Yulu is India's leading micro-mobility service provider, which offers unique vehicles for the daily commute. Starting off as a mission to eliminate traffic congestion in India, Yulu provides the safest commute solution through a user-friendly mobile app to enable shared, solo and sustainable commuting.

Yulu zones are located at all the appropriate locations (including metro stations, bus stands, office spaces, residential areas, corporate offices, etc) to make those first and last miles smooth, affordable, and convenient!

Yulu has recently suffered considerable dips in its revenues. They have contracted a consulting company to understand the factors on which the demand for these shared electric cycles depends. Specifically, they want to understand the factors affecting the demand for these shared electric cycles in the Indian market.

Unsupported Cell Type. Double-Click to inspect/edit the content.

```
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import ttest_ind,chisquare
from scipy.stats import f_oneway, kruskal
from scipy.stats import ttest_ind
from scipy.stats import shapiro
from scipy.stats import levene
from scipy.stats import ks_2samp
from statsmodels.graphics.gofplots import qqplot
```

```
df = pd.read_csv('bike_sharing.csv')
```

```
df.head()
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.39	81	0.0
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.63	80	0.0
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.63	80	0.0
	2011-01-01								

```
df.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
```

```
df.isna().sum()
```

```
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
```

```
df.describe()
```

	season	holiday	workingday	weather	temp	atemp	humidity	windspee
count	10886.00	10886.00	10886.00	10886.00	10886.00	10886.00	10886.00	10886.0
mean	2.51	0.03	0.68	1.42	20.23	23.66	61.89	12.8
std	1.12	0.17	0.47	0.63	7.79	8.47	19.25	8.1
min	1.00	0.00	0.00	1.00	0.82	0.76	0.00	0.0
25%	2.00	0.00	0.00	1.00	13.94	16.66	47.00	7.0
50%	3.00	0.00	1.00	1.00	20.50	24.24	62.00	13.0

75%	4.00	0.00	1.00	2.00	26.24	31.06	77.00	17.0(
max	4.00	1.00	1.00	4.00	41.00	45.45	100.00	57.0(

```
df.describe(include='object')
```

	datetime
count	10886
unique	10886
top	2011-01-01 00:00:00
freq	1

```
df.shape
```

```
(10886, 12)
```

```
df['season'].unique()
```

```
array([1, 2, 3, 4], dtype=int64)
```

```
df['holiday'].unique()
```

```
array([0, 1], dtype=int64)
```

```
df['workingday'].unique()
```

```
array([0, 1], dtype=int64)
```

```
df['weather'].unique()
```

```
array([1, 2, 3, 4], dtype=int64)
```

```
df['temp'].nunique()
```

```
49
```

```
df['atemp'].nunique()
```

```
60
```

```
df['humidity'].nunique()
```

```
89
```

```
df['windspeed'].nunique()
```

```
28
```

```
df['casual'].nunique()
```

```
309
```

```
df['registered'].nunique()
```

```
731
```

```
df['datetime']=pd.to_datetime(df['datetime'])
```

```
df['season']=df['season'].astype('object')
df['holiday']=df['holiday'].astype('object')
df['workingday']=df['workingday'].astype('object')
df['weather']=df['weather'].astype('object')
```

