#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.

#Business Problem

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

#Layout to solve the business case study:

To undersatnd the impact of various variables the following steps will be followed:

The code which would help us derive the result will be mentioned. Followed by the insights of the mentioned code. And at last all the major insights will be mentioned followed by recommendations ansd conclusion.

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

from scipy.stats import norm
from scipy.stats import ttest_1samp, ttest_ind,ttest_rel
from scipy.stats import chi2_contingency
from scipy.stats import chi2
from scipy.stats import geom, poisson, expon,binom
from scipy import stats
from scipy.stats import f_oneway, kruskal
from scipy.stats import poisson, binom
```

```
df = pd.read_csv('https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/428/original/bike_
        1642089089')
```

#EDA

```
df.head()
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casu
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0	3
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0	8
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0	5

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casu
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0	3
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0	0

```
df.tail()
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed
10881	2012-12-19 19:00:00	4	0	1	1	15.58	19.695	50	26.0027
10882	2012-12-19 20:00:00	4	0	1	1	14.76	17.425	57	15.0013
10883	2012-12-19 21:00:00	4	0	1	1	13.94	15.910	61	15.0013
10884	2012-12-19 22:00:00	4	0	1	1	13.94	17.425	61	6.0032
10885	2012-12-19 23:00:00	4	0	1	1	13.12	16.665	66	8.9981

```
df.shape
```

```
(10886, 12)
```

```
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
```

```
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.season.value_counts(ascending=True)
```

```
1    2686
2    2733
3    2733
4    2734
Name: season, dtype: int64
```

```
df.weather.value_counts()
```

```
1    7192
2    2834
3     859
4       1
Name: weather, dtype: int64
```

```
df.workingday.value_counts()
```

```
1    7412
0    3474
Name: workingday, dtype: int64
```

#Insight

we can see that bikes are highly used on working days and and there isnt much differnce between seasons and during heavy rains,foggy weather we see only one bike being used

#Statistical summary

