```
!wget -O yulu.csv
"https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/428/original/bike_sharing.csv?1642089089"
```

```
--2023-12-04 19:24:03--
https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/428
/original/bike_sharing.csv?1642089089
Resolving d2beiqkhq929f0.cloudfront.net
(d2beiqkhq929f0.cloudfront.net)... 13.224.9.181, 13.224.9.129,
13.224.9.103, ...
Connecting to d2beiqkhq929f0.cloudfront.net
(d2beiqkhq929f0.cloudfront.net)|13.224.9.181|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 648353 (633K) [text/plain]
Saving to: ‘yulu.csv’

 yulu.csv              0%[                    ]       0  --.-KB/s
yulu.csv            100%[===================>] 633.16K  --.-KB/s    in
0.04s

2023-12-04 19:24:03 (15.0 MB/s) - ‘yulu.csv’ saved [648353/648353]
```

Problem statement

Yulu is a micro-mobility service provider in India that offers shared electric cycles for daily commute. It aims to reduce traffic congestion and provide safe and sustainable commuting options. Yulu zones are located at convenient places for easy access. However, Yulu has faced a decline in its revenues and wants to know the factors that influence the demand for its service.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import ttest_ind,shapiro,levene,f_oneway,kruskal
import statsmodels.api as sm
```

read data

```
data = pd.read_csv('yulu.csv')

data.head()
```

```
              datetime  season  holiday  workingday  weather  temp
atemp  \
0  2011-01-01 00:00:00       1        0           0        1  9.84
14.395
1  2011-01-01 01:00:00       1        0           0        1  9.02
13.635
2  2011-01-01 02:00:00       1        0           0        1  9.02
13.635
3  2011-01-01 03:00:00       1        0           0        1  9.84
14.395
4  2011-01-01 04:00:00       1        0           0        1  9.84
14.395

   humidity  windspeed  casual  registered  count
0        81        0.0       3          13     16
1        80        0.0       8          32     40
2        80        0.0       5          27     32
3        75        0.0       3          10     13
4        75        0.0       0           1      1
```

```
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
```

```
data.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
```

No missing value

```
data.describe()
```

```
             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
```

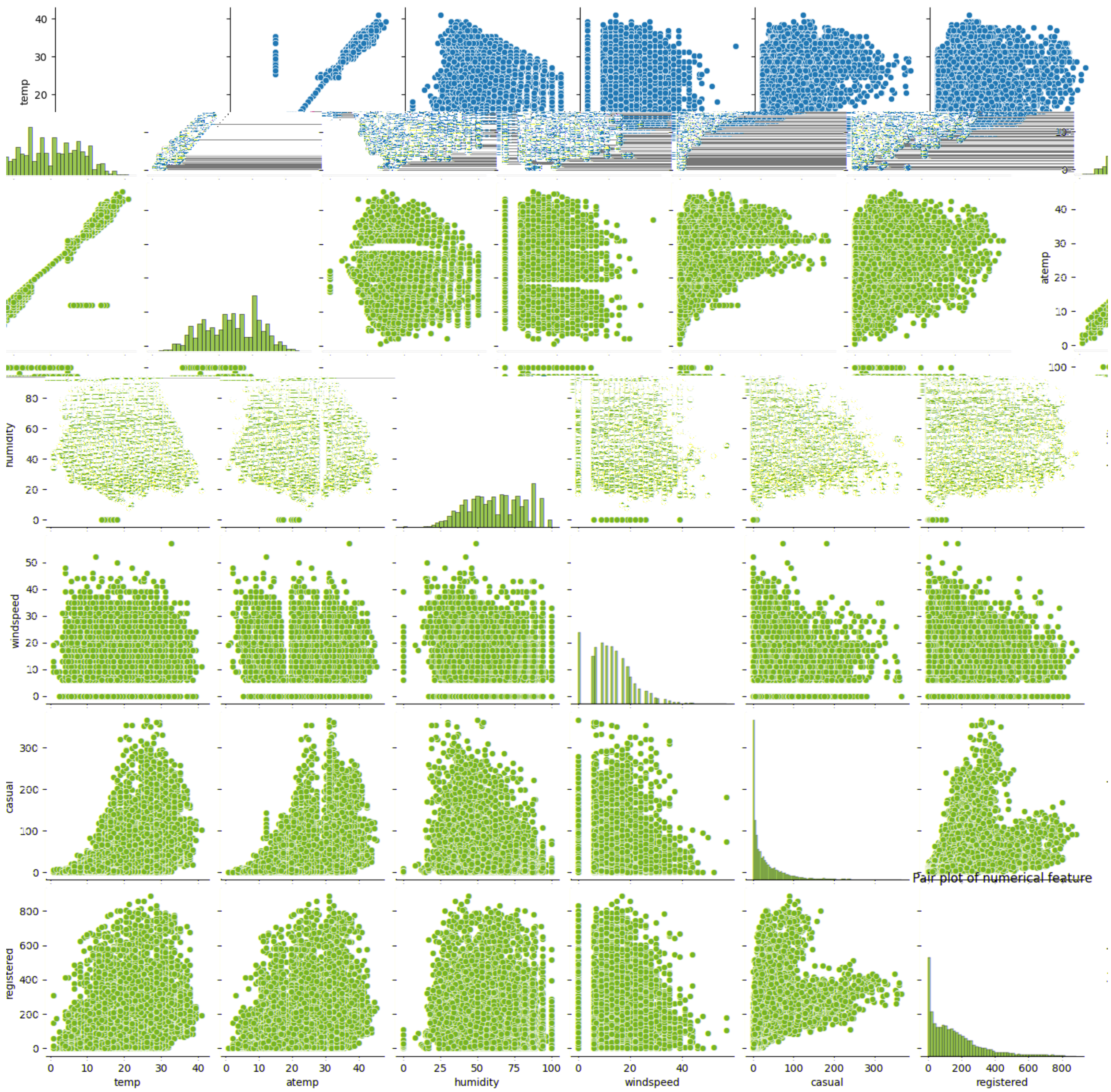
```
                count
count  10886.000000
mean     191.574132
std      181.144454
min        1.000000
25%       42.000000
50%      145.000000
75%      284.000000
max      977.000000
```

> 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: 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 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

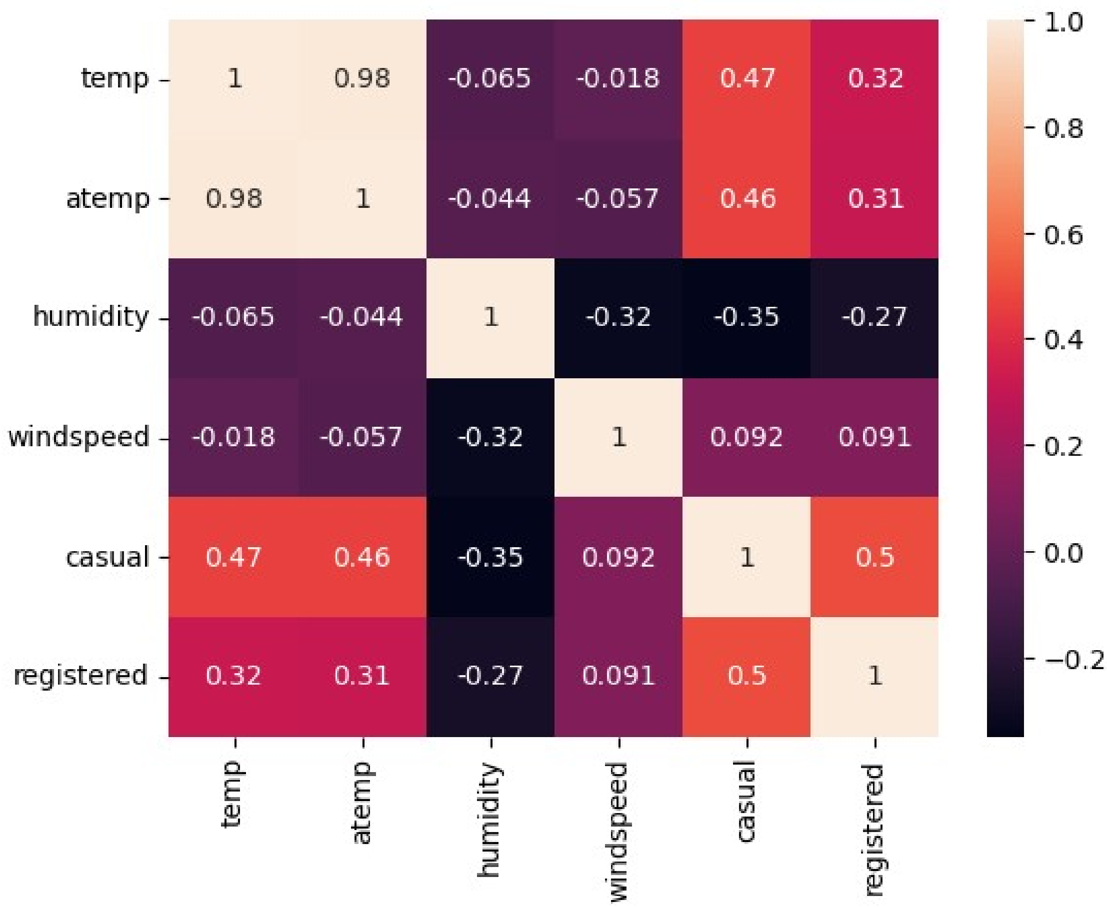
```
# no duplicates available
data.duplicated().sum()

0

categorical_features= ['season'  ,'holiday',      'workingday'
      ,'weather']
numerical_features =['temp',     'atemp',    'humidity' ,'windspeed'
      ,'casual', 'registered']
# transforming to datetime object
data['datetime']=pd.to_datetime(data['datetime'])
```

Eda

```
sns.pairplot(data[numerical_features])
plt.title("Pair plot of numerical feature")

