```
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/428/ori
```

```
--2025-02-01 06:45:09--  https://d2beiqkhq929f0.cloudfront.net/public_asset
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)
HTTP request sent, awaiting response... 200 OK
Length: 648353 (633K) [text/plain]
Saving to: 'bike_sharing.csv?1642089089.1'

bike_sharing.csv?16 100%[===================>] 633.16K  --.-KB/s    in 0.06

2025-02-01 06:45:09 (11.2 MB/s) - 'bike_sharing.csv?1642089089.1' saved [64
```

```
import pandas as pd
df=pd.read_csv('bike_sharing.csv?1642089089')
```

```
df.head()
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	win
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	
2	2011-01-01	1	0	0	1	9.02	13.635	80	

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
# 1. Define the Problem Statement, Import the required Libraries and perform Exp

#importing Libraries
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import ttest_ind,norm
import warnings
warnings.filterwarnings('ignore')
```

```
#a Examine dataset structure, characteristics, and statistical summary.
print('Shape is ',df.shape)
```

```
print('-'*50)
print(df.info())
print('-'*50)
print(df.describe())

#renaming count column
df.rename({'count':'Total_Users'},axis=1,inplace=True)
```

```
Shape is  (10886, 12)
--------------------------------------------------
<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
None
--------------------------------------------------
             season       holiday    workingday       weather          temp
count  10886.000000  10886.000000  10886.000000  10886.000000  10886.00000
mean       2.506614      0.028569      0.680875      1.418427     20.23086
std        1.116174      0.166599      0.466159      0.633839      7.79159
min        1.000000      0.000000      0.000000      1.000000      0.82000
25%        2.000000      0.000000      0.000000      1.000000     13.94000
50%        3.000000      0.000000      1.000000      1.000000     20.50000
75%        4.000000      0.000000      1.000000      2.000000     26.24000
max        4.000000      1.000000      1.000000      4.000000     41.00000

              atemp      humidity     windspeed        casual    registered
count  10886.000000  10886.000000  10886.000000  10886.000000  10886.000000
mean      23.655084     61.886460     12.799395     36.021955    155.552177
std        8.474601     19.245033      8.164537     49.960477    151.039033
min        0.760000      0.000000      0.000000      0.000000      0.000000
25%       16.665000     47.000000      7.001500      4.000000     36.000000
50%       24.240000     62.000000     12.998000     17.000000    118.000000
75%       31.060000     77.000000     16.997900     49.000000    222.000000
max       45.455000    100.000000     56.996900    367.000000    886.000000

              count
count  10886.000000
mean     191.574132
std      181.144454
min        1.000000
25%       42.000000
50%      145.000000
75%      284.000000
max      977.000000
```

```
#b Identify missing values and perform Imputation using an appropriate method.

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

	0
datetime	0
season	0
holiday	0
workingday	0
weather	0
temp	0
atemp	0
humidity	0
windspeed	0
casual	0
registered	0
Total_Users	0

dtype: int64

No missing Values

```
# c. Identify and remove duplicate records.

df[df.duplicated()]
```

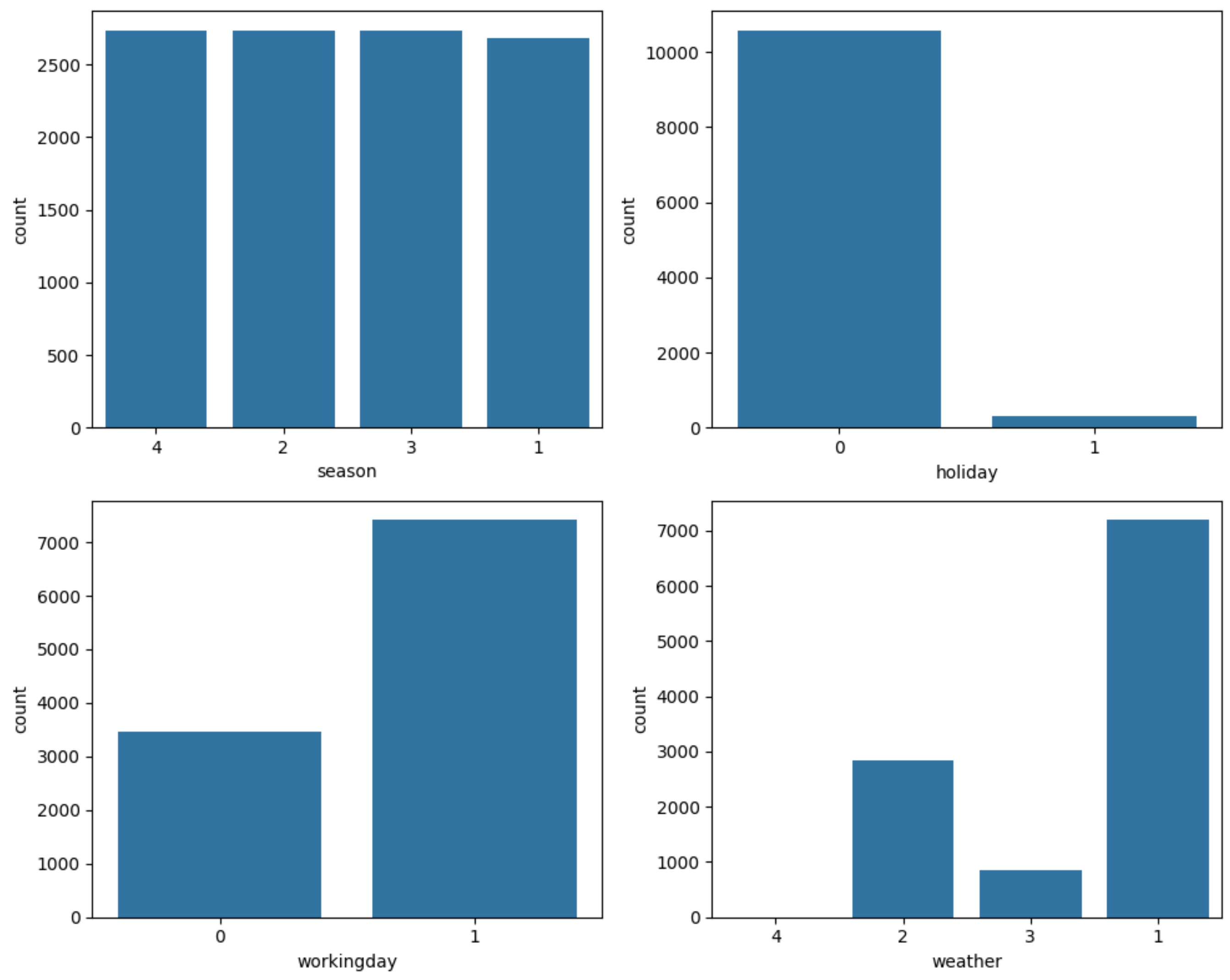
datetime	season	holiday	workingday	weather	temp	atemp	humidity	wind

No Duplicate records

```
#d. Analyze the distribution of Numerical & Categorical variables, separately

#Categorical variables
print(df['season'].unique(),'and',df['holiday'].unique(),'and',df['workingday']
```

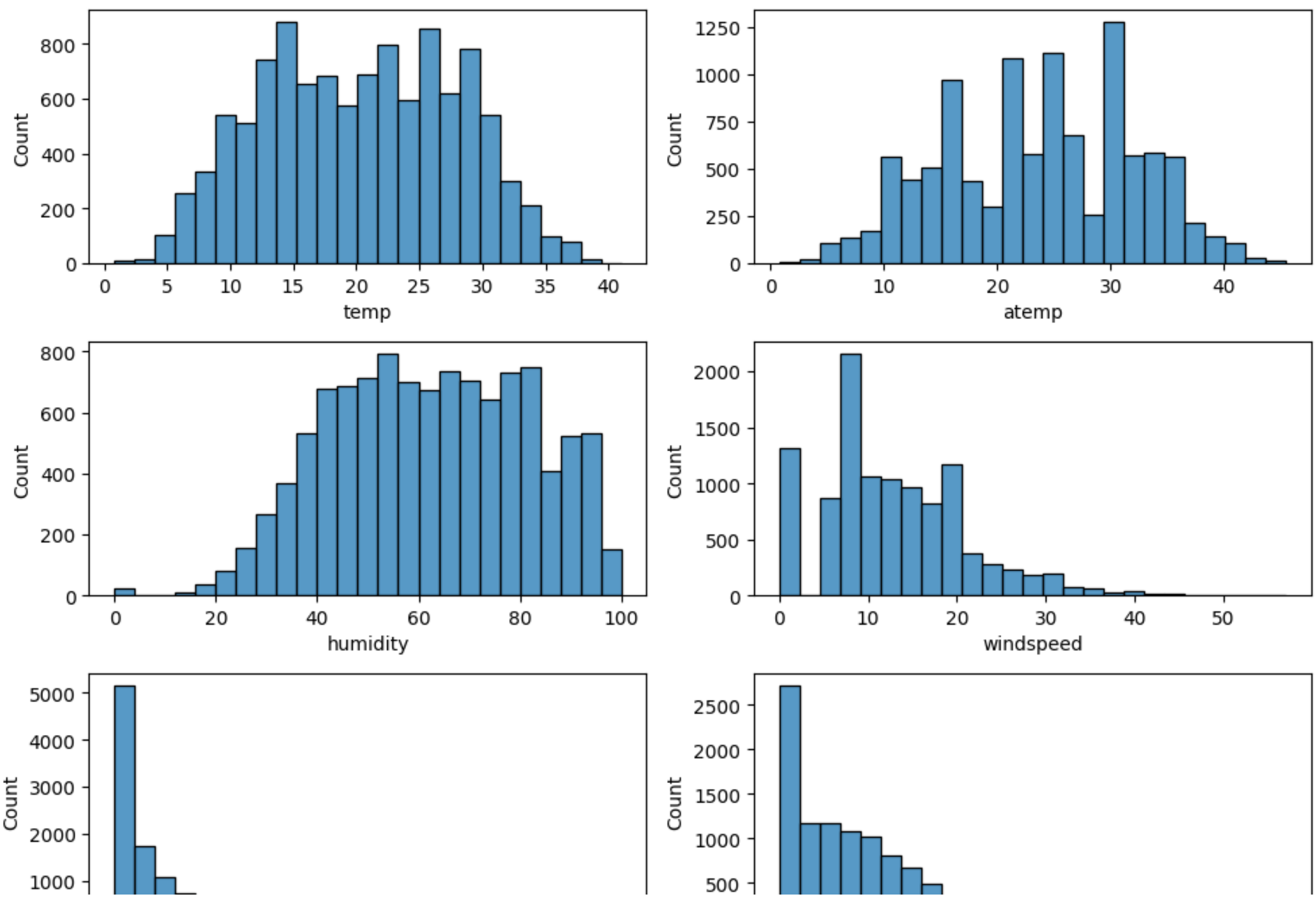
```
plt.figure(figsize=[10,8])

