Yulu – Case Study

About Yulu:

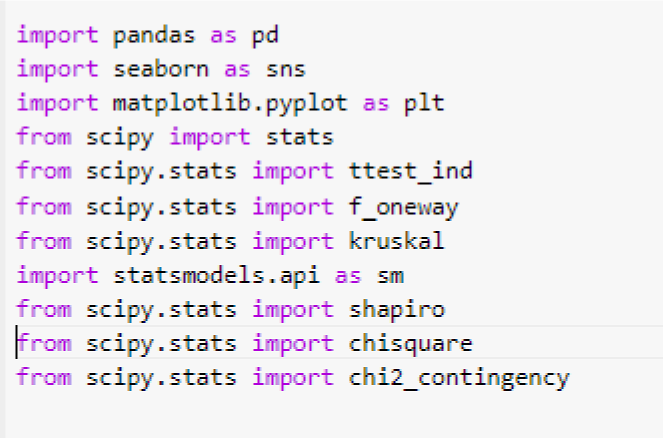
Yulu is a micro-mobility startup that operates bike-sharing and electric scooter-sharing services in several cities in India. Founded in 2017 by Amit Gupta, RK Misra, Naveen Dachuri, and Hemant Gupta.

Yulu aims to provide sustainable and affordable transportation solutions to urban commuters. Yulu offers dockless bicycles and electric scooters that users can rent through a mobile app, making it convenient for short-distance travels within cities. The company focuses on promoting eco-friendly transportation options, reducing traffic congestion, and enhancing last-mile connectivity in urban areas.

Problem Statement:

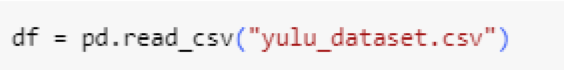
1. Yulu has recently suffered considerable dips in its revenues. We need to find the reason? Which variable is affecting the usage of Yulu bikes?
2. Which variables are significant in predicting the demand for shared electric cycles in the Indian market?
3. Key factors influencing the demand for shared electric cycles operated by Yulu in the Indian market.

Python Library imported for the case study:

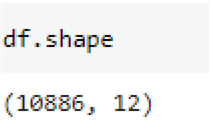


```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
from scipy.stats import ttest_ind
from scipy.stats import f_oneway
from scipy.stats import kruskal
import statsmodels.api as sm
from scipy.stats import shapiro
from scipy.stats import chisquare
from scipy.stats import chi2_contingency
```

1. Basic analysis:



```
df = pd.read_csv("yulu_dataset.csv")
```

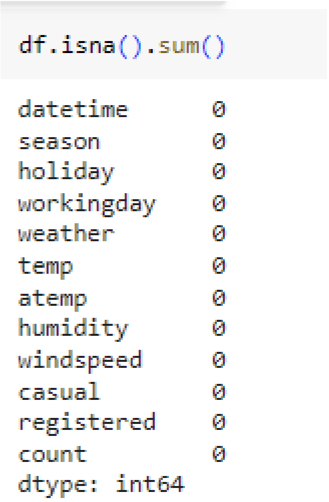


```
df.shape
```

```
(10886, 12)
```

Insight:

The dataset has 10886 rows and 12 columns

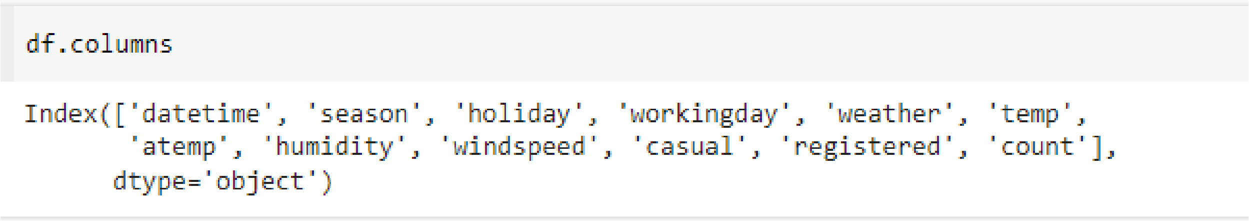


```
df.isna().sum()
```

```
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
```

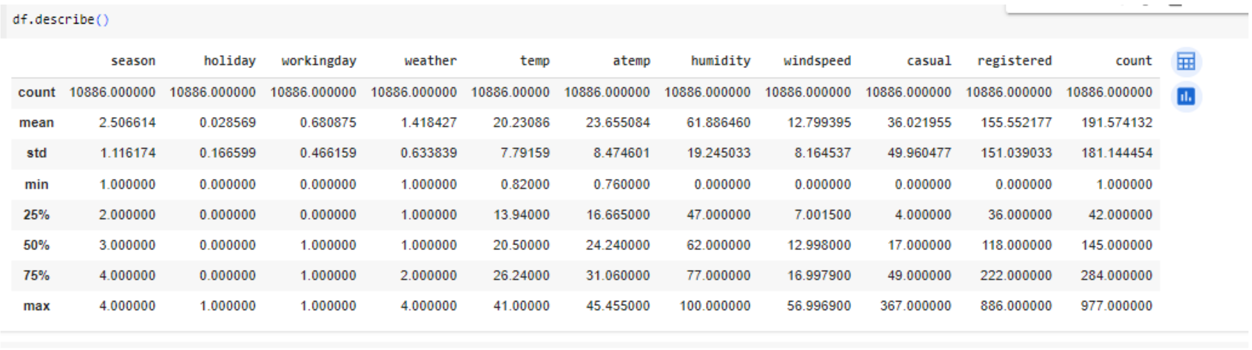
Insight:

The dataset doesnot have any null values



```
df.columns
```

```
Index(['datetime', 'season', 'holiday', 'workingday', 'weather', 'temp',
       'atemp', 'humidity', 'windspeed', 'casual', 'registered', 'count'],
      dtype='object')
```



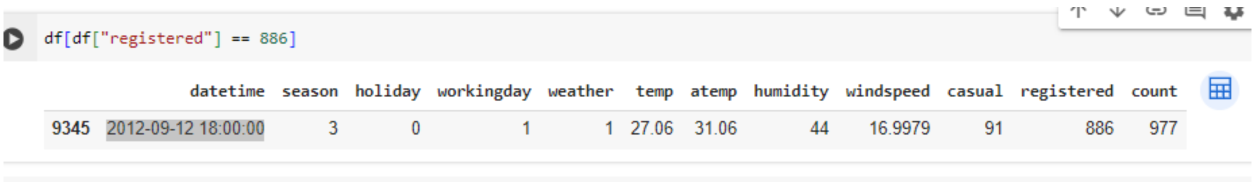
```
df.describe()
```

	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
count	10886.000000	10886.000000	10886.000000	10886.000000	10886.00000	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000
mean	2.506614	0.028569	0.680875	1.418427	20.23086	23.655084	61.886460	12.799395	36.021955	155.552177	191.574132
std	1.116174	0.166599	0.466159	0.633839	7.79159	8.474601	19.245033	8.164537	49.960477	151.039033	181.144454
min	1.000000	0.000000	0.000000	1.000000	0.82000	0.760000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	2.000000	0.000000	0.000000	1.000000	13.94000	16.665000	47.000000	7.001500	4.000000	36.000000	42.000000
50%	3.000000	0.000000	1.000000	1.000000	20.50000	24.240000	62.000000	12.998000	17.000000	118.000000	145.000000
75%	4.000000	0.000000	1.000000	2.000000	26.24000	31.060000	77.000000	16.997900	49.000000	222.000000	284.000000
max	4.000000	1.000000	1.000000	4.000000	41.00000	45.455000	100.000000	56.996900	367.000000	886.000000	977.000000

Insight:

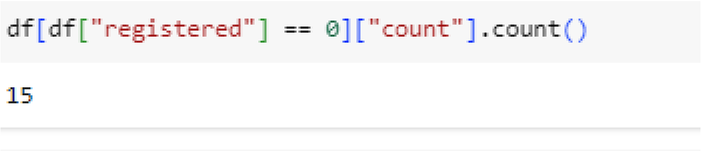
Eventhough season, holiday, workingday, weather has numerical data, They represent categorical values.

1. Working day – 1-holiday or weekend 0 – not a holiday
2. Weather – 4 seasons
    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
3. Seasons: 4 sesons
    1. Spring
    2. Summer
    3. fall
    4. winter
4. temp values ranges from 0 to 41
5. atemp values ranges from 0 tp 45
6. humidity ranges from 0 t0 45
7. windspeed ranges from 0 to 57
8. registered users with the max count of 886 on 2012-09-12 18:00:00and min count 0 is registered for 15 days.



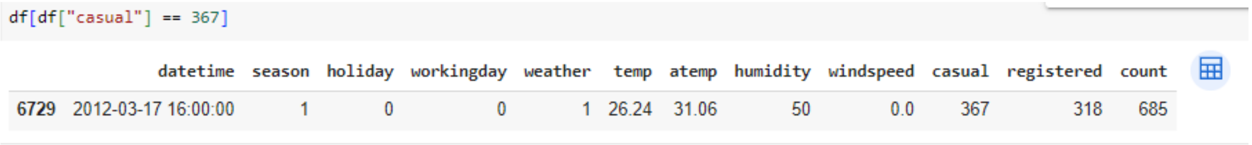
```
df[df["registered"] == 886]
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
9345	2012-09-12 18:00:00	3	0	1	1	27.06	31.06	44	16.9979	91	886	977



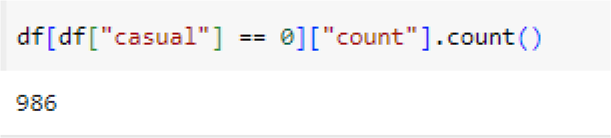
```
df[df["registered"] == 0]["count"].count()

15
```

9. causal users with max count of 367 was registered on 2012-03-17 16:00:00 and min count of 0 was registered for 986



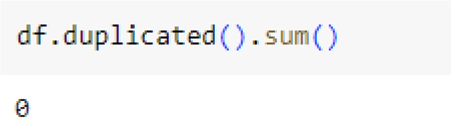
```
df[df["casual"] == 367]
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
6729	2012-03-17 16:00:00	1	0	0	1	26.24	31.06	50	0.0	367	318	685



```
df[df["casual"] == 0]["count"].count()
```

```
986
```

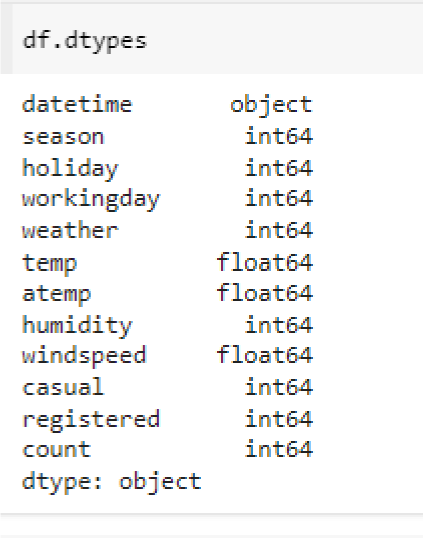


```
df.duplicated().sum()
```

```
0
```

Insight:

The dataset has no duplicate values.



```
df.dtypes
```

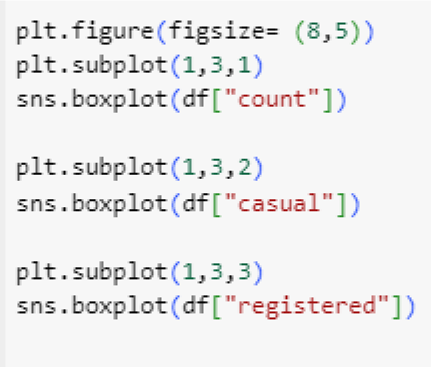
```
datetime       object
season          int64
holiday         int64
workingday      int64
weather         int64
temp          float64
atemp         float64
humidity        int64
windspeed     float64
casual          int64
registered      int64
count           int64
dtype: object
```

Insight:

Except datetime which is an object. All the other columns are numerical datatypes

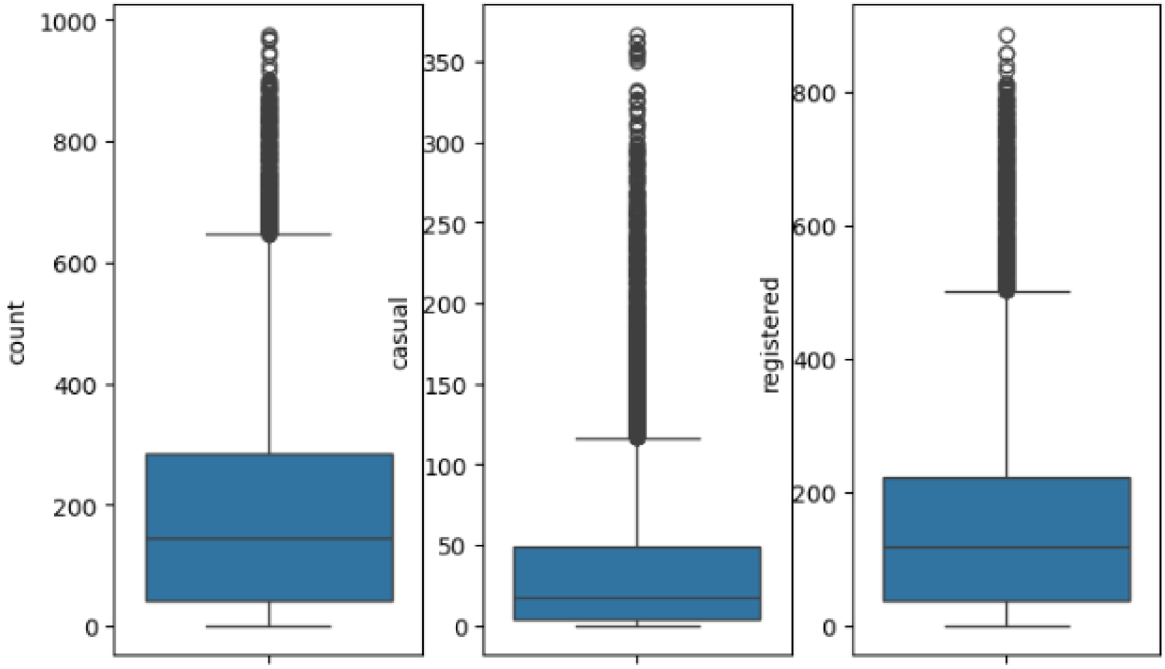
2. Outliers:

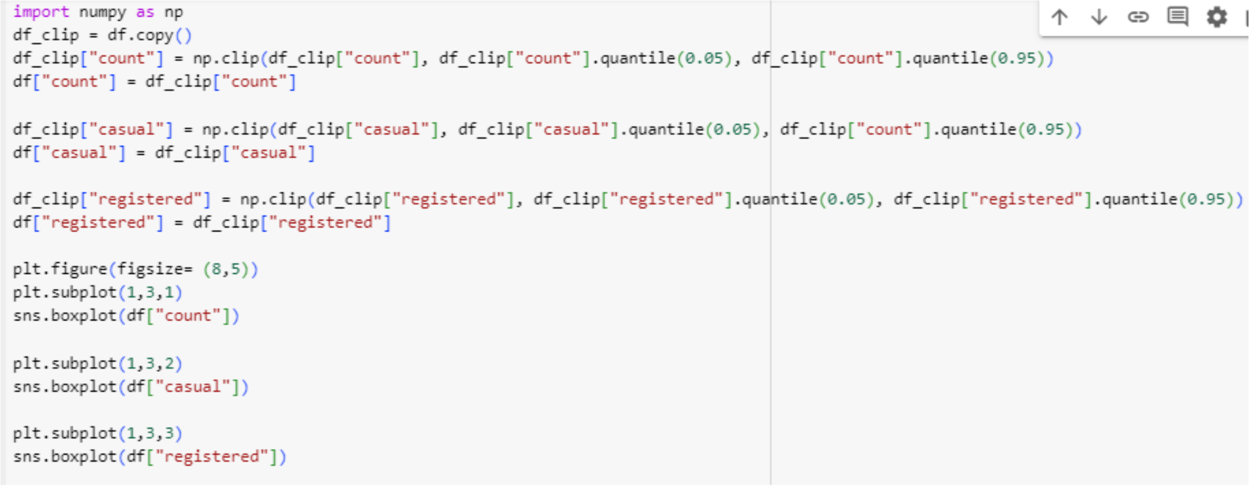
Outliers can be detected using boxplot.



```
plt.figure(figsize= (8,5))
plt.subplot(1,3,1)
sns.boxplot(df["count"])

plt.subplot(1,3,2)
sns.boxplot(df["casual"])

plt.subplot(1,3,3)
sns.boxplot(df["registered"])
```

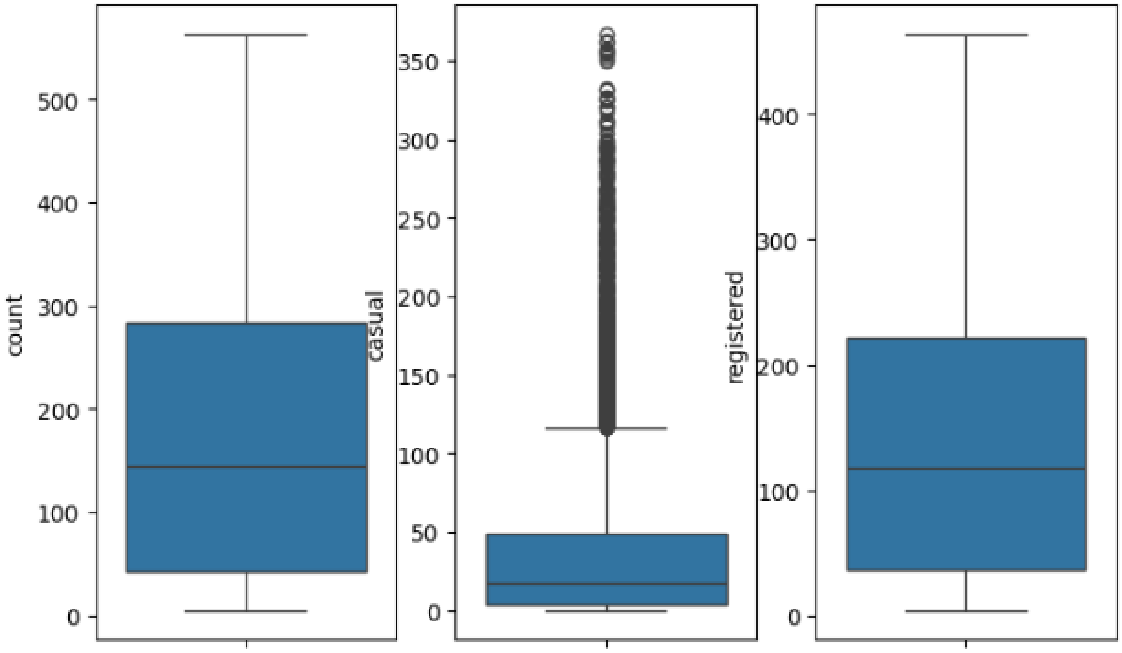
```
<Axes: ylabel='registered'>
```





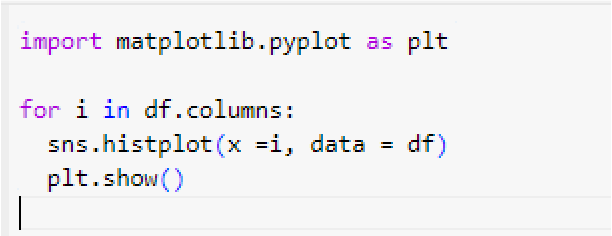
```
import numpy as np
df_clip = df.copy()
df_clip["count"] = np.clip(df_clip["count"], df_clip["count"].quantile(0.05), df_clip["count"].quantile(0.95))
df["count"] = df_clip["count"]

df_clip["casual"] = np.clip(df_clip["casual"], df_clip["casual"].quantile(0.05), df_clip["count"].quantile(0.95))
df["casual"] = df_clip["casual"]

df_clip["registered"] = np.clip(df_clip["registered"], df_clip["registered"].quantile(0.05), df_clip["registered"].quantile(0.95))
df["registered"] = df_clip["registered"]

plt.figure(figsize= (8,5))
plt.subplot(1,3,1)
sns.boxplot(df["count"])

plt.subplot(1,3,2)
sns.boxplot(df["casual"])

plt.subplot(1,3,3)
sns.boxplot(df["registered"])
```

