Yulu – Bikes

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.

How you can help here?

The company wants to know: • Which variables are significant in predicting the demand for shared electric cycles in the Indian market?

• How well those variables describe the electric cycle demands Dataset:

Column Profiling:

• **datetime:** datetime • **season:** season (1: spring, 2: summer, 3: fall, 4: winter)

• **holiday:** whether day is a holiday or not (extracted from http://dchr.dc.gov/page/holiday-schedule)

• **workingday:** if day is neither weekend nor holiday is 1, otherwise is 0.

• **weather:**

o 1: Clear, Few clouds, partly cloudy, partly cloudy

o 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist

o 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds

o 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog

• **temp:** temperature in Celsius

• **atemp:** feeling temperature in Celsius

• **humidity:** humidity

• **windspeed:** wind speed

• **casual:** count of casual users

- **registered:** count of registered users
- **count:** count of total rental bikes including both casual and registered
- Import the dataset and do usual exploratory data analysis steps like checking the structure & characteristics of the dataset
- Try establishing a relation between the dependent and independent variable (Dependent “Count” & Independent: Workingday, Weather, Season etc)
- Select an appropriate test to check whether:

o Working Day has effect on number of electric cycles rented

o No. of cycles rented similar or different in different seasons

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

o Weather is dependent on season (check between 2 predictor variable)

- Set up Null Hypothesis (H0)
- State the alternate hypothesis (H1)
- Check assumptions of the test (Normality, Equal Variance).

You can check it using Histogram, Q-Q plot or statistical methods like levene’s test, Shapiro-wilk test (optional)

o Please continue doing the analysis even If some assumptions fail (levene’s test or Shapiro-wilk test) but double check using visual analysis and report wherever necessary

- Set a significance level (alpha)
- Calculate test Statistics.
- Decision to accept or reject null hypothesis.
- Inference from the analysis

Solution

In [65]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

importing yulu dataset

In [66]:
```
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/428/origi
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/428/original/bike_sharing.csv?1642089089
To: /content/bike_sharing.csv?1642089089
100% 648k/648k [00:00<00:00, 8.16MB/s]
```

In [67]:
```
data = pd.read_csv("bike_sharing.csv?1642089089")
data
```

Out[67]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	cas
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0000	
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0000	
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0000	
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0000	
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0000	
...	...	...	...	...	...	...	...	...	...	
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	

10886 rows × 12 columns

checking the shape of dataset

In [68]:
```
data.shape
```

Out[68]:
```
(10886, 12)
```

In [69]:
```
data.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
```

from above dataset info we can say that there is no null values for all variables in the given dataset

now we will perform EDA for the given dataset

Univariate Analysis

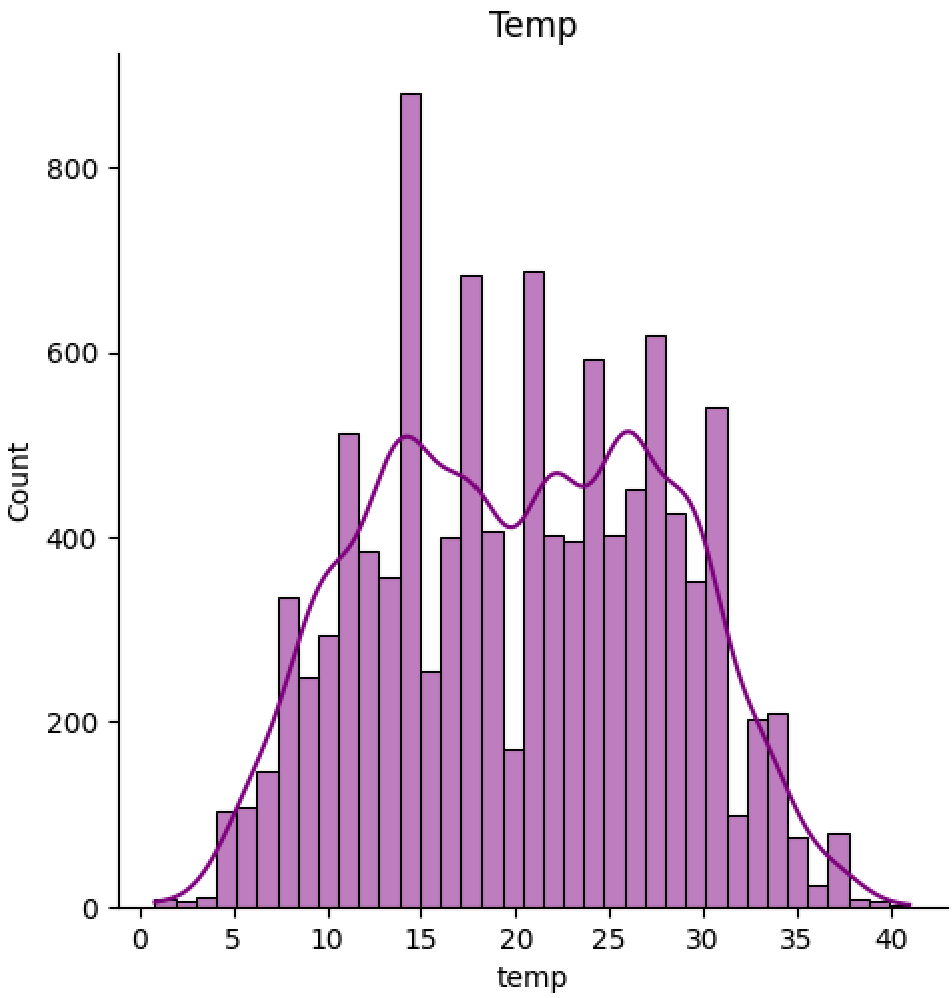
> > distribution plots of all the continuous variable(s)
> >
> > barplots/countplots of all the categorical variables

continuous variables are temp,atemp,humidity,windspeed,casual ,registered ,count

In [71]:

```
sns.displot(data['temp'],color='purple',kde=True)
plt.title("Temp")
```

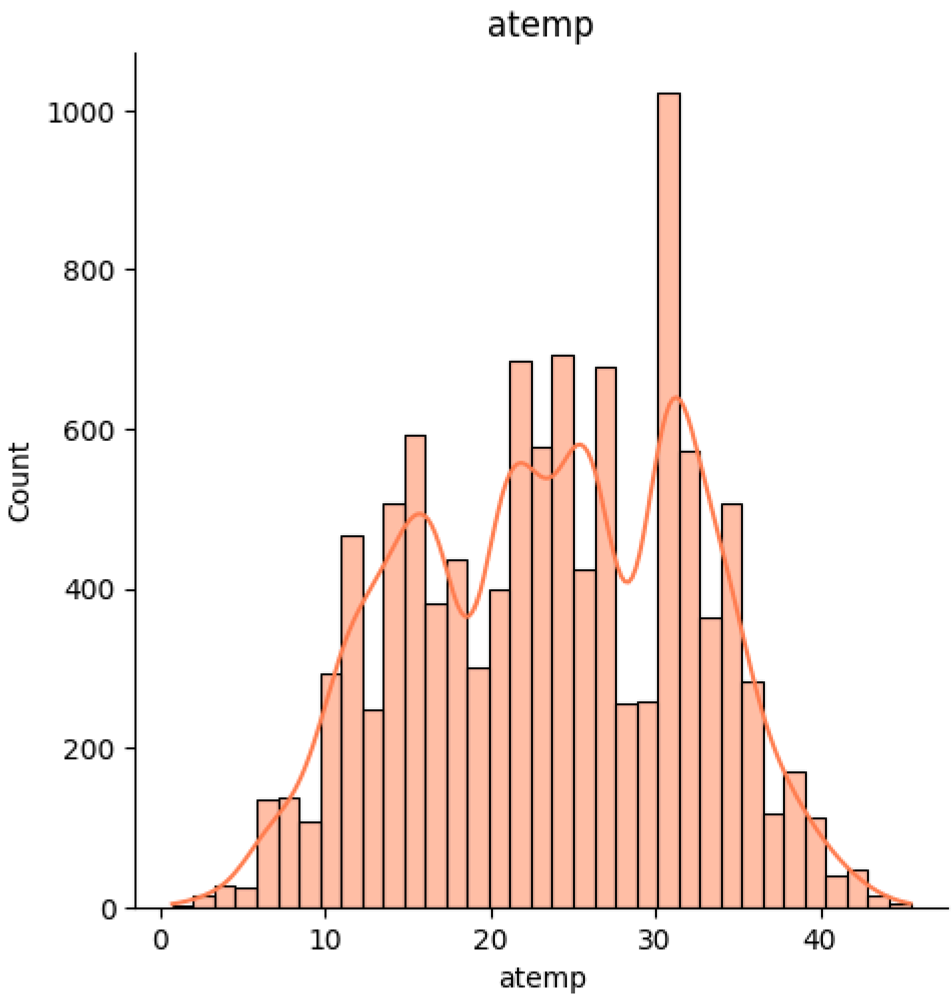
Out[71]: Text(0.5, 1.0, 'Temp')



Insights: from above graph we can say the maximum number of entries in data is for temerature close to 15 and subsequently most of the bike rented when temperature lies between 10-30 on celsius temp scale

In [72]:
```
sns.displot(data['atemp'],color='coral',kde=True)
plt.title("atemp")
```

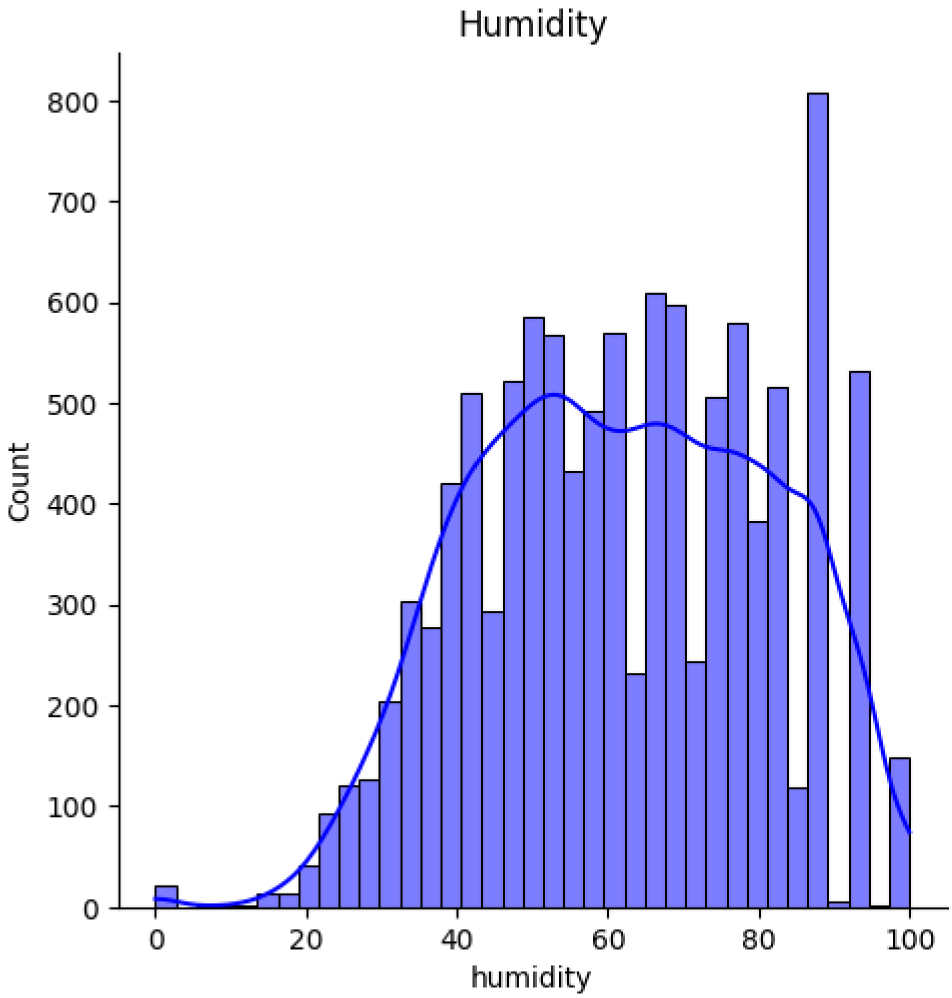
Out[72]:
```
Text(0.5, 1.0, 'atemp')
```



insights: air temperature mostly lies in between 10 -35 celcius temp

In [73]:
```
sns.displot(data['humidity'],color='Blue',kde=True)
plt.title("Humidity")
```