Text(0.5, 1.0, 'Pair plot of numerical feature')
```

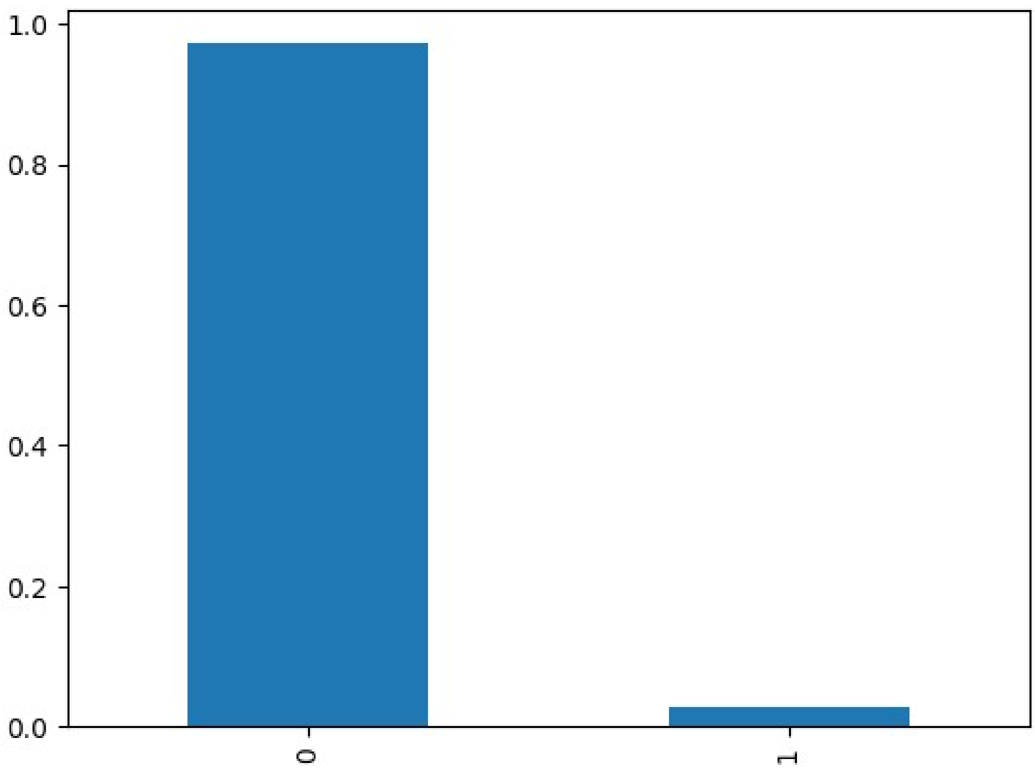



```
cor =data[numerical_features].corr()
sns.heatmap(cor , annot=True)

<Axes: >
```



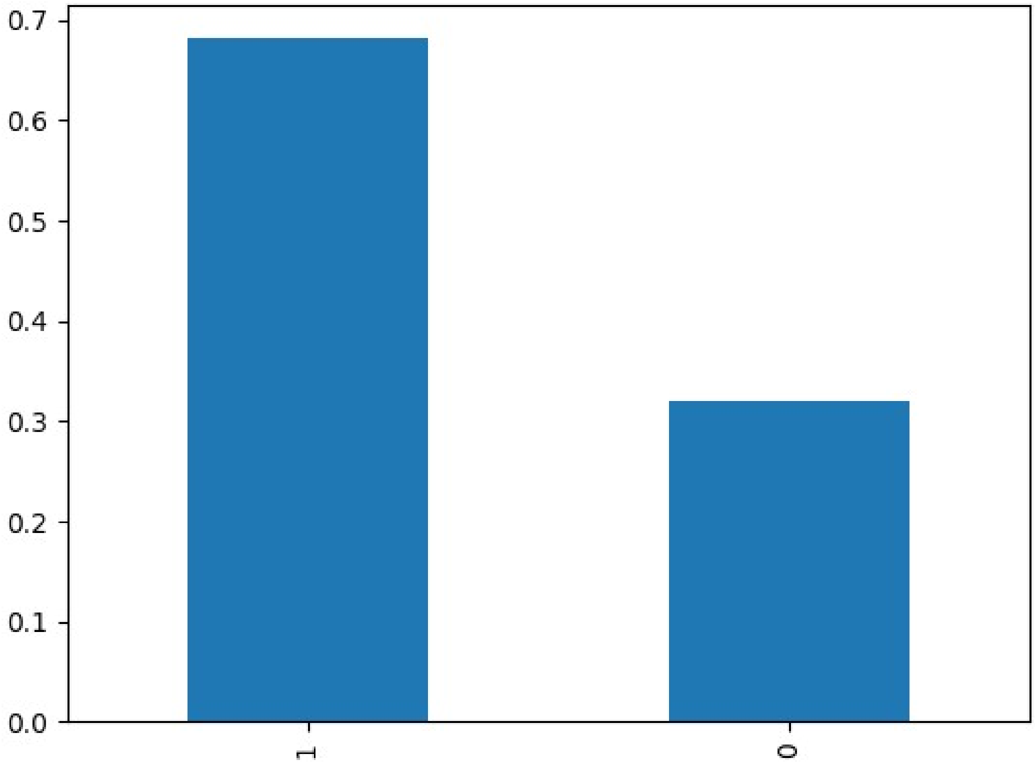
1. windspeed , casual , registred look like log normal dist very right squeed
2. Temp and atemp is higly correlated
3. Temp has some positive corellation with casual and registred
4. humidity is negatively correlated with windspeed , casual , registred

```
print(f"no of unique categorys in holiday:{data['holiday'].unique()}")
data['holiday'].value_counts(normalize =True).plot(kind='bar')
plt.title(print("holiday distribution"))
plt.show()

no of unique categorys in holiday:[0 1]
holiday distribution
```



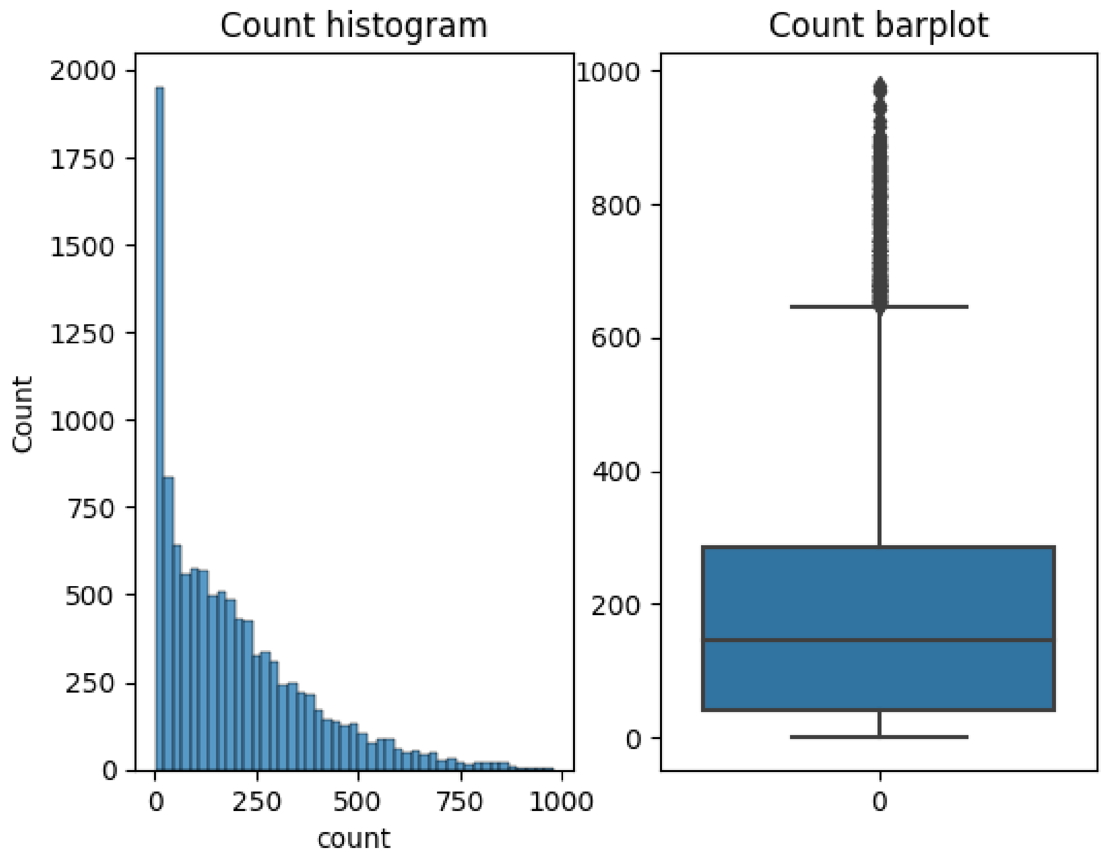
2% of days are holiday and 98% non holiday

```
print(f"no of unique categorys in workingday:
{data['workingday'].unique()}")
data['workingday'].value_counts(normalize =True).plot(kind='bar')
plt.title(print("workingday distribution"))
plt.show()

no of unique categorys in workingday:[0 1]
workingday distribution
```



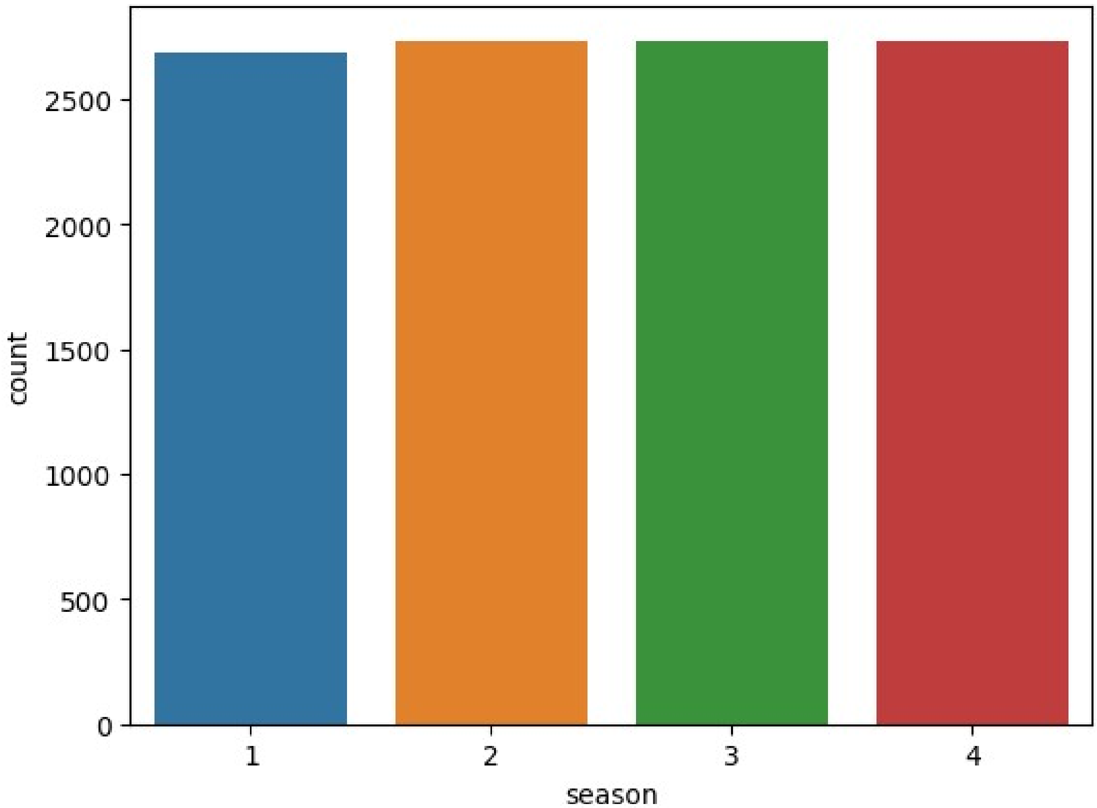
1. There are twocategory in workingday
2. About 68% is workingday and 32% of non working day