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

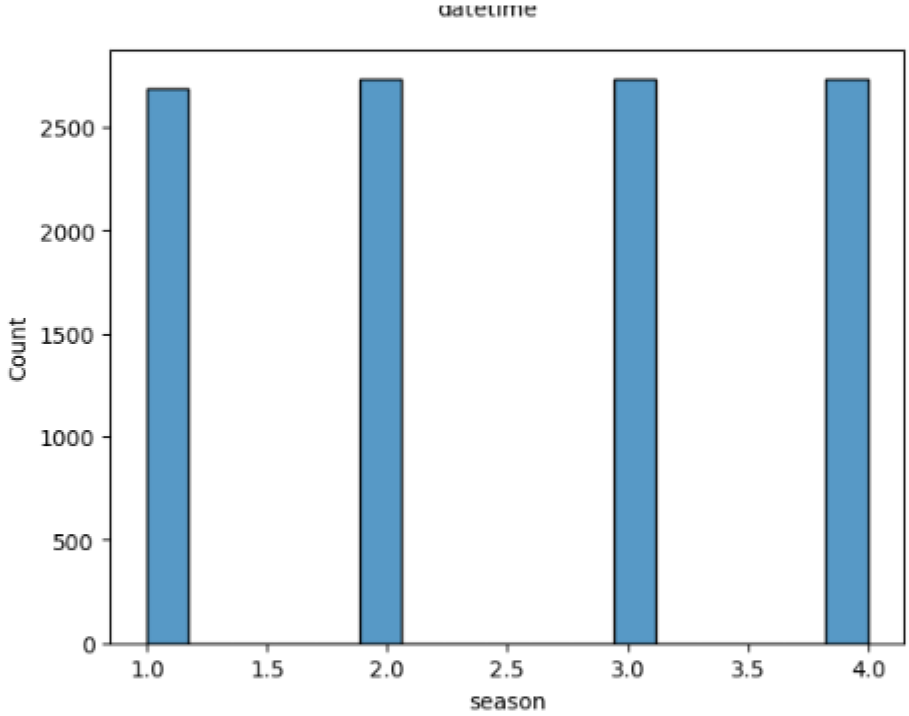


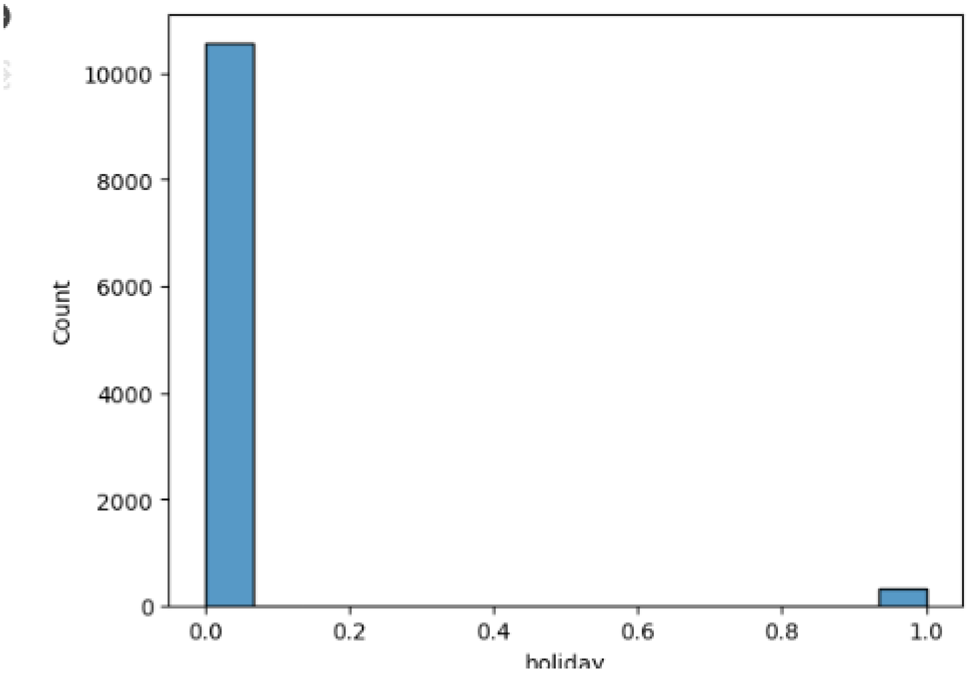
The outliers are removed.

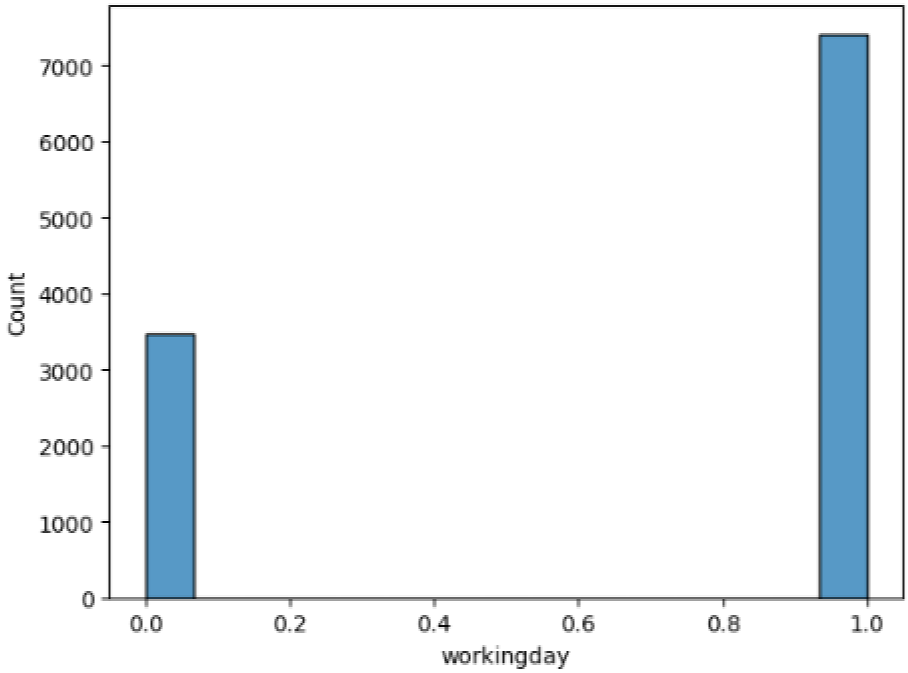
3. Univariant Analysis:

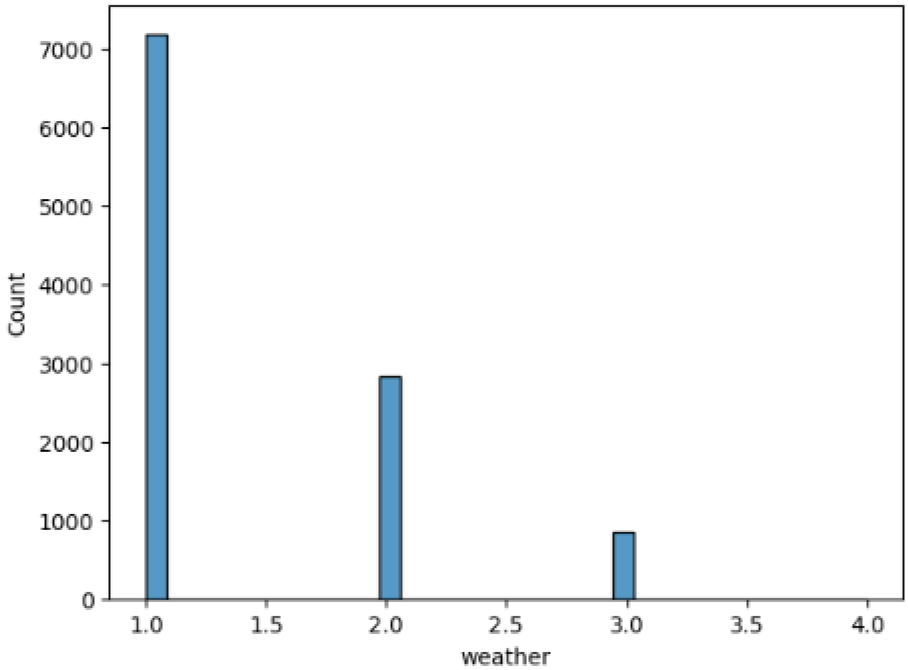


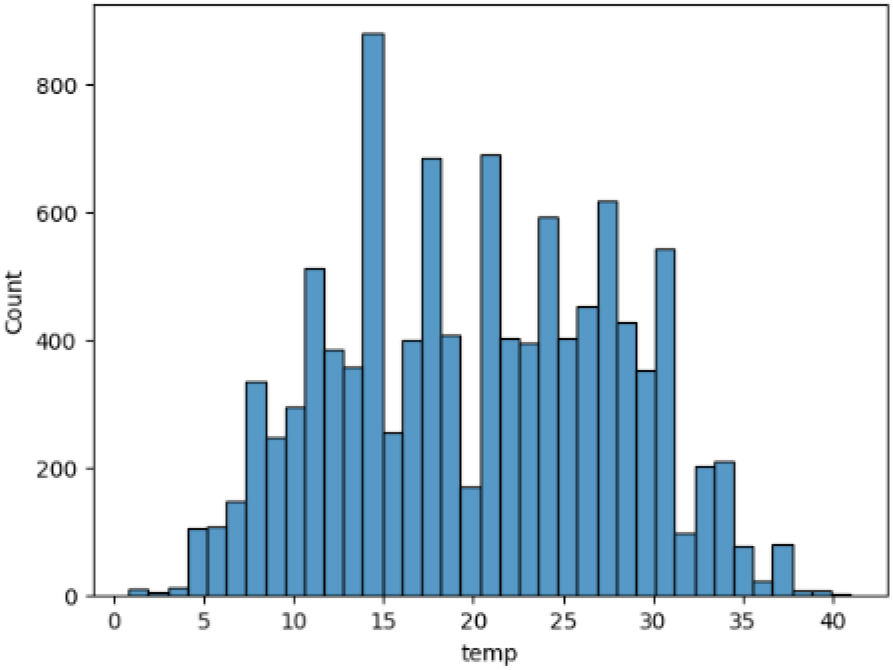
```
import matplotlib.pyplot as plt

for i in df.columns:
  sns.histplot(x =i, data = df)
  plt.show()
```

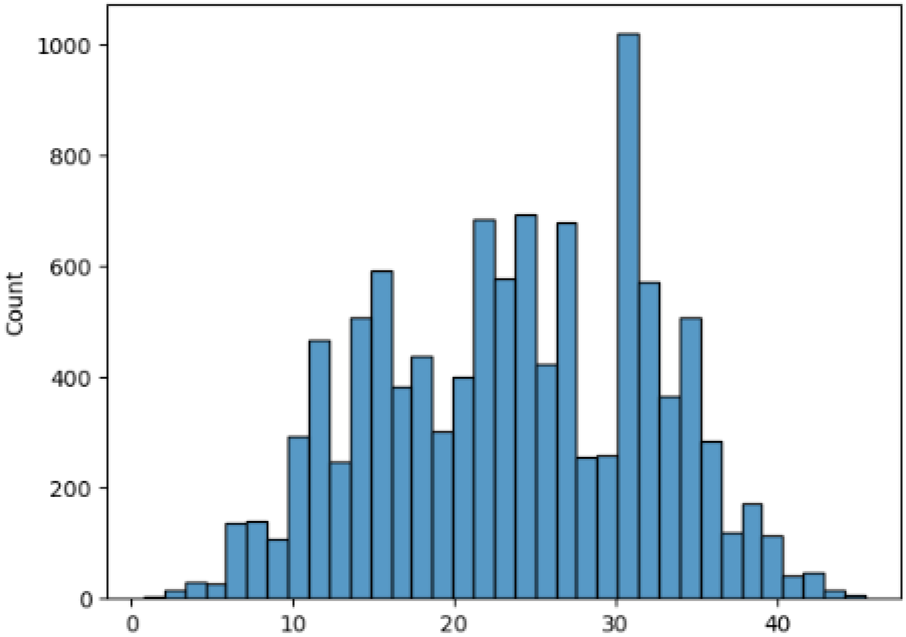


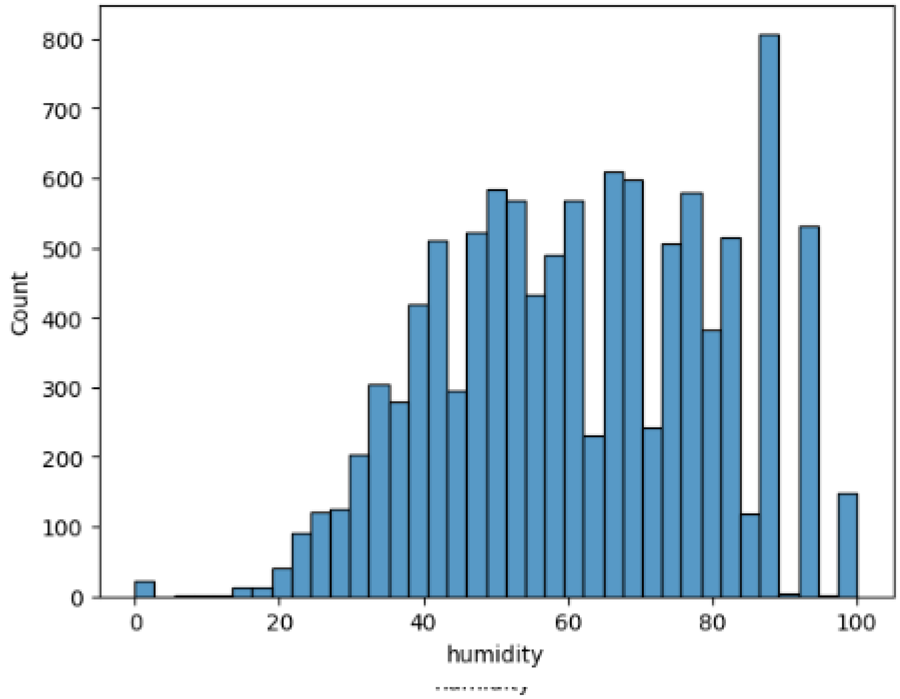



7000
6000
5000
4000
3000
2000
1000
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
workingday

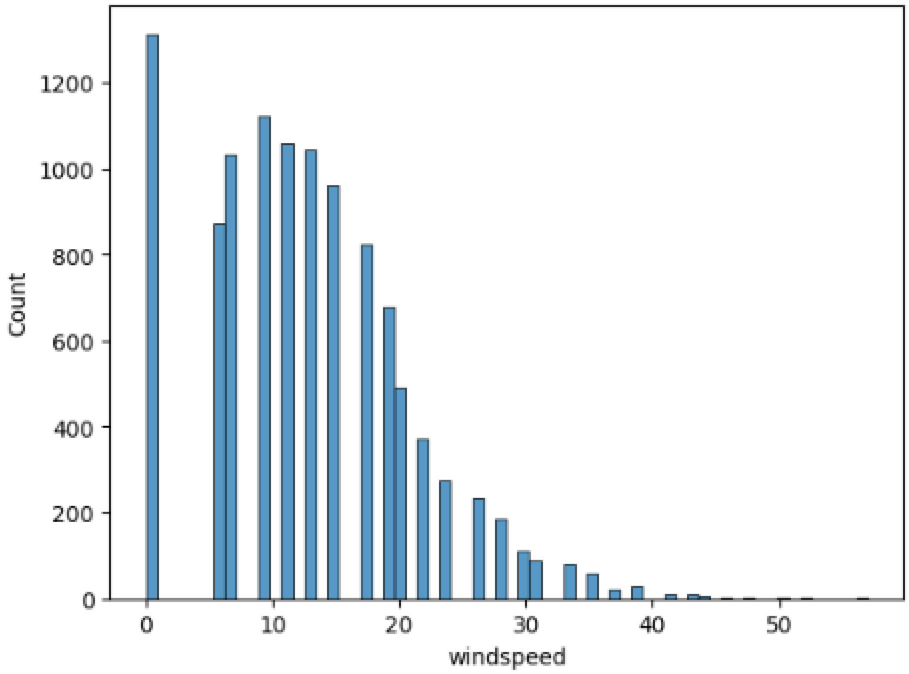

7000
6000
5000
4000
3000
2000
1000
0
Count
1.0
1.5
2.0
2.5
3.0
3.5
4.0
weather


Count
800
600
400
200
0
0
5
10
15
20
25
30
35
40
temp

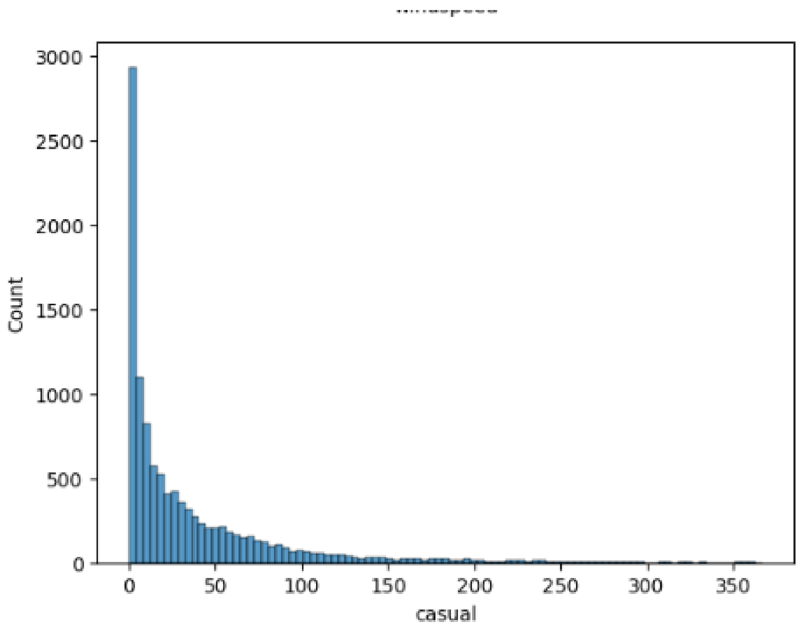

Count
1000
800
600
400
200
0
0
10
20
30
40

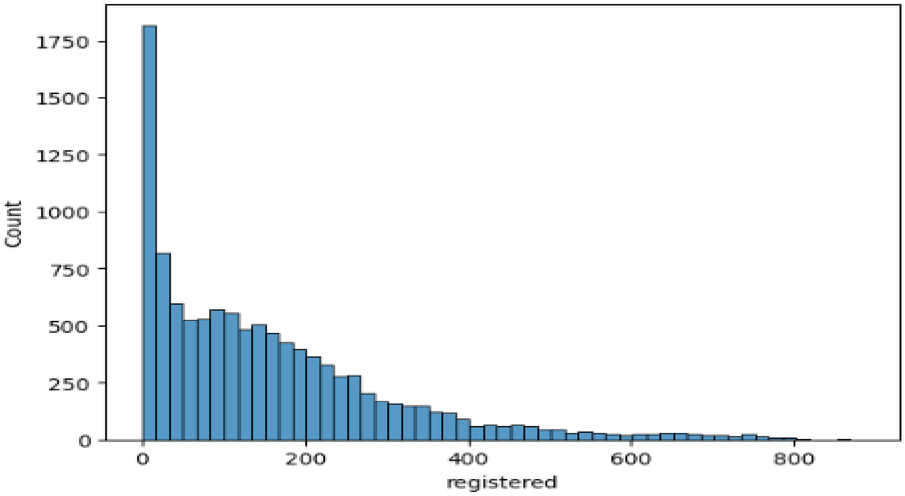


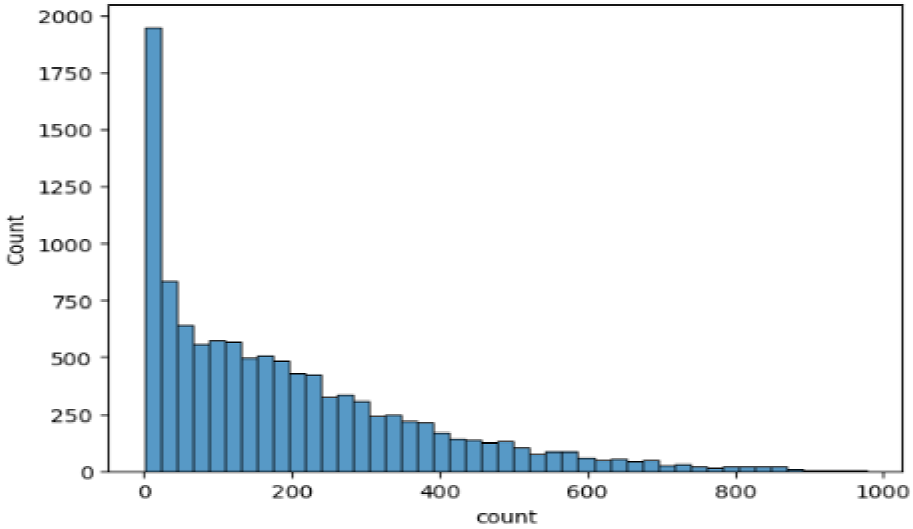
Count
800
700
600
500
400
300
200
100
0
0
20
40
60
80
100
humidity
humidity