plt.subplot(2,2,1)
sns.countplot(x='season',data=df,order=df['season'].value_counts().index)

plt.subplot(2,2,2)
sns.countplot(x='holiday',data=df)

plt.subplot(2,2,3)
sns.countplot(x='workingday',data=df)

plt.subplot(2,2,4)
sns.countplot(x='weather',data=df,order=df['season'].value_counts().index)
plt.tight_layout()
plt.show()
```

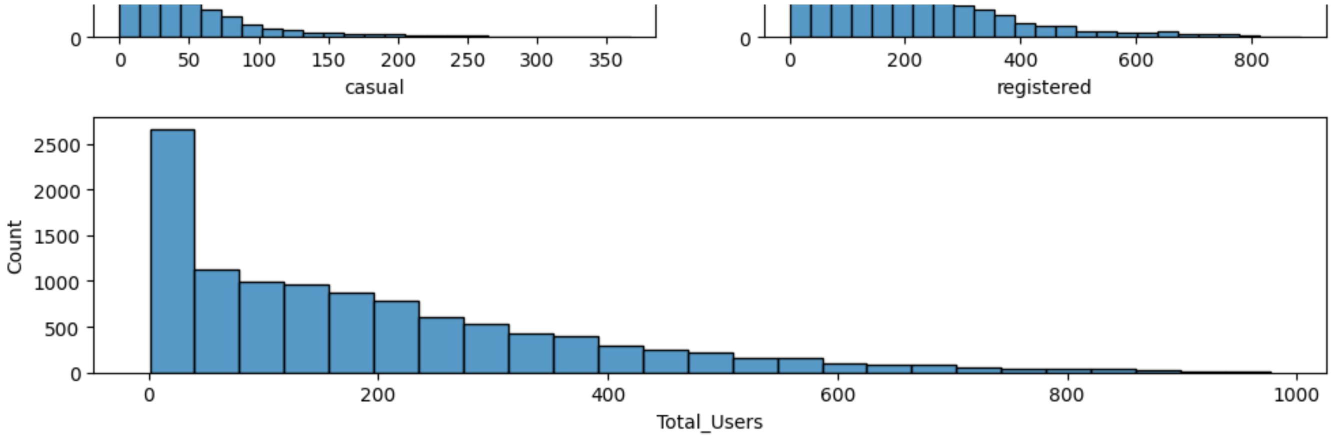
```
[1 2 3 4] and [0 1] and [0 1] and [1 2 3 4]
```



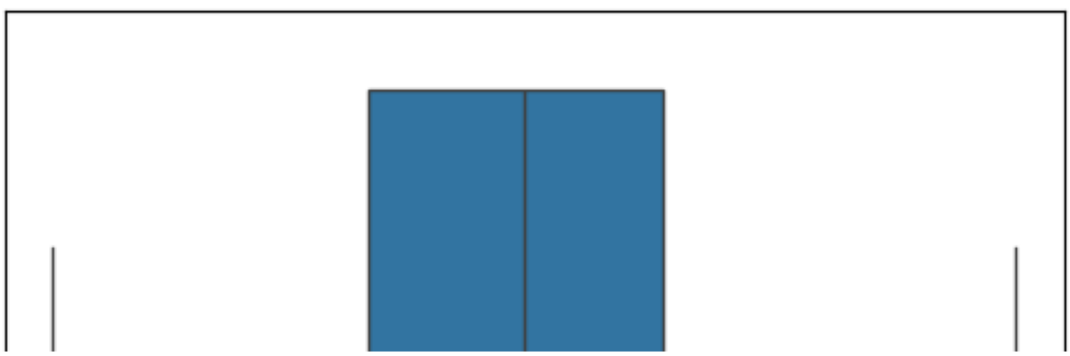

```
#Numercial variables

plt.figure(figsize=[10,10])
```

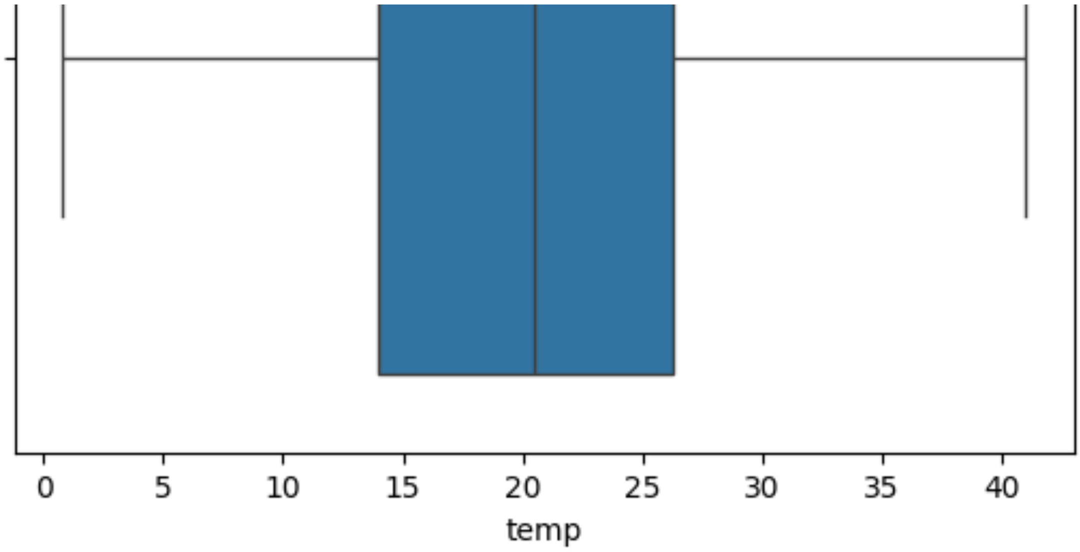
```
plt.subplot(4,2,1)
sns.histplot(df['temp'],bins=25)

plt.subplot(4,2,2)
sns.histplot(df['atemp'],bins=25)

plt.subplot(4,2,3)
sns.histplot(df['humidity'],bins=25)

plt.subplot(4,2,4)
sns.histplot(df['windspeed'],bins=25)

plt.subplot(4,2,5)
sns.histplot(df['casual'],bins=25)

plt.subplot(4,2,6)
sns.histplot(df['registered'],bins=25)

plt.subplot(4,1,4)
sns.histplot(df['Total_Users'],bins=25)

plt.tight_layout()

plt.show()
```

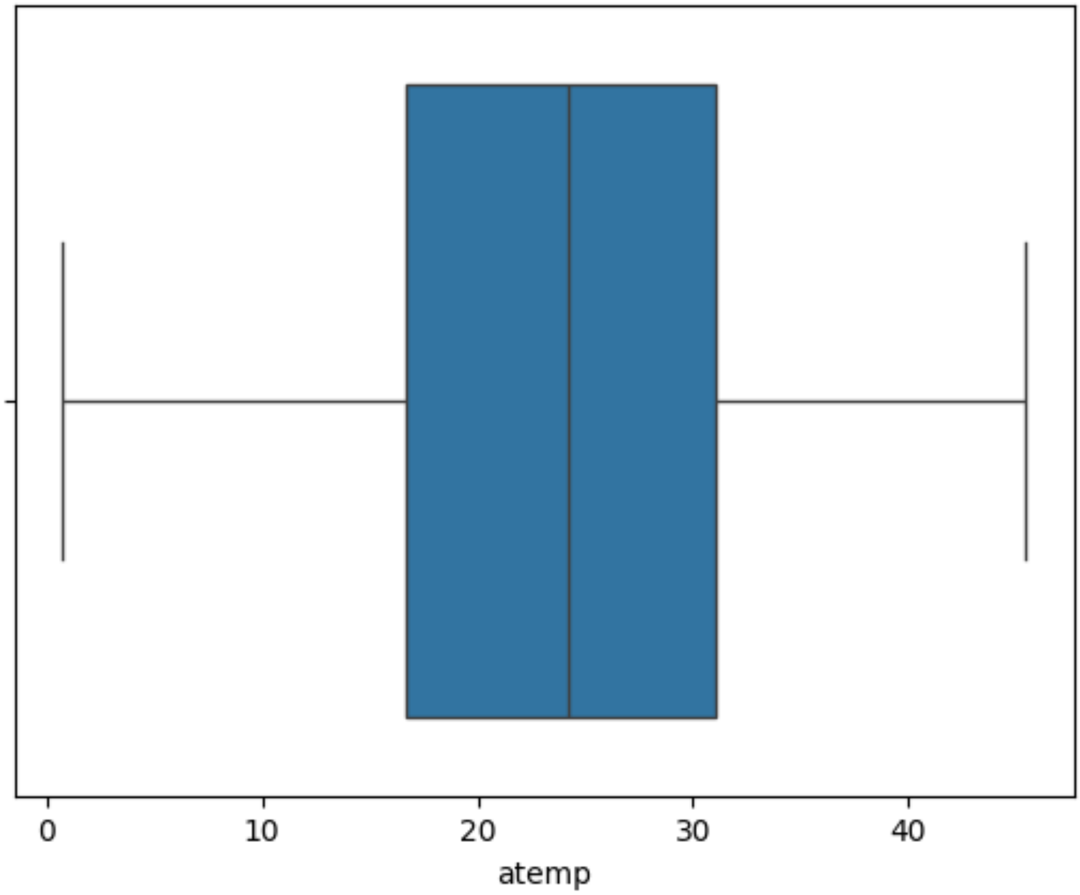


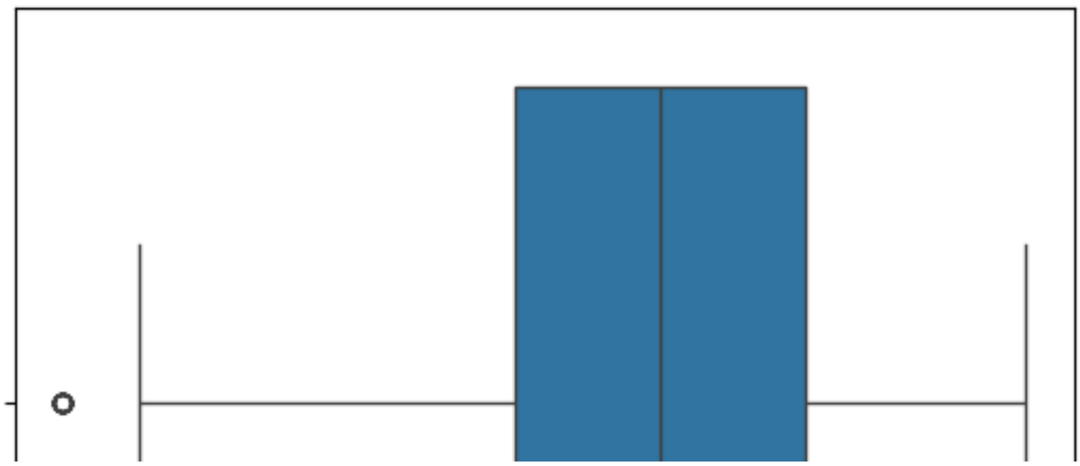


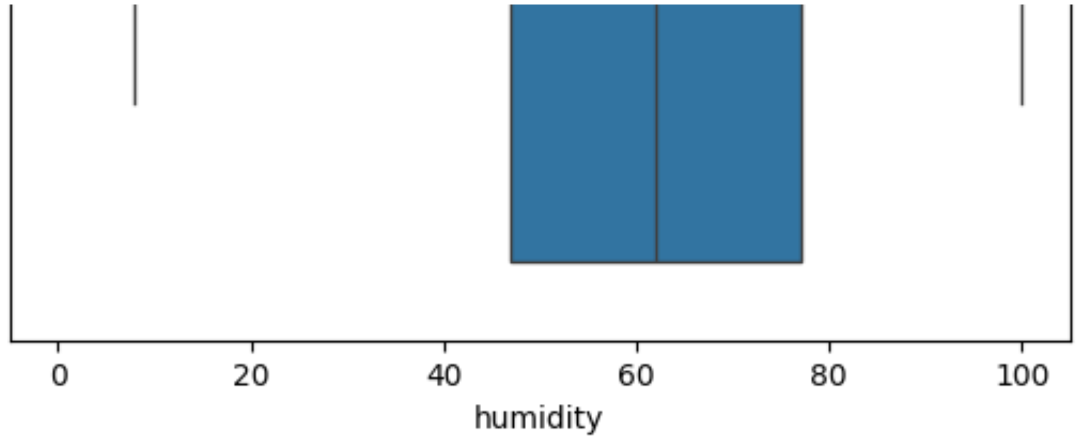

