YULU – HYPOTHESIS TESTING

BUSINESS CASE STUDY

ABOUT YULU: -

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.

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

1) Importing The Data Set & Doing usual Exploratory Data Analysis Steps like Checking the Structure & Characteristics of the Data Set: ----

```
[1]  import numpy as np
     import pandas as pd
     import seaborn as sns
     import matplotlib.pyplot as plt
     from scipy import stats

     csv_path = "/content/bike_sharing.csv"
     df = pd.read_csv(csv_path, delimiter = ",")
     df.head()
```

	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

2) Number of Columns & Rows in the Data Set: --

```
print(f"# Rows:- {df.shape[0]} \n# Columns:-{df.shape[1]}")
```

```
# Rows:- 10886
# Columns:-12
```

```
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
```

Data Type of Following Attributes Need to be changed to the Proper Data Type: -

- Datetime -> datetime
- Season -> categorical

- Holiday -> categorical
- Working day -> categorical
- Weather -> categorical

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

cat_cols = ["season","holiday","workingday","weather"]
for col in cat_cols:
  df[col] = df[col].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.iloc[:,1:].describe(include = "all")
```

	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
count	10886.0	10886.0	10886.0	10886.0	10886.00000	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000
unique	4.0	2.0	2.0	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	4.0	0.0	1.0	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freq	2734.0	10575.0	7412.0	7192.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	20.23086	23.655084	61.886460	12.799395	36.021955	155.552177	191.574132
std	NaN	NaN	NaN	NaN	7.79159	8.474601	19.245033	8.164537	49.960477	151.039033	181.144454
min	NaN	NaN	NaN	NaN	0.82000	0.760000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	NaN	NaN	NaN	NaN	13.94000	16.665000	47.000000	7.001500	4.000000	36.000000	42.000000
50%	NaN	NaN	NaN	NaN	20.50000	24.240000	62.000000	12.998000	17.000000	118.000000	145.000000
75%	NaN	NaN	NaN	NaN	26.24000	31.060000	77.000000	16.997900	49.000000	222.000000	284.000000
max	NaN	NaN	NaN	NaN	41.00000	45.455000	100.000000	56.996900	367.000000	886.000000	977.000000

- There are no missing values in the data set.
- Casual & Registered attributes might have outliers because their mean & median are very far away to one another & the value of standard deviation is also high which tells us that there is high variance in the data of this attributes.

3) Detecting Missing Values in the Data Set: --

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

- So found that there are no missing values present in the data set.

4) Minimum & Maximum Date time & Number of Unique Values in each Categorical Columns: --

```
print(df['datetime'].min(), df['datetime'].max())
df[cat_cols].melt().groupby(['variable', 'value'])[['value']].count()
```