```
plt.subplot(121)
sns.histplot(data['count'])
plt.title("Count histogram")

plt.subplot(122)
sns.boxplot(data['count'])
plt.title("Count barplot")

display(data['count'].describe())

count    10886.000000
mean       191.574132
std        181.144454
min          1.000000
25%         42.000000
50%        145.000000
75%        284.000000
max        977.000000
Name: count, dtype: float64
```



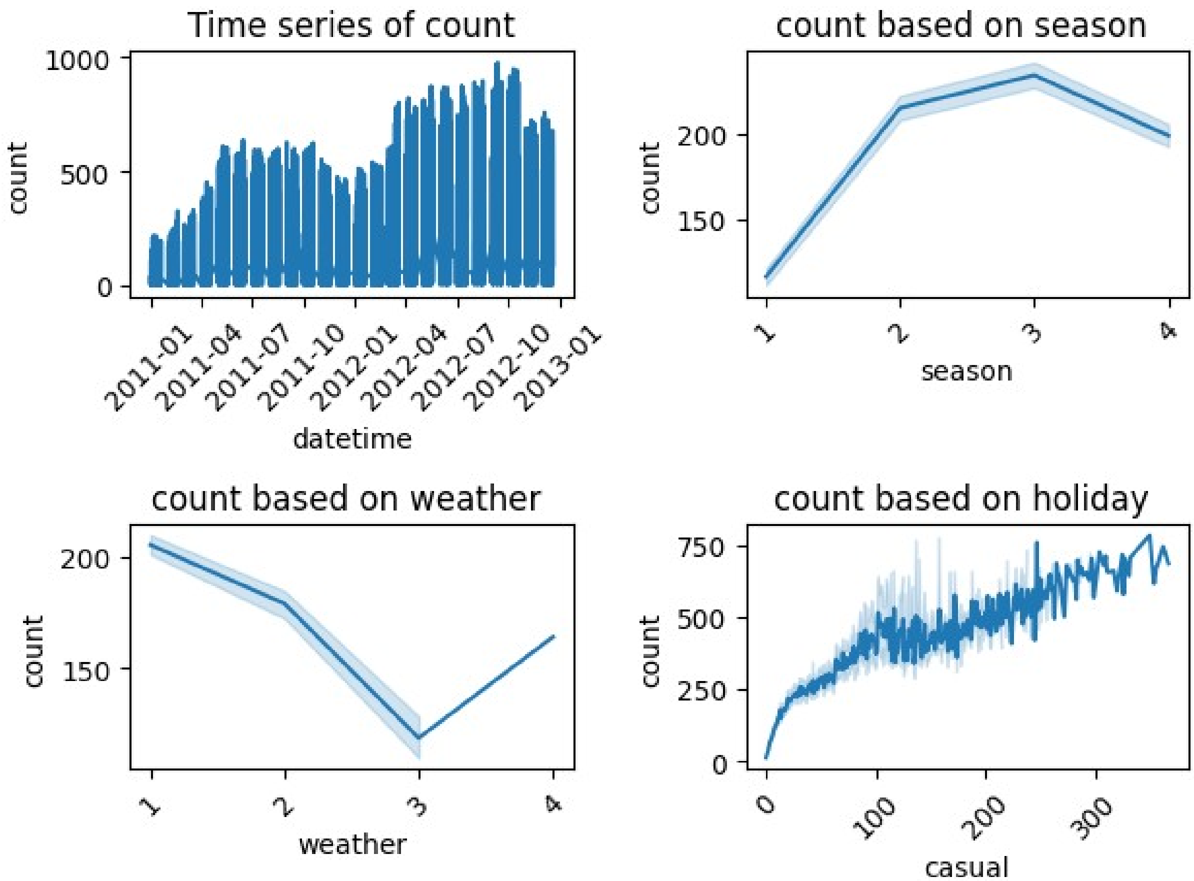
1. Count seem to be righ skewed.
2. Less then 50% of days <=145 rides where rented.
3. Mean of count is 191 and min max is (1,977).
4. There are outlier in count -- count above 650 are treated as outliers.

```
print(f"no of unique categorys in season:{data['season'].unique()}")
sns.countplot(data , x='season')
plt.title(print("season distribution"))
plt.show()

no of unique categorys in season:[1 2 3 4]
season distribution
```



1. All season data looks equally distrubuted

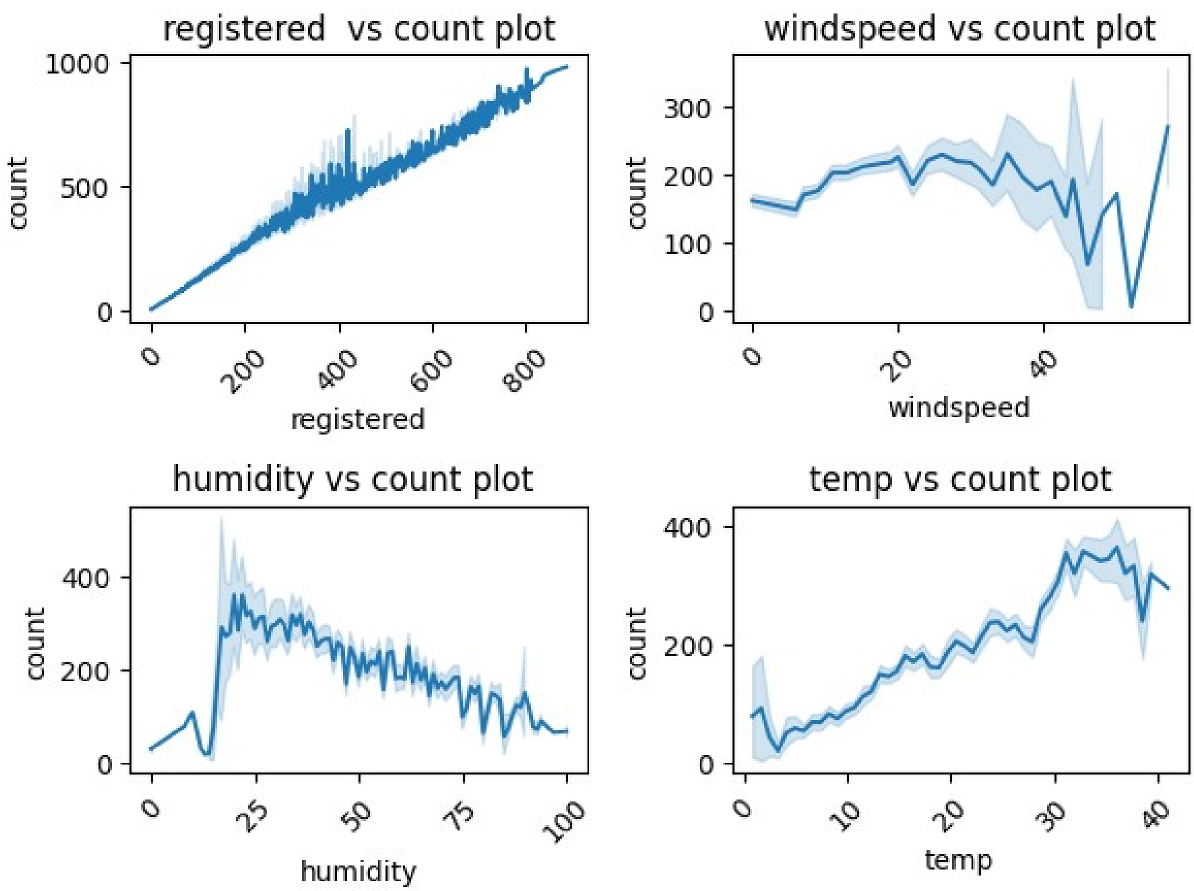
```
plt.subplot(221)
sns.lineplot(data , x= 'datetime'  , y ='count')
plt.xticks(rotation=45)
plt.title("Time series of count")

plt.subplot(222)
sns.lineplot(data , x= 'season'  , y ='count')
plt.xticks(rotation=45)
plt.title("count based on season ")

plt.subplot(223)
sns.lineplot(data , x= 'weather'  , y ='count')
plt.xticks(rotation=45)
plt.title("count based on weather ")

plt.subplot(224)
sns.lineplot(data , x= 'casual'  , y ='count')
plt.xticks(rotation=45)
plt.title("count based on holiday ")
plt.tight_layout()
```



1. Count data has seasonality and there is increase rental yoy but there drop from 2012-10 onward
2. Count increase from season 1-3 and drops at 4
3. Count keep drop from weather 1 to 3
4. As casual increase count increase

