_direction	visibility_in_miles	dew_point	temperature
73.0	63.0	1.0	27.0	4.0	4.0	285.15
251.0	63.0	1.0	27.0	4.0	4.0	285.15
75.0	56.0	1.0	0.0	1.0	1.0	285.15
98.0	56.0	1.0	351.0	2.0	2.0	284.79
283.0	56.0	1.0	351.0	1.0	1.0	284.79
...	...	...	...	...	...	...
NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [4]:

```
weather.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14454 entries, 0 to 14453
Data columns (total 14 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   date_time            199 non-null    object 
 1   is_holiday           199 non-null    object 
 2   air_pollution_index  199 non-null    float64
 3   humidity             199 non-null    float64
 4   wind_speed           199 non-null    float64
 5   wind_direction       199 non-null    float64
 6   visibility_in_miles  199 non-null    float64
 7   dew_point            199 non-null    float64
 8   temperature          199 non-null    float64
 9   rain_p_h             199 non-null    float64
 10  snow_p_h             199 non-null    float64
 11  clouds_all           199 non-null    float64
 12  weather_type         199 non-null    object 
 13  weather_description  199 non-null    object 
dtypes: float64(10), object(4)
memory usage: 1.5+ MB
```

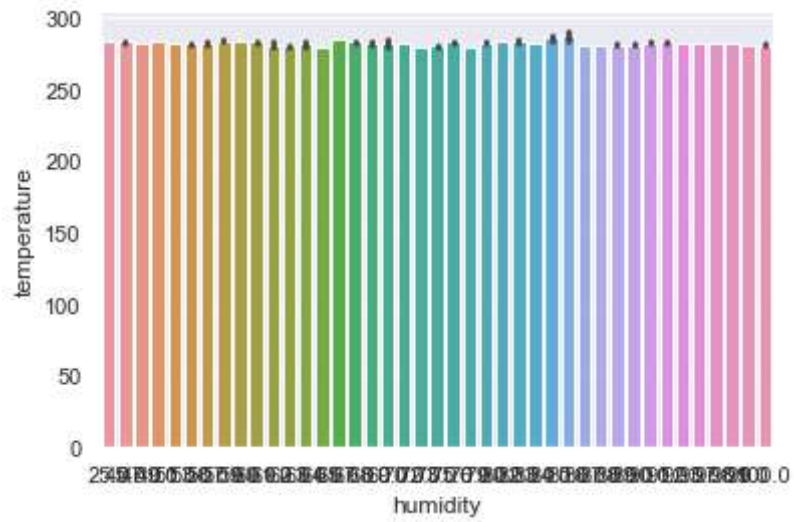
In [5]:

```
sns.barplot(weather['humidity'],weather['temperature'])
```

```
C:\Users\decos\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variables as keyword args: x, y. From version 0.
12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretat
ion.
  warnings.warn(
```

Out[5]:

```
<AxesSubplot:xlabel='humidity', ylabel='temperature'>
```



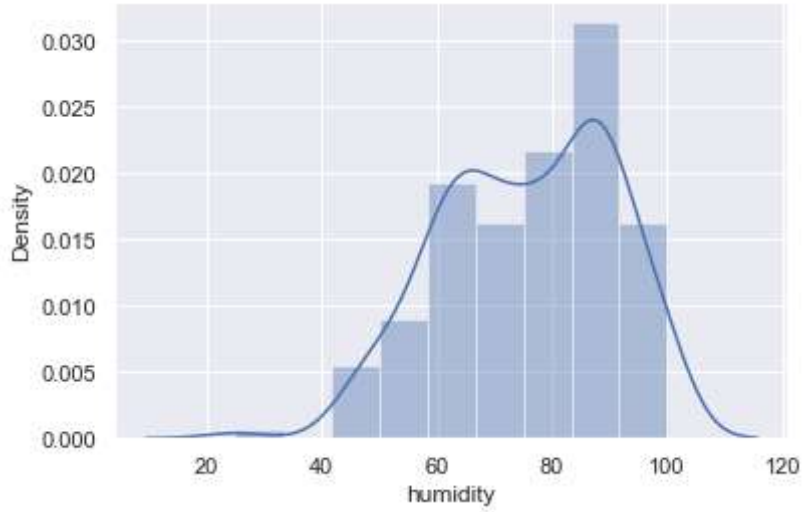
In [6]:

```
sns.distplot(weather['humidity'])
```

```
C:\Users\decos\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[6]:

```
<AxesSubplot:xlabel='humidity', ylabel='Density'>
```



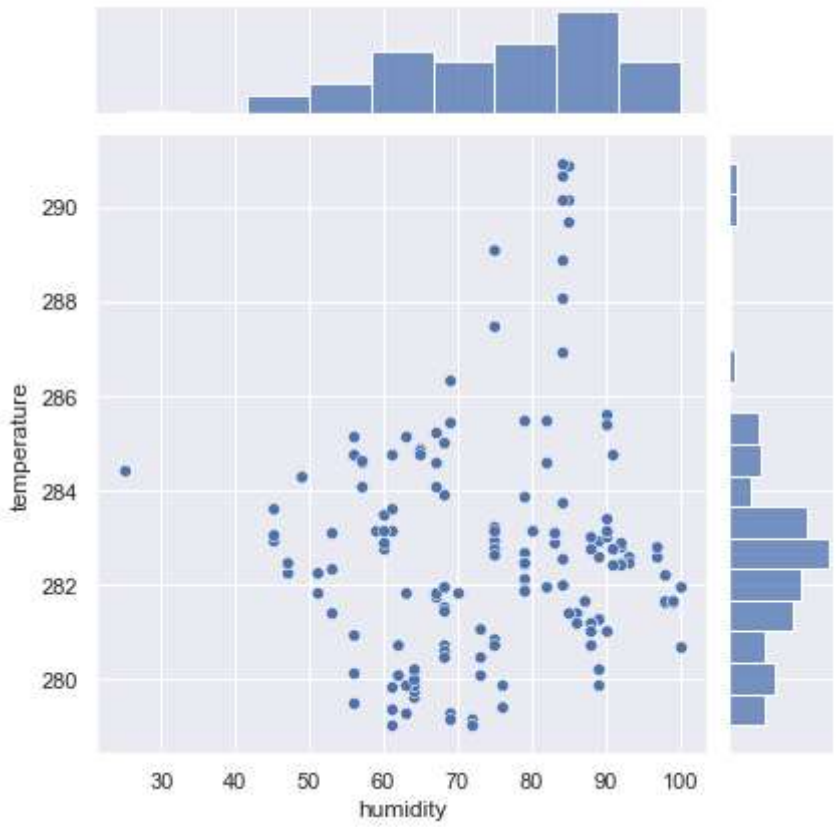
In [7]:

```
sns.jointplot(weather['humidity'],weather['temperature'])
```

```
C:\Users\decos\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variables as keyword args: x, y. From version 0.
12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretat
ion.
  warnings.warn(
```

Out[7]:

```
<seaborn.axisgrid.JointGrid at 0x201008c4d30>
```



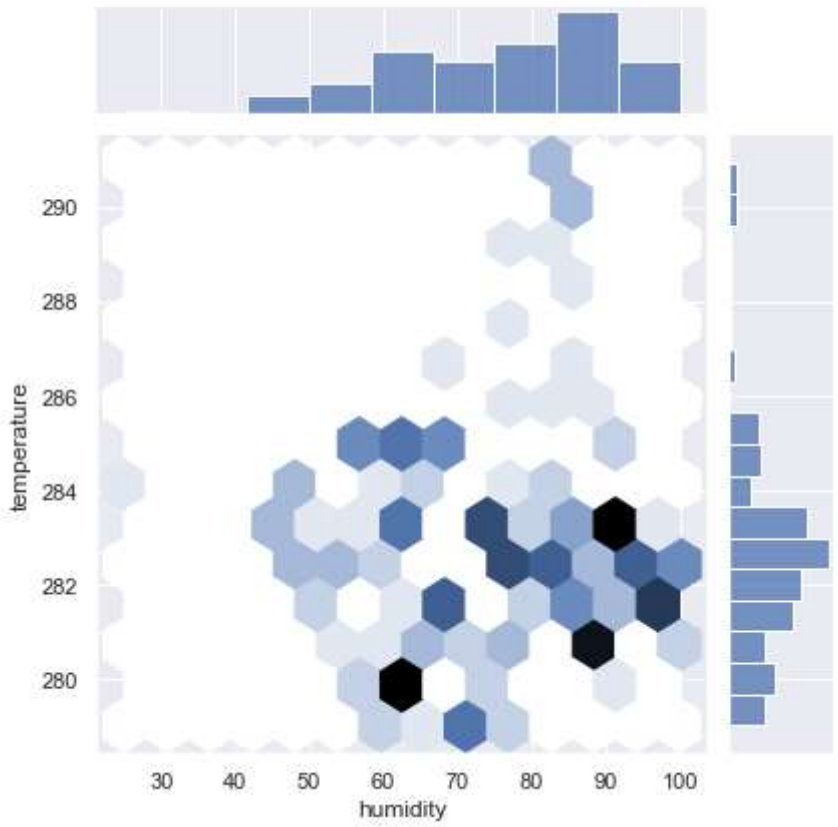
In [9]:

```
sns.jointplot(weather['humidity'],weather['temperature'],kind="hex")
```

```
C:\Users\decos\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variables as keyword args: x, y. From version 0.
12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretat
ion.
  warnings.warn(
```

Out[9]:

```
<seaborn.axisgrid.JointGrid at 0x20100b88be0>
```



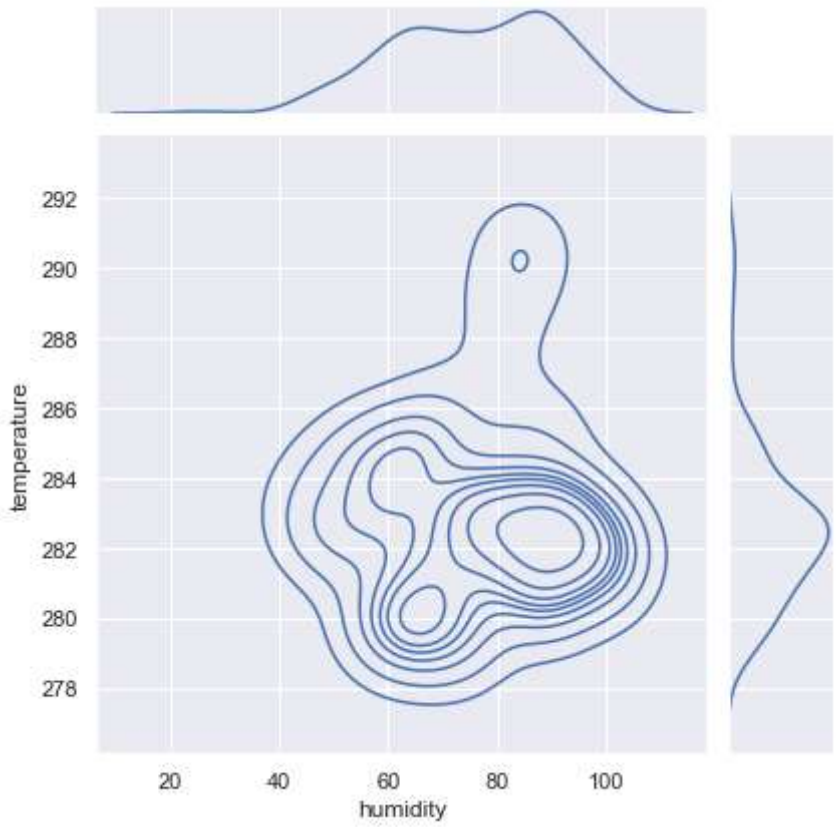
In [10]:

```
sns.jointplot(weather['humidity'],weather['temperature'],kind="kde")
```

```
C:\Users\decos\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variables as keyword args: x, y. From version 0.
12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretat
ion.
  warnings.warn(
```

Out[10]:

```
<seaborn.axisgrid.JointGrid at 0x20100cb20a0>
```

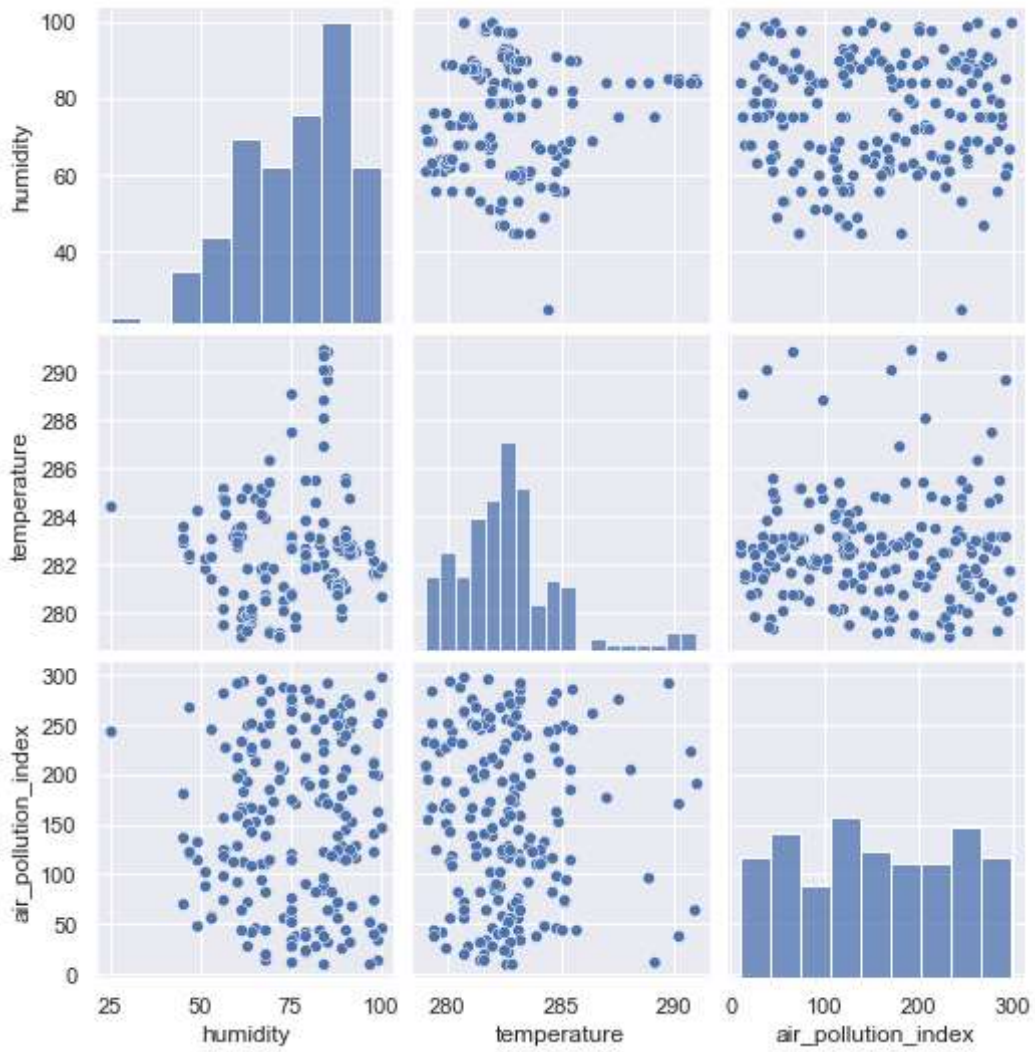


In [11]:

```
sns.pairplot(weather[['humidity', 'temperature', 'air_pollution_index']])
```

Out[11]:

```
<seaborn.axisgrid.PairGrid at 0x20101dc1850>
```



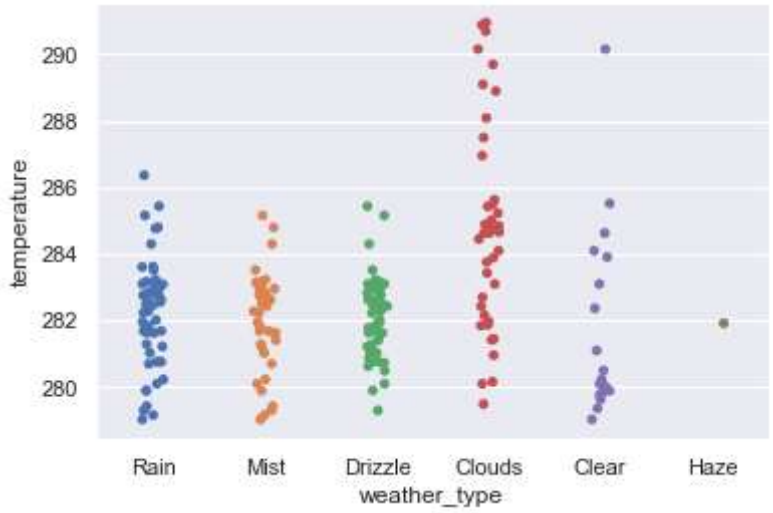
In [12]:

```
sns.stripplot(weather['weather_type'],weather['temperature'])
```

```
C:\Users\decos\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variables as keyword args: x, y. From version 0.
12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretat
ion.
  warnings.warn(
```

Out[12]:

```
<AxesSubplot:xlabel='weather_type', ylabel='temperature'>
```

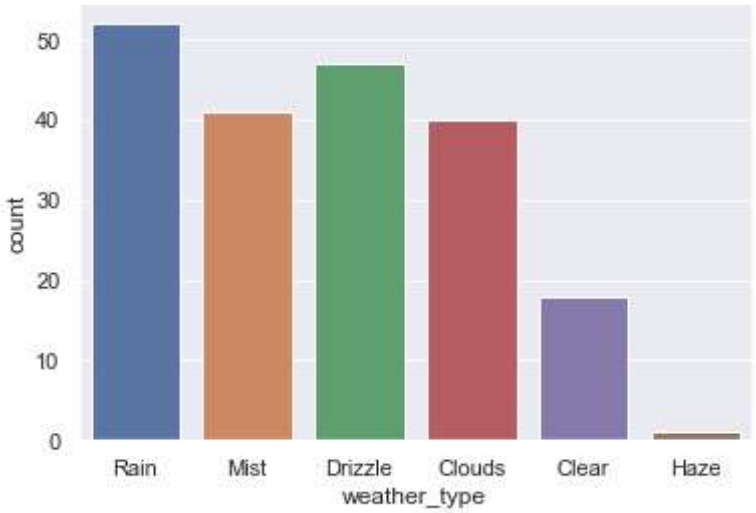


In [14]:

```
sns.countplot(x="weather_type", data=weather)
```

Out[14]:

```
<AxesSubplot:xlabel='weather_type', ylabel='count'>
```

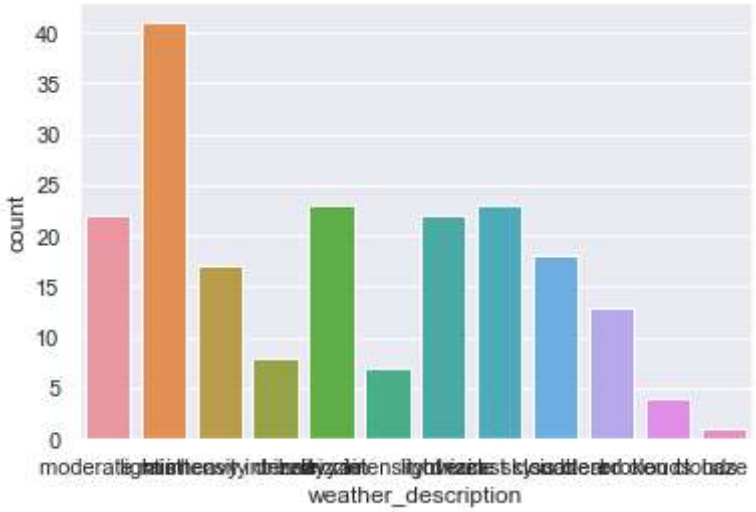


In [15]:

```
sns.countplot(x="weather_description", data=weather)
```

Out[15]:

```
<AxesSubplot:xlabel='weather_description', ylabel='count'>
```



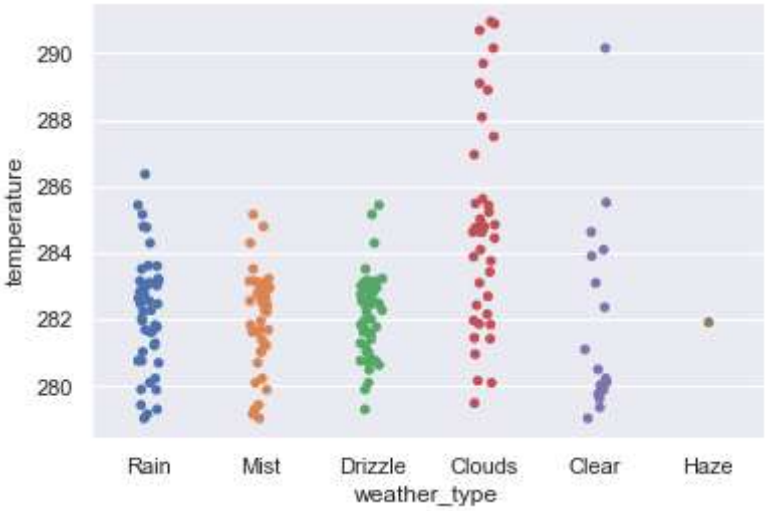
In [16]:

```
sns.stripplot(weather['weather_type'],weather['temperature'],jitter=True)
```

```
C:\Users\decos\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variables as keyword args: x, y. From version 0.
12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretat
ion.
  warnings.warn(
```

Out[16]:

```
<AxesSubplot:xlabel='weather_type', ylabel='temperature'>
```



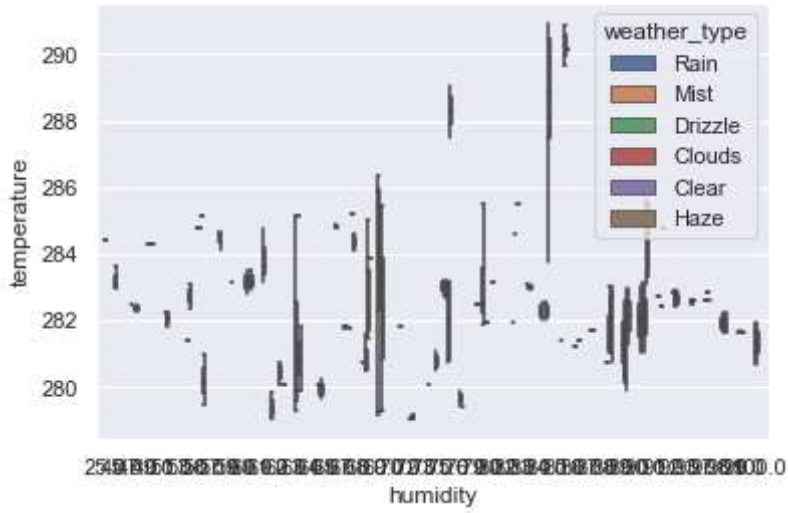
In [17]:

```
sns.boxplot(weather['humidity'],weather['temperature'],hue=weather['weather_type'])
```

```
C:\Users\decos\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variables as keyword args: x, y. From version 0.
12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretat
ion.
  warnings.warn(
```

Out[17]:

```
<AxesSubplot:xlabel='humidity', ylabel='temperature'>
```



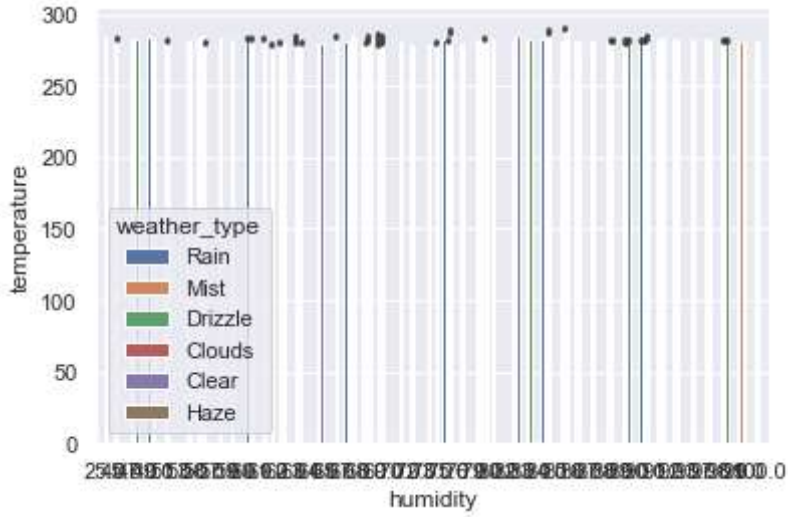
In [18]:

```
sns.barplot(weather['humidity'],weather['temperature'],hue=weather['weather_type'])
```

```
C:\Users\decos\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variables as keyword args: x, y. From version 0.
12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretat
ion.
  warnings.warn(
```

Out[18]:

```
<AxesSubplot:xlabel='humidity', ylabel='temperature'>
```



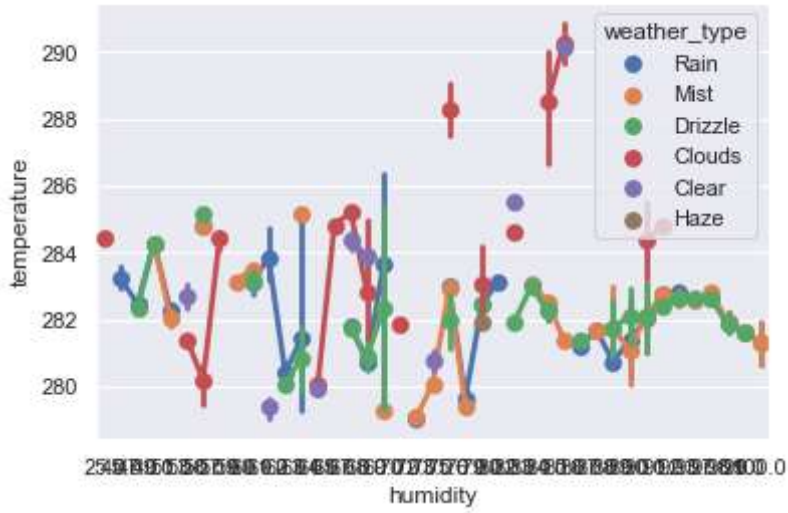
In [19]:

```
sns.pointplot(weather['humidity'],weather['temperature'],hue=weather['weather_type'])
```

```
C:\Users\decos\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variables as keyword args: x, y. From version 0.
12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretat
ion.
  warnings.warn(
```

Out[19]:

```
<AxesSubplot:xlabel='humidity', ylabel='temperature'>
```

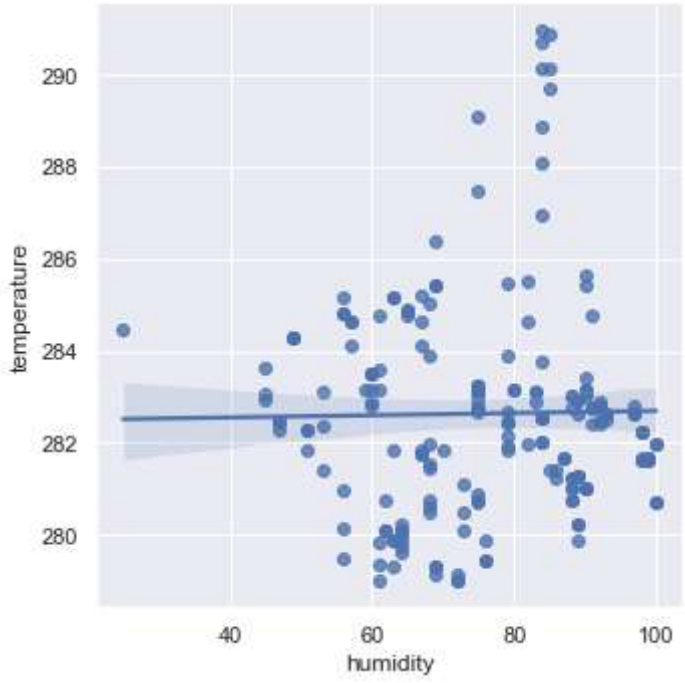


In [20]:

```
sns.lmplot(x="humidity",y="temperature",data=weather)
```

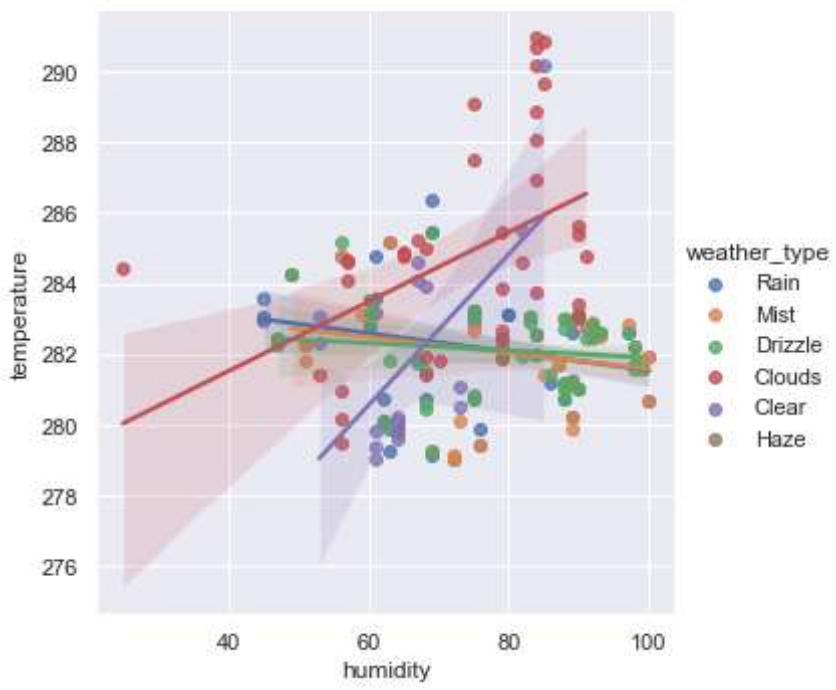
Out[20]:

```
<seaborn.axisgrid.FacetGrid at 0x20102f5bee0>
```



In [31]:

```
sns.lmplot(x="humidity", y="temperature", hue="weather_type", data=weather)
plt.savefig("figure.png")
```



In [32]:

```
weather['weather_type'].value_counts()
```

Out[32]:

```
Rain       52
Drizzle    47
Mist       41
Clouds     40
Clear      18
Haze        1
Name: weather_type, dtype: int64
```

In [34]:

```
weather['weather_type'].value_counts(ascending=True)
```

Out[34]:

```
Haze        1
Clear      18
Clouds     40
Mist       41
Drizzle    47
Rain       52
Name: weather_type, dtype: int64
```

In [33]:

```
weather['weather_description'].value_counts()
```

Out[33]:

```
mist                       41
drizzle                    23
overcast clouds            23
moderate rain              22
light rain                 22
sky is clear               18
light intensity drizzle    17
scattered clouds           13
heavy intensity rain        8
heavy intensity drizzle     7
broken clouds               4
haze                        1
Name: weather_description, dtype: int64
```

In [36]:

```
weather['weather_description'].value_counts(ascending=True)
```

Out[36]:

```
haze                        1
broken clouds               4
heavy intensity drizzle     7
heavy intensity rain        8
scattered clouds           13
light intensity drizzle    17
sky is clear               18
moderate rain              22
light rain                 22
drizzle                    23
overcast clouds            23
mist                       41
Name: weather_description, dtype: int64
```

In [38]:

```
weather['air_pollution_index'].describe()
```

Out[38]:

```
count    199.000000
mean     154.130653
std       83.713997
min       10.000000
25%       83.000000
50%      154.000000
75%      231.500000
max      298.000000
Name: air_pollution_index, dtype: float64
```

In [39]:

```
weather['humidity'].describe()
```

Out[39]:

```
count    199.000000
mean      75.778894
std       14.921050
min       25.000000
25%       64.000000
50%       75.000000
75%       88.500000
max      100.000000
Name: humidity, dtype: float64
```

In [40]:

```
weather['temperature'].describe()
```

Out[40]:

```
count    199.000000
mean     282.633216
std        2.311407
min      279.020000
25%      281.210000
50%      282.460000
75%      283.220000
max      290.950000
Name: temperature, dtype: float64
```

In [ ]:

```

```