WIND SPEED PREDICTION

Context:- High precision and reliable wind speed forecasting is a challenge for meteorologists. Severe wind due to convective storms, causes considerable damages (large scale forest damage, outage, buildings/houses damage, etc.). Convective events such as thunderstorms, tornadoes as well as large hail, strong winds, are natural hazards that have the potential to disrupt daily life, especially over complex terrain favoring the initiation of convection. Even ordinary convective events produce severe winds which causes fatal and costly damages. Therefore, wind speed prediction is an important task to get advanced severe weather warning. This dataset contains the responses of a weather sensor that collected different weather variables such as temperatures and precipitation.

Content:- The dataset contains 6574 instances of daily averaged responses from an array of 5 weather variables sensors embedded in a meteorological station. The device was located on the field in a significantly empty area, at 21M. Data were recorded from January 1961 to December 1978 (17 years). Ground Truth daily averaged precipitations, maximum and minimum temperatures, and grass minimum temperature were provided.

Attribute Information:-

DATE:(YYYY-MM-DD)

WIND: Average wind speed [knots]

IND: First indicator value

RAIN: Precipitation Amount (mm)

IND.1: Second indicator value

T.MAX: Maximum Temperature (°C)

IND.2: Third indicator value

T.MIN: Minimum Temperature (°C)

T.MIN.G: 09utc Grass Minimum Temperature (°C)

Import the libraries

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

Load the dataset

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

	DATE	WIND	IND	RAIN	IND.1	T.MAX	IND.2	T.MIN	T.MIN.G
0	1961-01-01	13.67	0	0.2	0.0	9.5	0.0	3.7	-1.0
1	1961-01-02	11.50	0	5.1	0.0	7.2	0.0	4.2	1.1
2	1961-01-03	11.25	0	0.4	0.0	5.5	0.0	0.5	-0.5
3	1961-01-04	8.63	0	0.2	0.0	5.6	0.0	0.4	-3.2
4	1961-01-05	11.92	0	10.4	0.0	7.2	1.0	-1.5	-7.5
...	...	...	...	...	...	...	...	...	...
6569	1978-12-27	14.46	0	16.8	0.0	9.8	0.0	4.0	0.0
6570	1978-12-28	14.33	0	16.0	0.0	9.1	0.0	8.5	8.0
6571	1978-12-29	19.17	0	14.7	0.0	5.0	0.0	3.5	3.2
6572	1978-12-30	18.08	0	4.9	0.0	2.9	0.0	0.3	-0.5
6573	1978-12-31	19.25	0	0.5	0.0	1.2	1.0	-1.5	-3.0

6574 rows × 9 columns

Data Exploration

```
df.head()
```

	DATE	WIND	IND	RAIN	IND.1	T.MAX	IND.2	T.MIN	T.MIN.G
0	1961-01-01	13.67	0	0.2	0.0	9.5	0.0	3.7	-1.0
1	1961-01-02	11.50	0	5.1	0.0	7.2	0.0	4.2	1.1
2	1961-01-03	11.25	0	0.4	0.0	5.5	0.0	0.5	-0.5
3	1961-01-04	8.63	0	0.2	0.0	5.6	0.0	0.4	-3.2
4	1961-01-05	11.92	0	10.4	0.0	7.2	1.0	-1.5	-7.5

```
df.tail()
```

	DATE	WIND	IND	RAIN	IND.1	T.MAX	IND.2	T.MIN	T.MIN.G
6569	1978-12-27	14.46	0	16.8	0.0	9.8	0.0	4.0	0.0
6570	1978-12-28	14.33	0	16.0	0.0	9.1	0.0	8.5	8.0
6571	1978-12-29	19.17	0	14.7	0.0	5.0	0.0	3.5	3.2
6572	1978-12-30	18.08	0	4.9	0.0	2.9	0.0	0.3	-0.5
6573	1978-12-31	19.25	0	0.5	0.0	1.2	1.0	-1.5	-3.0

```
df.describe()
```

	WIND	IND	RAIN	IND.1	T.MAX	IND.2	
count	6574.000000	6574.000000	6574.000000	6513.000000	5953.000000	6513.000000	59
mean	9.796834	0.391542	1.885169	0.356364	13.339123	0.464456	
std	4.977272	1.179092	4.030529	1.128552	4.890546	1.177571	
min	0.000000	0.000000	0.000000	0.000000	-0.100000	0.000000	-
25%	6.000000	0.000000	0.000000	0.000000	9.600000	0.000000	
50%	9.210000	0.000000	0.200000	0.000000	13.300000	0.000000	
75%	12.960000	0.000000	2.000000	0.000000	17.200000	0.000000	
max	30.370000	4.000000	67.000000	4.000000	26.800000	4.000000	

```
df.dtypes
```

```
DATE        object
WIND       float64
IND          int64
```

```
RAIN       float64
IND.1      float64
T.MAX      float64
IND.2      float64
T.MIN      float64
T.MIN.G    float64
dtype: object
```

```
df.shape
```

```
(6574, 9)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6574 entries, 0 to 6573
Data columns (total 9 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   DATE     6574 non-null   object 
 1   WIND     6574 non-null   float64
 2   IND      6574 non-null   int64  
 3   RAIN     6574 non-null   float64
 4   IND.1    6513 non-null   float64
 5   T.MAX    5953 non-null   float64
 6   IND.2    6513 non-null   float64
 7   T.MIN    5900 non-null   float64
 8   T.MIN.G  6214 non-null   float64
dtypes: float64(7), int64(1), object(1)
memory usage: 462.4+ KB
```

```
df.dropna()
```