```
df.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  datetime64[ns]
 1   season      10886 non-null  object
 2   holiday     10886 non-null  object
 3   workingday  10886 non-null  object
 4   weather     10886 non-null  object
 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: datetime64[ns](1), float64(3), int64(4), object(4)
memory usage: 1020.7+ KB
```

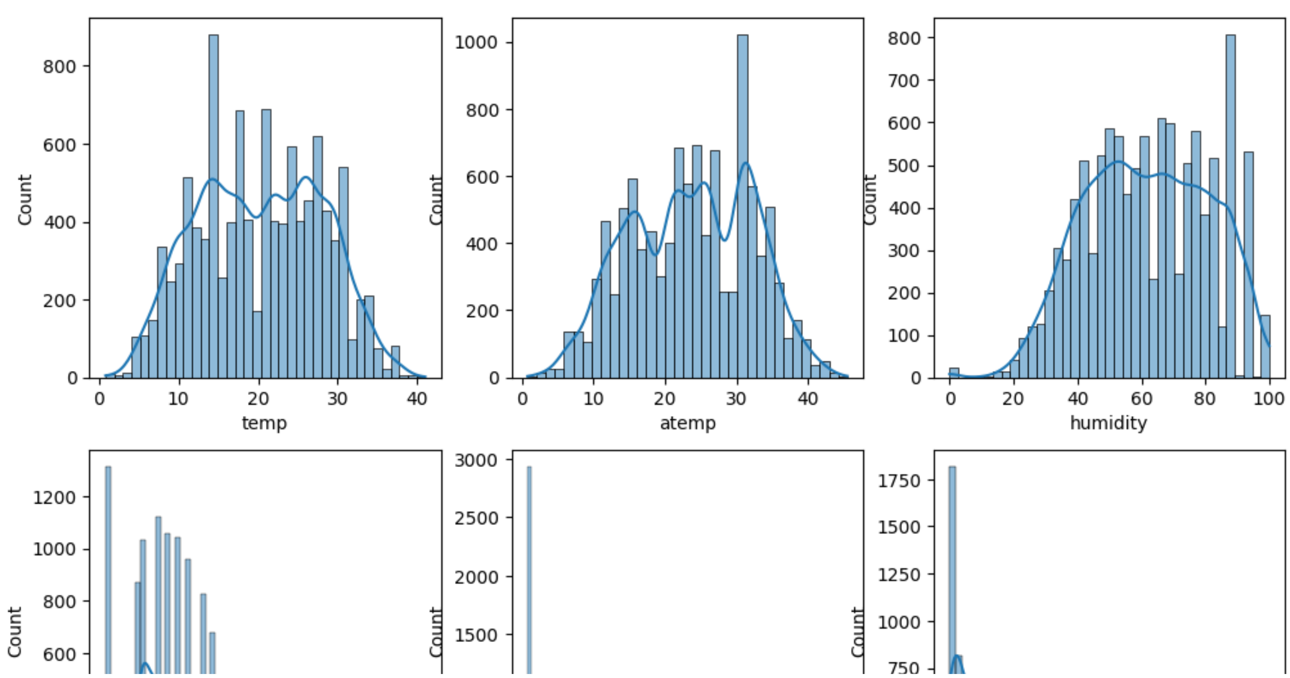
```
df.describe(include='all')
```

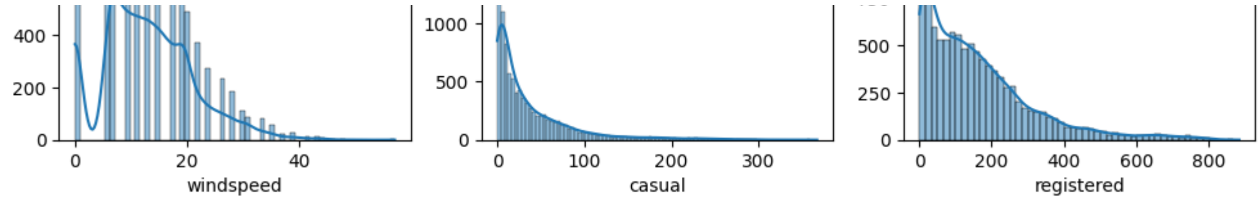
	datetime	**season**	**holiday**	**workingday**	**weather**	**temp**	**atemp**	**h**

count	10886	10886.0	10886.0	10886.0	10886.0	10886.00	10886.00	1
unique	NaN	4.0	2.0	2.0	4.0	NaN	NaN	
top	NaN	4.0	0.0	1.0	1.0	NaN	NaN	
freq	NaN	2734.0	10575.0	7412.0	7192.0	NaN	NaN	
mean	2011-12-27 05:56:22.399411968	NaN	NaN	NaN	NaN	20.23	23.66	
min	2011-01-01 00:00:00	NaN	NaN	NaN	NaN	0.82	0.76	
25%	2011-07-02 07:15:00	NaN	NaN	NaN	NaN	13.94	16.66	
50%	2012-01-01 20:30:00	NaN	NaN	NaN	NaN	20.50	24.24	
75%	2012-07-01 12:45:00	NaN	NaN	NaN	NaN	26.24	31.06	

1. There are no null values present in the data.
2. The standard deviation for registered users is quite high which means it has more outliers.

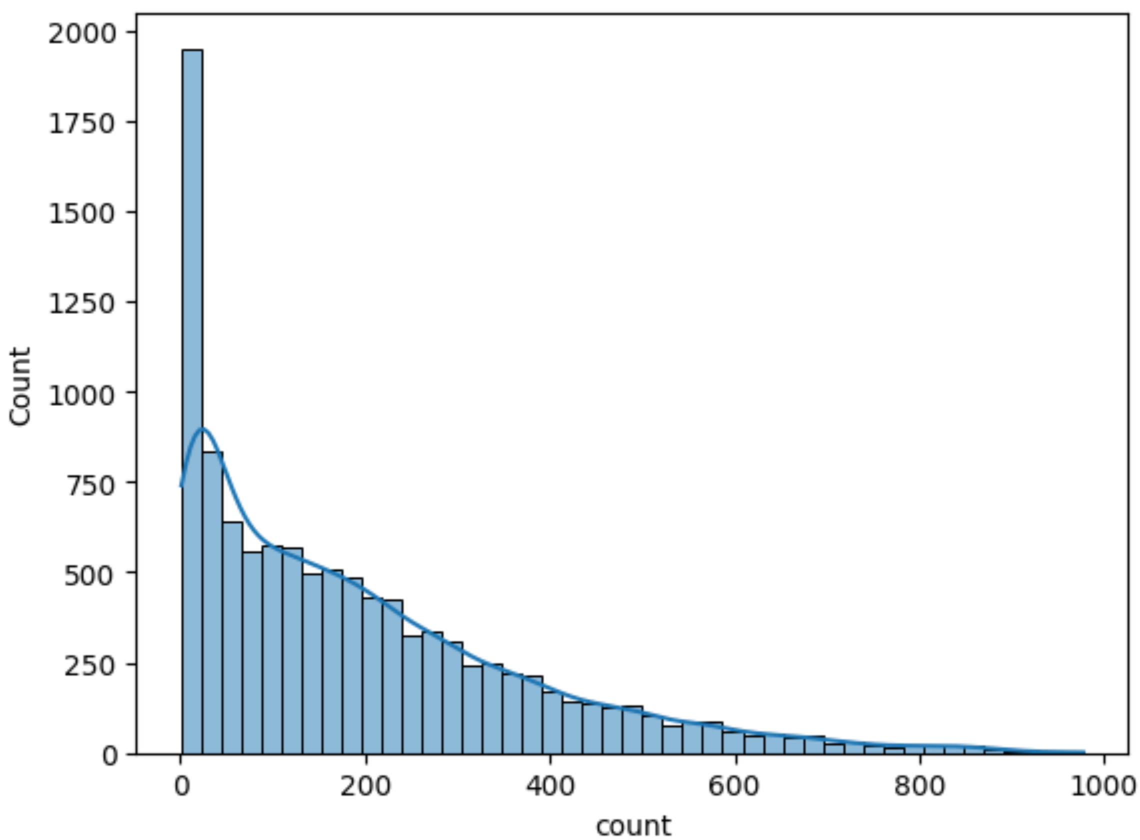
```
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(12, 8))
sns.histplot(df['temp'],kde=True,ax=axis[0][0])
sns.histplot(df['atemp'],kde=True,ax=axis[0][1])
sns.histplot(df['humidity'],kde=True,ax=axis[0][2])
sns.histplot(df['windspeed'],kde=True,ax=axis[1][0])
sns.histplot(df['casual'],kde=True,ax=axis[1][1])
sns.histplot(df['registered'],kde=True,ax=axis[1][2])
plt.show()
```



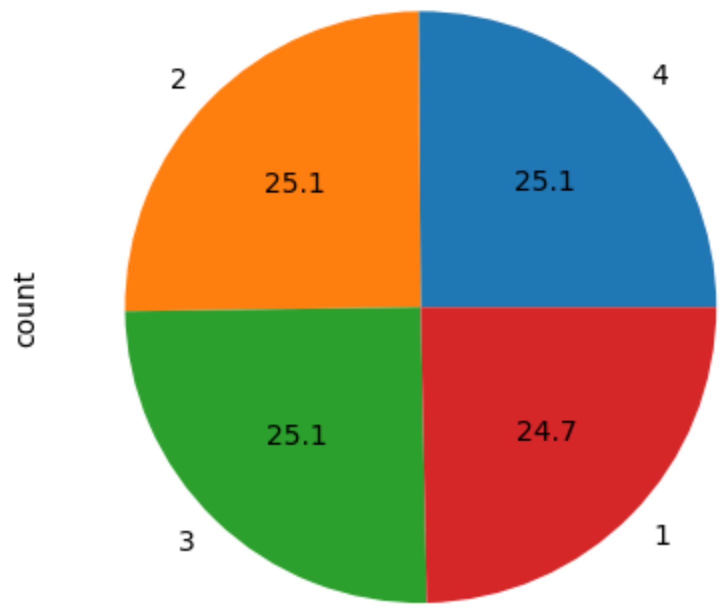