```
#e. Check for Outliers and deal with them accordingly
num_var= list(['temp','atemp','humidity','windspeed','casual','registered','Tot
for i in num_var:
  sns.boxplot(x=df[i])
  plt.show()
```

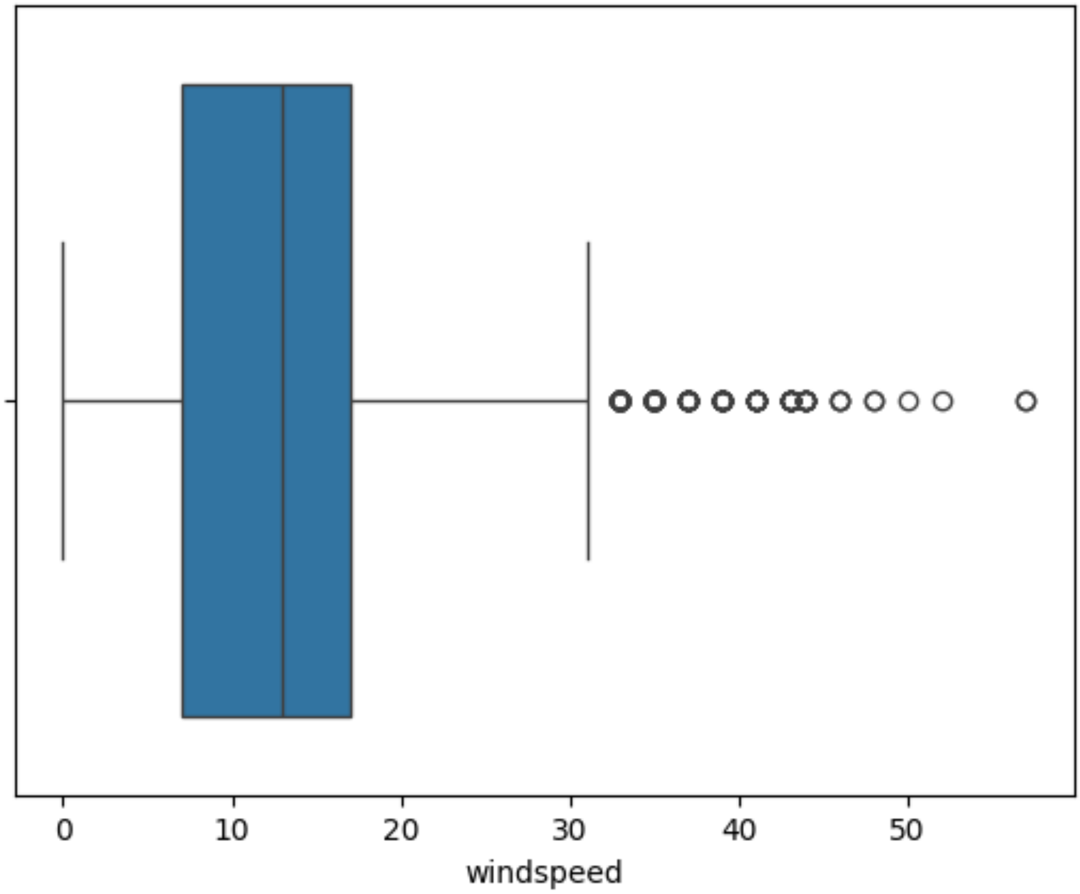


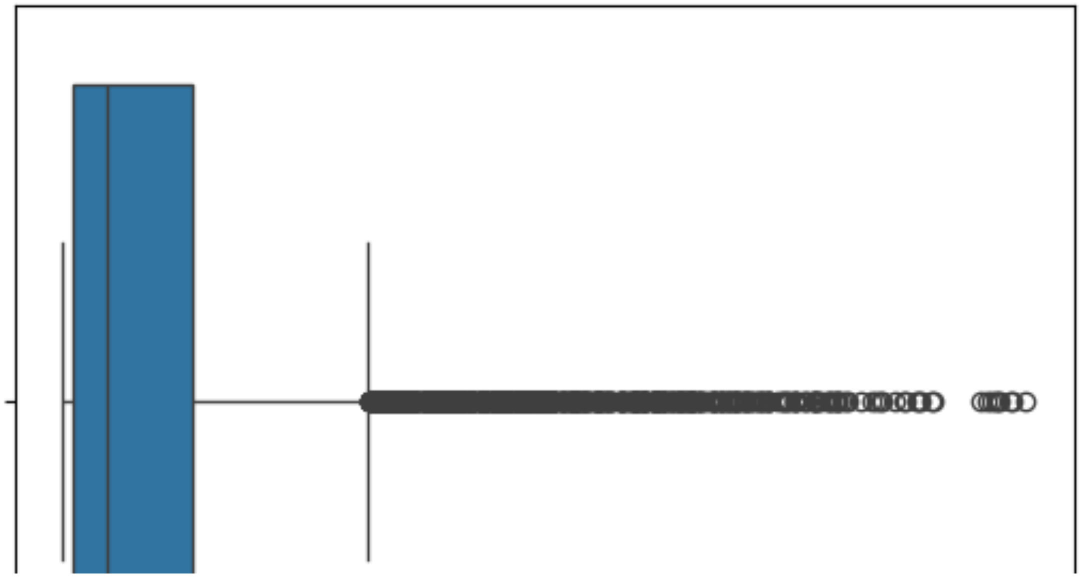

0 5 10 15 20 25 30 35 40
temp

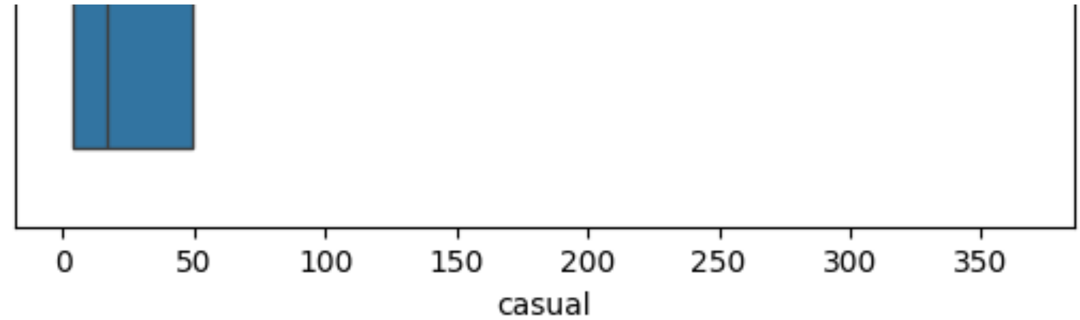

0 10 20 30 40
atemp

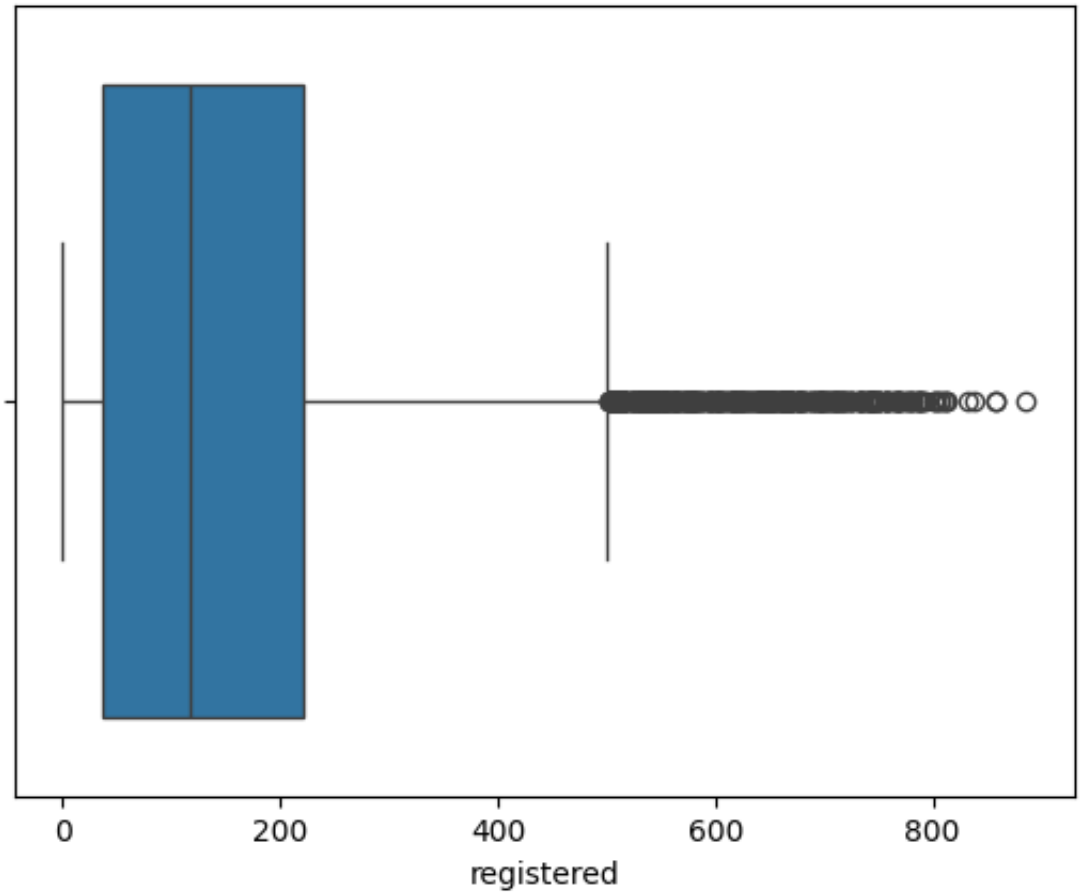