```
df.describe()
```

	season	holiday	workingday	weather	temp	atemp	hur
count	10886.000000	10886.000000	10886.000000	10886.000000	10886.00000	10886.000000	10886.0
mean	2.506614	0.028569	0.680875	1.418427	20.23086	23.655084	61.8864
std	1.116174	0.166599	0.466159	0.633839	7.79159	8.474601	19.2450
min	1.000000	0.000000	0.000000	1.000000	0.82000	0.760000	0.00000
25%	2.000000	0.000000	0.000000	1.000000	13.94000	16.665000	47.0000
50%	3.000000	0.000000	1.000000	1.000000	20.50000	24.240000	62.0000
75%	4.000000	0.000000	1.000000	2.000000	26.24000	31.060000	77.0000
max	4.000000	1.000000	1.000000	4.000000	41.00000	45.455000	100.000

#Insights:

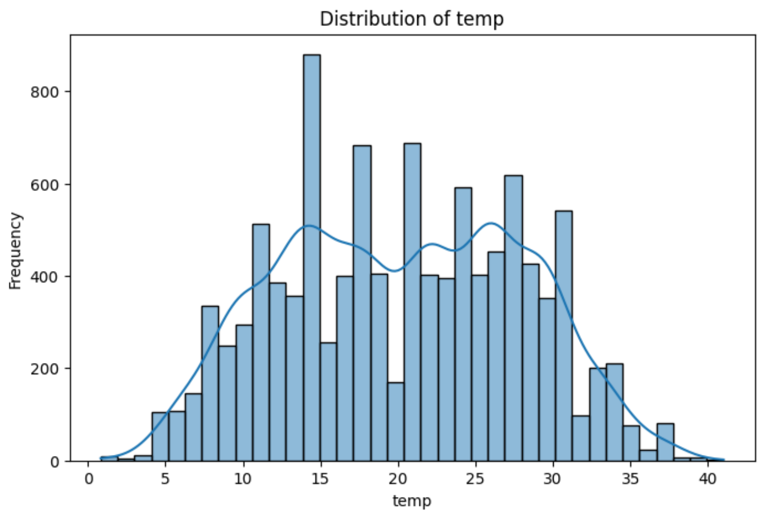
In the above data we can see that mean season value is around 2.51, which suggests an average season of between 2 and 3,which suggests that bikes are mostly used between summer and fall.Although there is not a higher differnce wrt to other seasons,we can say that around all seasons the demand doesnot vary much.

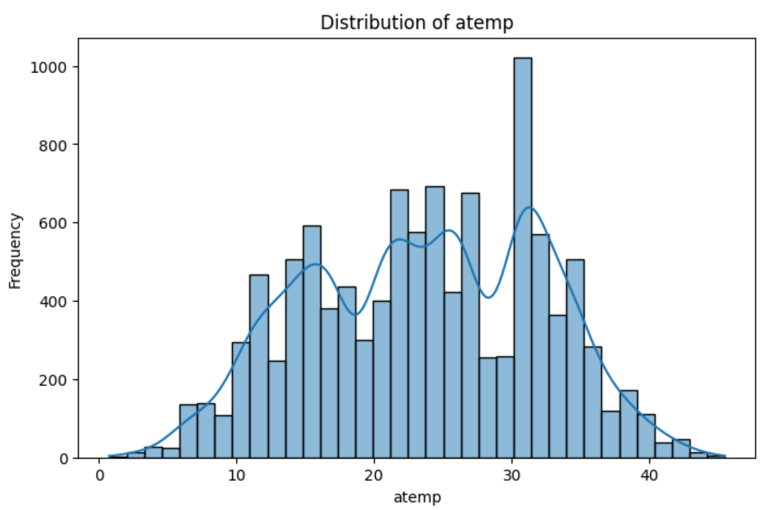
The mean weather value is around 1.42, indicating an average weather condition of 1 to 2,which means bike is mostly used on Clear and cloudy days.

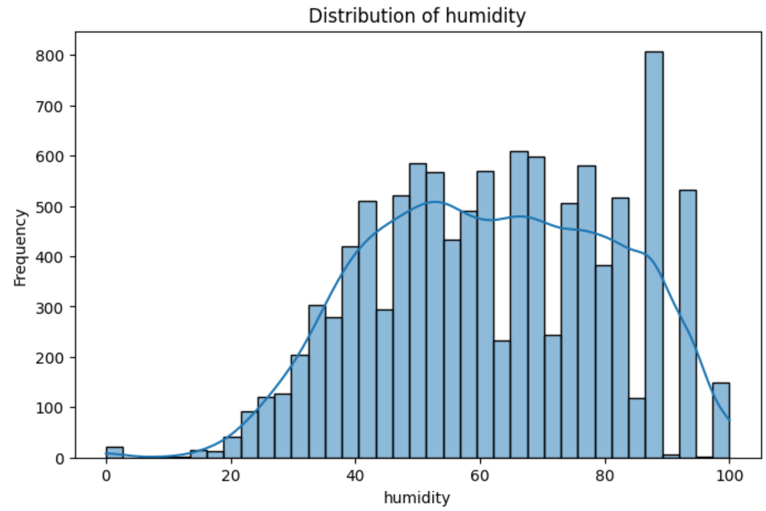
bikes on an average are used on days with average temprature of 20 celsius.

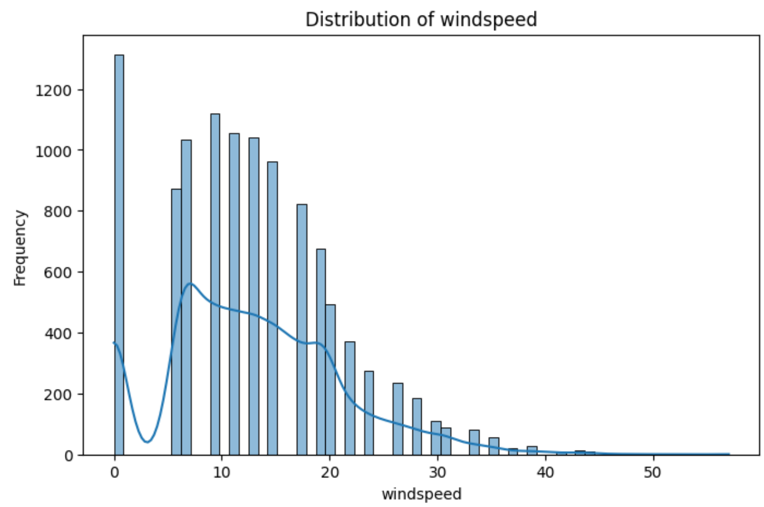
#Univariate and Bivariate anlysis

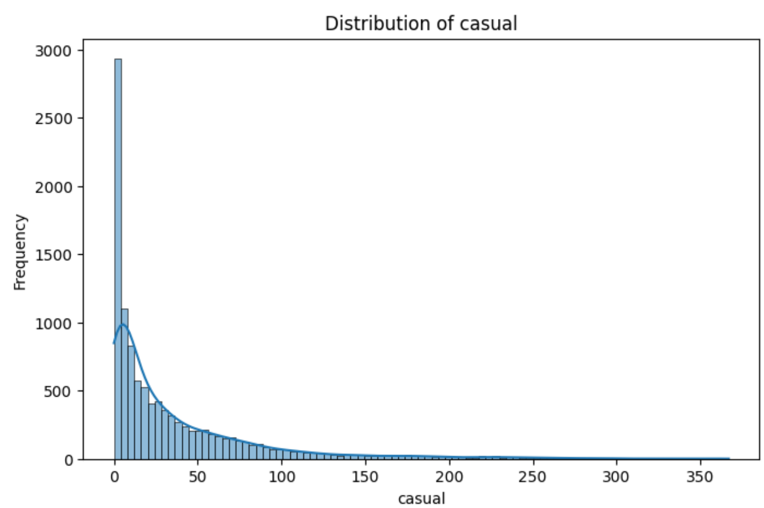
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


# List of continuous variables
continuous_vars = ['temp', 'atemp', 'humidity', 'windspeed', 'casual',
        'registered', 'count']

# distribution plots for  continuous variable
for var in continuous_vars:
    plt.figure(figsize=(8, 5))
    sns.histplot(df[var], kde=True)
    plt.title(f'Distribution of {var}')
    plt.xlabel(var)
    plt.ylabel('Frequency')
    plt.show()
```