	DATE	WIND	IND	RAIN	IND.1	T.MAX	IND.2	T.MIN	T.MIN.G
0	1961-01-01	13.67	0	0.2	0.0	9.5	0.0	3.7	-1.0
1	1961-01-02	11.50	0	5.1	0.0	7.2	0.0	4.2	1.1
2	1961-01-03	11.25	0	0.4	0.0	5.5	0.0	0.5	-0.5
3	1961-01-04	8.63	0	0.2	0.0	5.6	0.0	0.4	-3.2
4	1961-01-05	11.92	0	10.4	0.0	7.2	1.0	-1.5	-7.5
...	...	...	...	...	...	...	...	...	...
6569	1978-12-27	14.46	0	16.8	0.0	9.8	0.0	4.0	0.0
6570	1978-12-28	14.33	0	16.0	0.0	9.1	0.0	8.5	8.0
6571	1978-12-29	19.17	0	14.7	0.0	5.0	0.0	3.5	3.2
6572	1978-12-30	18.08	0	4.9	0.0	2.9	0.0	0.3	-0.5
6573	1978-12-31	19.25	0	0.5	0.0	1.2	1.0	-1.5	-3.0

5638 rows × 9 columns

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

```
DATE         0
WIND         0
IND          0
RAIN         0
IND.1       61
T.MAX      621
IND.2       61
T.MIN      674
T.MIN.G    360
dtype: int64
```

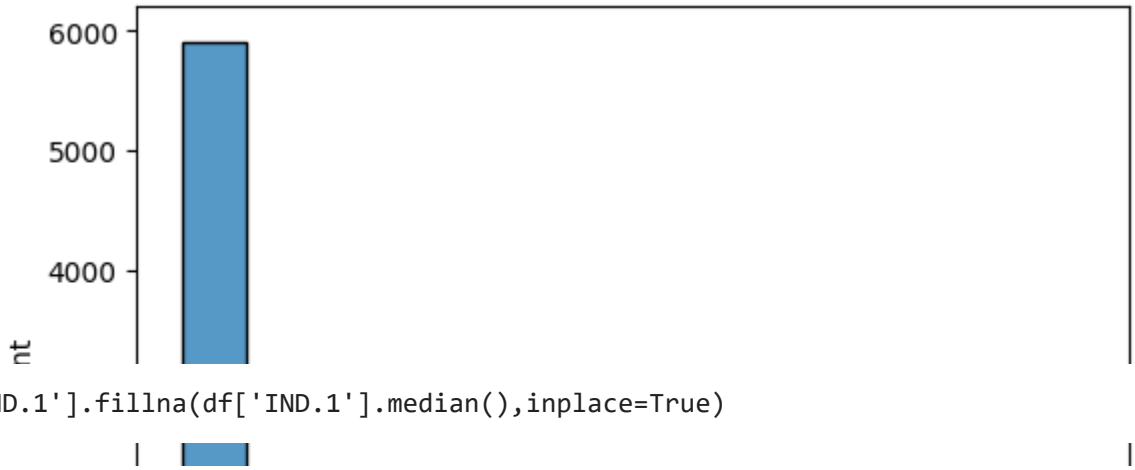
```
df.drop(['DATE'],axis=1,inplace=True)
df
```

	WIND	IND	RAIN	IND.1	T.MAX	IND.2	T.MIN	T.MIN.G
0	13.67	0	0.2	0.0	9.5	0.0	3.7	-1.0
1	11.50	0	5.1	0.0	7.2	0.0	4.2	1.1
2	11.25	0	0.4	0.0	5.5	0.0	0.5	-0.5
3	8.63	0	0.2	0.0	5.6	0.0	0.4	-3.2
4	11.92	0	10.4	0.0	7.2	1.0	-1.5	-7.5
...	...	...	...	...	...	...	...	...
6569	14.46	0	16.8	0.0	9.8	0.0	4.0	0.0
6570	14.33	0	16.0	0.0	9.1	0.0	8.5	8.0
6571	19.17	0	14.7	0.0	5.0	0.0	3.5	3.2
6572	18.08	0	4.9	0.0	2.9	0.0	0.3	-0.5
6573	19.25	0	0.5	0.0	1.2	1.0	-1.5	-3.0