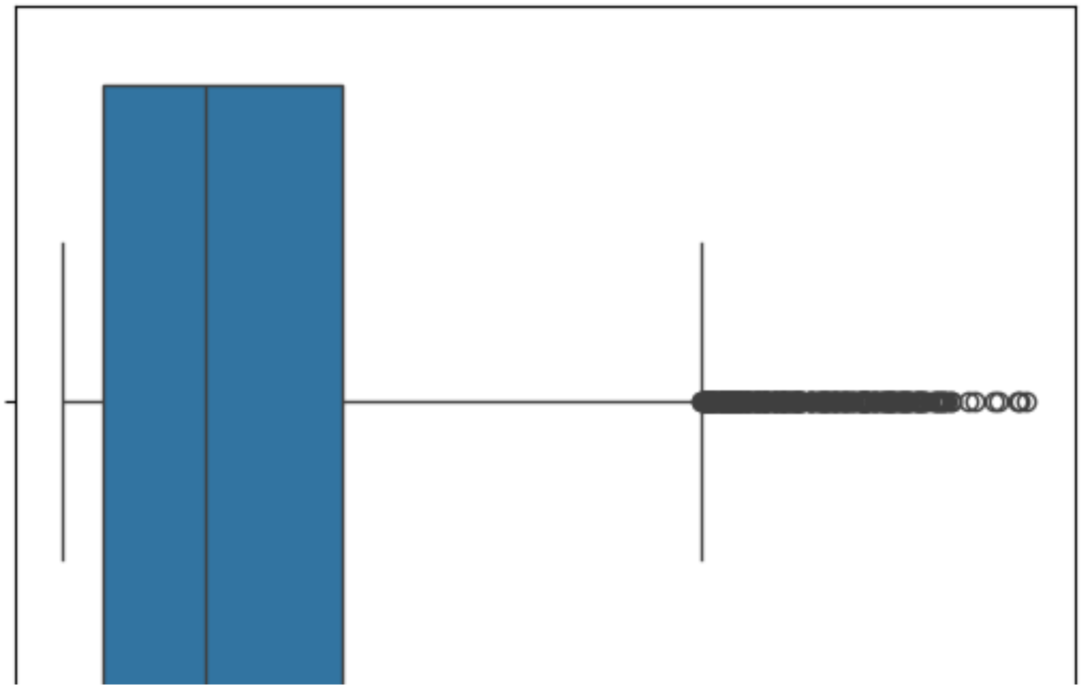

0
20
40
60
80
100
humidity

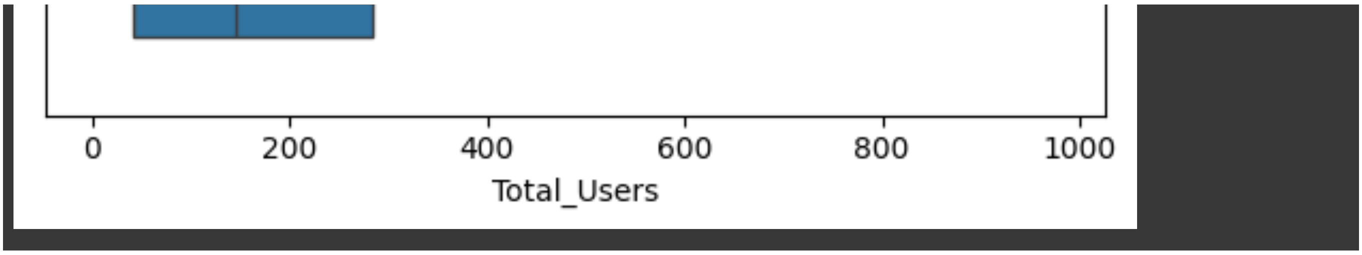

0
10
20
30
40
50
windspeed




0
50
100
150
200
250
300
350
casual


0
200
400
600
800
registered





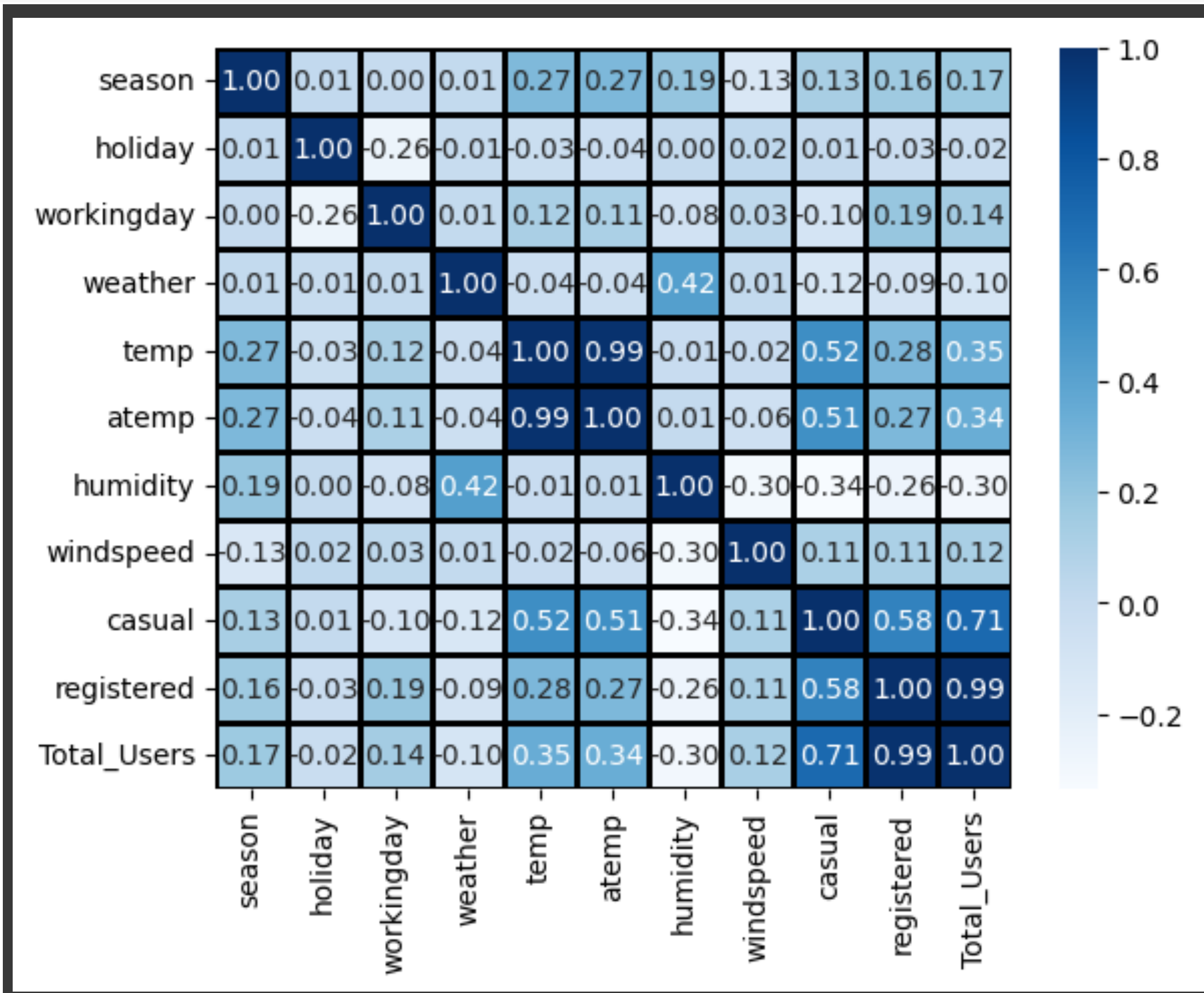
Temp and atemp column does not have any outliers whereas Humidity , windspeed , casual , registered and Total_Users have outliers.