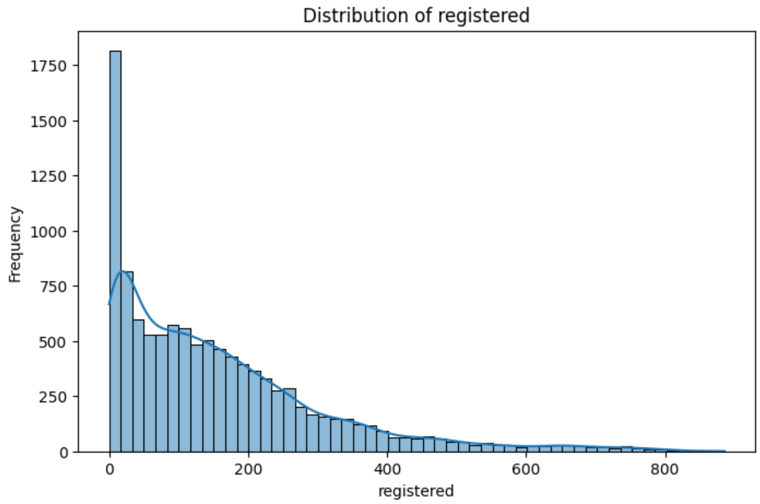


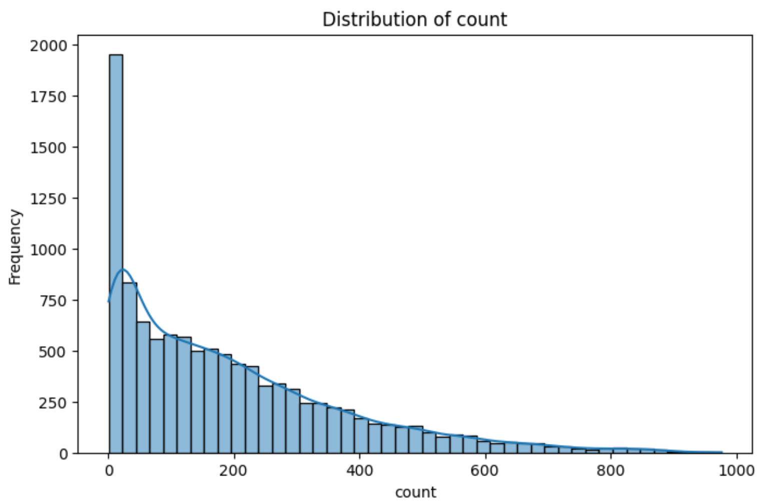



Distribution of humidity
Frequency
humidity
0
100
200
300
400
500
600
700
800
0
20
40
60
80
100


Distribution of windspeed
Frequency
windspeed
0
200
400
600
800
1000
1200
0
10
20
30
40
50

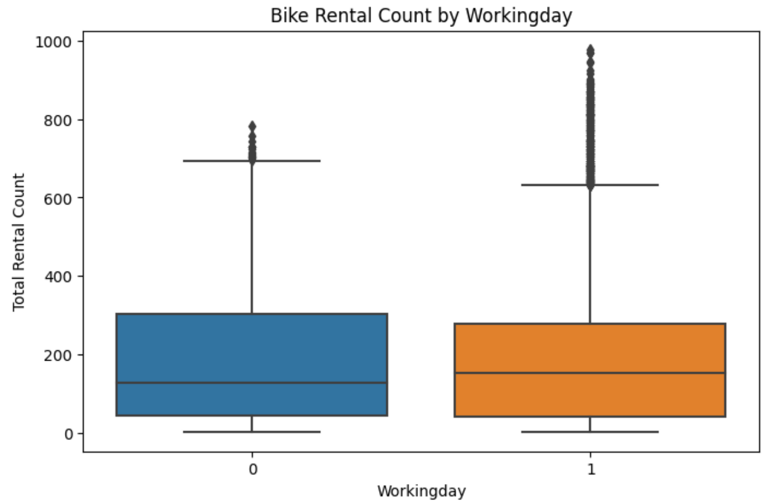

Distribution of casual
Frequency
casual
0
500
1000
1500
2000
2500
3000
0
50
100
150
200
250
300
350





#Bivariate

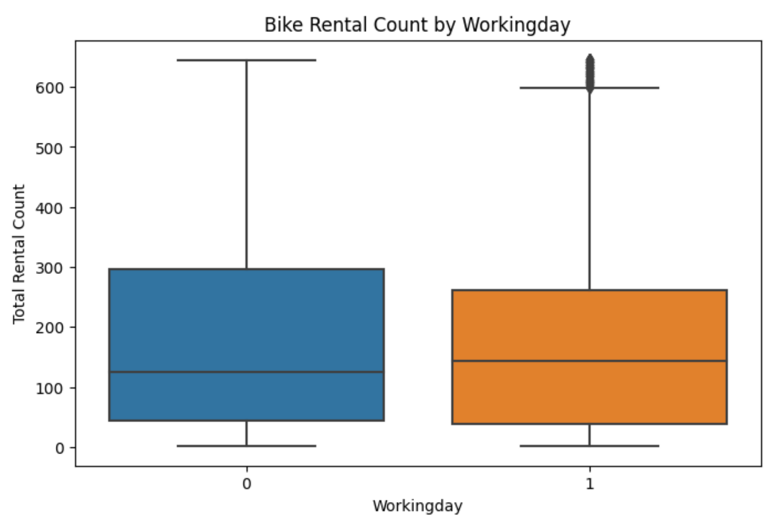
```
# Workingday vs. Count
plt.figure(figsize=(8, 5))
sns.boxplot(x='workingday', y='count', data=df)
plt.title('Bike Rental Count by Workingday')
plt.xlabel('Workingday')
plt.ylabel('Total Rental Count')
plt.show()
```



