Group Members:

Hamza Badar, Abdul Wahab Madni

In [3]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [4]:
```
df=pd.read_csv("solar_panel.csv")
```

In [20]:
```
df
```

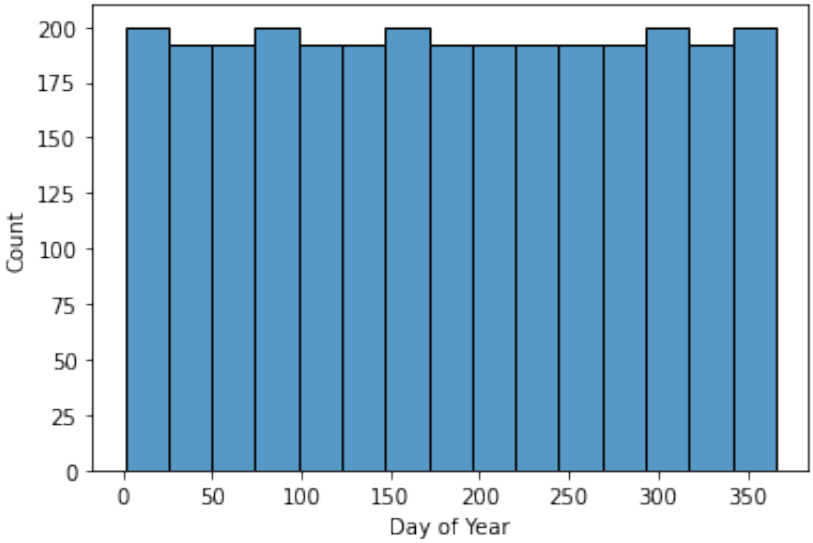
Out[20]:

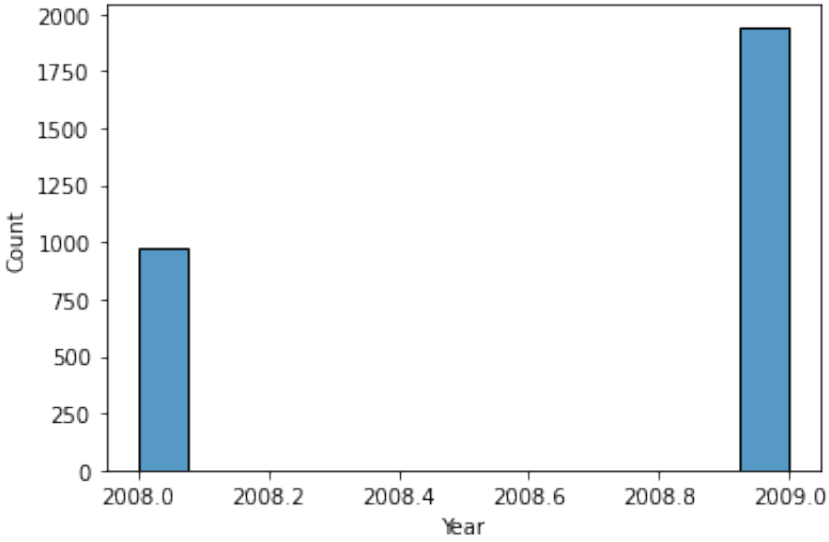
	Day of Year	Year	Month	Day	First Hour of Period	Is Daylight	Distance to Solar Noon	Average Temperature (Day)	Average Wind Direction (Day)	Aver W Sp (I
0	245	2008	9	1	1	False	0.859897	69	28	
1	245	2008	9	1	4	False	0.628535	69	28	
2	245	2008	9	1	7	True	0.397172	69	28	
3	245	2008	9	1	10	True	0.165810	69	28	
4	245	2008	9	1	13	True	0.065553	69	28	
...	...	...	...	...	...	...	...	...	...	
2915	243	2009	8	31	10	True	0.166453	63	27	
2916	243	2009	8	31	13	True	0.064020	63	27	
2917	243	2009	8	31	16	True	0.294494	63	27	
2918	243	2009	8	31	19	True	0.524968	63	27	
2919	243	2009	8	31	22	False	0.755442	63	27	