```
plt.subplot(221)
sns.lineplot(data , x= 'registered'   , y ='count')
plt.xticks(rotation=45)
plt.title("registered  vs count plot")

plt.subplot(222)
sns.lineplot(data , x= 'windspeed'   , y ='count')
plt.xticks(rotation=45)
plt.title("windspeed vs count plot")

plt.subplot(223)
sns.lineplot(data , x= 'humidity'   , y ='count')
plt.xticks(rotation=45)
plt.title("humidity vs count plot ")

plt.subplot(224)
sns.lineplot(data , x= 'temp'   , y ='count')
```

```
plt.xticks(rotation=45)
plt.title(" temp vs count plot ")
plt.tight_layout()
```



1. A registred increase count increase +ve corr
2. With increase on windspeed count tend to increase slightly , then it drops post 40 and then picks up again
3. With increase in humidity till 25 count seems to increase but then it keeps droping as humidity increases
4. As temp increase count seem to increase

Hypothesis Testing

Working Day has effect on number of electric cycles rented ?

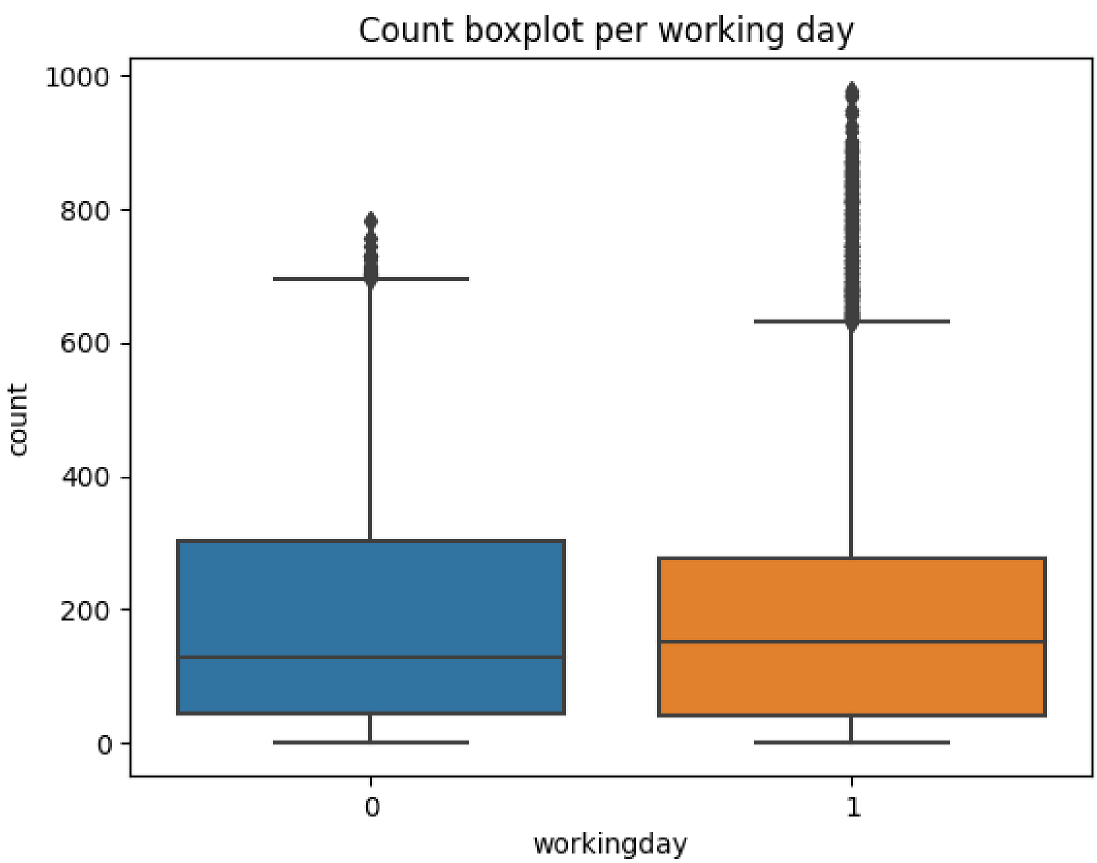
- H0 - There is no effect of working date on count

- Ha - There is effect of working date on count

TTest

```
print("Mean of count at different workingday ")
display(data.groupby('workingday')['count'].mean())
sns.boxplot(data , x='workingday' , y ='count')
plt.title("Count boxplot per working day ")
```

```
Mean of count at different workingday

workingday
0    188.506621
1    193.011873
Name: count, dtype: float64

Text(0.5, 1.0, 'Count boxplot per working day ')
```

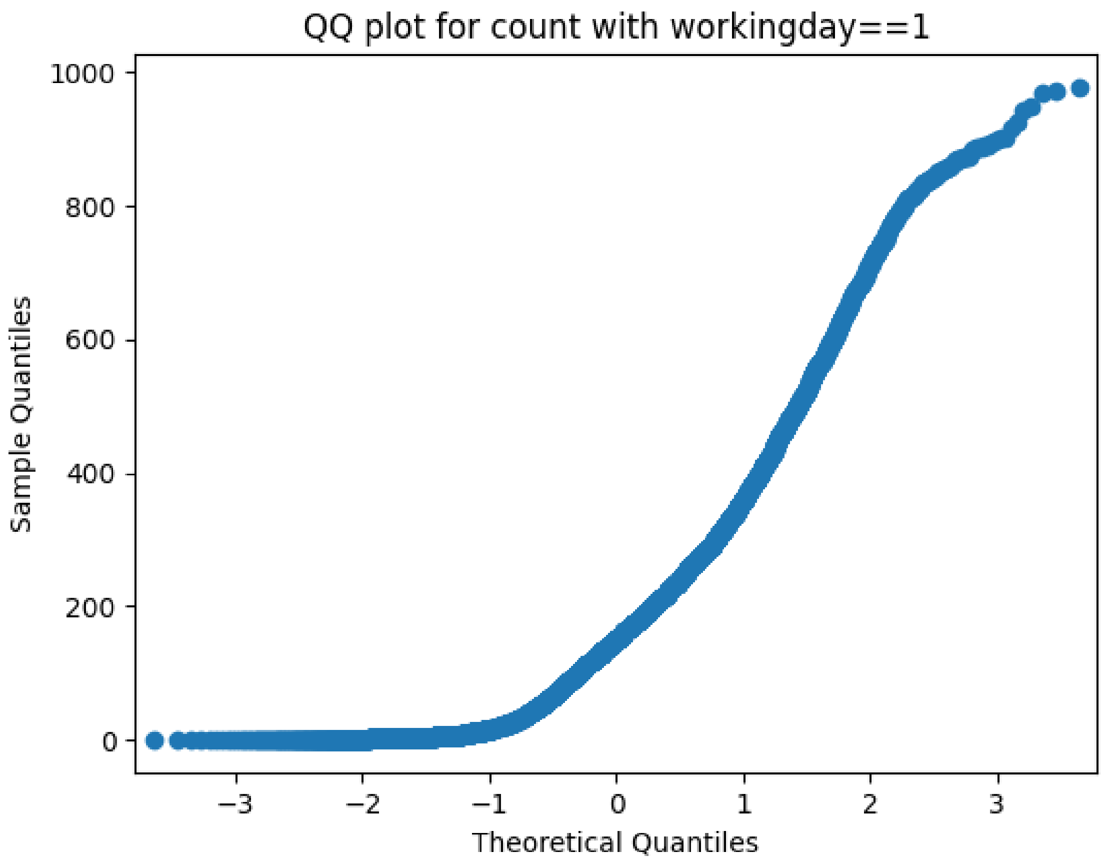


- There is not much difference between workingday 1 and 0

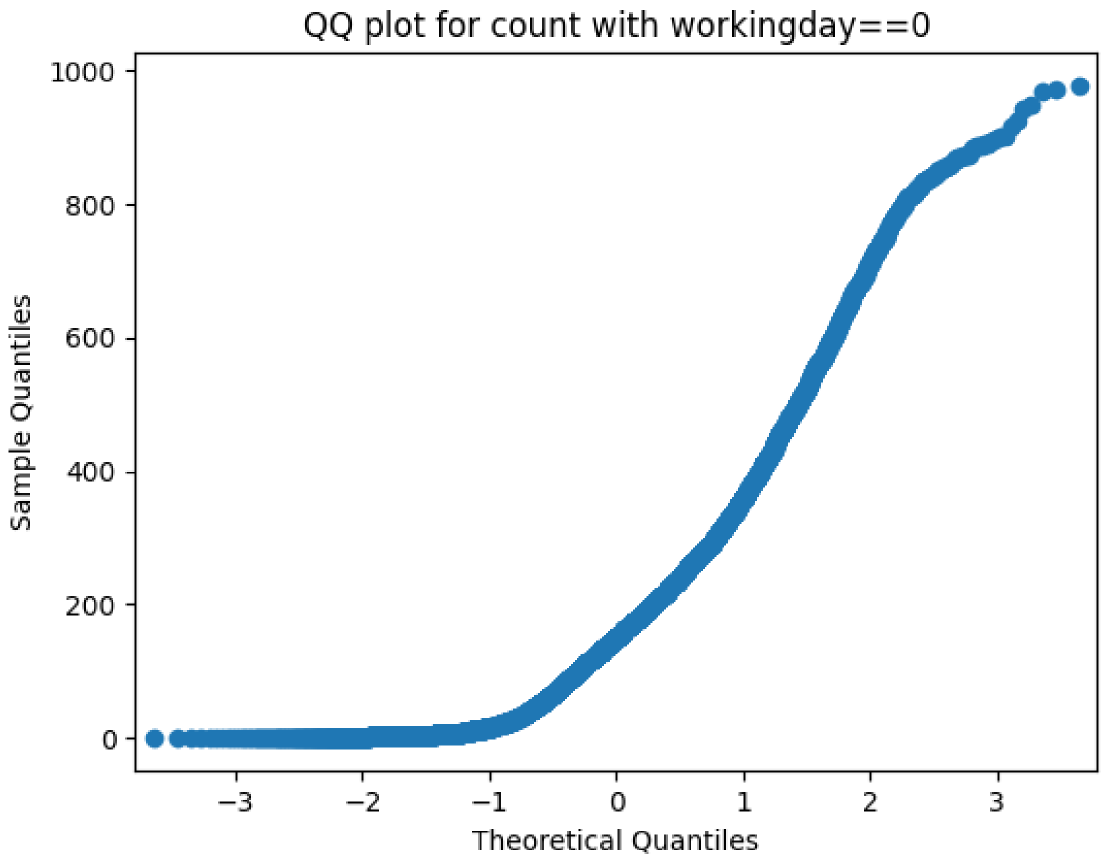
```
def check_normality(data , text ,alpha =0.05):

  sm.qqplot(data)
  plt.title(text)
```

```
  plt.show()
  n =100
  if  data.shape[0]<=n:
    n=data.shape[0]
  print("shapiro test")
  try:
    test =shapiro(data.sample(n))
    print(test)
    if test[1] <alpha:
      print("Reject Ho and  data is not normal")
    else:
      print("Fail Reject Ho and  data is normal")
  except Exception as e:
    print(e)


def check_equal_variability(data  ,alpha =0.05):


  print("levene test")
  try :
    test =levene()
    if test[1] <alpha:
      print("Reject Ho and  data is not normal")
    else:
      print("Fail Reject Ho and  data is normal")
  except Exception as e:
    print(e)



wd=data[data['workingday']==1]['count']
nwd=data[data['workingday']==1]['count']
print("#"* 50)
alpha = 0.05
print(f"Set a significance level (alpha) {alpha}")
print("#"* 50)
print("Test for Assumptions")
text ="QQ plot for count with workingday==1"
check_normality(wd ,text)
text ="QQ plot for count with workingday==0"
check_normality(wd ,text)
print("#"* 50)
print("Calculate test Statistics ttest")
res =ttest_ind(wd,nwd)
print(res)
print("Decision to accept or reject null hypothesis")
print("#"* 50)
if res[1] <alpha:
  print("Reject Ho :There is significant difference between working
```