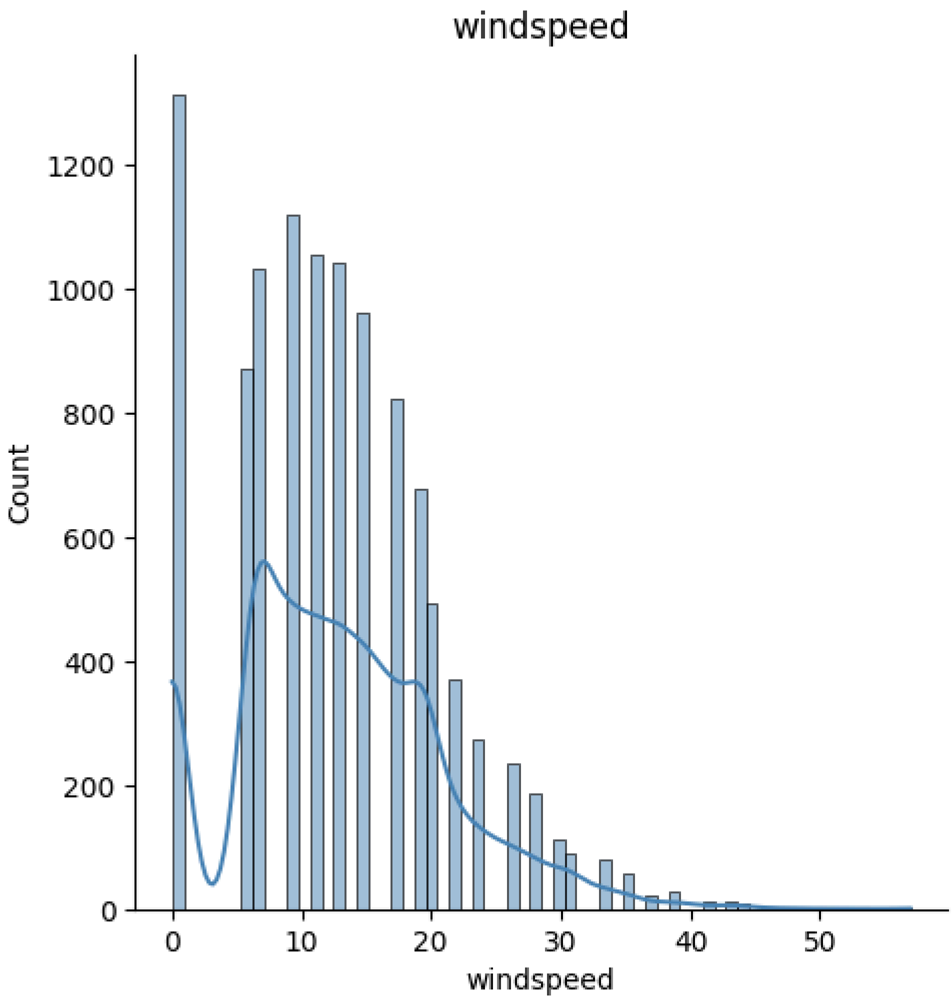
Out[73]: `Text(0.5, 1.0, 'Humidity')`



insights: humidity lies mostly > 40 and < 100 or 90 at humidity scale

In [74]:
```
sns.displot(data['windspeed'],color='steelblue',kde=True)
plt.title("windspeed")
```

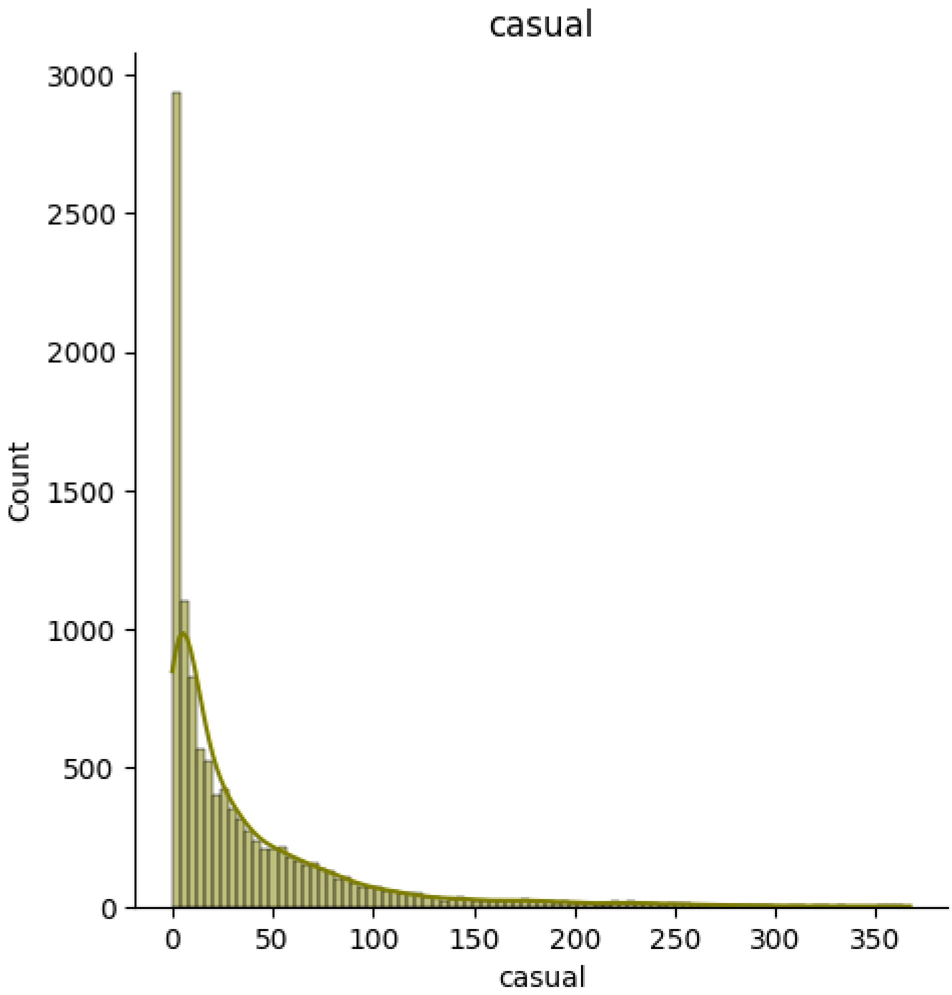
Out[74]: Text(0.5, 1.0, 'windspeed')



Insights: most of the bookings done when there is no speed or less speed and as windspeed keep on increasing the rented bike entries also steeply decreases

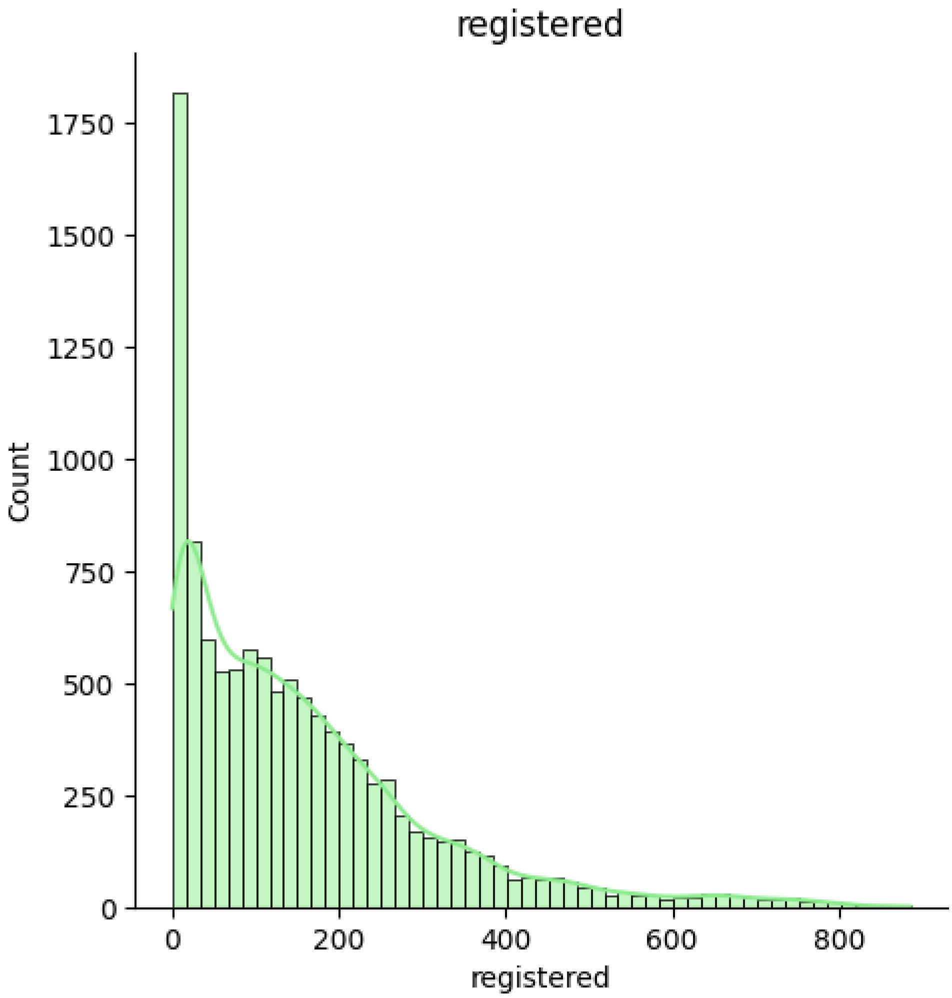
In [75]:
```
sns.displot(data['casual'],color='olive',kde=True)
plt.title("casual")
```

Out[75]:
```
Text(0.5, 1.0, 'casual')
```



In [76]:
```
sns.displot(data['registered'],color='lightgreen',kde=True)
plt.title("registered")
```

Out[76]:
```
Text(0.5, 1.0, 'registered')
```



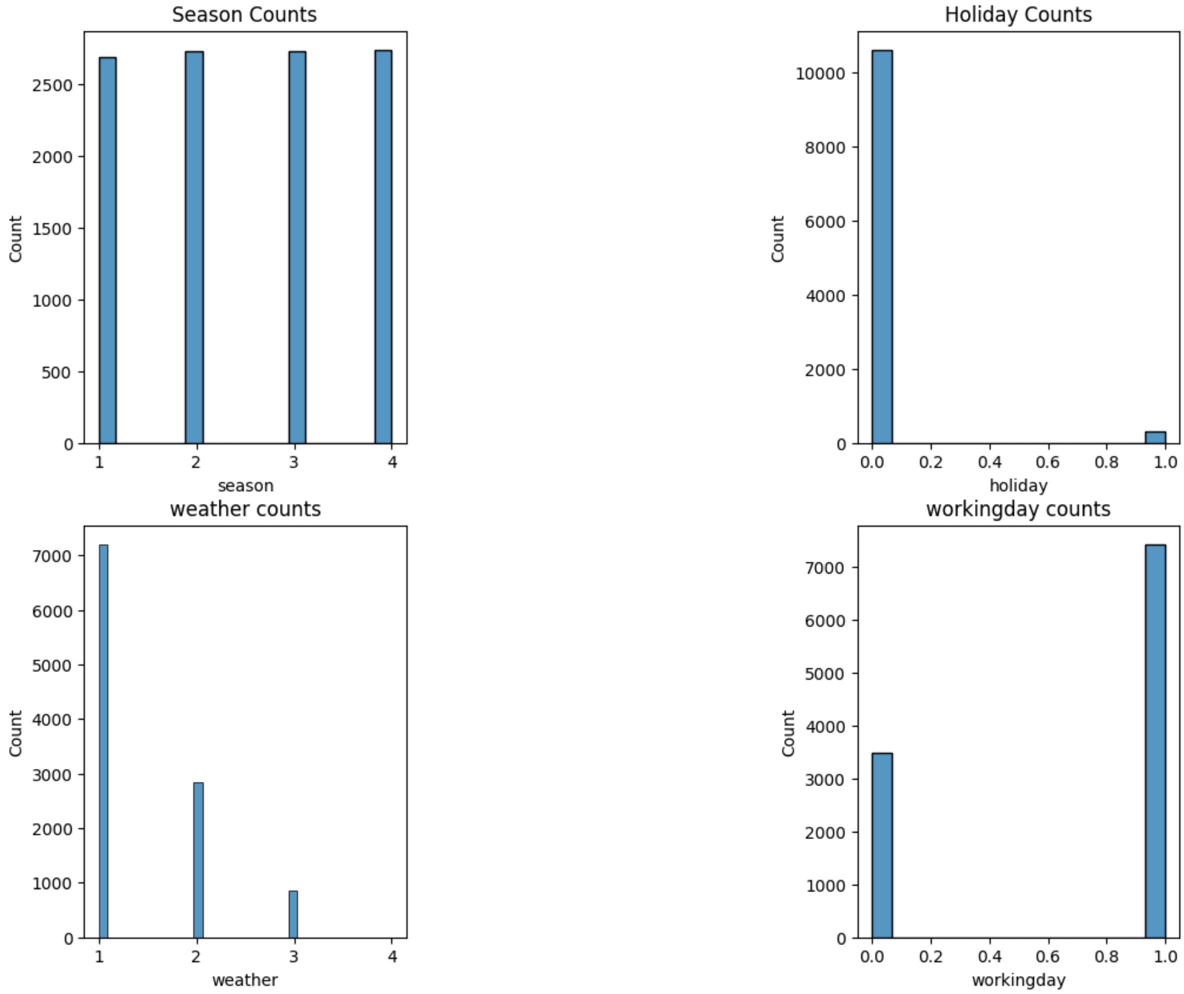
Insights: from above insights it is clear the number of registered and casual users keep on decreasing as the number of entries increases within short span of duration