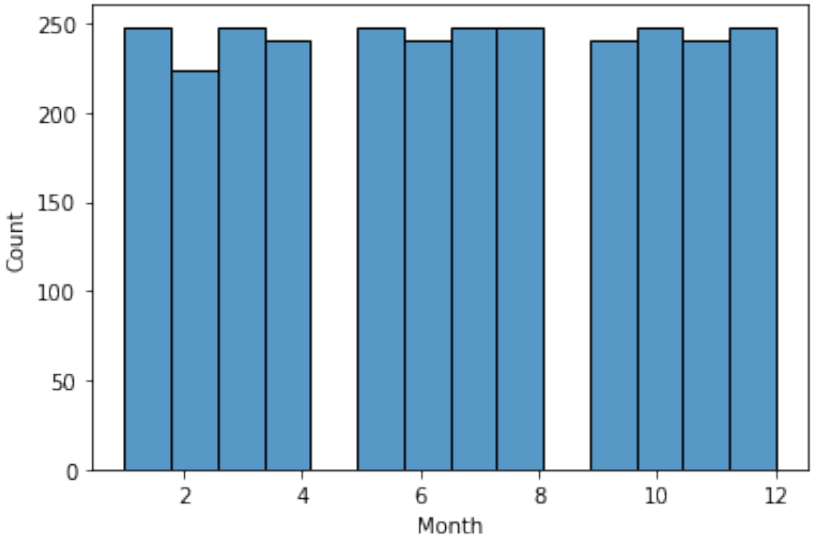
2920 rows × 16 columns

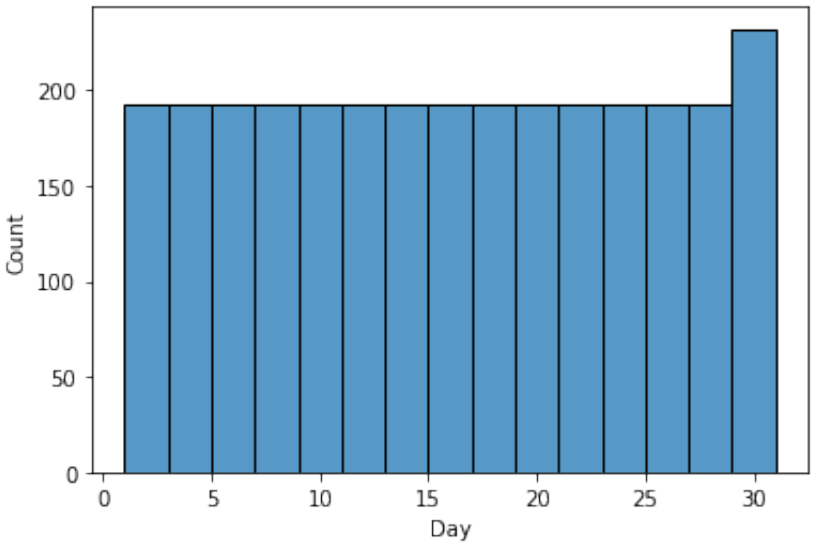
Plotting Histogram of all columns

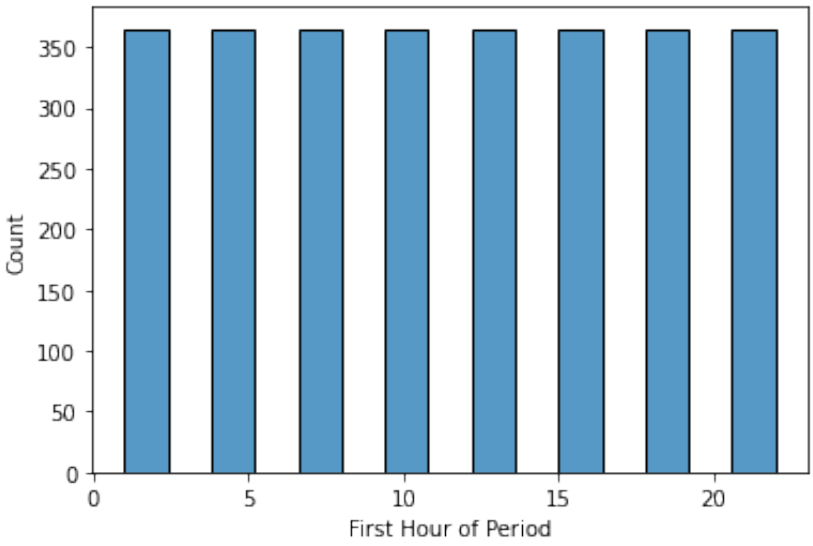
In [22]:
```
for c in df.columns:
    plt.figure()
    if(df[c].dtype==bool):
        # here 1 means daylight(true), 0 means no daylight (false)
        sns.histplot(data = df[c].astype('int'))
    else:
        sns.histplot(x = c, data = df)
```