#Outlier treatment:

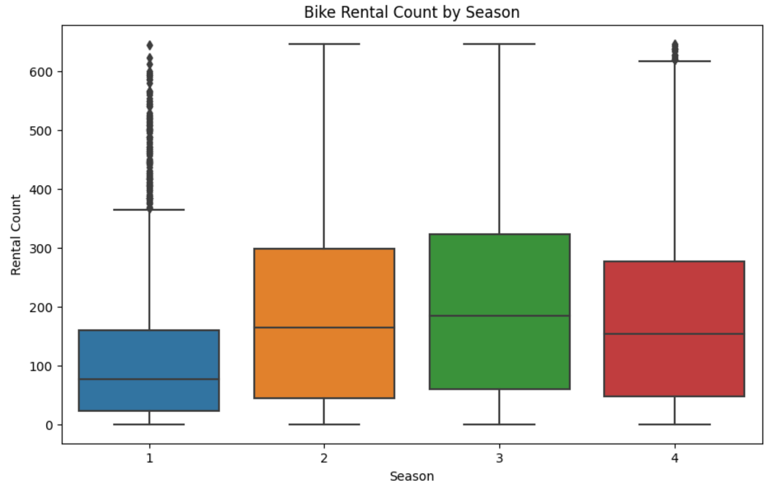
we are removing outliers because we have sufficent data here.

```
q1=df['count'].quantile(0.25)
q3=df['count'].quantile(0.75)
iqr=q3-q1

df=df[(df['count']>(q1-1.5*iqr))&(df['count']<(q3+1.5*iqr))]

# Workingday vs. Count after removing outliers.
plt.figure(figsize=(8, 5))
sns.boxplot(x='workingday', y='count', data=df)
plt.title('Bike Rental Count by Workingday')
plt.xlabel('Workingday')
plt.ylabel('Total Rental Count')
plt.show()
```



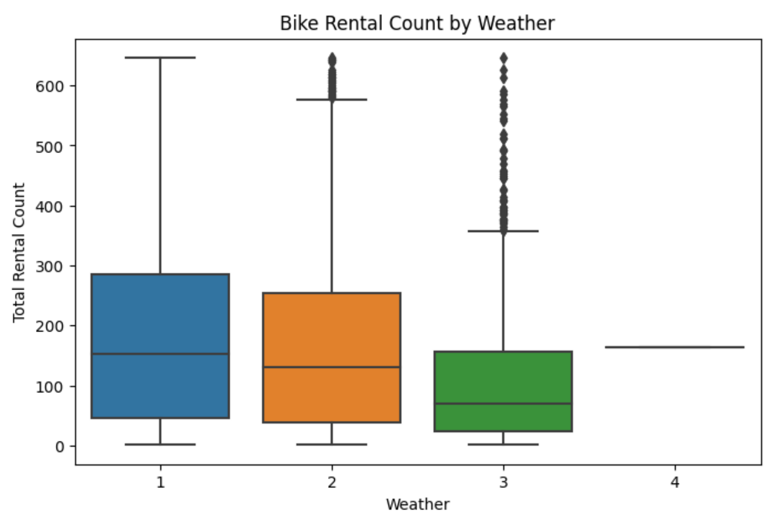
#Insights: There are a lot of outliers across working day and weekend. We dont have much clarity if weekend or weekday has higher effect on renting the bicycle. We will use hypothesis testing later.

```
# Season vs. Count
plt.figure(figsize=(10, 6))
sns.boxplot(x='season', y='count', data=df)
plt.title('Bike Rental Count by Season')
plt.xlabel('Season')
plt.ylabel('Rental Count')
plt.show()
```



Above we an see that the median for renting bicycle in season 1 is far below any season and there is no significant differnce between season 2,3,4

```
# Weather vs. Count
plt.figure(figsize=(8, 5))
sns.boxplot(x='weather', y='count', data=df)
plt.title('Bike Rental Count by Weather')
plt.xlabel('Weather')
plt.ylabel('Total Rental Count')
plt.show()
```



#Hypothesis Testing

we want to check if the week of the day have a effect on the demand of cycles.(ttest)

we use ttest because Population std is unknown.

Ho:The count on weekday is equal to the count on weekend.

Ha: The count on weekday is greater than count on weekend.

Ho: mu_1 = mu_2 Ha: mu_1 > mu_2

This is a one-tailed test concerning,the population standard deviations are unknown, the two sample independent t-test will be the appropriate test for this problem.

α = 0.05.

