Yulu Business Case: Hypothesis Testing

Problem Statement

The company wants to know:

Which variables are significant in predicting the demand for shared electric cycles in the Indian market?

How well those variables describe the electric cycle demands?

Whether the variables pose significant effect on demand for the electric cycles or not?

Exploratory Data Analysis

In [1]:

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

In [2]:

```
df_yulu=pd.read_csv('bike_sharing.txt')
```

In [3]:

```
df_yulu.head()
```

Out[3]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0	3	13	16
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0	8	32	40
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0	5	27	32
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0	3	10	13
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0	0	1	1

Description of the Dataframe

datetime: datetime

season: season (1: spring, 2: summer, 3: fall, 4: winter)

holiday: whether day is a holiday or not

workingday: if day is neither weekend nor holiday is 1, otherwise is 0.

weather: 1: Clear, Few clouds, partly cloudy, partly cloudy 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog

temp: temperature in Celsius

atemp: feeling temperature in Celsius

humidity: humidity

windspeed: wind speed

casual: count of casual users

registered: count of registered users

count: count of total rental cycles including both casual and registered

In [4]:

```
np.shape(df_yulu)
```

Out[4]:

```
(10886, 12)
```

In [5]:

```
np.ndim(df_yulu)
```

Out[5]:

```
2
```

In [6]:

```
len(df_yulu)
```

Out[6]:

```
10886
```

In [7]:

```
#Checking the data typeof all the attributes
df_yulu.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   datetime    10886 non-null  object 
 1   season      10886 non-null  int64  
 2   holiday     10886 non-null  int64  
 3   workingday  10886 non-null  int64  
 4   weather     10886 non-null  int64  
 5   temp        10886 non-null  float64
 6   atemp       10886 non-null  float64
 7   humidity    10886 non-null  int64  
 8   windspeed   10886 non-null  float64
 9   casual      10886 non-null  int64  
 10  registered  10886 non-null  int64  
 11  count       10886 non-null  int64  
dtypes: float64(3), int64(8), object(1)
memory usage: 1020.7+ KB
```

Looking at the basic EDA we can observe that the data is a 2D list with length 10886 and shape (10886,12), which means the data frame has 10866 rows and 12 columns

In [8]:

```
df_yulu.describe()
```

Out[8]:

	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	
count	10886.000000	10886.000000	10886.000000	10886.000000	10886.00000	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000	1088
mean	2.506614	0.028569	0.680875	1.418427	20.23086	23.655084	61.886460	12.799395	36.021955	155.552177	19
std	1.116174	0.166599	0.466159	0.633839	7.79159	8.474601	19.245033	8.164537	49.960477	151.039033	18
min	1.000000	0.000000	0.000000	1.000000	0.82000	0.760000	0.000000	0.000000	0.000000	0.000000	
25%	2.000000	0.000000	0.000000	1.000000	13.94000	16.665000	47.000000	7.001500	4.000000	36.000000	4
50%	3.000000	0.000000	1.000000	1.000000	20.50000	24.240000	62.000000	12.998000	17.000000	118.000000	14
75%	4.000000	0.000000	1.000000	2.000000	26.24000	31.060000	77.000000	16.997900	49.000000	222.000000	28
max	4.000000	1.000000	1.000000	4.000000	41.00000	45.455000	100.000000	56.996900	367.000000	886.000000	97

Range of the attributes:

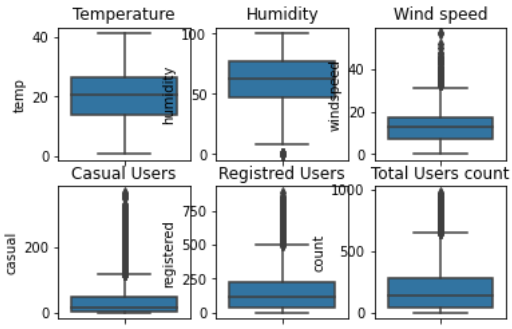
1. The maximum temperature that was observed was 41 celcisus and minimum was 0.82 celcius with the overall mean temperature being 20.23 Celciusm. When we observe the median temp, it is 20.5 which is almost close to the mean temperature which means there is no much possibility of outliers.
2. The maximum humidity was 100 and minimum was 0 and mean humidity being 61.87. The median of huidity is 62 which is close to the mean value of humidity, which implies less outliers
3. The maximum value of windspeed was found to be 56.99 and the minimum was found to be 0, with mean being 12.8. The median of the windspeed is 12.99 which is close to the mean value of the windspeed, reducing the outliers
4. The maximum number of casual users on a single day was 367 and the minimum was found to be 0 whereas the average number of casual users per day is 36. The median value of the casual users was found to be 17 which implies there is a huge possibility of outliers being present in this attribute.
5. The maximum number of registered users on a single day was 886 and minimum was 0,mean being 155.55. The median of this attribute is 118 which is not very close to the mean value. Therefore there is a possibility of outliers being present in this attribute.
6. The maximum of total count of users in a single day was found to be 977 and minimum was found to be 1,average being 191.57. The median of this attribute is 145 which indicates the presence of outliers.