```
sns.histplot(df['count'],kde=True)
```

```
<Axes: xlabel='count', ylabel='Count'>
```

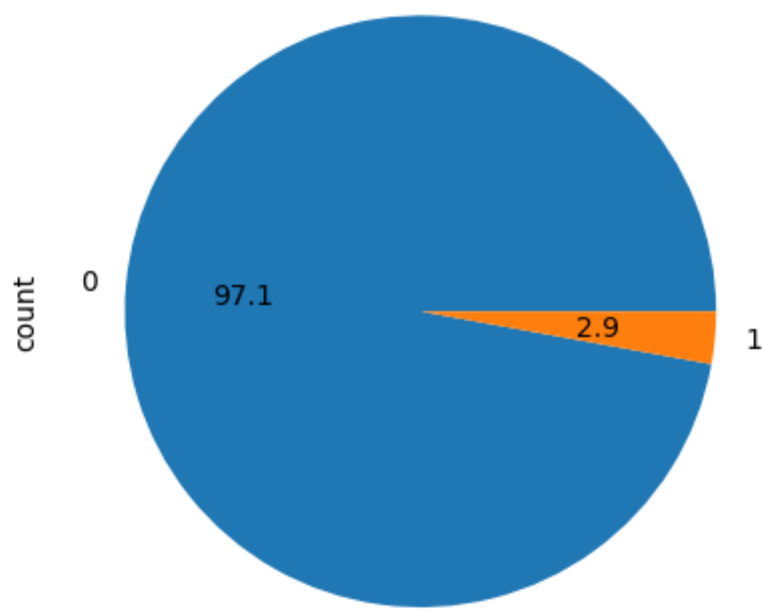



```
df['season'].value_counts().plot(kind='pie',autopct="%.1f")
plt.show()
```



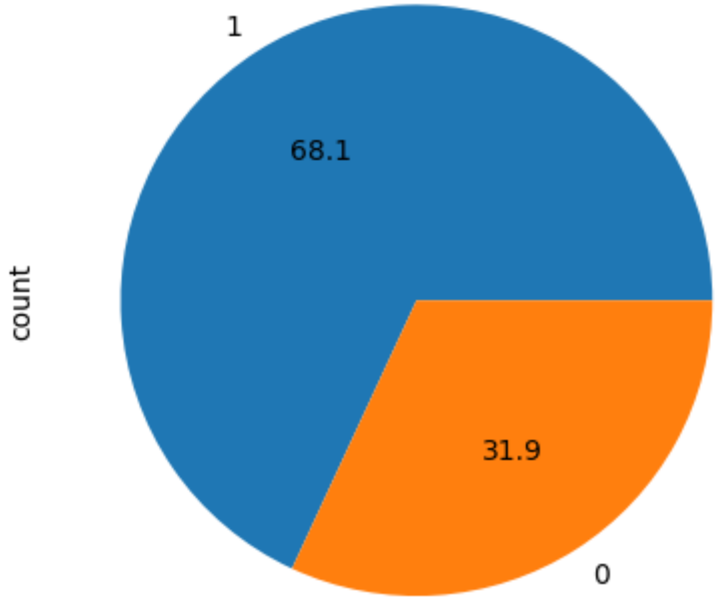

```
df['holiday'].value_counts().plot(kind='pie',autopct="%.1f")