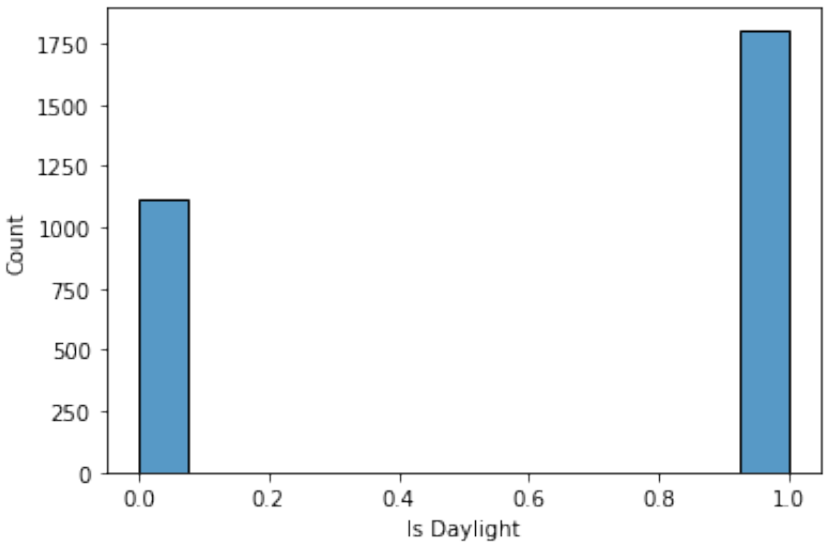

Count
Day of Year
0
25
50
75
100
125
150
175
200
0
50
100
150
200
250
300
350

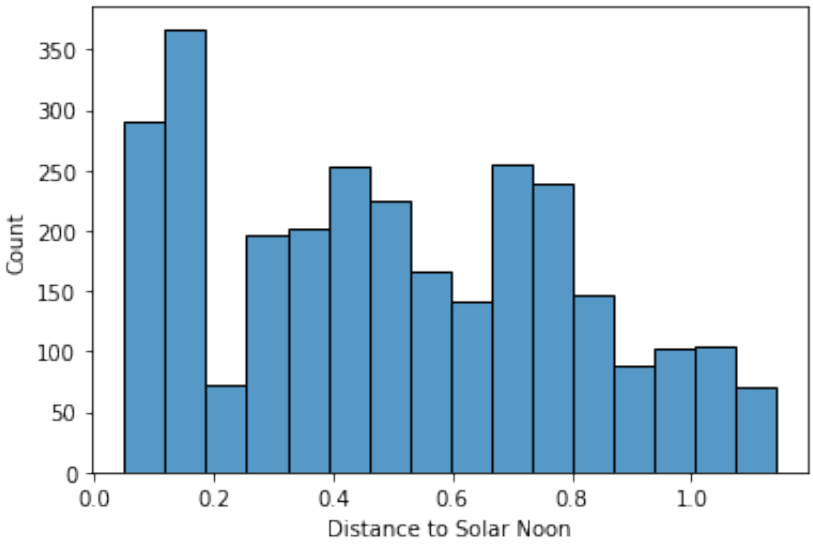

Count
Year
0
250
500
750
1000
1250
1500
1750
2000
2008.0
2008.2
2008.4
2008.6
2008.8
2009.0