Count
1200
1000
800
600
400
200
0
0
10
20
30
40
50
windspeed

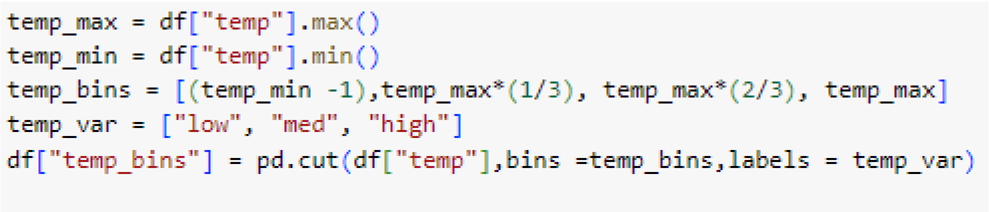

windspeed
3000
2500
2000
1500
1000
500
0
Count
0
50
100
150
200
250
300
350
casual


1750
1500
1250
1000
750
500
250
0
Count
0
200
400
600
800
registered

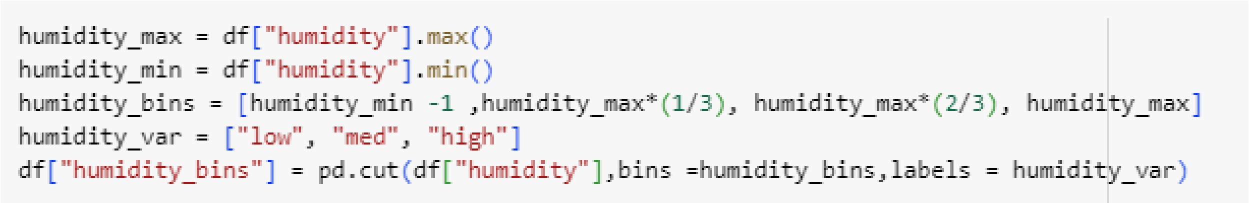

2000
1750
1500
1250
1000
750
500
250
0
Count
0
200
400
600
800
1000
count

4. Bivariant Analysis:

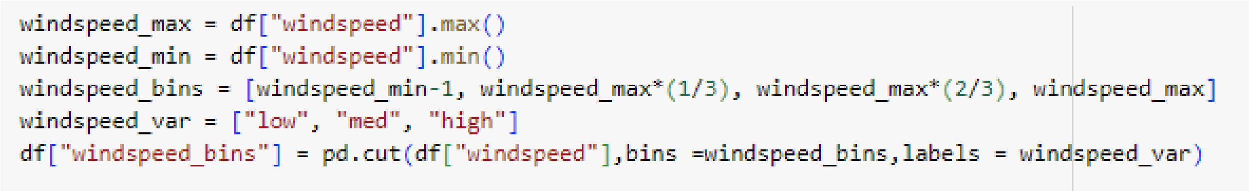
To understand how temp, atemp, humidity and windspeed affect the usage of yulu bikes, they are converted into categorical values - (low, medium, high)



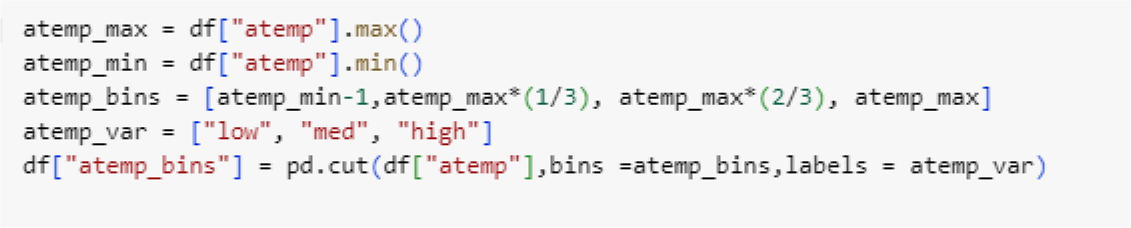
```
temp_max = df["temp"].max()
temp_min = df["temp"].min()
temp_bins = [(temp_min -1),temp_max*(1/3), temp_max*(2/3), temp_max]
temp_var = ["low", "med", "high"]
df["temp_bins"] = pd.cut(df["temp"],bins =temp_bins,labels = temp_var)
```



```
humidity_max = df["humidity"].max()
humidity_min = df["humidity"].min()
humidity_bins = [humidity_min -1 ,humidity_max*(1/3), humidity_max*(2/3), humidity_max]
humidity_var = ["low", "med", "high"]
df["humidity_bins"] = pd.cut(df["humidity"],bins =humidity_bins,labels = humidity_var)
```

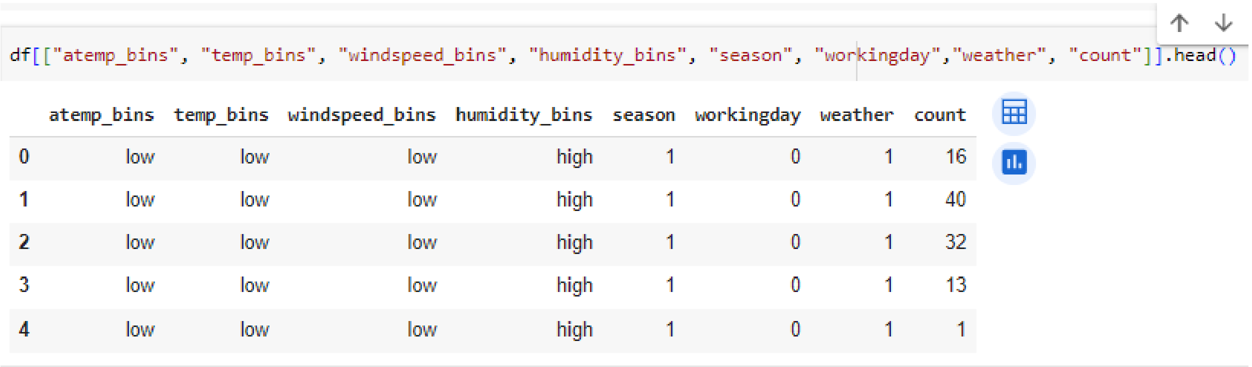


```
windspeed_max = df["windspeed"].max()
windspeed_min = df["windspeed"].min()
windspeed_bins = [windspeed_min-1, windspeed_max*(1/3), windspeed_max*(2/3), windspeed_max]
windspeed_var = ["low", "med", "high"]
df["windspeed_bins"] = pd.cut(df["windspeed"],bins =windspeed_bins,labels = windspeed_var)
```



```
atemp_max = df["atemp"].max()
atemp_min = df["atemp"].min()
atemp_bins = [atemp_min-1,atemp_max*(1/3), atemp_max*(2/3), atemp_max]
atemp_var = ["low", "med", "high"]
df["atemp_bins"] = pd.cut(df["atemp"],bins =atemp_bins,labels = atemp_var)
```

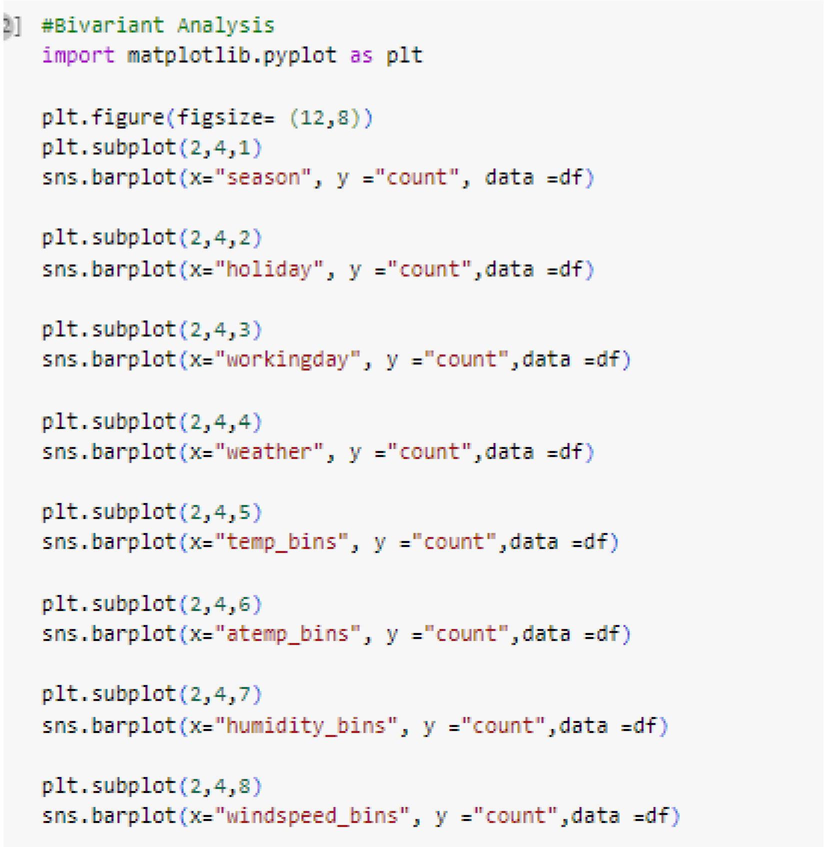
After executing the above code, the data we are using for analysis is,



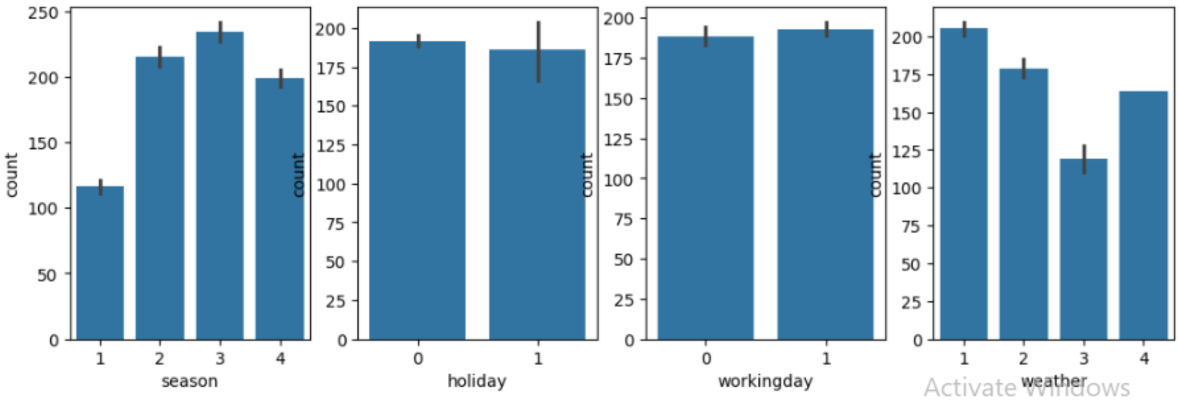
```
df[["atemp_bins", "temp_bins", "windspeed_bins", "humidity_bins", "season", "workingday","weather", "count"]].head()
```

	atemp_bins	temp_bins	windspeed_bins	humidity_bins	season	workingday	weather	count
0	low	low	low	high	1	0	1	16
1	low	low	low	high	1	0	1	40
2	low	low	low	high	1	0	1	32
3	low	low	low	high	1	0	1	13
4	low	low	low	high	1	0	1	1

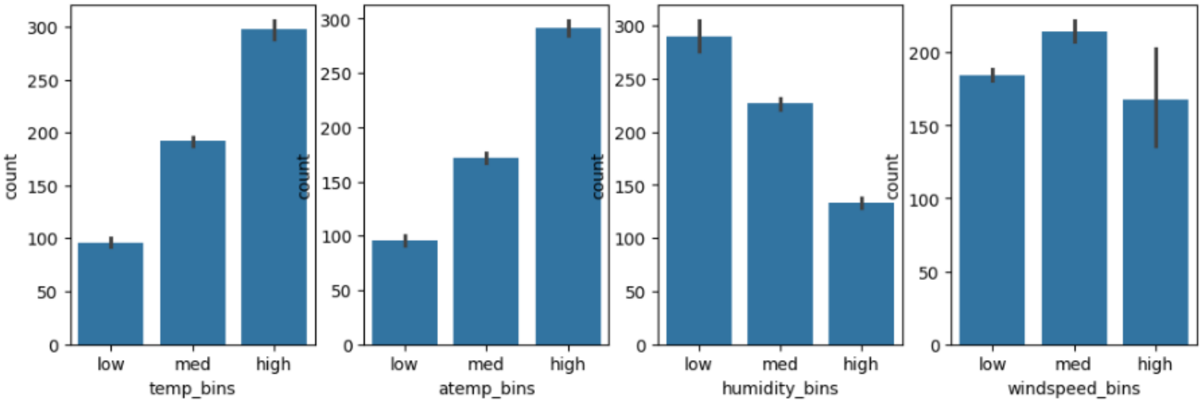
Bivariant Analysis on Count vs Categorical Columns:



```
#Bivariant Analysis
import matplotlib.pyplot as plt

plt.figure(figsize= (12,8))
plt.subplot(2,4,1)
sns.barplot(x="season", y ="count", data =df)

plt.subplot(2,4,2)
sns.barplot(x="holiday", y ="count",data =df)

plt.subplot(2,4,3)
sns.barplot(x="workingday", y ="count",data =df)

plt.subplot(2,4,4)
sns.barplot(x="weather", y ="count",data =df)

plt.subplot(2,4,5)
sns.barplot(x="temp_bins", y ="count",data =df)

plt.subplot(2,4,6)
sns.barplot(x="atemp_bins", y ="count",data =df)

plt.subplot(2,4,7)
sns.barplot(x="humidity_bins", y ="count",data =df)

plt.subplot(2,4,8)
sns.barplot(x="windspeed_bins", y ="count",data =df)
```



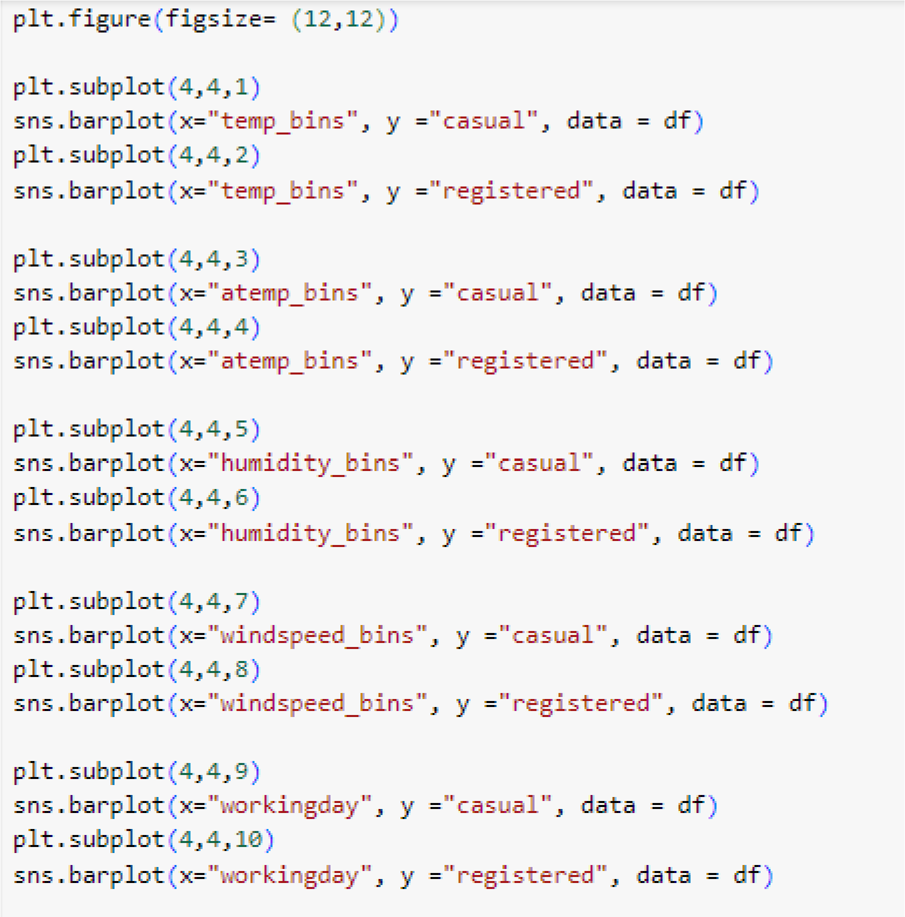
Activate Windows



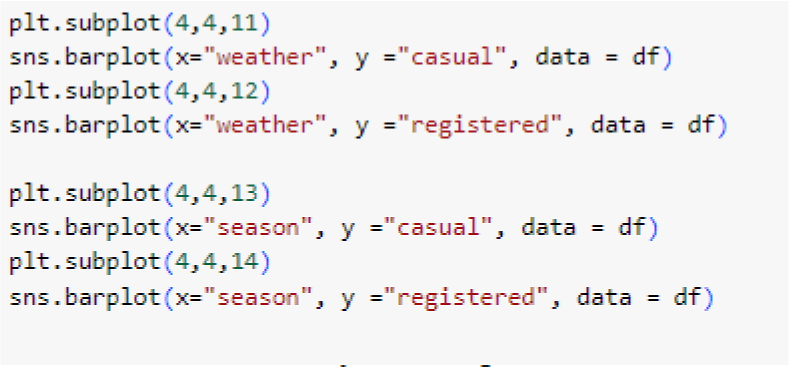
Insights:

- Bike usage remains consistent regardless of whether it's a working day or a holiday. The data shows no discernible difference in bike usage between these two types of days, with similar values recorded across both categories.
- Weather conditions significantly influence bike usage patterns. Days characterized by Light Snow, Light Rain accompanied by Thunderstorms, or Scattered Clouds tend to exhibit the lowest bike usage rates. Conversely, when the weather is Clear, with Few Clouds or Partly Cloudy conditions prevailing, bike usage peaks.
- Bike usage tends to decrease during the spring and winter seasons. Conversely, the highest levels of bike usage are observed during the summer and fall seasons.
- Low bike usage is typically observed when both the temperature and the perceived temperature (atemp) are low. Conversely, the highest levels of bike usage are recorded when both the temperature and perceived temperature are high. Therefore, there is a positive correlation between temperature and bike usage.
- Bike usage peaks when humidity levels are low, while it decreases when humidity is high. Thus, there exists a negative correlation between bike usage and humidity.
- Bike usage reaches its peak when the windspeed falls within the medium range. Conversely, bike usage tends to be lower when windspeed is either low or high. Notably, there is no discernible correlation between windspeed and bike usage.