6574 rows × 8 columns

```
sns.histplot(df['IND.1'])
```

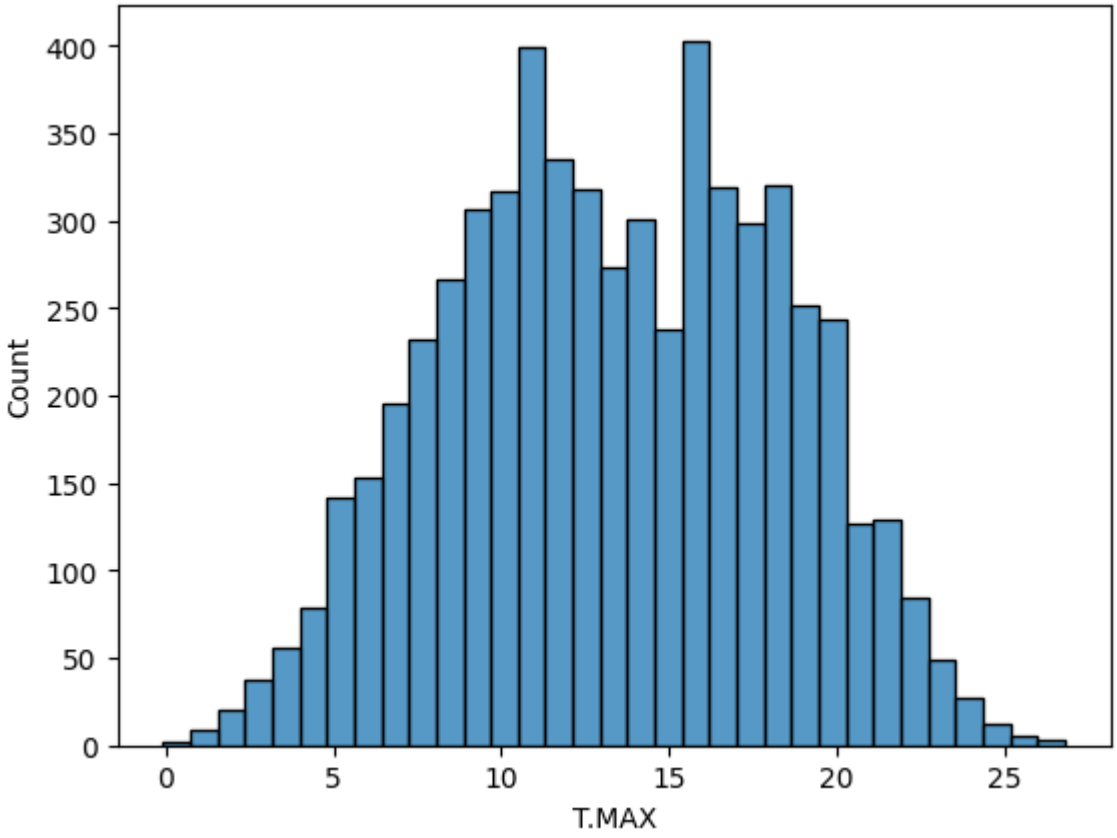
```
<Axes: xlabel='IND.1', ylabel='Count'>
```




```
df['IND.1'].fillna(df['IND.1'].median(),inplace=True)
```

```
sns.histplot(df['T.MAX'])
```

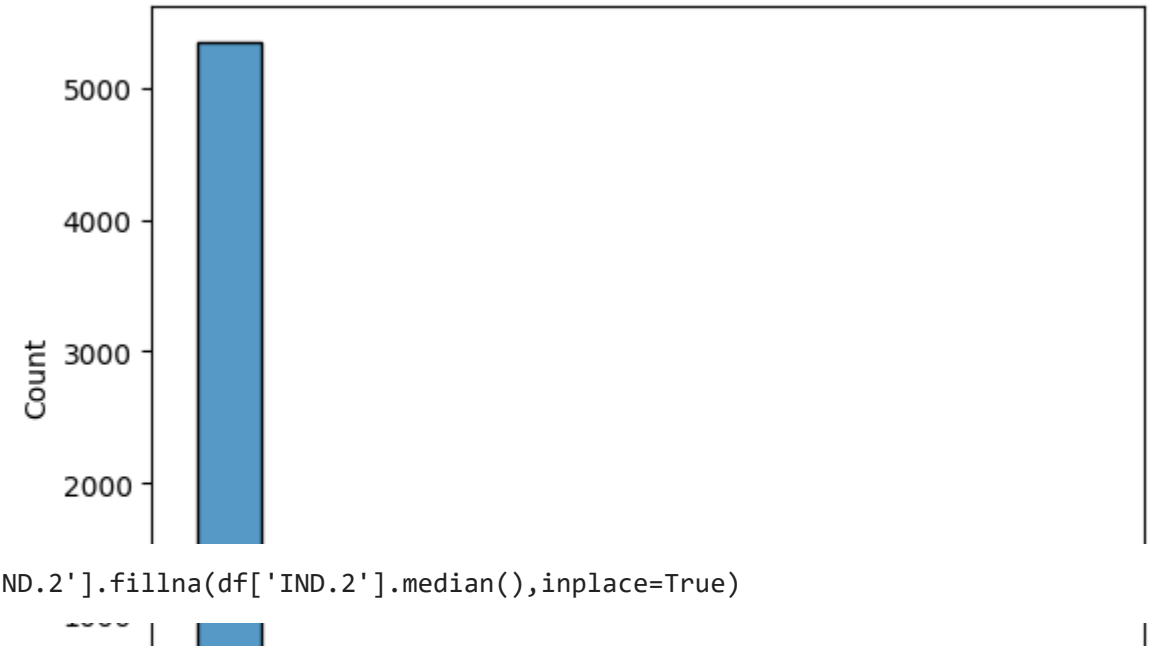
```
<Axes: xlabel='T.MAX', ylabel='Count'>
```




```
df['T.MAX'].fillna(df['T.MAX'].mean(),inplace=True)
```

```
sns.histplot(df['IND.2'])
```