In [39]:

```
#Checking for Outliers
plt.subplot(2,3,1)
sns.boxplot(data=df_yulu,y='temp')
plt.title('Temperature')

plt.subplot(2,3,2)
sns.boxplot(data=df_yulu,y='humidity')
plt.title('Humidity')

plt.subplot(2,3,3)
sns.boxplot(data=df_yulu,y='windspeed')
plt.title('Wind speed')

plt.subplot(2,3,4)
sns.boxplot(data=df_yulu,y='casual')
plt.title('Casual Users')

plt.subplot(2,3,5)
sns.boxplot(data=df_yulu,y='registered')
plt.title('Registred Users')

plt.subplot(2,3,6)
sns.boxplot(data=df_yulu,y='count')
plt.title('Total Users count')
```

Out[39]:

```
Text(0.5, 1.0, 'Total Users count')
```



From the above plots we can see the presence of outliers in the attributes: windspeed, casual users, registred users, and count

In [9]:

```
#Checking for null values
df_yulu.isna().sum()
```

Out[9]:

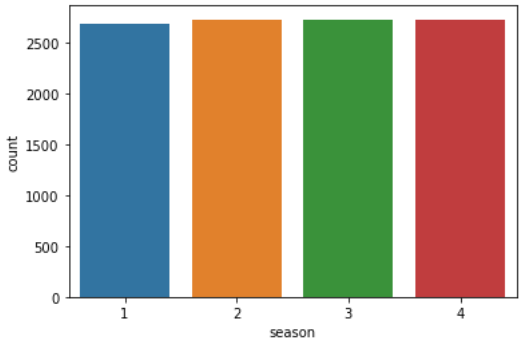
```
datetime      0
season        0
holiday       0
workingday    0
weather       0
temp          0
atemp         0
humidity      0
windspeed     0
casual        0
registered    0
count         0
dtype: int64
```

This implies there are no null values present in the entire dataset

Univariate Analysis

In [10]:

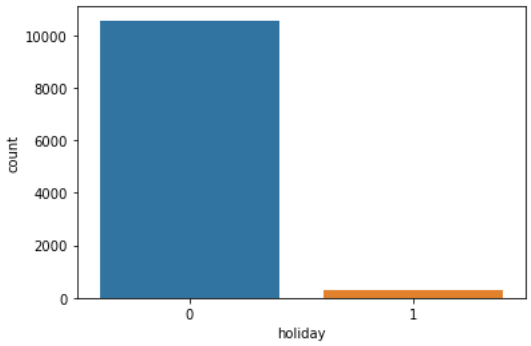
```
sns.countplot(data=df_yulu,x='season')
plt.show()
```



From the above plot we can observe that, in all the seasons there was similar demand for the electric cycle. Which means we can assume the demand of the electric cycle does not vary from season to season

In [11]:

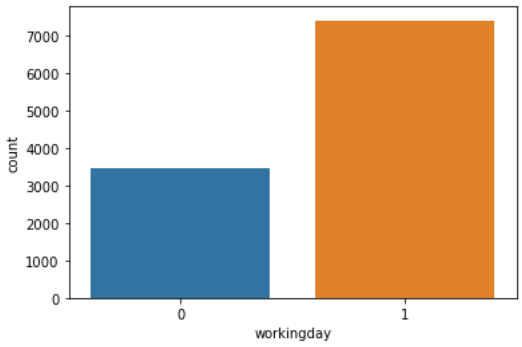
```
sns.countplot(data=df_yulu,x='holiday')
plt.show()
```



From the plot we can see that the moreumber of electric cycles were rented during the day which is not a holiday.