```
day and non working day")
else:
  print("Fail Reject Ho: There is no significant difference between
working day and non working day")

##################################################
Set a significance level (alpha) 0.05
##################################################
Test for Assumptions
```




```
shapiro test
ShapiroResult(statistic=0.8560823202133179,
pvalue=2.0027924207965953e-08)
Reject Ho and  data is not normal
```




```
shapiro test
ShapiroResult(statistic=0.8781797885894775,
pvalue=1.5210953563382645e-07)
Reject Ho and  data is not normal
##################################################
Calculate test Statistics ttest
TtestResult(statistic=0.0, pvalue=1.0, df=14822.0)
Decision to accept or reject null hypothesis
##################################################
Fail Reject Ho: There is no significant difference between working day
and non working day
```

Summary

- There is no significant difference between working day and non working day

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

- H0 - There is no effect of season on count
- Ha - Season effects count

Anova

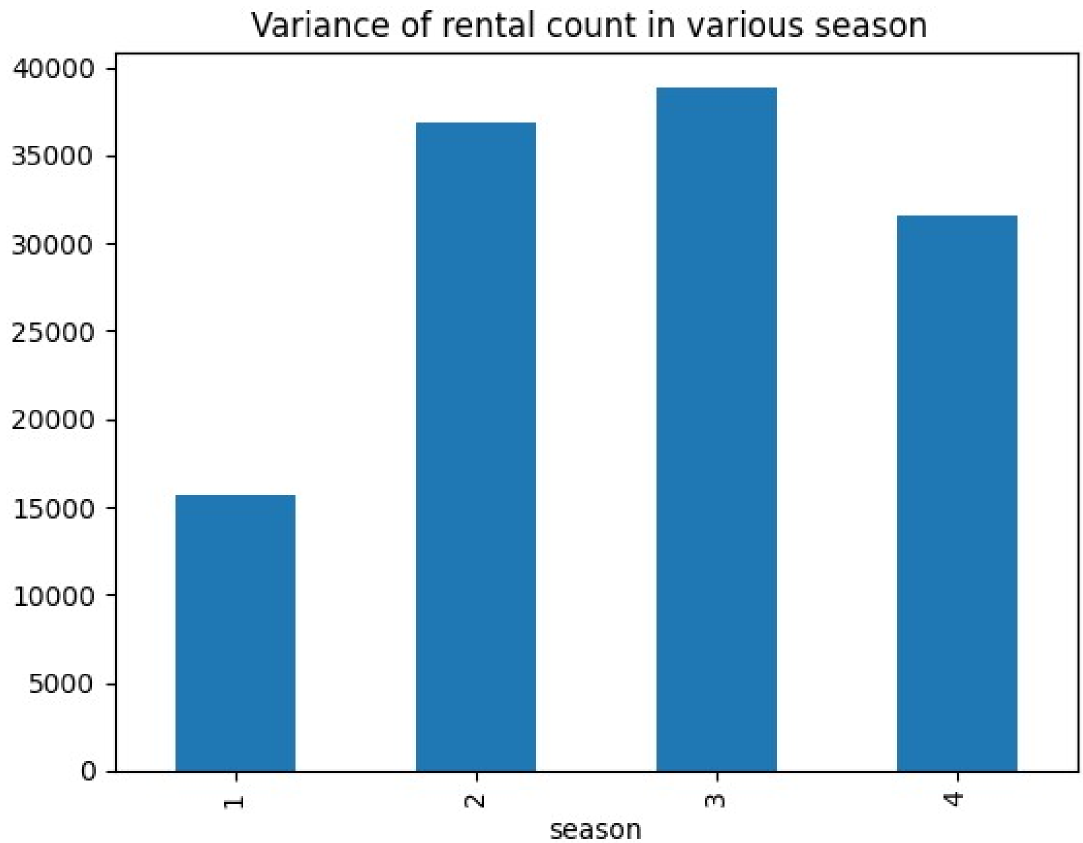
```
print("Mean of count at different season ")
display(data.groupby('season')['count'].mean())
print("varaince of count at different season ")
print(data.groupby('season')['count'].var())
data.groupby('season')['count'].var().plot(kind='bar')
plt.title("Variance of rental count in various season ")
```

```
Mean of count at different season
```

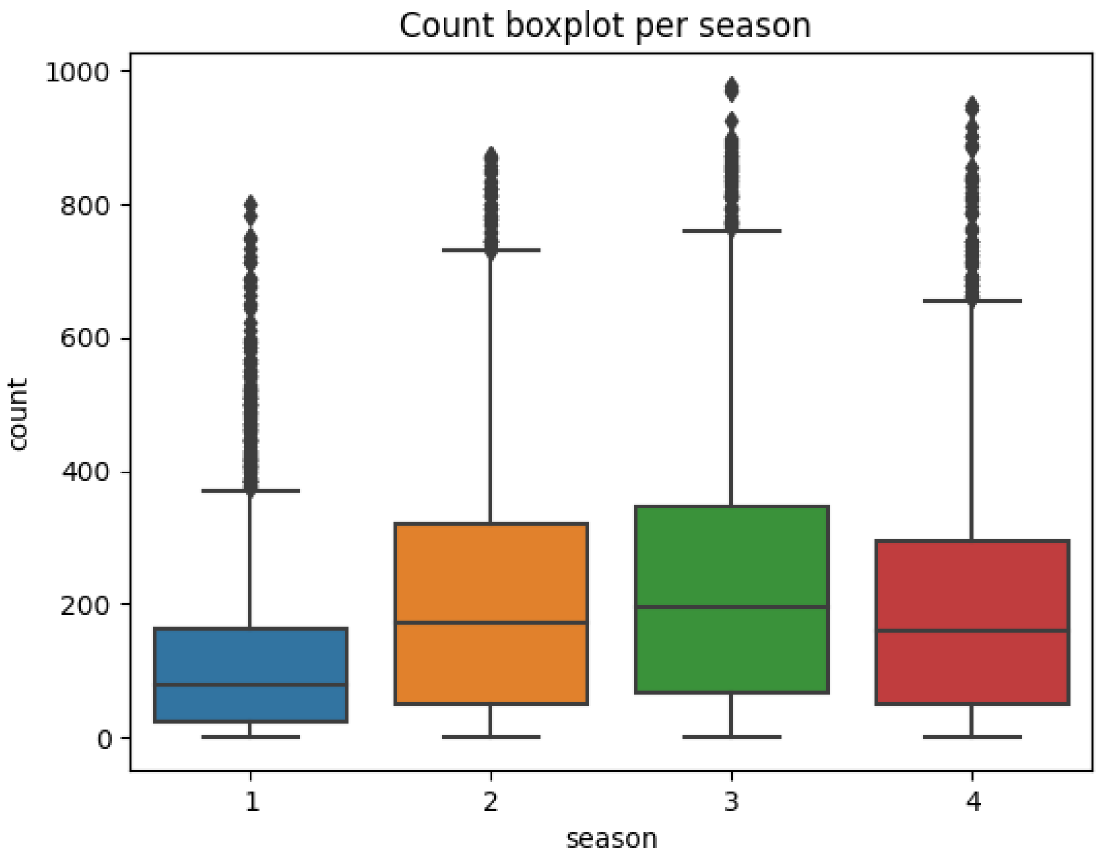
```
season
1    116.343261
2    215.251372
3    234.417124
4    198.988296
Name: count, dtype: float64
```

```
varaince of count at different season
season
1    15693.568534
2    36867.011826
3    38868.517013
4    31549.720317
Name: count, dtype: float64
```