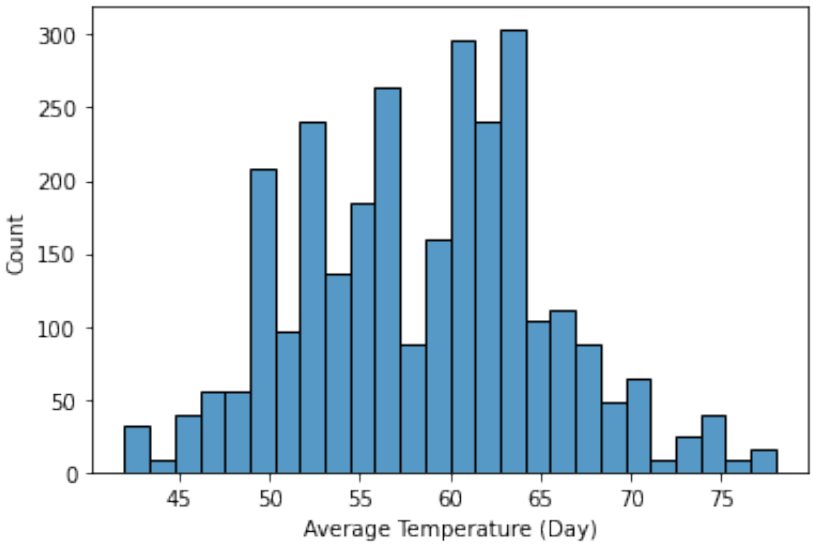

Count
Month
0
50
100
150
200
250
2
4
6
8
10
12

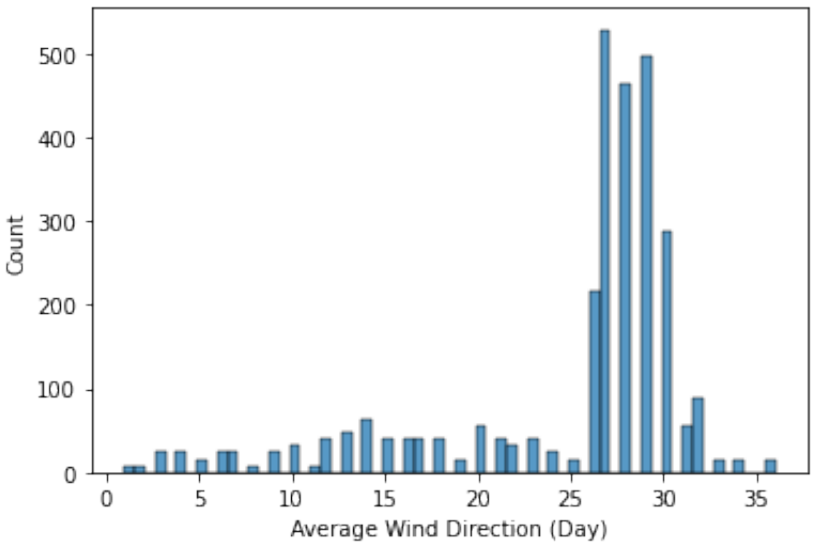

Count
Day
0
50
100
150
200
0
5
10
15
20
25
30

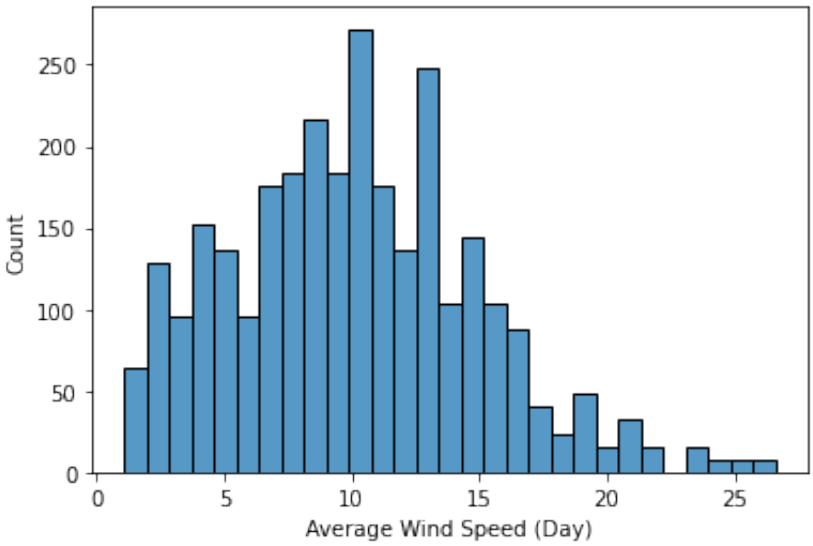

Count
First Hour of Period
0
50
100
150
200
250