In [12]:

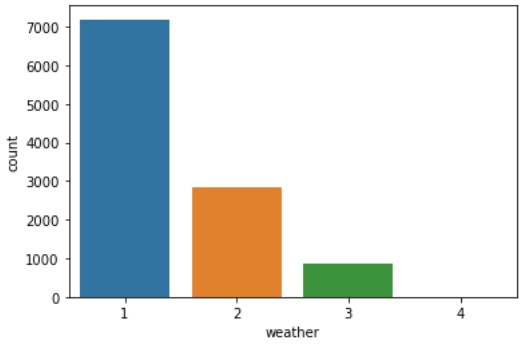
```
sns.countplot(data=df_yulu,x='workingday')
plt.show()
```



From the above plot we can asume that the demand for the electric cycles was the maximum during a working day rather than a holiday

In [13]:

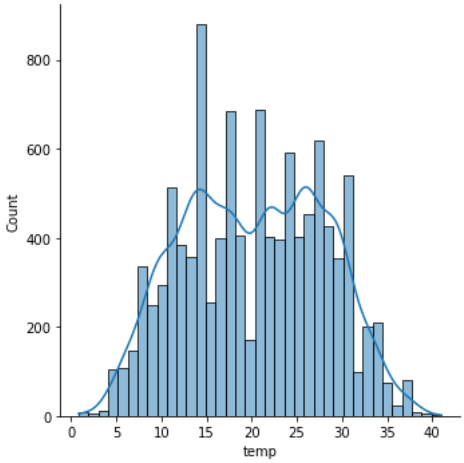
```
sns.countplot(data=df_yulu,x='weather')
plt.show()
```



From the above plot we can see that the maximum number of cycles were rented in the weather condition 1 which is clear sky or partly cloudy. We also can see there was no demand for the cycles in the extreme weather condition which is 4

In [14]:

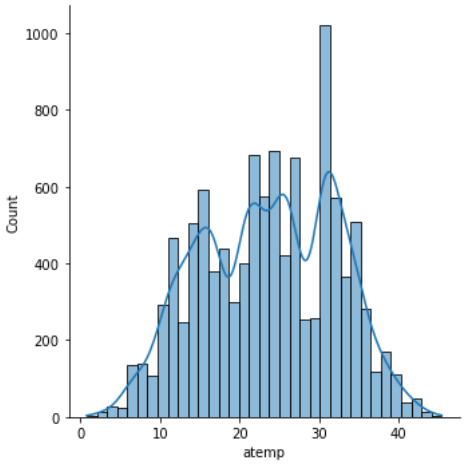
```
sns.displot(data=df_yulu,x='temp',kde=True)
plt.show()
```



From the above distribution we can say the mean of the temperature lies roughly around 20 Celsius

In [15]:

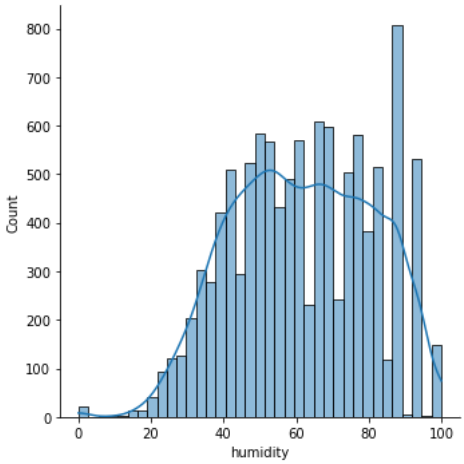
```
sns.displot(data=df_yulu,x='atemp',kde=True)
plt.show()
```



From the above distribution we can say the mean of the distribution lies roughly around 25 Celsius

In [16]:

```
sns.displot(data=df_yulu,x='humidity',kde=True)
plt.show()
```

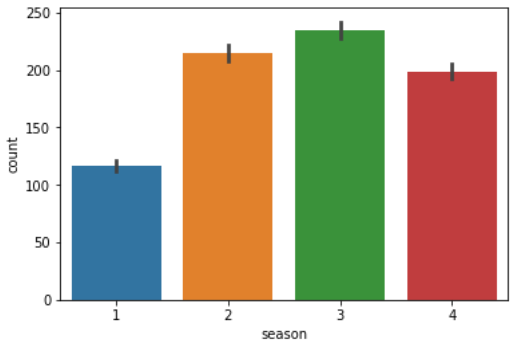


From the distribution we can say, the mean of distribution lies around 60

Bivariate Analysis

In [17]:

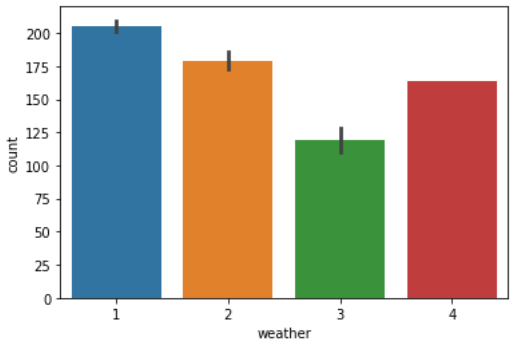
```
#Count of cycles rented Vs Season
sns.barplot(data=df_yulu,
            x='season',
            y='count')
plt.show()
```



From the above plot, we can observe that the maximum number of electric cycles were rented during the season 3 which is fall