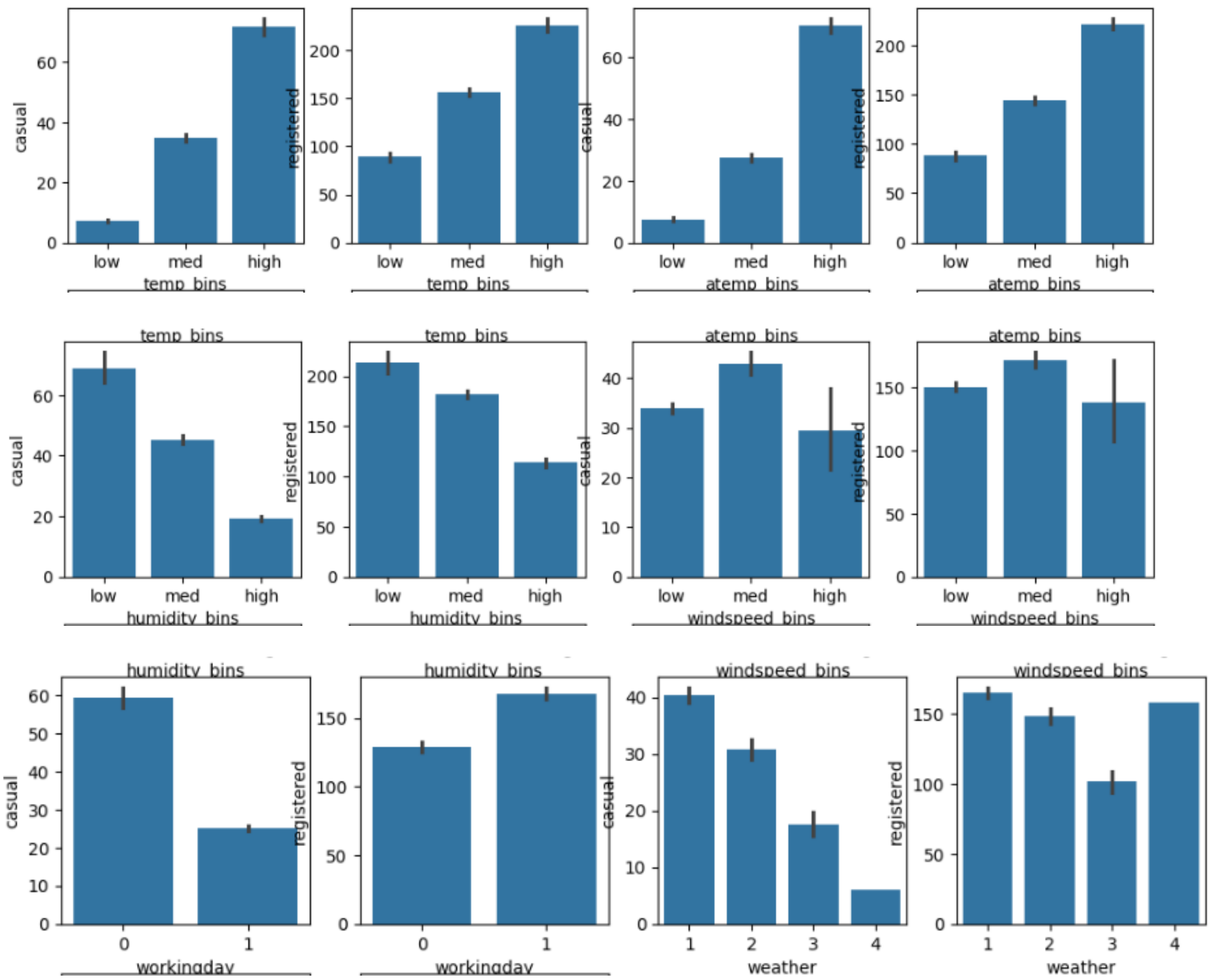
Bivariant Analysis of casual, registered vs categorical column:

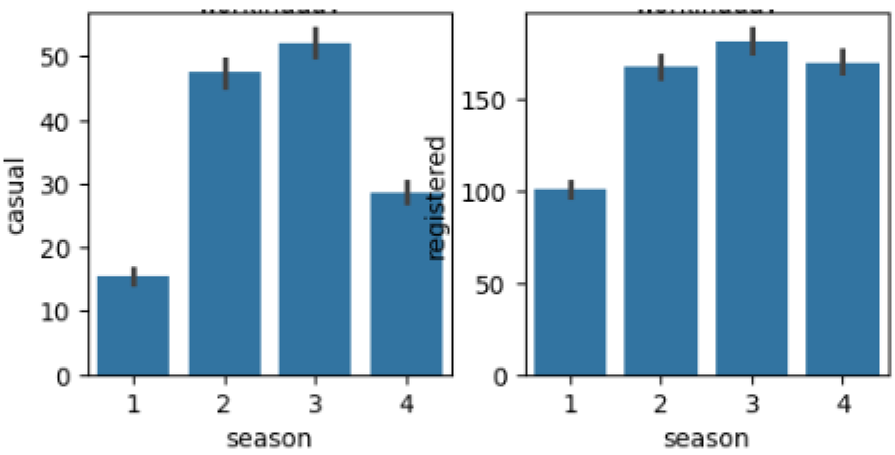


```
plt.figure(figsize= (12,12))

plt.subplot(4,4,1)
sns.barplot(x="temp_bins", y ="casual", data = df)
plt.subplot(4,4,2)
sns.barplot(x="temp_bins", y ="registered", data = df)

plt.subplot(4,4,3)
sns.barplot(x="atemp_bins", y ="casual", data = df)
plt.subplot(4,4,4)
sns.barplot(x="atemp_bins", y ="registered", data = df)

plt.subplot(4,4,5)
sns.barplot(x="humidity_bins", y ="casual", data = df)
plt.subplot(4,4,6)
sns.barplot(x="humidity_bins", y ="registered", data = df)

plt.subplot(4,4,7)
sns.barplot(x="windspeed_bins", y ="casual", data = df)
plt.subplot(4,4,8)
sns.barplot(x="windspeed_bins", y ="registered", data = df)

plt.subplot(4,4,9)
sns.barplot(x="workingday", y ="casual", data = df)
plt.subplot(4,4,10)
sns.barplot(x="workingday", y ="registered", data = df)
```



```
plt.subplot(4,4,11)
sns.barplot(x="weather", y ="casual", data = df)
plt.subplot(4,4,12)
sns.barplot(x="weather", y ="registered", data = df)

plt.subplot(4,4,13)
sns.barplot(x="season", y ="casual", data = df)
plt.subplot(4,4,14)
sns.barplot(x="season", y ="registered", data = df)
```


casual
registered
low
med
high
temp_bins
atemp_bins
humidity_bins
windspeed_bins
workingday
weather
0
1
2
3
4
0
20
40
60
50
100
150
200
10
30


casual
registered
season
1
2
3
4
0
10
20
30
40
50
0
50
100
150

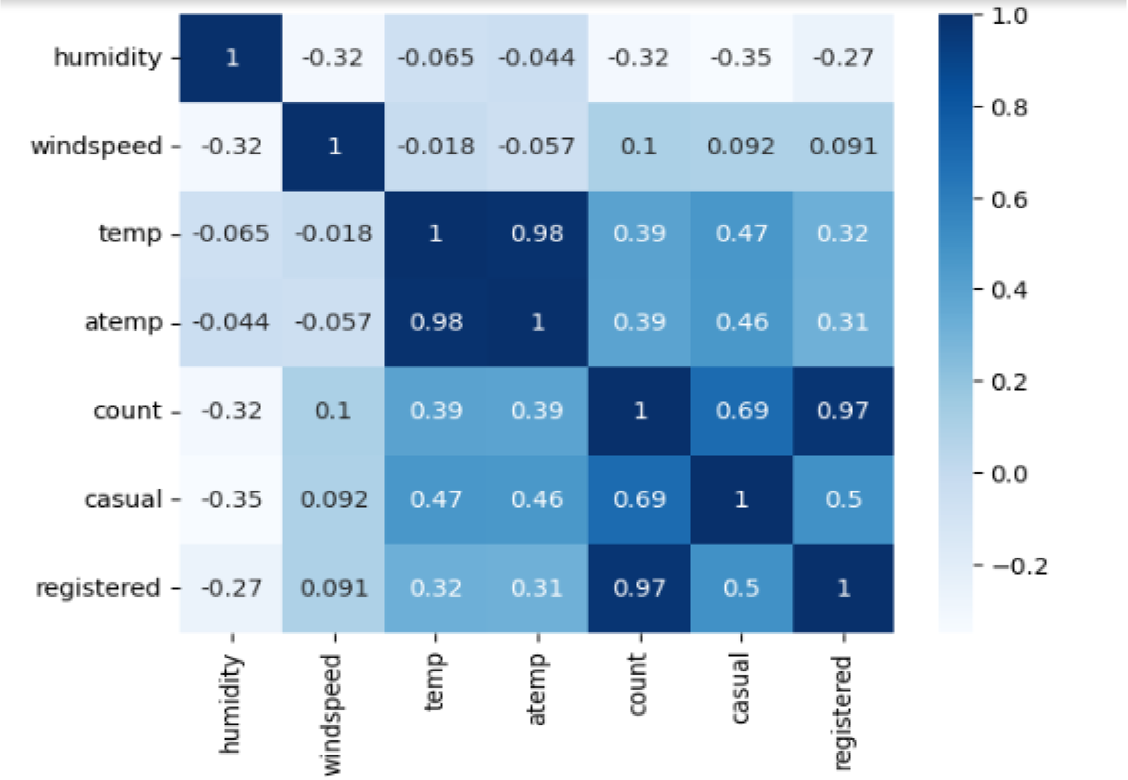
- In comparison to casual users, registered users consistently exhibit higher bike usage levels regardless of holidays, weather conditions, seasons, temperature, humidity, or windspeed.
- Both registered and casual users exhibit peak bike usage when temperatures are high or medium. Conversely, bike usage decreases significantly when temperatures are low for both user categories.
- High bike usage is observed among both casual and registered users during periods of low humidity. Conversely, when humidity levels are high, bike usage decreases across both user categories.
- Moderate windspeed levels coincide with increased bike usage for both casual and registered users. Conversely, during periods of low or high windspeed, bike usage decreases across both user categories.
- Differences in bike usage between working days and holidays are evident across both casual and registered user categories. Casual users exhibit lower bike usage on holidays compared to working days. In contrast, registered users show heightened bike usage during holidays, while their working day usage remains lower. Overall, registered users demonstrate higher bike usage on both working days and holidays compared to casual users.
- Casual users peak in fall, followed by summer, spring, and winter in descending order. Registered users show a peak in fall, then winter, and summer, with spring recording the least usage. Nevertheless, registered users consistently maintain higher mean usage across all four seasons compared to casual users.

5. Correlation:

We are trying to understand the correlation between the numerical data independent Vs dependent.

```
sns.heatmap(df[[ "humidity","windspeed", "temp", "atemp", "count",
"casual", "registered"]].corr(), annot = True, cmap ="Blues")
```

.



- Humidity and bike user count has a correlation value of -0.32 indicates a moderate negative correlation.
- Temperature and Bike usage count has correlation value of 0.39 suggests a moderate positive correlation.
- Windspped and bike usage count has a correlation value of 0.1 indicates a weak positive correlation.

6. Hypothesis Testing:

1.No. of bike rides on weekdays and weekends:

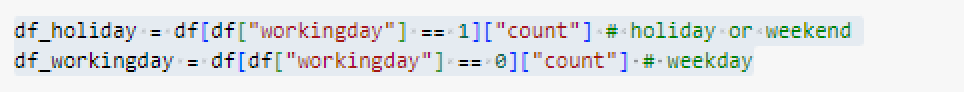
Step 1:

Ho: There is no significant difference between the no. of bike rides on weekdays and weekends.

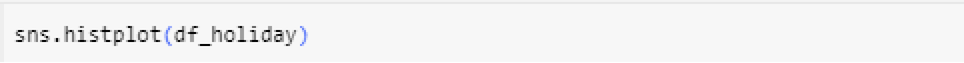
Ha: There is significant difference between the no. of bike rides on weekdays and weekends.

Step 2:

Distribution is not normal.

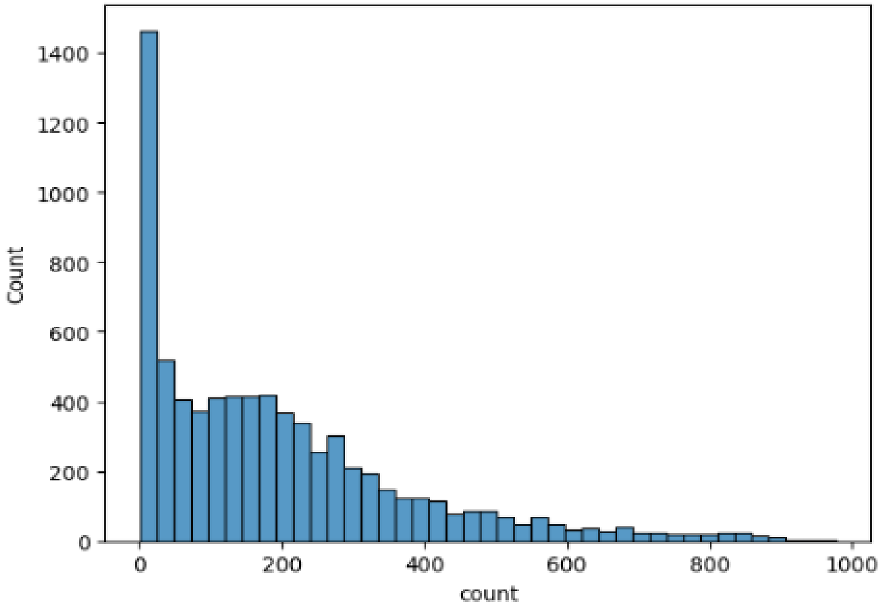


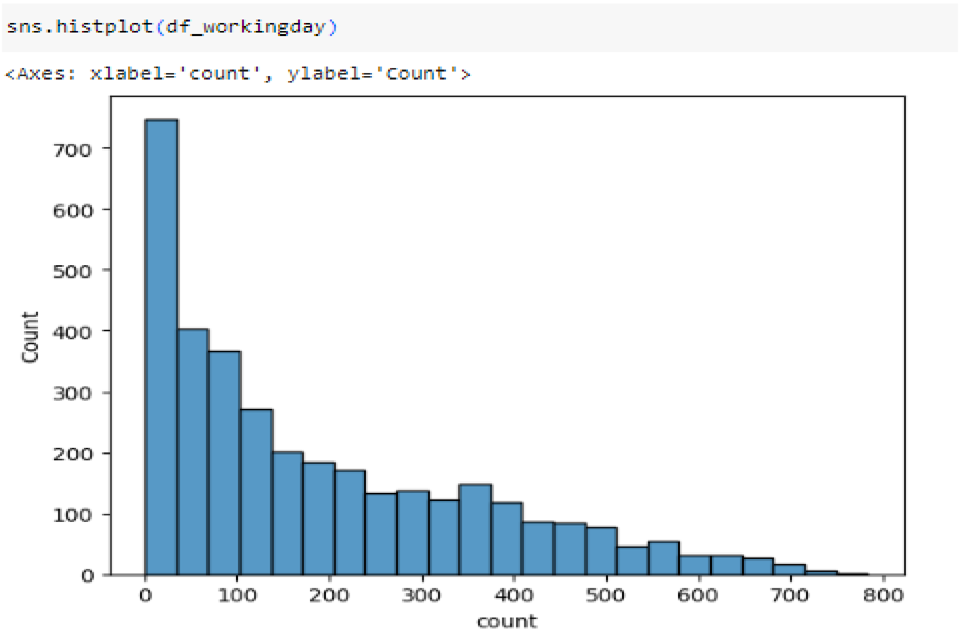
```
df_holiday = df[df["workingday"] == 1]["count"] # holiday or weekend
df_workingday = df[df["workingday"] == 0]["count"] # weekday
```



```
sns.histplot(df_holiday)
```

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





```
sns.histplot(df_workingday)
```

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

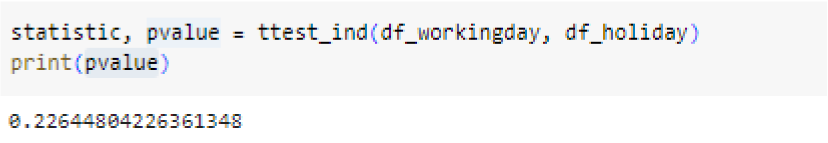
Step 3:

Significant level (alpha) = 0.05

Which means confidence level is 95%. We are accepting 5% error in this testing.

