In [5]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("weather.csv")
```

In [2]:
```
#Total number of rows and columns
df.shape
```

Out[2]: (366, 22)

In [3]:
```
#Top 5 rows
df.head()
```

Out[3]:

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	...	Humidity3pm
0	8.0	24.3	0.0	3.4	6.3	NW	30.0	SW	NW	6.0	...	29
1	14.0	26.9	3.6	4.4	9.7	ENE	39.0	E	W	4.0	...	36
2	13.7	23.4	3.6	5.8	3.3	NW	85.0	N	NNE	6.0	...	69
3	13.3	15.5	39.8	7.2	9.1	NW	54.0	WNW	W	30.0	...	56
4	7.6	16.1	2.8	5.6	10.6	SSE	50.0	SSE	ESE	20.0	...	49

5 rows × 22 columns

In [4]:
```
#Bottom 5 rows
df.tail()
```

Out[4]:

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	...	Humidity3p
361	9.0	30.7	0.0	7.6	12.1	NNW	76.0	SSE	NW	7.0	...	
362	7.1	28.4	0.0	11.6	12.7	N	48.0	NNW	NNW	2.0	...	
363	12.5	19.9	0.0	8.4	5.3	ESE	43.0	ENE	ENE	11.0	...	
364	12.5	26.9	0.0	5.0	7.1	NW	46.0	SSW	WNW	6.0	...	
365	12.3	30.2	0.0	6.0	12.6	NW	78.0	NW	WNW	31.0	...	

5 rows × 22 columns

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 366 entries, 0 to 365
Data columns (total 22 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   MinTemp        366 non-null    float64
 1   MaxTemp        366 non-null    float64
 2   Rainfall       366 non-null    float64
 3   Evaporation    366 non-null    float64
 4   Sunshine       363 non-null    float64
 5   WindGustDir    363 non-null    object 
 6   WindGustSpeed  364 non-null    float64
 7   WindDir9am     335 non-null    object 
 8   WindDir3pm     365 non-null    object 
 9   WindSpeed9am   359 non-null    float64
 10  WindSpeed3pm   366 non-null    int64  
 11  Humidity9am    366 non-null    int64  
 12  Humidity3pm    366 non-null    int64  
 13  Pressure9am    366 non-null    float64
 14  Pressure3pm    366 non-null    float64
 15  Cloud9am       366 non-null    int64  
 16  Cloud3pm       366 non-null    int64  
 17  Temp9am        366 non-null    float64
 18  Temp3pm        366 non-null    float64
 19  RainToday      366 non-null    object 
 20  RISK_MM        366 non-null    float64
 21  RainTomorrow   366 non-null    object 
dtypes: float64(12), int64(5), object(5)
memory usage: 63.0+ KB
```

In [151]:
```
df = df.drop_duplicates()
```

In [55]:
```
#All column names
df.columns
```

Out[55]:
```
Index(['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine',
       'WindGustDir', 'WindGustSpeed', 'WindDir9am', 'WindDir3pm',
       'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm',
       'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm', 'Temp9am',
       'Temp3pm', 'RainToday', 'RISK_MM', 'RainTomorrow'],
      dtype='object')
```

In [56]:
```
#For statistical analysis
df.describe()
```

Out[56]:

	MinTemp	MaxTemp	Rainfall	Evaporation	WindSpeed3pm	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm	Cloud9am	Clo
count	366.000000	366.000000	366.000000	366.000000	366.000000	366.000000	366.000000	3.660000e+02	366.000000	366.000000	366.
mean	7.265574	20.550273	1.428415	4.521858	17.986339	72.035519	44.519126	1.015350e+03	1016.810383	3.890710	4.
std	6.025800	6.690516	4.225800	2.669383	8.856997	13.137058	16.850947	3.756801e-12	6.469422	2.956131	2.
min	-5.300000	7.600000	0.000000	0.200000	0.000000	36.000000	13.000000	1.015350e+03	996.800000	0.000000	0.
25%	2.300000	15.025000	0.000000	2.200000	11.000000	64.000000	32.250000	1.015350e+03	1012.800000	1.000000	1.
50%	7.450000	19.650000	0.000000	4.200000	17.000000	72.000000	43.000000	1.015350e+03	1017.400000	3.500000	4.
75%	12.500000	25.500000	0.200000	6.400000	24.000000	81.000000	55.000000	1.015350e+03	1021.475000	7.000000	7.
max	20.900000	35.800000	39.800000	13.800000	52.000000	99.000000	96.000000	1.015350e+03	1033.200000	8.000000	8.