```
2011-01-01 00:00:00 2012-12-19 23:00:00
```

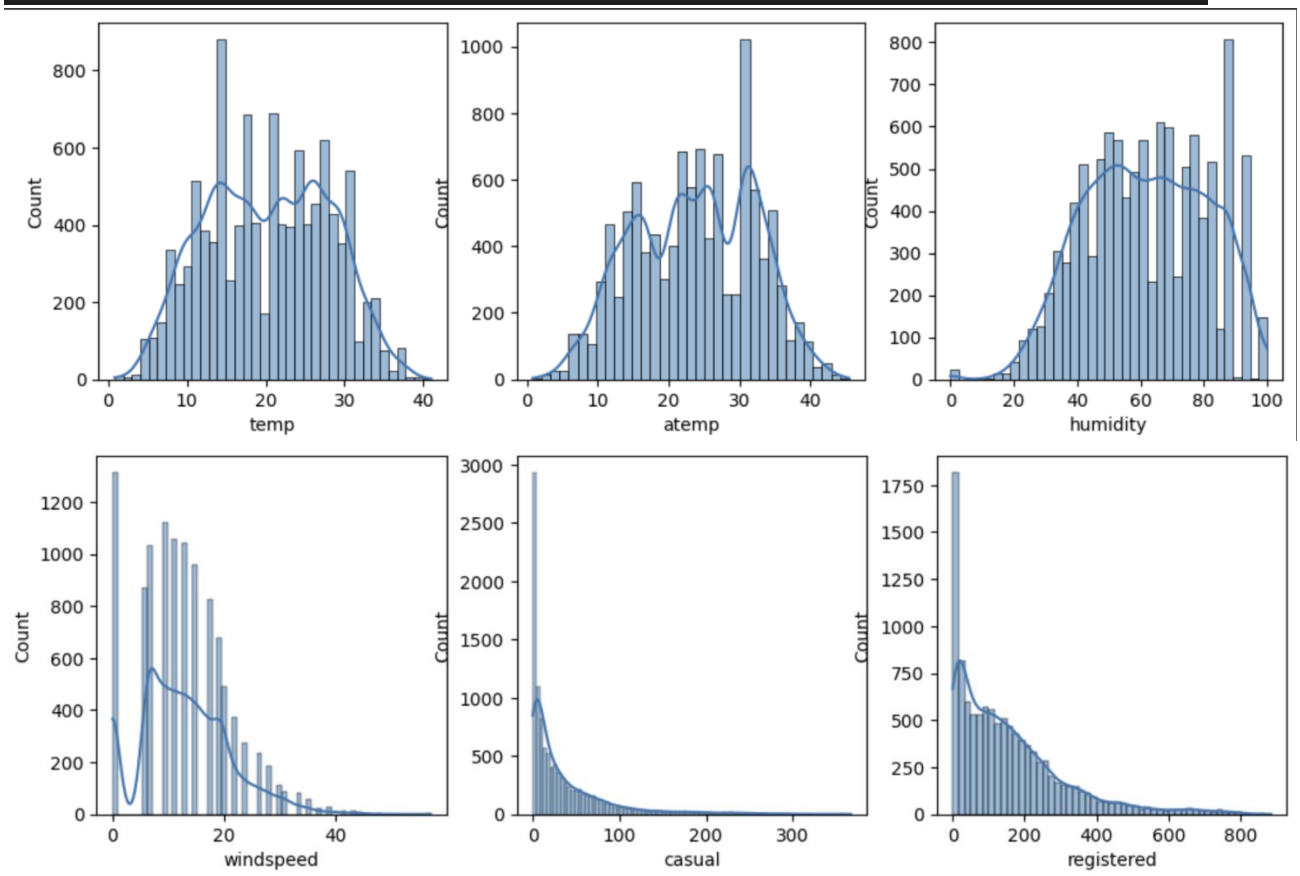
variable	value	value
holiday	0	10575
	1	311
season	1	2686
	2	2733
	3	2733
	4	2734
weather	1	7192
	2	2834
	3	859
	4	1
workingday	0	3474
	1	7412

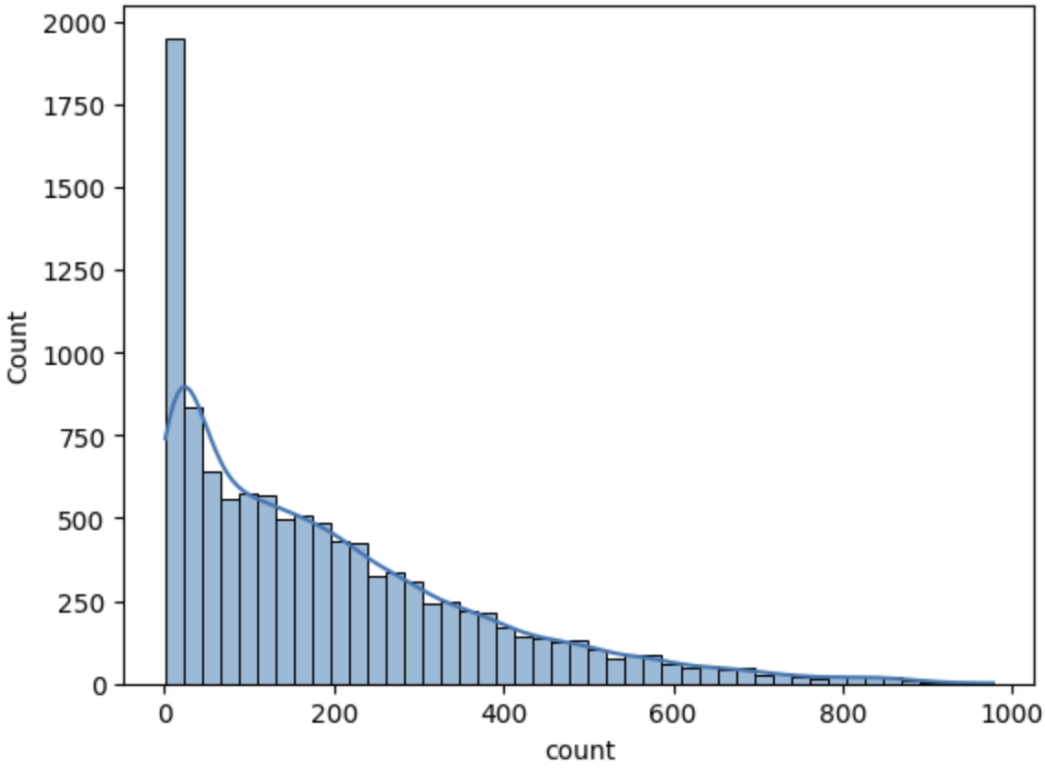
5) Trying to establishing a relation between the Dependent & Independent Variables (Dependent “Count” & Independent: Working day Weather Season etc.):--

- **Univariate Analysis: --- (Understanding the distribution for numerical variables)**

```
num_cols = ['temp', 'atemp', 'humidity', 'windspeed', 'casual',
'registered','count']
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(12, 8))
index = 0
for row in range(2):
  for col in range(3):
    sns.histplot(df[num_cols[index]], ax=axis[row, col], kde=True)
    index += 1