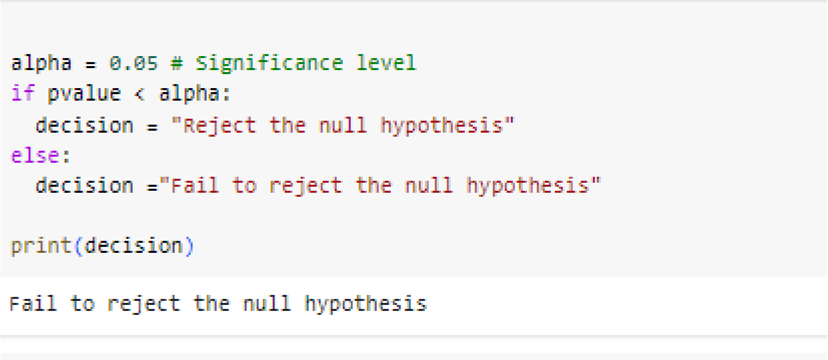
Step 4:

a) Find the Pvalue for count(casual + registered) and compare it with alpha



```
statistic, pvalue = ttest_ind(df_workingday, df_holiday)
print(pvalue)
```

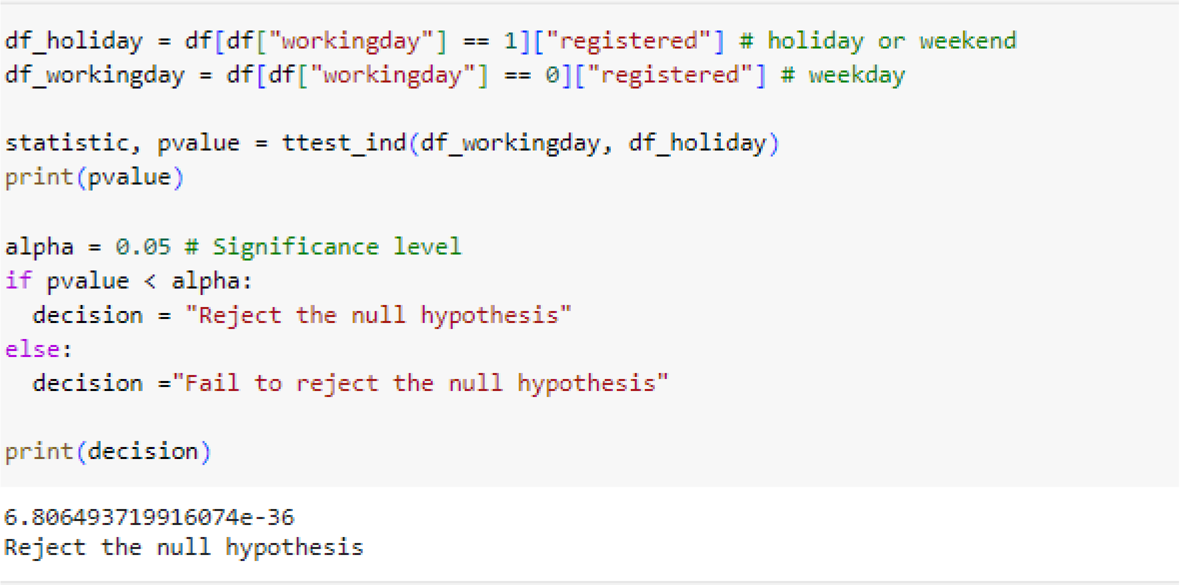
```
0.22644804226361348
```



```
alpha = 0.05 # Significance level
if pvalue < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"

print(decision)
```

```
Fail to reject the null hypothesis
```

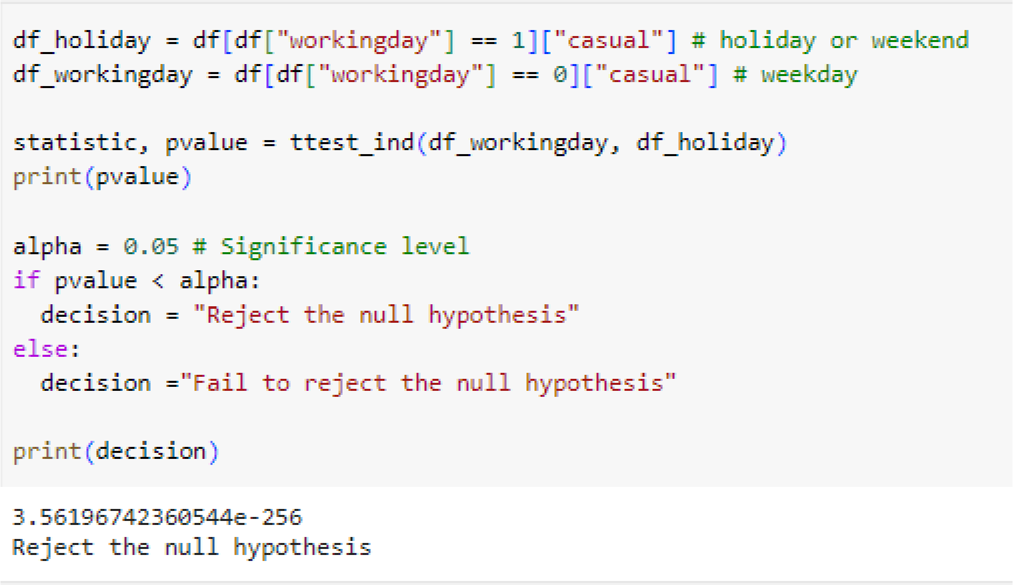
b) Find the Pvalue for registered and compare it with alpha



```
df_holiday = df[df["workingday"] == 1]["registered"] # holiday or weekend
df_workingday = df[df["workingday"] == 0]["registered"] # weekday

statistic, pvalue = ttest_ind(df_workingday, df_holiday)
print(pvalue)

alpha = 0.05 # Significance level
if pvalue < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"

print(decision)
```

```
6.806493719916074e-36
Reject the null hypothesis
```

c) Find the Pvalue for casual and compare it with alpha



```
df_holiday = df[df["workingday"] == 1]["casual"] # holiday or weekend
df_workingday = df[df["workingday"] == 0]["casual"] # weekday

statistic, pvalue = ttest_ind(df_workingday, df_holiday)
print(pvalue)

alpha = 0.05 # Significance level
if pvalue < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"

print(decision)
```

```
3.561967423605440e-256
Reject the null hypothesis
```

Inference:

➢ When conducting a t-test on the combined total count of casual and registered users, the resulting p-value was found to be greater than the chosen significance level (alpha). This suggests that there is no statistically significant difference between bike usage on working days and holidays when considering both types of users together.

- However, upon separately analyzing casual users and registered users, the p-value was found to be less than the chosen alpha level. This indicates that there is a significant difference in bike usage between working days and holidays for these two user groups.
- In essence, while the total count analysis didn't reveal a significant difference, further examination focusing on individual user types uncovered a notable distinction in usage patterns between working days and holidays.

Recommendation:

- To cater to both casual and registered users effectively, the company should devise distinct strategies aligned with their usage patterns.
- For casual users who utilize bikes most on working days, the company could introduce weekday-specific promotions and incentives.
- Meanwhile, for registered users who show peak usage during holidays, holiday-themed offers and extended rental options would be more appealing.
- By tailoring plans to match these usage trends, the company can attract and retain both user groups more effectively.

2. Demand of bicycles on rent Vs Different Weather conditions:

Step 1:

Ho: There is no significant difference in bike demand across different weather conditions.

Ha: There is significant difference in bike demand across different weather conditions.

Step 2:

Test : ANOVA

The conditions for ANOVA:

1. Data should be normally distributed.

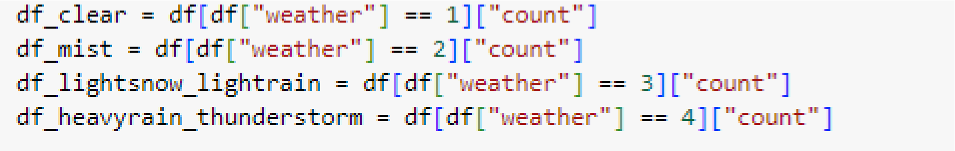
2. Data should be independent across each record.

3. Equal variance in different groups

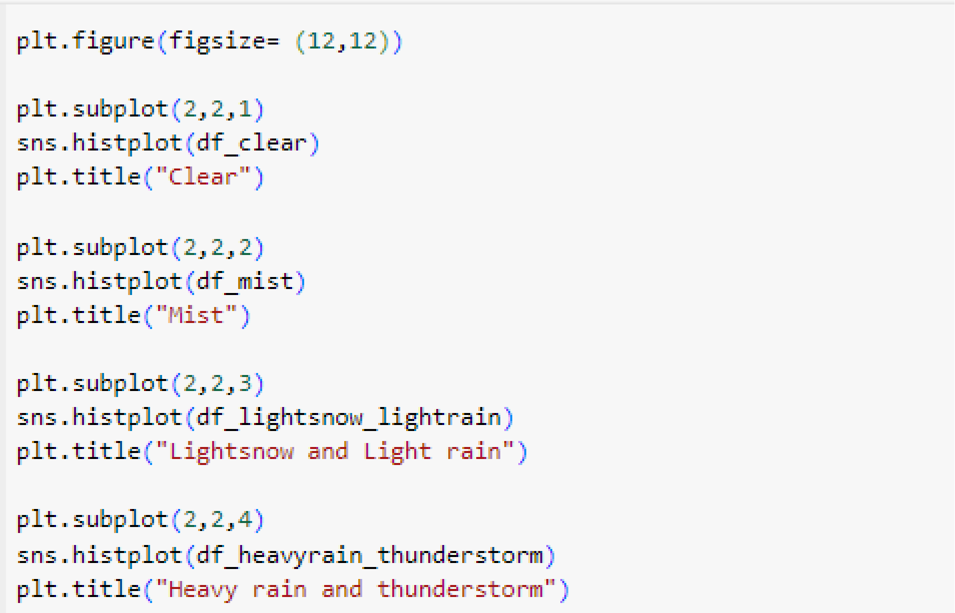
To check whether the data is normally distributed or not we can use Histplot, QQ plot, Shapiro-Wilk test.

Test to check Normality:

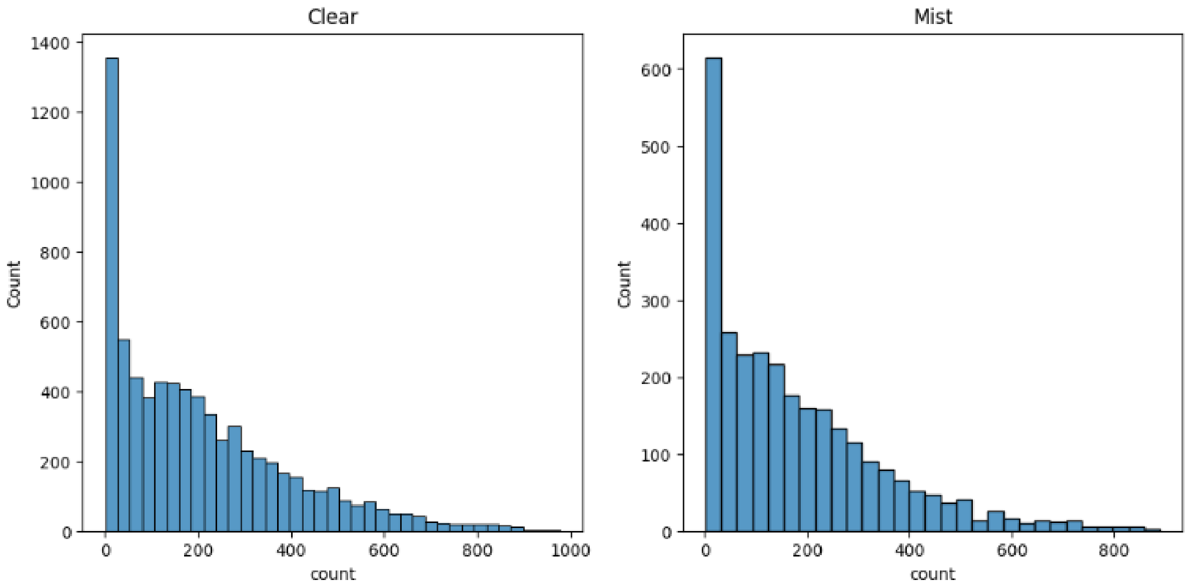
a) Histogram:

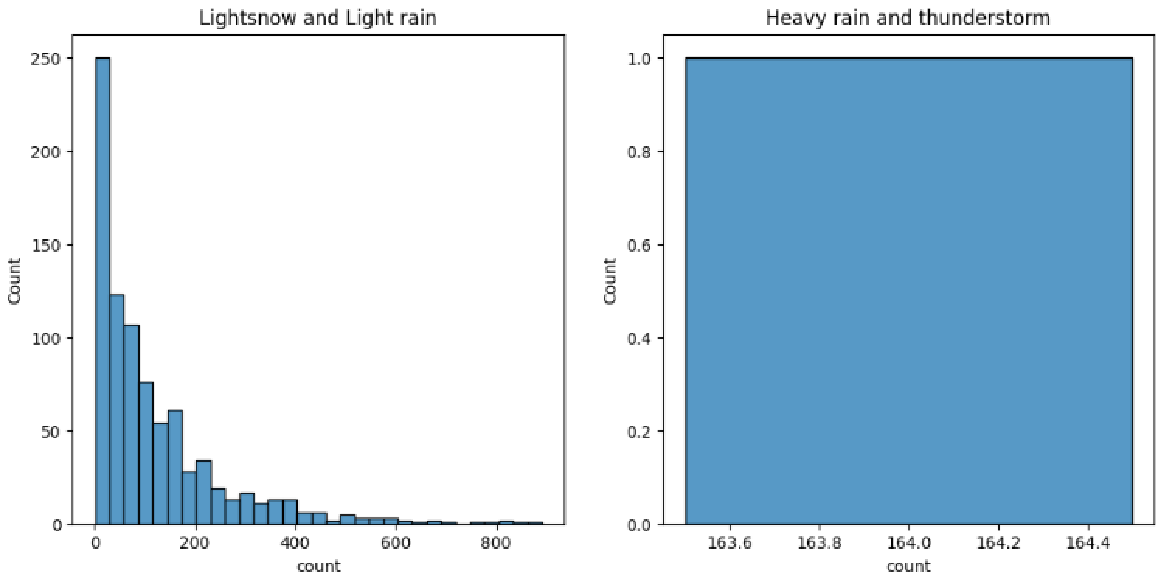


```
df_clear = df[df["weather"] == 1]["count"]
df_mist = df[df["weather"] == 2]["count"]
df_lightsnow_lightrain = df[df["weather"] == 3]["count"]
df_heavyrain_thunderstorm = df[df["weather"] == 4]["count"]
```



```
plt.figure(figsize= (12,12))

plt.subplot(2,2,1)
sns.histplot(df_clear)
plt.title("Clear")

plt.subplot(2,2,2)
sns.histplot(df_mist)
plt.title("Mist")

plt.subplot(2,2,3)
sns.histplot(df_lightsnow_lightrain)
plt.title("Lightsnow and Light rain")

plt.subplot(2,2,4)
sns.histplot(df_heavyrain_thunderstorm)
plt.title("Heavy rain and thunderstorm")
```

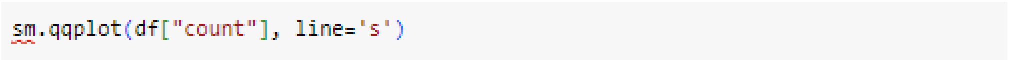




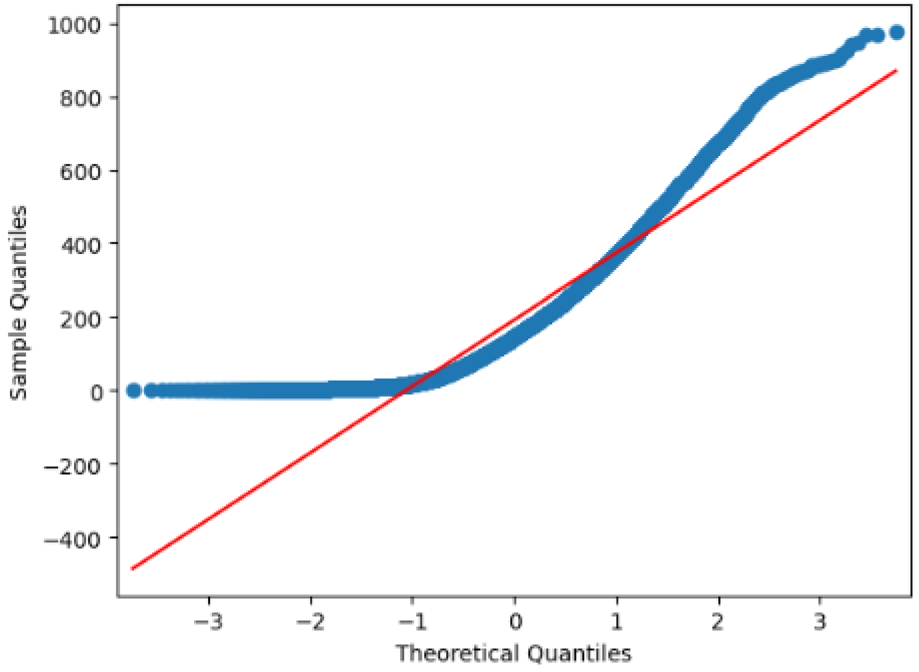
Insights

The Data is not normally distributed.

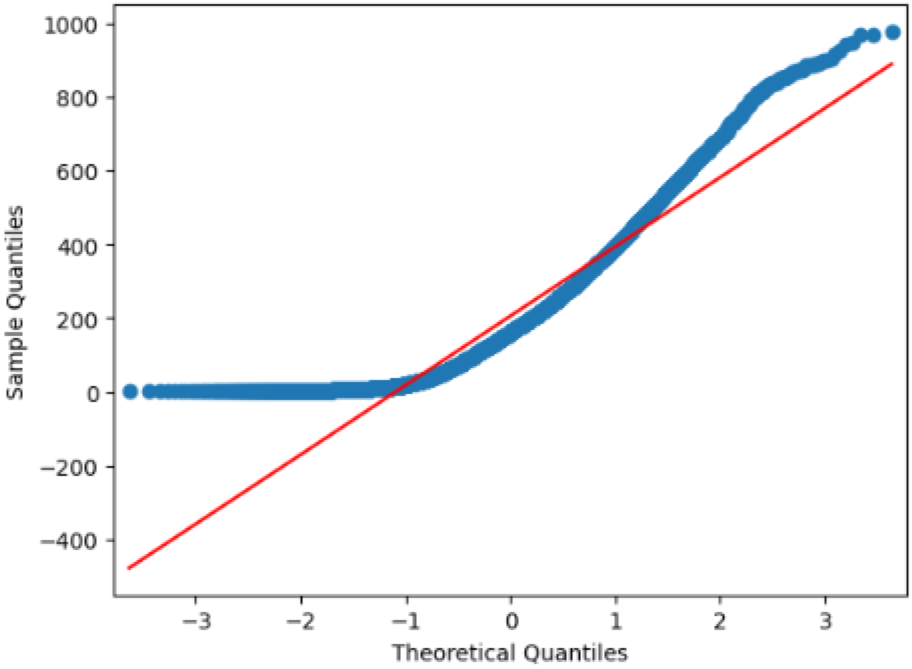
b) QQ Plot:



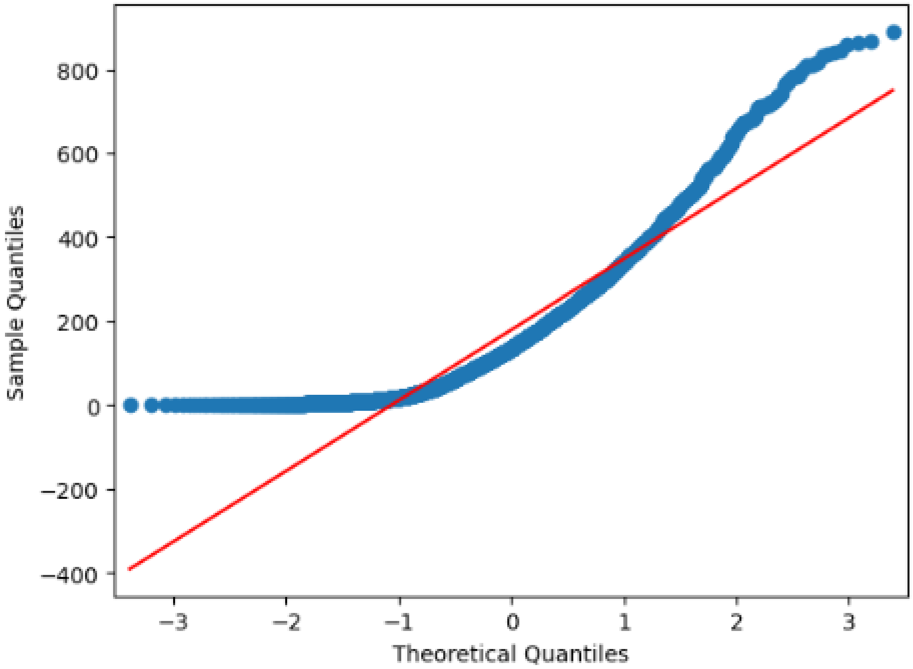
```
sm.qqplot(df["count"], line='s')
```

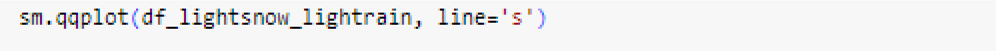


```
sm.qqplot(df_clear , line='s')
```