300
350
0
5
10
15
20

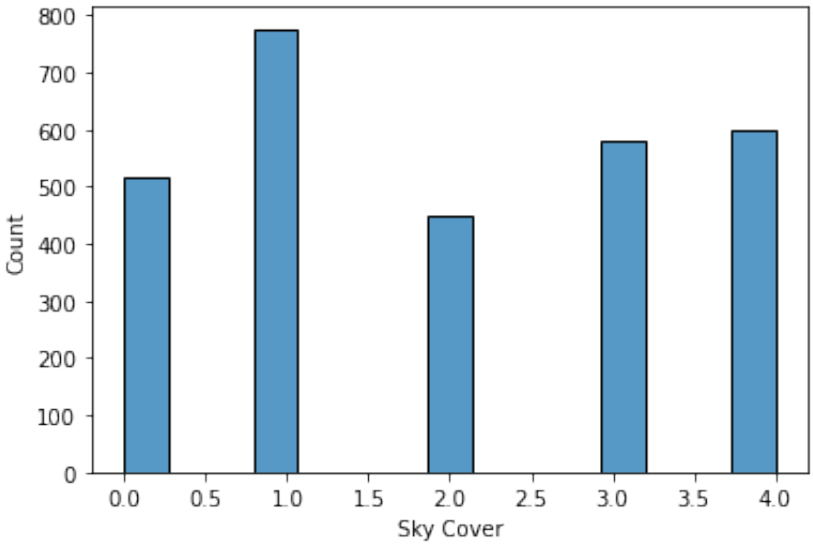

Count
Is Daylight
0
250
500
750
1000
1250
1500
1750
0.0
0.2
0.4
0.6
0.8
1.0

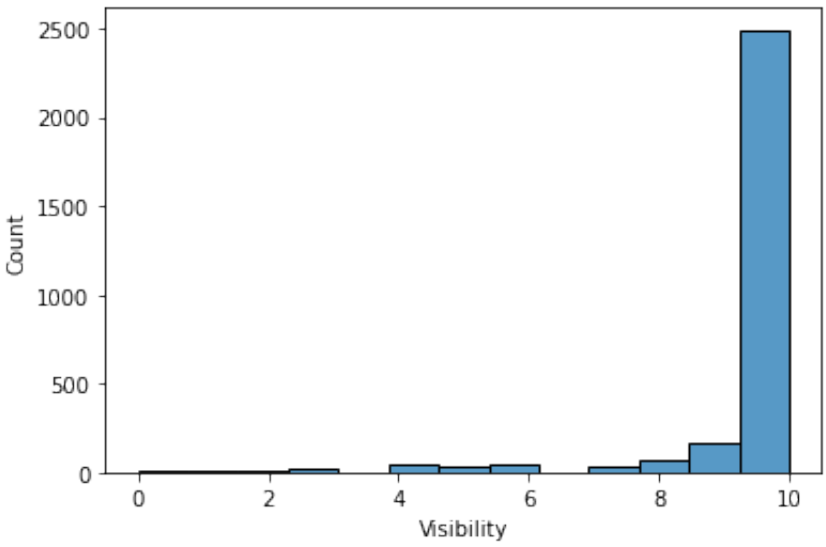

Count
Distance to Solar Noon
0
50
100
150
200
250
300
350
0.0
0.2
0.4
0.6
0.8
1.0