plt.show()
sns.histplot(df[num_cols[-1]], kde=True)
plt.show()
```

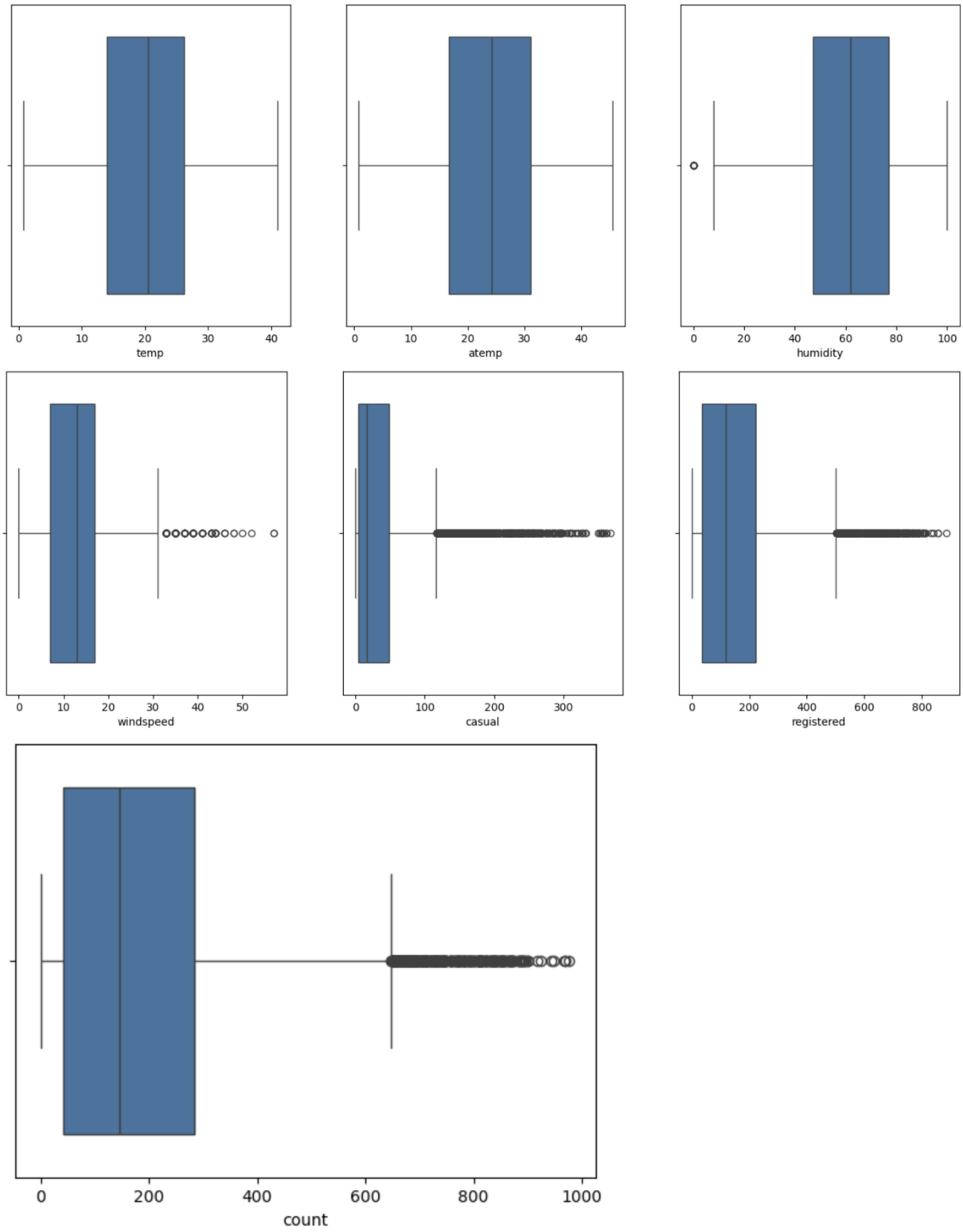




- Casual, registered & count somewhat looks like Log Normal Distribution.
- Temp, atemp & humidity looks like they follow the Normal Distribution.
- Windspeed follows the binomial distribution.

Plotting Box Plots to detect The Outliers in the Data: --

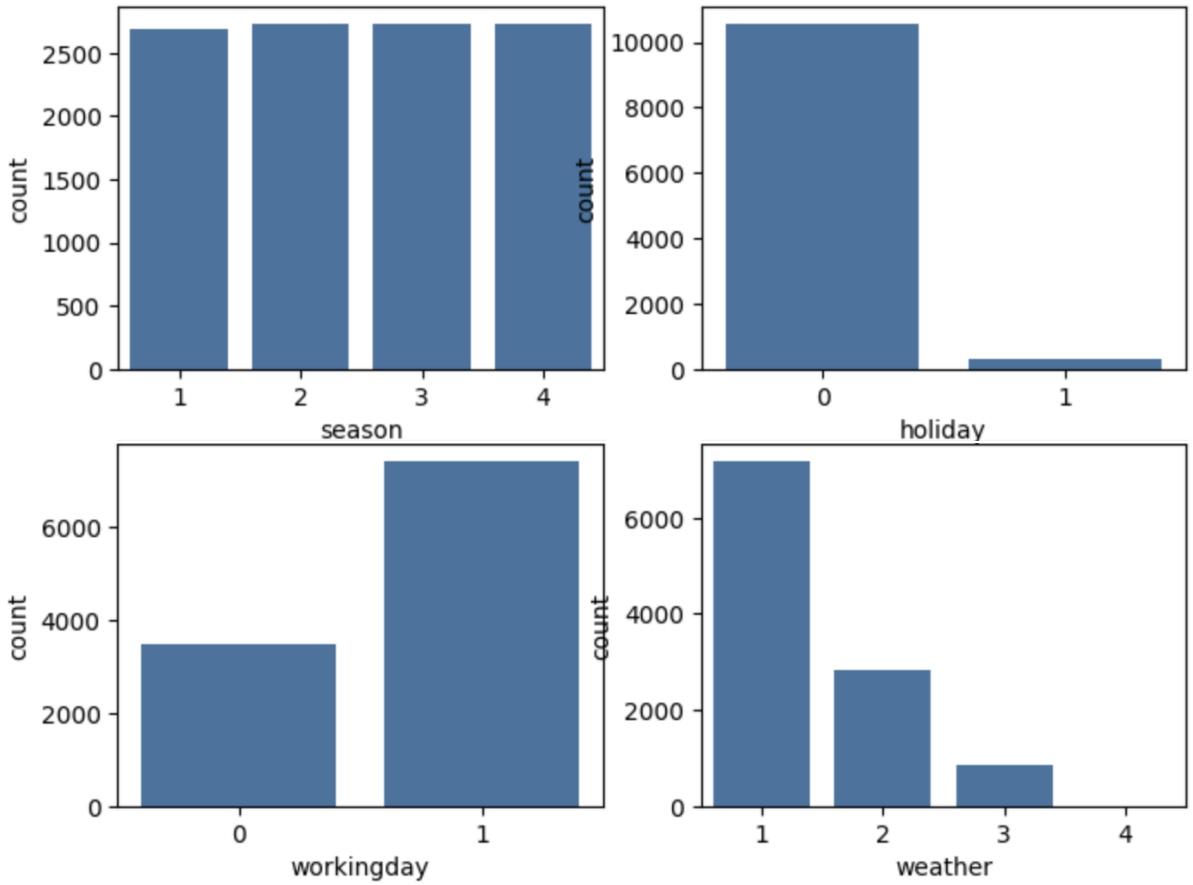
```
[12] fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(16, 12))
     index = 0
     for row in range(2):
       for col in range(3):
         sns.boxplot(x=df[num_cols[index]], ax=axis[row, col])
         index += 1
     plt.show()
     sns.boxplot(x=df[num_cols[-1]])
     plt.show()
```



So, it looks like that humidity, Casual, registered & count have Outliers in the data.

- **Count Plot of Each Categorical Column: --**

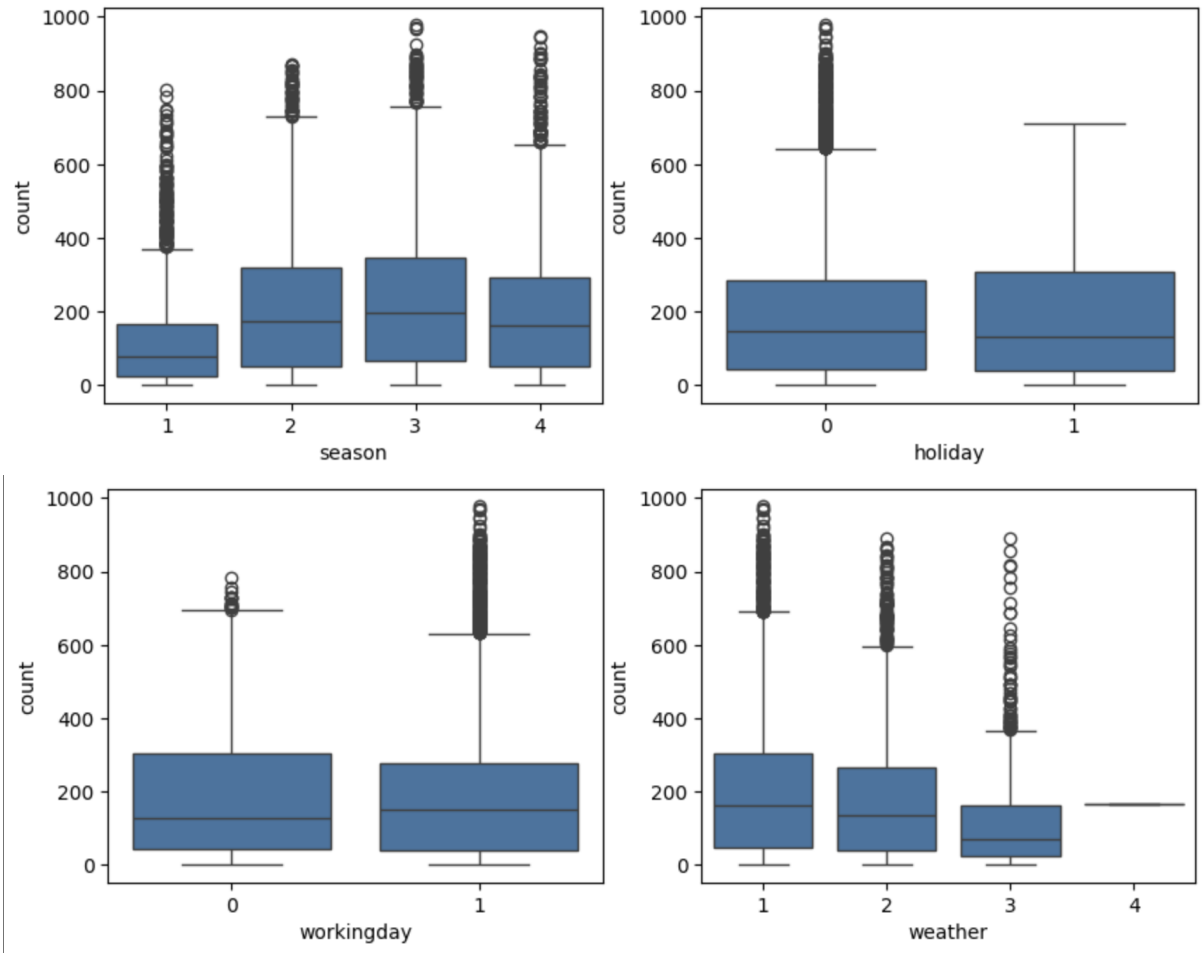