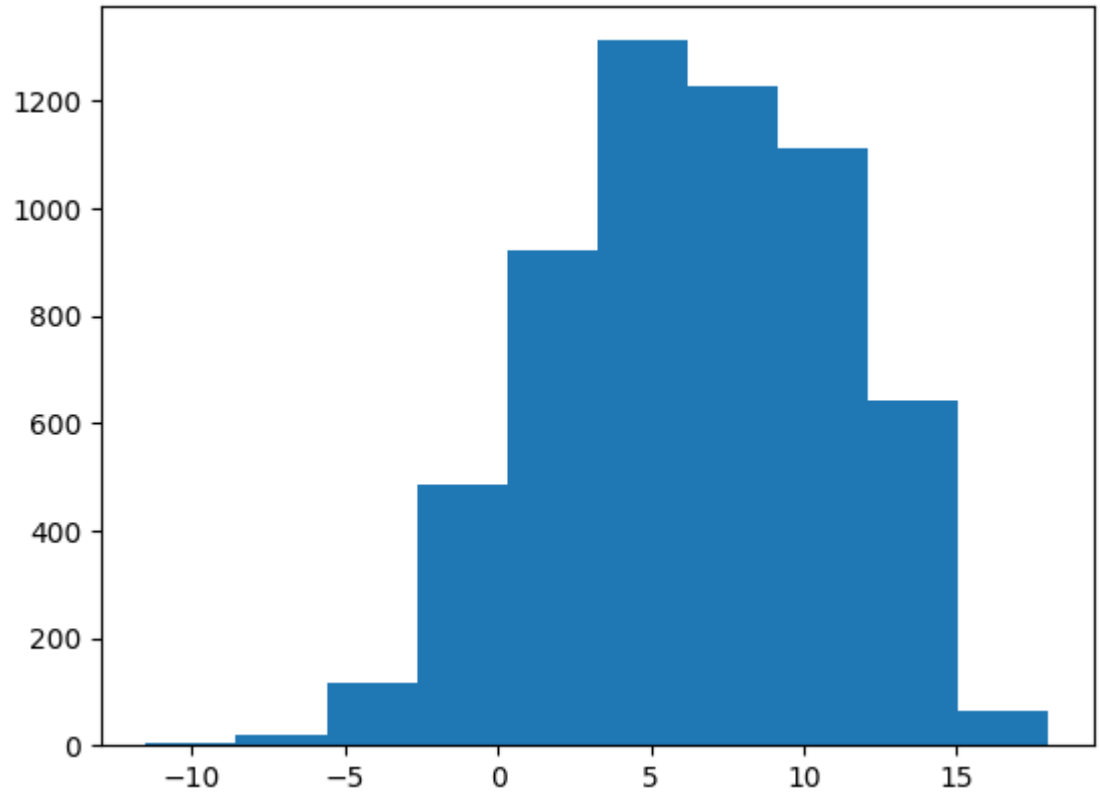
```
<Axes: xlabel='IND.2', ylabel='Count'>
```




```
df['IND.2'].fillna(df['IND.2'].median(),inplace=True)
```

```
plt.hist(df['T.MIN'])
```

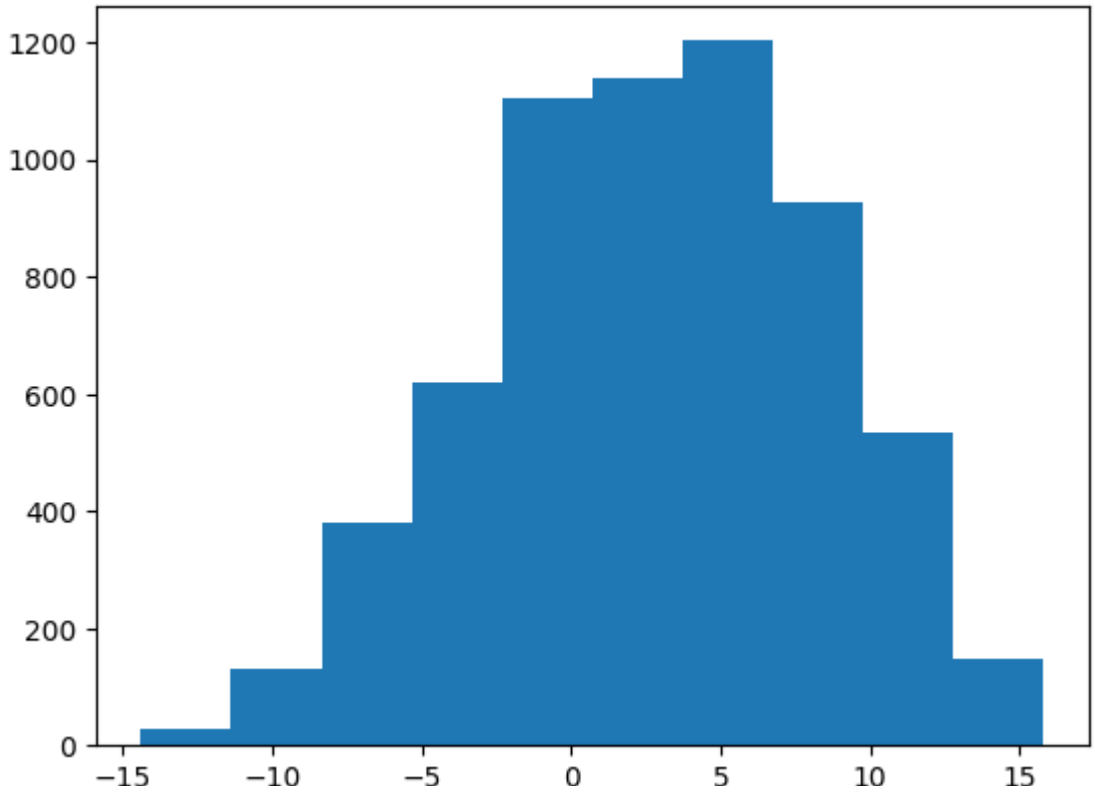
```
(array([   4.,   21.,  115.,  485.,  920., 1312., 1228., 1111.,  641.,
          63.]),
 array([-11.5 ,  -8.55,  -5.6 ,  -2.65,   0.3 ,   3.25,   6.2 ,   9.15,
         12.1 ,  15.05,  18.  ]),
 <BarContainer object of 10 artists>)
```




```
df['T.MIN'].fillna(df['T.MIN'].median(),inplace=True)
```

```
plt.hist(df['T.MIN.G'])
```

```
(array([  27.,  131.,  380.,  620., 1105., 1139., 1203.,  927.,  535.,
        147.]),
 array([-14.4 , -11.38,  -8.36,  -5.34,  -2.32,   0.7 ,   3.72,   6.74,
         9.76,  12.78,  15.8 ]),
 <BarContainer object of 10 artists>)
```