```
Text(0.5, 1.0, 'Variance of rental count in various season ')
```



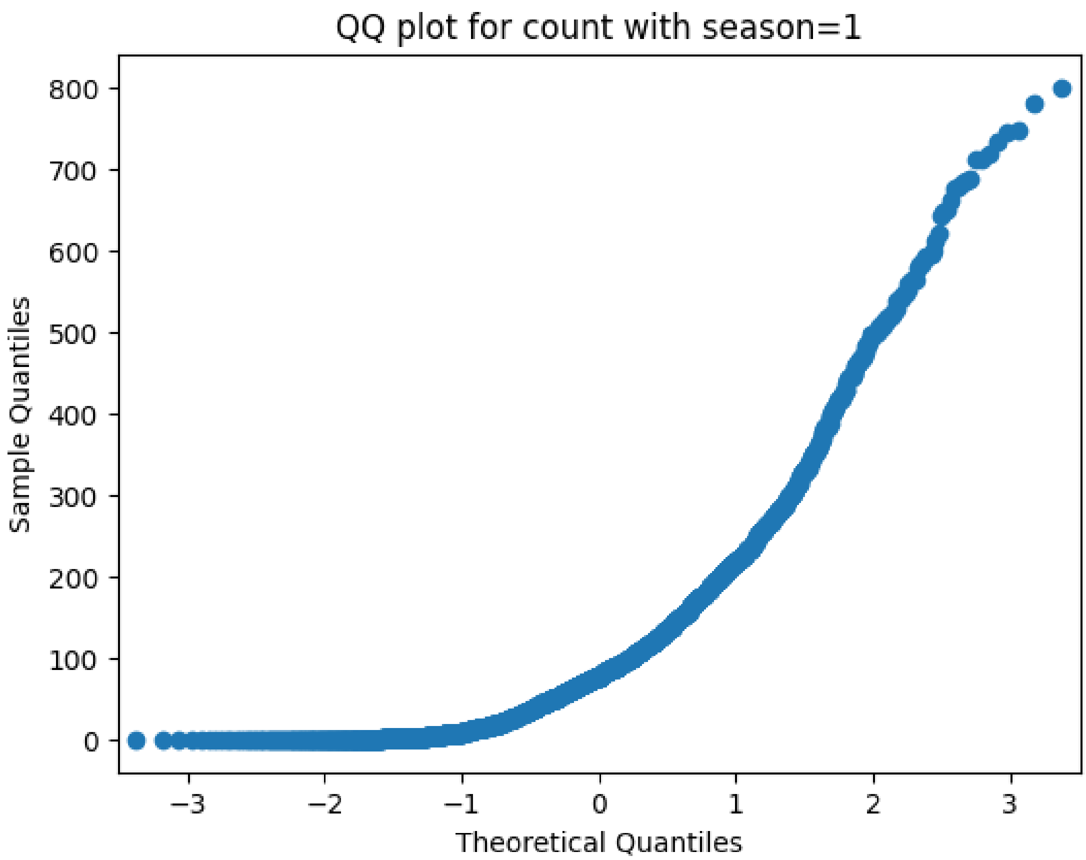

```
sns.boxplot(data , x='season' , y ='count')
plt.title("Count boxplot per season ")

Text(0.5, 1.0, 'Count boxplot per season ')
```



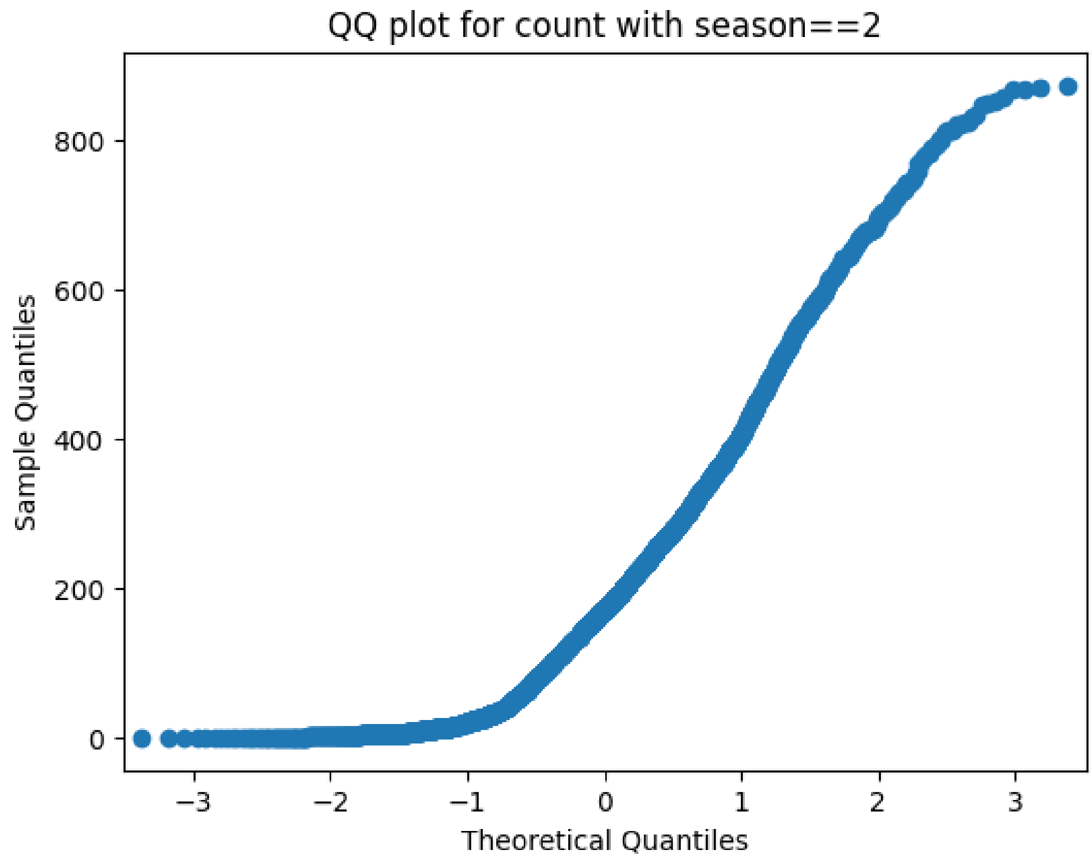
- Season 3 has highest mean count [fall]
- season 1 has lowest mean count [spring]
- looks like of variance groups are different

```
seasondata={  s : data[data['season']==s]['count'] for s in
data['season'].unique()}

print("#"* 50)
alpha = 0.05
print(f"Set a significance level (alpha) {alpha}")
print("#"* 50)
print("Test for Assumptions")
print('Normality')
text ="QQ plot for count with season=1"
check_normality( seasondata[1],text)
text ="QQ plot for count with season==2"
check_normality(seasondata[2] ,text)
text ="QQ plot for count with season==3"
check_normality(seasondata[3] ,text)
text ="QQ plot for count with season==4"
check_normality(seasondata[4] ,text)
print("#"* 50)
print("Checking for equal variance using levene")
```

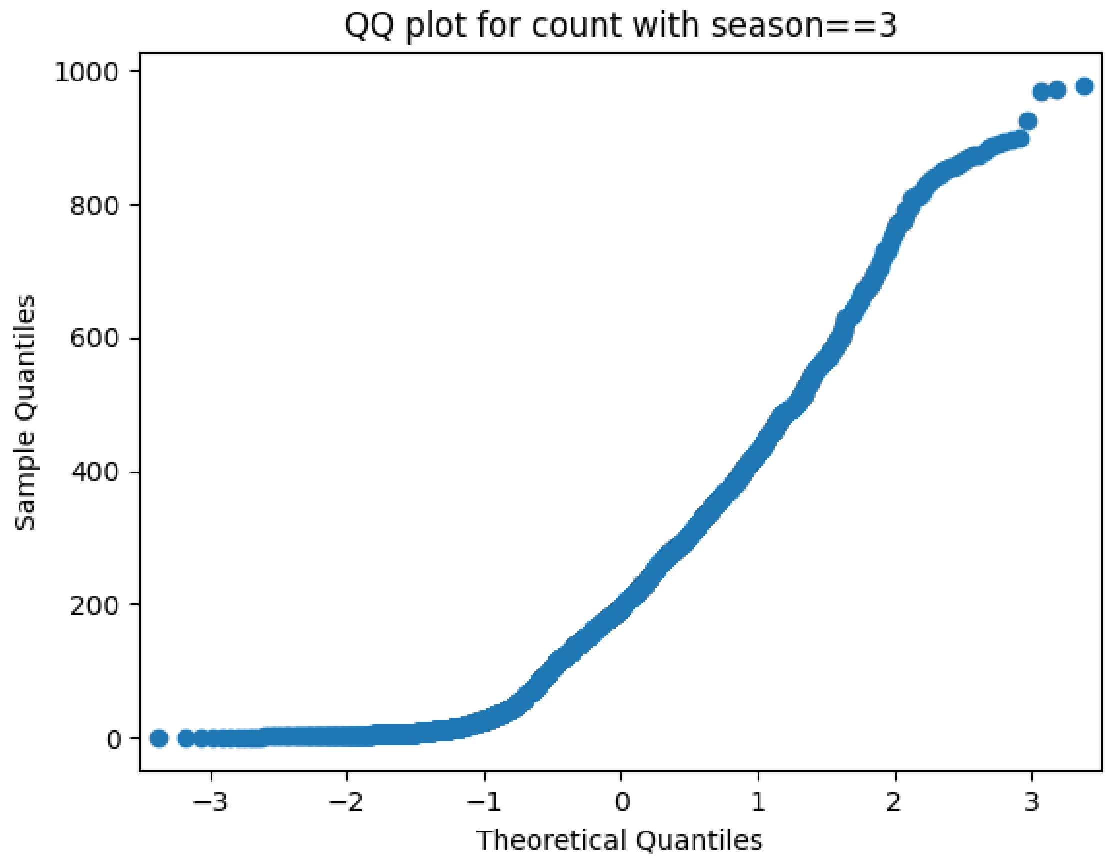
```
test =levene(seasondata[1] ,seasondata[2] , seasondata[3] ,
seasondata[4])
print(test)
if test[1] <alpha:
  print("Reject Ho and  data doesnt have equal variance")
else:
  print("Fail Reject Ho and  data  has equal variance")