```
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(8, 6))
index = 0
for row in range(2):
  for col in range(2):
    sns.countplot(data=df, x=cat_cols[index], ax=axis[row, col])
    index += 1
plt.show()
```



So, the data looks common as it should be like equal number of days in each season, more working & weather is mostly clear, few clouds, partly clouds, partly cloudy.

- **<u>Bi-Variate Analysis</u>: -**
    - **Plotting Categorical Variables against Count using boxplot.**

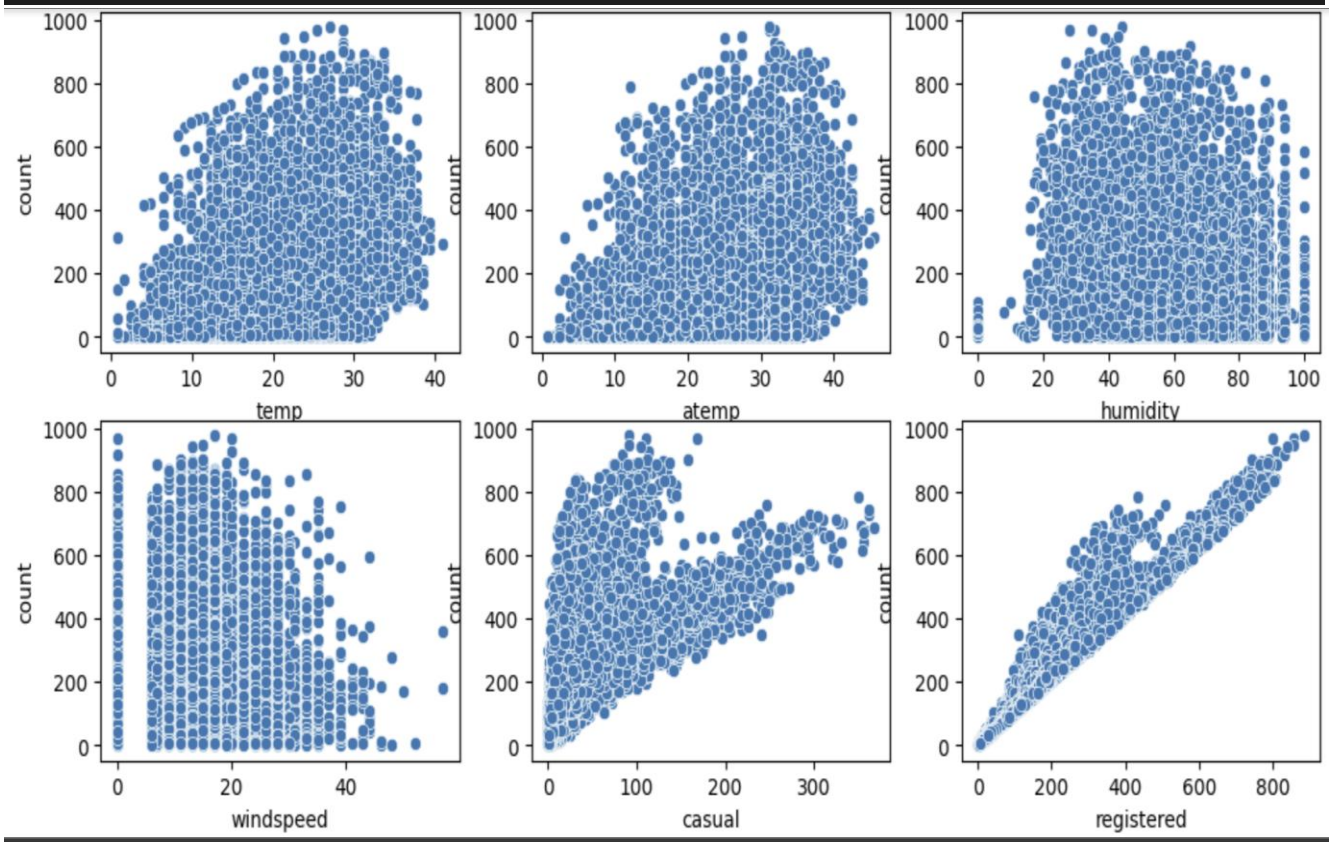
```
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(10, 8))
index = 0
for row in range(2):
  for col in range(2):
    sns.boxplot(data=df, x=cat_cols[index], y='count', ax=axis[row,col])
    index += 1
plt.show()
```



- In Summer & fall seasons more bikes are rented as compared to other seasons. Whenever it's a holiday more bikes are rented.