Checking for Assumptions of ttest:

Assumptions of ttest:

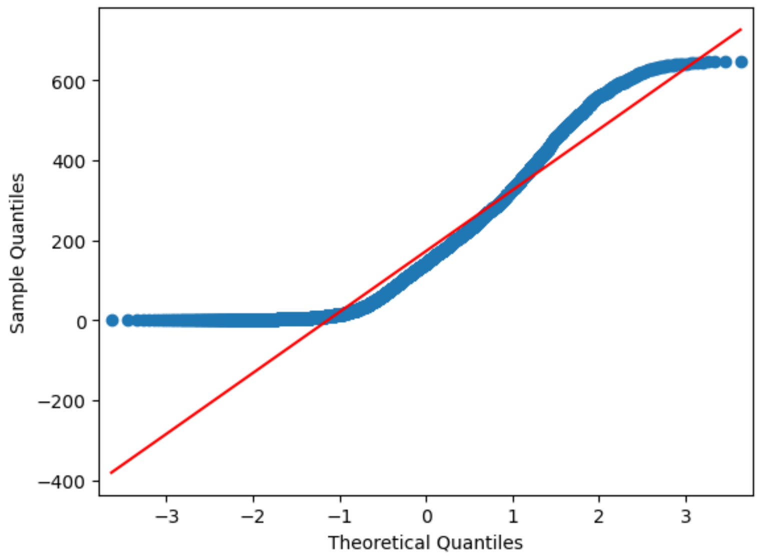
Observations in each sample are normally distributed.

Observations in each sample have the same variance

```
# defining the two groups

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

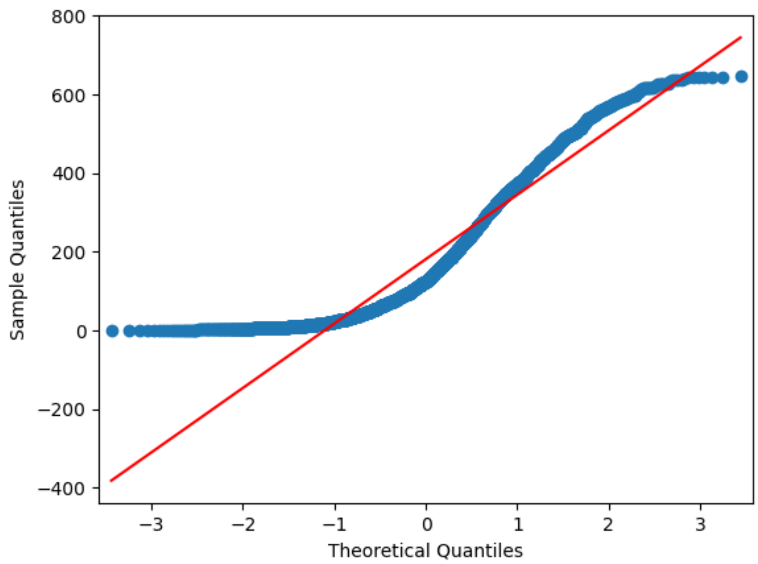
```
from statsmodels.graphics.gofplots import qqplot

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

```
<function matplotlib.pyplot.show(close=None, block=None)>
```




```
from statsmodels.graphics.gofplots import qqplot

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

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Doesnot follow normality.

Now checking for variance using levenes test:

```
#levens test
# Ho:All the count variances are equal
# Ha:All the count variances aredifffernt.

from scipy.stats import levene
test, p_val= levene(weekday,weekend)
print('The p-value is',p_val)
```

```
The p-value is 7.450676972537556e-07
```

```
if p_val<0.05:
    print("Reject Ho")

else:
    print("Fail to reject null hypothesis(Ho)")
```

```
Reject Ho
```

Thus both the assumptions of ttest have failed.

```
# doing the test ,its a one tailed test
test_stat,p_value=ttest_ind(weekday,weekend,alternative='greater')
```

```
p_value
```

```
0.9928730118891328
```

```
if p_value<0.05:
    print("Reject Ho")
else:
    print("Fail to reject null hypothesis(Ho)")
```

```
Fail to reject null hypothesis(Ho)
```

#Insight: As we can see our p-value comes out higher we can say that,working day has no affect on the cycles rented

#ANOVA

No. of cycles rented is similar or different in different weather

We are using Anova because we are doing comparisons for multiple groups.

Ho:The mean count in different weather are equal

Ha: The mean count in different weather are different

$\alpha = 0.05$

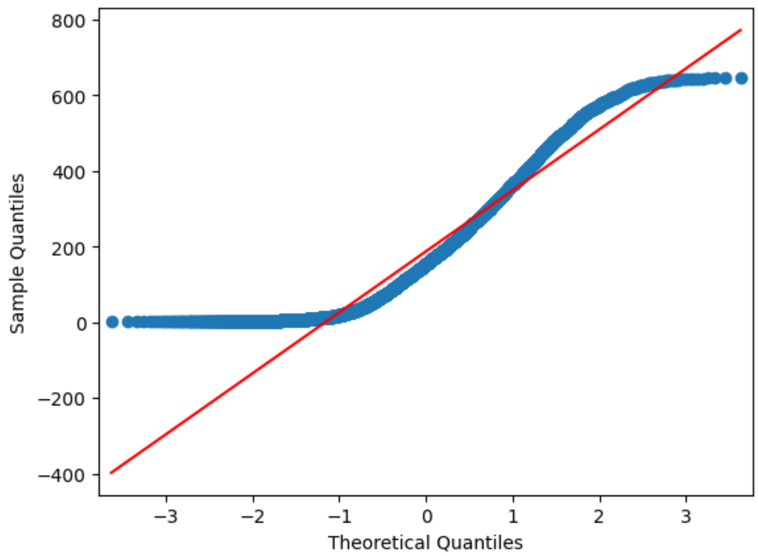
We will be using Annnova here because we are compring means of multiple groups

Testing the assumptions of Annova

```
#checking for equal variance - levenes test


w1=df[df['weather']==1]['count']
w2=df[df['weather']==2]['count']
w3=df[df['weather']==3]['count']
# we have dropped weather 4 because there is only 1 data point in it.