plt.show()
```



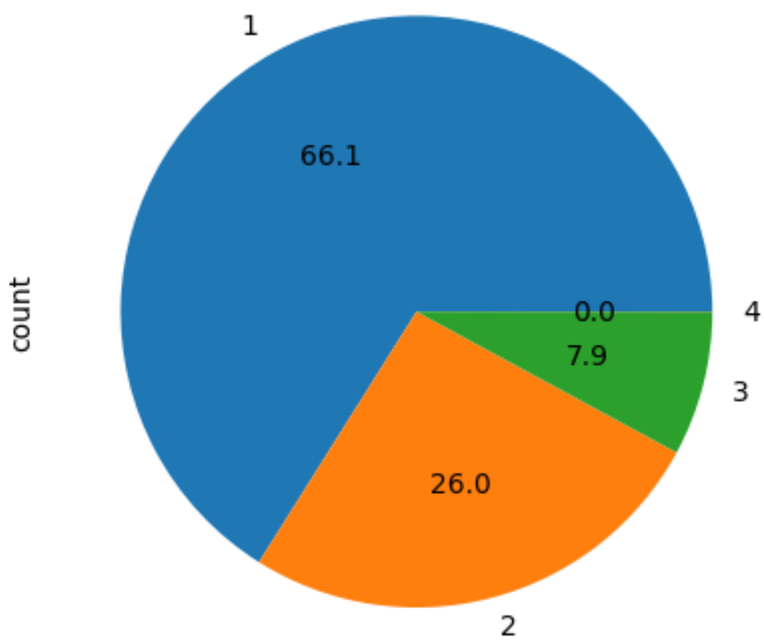

```
df['workingday'].value_counts().plot(kind='pie',autopct="%.1f")
```

```
<Axes: ylabel='count'>
```




```
df.groupby('weather')['weather'].value_counts().plot(kind='pie',autopct="%.1f")
```

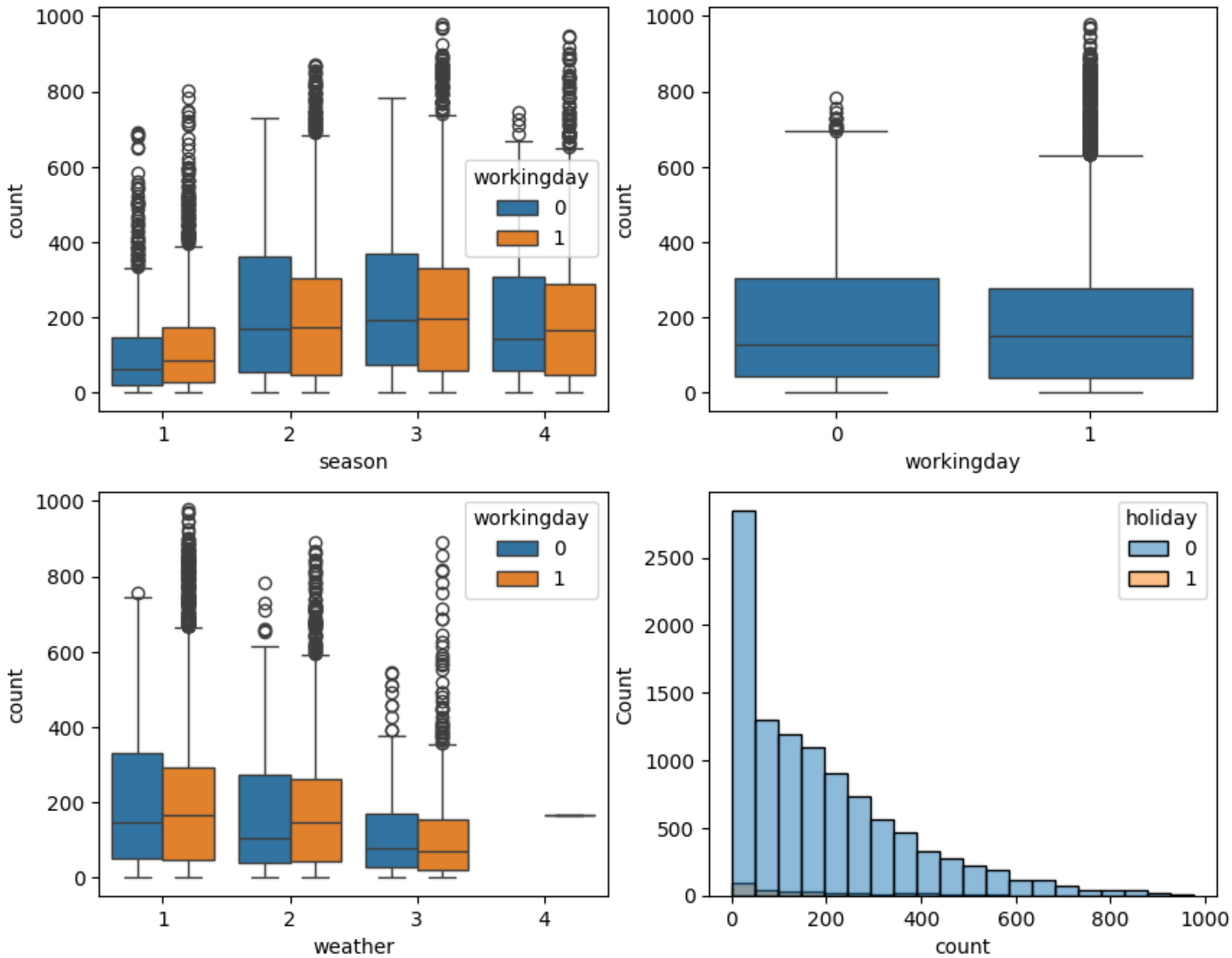
```
<Axes: ylabel='count'>
```




```
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(10, 8))
```

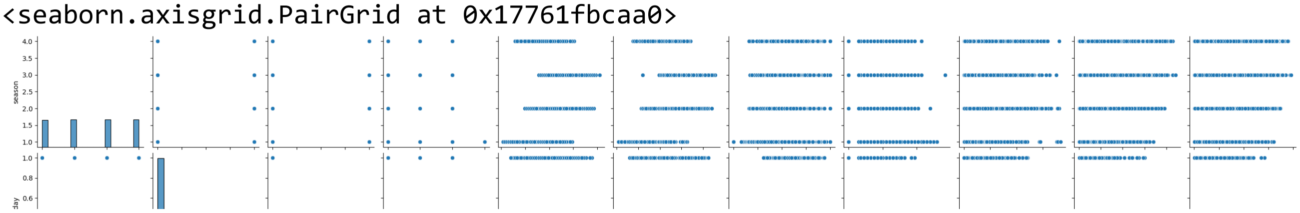
```
sns.boxplot(data = df,x= 'season',y='count',hue='workingday',ax=axis[0][0])
sns.boxplot(data = df,x= 'weather',y='count',hue='workingday',ax=axis[1][0])
sns.boxplot(data = df,x= 'workingday',y='count',ax=axis[0][1])
sns.histplot(data= df , x= 'count',bins = 20,hue = 'holiday', ax=axis[1][1])
```

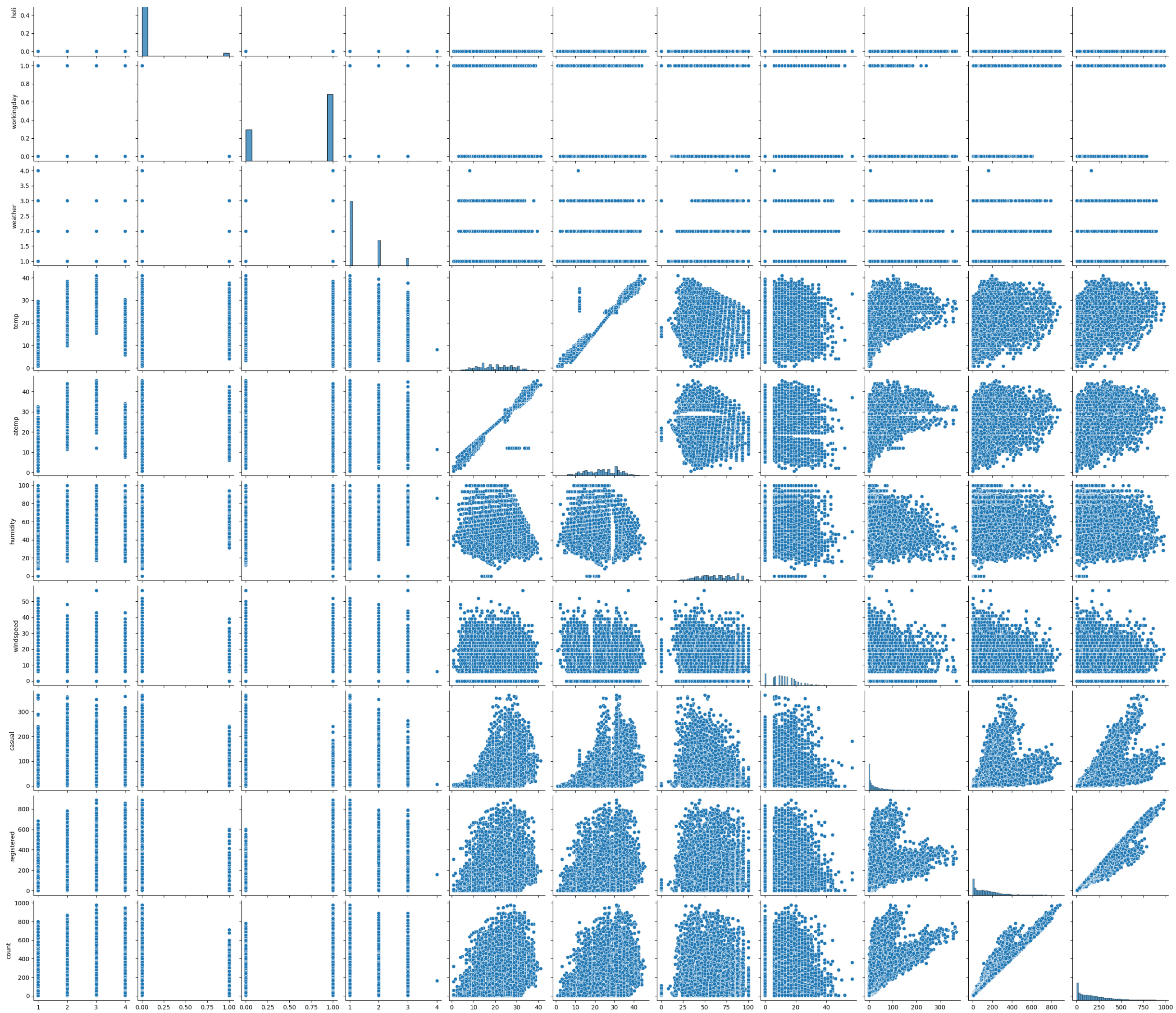
```
<Axes: xlabel='count', ylabel='Count'>
```




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