```
#Handling Outliers

df_num = df.select_dtypes(include=np.number)
df_num.drop(['season','holiday','workingday','weather'],axis=1,inplace=True)

Q1=df_num.quantile(0.25)
Q3=df_num.quantile(0.75)

IQR=Q3-Q1

df_iqr = df[((df_num>=(Q1-1.5*IQR)) & (df_num<=(Q3+1.5*IQR))).all(axis=1)]

df_iqr.shape
```

```
(9518, 12)
```

After removing outliers, Length of the entire dataframe reduced to 9518 rows from 10886 rows.

```
#2. Try establishing a Relationship between the Dependent and Independent Variab

df_iqr_corr = df_iqr.drop(['datetime'],axis=1)

sns.heatmap(df_iqr_corr.corr(),annot=True,cmap='Blues',linewidths=1,fmt='.2f',li
plt.show()
```



```
#3. Check if there any significant difference between the no. of bike rides on

#weekdays
workingday_bike = df_iqr[df_iqr['workingday']==1]['Total_Users']

#weekends
holiday_bike = df_iqr[df_iqr['workingday']==0]['Total_Users']

print('Average count in working days',np.round(workingday_bike.mean(),2)) #161
print('Average count in Holidays ',np.round(holiday_bike.mean(),2))  #120
print('-'*100)

#NULL Hypothesis (H0) : There is no significant difference in number of bike ri
#Alternate Hypothesis(Ha) : There is a significant difference , number of bike
alpha=0.05

#2 sample Independent ttest
pvalue= ttest_ind(workingday_bike,holiday_bike,alternative='greater')[1]
print('P value is ',pvalue)
if(pvalue<alpha):
  print('Reject the Null Hypothesis')
else:
  print('Failed to reject the Null Hypothesis')
```

```
Average count in working days 161.97
Average count in Holidays  120.68
----------------------------------------------------------------------------------------------------
P value is  2.6924480901178837e-44
Reject the Null Hypothesis
```

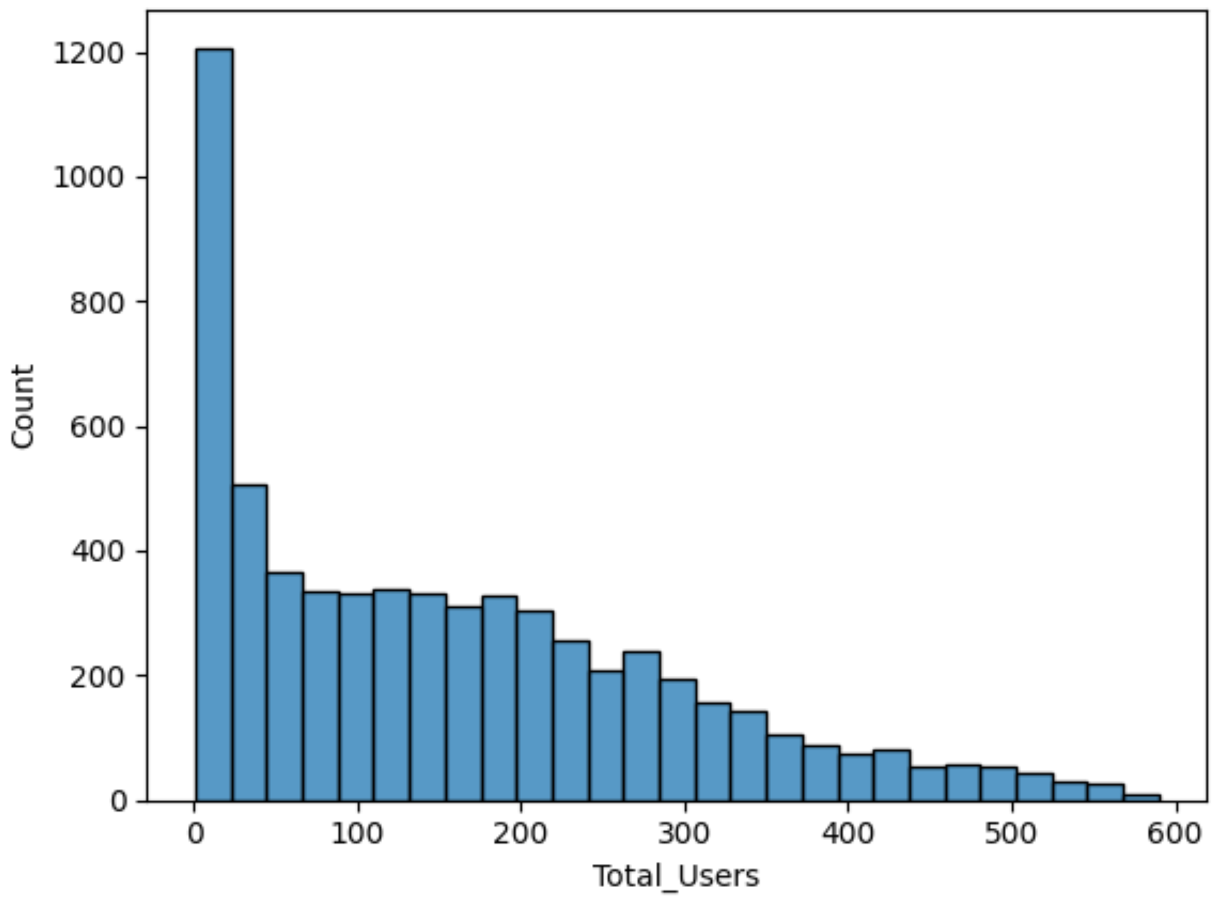
Hence , we see there is significant difference in number of bike rides. Number of bike rides in weekdays/working days are significantly higher than those in holidays/weekends.

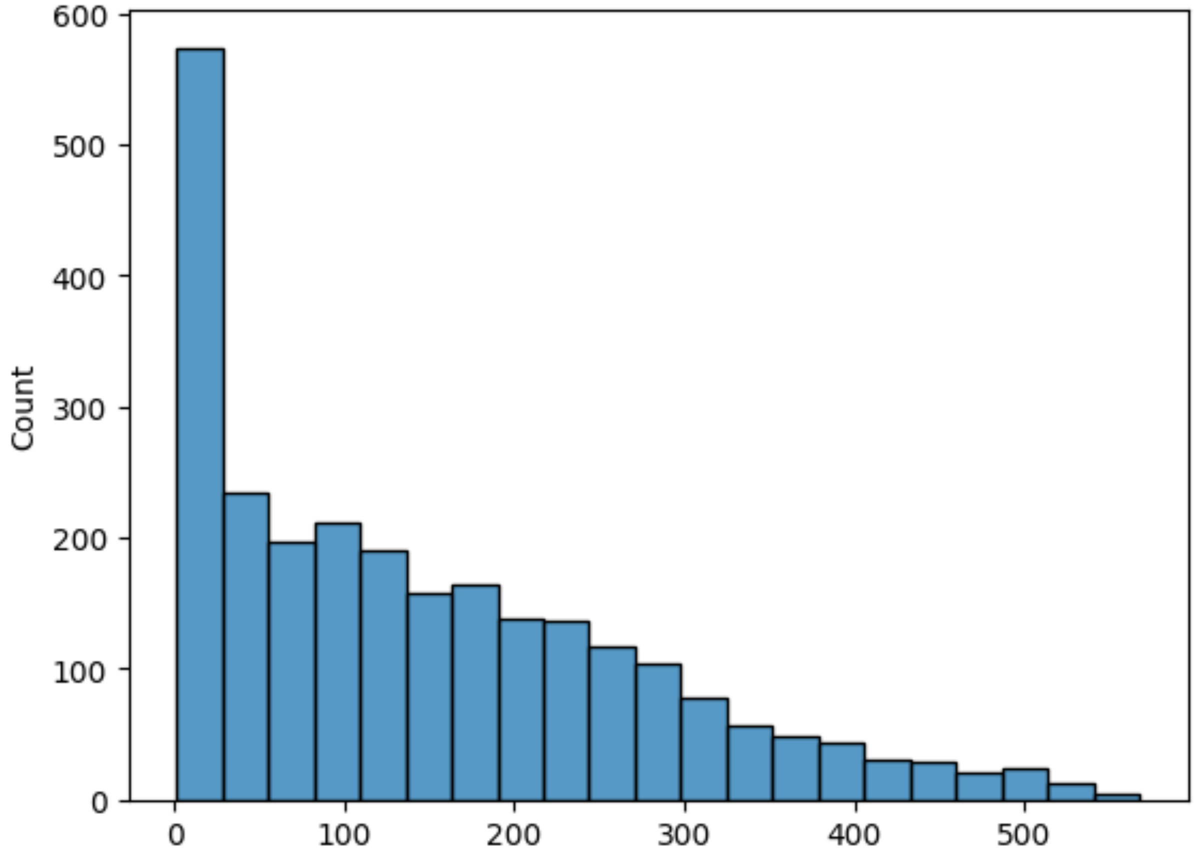
Recommendations: Company can introduce some offers for users in weekdays to increase the demands further.

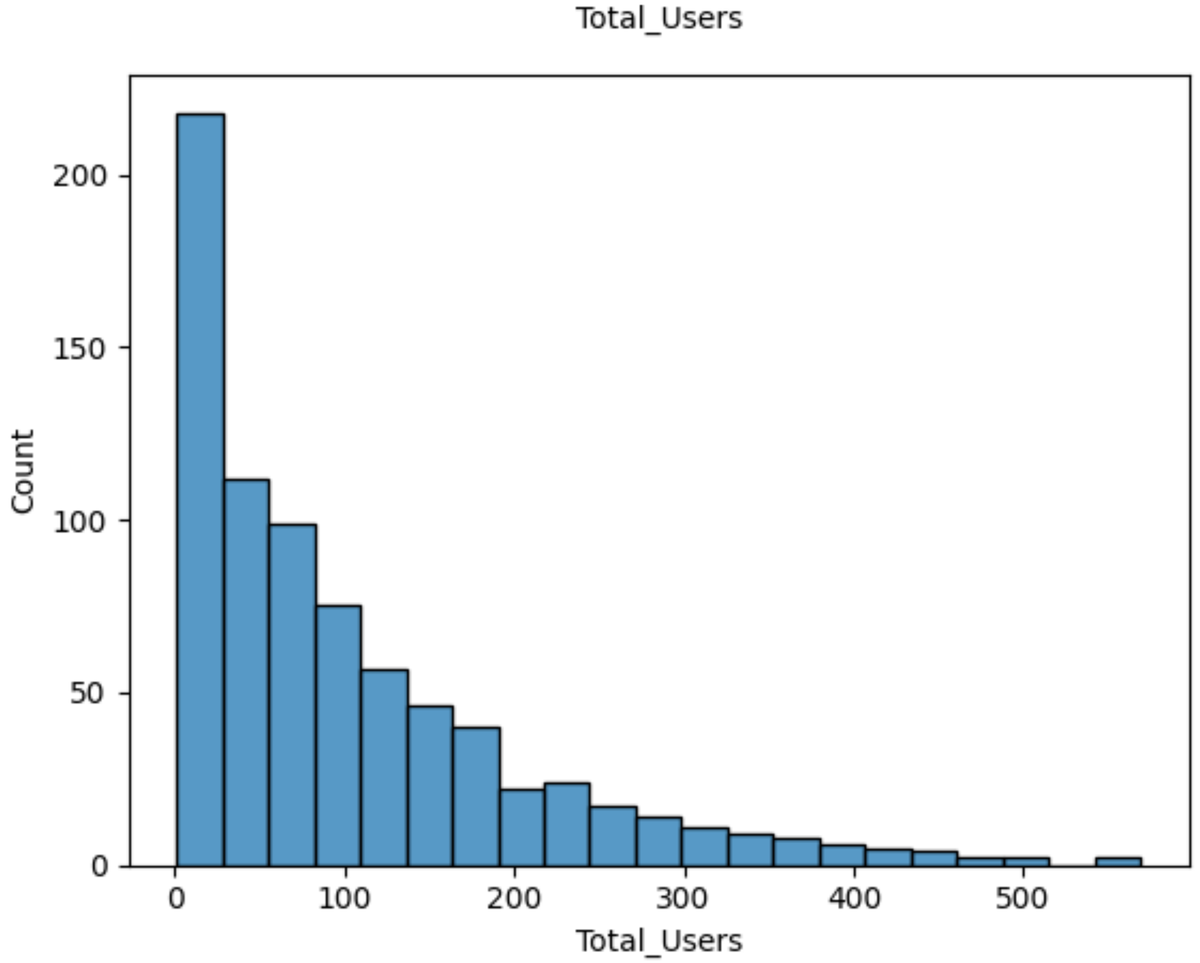
```
#4. Check if the demand of bicycles on rent is the same for different Weather c