# checking if the data is normal-QQ plot
from statsmodels.graphics.gofplots import qqplot
```

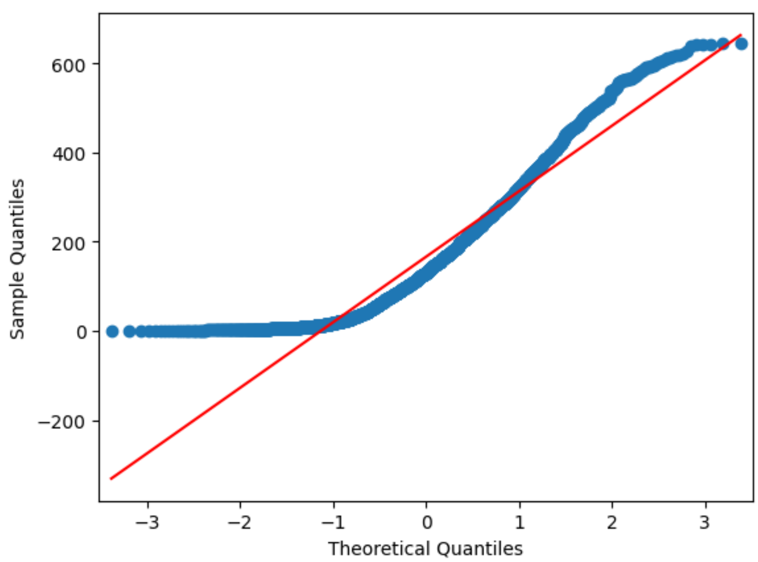
```
qqplot(w1,line='s')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



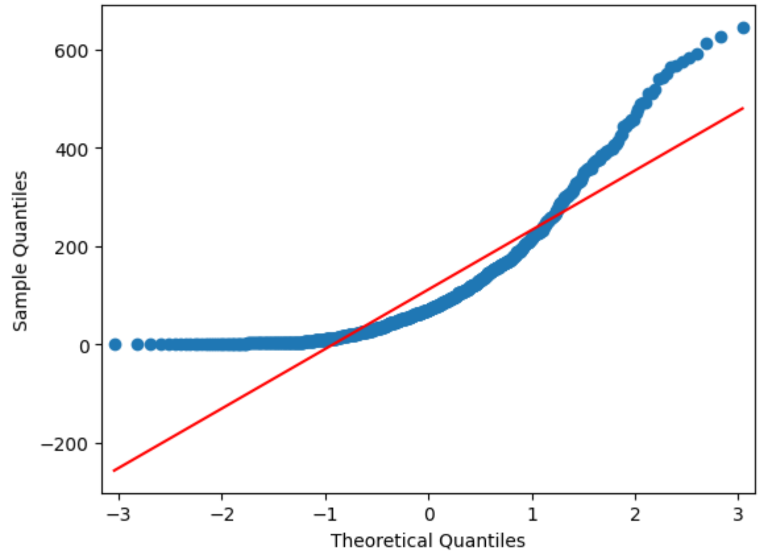

```
qqplot(w2,line='s')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```




```
qqplot(w3,line='s')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```




```
#levens test
# Ho:All the count variances are equal
# Ha:All the count variances aredifffernt.
```

```
from scipy.stats import levene
test, p_val= levene(w1,w2,w3)
print('The p-value is',p_val)
```

```
The p-value is 4.0910836644825954e-38
```

```
if p_val<0.05:
    print("Reject Ho")
else:
    print("Fail to reject null hypothesis(Ho)")
```

```
Reject Ho
```

The data is not normally distributed and varince is different which means that assumptions of Anova have failed.

#ANOVA

Ho:The mean count in different weather are equal

Ha: The mean count in different weather are different

```
test, p_val=f_oneway(w1,w2,w3)
p_val
```

```
2.749873188252358e-42
```

```
if p_val<0.05:
    print("Reject Ho")
else:
    print("Fail to reject null hypothesis,Since the p-value is greater
        than the 5% signicance level, we fail to reject the null
        hypothesis.")
```

```
Reject Ho
```

Insights:

As the p-values comes out to be much lower then 0.05 we will reject our null hypothesis and can say that the population count means under different weather conditions are the same meaning there is a difference in the usage of Yulu bikes in different weather conditions.

inshort weather impacts the usage of bicycle renting.

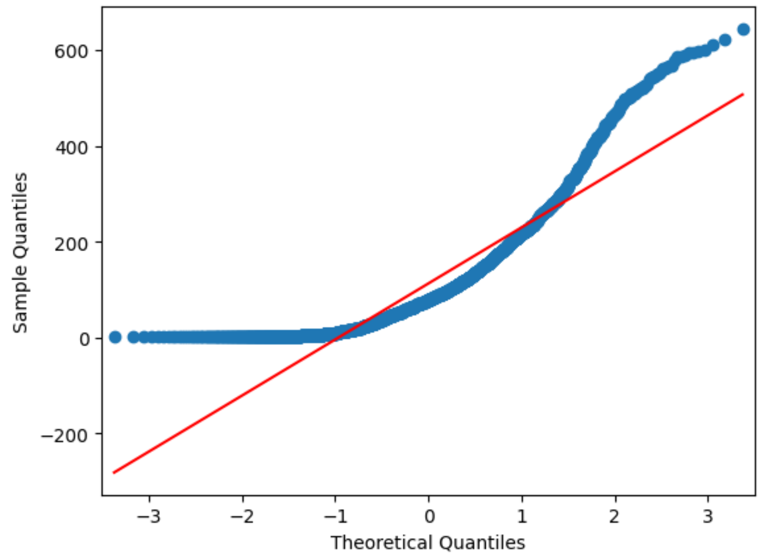
#No. of cycles rented is similar or different in different season

```
s1=df[df['season']==1]['count']
s2=df[df['season']==2]['count']
s3=df[df['season']==3]['count']
s4=df[df['season']==4]['count']
```