```
<seaborn.axisgrid.PairGrid at 0x17761fbcaa0>
```

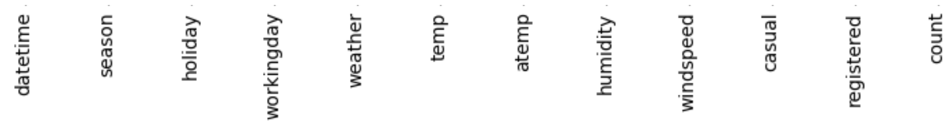



holi
workingday
weather
temp
atemp
humidity
windspeed
casual
registered
count
season
holiday
workingday
weather
temp
atemp
humidity
windspeed
casual
registered
count

```
pd.set_option('display.precision',2)
```

```
fig, axis = plt.subplots(nrows=1, ncols=1, figsize=(12, 8))
sns.heatmap(df.corr(),annot=True)
```

<Axes: >



1. There is a strong positive correlation between casual users and count of total rented bikes.
2. There is a strong positive correlation between registered users and count of total rented bikes.
3. There is a strong positive correlation between actual temperature and temperature felt.
4. The total number of bikes rented are highest in clear weather and lowest in rainy weather.
5. The total no. of bike rental is highest in summer and fall season.

Hypothesis Testing

Null hypothesis - Weather and season are independent

---

alternative hypothesis - Weather and season are dependent Significance level - 5%(0.05)

CHI Square Test

Significance level = 0.05 (5%) Assumptions: The observations are independently and randomly sampled from the population of all possible observations. The expected frequency for each cell is nonzero.

```
a=pd.crosstab(df['season'],df['weather'])
```

```
a
```

weather	1	2	3	4
season				

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

```
stats.chi2_contingency(a)
```

```
Chi2ContingencyResult(statistic=49.158655596893624, pvalue=1.549925073686492e-07,
dof=9, expected_freq=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]]))
```

```
Since p value is much less than 0.05 we will reject the null hypothesis and conclude that w
```

Null hypothesis - holiday and season are independent alternative hypothesis - holiday and season are dependent Significance level - 5%(0.05)

```
b=pd.crosstab(df['season'],df['holiday'])
```

```
b
```

holiday	0	1
season		
1	2615	71
2	2685	48
3	2637	96
4	2638	96

```
stats.chi2_contingency(b)
```

```
Chi2ContingencyResult(statistic=20.82338817816167, pvalue=0.00011455163312609901,
dof=3, expected_freq=array([[2609.26419254,   76.73580746],
       [2654.92145875,   78.07854125],
       [2654.92145875,   78.07854125],
       [2655.89288995,   78.10711005]]))
```

Since p value is much less than 0.05 we will reject the null hypothesis and conclude that holiday

Since p value is much less than 0.05 we will reject the null hypothesis and conclude that holiday and season are dependent on each other.

Null hypothesis - holiday and weather are independent alternative hypothesis - holiday and weather are dependent Significance level - 5%(0.05)

```
c=pd.crosstab(df['weather'],df['holiday'])
```

```
stats.chi2_contingency(c)
```

```
Chi2ContingencyResult(statistic=5.406882723976633, pvalue=0.1443153629276037, dof=3,
expected_freq=array([[6.98653316e+03, 2.05466838e+02],
       [2.75303601e+03, 8.09639904e+01],
       [8.34459397e+02, 2.45406026e+01],
       [9.71431196e-01, 2.85688040e-02]]))
```

Unsupported Cell Type. Double-Click to inspect/edit the content.

```
weekday = df[df['workingday'] == 1]['count']
weekend = df[df['workingday'] == 0]['count']
```

```
weekday
```

```
47        5
48        2
49        1
50        3
51       30
       ...
10881   336
10882   241
10883   168
10884   129
10885    88
Name: count, Length: 7412, dtype: int64
```