Categorical

(season,holiday,workingday,weather)

In [14]:
```
plt.figure(figsize=(12,10))
plt.subplot(231)
sns.histplot(data['season'])
plt.title('Season Counts')
plt.subplot(233)
sns.histplot(data['holiday'])
plt.title('Holiday Counts')
plt.subplot(236)
sns.histplot(data['workingday'])
plt.title('workingday counts')
plt.subplot(234)
sns.histplot(data['weather'])
plt.title('weather counts')
```

Out[14]:
```
Text(0.5, 1.0, 'weather counts')
```



Insights:

1. people prefer yulu bikes from all seasons (1: spring, 2: summer, 3: fall, 4: winter)

2. mostly people rented bikes when the workingday was 1 then 0

3. people preferred bikes during weather 1 { which is Clear, Few clouds, partly cloudy, partly cloudy} then

   2 {Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist} then

   3 {Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds} 1 -- > 2 -- > 3

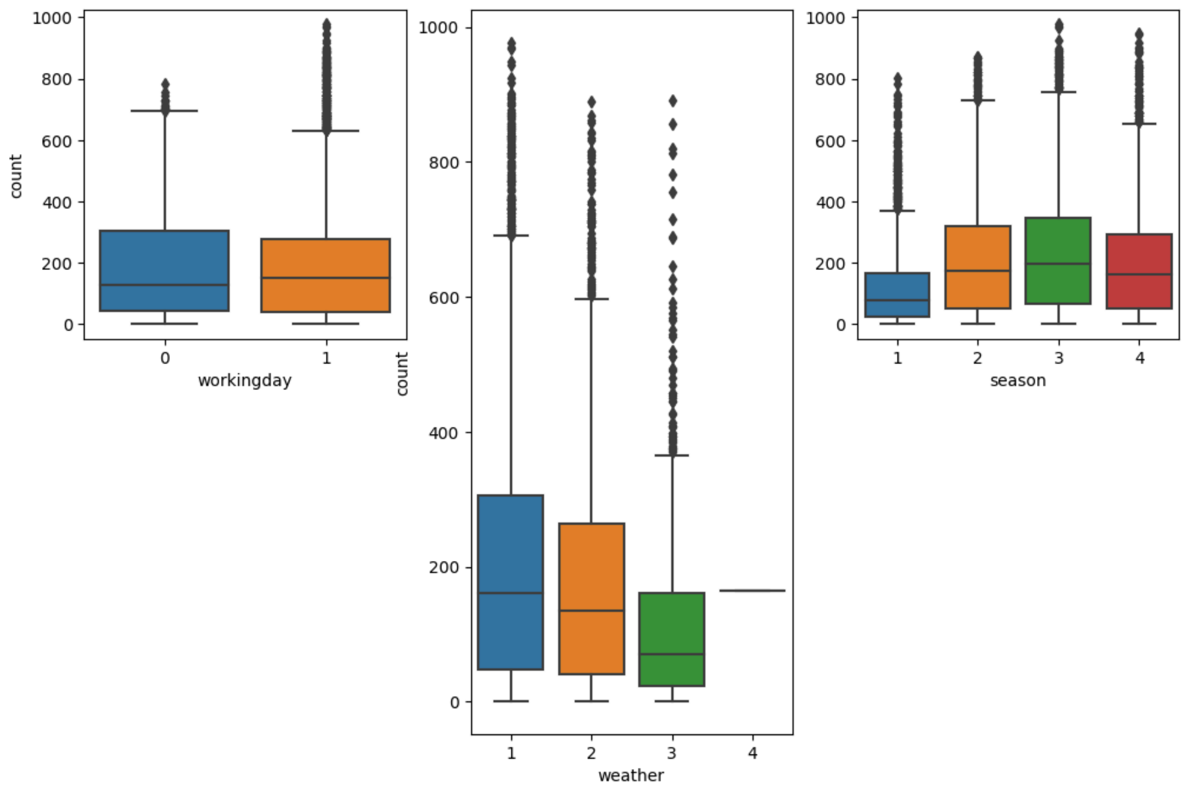
Bivariate Analysis

(Relationships between important variables such as workday and count, season and count, weather and count

In [15]:
```
plt.figure(figsize=(12,8))
plt.subplot(2,3,1)
sns.boxplot(data=data,x=data['workingday'],y=data['count'])
plt.subplot(2,3,3)
sns.boxplot(data=data,x=data['season'],y=data['count'])
```

```python
plt.subplot(1,3,2)
sns.boxplot(data=data,x=data['weather'],y=data['count'])
```

Out[15]: <Axes: xlabel='weather', ylabel='count'>



from above graph we can see there are ample of outlier , for the observation we can check for some sample let say for weather , to detect it we can do further process

1. the median of number of rented bikes are equals for both 0 and 1 working day
2. For weather 1 there is a highest median then for 2 weather and then 3
3. for season 2 and 3 the median of yulu rented is higher than season1 and season 4

checking **Weather outliers** when weather is

```
1: Clear, Few clouds, partly cloudy, partly cloudy
```

In [79]:
```python
import numpy  as np
```

In [81]:
```python
w_25 = np.percentile(data[data['weather']==1]['count'],25)
## it is 25% percentile value of bike rented on working day for 50 percentile and 7
w_50 = np.percentile(data[data['weather']==1]['count'],50)
w_75 = np.percentile(data[data['weather']==1]['count'],75)
w_25,w_50,w_75
```

Out[81]: (48.0, 161.0, 305.0)

In [82]:
```python
IQR = w_75- w_25 # Q3- Q1
IQR
```

Out[82]: 257.0

In [83]:
```python
upper_line = w_75 + 1.5 * IQR
lower_line = max(w_25 - 1.5 * IQR,0)
lower_line,upper_line
```

Out[83]:
```
(0, 690.5)
```

checking how many outlier in rented bikes are there for weather 1

In [84]:
```
data_outlier_weather_1_rented_bike = data[data['weather']==1]['count'][data[data['w
data_outlier_weather_1_rented_bike.count()
```

Out[84]:
```
160
```

From above observation we can say **that there are 160 outliers in rented bike on weather is 1** i.e o **1: Clear, Few clouds, partly cloudy, partly cloudy** similary we can check for others outliers too

• Try establishing a relation between the dependent and independent variable (Dependent "Count" & Independent: Workingday, Weather, Season etc)

Checking its correlation and observing its relationship among them

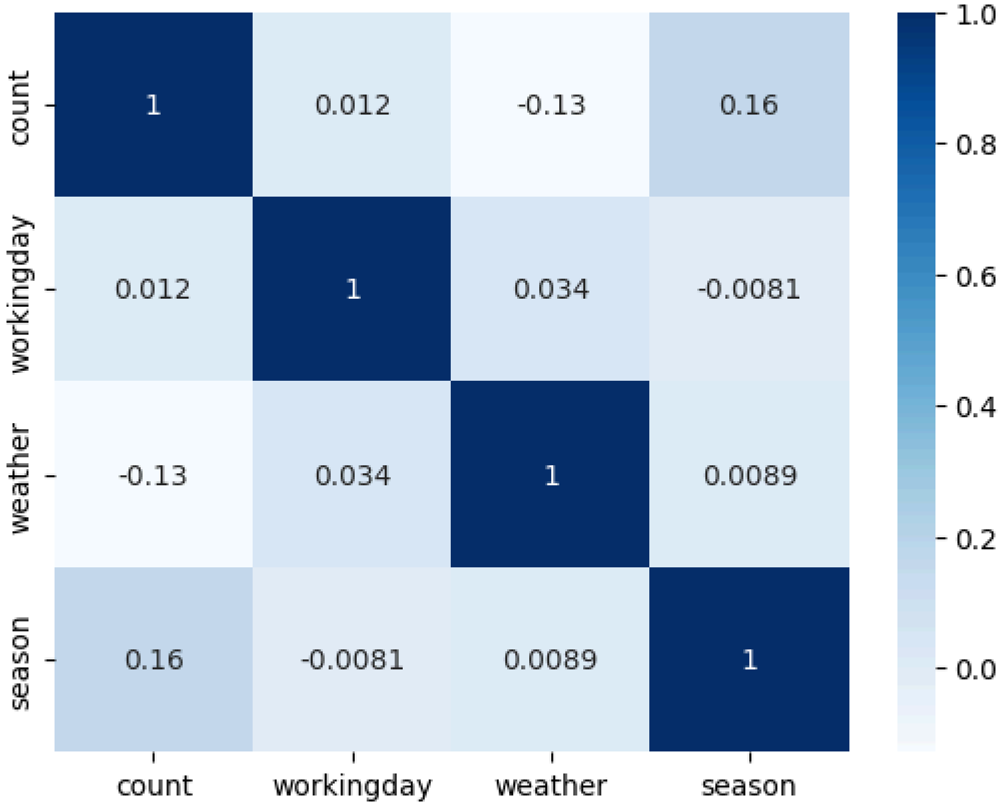
In [16]:
```
data[['count','workingday','weather','season']].corr()
```

Out[16]:

	count	workingday	weather	season
count	1.000000	0.011594	-0.128655	0.163439
workingday	0.011594	1.000000	0.033772	-0.008126
weather	-0.128655	0.033772	1.000000	0.008879
season	0.163439	-0.008126	0.008879	1.000000

In [17]:
```
sns.heatmap(data[['count','workingday','weather','season']].corr(),cmap ='Blues',ar
```

Out[17]:
```
<Axes: >
```



In [18]:
```python
data_rel = data[['count','workingday','weather','season']]
data_rel
```

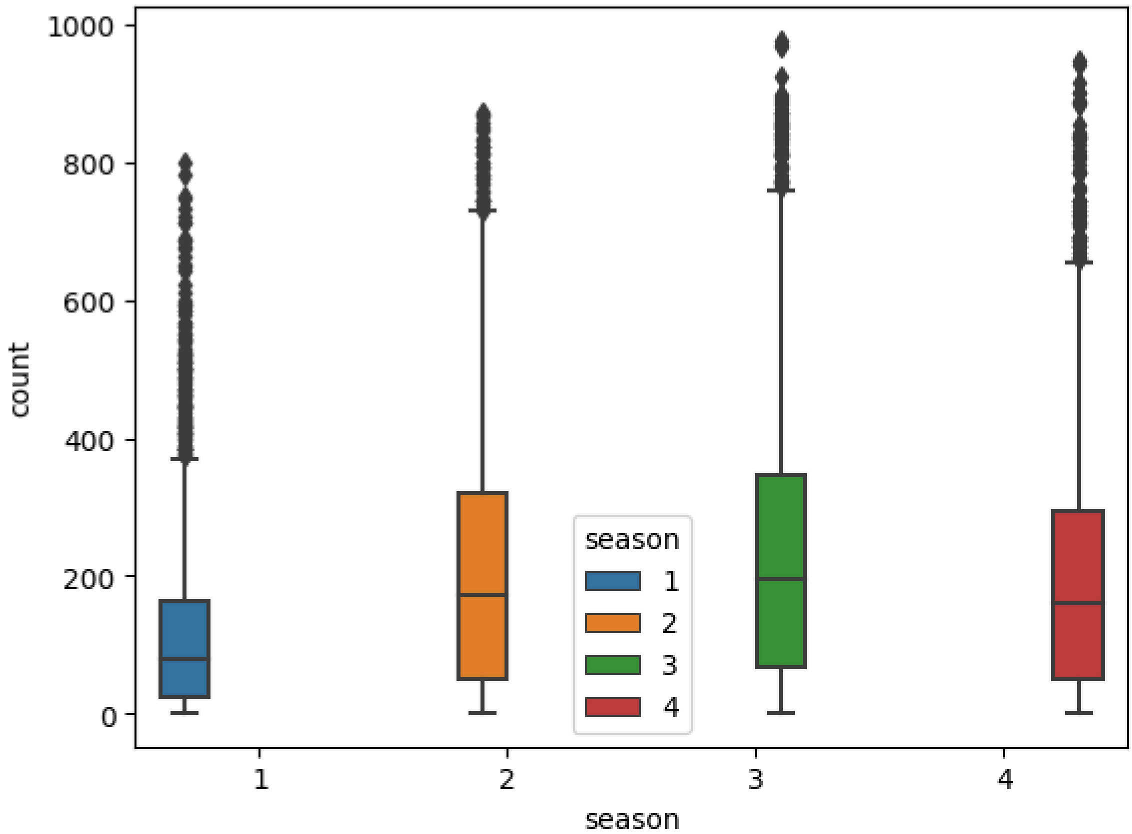
Out[18]:

	count	workingday	weather	season
0	16	0	1	1
1	40	0	1	1
2	32	0	1	1
3	13	0	1	1
4	1	0	1	1
...	...	...	...	...
10881	336	1	1	4
10882	241	1	1	4
10883	168	1	1	4
10884	129	1	1	4
10885	88	1	1	4