Checking assumptions of Annova test

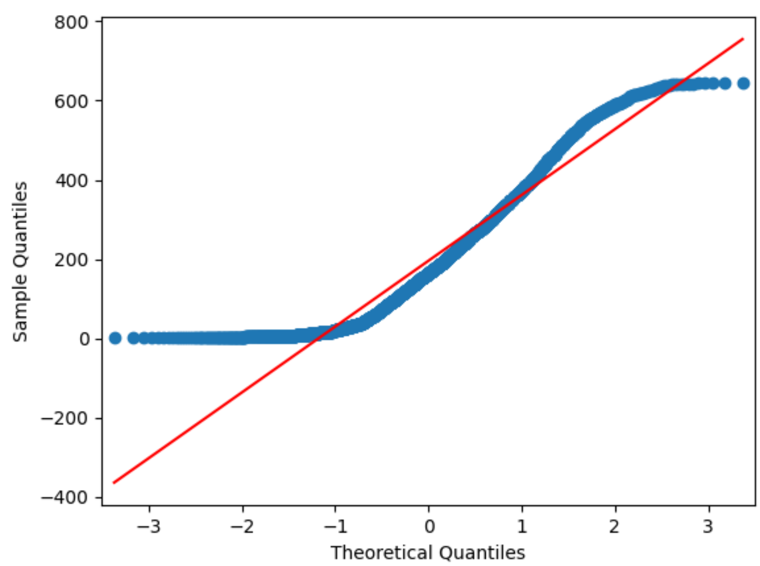
```
# checking if the data is normal
from statsmodels.graphics.gofplots import qqplot

qqplot(s1,line='s')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



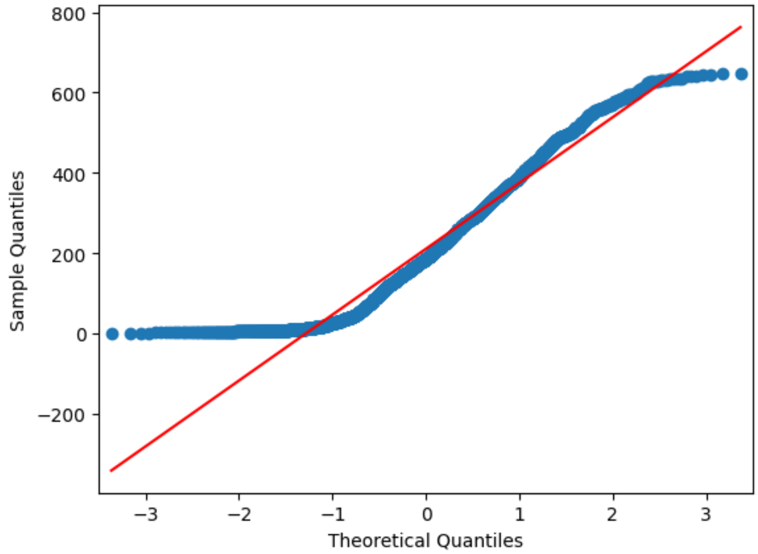

```
qqplot(s2,line='s')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



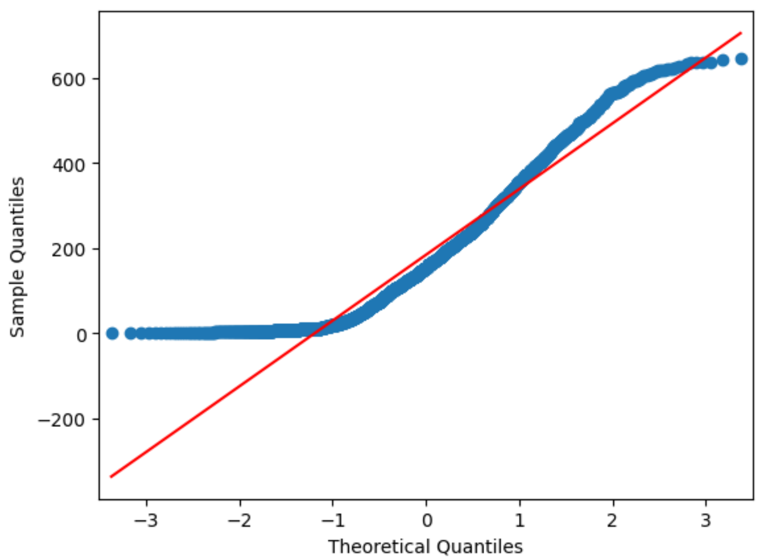

```
qqplot(s3,line='s')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```




```
qqplot(s4,line='s')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```




```
#levens test
# Ho:All the count variances are equal
# Ha: All the count variances are  differnt.
# α = 0.05
test, p_val=levene(s1,s2,s3,s4)


p_val
```

```
2.6643548968275643e-112
```

```
if p_val<0.05:
    print("Reject Ho")

else:
    print("Fail to reject null hypothesis(Ho)")
```

```
Reject Ho
```

There are different variences among different seasons,and the data isnt normlly distributed which means assumptions of Anova have failed.