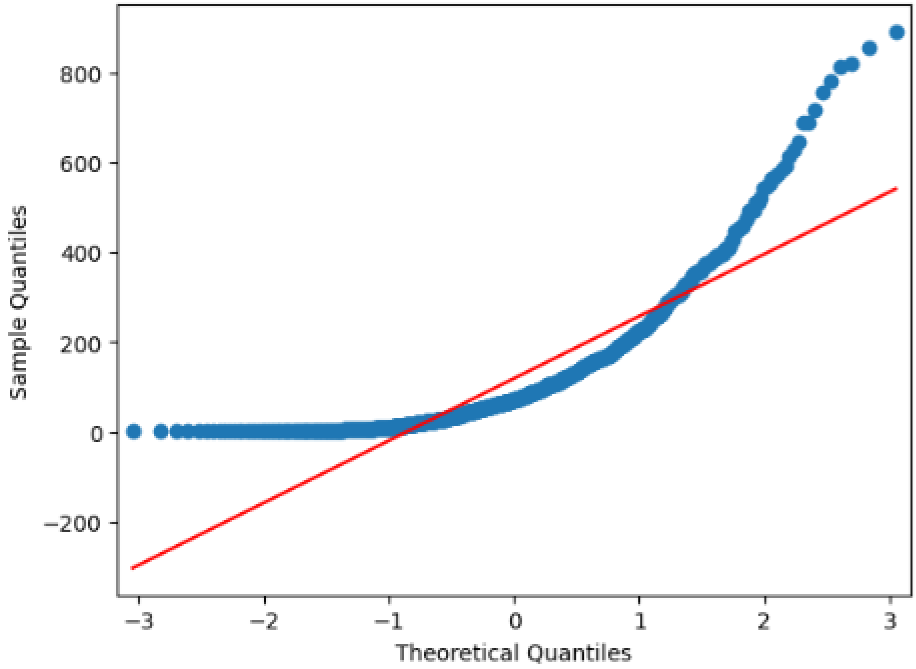


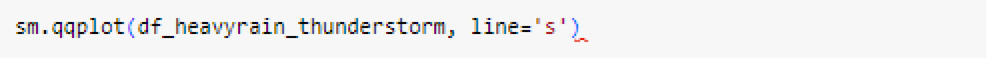
```
sm.qqplot(df_mist, line='s')
```



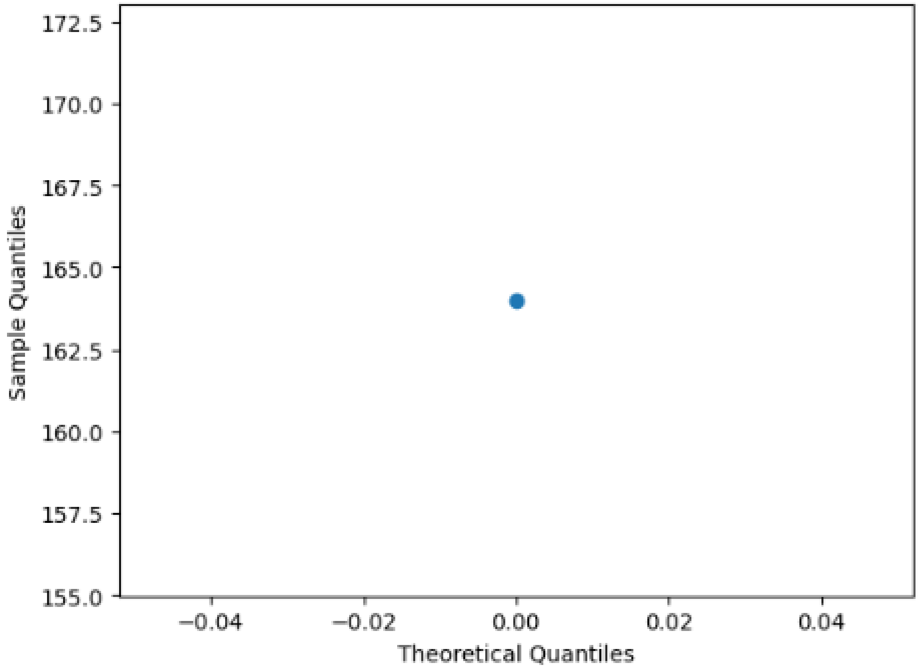


```
sm.qqplot(df_lightsnow_lightrain, line='s')
```





```
sm.qqplot(df_heavyrain_thunderstorm, line='s')
```

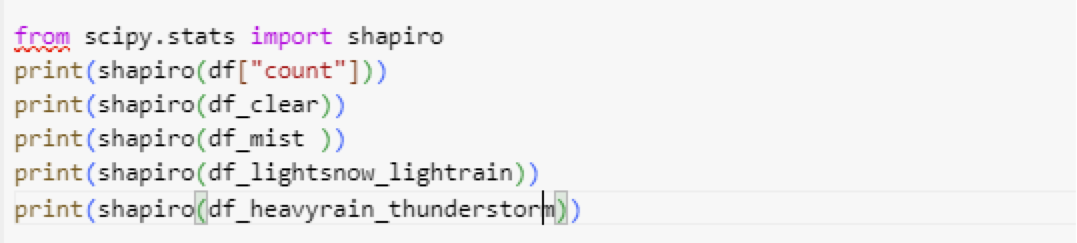


<u>Insights</u>

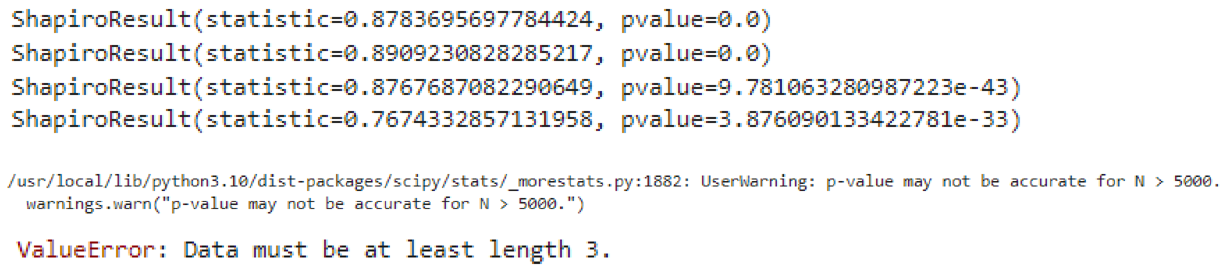
The QQ plot helps to understand whether the data follows normal distribution or not. The plot clearly shows that the data is not normally distributed

c) Shapiro –Wilk test:

This is a statistical test. If the pvalue is greater than 0.05 the data follows normal distribution. Conversely, if the pvalue is low, the data distribution is significantly different from normal distribution.



```
from scipy.stats import shapiro
print(shapiro(df["count"]))
print(shapiro(df_clear))
print(shapiro(df_mist ))
print(shapiro(df_lightsnow_lightrain))
print(shapiro(df_heavyrain_thunderstorm))
```



```
ShapiroResult(statistic=0.8783695697784424, pvalue=0.0)
ShapiroResult(statistic=0.8909230828285217, pvalue=0.0)
ShapiroResult(statistic=0.8767687082290649, pvalue=9.781063280987223e-43)
ShapiroResult(statistic=0.7674332857131958, pvalue=3.876090133422781e-33)

/usr/local/lib/python3.10/dist-packages/scipy/stats/_morestats.py:1882: UserWarning: p-value may not be accurate for N > 5000.
  warnings.warn("p-value may not be accurate for N > 5000.")

ValueError: Data must be at least length 3.
```

Insights

- The data analysis reveals that the p-values for mist and light rain/light snow conditions are significantly low, indicating non-normal distributions.
- However, the sample size for heavy rain/thunderstorm is insufficient for p-value calculation due to fewer than three data points.
- As for the overall count and clear conditions, the sample size exceeds 5000, making p-value calculation impractical.
- The data is not Gaussian /Normal.

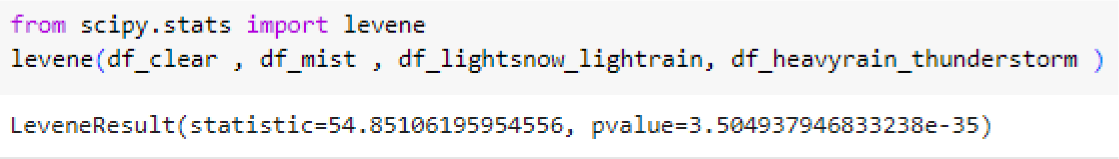
Test to find whether different groups have equal Variance:

LeveneTest:

The pvalue from the levene test, tells us whether thevariances across the groups are statistically similar or if there are significant differences

Here we are taking alpha = 0.05

If the pvalue is high (>0.05) it implies that the variance are relatively equal across the groups. On the other hand if the pvalue is low (<0.05) variance significantly differ across the group.



```
from scipy.stats import levene
levene(df_clear , df_mist , df_lightsnow_lightrain, df_heavyrain_thunderstorm )
```

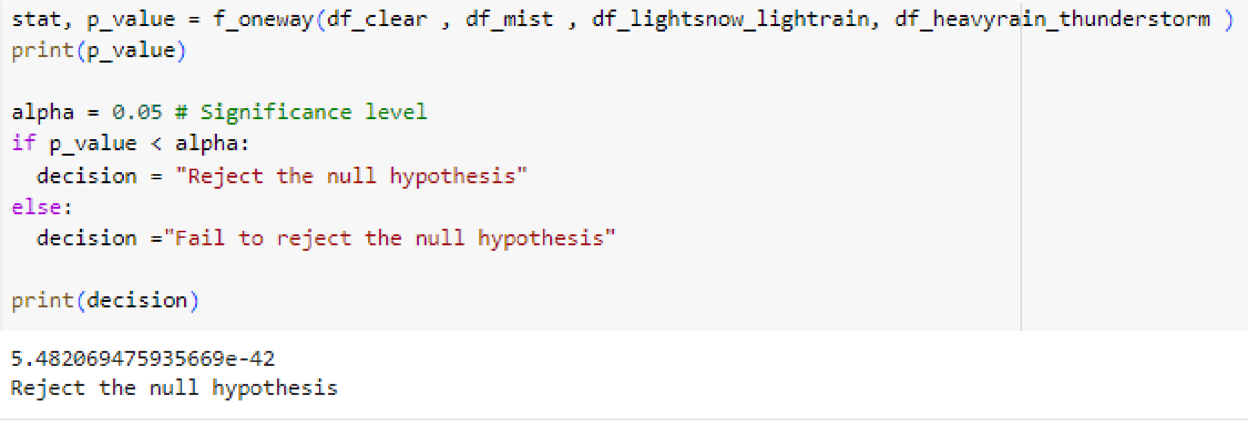
```
LeveneResult(statistic=54.85106195954556, pvalue=3.504937946833238e-35)
```

The pvalue is <0.05, so the variance across different groups are not same.

Step 3:

Lets conduct both ANOVA and Kruskal test to find whether there is significant difference is the bike usage due to weather.

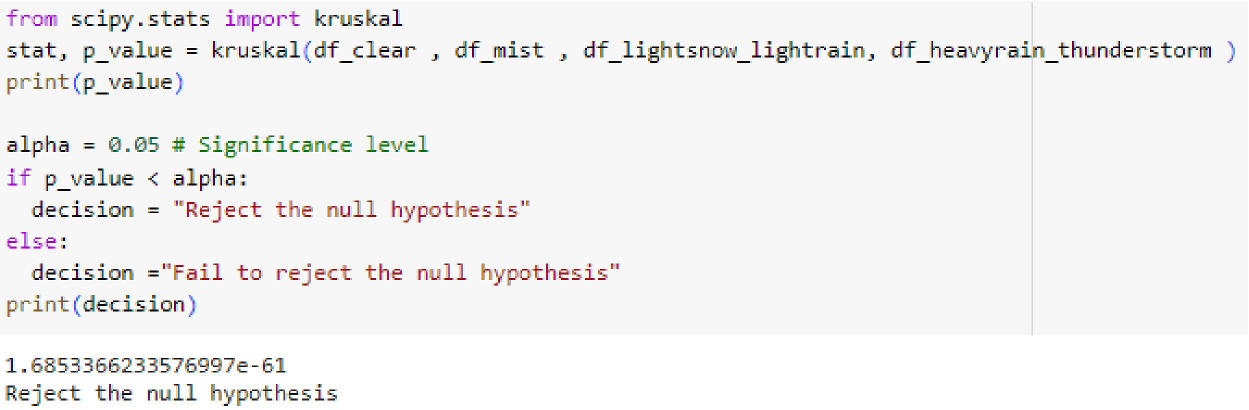
ANOVA on count:



```
stat, p_value = f_oneway(df_clear , df_mist , df_lightsnow_lightrain, df_heavyrain_thunderstorm )
print(p_value)

alpha = 0.05 # Significance level
if p_value < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"

print(decision)
```

```
5.482069475935669e-42
Reject the null hypothesis
```

Kruskal on count:

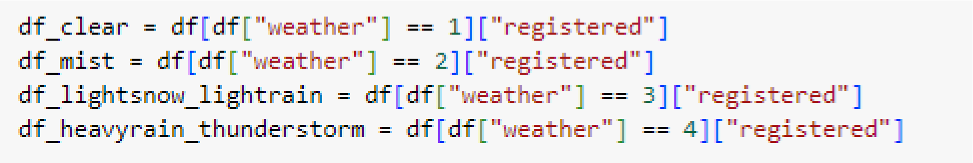


```
from scipy.stats import kruskal
stat, p_value = kruskal(df_clear , df_mist , df_lightsnow_lightrain, df_heavyrain_thunderstorm )
print(p_value)

alpha = 0.05 # Significance level
if p_value < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"
print(decision)
```

```
1.68533662335769970e-61
Reject the null hypothesis
```

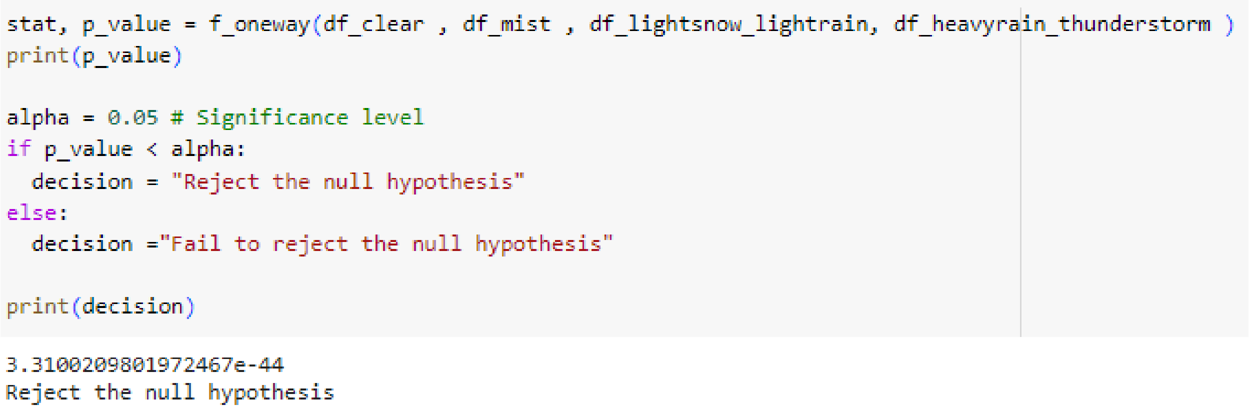
Insights:

In both the test pvalue is very less, which implies that there is significant difference on bike usage under different weather condition.



```
df_clear = df[df["weather"] == 1]["registered"]
df_mist = df[df["weather"] == 2]["registered"]
df_lightsnow_lightrain = df[df["weather"] == 3]["registered"]
df_heavyrain_thunderstorm = df[df["weather"] == 4]["registered"]
```

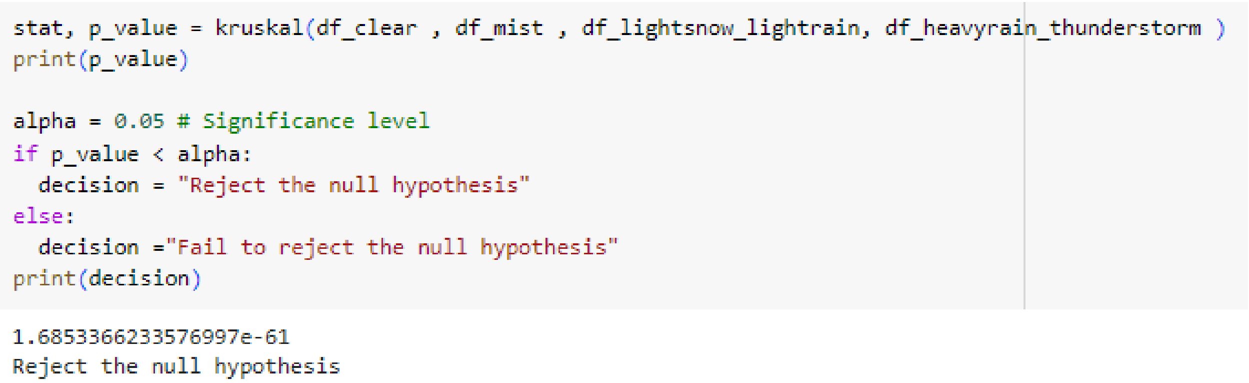
ANOVA on registered:



```
stat, p_value = f_oneway(df_clear , df_mist , df_lightsnow_lightrain, df_heavyrain_thunderstorm )
print(p_value)

alpha = 0.05 # Significance level
if p_value < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"

print(decision)
```

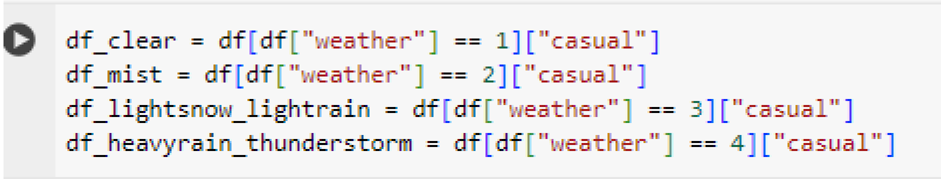
```
3.3100209801972467e-44
Reject the null hypothesis
```

Kruskal on registered users:



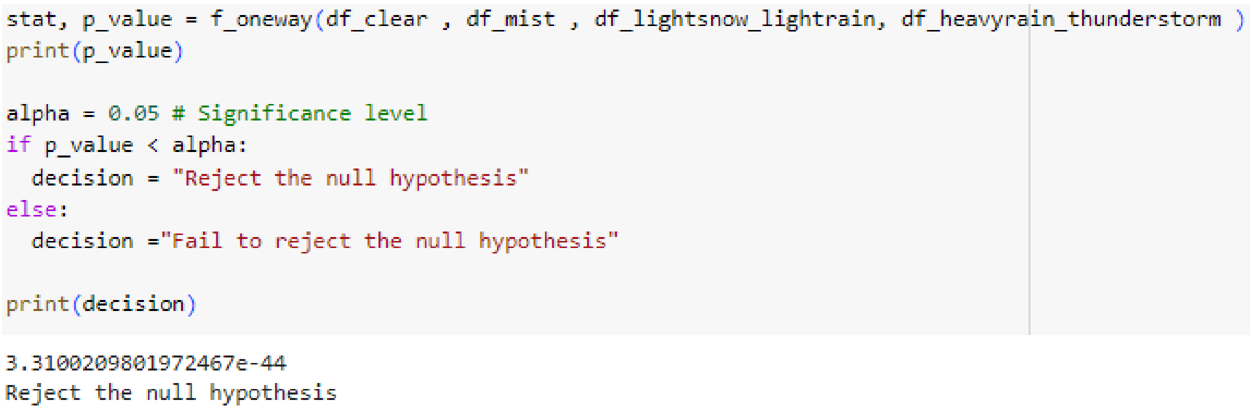
```
stat, p_value = kruskal(df_clear , df_mist , df_lightsnow_lightrain, df_heavyrain_thunderstorm )
print(p_value)

alpha = 0.05 # Significance level
if p_value < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"
print(decision)
```

```
1.68533662335769970e-61
Reject the null hypothesis
```



```
df_clear = df[df["weather"] == 1]["casual"]
df_mist = df[df["weather"] == 2]["casual"]
df_lightsnow_lightrain = df[df["weather"] == 3]["casual"]
df_heavyrain_thunderstorm = df[df["weather"] == 4]["casual"]
```

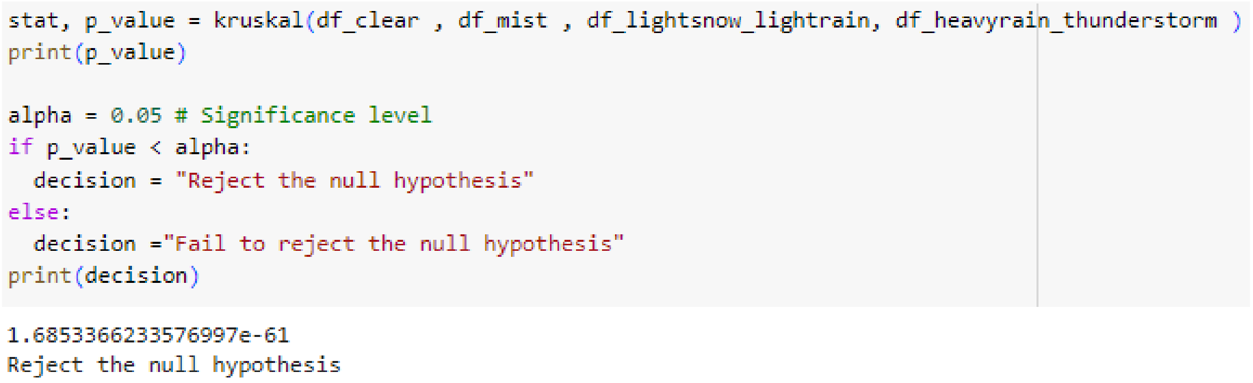
ANOVA on Casual users:



```
stat, p_value = f_oneway(df_clear , df_mist , df_lightsnow_lightrain, df_heavyrain_thunderstorm )
print(p_value)

alpha = 0.05 # Significance level
if p_value < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"

print(decision)
```

```
3.3100209801972467e-44
Reject the null hypothesis
```

Kruskal on Casual users:



```
stat, p_value = kruskal(df_clear , df_mist , df_lightsnow_lightrain, df_heavyrain_thunderstorm )
print(p_value)

alpha = 0.05 # Significance level
if p_value < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"
print(decision)
```

```
1.6853366233576997e-61
Reject the null hypothesis
```

Inference:

- Based on the results of both the ANOVA and Kruskal-Wallis tests conducted on Casual users, Registered users, and overall counts, it is evident that there exists a significant difference in bike usage across varying weather conditions.