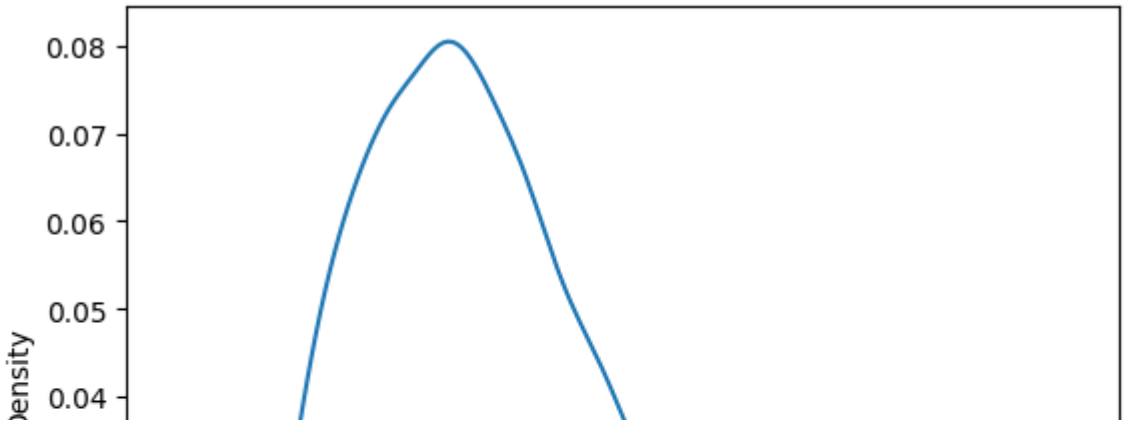

```
df['T.MIN.G'].fillna(df['T.MIN.G'].mean(),inplace=True)
```

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

```
WIND       0
IND        0
RAIN       0
IND.1      0
T.MAX      0
IND.2      0
T.MIN      0
T.MIN.G    0
dtype: int64
```

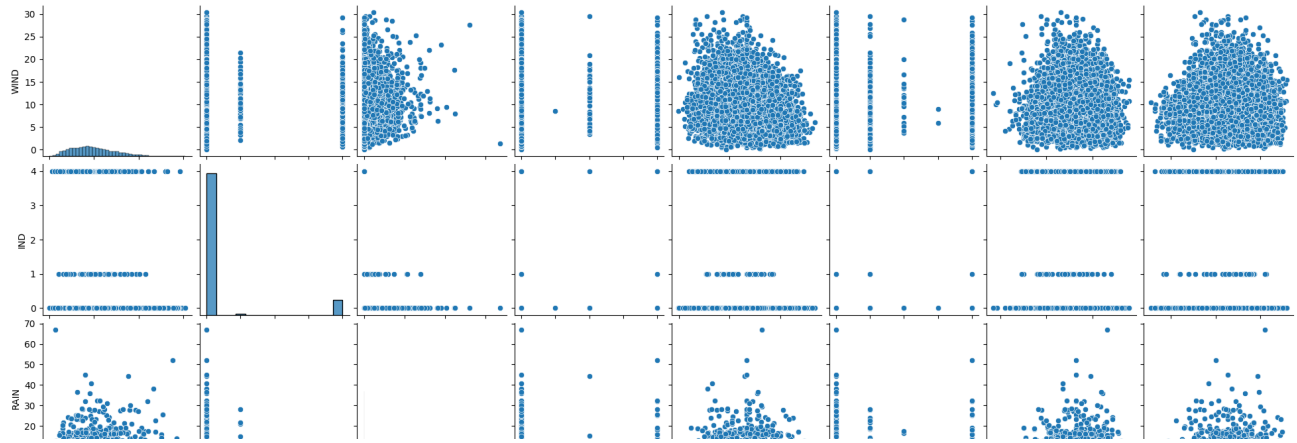
```
sns.kdeplot(df['WIND'])
```

```
<Axes: xlabel='WIND', ylabel='Density'>
```