- It is also clear from the working day also that whenever day is holiday or weekend, slightly more bikes were rented.
- Whenever there is rain, thunderstorm, snow or fog, there were less bikes were rented.

- **Plotting Numerical Variables against count using Scatterplot: -**

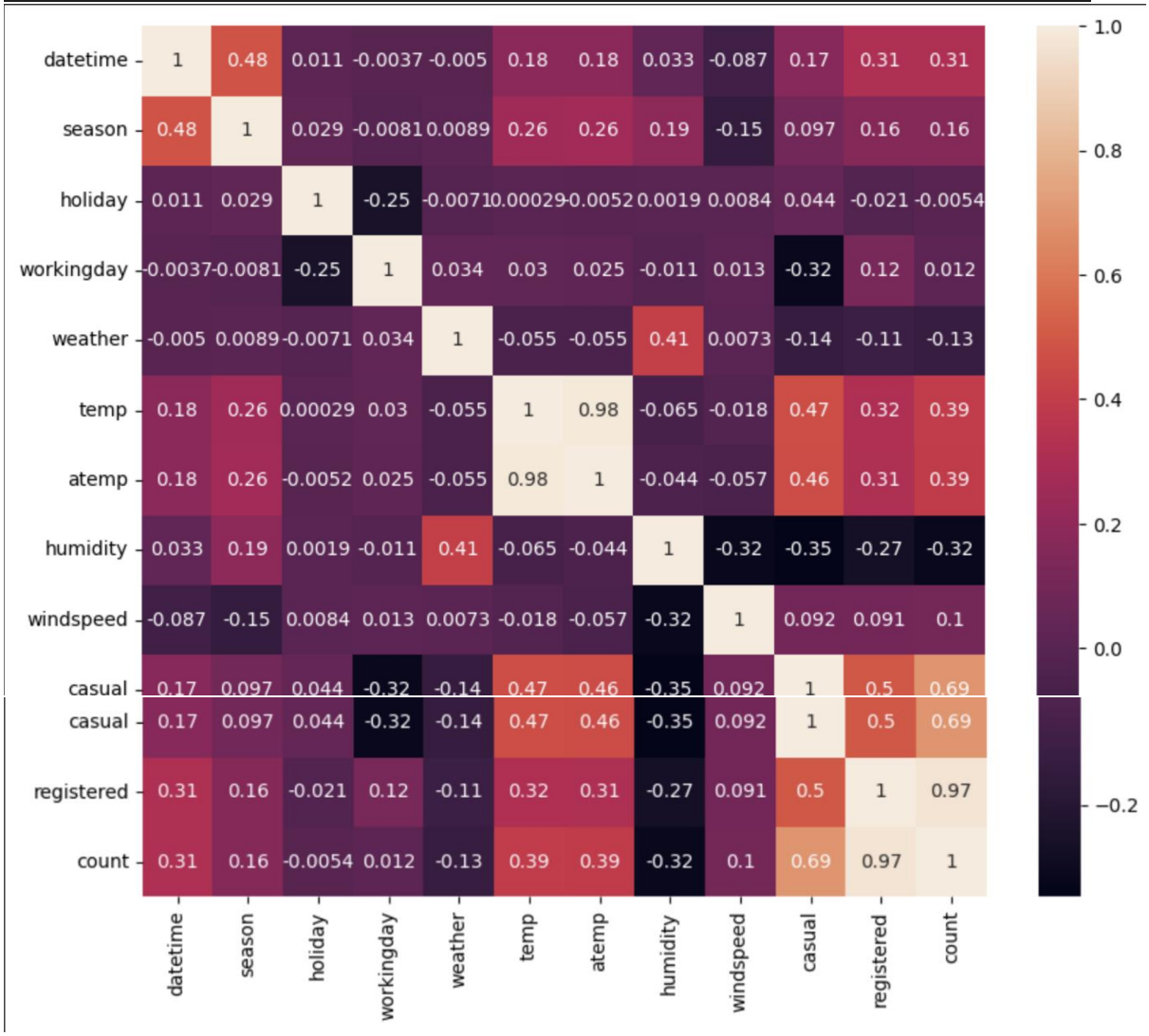
```
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(12, 6))
index = 0
for row in range(2):
  for col in range(3):
    sns.scatterplot(data=df, x=num_cols[index], y='count',ax=axis[row, col])
    index += 1
plt.show()
```



- Whenever the humidity is less than 20, number of bikes rented is very low.
- Whenever the temperature is less than 10, number of bikes rented is less.
- Whenever the windspeed is greater than 35, number of bikes rented is less.

- **Understanding the Correlation between Count & Numerical Variables: -**