10886 rows × 4 columns

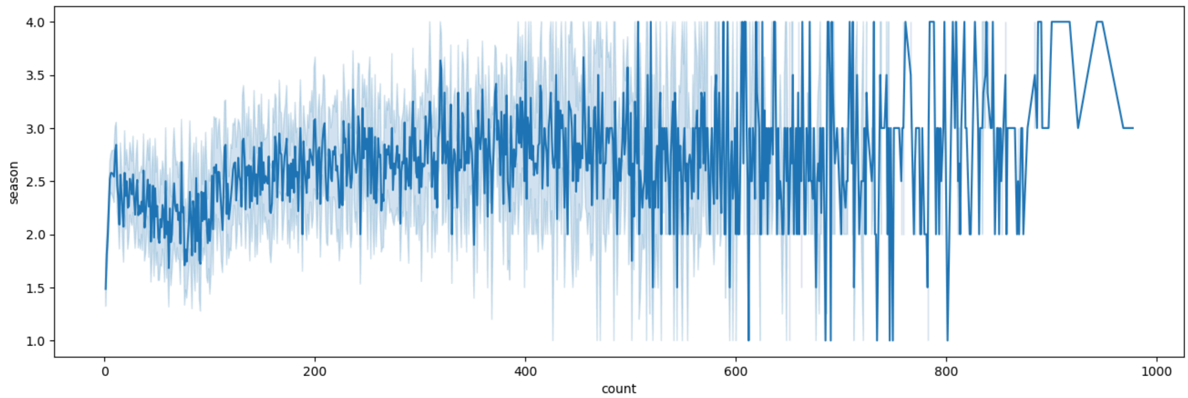
In [19]:
```python
sns.boxplot(y=data_rel['count'],x=data_rel['season'],hue = data_rel['season'] )
```

Out[19]:
```
<Axes: xlabel='season', ylabel='count'>
```



In [20]:
```
plt. figure(figsize=(16, 5))
sns.lineplot(x=data_rel['count'],y=data_rel['season'])
```

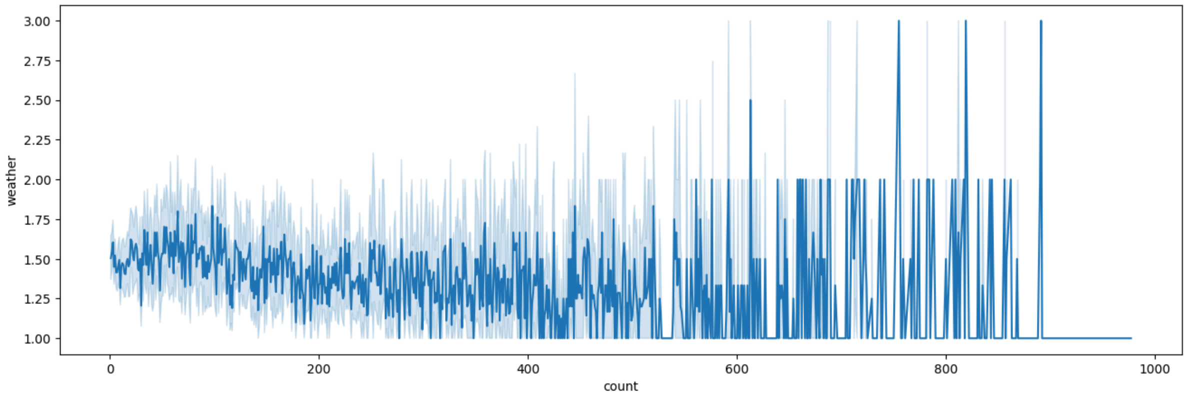
Out[20]: <Axes: xlabel='count', ylabel='season'>



Insights:from above graph we can say that the count mostly lies when the season is above 2

In [21]:
```
plt. figure(figsize=(16, 5))
sns.lineplot(x=data_rel['count'],y=data_rel['weather'])
```

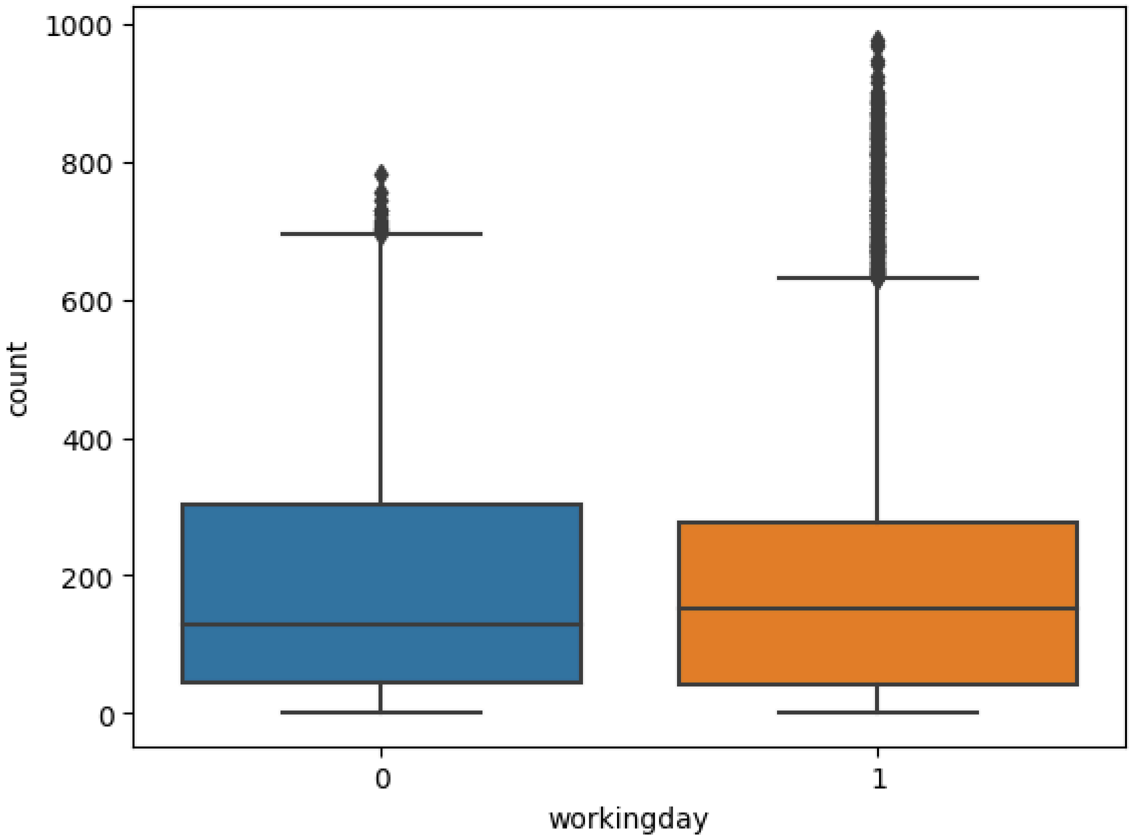
Out[21]: <Axes: xlabel='count', ylabel='weather'>



insights: most of the bikes rented are when weather is 2 or 1 and bike renting rarely occurs for 3

In [22]:
```
sns.boxplot(y=data_rel['count'],x=data_rel['workingday'])
```

Out[22]:
```
<Axes: xlabel='workingday', ylabel='count'>
```



Insights: the median of number of bike rented lies parallel for 0 and 1 working day

In [23]:
```
data_workingday = pd.crosstab(columns =data_rel['workingday'],index=data_rel['count
data_workingday
```

Out[23]:

workingday	0	1
count		
1	21	84
2	25	107
3	27	117
4	27	122
5	35	134
...	...	...
943	0	1
948	0	1
968	0	1
970	0	1
977	0	1

822 rows × 2 columns

In [24]:
```
data_workingday = data_workingday.reset_index()
data_workingday
```

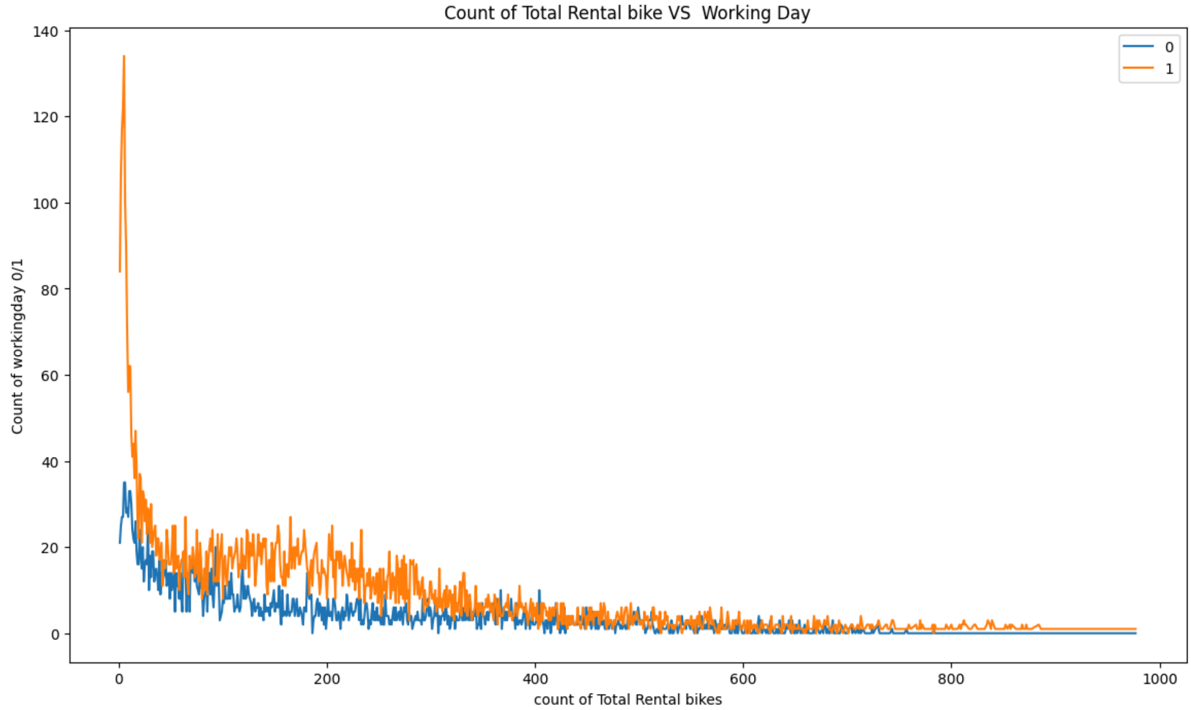
Out[24]:

workingday	count	0	1
0	1	21	84
1	2	25	107
2	3	27	117
3	4	27	122
4	5	35	134
...	...	...	...
817	943	0	1
818	948	0	1
819	968	0	1
820	970	0	1
821	977	0	1

822 rows × 3 columns

In [25]:
```
plt.figure(figsize=(14,8))
plt.plot(data_workingday['count'],data_workingday[0])
plt.plot(data_workingday['count'],data_workingday[1])
plt.ylabel("Count of workingday 0/1")
plt.xlabel('count of Total Rental bikes')
plt.title("Count of Total Rental bike VS  Working Day")
plt.legend(['0','1'])
```

Out[25]:
```
<matplotlib.legend.Legend at 0x7fcf42bf9b70>
```



Insights:

1. for 0 working day or we can say holiday or weekend rented bike counts were maximum for range of bike rented in between 0-200 lies around 20- 40
2. for weekday or no holiday that is 1 working day the bike count ranges from 0- 200 have maximum in renting 20- 140 and it keep on decreasing as the number of bike rented on 1 workingday increase that is if count of rental bikes increases to 600-800-100 then there is less chance of booking which had happened

Select an appropriate test to check whether:

o Working Day has effect on number of electric cycles rented

o No. of cycles rented similar or different in different seasons

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

o Weather is dependent on season (check between 2 predictor variable)