#Null Hypothesis (H0) : There is no impact on demand of bicycles for different
#Alternate (Ha) : Demand of bicyles are significantly different for different w

clear_weather = df_iqr[df_iqr['weather']==1]
cloudy = df_iqr[df_iqr['weather']==2]
LightRain = df_iqr[df_iqr['weather']==3]
HeavyRain = df_iqr[df_iqr['weather']==4]
```

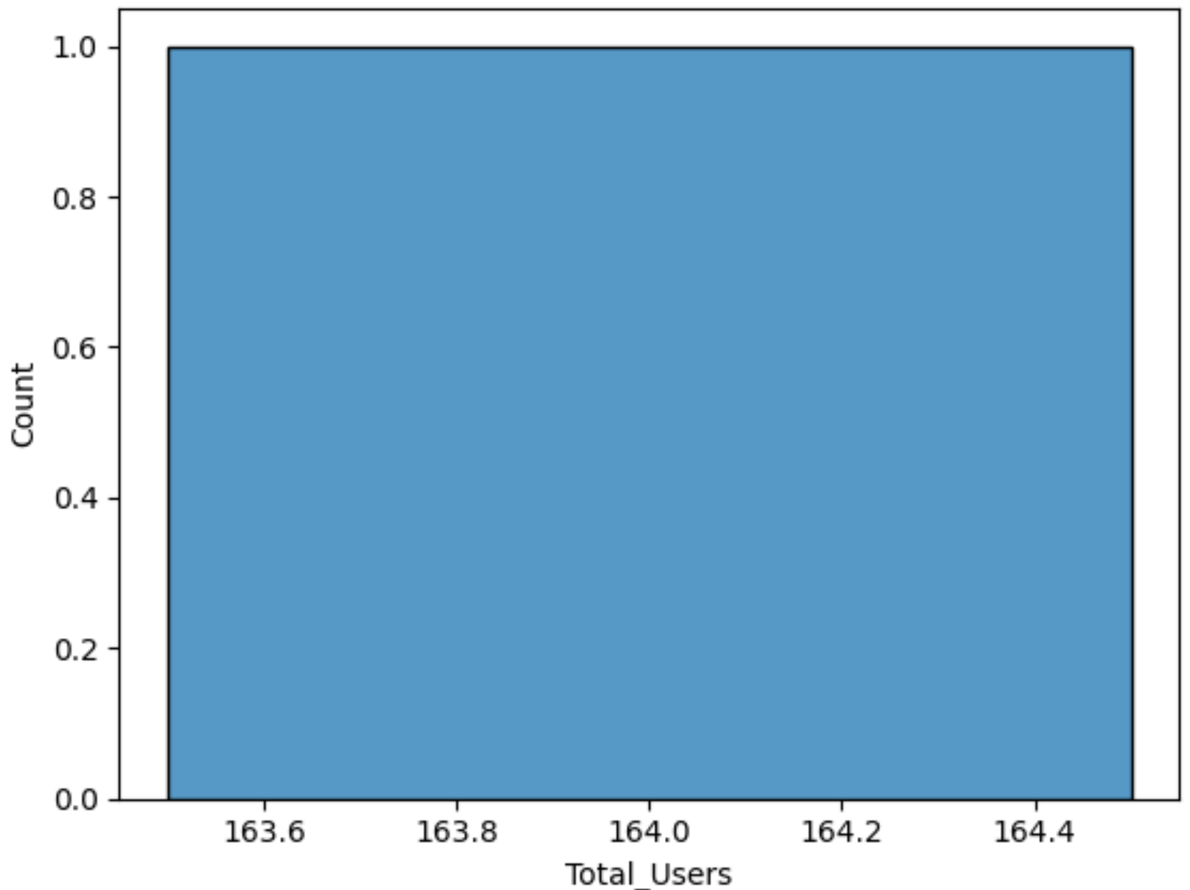
```
#Checking assumptions for one way ANOVA

#Normality(shapiro Wilk)
print('SHAPIRO WILK TEST')
from scipy.stats import shapiro

p1 = shapiro(clear_weather['Total_Users'])[1]
p2 = shapiro(cloudy['Total_Users'])[1]
p3 = shapiro(LightRain['Total_Users'])[1]
#p4 = shapiro(HeavyRain['Total_Users'])[1]

print('p values are',p1,p2,p3)  #Very small value , Data is not gaussian. Same

sns.histplot(clear_weather['Total_Users'])
plt.show()
sns.histplot(cloudy['Total_Users'])
plt.show()
sns.histplot(LightRain['Total_Users'])
plt.show()
sns.histplot(HeavyRain['Total_Users'])
plt.show()



#Checking Variance through Levene's test
print('-'*100)
print('LEVENE TEST')
from scipy.stats import levene
p=levene(clear_weather['Total_Users'],cloudy['Total_Users'],LightRain['Total_Us
print('P value for levene test is ',p )  #Very small p value , variances are nc

sns.boxplot(y='Total_Users',hue='weather',data=df_iqr)
plt.xlabel('Weather Category')
plt.show()  # As visible from boxplot , variances are not equal


#One way ANOVA(as asked in question , even though assumptions fail)
print('*'*100)
print('One Way ANOVA Testing')
alpha=0.05
from scipy.stats import f_oneway
pvalue = f_oneway(clear_weather['Total_Users'],cloudy['Total_Users'],LightRain[
print('P value from Anova test is',pvalue)

if(pvalue<alpha):
  print('Reject the Null Hypothesis')
else:
  print('Failed to reject the Null Hypothesis')
```

```
SHAPIRO WILK TEST
p values are 3.124536629582929e-50 2.1028135244186288e-36 7.518848257749986
```



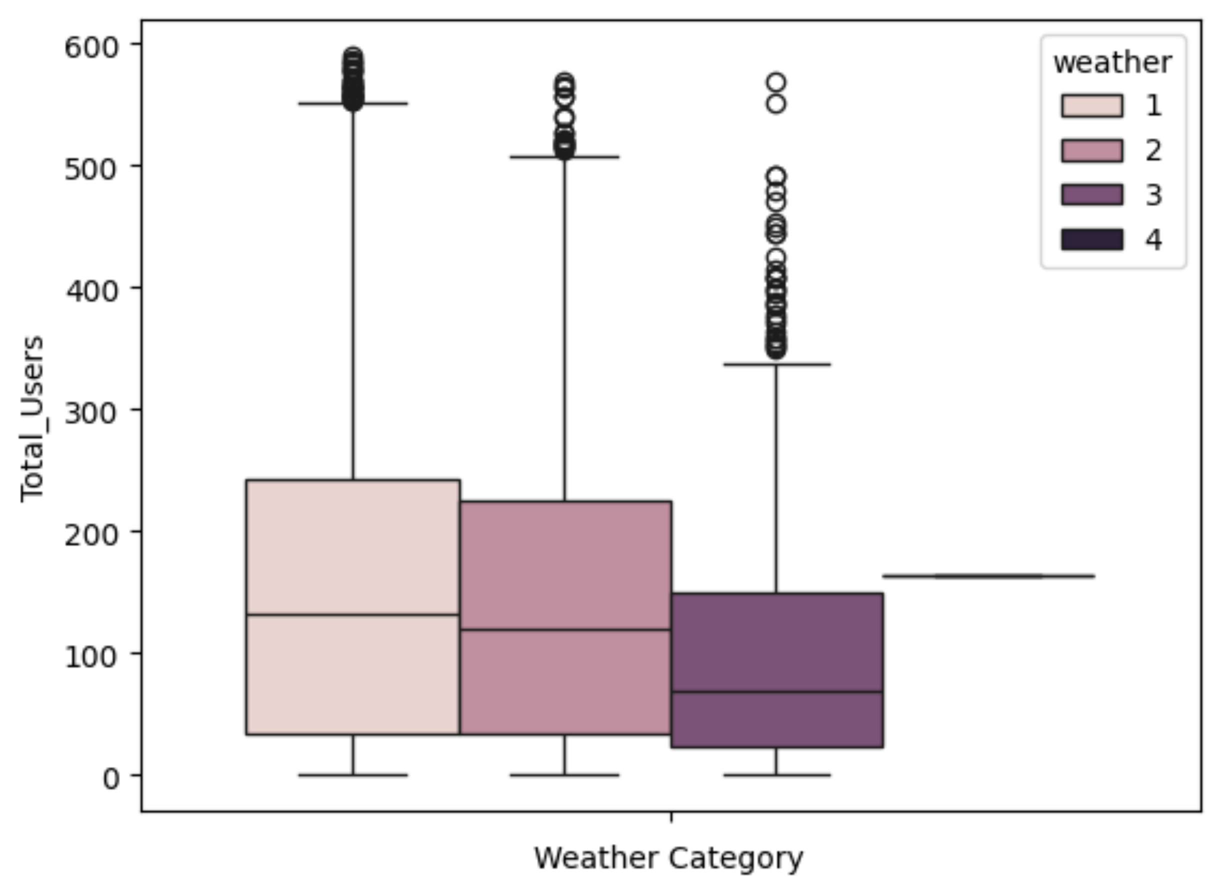



Total_Users
Count
0
50
100
150
200
0
100
200
300
400
500
Total_Users


Count
0.0
0.2
0.4
0.6
0.8
1.0
163.6
163.8
164.0
164.2
164.4
Total_Users