In [57]:
```
df["WindGustDir"].unique()
```

Out[57]:
```
array(['NW', 'ENE', 'SSE', 'SE', 'E', 'S', 'N', 'WNW', 'ESE', 'NE', 'NNE',
       'NNW', 'SW', 'W', 'WSW', 'SSW', 'N/A'], dtype=object)
```

In [85]:
```
df.nunique()
```

Out[85]:
```
MinTemp          180
MaxTemp          187
Rainfall          47
Evaporation       55
Sunshine         114
WindGustDir       16
WindGustSpeed     35
WindDir9am        16
WindDir3pm        16
WindSpeed9am      22
WindSpeed3pm      26
Humidity9am       60
Humidity3pm       74
Pressure9am      190
Pressure3pm      193
Cloud9am           9
Cloud3pm           9
Temp9am          178
Temp3pm          200
RainToday          2
RISK_MM           47
RainTomorrow       2
dtype: int64
```

In [86]:
```
#To remove null values
df.isnull().sum()
```

Out[86]:
```
MinTemp          0
MaxTemp          0
Rainfall         0
Evaporation      0
Sunshine         3
WindGustDir      3
WindGustSpeed    2
WindDir9am      31
WindDir3pm       1
WindSpeed9am     7
WindSpeed3pm     0
Humidity9am      0
Humidity3pm      0
Pressure9am      0
Pressure3pm      0
Cloud9am         0
Cloud3pm         0
Temp9am          0
Temp3pm          0
RainToday        0
RISK_MM          0
RainTomorrow     0
dtype: int64
```

In [87]:
```
df = df.fillna("N/A")
```

In [88]:
```
df.isnull().sum()
```

Out[88]:
```
MinTemp          0
MaxTemp          0
Rainfall         0
Evaporation      0
Sunshine         0
WindGustDir      0
WindGustSpeed    0
WindDir9am       0
WindDir3pm       0
WindSpeed9am     0
WindSpeed3pm     0
Humidity9am      0
Humidity3pm      0
Pressure9am      0
Pressure3pm      0
Cloud9am         0
Cloud3pm         0
Temp9am          0
Temp3pm          0
RainToday        0
RISK_MM          0
RainTomorrow     0
dtype: int64
```

In [196]:
```
#Outliers using IQR
q1 = df["Humidity9am"].quantile(0.25)
print("q1 - ",q1)
q3 = df["Humidity9am"].quantile(0.75)
print("q3 - ",q3)
```

```
q1 -  64.0
q3 -  81.0
```

In [197]:
```
IQR = q3-q1
IQR
```

Out[197]:
```
17.0
```

In [198]:
```
#Outlier Threshold
lower_bound = q1-1.5*IQR
upper_bound = q3+1.5*IQR
print("Lower bound",lower_bound)
print("Upper bound",upper_bound)
```

```
Lower bound 38.5
Upper bound 106.5
```

In [199]:
```
upper_array = np.where(df['Humidity9am']>=upper_bound)[0]
lower_array = np.where(df['Humidity9am']<=lower_bound)[0]
print(upper_array)
print(lower_array)
```

```
[]
[332 361]
```

In [200]:
```
#To remove outliers
df.drop(index=lower_array,inplace = True)
```

In [202]:
```
df.shape
```

Out[202]: (364, 22)

In [211]:
```
#correlation Analysis
correlation = df[["Rainfall","Evaporation"]].corr()
correlation
```

Out[211]:

	Rainfall	Evaporation
Rainfall	1.000000	-0.004548
Evaporation	-0.004548	1.000000

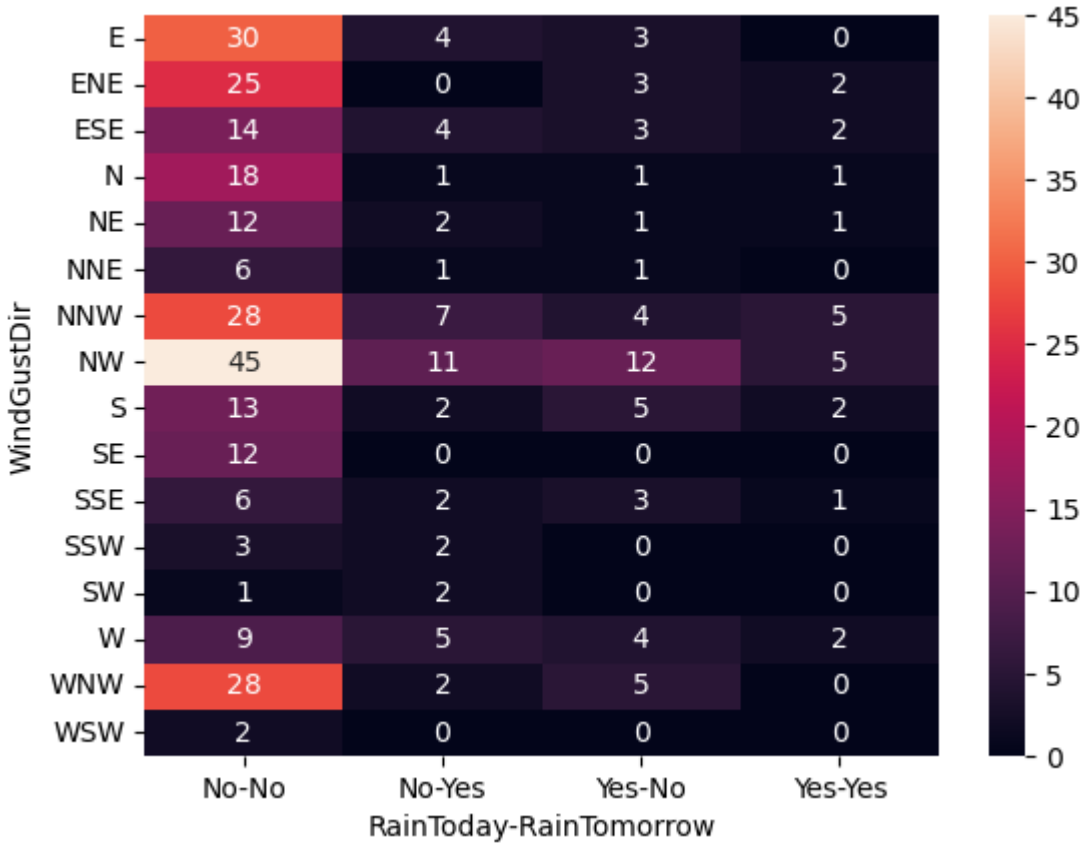
In [216]:
```
sns.heatmap(correlation,annot=True)
plt.show()
```

In [14]:
```
#Regression Analysis
regression_analysis = pd.crosstab(index=df["WindGustDir"],columns=[df["RainToday"],df["RainTomorrow"]])
regression_analysis
```

Out[14]:

RainToday	No		Yes	
RainTomorrow	No	Yes	No	Yes
WindGustDir				
E	30	4	3	0
ENE	25	0	3	2
ESE	14	4	3	2
N	18	1	1	1
NE	12	2	1	1
NNE	6	1	1	0
NNW	28	7	4	5
NW	45	11	12	5
S	13	2	5	2
SE	12	0	0	0
SSE	6	2	3	1
SSW	3	2	0	0
SW	1	2	0	0
W	9	5	4	2
WNW	28	2	5	0
WSW	2	0	0	0

In [16]:
```
sns.heatmap(regression_analysis,annot=True,fmt="d")
plt.show()
```



In [ ]:
```
```