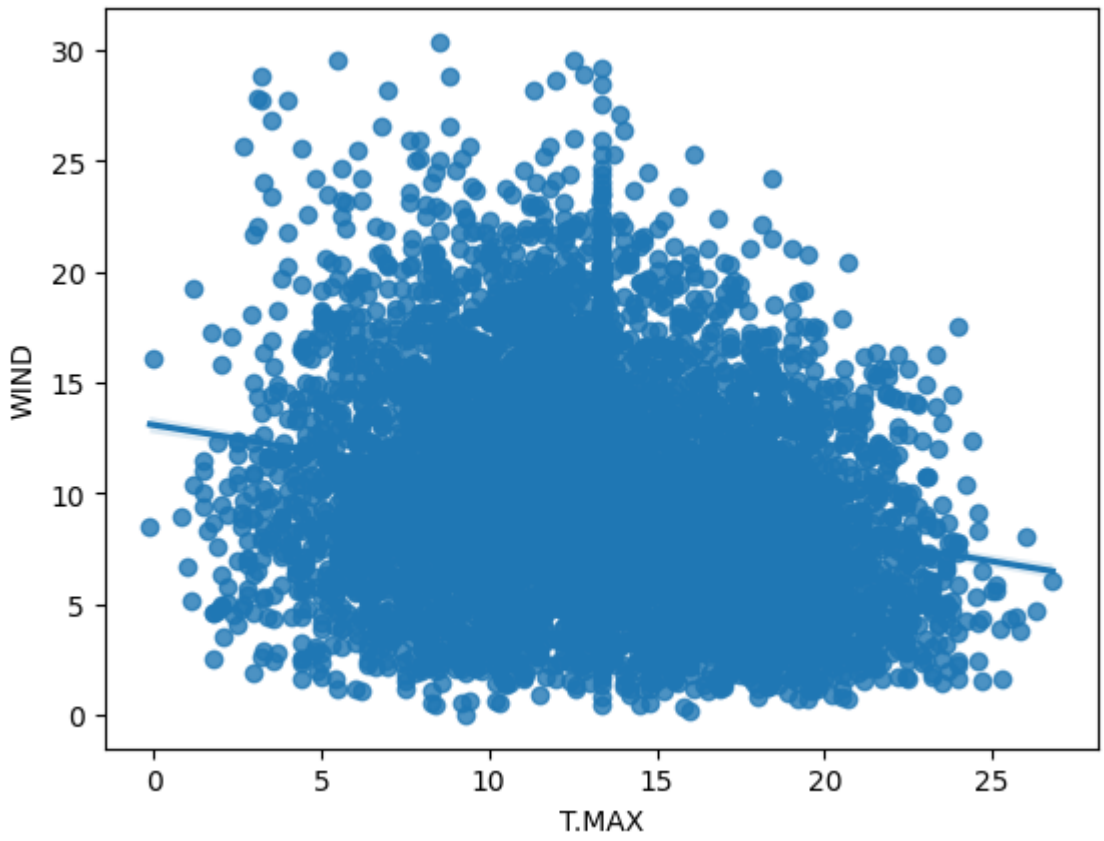

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7c9250658d90>
```




```
sns.regplot(x='T.MAX',y='WIND',data=df)
```

```
<Axes: xlabel='T.MAX', ylabel='WIND'>
```

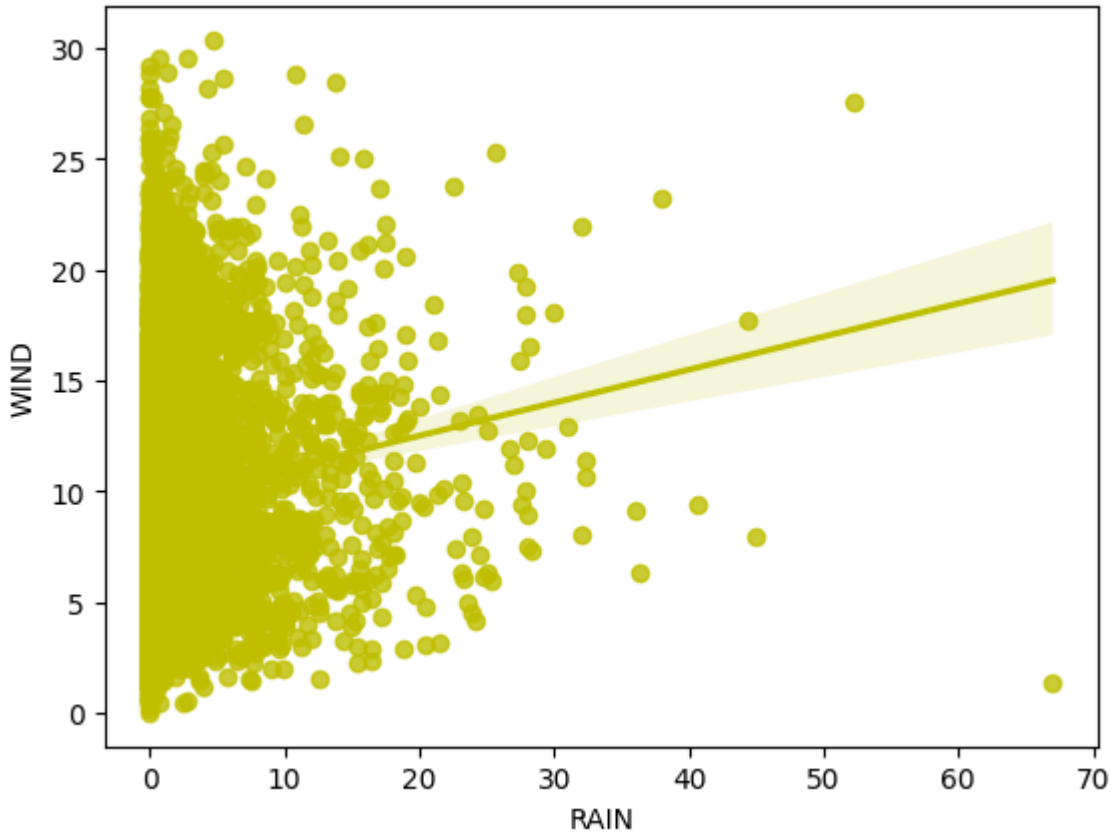



```