In [18]:

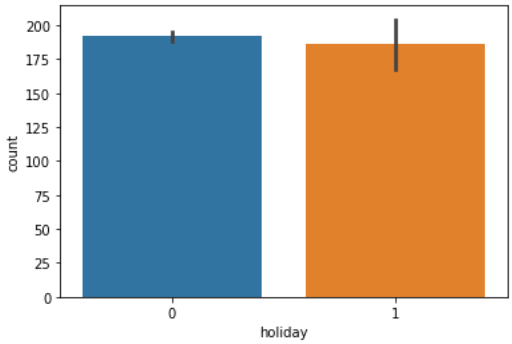
```
#Count of cycles rented vs Weather Condition
sns.barplot(data=df_yulu,
            x='weather',
            y='count')
plt.show()
```



From the above plot we can observe that the maximum number of electric cycles were rented during the weather condition 1 which is clear or partly cloudy, followed by weather condition 2 which is mist and few clouds.

In [19]:

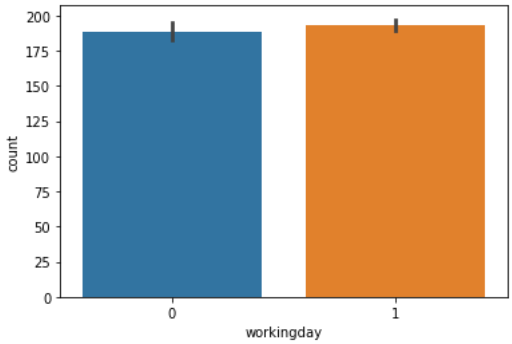
```
#Count of cycles rented vs holiday
sns.barplot(data=df_yulu,
            x='holiday',
            y='count')
plt.show()
```



From the above plot we can understand that there is no much effect on the demand of the elctric cycles whether it is holiday or not. We can see almost similar number of electric vehicles being rented in both the cases

In [20]:

```
#Count of cycles rented vs working day
sns.barplot(data=df_yulu,
            x='workingday',
            y='count')
plt.show()
```



From the above plot we can understand that the there is no much effect whether the day is working day or not. We can see similar number of electric vehicles being rented in both the cases

2 Sample T Test to check if Working Day has an effect on the number of electric cycles rented.

In [3]:

```
from scipy.stats import ttest_ind
```

In [4]:

```
df_yulu.head()
```

Out[4]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0	3	13	16
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0	8	32	40
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0	5	27	32
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0	3	10	13
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0	0	1	1

In [5]:

```
df_workingday=df_yulu[df_yulu['workingday']==1]['count']
```

In [6]:

```
df_non_workingday=df_yulu[df_yulu['workingday']==0]['count']
```

In [9]:

```
#Null Hypothesis(H0): There is no effect of working day on the number of electric cycles rented
#Alternate Hypothesis(Ha): There is a significant effect of working day on the number of electric cycles rented

#Let us set a significance level alpha=0.05

alpha=0.05
tstat,p_value=ttest_ind(df_workingday,df_non_workingday)

print('p_value:',p_value)
if p_value<alpha:
    print('Reject Null Hypothesis')
    print(' There is a significant effect of working day on the number of electric cycles rented')
else:
    print('Fail to reject Null Hypothesis')
    print('There is no significant effect of workingday on the number of electric cycles rented')
```

```
p_value: 0.22644804226361348
Fail to reject Null Hypothesis
There is no significant effect of workingday on the number of electric cycles rented
```

From the above test we can see that the p value is 0.22 which is very higher than the significance level. Which means we cannot reject the null hypothesis and in case the null hypothesis holds true. Therefore, we can say that there is no significant effect of working day on the number of electric cycles being rented.

ANNOVA to check if No. of cycles rented is similar or different in different 1. weather 2. season

In [10]:

```
from scipy.stats import f_oneway
```

1. No.of cycles rented Vs Different weather conditions

In [13]:

```
df_yulu.head()
```

Out[13]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0	3	13	16
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0	8	32	40
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0	5	27	32
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0	3	10	13
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0	0	1	1