Recommendations:

Based on the insight that there is a significant difference in bike usage across different weather conditions for both casual and registered users, the company can tailor its operational and marketing strategies accordingly. Here are some recommendations:

- Weather-Based Promotions
- Targeted Marketing Campaigns

3. Demand of bicycle on rent Vs Different Season:

Step 1:

Ho: Demand of bicycle on rent is same for different season

Ha: Demand of bicycle on rent is not same for different season

Step 2:

Test : ANOVA

The conditions for ANOVA:

1. Data should be normally distributed.

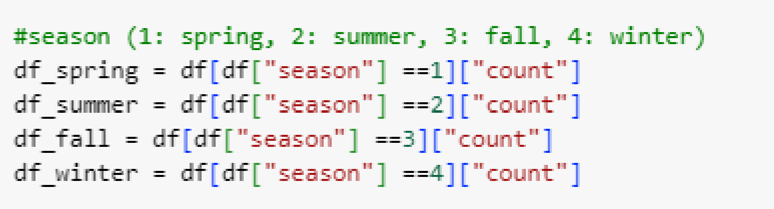
2. Data should be independent across each record.

3. Equal variance in different groups

To check whether the data is normally distributed or not we can use Histplot, QQ plot, Shapiro-Wilk test.

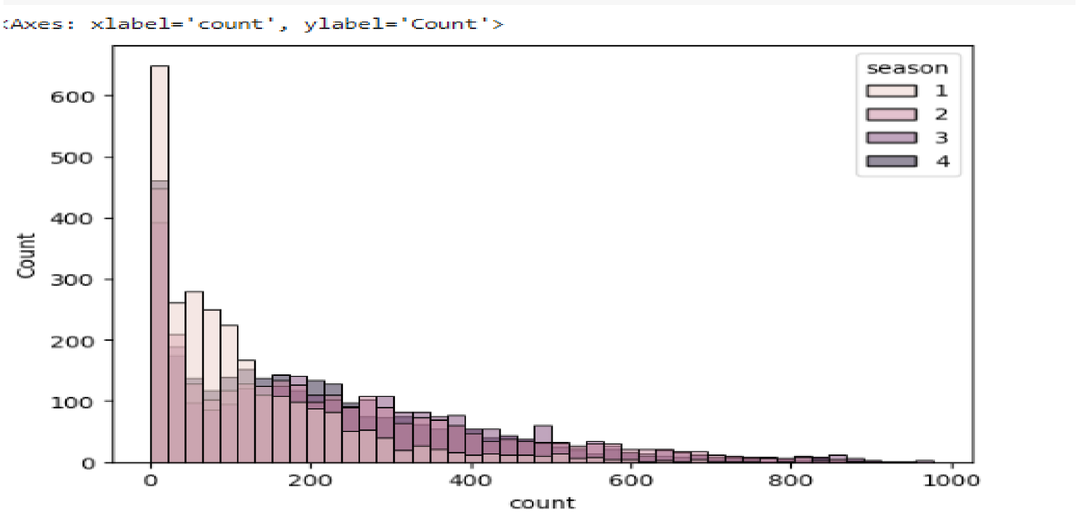
Test for Normality:

a) Histogram:



```
#season (1: spring, 2: summer, 3: fall, 4: winter)
df_spring = df[df["season"] ==1]["count"]
df_summer = df[df["season"] ==2]["count"]
df_fall = df[df["season"] ==3]["count"]
df_winter = df[df["season"] ==4]["count"]
```

```
sns.histplot(data = df, x = "count", hue = "season")
```

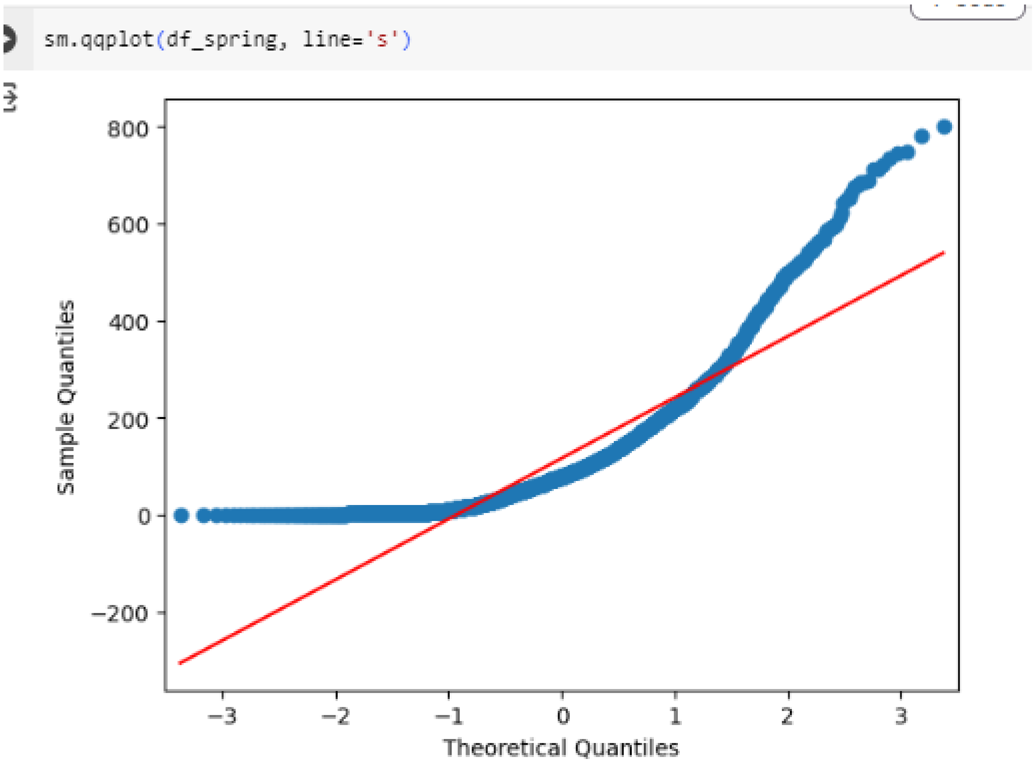


```
<Axes: xlabel='count', ylabel='Count'>
```

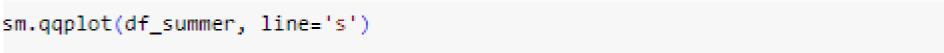
Insights:

Histrogram of different seasons shows that the data is not Gaussian.

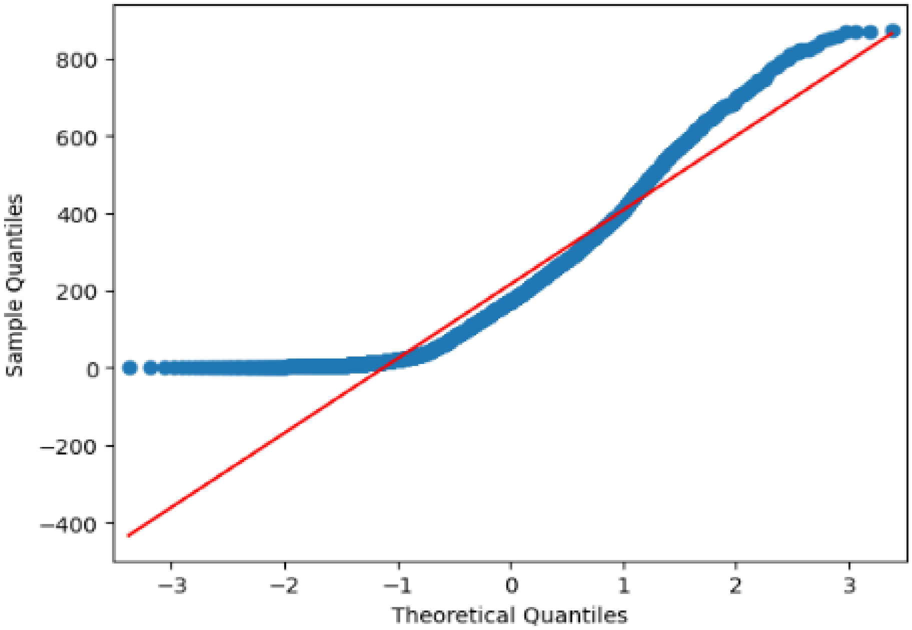
b) QQ Plot

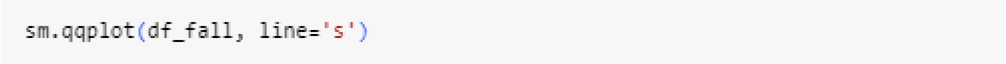


```
sm.qqplot(df_spring, line='s')
```

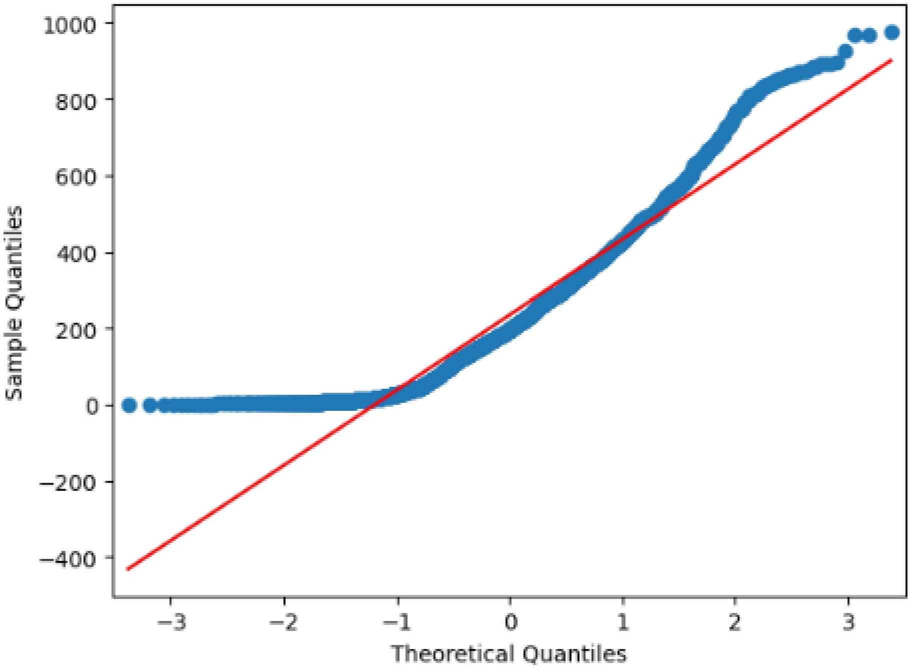


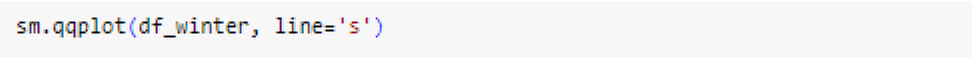
```
sm.qqplot(df_summer, line='s')
```



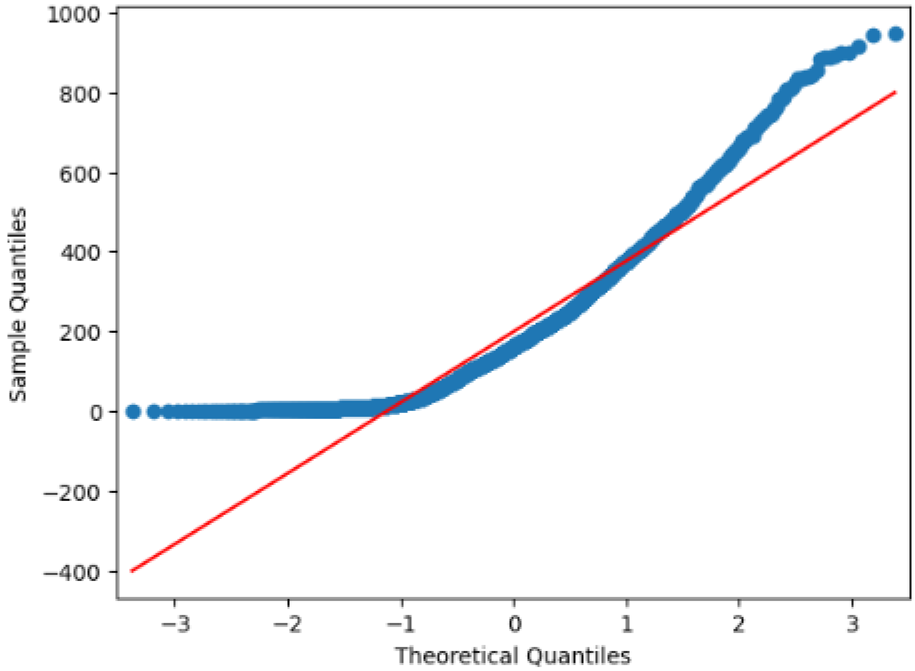


```
sm.qqplot(df_fall, line='s')
```





```
sm.qqplot(df_winter, line='s')
```

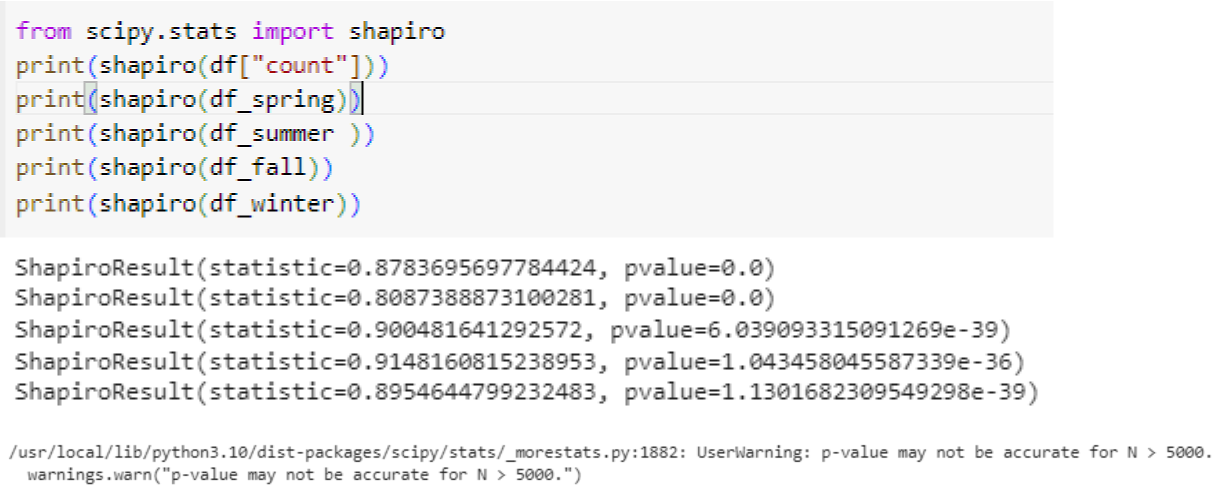


Insights:

The QQ plot shows that the data is not normal/ Gaussian.

c) Shapiro – Wilk Test:

This is a statistical test. If the pvalue is greater than 0.05 the data follows normal distribution. Conversely, if the pvalue is low, the data distribution is significantly different from normal distribution.



```
from scipy.stats import shapiro
print(shapiro(df["count"]))
print(shapiro(df_spring))
print(shapiro(df_summer ))
print(shapiro(df_fall))
print(shapiro(df_winter))
```

```
ShapiroResult(statistic=0.8783695697784424, pvalue=0.0)
ShapiroResult(statistic=0.8087388873100281, pvalue=0.0)
ShapiroResult(statistic=0.900481641292572, pvalue=6.039093315091269e-39)
ShapiroResult(statistic=0.9148160815238953, pvalue=1.043458045587339e-36)
ShapiroResult(statistic=0.8954644799232483, pvalue=1.1301682309549298e-39)
```

```
/usr/local/lib/python3.10/dist-packages/scipy/stats/_morestats.py:1882: UserWarning: p-value may not be accurate for N > 5000.
  warnings.warn("p-value may not be accurate for N > 5000.")
```

Insights:

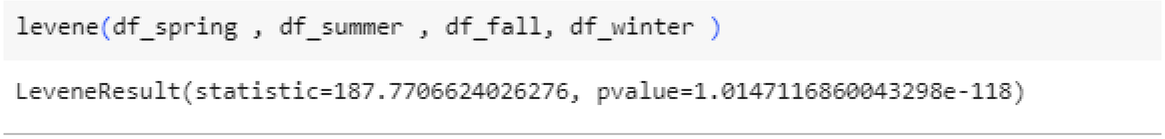
- ➢ The data analysis reveals that the p-values for summer, fall and winter seasons are significantly low, indicating non-normal distributions.
- ➢ As for the overall count and Spring, the sample size exceeds 5000, making p-value calculation impractical.
- ➢ The data is not Gaussian /Normal.

d) LeveneTest:

The pvalue from the levene test, tells us whether thevariances across the groups are statistically similar or if there are significant differences

Here we are taking alpha = 0.05

If the pvalue is high (>0.05) it implies that the variance are relatively equal across the groups. On the other hand if the pvalue is low (<0.05) variance significantly differ across the group.



```
levene(df_spring , df_summer , df_fall, df_winter )
```

```
LeveneResult(statistic=187.7706624026276, pvalue=1.0147116860043298e-118)
```

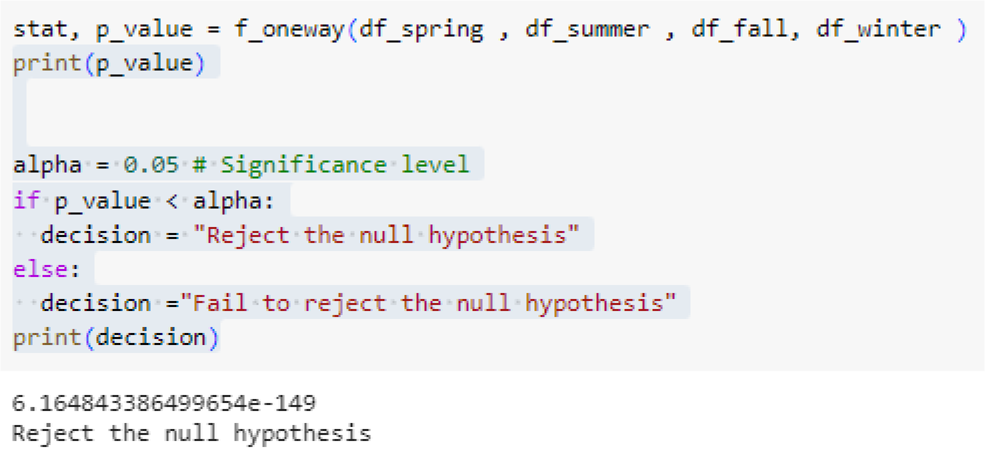
Insights:

The data groups doesnot have equal variance.