Working Day has effect on number of electric cycles rented

workingday: if day is neither weekend nor holiday is 1, otherwise is 0.

count: count of total rental bikes including both casual and registered

this a case of **categorical(2- category) V/S Numerical columns** so we will consider

2-Sample-Ttest

In [26]:
```
data_wd = data[['count','workingday']]
data_wd
```

Out[26]:

	count	workingday
0	16	0
1	40	0
2	32	0
3	13	0
4	1	0
...	...	...
10881	336	1
10882	241	1
10883	168	1
10884	129	1
10885	88	1

10886 rows × 2 columns

In [97]:
```
data_wd.groupby(['workingday'])['count'].aggregate({'sum'})
```

Out[97]:

workingday	sum
0	654872
1	1430604

insights: Number of bike rented is is more on working day 1 i.e, 1430604 but on weekend and holidays it is less which is 654872

In [27]:
```
m0 = data_wd.loc[data['workingday']==0]
m1 = data_wd.loc[data['workingday']==1]
```

In [28]:
```
print(f"number of rental bike mean of m0 is : {m0['count'].mean()}")
print(f"number of rental bike mean of m1 is : {m1['count'].mean()}")
```

```
number of rental bike mean of m0 is : 188.50662061024755
number of rental bike mean of m1 is : 193.01187263896384
```

To check whether above mean are statically different or not we would do 2 sample - t test

Hypothesis Testing: 2 Sample -t test

Null Hypothesis (H0):

number of rental bike mean on working day 0 $=$ number of rental bike mean on working day 0

Alternative Hypothesis (H1):

number of rental bike mean on working day 0 $! =$ number of rental bike mean on working day 0

In [29]:
```
from scipy.stats import ttest_ind
```

In [30]:
```
t_stat, p_value = ttest_ind(m0,m1,alternative='two-sided')
print("t_stat :",t_stat)
print("p_value :",p_value)
```

```
t_stat : [-1.20962774        -inf]
p_value : [0.22644804 0.        ]
/usr/local/lib/python3.10/dist-packages/scipy/stats/_axis_nan_policy.py:551: RuntimeWarning: Precision loss occurred in moment calculation due to catastrophic cancellation. This occurs when the data are nearly identical. Results may be unreliable.
  res = hypotest_fun_out(*samples, axis=axis, **kwds)
```

In [31]:
```
p_value[0] < 0.05
```

Out[31]:
```
False
```

since p_value is not less than alpha so **null hypothesis is not rejected** hence mean of rental bikes on 0 working day is **statistically similar** to mean of rental bike on 1 working day which concludes **that Working day has no effect on the number of electric cycles rented**

2) to check if No. of cycles rented is similar or different in different 1. weather 2. season (10 points)

Checking for weather

In [32]:
```
data_weather = data[['count','weather']]
data_weather
```

Out[32]:

	count	weather
0	16	1
1	40	1
2	32	1
3	13	1
4	1	1
...	...	...
10881	336	1
10882	241	1
10883	168	1
10884	129	1
10885	88	1

10886 rows × 2 columns

In [99]:
```
data_weather.groupby(['weather'])['count'].aggregate({'sum'})
```

Out[99]:

	sum
weather	
1	1476063
2	507160
3	102089
4	164

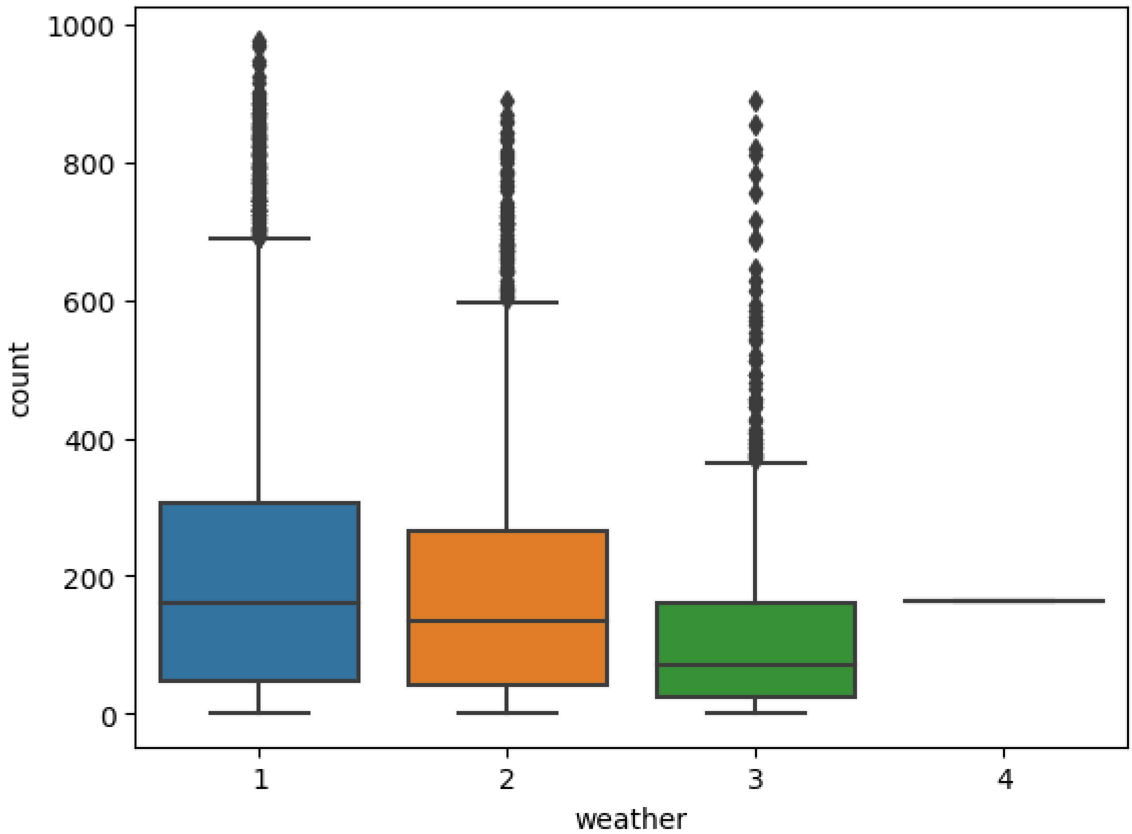
Insights: mostly people preffered to rent the bike in weather 1 with total booking of 1476063 then 2 507160 then 3 102089 then least among is 4 - 164

In [33]:
```
data_weather['weather'].unique()
```

Out[33]:
```
array([1, 2, 3, 4])
```

In [34]:
```
sns.boxplot(x='weather',y='count',data=data_weather)
```

Out[34]:
```
<Axes: xlabel='weather', ylabel='count'>
```



In [35]:
```
w1 = data_weather[data_weather['weather']==1]['count']
w2 = data_weather[data_weather['weather']==2]['count']
w3 = data_weather[data_weather['weather']==3]['count']
w4 = data_weather[data_weather['weather']==4]['count']
```

Checking whether The rented bikes for every weather follow Annova assumptions Annova Assumption

1.Data should be gaussian - Q-Q plot

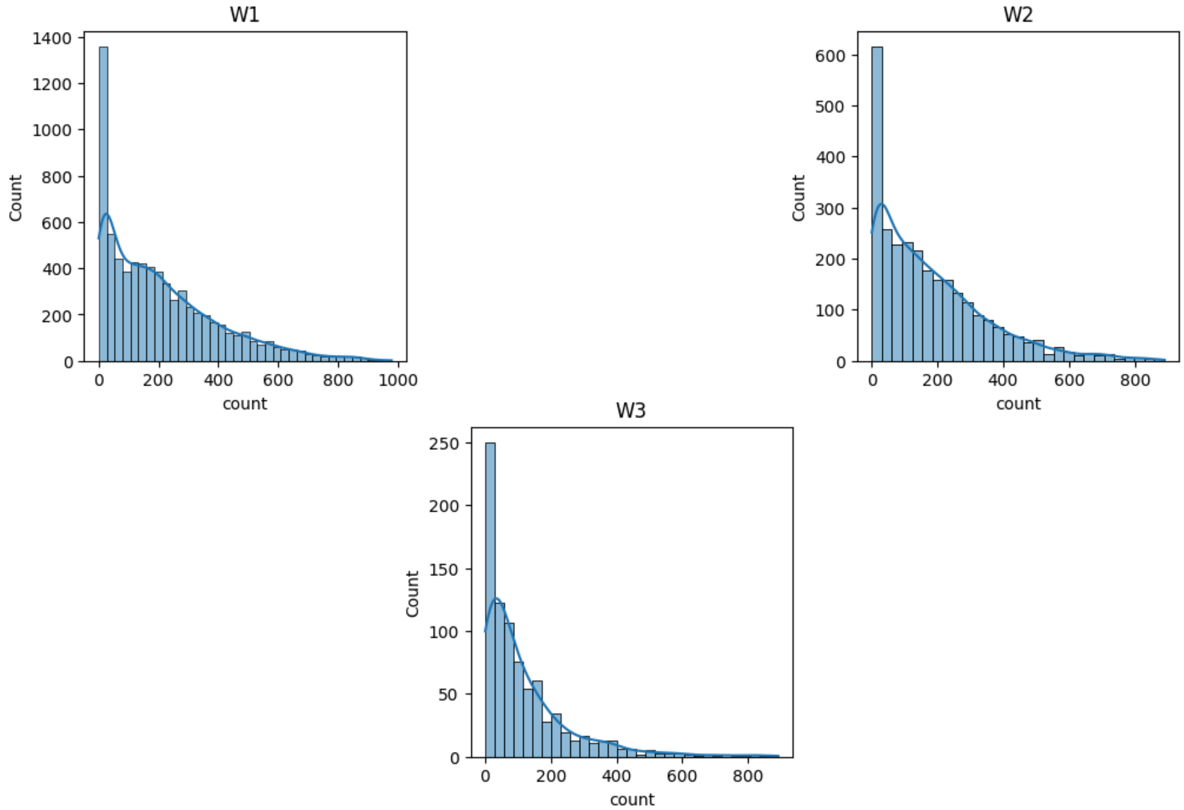
2.independence

3.equal variance in different groups - levene test

In [36]:
```
from statsmodels.graphics.gofplots import qqplot
```

In [37]:
```
plt.figure(figsize=(12,8))
plt.subplot(231)
sns.histplot(w1,kde=True)
plt.title("W1")
plt.subplot(233)
sns.histplot(w2,kde=True)
plt.title("W2")
plt.subplot(235)
sns.histplot(w3,kde=True)
plt.title("W3")
```