In [14]:

```
weather1=df_yulu[df_yulu['weather']==1]['count']
```

In [16]:

```
weather2=df_yulu[df_yulu['weather']==2]['count']
```

In [17]:

```
weather3=df_yulu[df_yulu['weather']==3]['count']
```

In [18]:

```
weather4=df_yulu[df_yulu['weather']==4]['count']
```

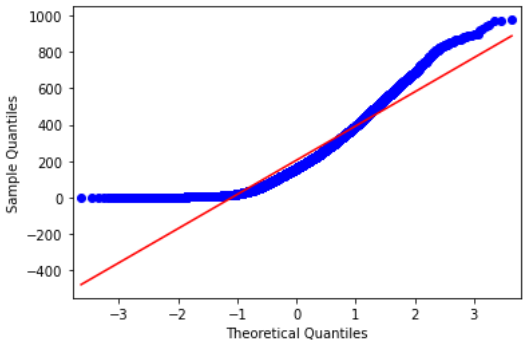
Checking the Assumptions of Anova

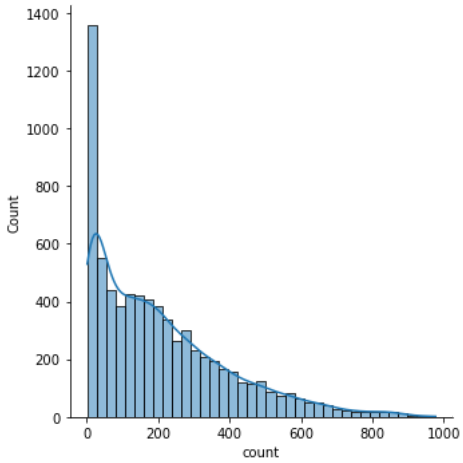
In [41]:

```
#Checking whether the distirbution is Gaussian or not using qqplot
from statsmodels.graphics.gofplots import qqplot
```

In [75]:

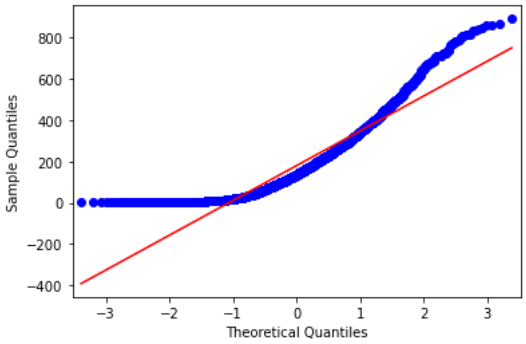
```
qqplot(weather1,line='s')
plt.show()

#Visual representation of the distribution
sns.displot(weather1,kde=True)
plt.show()
```

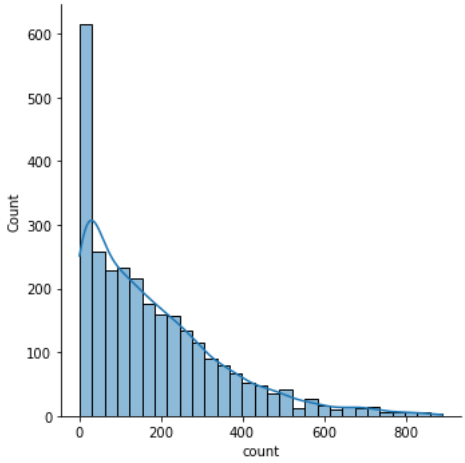




In [77]:

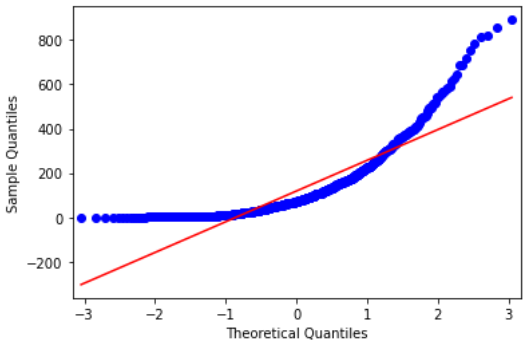
```
qqplot(weather2,line='s')
plt.show()

sns.displot(weather2,kde=True)
plt.show()
```

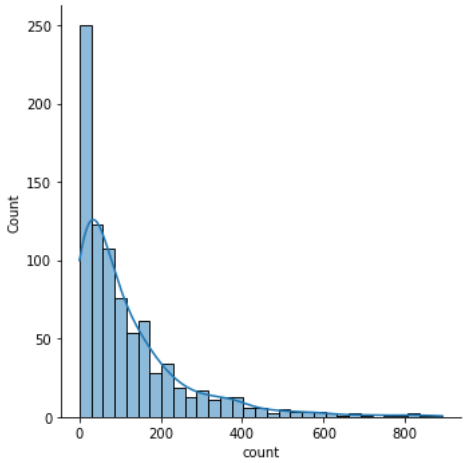




In [79]:

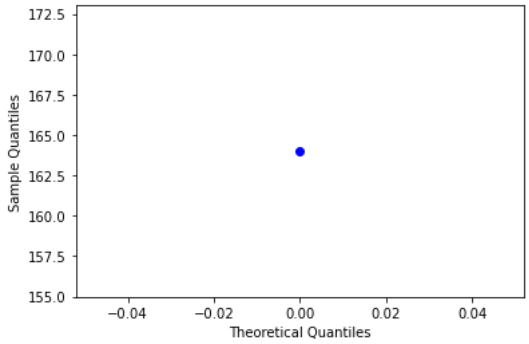
```
qqplot(weather3,line='s')
plt.show()

sns.displot(weather3,kde=True)
plt.show()
```



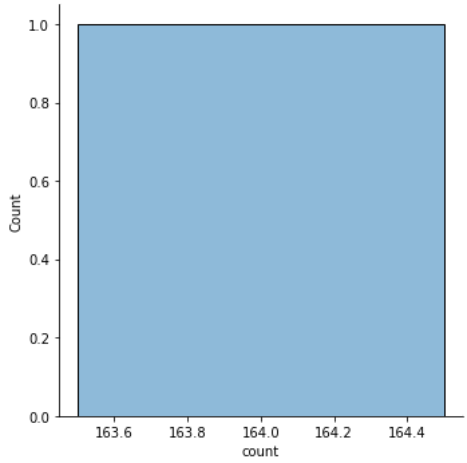


In [80]:

```
qqplot(weather4,line='s')
plt.show()

sns.displot(weather4,kde=True)
plt.show()
```




```
C:\Users\PC\anaconda3\lib\site-packages\seaborn\distributions.py:306: UserWarning: Dataset has 0 variance; skipping density
estimate.
  warnings.warn(msg, UserWarning)
```



From the above test using qqplot and distribution plots, we can observe that the data does not follow Gaussian Distribution

In [55]:

```
#Checking whether the variances are equal or not using levene test
from scipy.stats import levene

#Null Hypothesis(H0): Variances are equal
#Alternate Hypothesis(Ha): Variances are not equal

#significane level aplha=0.05

levene_stat,p_value=levene(weather1,weather2,weather3,weather4)

alpha=0.05

print('p_value for levene test(different weather conditions):',p_value)

if p_value<alpha:
    print('Reject Null Hypothesis')
    print('Variances are not equal')
else:
    print('Fail to reject Null Hypothesis')
    print('Variances are equal')
```

```
p_value for levene test(different weather conditions): 3.504937946833238e-35
Reject Null Hypothesis
Variances are not equal
```

From the above test for variances it is observed that the variances of all the different weather conditions is not equal.

Since the assumptions of Anova do not hold true in this case, we will use Kruskal Wallis test for testing the Hypothesis

In [56]:

```
from scipy.stats import kruskal
```

In [57]:

```
#Null Hypothesis(H0): The number of electric cycles rented in different weather conditions is same
#Alternate Hypothesis(Ha): The number of electric cycles rented in different weather conditions is different

#Let us set a significance level alpha=0.05

alpha=0.05
kruskal_stat,p_value=kruskal(weather1,weather2,weather3,weather4)

print('p_value(kruskal test for different weather conditions):',p_value)
if p_value<alpha:
    print('Reject Null Hypothesis')
    print('The number of electric cycles rented in different weather conditions is different')
else:
    print('Fail to reject Null Hypothesis')
    print(' The number of electric cycles rented in different weather conditions is same')
```

```
p_value(kruskal test for different weather conditions): 3.501611300708679e-44
Reject Null Hypothesis
The number of electric cycles rented in different weather conditions is different
```

From the above test we can observe that the p value is significantly low, which prompts us to reject the null hypothesis. Therefore we can say that the number of electric cycles rented in different weather conditions is different