```
ttest_ind(weekday,weekend)
```

```
TtestResult(statistic=1.2096277376026694, pvalue=0.22644804226361348, df=10884.0)
```

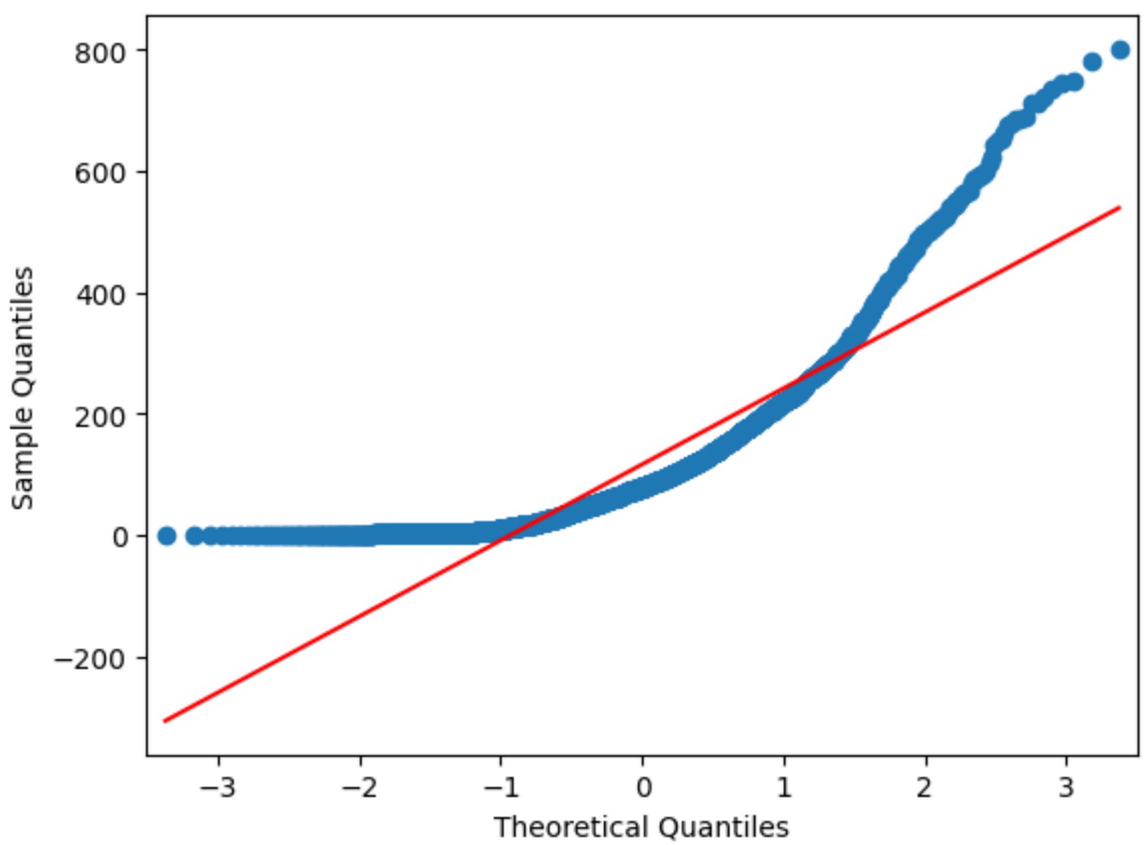
Since the p value is more than 0.05 then we fail to reject the null hypothesis and conclude that no of vehicles rented are independent of the fact if it is working day or non working day.

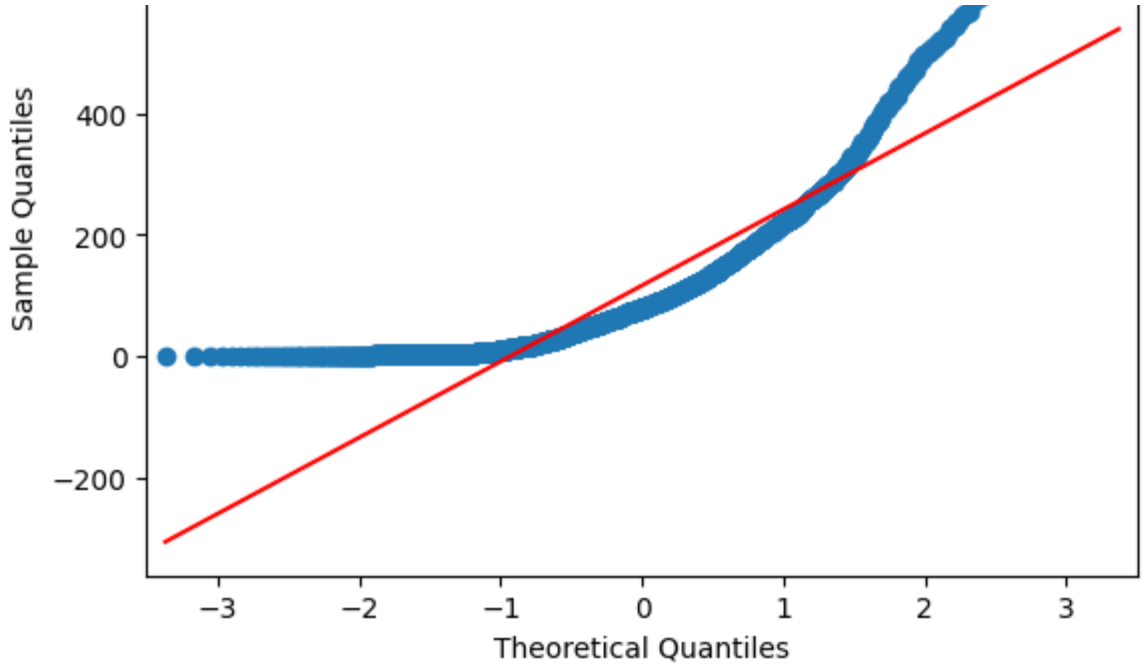
QQ Plot to check if we can use one way test or do we have to use kruskal Assumptions: Samples are random samples, or allocation is random. The two samples are mutually independent. The are random samples, or allocation is random. The two samples are mutually independent. The measurement scale is at least ordinal, and the variable is contino. Null hypothesis (H0) - The no. of bikes rented are independent of season . Alternate hypothesis (Ha) - The no. of bikes rented are dependent on season . Significance level - 5% (0.05)us