Out[37]:
```
Text(0.5, 1.0, 'W3')
```

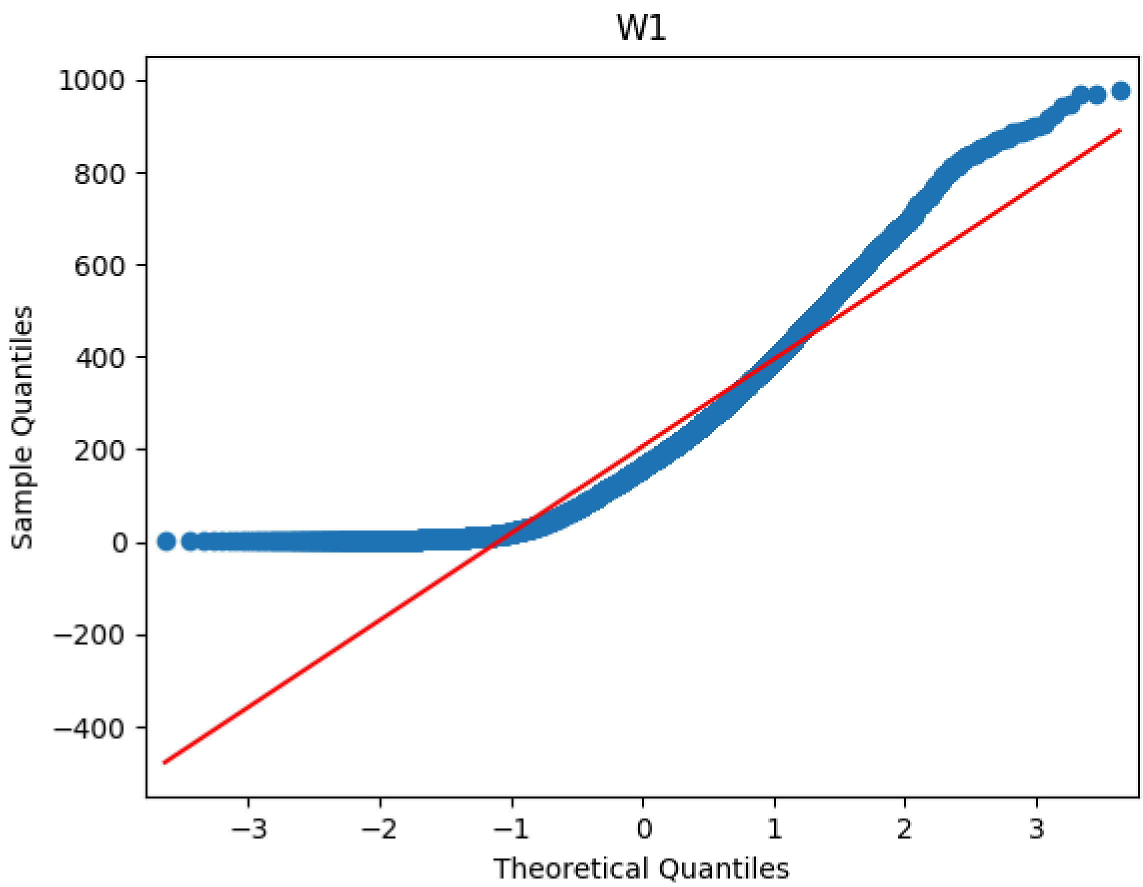


Checking q-q plot for gaussian distribution check

In [38]:
```
plt.figure(figsize=(5,3))
qqplot(w1,line='s')
plt.title("W1")
```

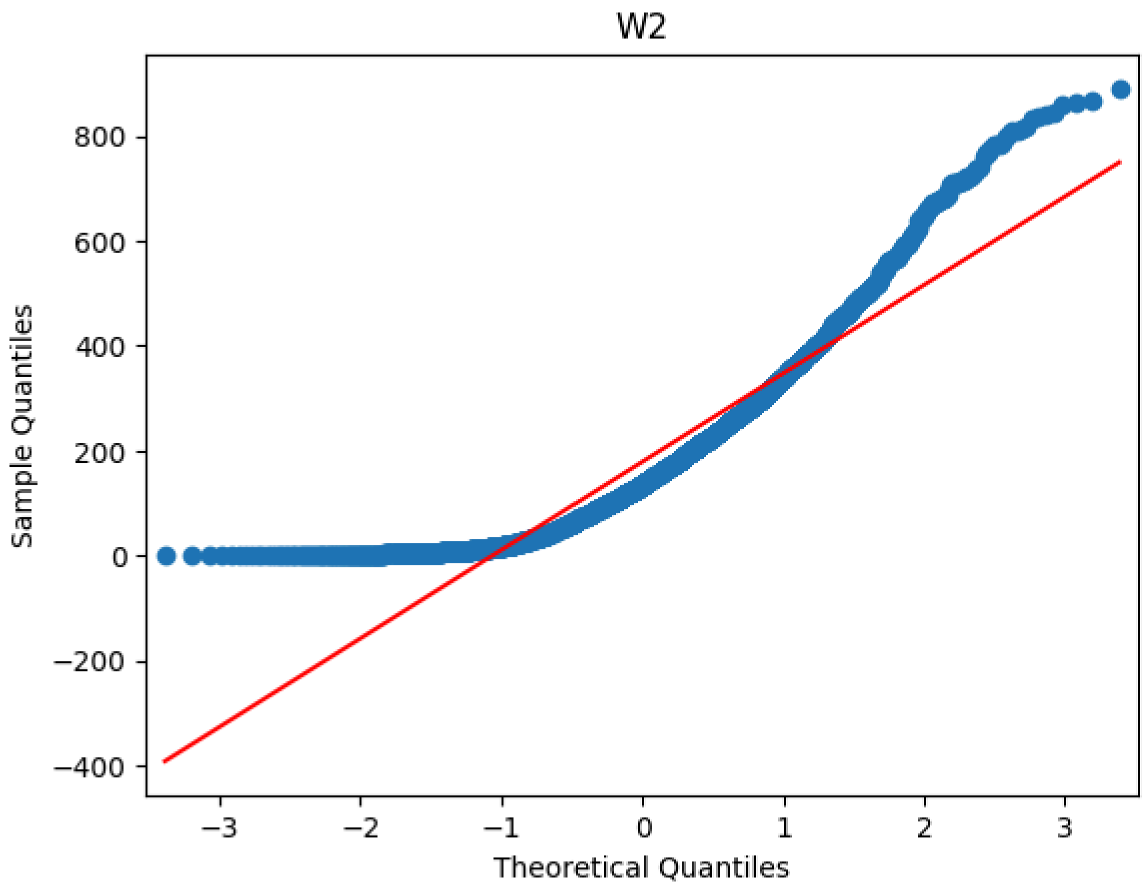
Out[38]:
```
Text(0.5, 1.0, 'W1')
```

```
<Figure size 500x300 with 0 Axes>
```



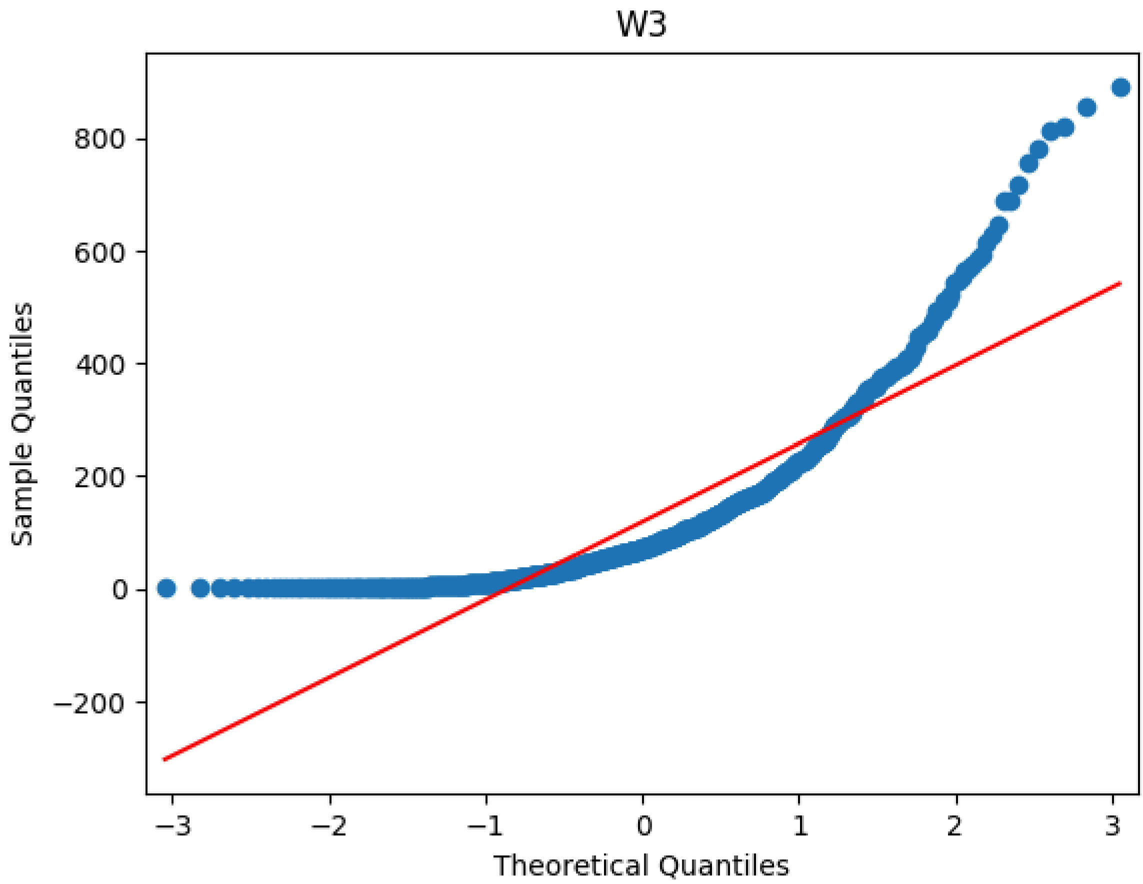
In [39]:
```
qqplot(w2,line='s')
plt.title("W2")
```

Out[39]: Text(0.5, 1.0, 'W2')



In [40]:
```
qqplot(w3,line='s')
plt.title("W3")
```

Out[40]: Text(0.5, 1.0, 'W3')



Checking Variance are equal for satisfying assumptions we will do levene test

H0: Variances are same

Ha: variances are different

alpha = 0.05

In [41]:
```
from scipy.stats import levene
```

In [42]:
```
s_stat, p_value_variance = levene(w1,w2,w3,w4)
print("s_stats :",s_stat)
print("p_value :",p_value_variance)
```

```
s_stats : 54.85106195954556
p_value : 3.504937946833238e-35
```

since p_value < 0.05 so we reject null hypothesis so variance are different which doesn't satisfy Annova assumptions so we conclude further as given below

From above Q-Q plot and Variance test we can say the data doesn't follow Annova Assumption of Gaussian distribution so we have to Use alternative i.e, KRUSKAL Test

In [43]:
```
print("mean of bike rented on weather = 1 :",w1.mean())
print("mean of bike rented on weather = 2 :",w2.mean())
print("mean of bike rented on weather = 3 :",w3.mean())
print("mean of bike rented on weather = 4 :",w4.mean())
```

```
mean of bike rented on weather = 1 : 205.23679087875416
mean of bike rented on weather = 2 : 178.95553987297106
mean of bike rented on weather = 3 : 118.84633294528521
mean of bike rented on weather = 4 : 164.0
```

checking statistical differnce in mean we have to use Annova hypothesis testing

Hypothesis Testing: Kruskal wallis Test

Null Hypothesis (H0):

mean of bike rented on different weather are equal

Alternative Hypothesis (H1): mean of bike rented on different weather are not equal

Alpha = 0.05

In [44]:
```
from scipy.stats import kruskal
```

In [45]:
```
s_stats , p_value_weather = kruskal(w1,w2,w3,w4)
print("s_stats : ",s_stats)
print("p_value_weather :", p_value_weather)
```

```
s_stats :  205.00216514479087
p_value_weather : 3.501611300708679e-44
```

In [46]:
```
if p_value_weather < 0.05:
  print("Reject Null Hypothesis")
else:
  print("accpet Null Hypothesis")
```

```
Reject Null Hypothesis
```

From Above result of Hypothesis testing we can say mean of number of bike rented on different weather are not equal which concludes **that Number of rented bikes are significantly different for different weather**

Rented Bike V/S Seasons

In [47]:
```
data_season = data[['count','season']]
data_season
```

Out[47]:

	count	season
0	16	1
1	40	1
2	32	1
3	13	1
4	1	1
...	...	...
10881	336	4
10882	241	4
10883	168	4
10884	129	4
10885	88	4

10886 rows × 2 columns

In [98]:
```
data_season.groupby(['season'])['count'].aggregate({'sum'})
```

Out[98]:

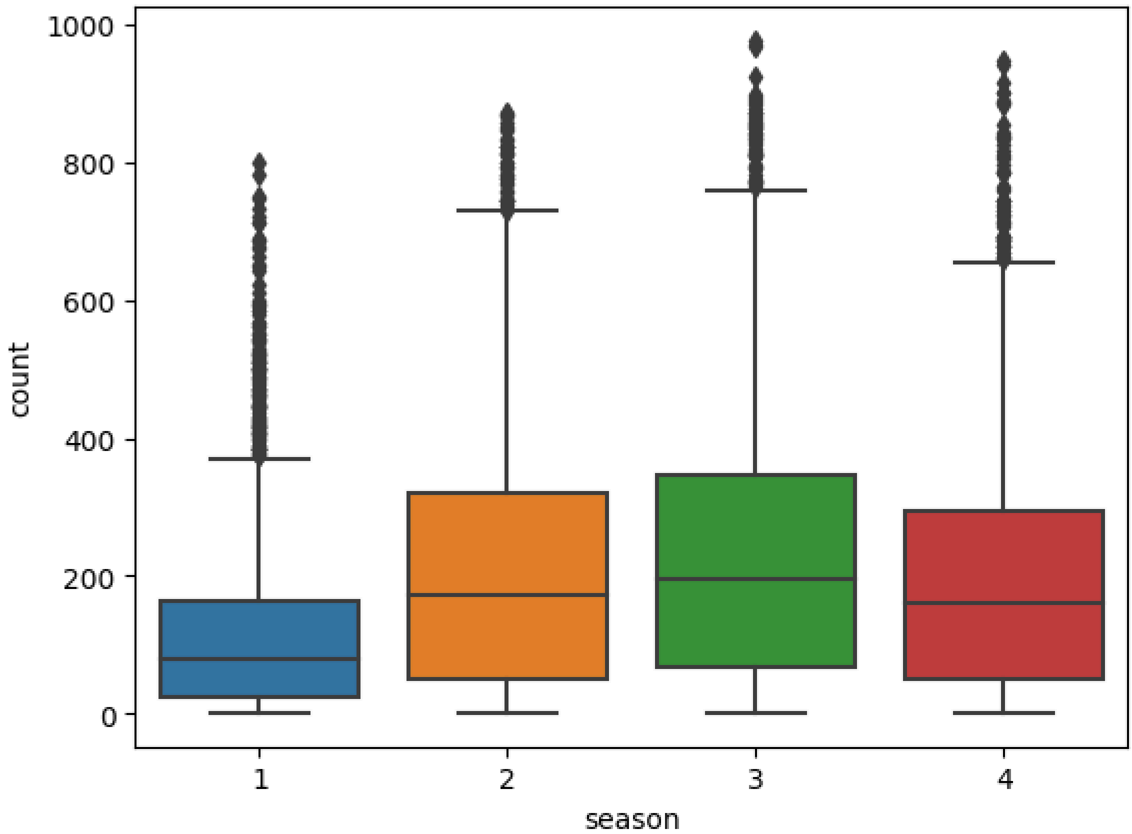
	sum
season	
1	312498
2	588282
3	640662
4	544034