sns.regplot(x='IND',y='WIND',data=df,color='r')
```

```
<Axes: xlabel='IND', ylabel='WIND'>
```

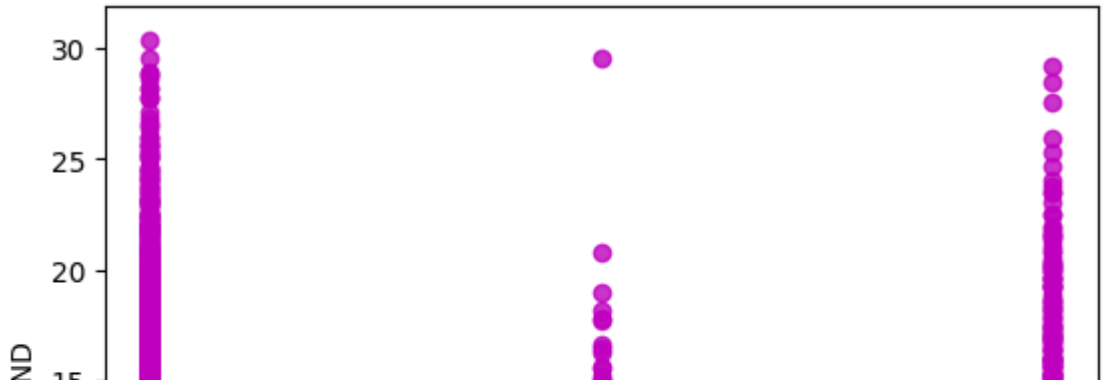

```
sns.regplot(x='RAIN',y='WIND',data=df,color='y')
```

```
<Axes: xlabel='RAIN', ylabel='WIND'>
```



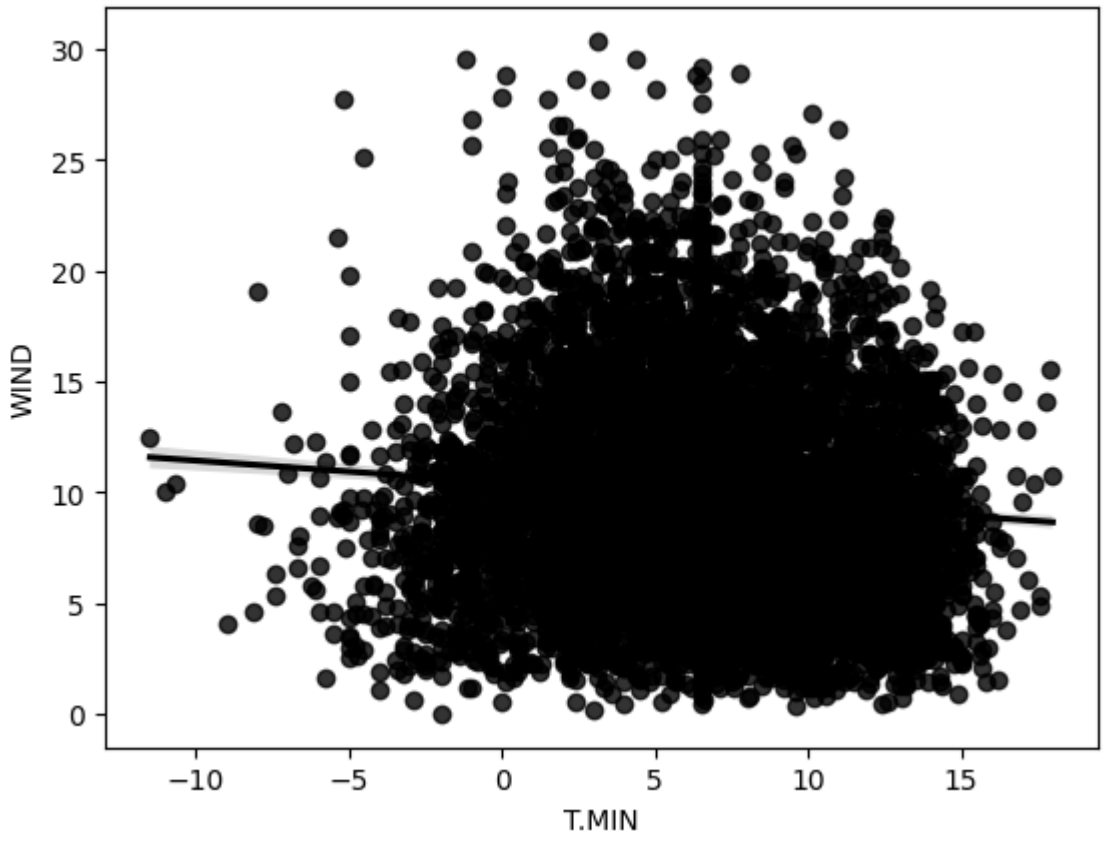