2. No.of cycles rented Vs Different Seasons

In [26]:

```
season1=df_yulu[df_yulu['season']==1]['count']
```

In [27]:

```
season2=df_yulu[df_yulu['season']==2]['count']
```

In [28]:

```
season3=df_yulu[df_yulu['season']==3]['count']
```

In [29]:

```
season4=df_yulu[df_yulu['season']==4]['count']
```

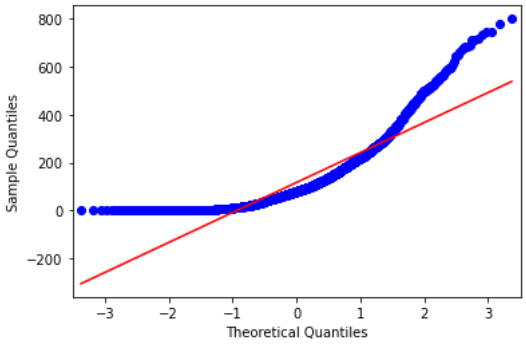
Checking the assumptions of Anova

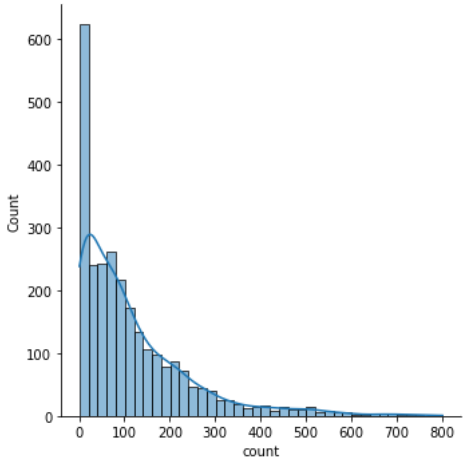
In [58]:

```
#Checking whether the distirbution is Gaussian or not using qqplot
from statsmodels.graphics.gofplots import qqplot
```

In [83]:

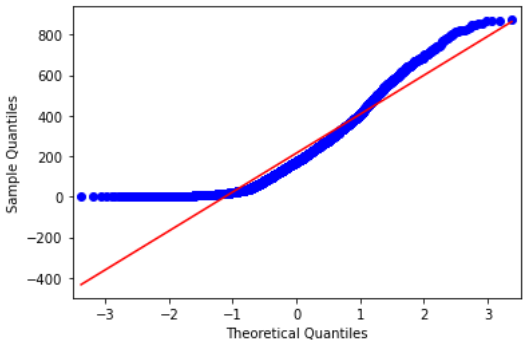
```
qqplot(season1,line='s')
plt.show()

sns.displot(season1,kde=True)
plt.show()
```

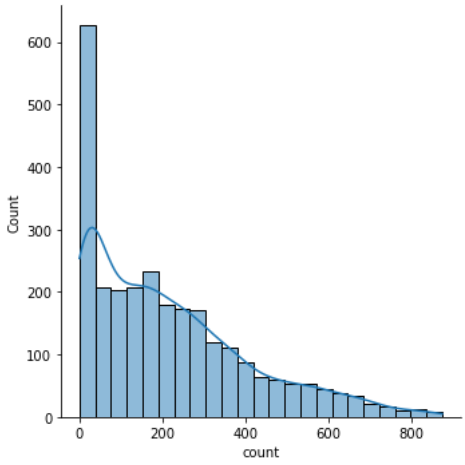




In [84]:

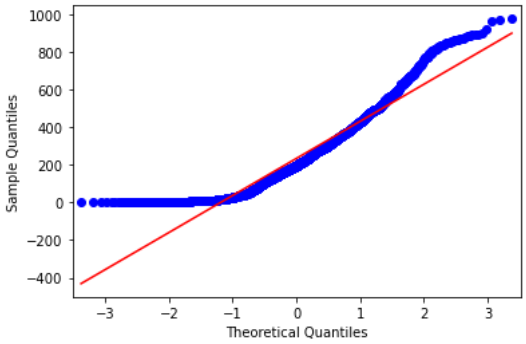
```
qqplot(season2,line='s')
plt.show()

sns.displot(season2,kde=True)
plt.show()
```

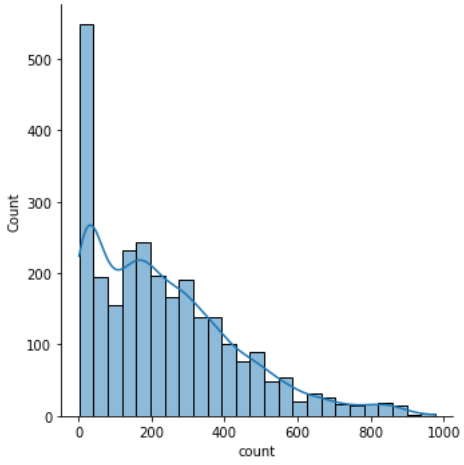




In [85]:

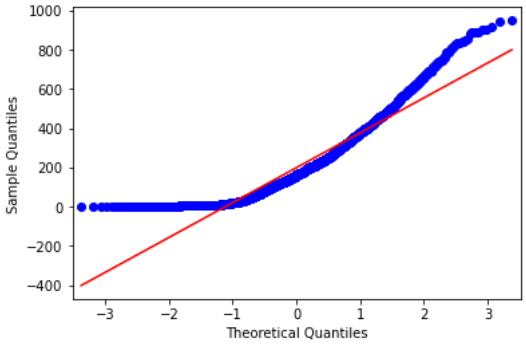
```
qqplot(season3,line='s')
plt.show()

sns.displot(season3,kde=True)
plt.show()
```