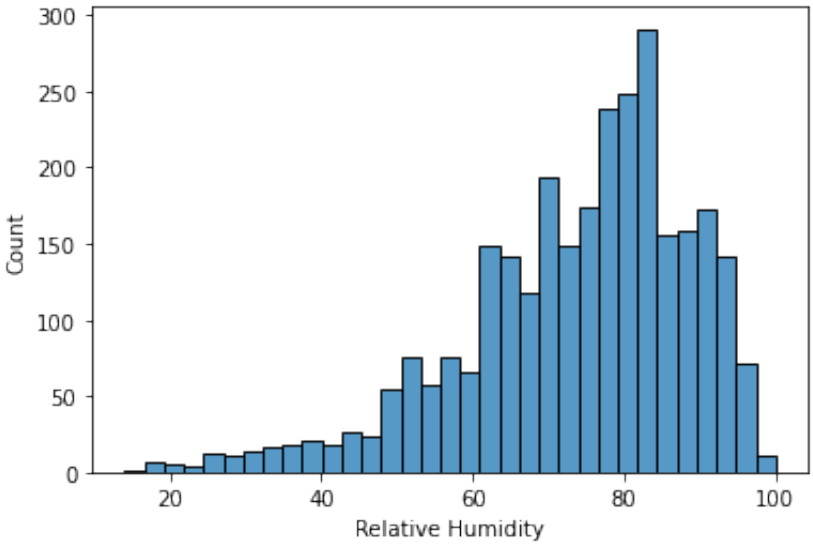

Count
Average Temperature (Day)
0
50
100
150
200
250
300
45
50
55
60
65
70
75

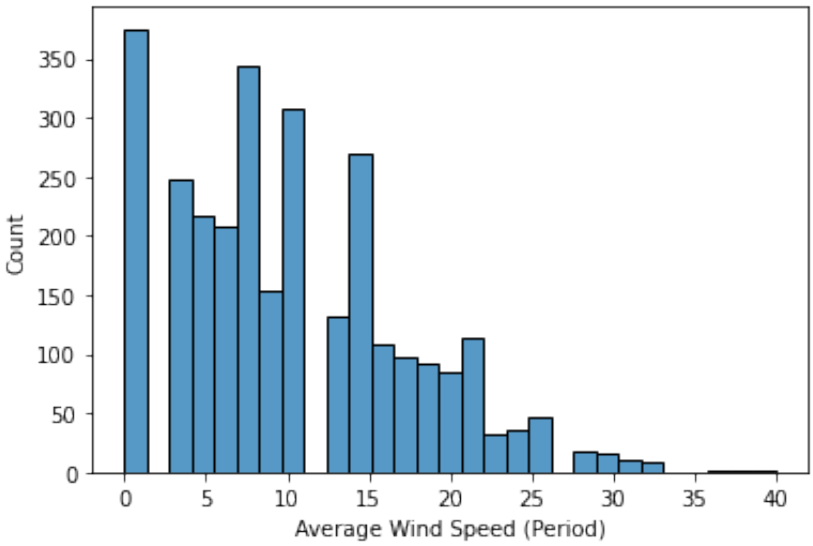

Count
Average Wind Direction (Day)
0
100
200
300
400
500
0
5
10
15
20
25
30
35

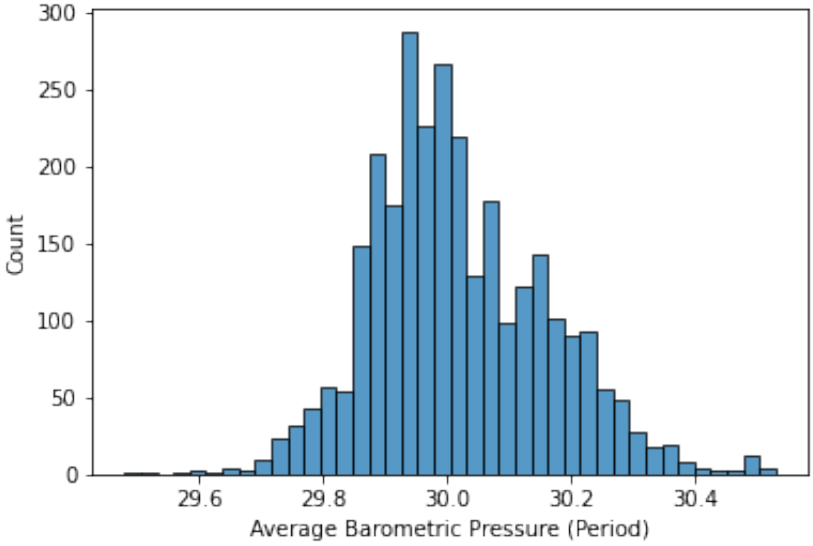

Count
Average Wind Speed (Day)

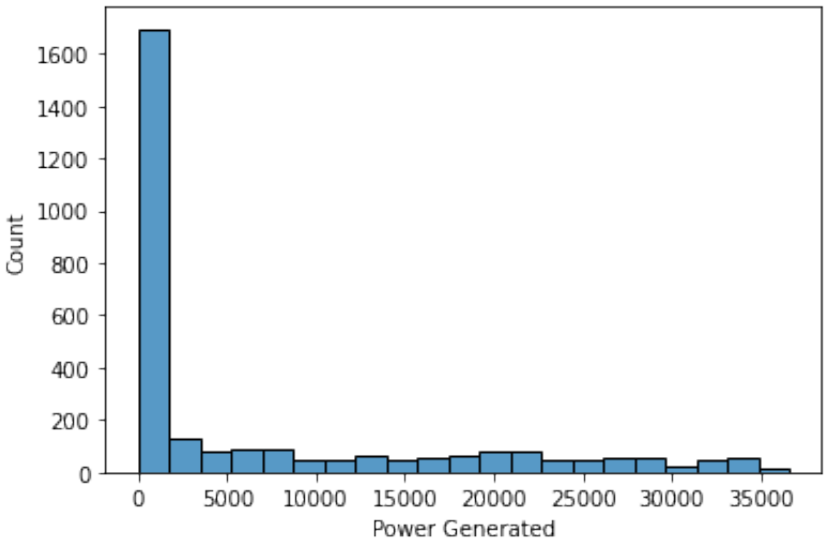

Count
Sky Cover


Count
Visibility


Count
Relative Humidity
0
50
100
150
200
250
300
20
40
60
80
100


Count
Average Wind Speed (Period)
0
50
100
150
200
250
300
350
0
5
10
15
20
25
30
35
40


Count
Average Barometric Pressure (Period)
0
50
100
150
200
250
300
29.6
29.8
30.0
30.2
30.4



Generated Power over the year

We have noticed that there are multiple power values per day as we have a data on the hour level. Since we are interested in finding the overall trend and at the end finding the max power generated, we aggregated our data to day level by taking the sum of power generated per day.

From the below graph & analysis, we have concluded that the maximum power generated is 97262 (day 140)

In [24]:
```
max_power_per_day = df[["Day of Year","Power Generated"]].groupby(["Day o
```