```
plt.figure(figsize=(10, 8))
df.corr()['count']
sns.heatmap(df.corr(), annot=True)
plt.show()
```



6) Hypothesis Testing: --

- **Chi-Square test to check if Weather is dependent on the season: -**

Null Hypothesis (HO) :- Weather is independent of the season.

Alternate Hypothesis (H1) :- Weather is not independent of the season.

Significance Level (alpha) :- 0.05

```
data_table = pd.crosstab(df['season'], df['weather'])
print("Observed values:")
data_table
```

```
Observed values:
```

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

```
val = stats.chi2_contingency(data_table)
print(val)
expected_values = val[3]
print(expected_values)
nrows, ncols = 4, 4
dof = (nrows-1)*(ncols-1)
print("degrees of freedom: ", dof)
alpha = 0.05
chi_sqr = sum([(o-e)**2/e for o, e in zip(data_table.values, expected_values)])
chi_sqr_statistic = chi_sqr[0] + chi_sqr[1]
print("chi-square test statistic: ", chi_sqr_statistic)
critical_val = stats.chi2.ppf(q=1-alpha, df=dof)
print(f"critical value: {critical_val}")
p_val = 1-stats.chi2.cdf(x=chi_sqr_statistic, df=dof)
print(f"p-value: {p_val}")
if p_val <= alpha:
  print("\nSince p-value is less than the alpha 0.05, We reject the Null Hypothesis. Meaning that \ Weather is dependent on the season.")
else:
  print("Since p-value is greater than the alpha 0.05, We do not reject the Null Hypothesis")
```

```
Chi2ContingencyResult(statistic=49.158655596893624, pvalue=1.549925073686492e-07, dof=9, expected_freq=array([[1.77454639e+03, 6.99258130e+02, 2.11948742e+02, 2.46738931e-
       [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]]))
[[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]]
degrees of freedom:  9
chi-square test statistic:  44.09441248632364
critical value: 16.918977604620448
p-value: 1.3560001579371317e-06

Since p-value is less than the alpha 0.05, We reject the Null Hypothesis. Meaning that \ Weather is dependent on the season.
```

Sample T Test – To Check if Working Day has an effect on the number of electric cycles rented: -

Null Hypothesis (HO) :- Working day has no effect on the number of cycles being rented.

Significance level (alpha) :- 0.05

So, we will use the 2-**Sample T-Test** to test the hypothesis defined above

```
data_group1 = df[df['workingday']==0]['count'].values
data_group2 = df[df['workingday']==1]['count'].values
print(np.var(data_group1), np.var(data_group2))
np.var(data_group2)// np.var(data_group1)
```

So before conducting the sample T – Test we need to find if the given data groups have the same variance. If the ratio of the larger data groups

to the small data group is less than 4:1 then we can consider that the given data groups have equal variance.

```
30171.346098942427 34040.69710674686
1.0
```

Here the ratio is 34040.70 / 30171.35 Which Is less than 4:1

```
stats.ttest_ind(a=data_group1, b=data_group2, equal_var=True)

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

Since the pvalue is greater than 0.05, So we cannot reject the Null Hypothesis. We don't have the sufficient evidence to say that working day has effect on the number of cycles being rented.

ANNOVA to check if No. of cycles rented is similar or different in different 1. Weather 2. Season: -

Null Hypothesis (HO) :- Number of cycles rented is similar in different weather & season.

Alternate Hypothesis (Ha) :- Number of cycles rented is not similar in different Weather & Season.