##################################################
Set a significance level (alpha) 0.05
##################################################
Test for Assumptions
Normality
```



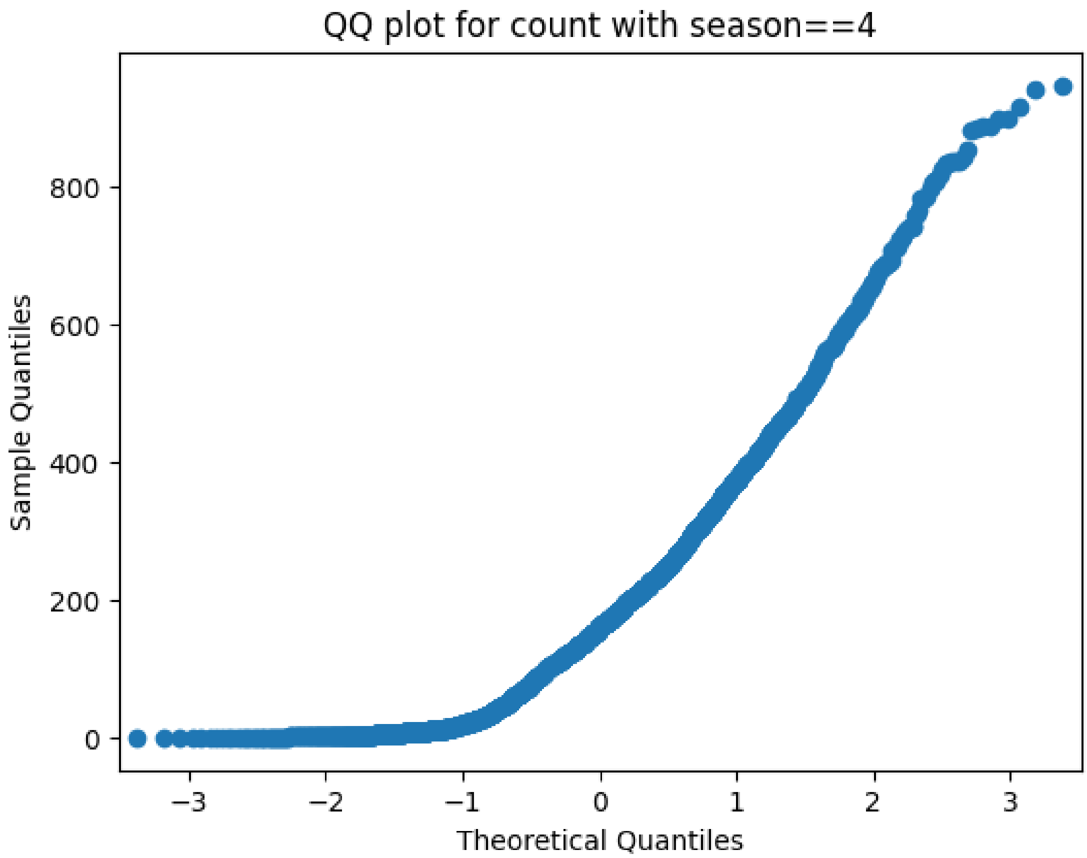

```
shapiro test
ShapiroResult(statistic=0.7658171653747559,
pvalue=2.5049070456750755e-11)
Reject Ho and  data is not normal
```




```
shapiro test
ShapiroResult(statistic=0.9162874817848206, pvalue=8.991204595076852e-06)
Reject Ho and  data is not normal
```




```
shapiro test
ShapiroResult(statistic=0.9164454936981201, pvalue=9.162337846646551e-06)
Reject Ho and  data is not normal
```

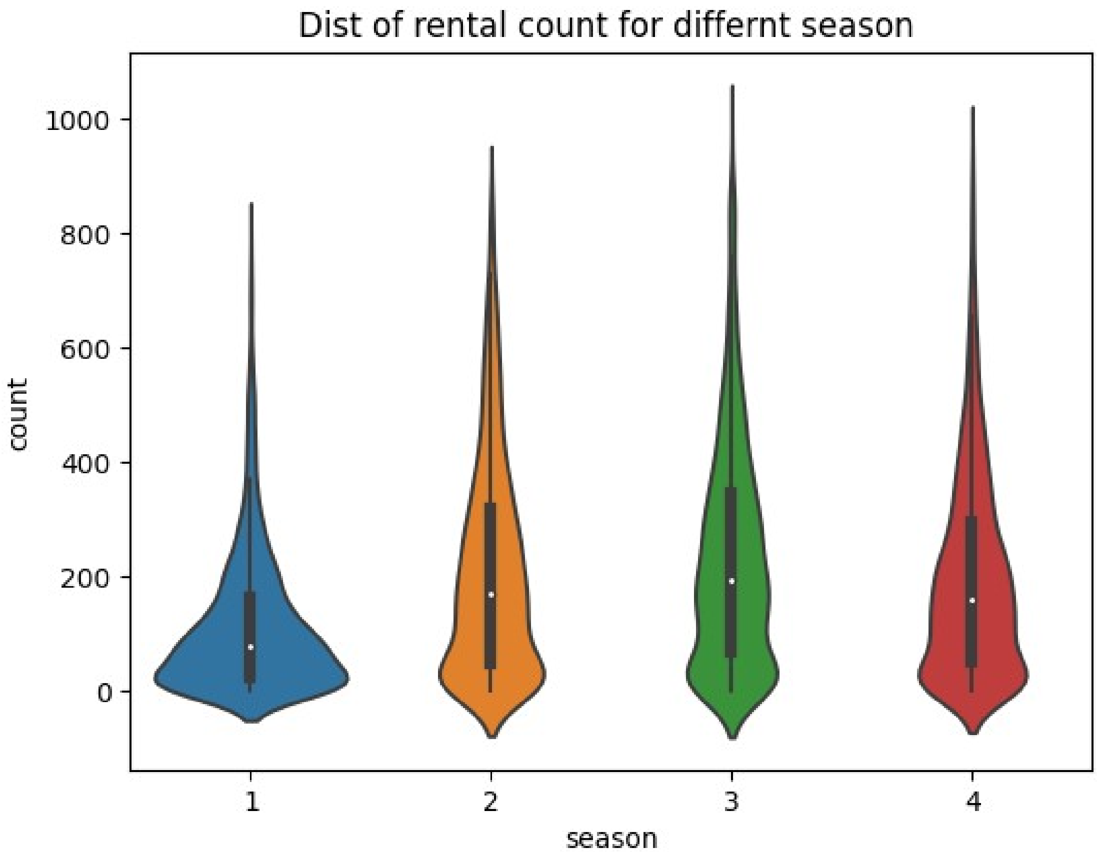



```
shapiro test
ShapiroResult(statistic=0.8377450108528137, pvalue=4.289978150495699e-09)
Reject Ho and  data is not normal
##################################################
Checking for equal variance using levene
LeveneResult(statistic=187.7706624026276, pvalue=1.0147116860043298e-118)
Reject Ho and  data doesnt have equal variance
```

- We failed assumption of Anova -- we will ks test if group are similar
- I will also give Anova a try -- in real world meeting all the assumption would be hard

```
print("#"* 50)
print("Anova test:")
res =f_oneway(seasondata[1] ,seasondata[2] , seasondata[3] ,
seasondata[4])
print(res)
print("Decision to accept or reject null hypothesis")

if res[1] <alpha:
  print("Reject Ho :There is significant difference between rental
count across season")
```

```
else:
  print("Fail Reject Ho: There is no significant difference between 
rental count across season")

print("#"* 50)
print("kruskal test:")
res =kruskal(seasondata[1] ,seasondata[2] , seasondata[3] , 
seasondata[4])
print(res)
print("Decision to accept or reject null hypothesis")

if res[1] <alpha:
  print("Reject Ho :There is significant difference between rental 
count across season")
else:
  print("Fail Reject Ho: There is no significant difference between 
rental count across season")
print("#"* 50)

##################################################
Anova test:
F_onewayResult(statistic=236.94671081032106, 
pvalue=6.164843386499654e-149)
Decision to accept or reject null hypothesis
Reject Ho :There is significant difference between rental count across 
season
##################################################
kruskal test:
KruskalResult(statistic=699.6668548181988, pvalue=2.479008372608633e-
151)
Decision to accept or reject null hypothesis
Reject Ho :There is significant difference between rental count across 
season
##################################################

sns.violinplot(data , x ='season' , y='count')
plt.title("Dist of rental count for differnt season ")