#ANOVA

Ho:The mean count in different seasons are equal

Ha: The mean count in different seasons are different

```
test,p_val=f_oneway(s1,s2,s3,s4)
p_val
```

```
8.891092521664531e-137
```

```
if p_val<0.05:
    print("Reject Ho")
else:
    print("Fail to reject null hypothesis(Ho)")
```

```
Reject Ho
```

#Insight:

As the p_val comes out to be lower then 0.05 we can say that population count means under different seasons are not the same, meaning there is a difference in the usage of Yulu bikes in different seasons.

#Chi2test

We use chi2 because we are comparing two categorical variables.

H0: Weather is not dependent on the season

Ha: Weather is dependent on the season

α = 0.05

Checking the assumptions of Chi2 test,which are as follows:

1. Variables must be categorical.
2. Expected frequency in each cell of contingency table is reasonably high.

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

season	**1**	**2**	**3**	**4**
weather				
1	1744	1720	1842	1656
2	714	690	579	787
3	211	223	195	221
4	1	0	0	0

```
result = chi2_contingency(weather_season)
result
```

```
Chi2ContingencyResult(statistic=47.16590591959627,
pvalue=3.6550317439064896e-07, dof=9,
expected_freq=array([[1.75645280e+03, 1.73211244e+03, 1.72092904e+03,
1.75250572e+03],
       [6.98847208e+02, 6.89162808e+02, 6.84713219e+02,
6.97276765e+02],
       [2.14447699e+02, 2.11475952e+02, 2.10110555e+02,
2.13965794e+02],
       [2.52291411e-01, 2.48795238e-01, 2.47188888e-01, 2.51724464e-
01]]))
```

We can see that the frequency in each cell of contingency table is reasonably high and we have seen in earlier cases as wel that weather and season are categorical variables withvalues ranging from 1-4.

Thus, the assumptions of chi2 test are followed.

```
p_val = result.pvalue
p_val
```

```
3.6550317439064896e-07
```

```
if p_val<0.05:
    print("Reject Ho")
else:
    print("Fail to reject null hypothesis(Ho)")
```

```
Reject Ho
```

Insights: As the p-value comes out to be lower we can conclude that weather is dependent on season.

```
# Group by humidity and calculate the sum of counts
humidity_count_sum = df.groupby('humidity')['count'].sum()


# Find the maximum sum of counts and corresponding humidity
max_count_sum = humidity_count_sum.max()
humidity_with_max_count_sum = humidity_count_sum[humidity_count_sum ==
        max_count_sum].index[0]

print("Humidity with the highest count sum:",
        humidity_with_max_count_sum)
```

```
Humidity with the highest count sum: 46
```

```
temp_count_sum = df.groupby('temp')['count'].sum()


# Find the maximum sum of counts and corresponding temperature
max_count_sum = temp_count_sum.max()
temp_with_max_count_sum = temp_count_sum[temp_count_sum ==
        max_count_sum].index[0]

print("Temperature with the highest count sum:",
        temp_with_max_count_sum)
```

```
Temperature with the highest count sum: 28.7
```

#Insights and Recommendations:

In ttest for weekend and weekday the p-value comes out higher we can say that,working day has no affect on the cycles rented.

For effect of weather on cycles rented,the p-value comes out to be much lower thus the population count means under different weather conditions are the same meaning there is a difference in the usage of Yulu bikes in different weather conditions. inshort weather impacts the usage of bicycle renting.

For effect of season on cycles rented,population count means under different seasons are not the same, meaning there is a difference in the usage of Yulu bikes in different seasons.

The above two points were proved using Anova.

From running a chi2test on dependancy of weather on season we see that weather and season depend on each other.

We can also see that humidity and temprature have some effect on the number of cycles rented.

#Recommendations:

The variables that are significant in predicting the demand for shared electric cycles in the Indian market are weather and season.We saw that mostly in cloudy weather yulu bikes were preffered and to improve the bike rentals some small coolerfans can be attached to the handle of the bikes which will help with better ride for the customer.To increse renting during rainy seasson some type of protective sheilds can be placed on the cycles, durable rainy coates can also be provided.

Some other factors which impact the renting are workday and weekend,temprature and humidity.To imprve the usage during weekend city exploration drives can be conducted,providing facilties for city exploration with family and friends,since yulu is providing environment friendly option of commuting some cleanliness drives can be conducted which will attract the attention of people who are looking for eco friendly options.

To attract more customers special discounts and reward points can also be given and to get more regular customers the cncept of reward points can be really helpful and loyalty programs like refferl points etc can be used.

Areas with high foot traffic should be identified, ares were there are transportation problems and by collaborating with universities and corporates with bigger campuses will ease commuting within such campuses.

#Conclusion:

There is a good impact of all of the factors discussed on the demand for cycles; taking the above-given insights and recommendations into consideration, it can be possible to build a strong customer base for Yulu.