Significance Level (alpha) :- 0.05

- **Defining The Data Groups for the ANOVA: -**

```
from statsmodels.graphics.gofplots import qqplot
gp1 = df[df['weather']==1]['count'].values
gp2 = df[df['weather']==2]['count'].values
gp3 = df[df['weather']==3]['count'].values
gp4 = df[df['weather']==4]['count'].values

gp5 = df[df['season']==1]['count'].values
gp6 = df[df['season']==2]['count'].values
gp7 = df[df['season']==3]['count'].values
gp8 = df[df['season']==4]['count'].values
groups=[gp1,gp2,gp3,gp4,gp5,gp6,gp7,gp8]
```

```
fig, axis = plt.subplots(nrows=4, ncols=2, figsize=(8, 8))
index = 0
for row in range(4):
  for col in range(2):
    sns.histplot(groups[index], ax=axis[row, col], kde=True)
    index += 1
plt.show()
```

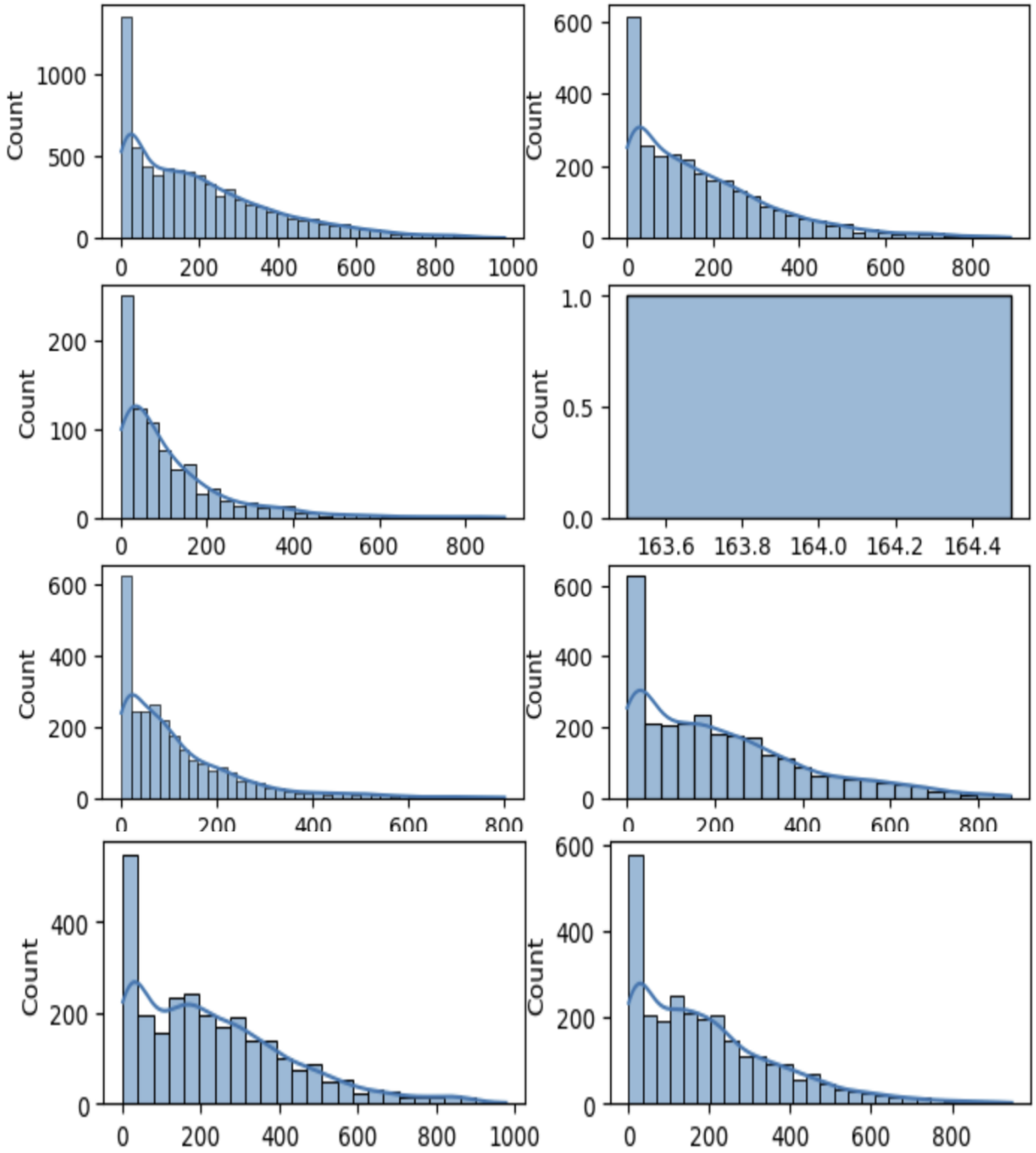
```
index = 0
for row in range(4):
  for col in range(2):
    qqplot(groups[index], line="s")
    index += 1
plt.show()
```