In [48]:
```
data_season['season'].unique()
```

Out[48]:
```
array([1, 2, 3, 4])
```

In [49]:
```
sns.boxplot(x='season',y='count',data=data_season)
```

Out[49]:
```
<Axes: xlabel='season', ylabel='count'>
```



In [50]:
```
s1 = data_season[data_season['season']==1]['count']
s2 = data_season[data_season['season']==2]['count']
s3 = data_season[data_season['season']==3]['count']
s4 = data_season[data_season['season']==4]['count']
```

Checking whether The rented bikes for every season follow Annova assumptions Annova Assumption

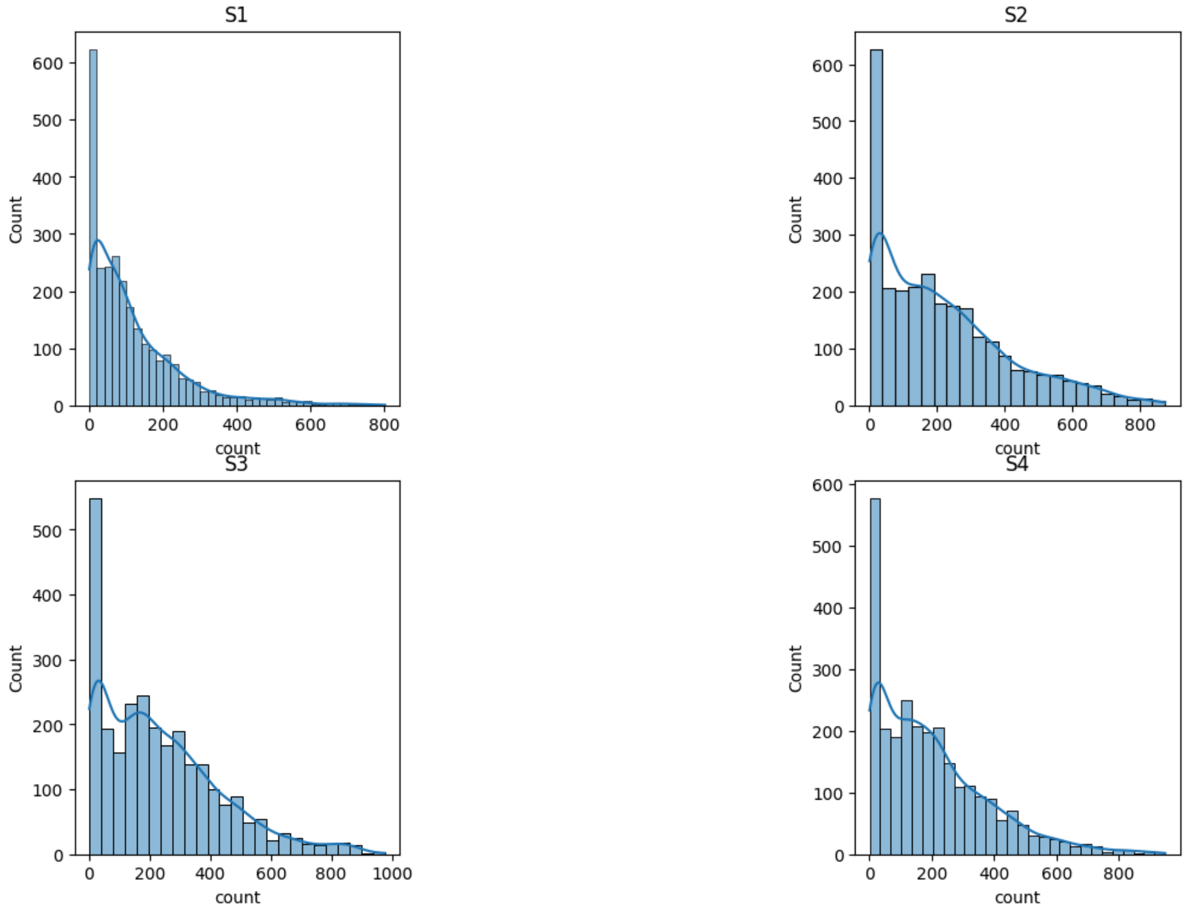
1.Data should be gaussian - Q-Q plot

2.independence

3.equal variance in different groups - levene test

In [51]:
```
plt.figure(figsize=(12,9))
plt.subplot(231)
sns.histplot(s1,kde=True)
plt.title("S1")
plt.subplot(233)
sns.histplot(s2,kde=True)
plt.title("S2")
plt.subplot(234)
sns.histplot(s3,kde=True)
plt.title("S3")
plt.subplot(236)
sns.histplot(s4,kde=True)
plt.title("S4")
```

Out[51]:
```
Text(0.5, 1.0, 'S4')
```

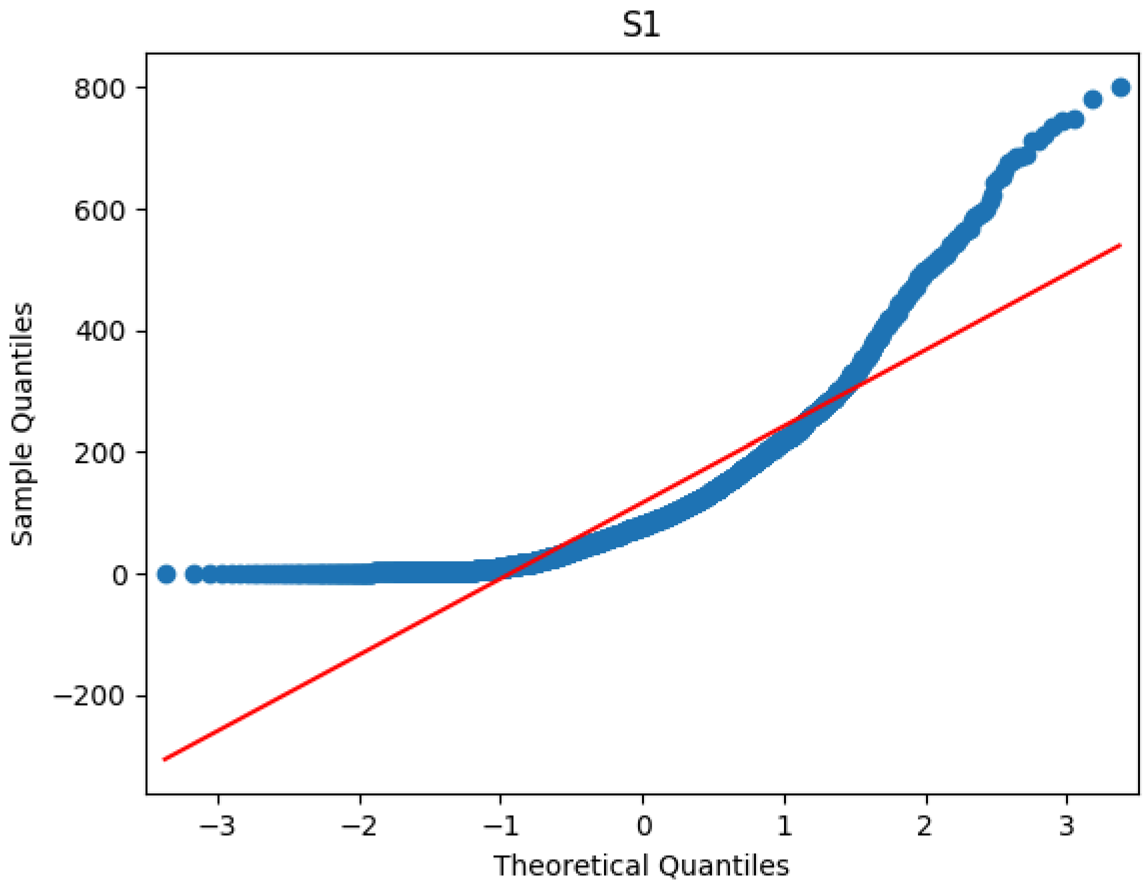


Checking q-q plot for gaussian distribution check

In [52]:
```
plt.figure(figsize=(5,3))
qqplot(s1,line='s')
plt.title("S1")
```

Out[52]:
```
Text(0.5, 1.0, 'S1')
```

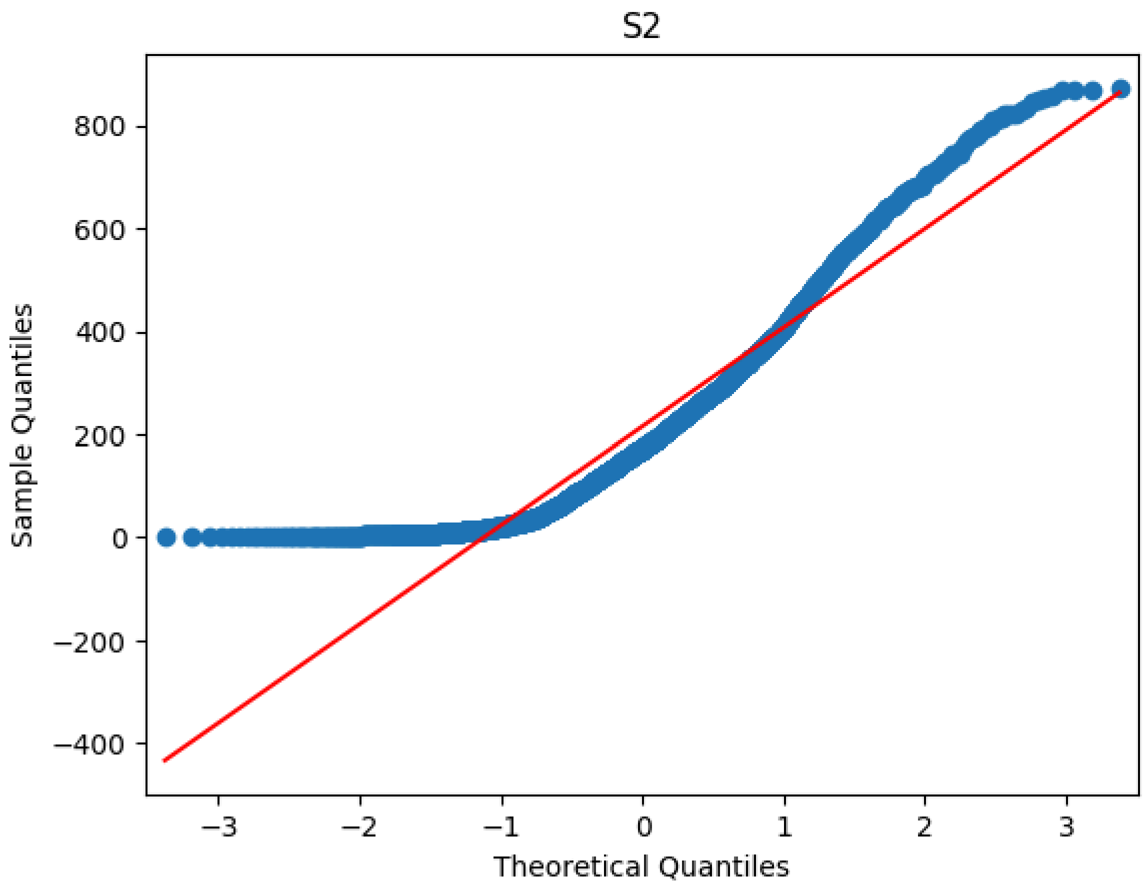
```
<Figure size 500x300 with 0 Axes>
```