```
--------------------------------------------------------------------------------
LEVENE TEST
```

```
LEVENE TEST
P value for levene test is  4.67345079664337e-28
```




```
****************************************************************************
One Way ANOVA Testing
P value from Anova test is 7.867839321549754e-27
Reject the Null Hypothesis
```

As clear from above test that p value calculated from Anova is less than significance. Hence , we reject the null hypothesis and therefor there is significant difference in demands of bicycles for different weather conditions.

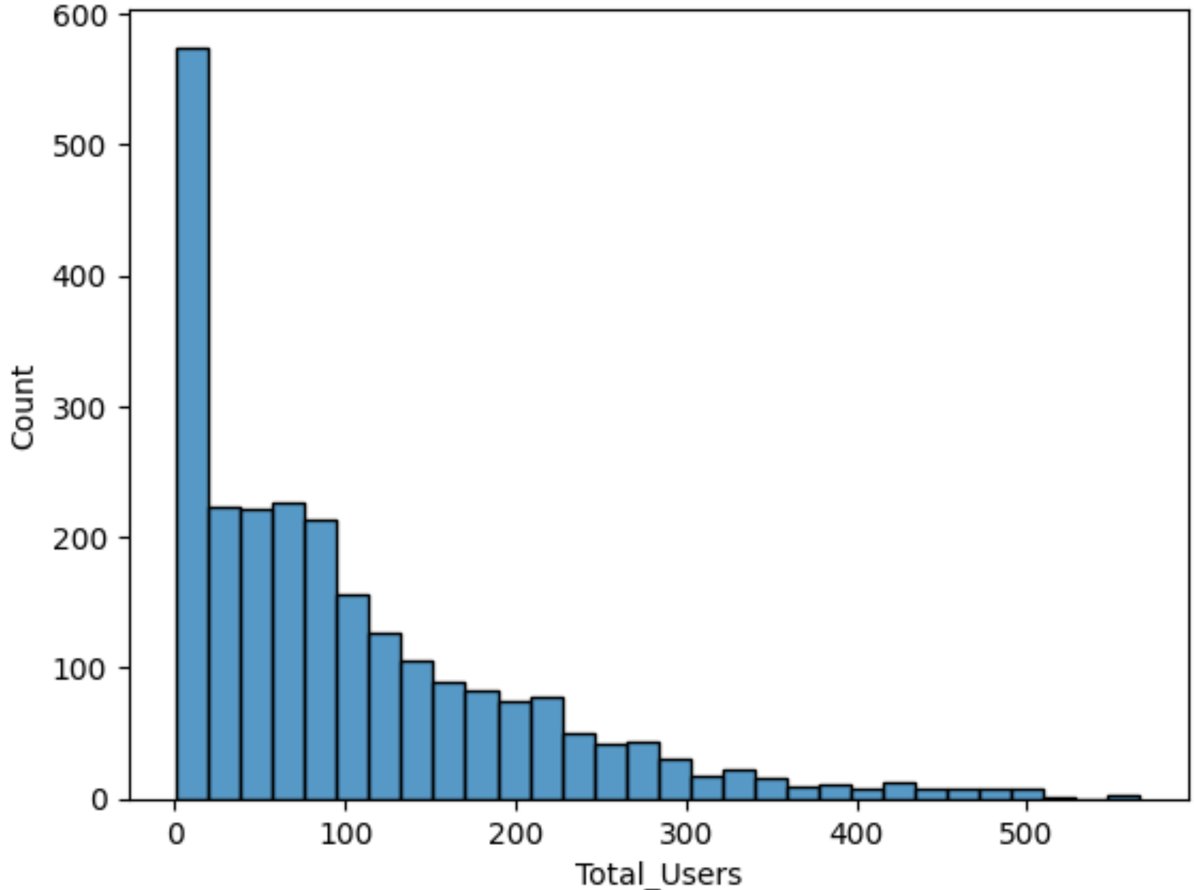
Recommendations: Company can look further more as to which weather condition is supporting the demand and which is leading to decline and make appropriate strategies

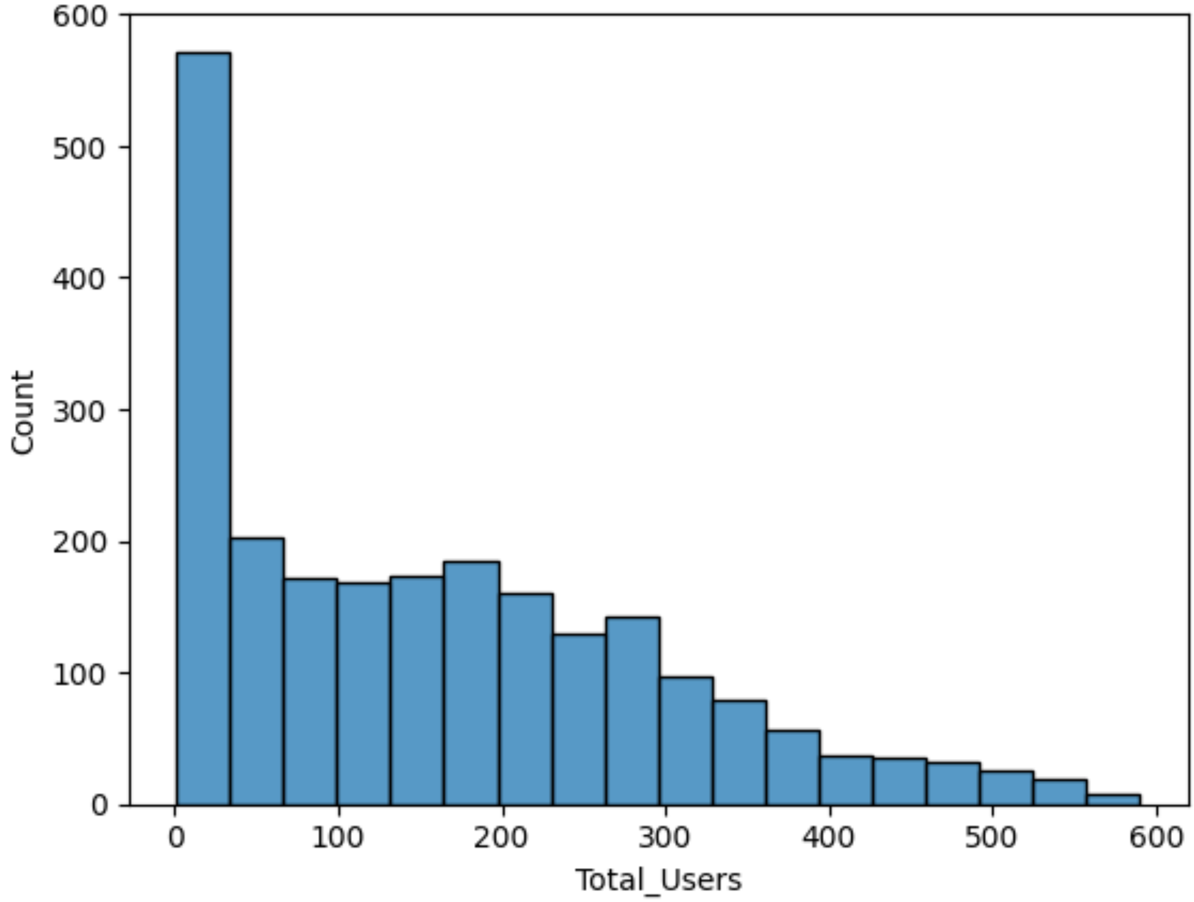
```
#5 Check if the demand of bicycles on rent is the same for different Seasons?

#Null Hypothesis (H0) : There is no impact on demand of bicycles for different
#Alternate (Ha) : Demand of bicyles are significantly different for different S

spring=df_iqr[df_iqr['season']==1]
summer=df_iqr[df_iqr['season']==2]
fall=df_iqr[df_iqr['season']==3]
winter=df_iqr[df_iqr['season']==4]
```

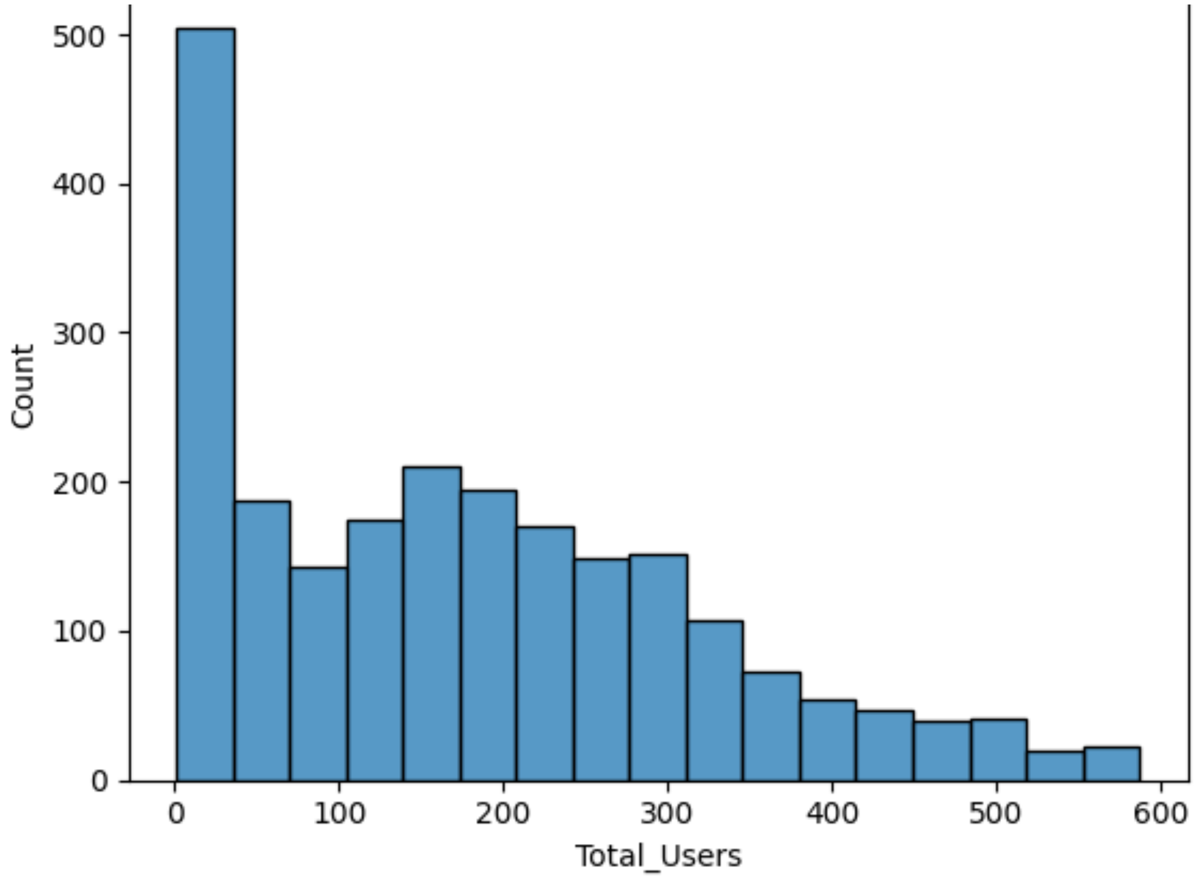
```
#Assumptions of One way ANOVA
#Checking Normality
print('Shapiro Wilk Test')
p1=shapiro(spring['Total_Users'])[1]
p2=shapiro(summer['Total_Users'])[1]
p3=shapiro(fall['Total_Users'])[1]
p4=shapiro(winter['Total_Users'])[1]
print('P values for Shapiro Wilk Test are ',p1,p2,p3,p4)
print('Very small value , Data is not gaussian. Same is visible from below hist

sns.histplot(spring['Total_Users'])
plt.show()
sns.histplot(summer['Total_Users'])
plt.show()
sns.histplot(fall['Total_Users'])
plt.show()
sns.histplot(winter['Total_Users'])
plt.show()