```
sns.regplot(x='IND.1',y='WIND',data=df,color='m')
```

```
<Axes: xlabel='IND.1', ylabel='WIND'>
```




```
sns.regplot(x='T.MIN',y='WIND',data=df,color='k')
```

```
<Axes: xlabel='T.MIN', ylabel='WIND'>
```

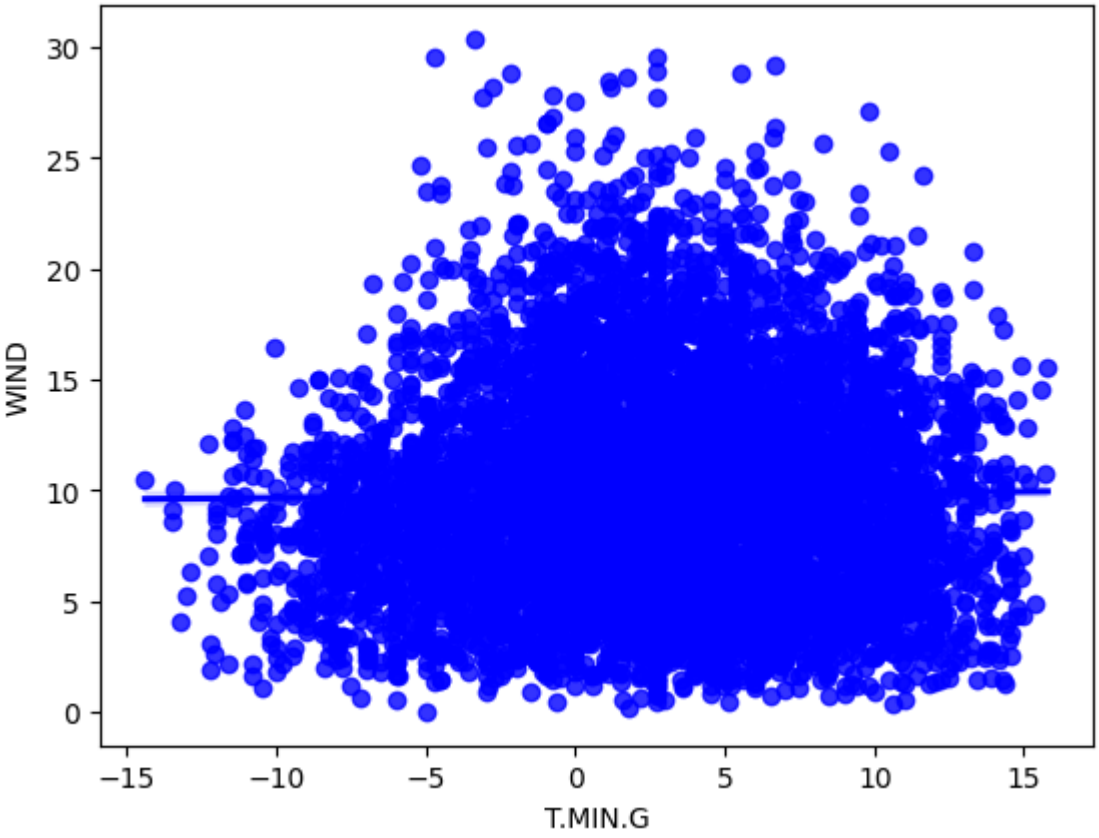



```
sns.regplot(x='IND.2',y='WIND',data=df,color='g')
```

```
<Axes: xlabel='IND.2', ylabel='WIND'>
```


```
sns.regplot(x='T.MIN.G',y='WIND',data=df,color='b')
```

```
<Axes: xlabel='T.MIN.G', ylabel='WIND'>
```




```
sns.distplot(df['T.MAX'])
```

```
<ipython-input-39-f520710037a9>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['T.MAX'])
<Axes: xlabel='T.MAX', ylabel='Density'>
```


```
sns.heatmap(df.corr(),annot=True)
```

```
<Axes: >
```