```
a =df[df['season']==1]['count']
```

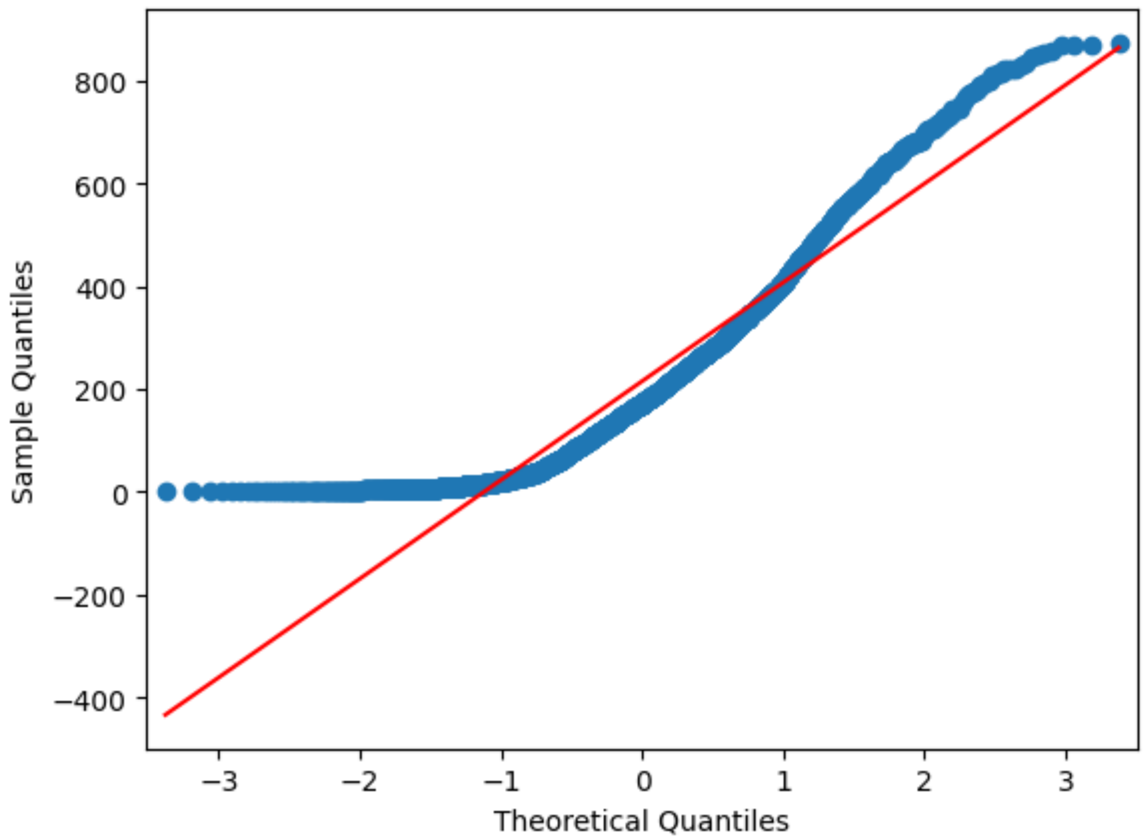
```
0        16
1        40
2        32
3        13
4         1
       ...
6780    549
6781    330
6782    223
6783    148
6784     54
Name: count, Length: 2686, dtype: int64
```

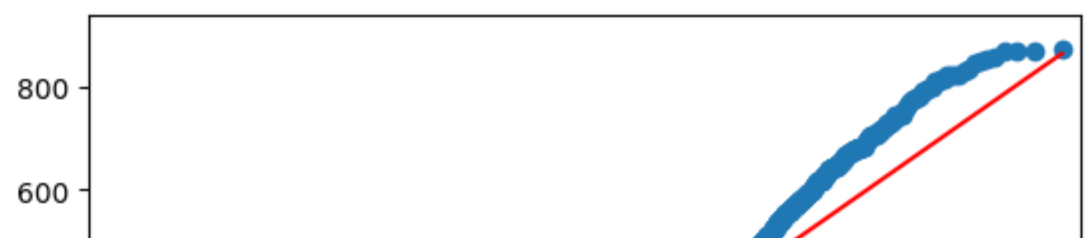
```
qqplot(df[df['season']==1]['count'],line='s')
```

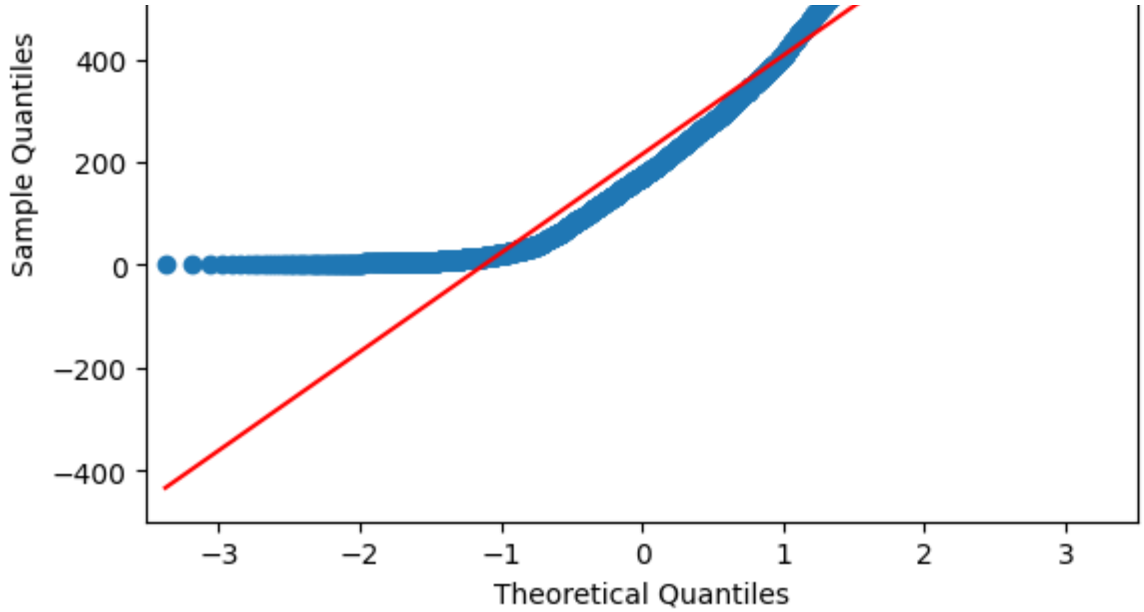





```
qqplot(df[df['season']==2]['count'],line='s')
```








```
kruskal(df['count'][df['season']==1],
                          df['count'][df['season']==2],
                          df['count'][df['season']==3],
                          df['count'][df['season']==4])
```