In [25]:
```
max_power_per_day
```

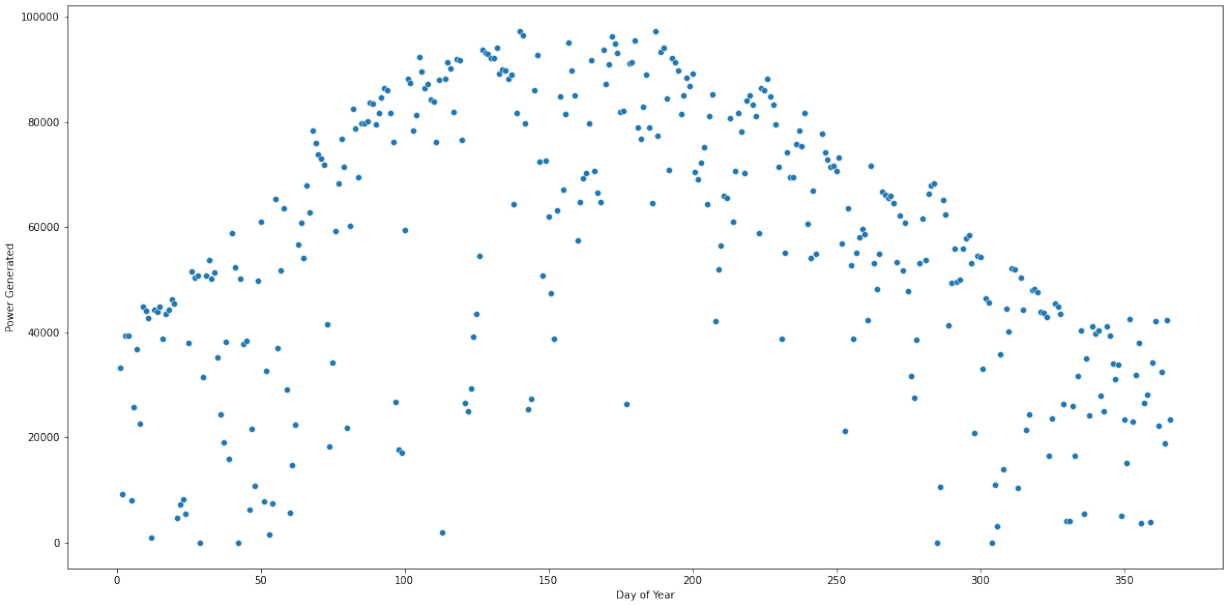
Out[25]:

	Day of Year	Power Generated
0	1	33282
1	2	9197
2	3	39376
3	4	39309
4	5	8010
...	...	...
360	362	22187
361	363	32431
362	364	18916
363	365	42352
364	366	23504

365 rows × 2 columns

In [26]:
```
plt.figure(figsize = (20,10))
sns.scatterplot(data=max_power_per_day, x="Day of Year", y="Power Generat
```

Out[26]:
```
<AxesSubplot:xlabel='Day of Year', ylabel='Power Generated'>
```



In [27]:
```
max_power = max(max_power_per_day["Power Generated"])
max_power_per_day[max_power_per_day["Power Generated"]==max_power]
```

Out[27]:

	Day of Year	Power Generated
139	140	97262

Relation between weather & power generated

In [5]:
```
import numpy as np
```

In [18]:
```
aggdf = df[["Day of Year","Sky Cover","Power Generated"]].groupby(["Day o

aggdf
```

Out[18]:

	Day of Year	avg_sky_cover	power_generated
0	1	3.250	33282
1	2	3.250	9197
2	3	1.750	39376
3	4	1.750	39309
4	5	3.875	8010
...	...	...	...
360	362	2.375	22187
361	363	3.000	32431
362	364	2.250	18916
363	365	1.625	42352
364	366	3.375	23504