Assumption For ANOVA----------

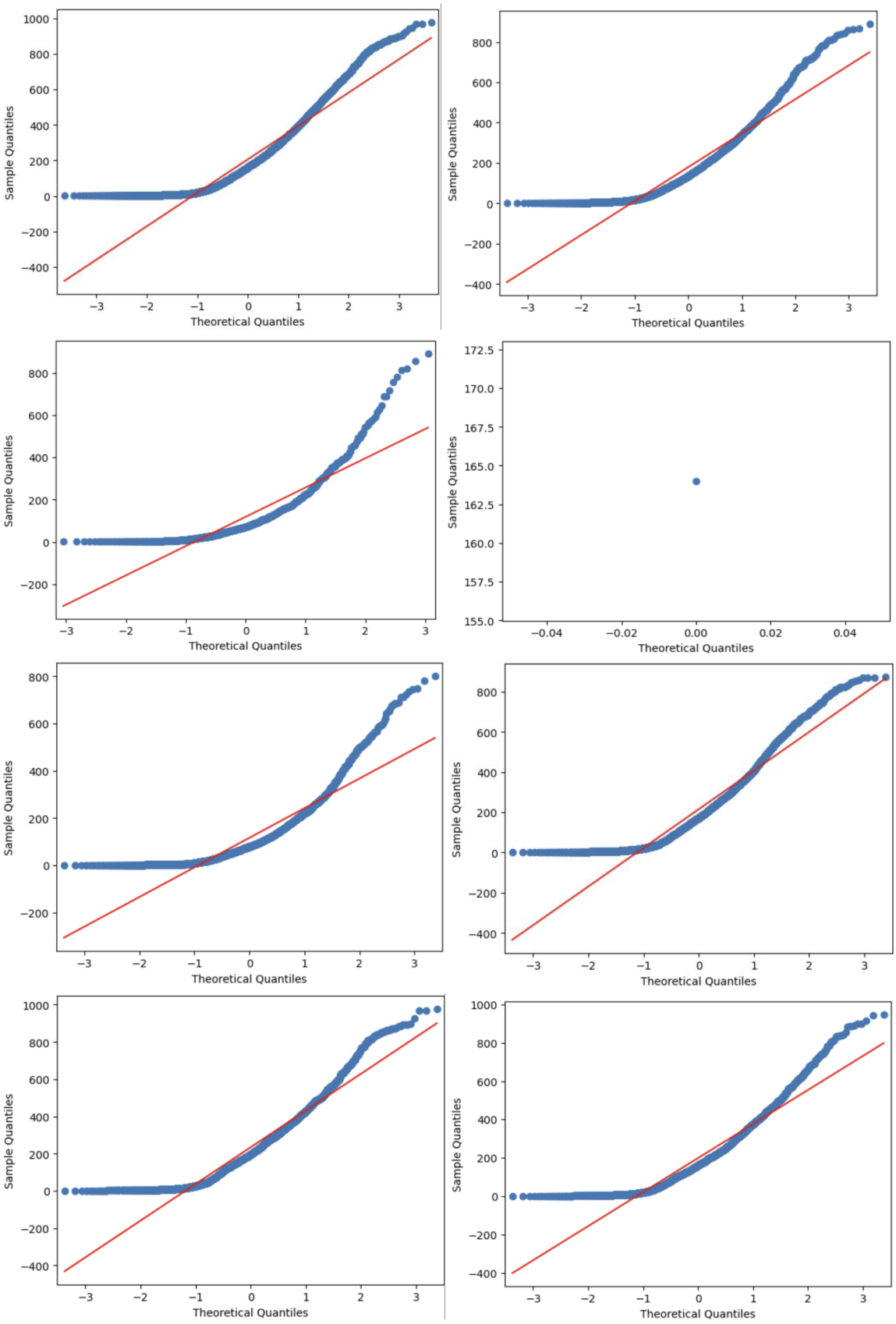
----

1) **Gaussian**

Histogram


Count
1000
500
0
0 200 400 600 800 1000
Count
600
400
200
0
0 200 400 600 800
Count
200
100
0
0 200 400 600 800
Count
1.0
0.5
0.0
163.6 163.8 164.0 164.2 164.4
Count
600
400
200
0
0 200 400 600 800
Count
600
400
200
0
0 200 400 600 800
Count
400
200
0
0 200 400 600 800 1000
Count
600
400
200
0
0 200 400 600 800

QQ -Plot


Sample Quantiles
Theoretical Quantiles
Sample Quantiles
Theoretical Quantiles
Sample Quantiles
Theoretical Quantiles
Sample Quantiles
Theoretical Quantiles
Sample Quantiles
Theoretical Quantiles
Sample Quantiles
Theoretical Quantiles
Sample Quantiles
Theoretical Quantiles
Sample Quantiles
Theoretical Quantiles

As per the above graphs, all groups are not following Gaussian distribution.

2) **Data is Independent.**
3) **Equal variance: - LEVENE's Test**

Null Hypothesis (HO) :- Variance is similar in different weather & season.

Alternate Hypothesis (Ha) :- Variance is not similar in different weather & season.

Significance Level (alpha) :- 0.05

```
levene_stat, p_value = stats.levene(gp1,gp2,gp3,gp4,gp5,gp6,gp7,gp8)
print(p_value)
if p_value < 0.05:
  print("Reject the Null hypothesis.Variances are not equal")
else:
  print("Fail to Reject the Null hypothesis.Variances are equal")
```

```
3.4635318888975 94e-148
Reject the Null hypothesis.Variances are not equal
```

As per QQ-plot & Levene's Test, We cannot do ANOVA Test.

- **Assumptions of ANOVA fail, Use Kruskal.**
  So assumption of ANOVA don't hold, We need Kruskal Wallis

```
kruskal_stat, p_value = stats.kruskal(gp1,gp2,gp3,gp4,gp5,gp6,gp7,gp8)
print("p_value===",p_value)
if p_value<0.05:
  print("Since p-value is less than 0.05, we reject the null hypothesis")
```

```
p_value=== 4.614440933900297e-191
Since p-value is less than 0.05, we reject the null hypothesis
```

Since p-value is less than 0.05, We reject the Null Hypothesis. This implies that Number of cycles rented is not similar in different weather & season conditions.

Insights: -

- **In Summer with in full seasons more bikes are rented as compared to other seasons.**
- **Whenever it's a holiday than more bikes are rented.**
- **It is also clear from the working day also that whenever a day is holiday or a weekend than it has been observed that slightly more bikes were rented.**
- **Whenever there is rain & thunderstorm, snow or fog, it has been observed that there were less bikes were rented.**
- **Whenever the humidity is less than 20, number of bikes rented is very low.**
- **Whenever the temperature is less than 10, the number of bikes rented is less/**
- **Whenever the windspeed is greater than 35, number of bikes rented is less.**

Recommendations: -

- **In summer with in full season the company should increase the stock to cater to the increasing customers because the demand is seen higher as compared to the other seasons.**
- **With a Significance level of 0.05, Working day has no effect on the number of bikes being rented.**
- **In very low humid days the company may allow to have less stock for rent.**
- **Whenever temperature is less than 10 or in very cold days company may have less bikes.**
- **When ever the windspeed is greater than 35 or in thunderstorms, the company may have less bikes in stock for rent.**