```
KruskalResult(statistic=699.6668548181988, pvalue=2.479008372608633e-151)
```

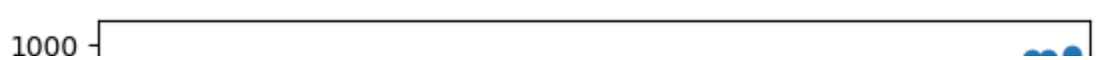
p value is much less than 0.05 which says that they are no. of bikes rented is heavily dependent on the season.

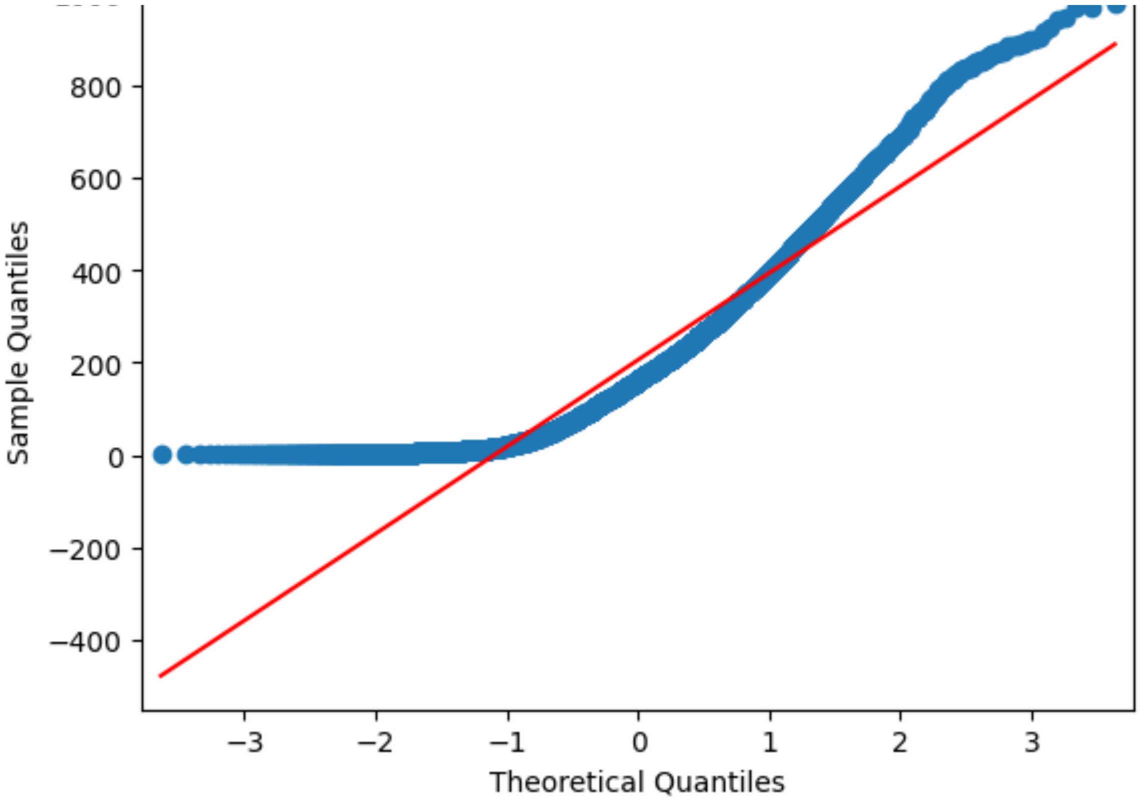
```
a =df[df['season']==1]['count']
```

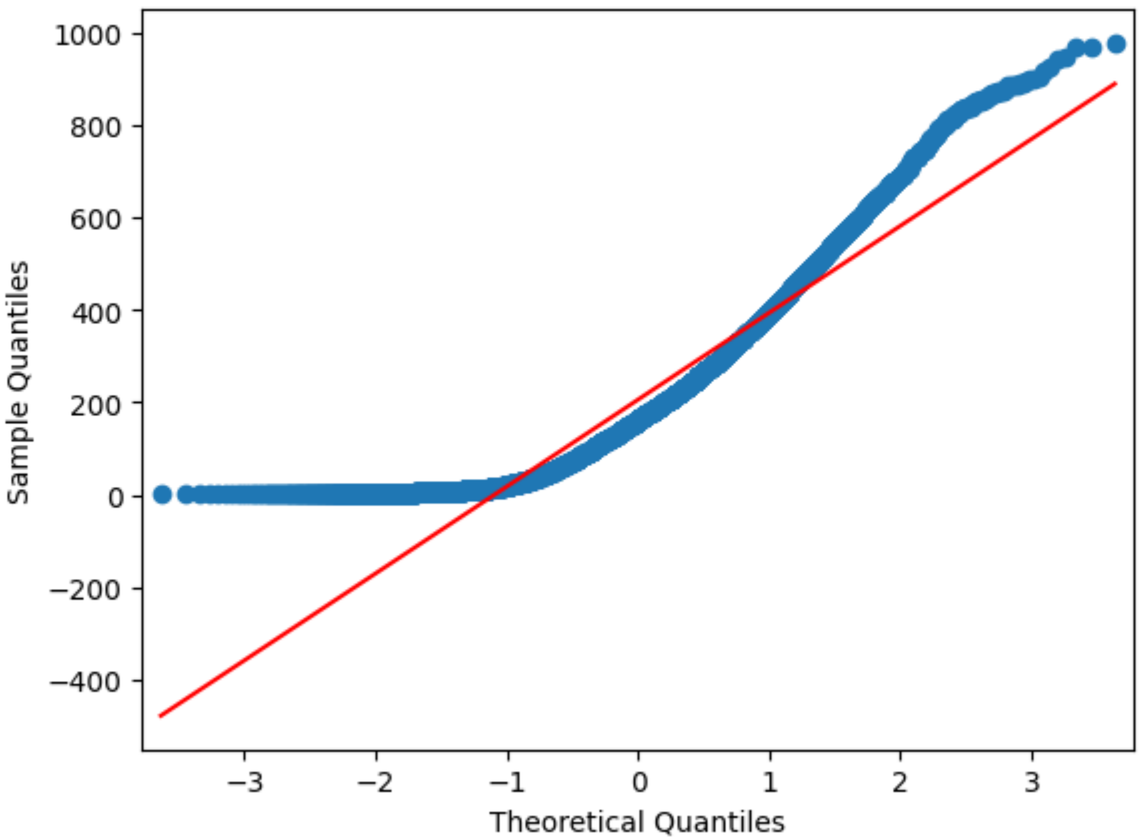
```
a
```

```
0        16
1        40
2        32
3        13
4         1
       ...
6780    549
6781    330
6782    223
6783    148
6784     54
Name: count, Length: 2686, dtype: int64
```

```
qqplot(df[df['weather']==1]['count'],line ='s')
```







Null hypothesis (H0) - The no. of bikes rented are independent of weather . Alternate hypothesis (Ha) - The no. of bikes rented are dependent on weather . Significance level - 5% (0.05)

```
clean = df[df['weather']==1]['count']
```

```
clear = df[df['weather']==1]['count']
Mist = df[df['weather']==2]['count']
Light_rain = df[df['weather']==3]['count']
Heavy_rain = df[df['weather']==4]['count']
```

```
kruskal(df[df['weather']==1]['count'],df[df['weather']==2]['count'],df[df['weather']==3]['c
```

```
KruskalResult(statistic=205.00216514479087, pvalue=3.501611300708679e-44)
```

p value is much less than 0.05 which says that they are no. of bikes rented is heavily dependent on the weather.

Insights and Recommendations

Customers rent electric bikes during clear or cloudy weather, majorly during January to August. This can help business to provide and make available of bikes during these seasons.

It is observed that whenever there is heavy rain, thunderstorm, snow or fog, less number bikes were rented .

whenever the temperature is less , number of bikes rented is less and when the windspeed is high, number of bikes rented is also low.

Registered users prefer renting bike on working days, mostly during office start and end hours, while casual riders rent on holidays.

Registered riders are significantly higher than casual riders.

we can say that weather depend on the season. Moreover, weather and season impact the numbers of bikes rented.

In summer and fall seasons, during clear or cloudy weather, the company should have more bikes in stock to be rented, as the demand during these seasons is higher as compared to other seasons.

With a significance level of 0.05, working day does have effect on the number of bikes rented and the bikes must be easily available at office hours i.e. morning and evening , infact bike stands should be installed near the corporate offices.

We neet to find the kind of bikes rented on holidays and the maintenance at these times should be high and a higher no. of bikes available during holidays will cater the needs of casual riders.