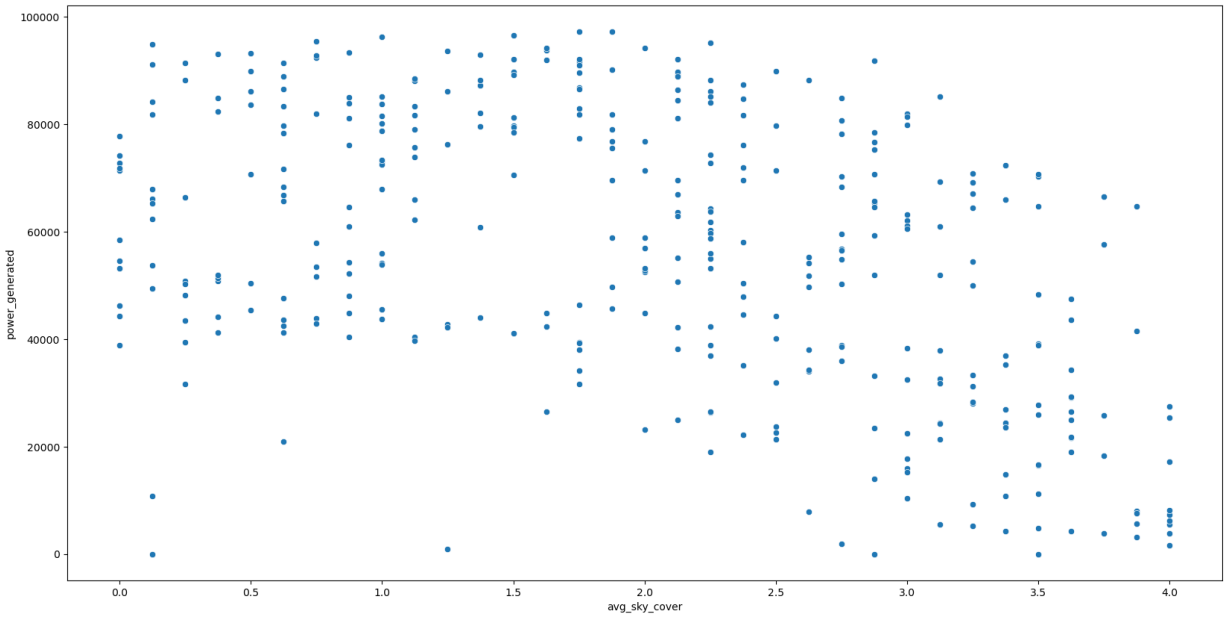
365 rows × 3 columns

In [26]:
```
aggdf["avg_sky_cover"].describe()
```

Out[26]:
```
count    365.000000
mean       1.987671
std        1.141298
min        0.000000
25%        1.000000
50%        2.125000
75%        2.875000
max        4.000000
Name: avg_sky_cover, dtype: float64
```

In [19]:
```
plt.figure(figsize = (20,10))
sns.scatterplot(data=aggdf, x='avg_sky_cover', y='power_generated')
```

Out[19]:
```
<Axes: xlabel='avg_sky_cover', ylabel='power_generated'>
```



In [23]:
```
Threshold = 2.5
```

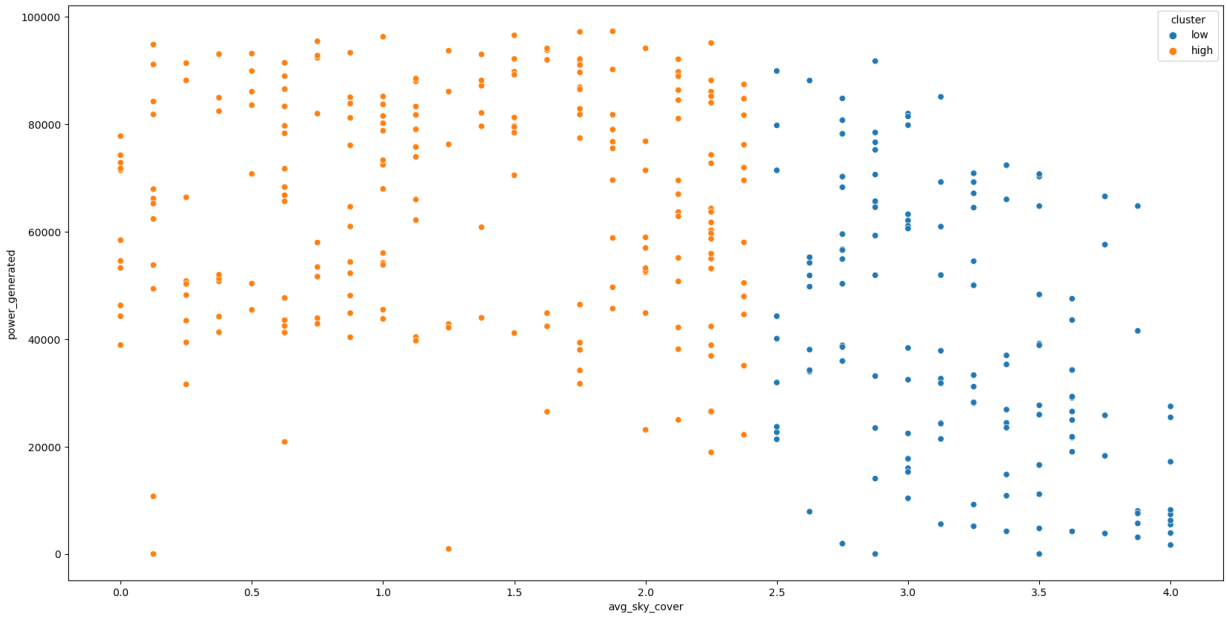
From the data, we can observe that the weather affects the power generation.

The more the sky is covered, less the electricity is generated. This connection remains same over the year. The threshold can be observed at 2.5 sky coverage. To understand it better, we have made a column as 'cluster' containing lower and higher values of 'power_generated' than the threshold.

In [24]:
```
aggdf["cluster"] = np.where(aggdf["avg_sky_cover"]>=Threshold, 'low', 'hi
```

In [25]:
```
plt.figure(figsize = (20,10))
sns.scatterplot(data=aggdf, x="avg_sky_cover", y="power_generated", hue =
```

Out[25]:
```
<Axes: xlabel='avg_sky_cover', ylabel='power_generated'>
```


cluster
low
high
power_generated
avg_sky_cover
100000
80000
60000
40000
20000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0