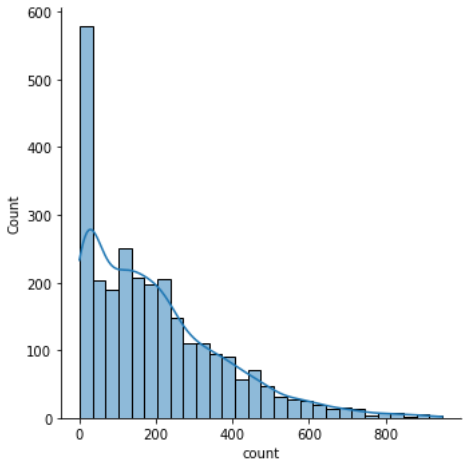




In [86]:

```
qqplot(season4,line='s')
plt.show()

sns.displot(season4,kde=True)
plt.show()
```





From the above test using qqplot and distributon plots, we can observe that the data does not follow Gaussian Distribution

In [63]:

```
#Checking whether the variances are equal or not using levene test
from scipy.stats import levene

#Null Hypothesis(H0): Variances are equal
#Alternate Hypothesis(Ha): Variances are not equal

#significane level aplha=0.05

levene_stat,p_value=levene(season1,season2,season3,season4)

alpha=0.05

print('p_value for levene test(different seasons):',p_value)

if p_value<alpha:
    print('Reject Null Hypothesis')
    print('Variances are not equal')
else:
    print('Fail to reject Null Hypothesis')
    print('Variances are equal')
```

```
p_value for levene test(different seasons): 1.0147116860043298e-118
Reject Null Hypothesis
Variances are not equal
```

From the above test for variances it is observed that the variances of all the different weather conditions is not equal.

Since the assumptions of Anova do not hold true in this case, we will use Kruskal Wallis test for testing the Hypothesis

In [64]:

```
#Null Hypothesis(H0): The number of electric cycles rented in different seasons is same
#Alternate Hypothesis(Ha): The number of electric cycles rented in different seasons is different

#Let us set a significance level alpha=0.05

alpha=0.05
kruskal_stat,p_value=kruskal(season1,season2,season3,season4)

print('p_value(kruskal test for different seasons):',p_value)
if p_value<alpha:
    print('Reject Null Hypothesis')
    print('The number of electric cycles rented in different seasons is different')
else:
    print('Fail to reject Null Hypothesis')
    print(' The number of electric cycles rented in different seasons is same')
```

```
p_value(kruskal test for different seasons): 2.479008372608633e-151
Reject Null Hypothesis
The number of electric cycles rented in different seasons is different
```

From the above test we can observe that the p value is significantly low, which prompts us to reject the null hypothesis. Therefore we can say that the number of electric cycles rented in different seasons is different

Chi-square test to check if Weather is dependent on the season

In [38]:

```
from scipy.stats import chi2_contingency
```

In [35]:

```
df_yulu.head()
```

Out[35]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0	3	13	16
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0	8	32	40
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0	5	27	32
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0	3	10	13
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0	0	1	1

In [36]:

```
contingency=pd.crosstab(df_yulu['season'],df_yulu['weather'])
```

In [37]:

```
contingency
```

Out[37]:

weather	1	2	3	4
season				
1	1759	715	211	1
2	1801	708	224	0
3	1930	604	199	0
4	1702	807	225	0

In [39]:

```
chi2_contingency(contingency)
```

Out[39]:

```
(49.158655596893624,
 1.549925073686492e-07,
 9,
 array([[1.77454639e+03, 6.99258130e+02, 2.11948742e+02, 2.46738931e-01],
        [1.80559765e+03, 7.11493845e+02, 2.15657450e+02, 2.51056403e-01],
        [1.80559765e+03, 7.11493845e+02, 2.15657450e+02, 2.51056403e-01],
        [1.80625831e+03, 7.11754180e+02, 2.15736359e+02, 2.51148264e-01]]))
```

In [40]:

```
#Null Hypothesis(H0): Weather and season are independent
#Alternate Hypothesis(Ha): weather is dependent on season

#Let us set a significance level alpha=0.05

alpha=0.05
chi_stat,p_value3,dof,exp_frequency=chi2_contingency(contingency)

print('p_value:',p_value3)
if p_value3<alpha:
    print('Reject Null Hypothesis')
    print('Weather is dependent on season')
else:
    print('Fail to reject Null Hypothesis')
    print(' Weather and season are independent')
```

```
p_value: 1.549925073686492e-07
Reject Null Hypothesis
Weather is dependent on season
```

From the above test, we can observe p value is low which means the asumption that season and weather are independent does not hold true. From the test results we can conclude that the weather is dependent on season.

Insights and Observations

1. There is no significant effect of working day on the number of electric cycles being rented
2. The number of electric cycles rented in different weather conditions is different
3. The number of electric cycles rented in different seasons is different
4. Weather conditions are dependent on season