In [53]:
```
plt.figure(figsize=(5,3))
qqplot(s2,line='s')
plt.title("S2")
```

Out[53]:
```
Text(0.5, 1.0, 'S2')
```

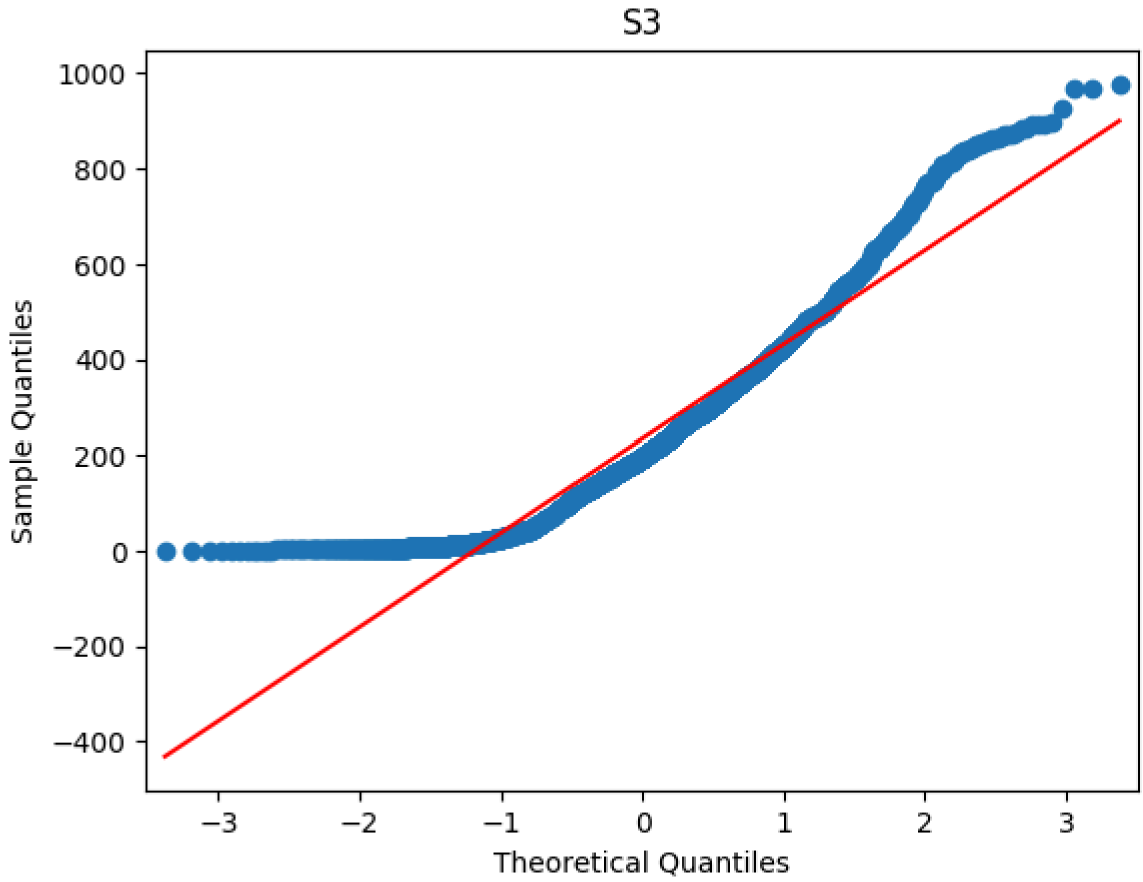
```
<Figure size 500x300 with 0 Axes>
```



In [54]:
```
plt.figure(figsize=(5,3))
qqplot(s3,line='s')
plt.title("S3")
```

Out[54]:
```
Text(0.5, 1.0, 'S3')
```

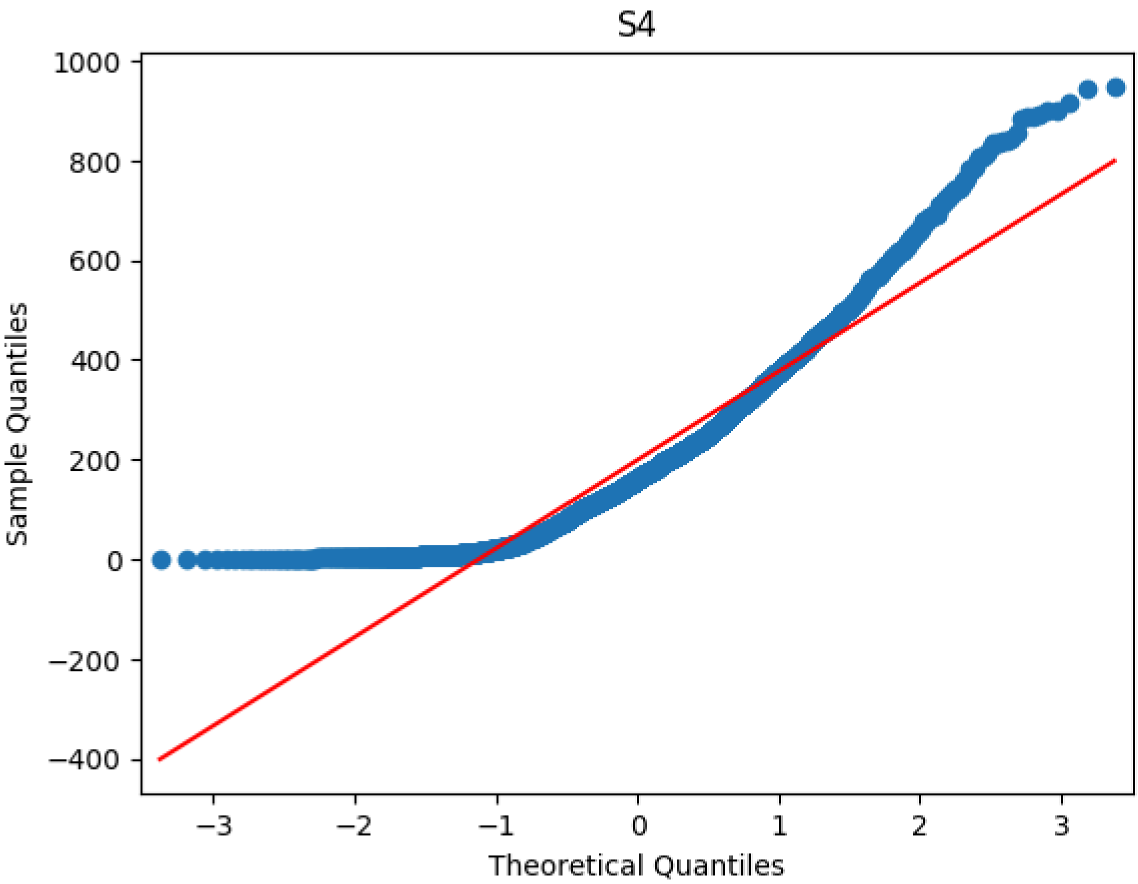
```
<Figure size 500x300 with 0 Axes>
```



In [55]:
```
plt.figure(figsize=(5,3))
qqplot(s4,line='s')
plt.title("S4")
```

Out[55]:
```
Text(0.5, 1.0, 'S4')
```

```
<Figure size 500x300 with 0 Axes>
```



Checking Variance are equal for satisfying assumptions we will do levene test

H0: Variances are same

Ha: variances are different

alpha = 0.05

In [56]:
```
s_stat, p_value_variance = levene(s1,s2,s3,s4)
print("s_stats :",s_stat)
print("p_value :",p_value_variance)
```

```
s_stats : 187.7706624026276
p_value : 1.0147116860043298e-118
```

In [56]:

since p_value < 0.05 so we reject null hypothesis so variance are different which doesn't satisfy Annova assumptions so we conclude further as given below

From above Q-Q plot and Variance test on season data we can say the data doesn't follow Annova Assumption of Gaussian distribution so we have to Use alternative i.e, KRUSKAL Test

In [57]:
```
print("mean of bike rented on season = 1 :",s1.mean())
print("mean of bike rented on season = 2 :",s2.mean())
print("mean of bike rented on season = 3 :",s3.mean())
print("mean of bike rented on season = 4 :",s4.mean())
```

```
mean of bike rented on season = 1 : 116.34326135517499
mean of bike rented on season = 2 : 215.25137211855105
mean of bike rented on season = 3 : 234.417124039517
mean of bike rented on season = 4 : 198.98829553767374
```

Hypothesis Testing: Kruskal wallis Test

Null Hypothesis (H0):

mean of bike rented on different season are equal

Alternative Hypothesis (H1): mean of bike rented on different season are not equal

Alpha = 0.05

In [58]:
```
s_stats , p_value_season = kruskal(s1,s2,s3,s4)
print("s_stats : ",s_stats)
print("p_value_season :",p_value_season)
```

```
s_stats :  699.6668548181988
p_value_season : 2.479008372608633e-151
```

In [59]:
```
if p_value_season < 0.05:
  print("Reject Null Hypothesis")
else:
  print("accpet Null Hypothesis")
```

```
Reject Null Hypothesis
```

From Above result of Hypothesis testing we can say mean of number of bike rented on different season are not equal which concludes **that Number of rented bikes are significantly different for different season**

3.) Chi-square test to check if Weather is dependent on the season

Weather V/s Season { categorical - categorical columns}

In [60]:
```
data_w_s = data[['weather','season']]
data_w_s
```

Out[60]:

	weather	season
0	1	1
1	1	1
2	1	1
3	1	1
4	1	1
...	...	...
10881	1	4
10882	1	4
10883	1	4
10884	1	4
10885	1	4

10886 rows × 2 columns

In [61]:
```
s_w = pd.crosstab(data_w_s['weather'],data_w_s['season'])
s_w
```

Out[61]:

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

since the weather and season data are categorical columns so we have to use **chisquare test** for checking

$$Null Hypothesis$$

H0 : weather is independent of season

$$Alternative Hypothesis$$

H1: weather is dependent on season

alpha = 0.05

In [62]:
```
from scipy.stats import chi2_contingency
```

In [63]:
```
table = [[1759,1801,1930,1702],[715,708,604,807],[211,224,199,225],[1,0,0,0]]
stats, p_value , dof , expected = chi2_contingency(table)
print("stats:",stats)
print("p_value: ",p_value)
print("dof :",dof)
print("expected: ",expected)
```

```
stats: 49.15865559689363
p_value:  1.5499250736864862e-07
dof : 9
expected:  [[1.77454639e+03 1.80559765e+03 1.80559765e+03 1.80625831e+03]
 [6.99258130e+02 7.11493845e+02 7.11493845e+02 7.11754180e+02]
 [2.11948742e+02 2.15657450e+02 2.15657450e+02 2.15736359e+02]
 [2.46738931e-01 2.51056403e-01 2.51056403e-01 2.51148264e-01]]
```

In [64]:
```
if p_value < 0.05:
  print("Rejct the null hypothesis")
else:
  print("accept the null hypothesis")
```

```
Rejct the null hypothesis
```

from Above result we can conclude that **the null hypothesis is rejected** and further **we can say Weather is dependent on seasons**

$$Summary$$

1. people prefer yulu bikes from all seasons (1: spring, 2: summer, 3: fall, 4: winter)
2. mostly people rented bikes when the workingday was 1 then 0
3. people preferred bikes during weather

   1 { which is Clear, Few clouds, partly cloudy, partly cloudy} then

   2 {Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist} then

   3 {Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds} 1 -- > 2 -- > 3

1. the median of number of rented bikes are equals for both 0 and 1 working day

   a.) Number of bike rented is is more on working day 1 i.e, 1430604 but on weekend and holidays it is less which is 654872

2. For weather 1 there is a highest median then for 2 weather and then 3

   a) mostly people preffered to rent the bike in weather 1 with total booking of 1476063 then 2 507160 then 3 102089 then least among is 4 - 164

3. for season 2 and 3 the median of yulu rented is higher than season1 and season 4

4. for 0 working day or we can say holiday or weekend rented bike counts were maximum for range of bike rented in between 0-200 lies around 20- 40

5. for weekday or no holiday that is 1 working day the bike count ranges from 0- 200 have maximum in renting 20- 140 and it keep on decreasing as the number of bike rented on 1 workingday increase that is if count of rental bikes increases to 600-800-100 then there is less chance of booking which had happened

Hypothesis results

1. Working day has no effect on the number of electric cycles rented
2. Number of rented bikes are significantly different for different weather
3. Number of rented bikes are significantly different for different season
4. we can say Weather is dependent on seasons

Recommendations-

1.from statistical hypothesis testing we can say that workinday doesn't effect electric cycle renting behaviour

1. working day 1 have more number of bike rented so yulu firm should priortise in providing offer and accesibilities on these days

2. when the weather was 1 i.e, is Clear, Few clouds, partly cloudy, partly cloudy then booking was highest so firm can priortise there focus on these days and for more engagement of customers they should give some concession or discounted ride on other weather condition like 2, 3 or 4