#Checking Variance
print('-'*100)
print('LEVENE TEST')
p = levene(spring['Total_Users'],summer['Total_Users'],fall['Total_Users'],wint
print('P value for Levene Test is ',p)  #
print('Very small p value ,variances are not equal for different groups.Below i
sns.boxplot(y='Total_Users',hue='season',data=df_iqr)
plt.show()



#One way ANOVA(as asked in question , even though assumptions fail)
print('*'*100)
print('One Way ANOVA Testing')
alpha=0.05

pvalue=f_oneway(spring['Total_Users'],summer['Total_Users'],fall['Total_Users']
print('P value from Anova test is',pvalue)

if(pvalue<alpha):
  print('Reject the Null Hypothesis')
else:
  print('Failed to reject the Null Hypothesis')
```

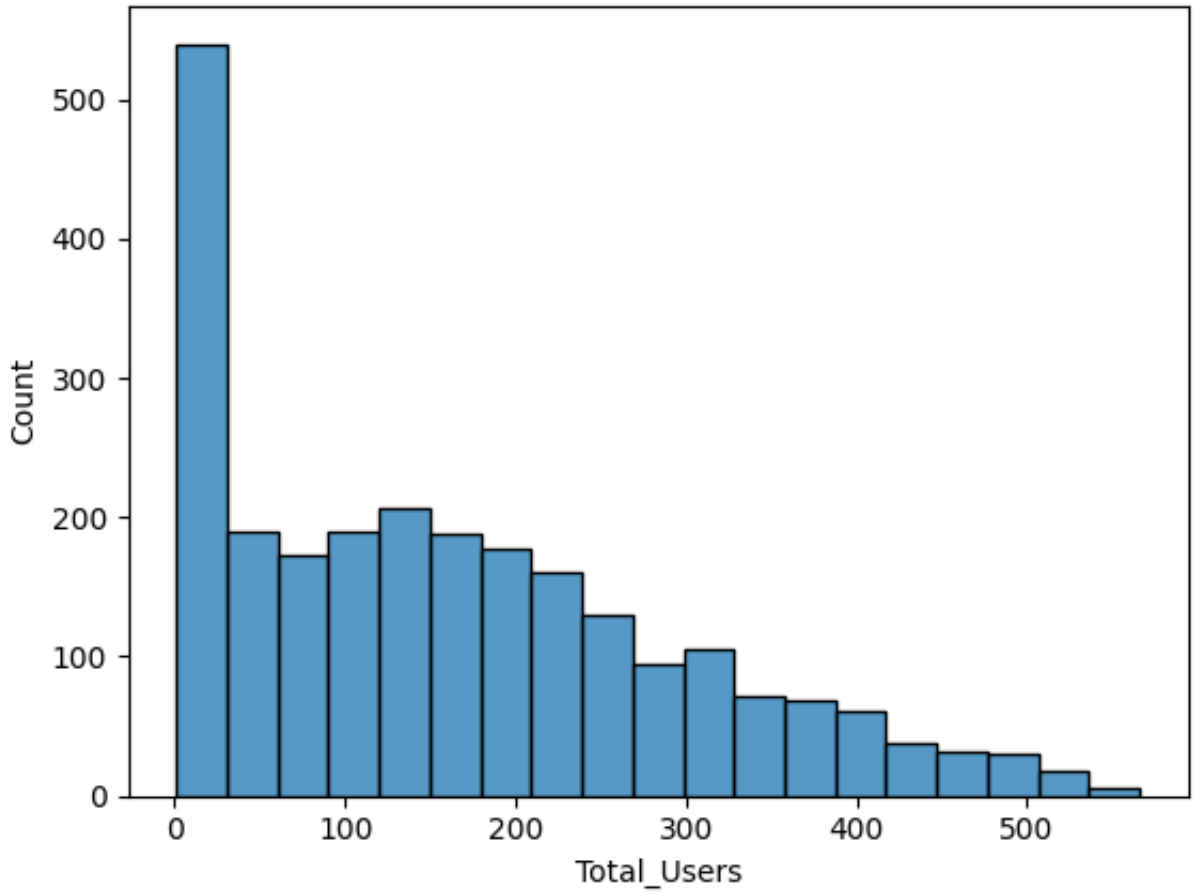
```
Shapiro Wilk Test
P values for Shapiro Wilk Test are  2.0714964386977434e-42 1.2437934617894
```

```
P values for Shapiro Wilk Test are  2.0714964386977434e-42 1.2427934617894
Very small value , Data is not gaussian. Same is visible from below histplo
```

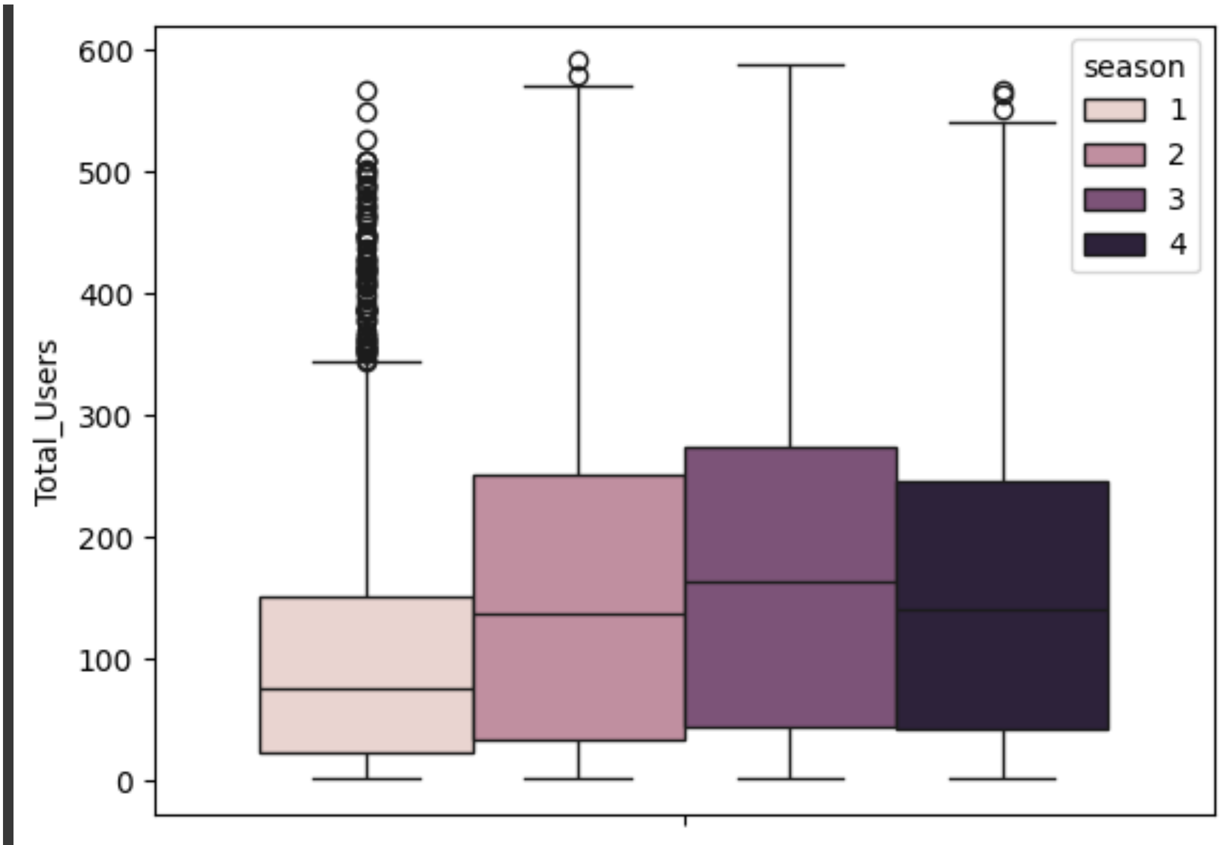









```
--------------------------------------------------------------------------------
LEVENE TEST
P value for Levene Test is  6.687186315723853e-87
Very small p value ,variances are not equal for different groups.Below is t
```




```
***************************************************************************
One Way ANOVA Testing
P value from Anova test is 1.328514170995064e-98
Reject the Null Hypothesis
```

As clear from above test that p value calculated from Anova is less than significance. Hence , we reject the null hypothesis and therefor there is significant difference in demands of bicycles for different seasons

Recommendations: Company can look further more as to which season is supporting the demand and which is leading to decline and make appropriate strategies

```
#6 Check if the Weather conditions are significantly different during different

#Null Hypothesis (H0) : Weather condition and seasons are independent of each c
#Alternate (Ha) : Weather conditions are affected by seasons or vice-versa

df_iqr['season_names']=df_iqr['season'].replace({1:'spring',2:'summer',3:'fall'
df_iqr['weather_names']=df_iqr['weather'].replace({1:'clear',2:'cloudy',3:'Ligh

df_iqr.drop(df_iqr[df_iqr['weather_names']=='HeavyRain'].index,inplace=True)  #

observed=pd.crosstab(df_iqr['season_names'],df_iqr['weather_names'])

#Chi Square test for independence
alpha=0.05
from scipy.stats import chi2_contingency
p_value=chi2_contingency(observed)[1]
print('P value is ',p_value)
if(p_value<alpha):
  print('Reject the Null Hypothesis')
else:
  print('Failed to reject the Null Hypothesis')
```

```
P value is  1.797264261807596e-08
Reject the Null Hypothesis
```

As clear from above test that p value calculated is less than significance. Hence , we reject the null hypothesis and therefor there is significant difference in weather conditions for different seasons

Recommendations: Company can look further into a particular season and weather which is supporting the demand and which one is leading to decline and make appropriate strategies