Step 3:

Significane level (alpha) = 0.05

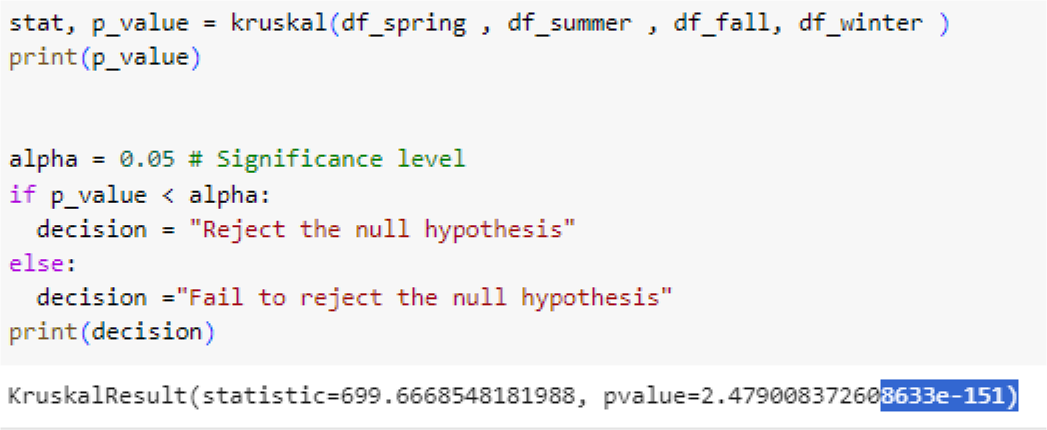
ANOVA on overall count:



```
stat, p_value = f_oneway(df_spring , df_summer , df_fall, df_winter )
print(p_value)


alpha = 0.05 # Significance level
if p_value < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"
print(decision)
```

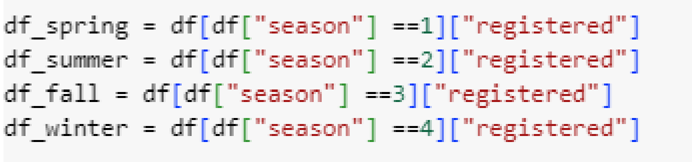
```
6.164843386499654e-149
Reject the null hypothesis
```

Kruskal on overall count:



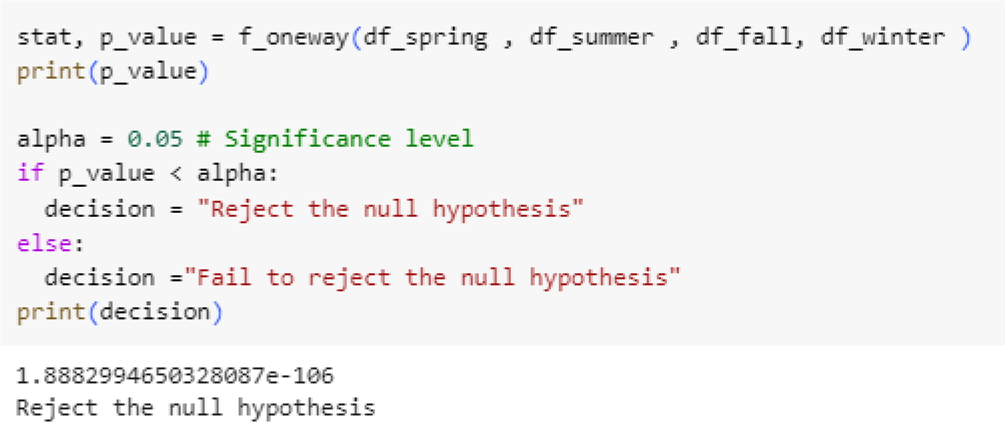
```
stat, p_value = kruskal(df_spring , df_summer , df_fall, df_winter )
print(p_value)


alpha = 0.05 # Significance level
if p_value < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"
print(decision)
```

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



```
df_spring = df[df["season"] ==1]["registered"]
df_summer = df[df["season"] ==2]["registered"]
df_fall = df[df["season"] ==3]["registered"]
df_winter = df[df["season"] ==4]["registered"]
```

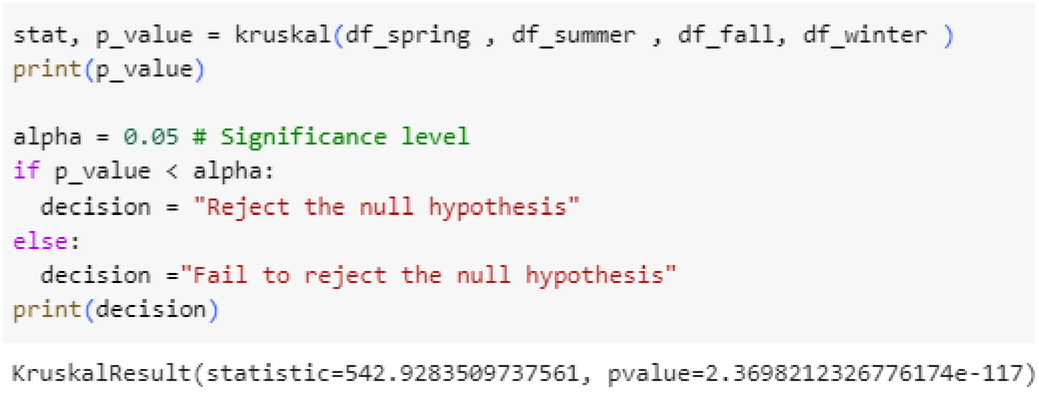
ANOVA on Registered users:



```
stat, p_value = f_oneway(df_spring , df_summer , df_fall, df_winter )
print(p_value)

alpha = 0.05 # Significance level
if p_value < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"
print(decision)
```

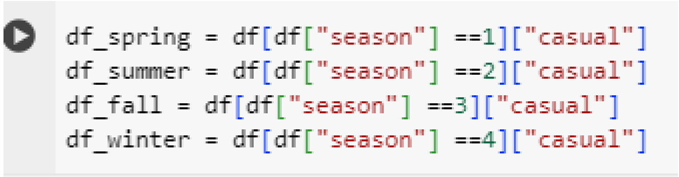
```
1.8882994650328087e-106
Reject the null hypothesis
```

Kruskal on registered users:



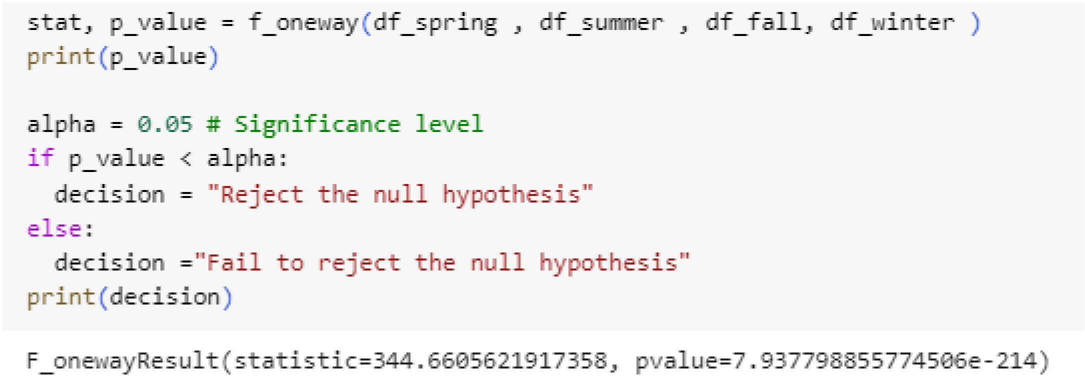
```
stat, p_value = kruskal(df_spring , df_summer , df_fall, df_winter )
print(p_value)

alpha = 0.05 # Significance level
if p_value < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"
print(decision)
```

```
KruskalResult(statistic=542.9283509737561, pvalue=2.3698212326776174e-117)
```



```
df_spring = df[df["season"] ==1]["casual"]
df_summer = df[df["season"] ==2]["casual"]
df_fall = df[df["season"] ==3]["casual"]
df_winter = df[df["season"] ==4]["casual"]
```

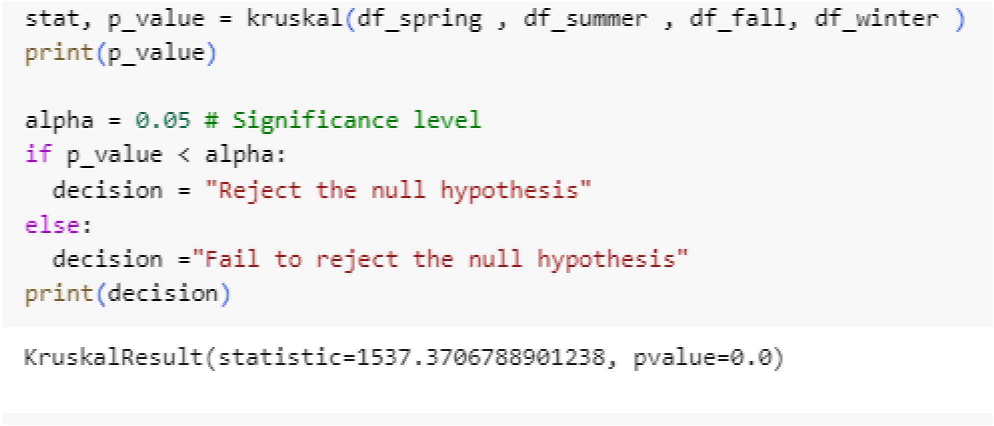
ANOVA on casual users:



```
stat, p_value = f_oneway(df_spring , df_summer , df_fall, df_winter )
print(p_value)

alpha = 0.05 # Significance level
if p_value < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"
print(decision)
```

```
F_onewayResult(statistic=344.6605621917358, pvalue=7.937798855774506e-214)
```

Kruskal on casual users:



```
stat, p_value = kruskal(df_spring , df_summer , df_fall, df_winter )
print(p_value)

alpha = 0.05 # Significance level
if p_value < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"
print(decision)
```

```
KruskalResult(statistic=1537.3706788901238, pvalue=0.0)
```

Inference:

- Based on the results of both the ANOVA and Kruskal-Wallis tests conducted on Casual users, Registered users, and overall counts, it is evident that there exists a significant difference in bike usage across different seasons.

Recommendations:

Based on the significant difference in bike usage across different seasons as indicated by both the ANOVA and Kruskal-Wallis tests, here are some recommendations that could be suggested to the company:

- **Seasonal Marketing Strategies**:The company can tailor their marketing efforts to capitalize on the variations in bike usage across different seasons.
- **Inventory Management:** Understanding seasonal trends in bike usage can help the company better manage their inventory.
- **Service and Maintenance**: During peak seasons, when bike usage is high, the company may need to increase their focus on servicing and maintaining bikes to ensure they are in good working condition and can handle the higher usage levels.

4. Weather conditions Vs Seasons:

Step 1:

Ho: Weather conditions are the same during different seasons are same.

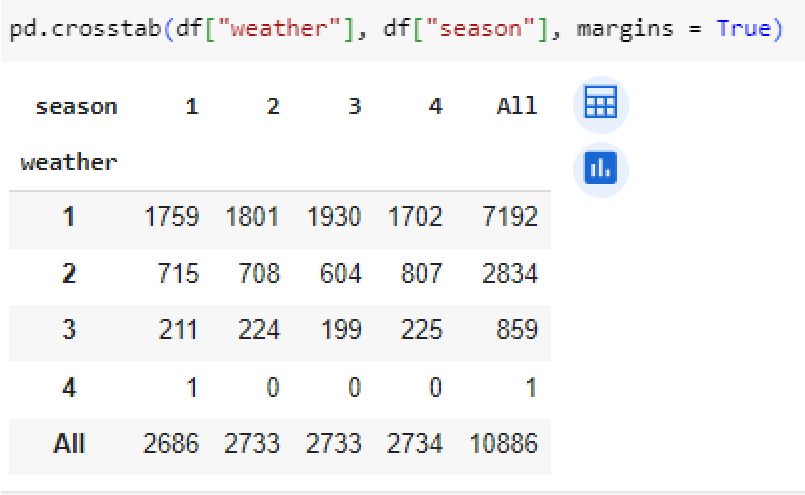
Ha: Weather conditions are significantly different during different seasons.

Step 2:

Test : ChiSquare Test

Step 3:

Contingency Table against ‘Weather’ & ‘Season’ columns:

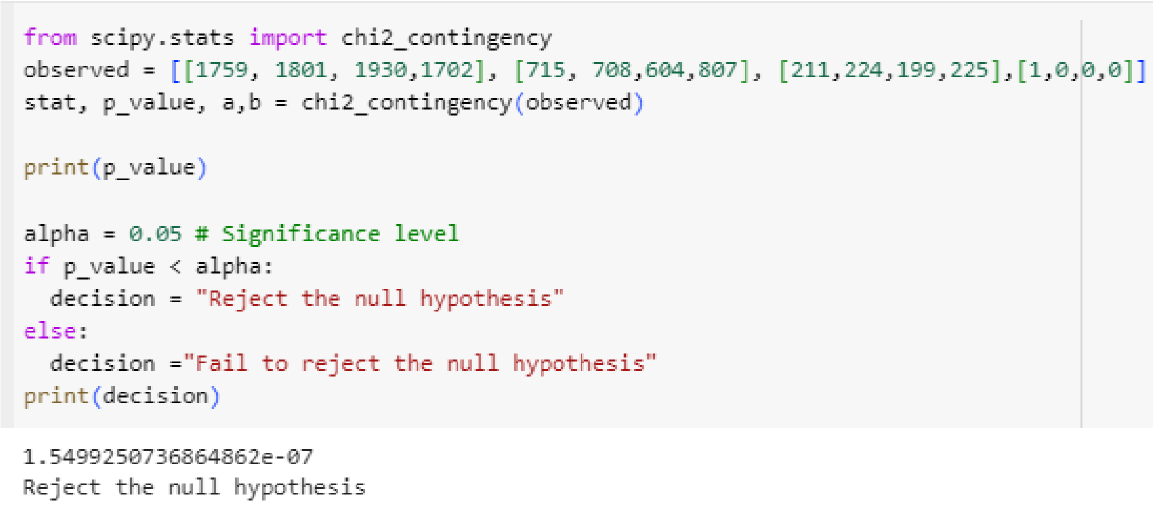


```
pd.crosstab(df["weather"], df["season"], margins = True)
```

season weather	1	2	3	4	All
1	1759	1801	1930	1702	7192
2	715	708	604	807	2834
3	211	224	199	225	859
4	1	0	0	0	1
All	2686	2733	2733	2734	10886

Step 4:

Significance level (alpha) = 0.05



```
from scipy.stats import chi2_contingency
observed = [[1759, 1801, 1930,1702], [715, 708,604,807], [211,224,199,225],[1,0,0,0]]
stat, p_value, a,b = chi2_contingency(observed)

print(p_value)

alpha = 0.05 # Significance level
if p_value < alpha:
  decision = "Reject the null hypothesis"
else:
  decision ="Fail to reject the null hypothesis"
print(decision)
```

```
1.5499250736864862e-07
Reject the null hypothesis
```

Inference:

- Based on the chi-square contingency test results, it's evident that there is a notable disparity in weather conditions across various seasons.

Recommendations:

Based on the significant differences in weather conditions across various seasons, several recommendations can be made to the company:

- ❖ Weather-Responsive Marketing
- ❖ Weather Forecast Integration

7. Conclusion:

Consolidated Insights:

- ➢ Registered users reached a peak usage count of 886 in a single hour, while casual users peaked at 367 in the same timeframe.
- ➢ Bike usage remains consistent regardless of whether it's a working day or a holiday.
- ➢ Weather conditions significantly influence bike usage patterns.
- ➢ Bike usage tends to decrease during the spring and winter seasons. Conversely, the highest levels of bike usage are observed during the summer and fall seasons.
- ➢ Positive correlation is observed between temperature and bike usage.
- ➢ Bike usage peaks when humidity levels are low, while it decreases when humidity is high.
- ➢ Bike usage reaches its peak when the windspeed falls within the medium range. Conversely, bike usage tends to be lower when windspeed is either low or high.
- ➢ Differences in bike usage between working days and holidays are evident across both casual and registered user categories. Casual users exhibit lower bike usage on holidays compared to working days. In contrast, registered users show heightened bike usage during holidays, while their working day usage remains lower.
- ➢ Casual users peak in fall, followed by summer, spring, and winter in descending order. Registered users show a peak in fall, then winter, and summer, with spring recording the least usage.
- ➢ Upon separately analyzing casual users and registered users, The pvalue indicates that there is a significant difference in bike usage between working days and holidays for these two user groups.
- ➢ Based on the results of both the ANOVA and Kruskal-Wallis tests conducted on Casual users, Registered users, and overall counts, it is evident that there exists a significant difference in bike usage across varying weather conditions and across different seasons.
- ➢ Based on the chi-square contingency test results, it's evident that there is a notable disparity in weather conditions across various seasons.

Consolidated Recommendations:

- ✓ To cater to both casual and registered users effectively, the company should devise distinct strategies aligned with their usage patterns.For casual users who utilize bikes most on working days, the company could introduce weekday-specific promotions and incentives. Meanwhile, for registered users who show peak usage during holidays, holiday-themed offers and extended rental options would be more appealing.

- ✓ Based on the insight that there is a significant difference in bike usage across different weather conditions for both casual and registered users, the company can tailor its operational and marketing strategies accordingly. Here are some recommendations:
    - o Weather-Based Promotions
    - o Targeted Marketing Campaigns
- ✓ Based on the significant difference in bike usage across different seasons as indicated by both the ANOVA and Kruskal-Wallis tests, here are some recommendations that could be suggested to the company:
    - **Seasonal Marketing Strategies**:The company can tailor their marketing efforts to capitalize on the variations in bike usage across different seasons.
    - **Inventory Management:** Understanding seasonal trends in bike usage can help the company better manage their inventory.