Text(0.5, 1.0, 'Dist of rental count for differnt season ')
```



Summary

- we can conclude that Season does effect bike rental based on test
- From dist we can see that median of group are different season to season
- season 1 is data is not very spread compared to 2,3,4 , people usually take bike for short ride may they prefer to walk when weather is good

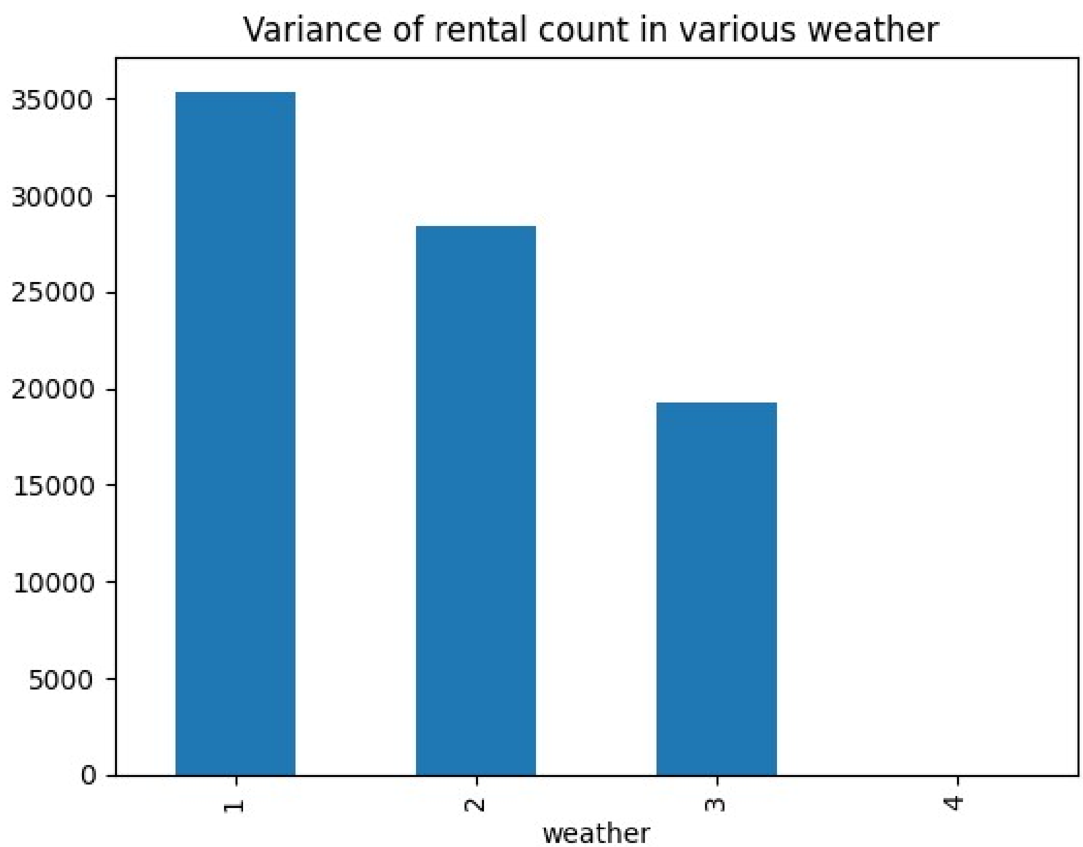
No. of cycles rented similar or different in different weather ?

- H0 - There is no effect of weather on count
- Ha - weather effects count

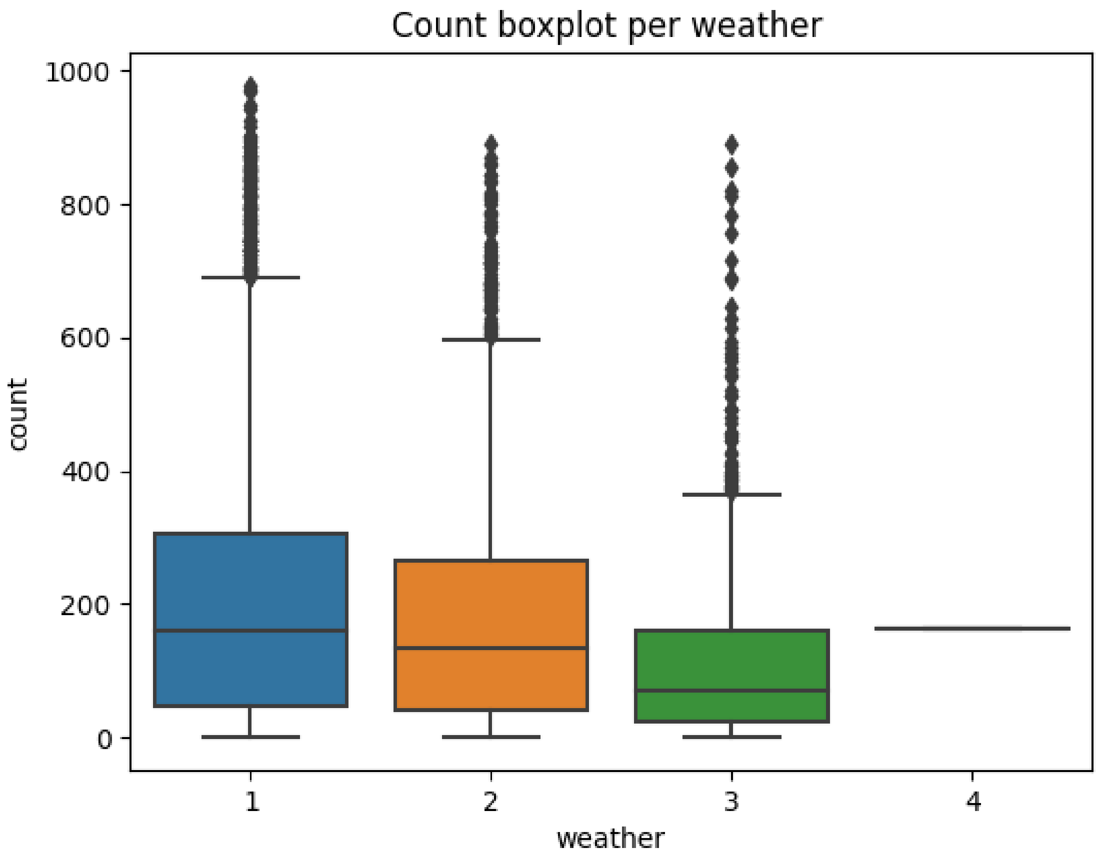
Test =Anova

```
print("Mean of count at different weather ")
display(data.groupby('weather')['count'].mean())
print("varaince of count at different weather ")
print(data.groupby('weather')['count'].var())
data.groupby('weather')['count'].var().plot(kind='bar')
plt.title("Variance of rental count in various weather ")
```

```
Mean of count at different weather

weather
1    205.236791
2    178.955540
3    118.846333
4    164.000000
Name: count, dtype: float64

varaince of count at different weather
weather
1    35328.798463
2    28347.248993
3    19204.775893
4             NaN
Name: count, dtype: float64

Text(0.5, 1.0, 'Variance of rental count in various weather ')
```




```
sns.boxplot(data , x='weather' , y ='count')
plt.title("Count boxplot per weather ")

Text(0.5, 1.0, 'Count boxplot per weather ')
```



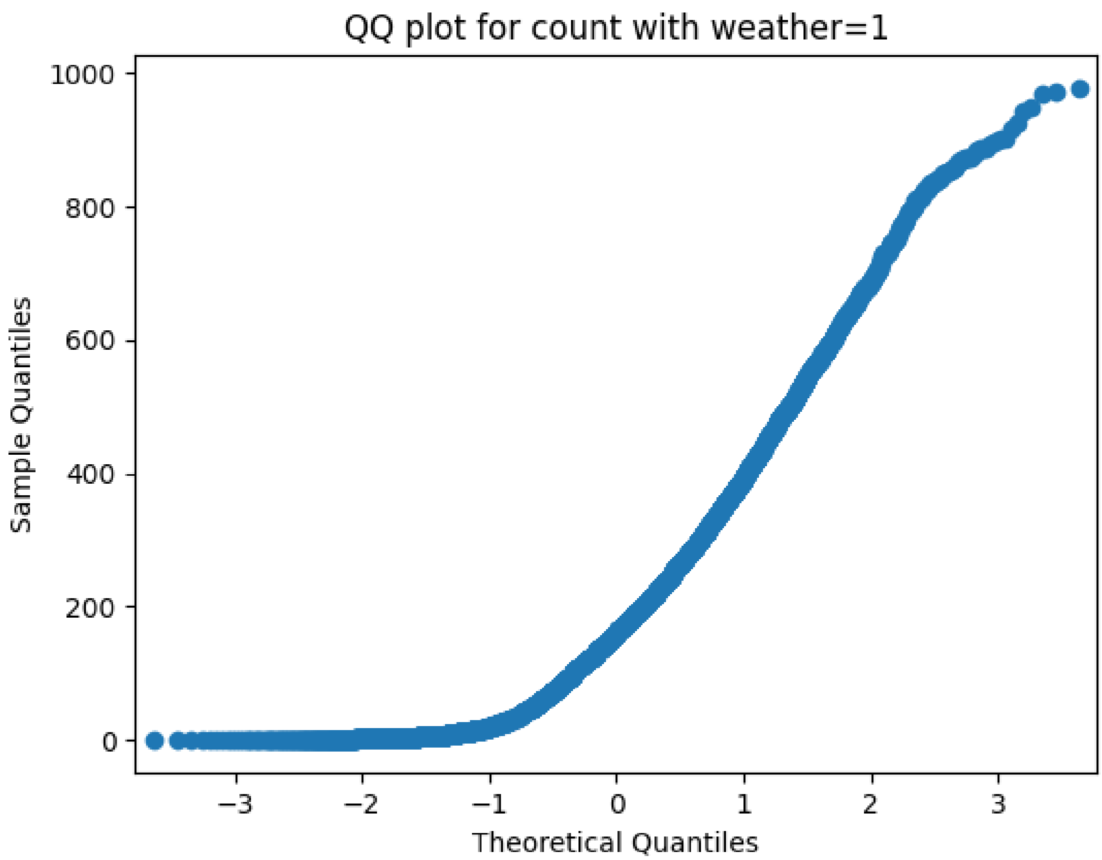
- Weather 1 has highest variance followed by 2 and 3
- mean of weather 1 is highest followed by 2 ,3

Testing for Assumption of Anova

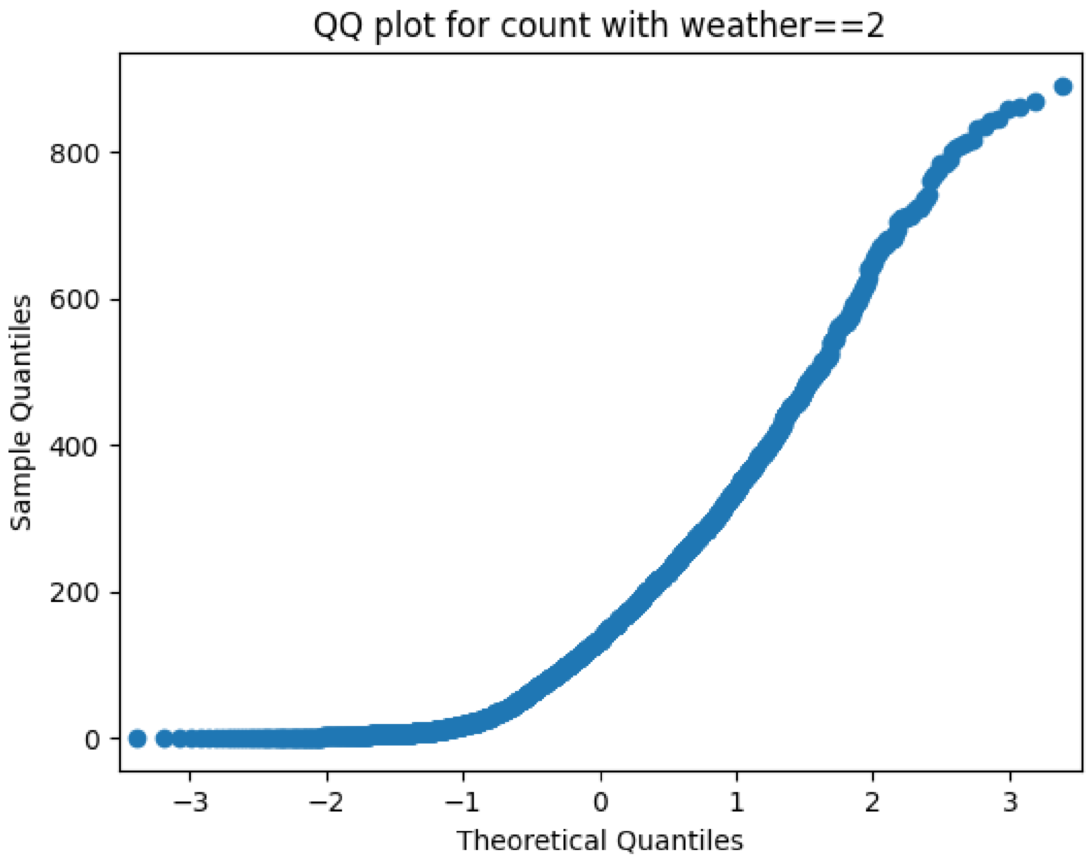
```
weatherdata={  s : data[data['weather']==s]['count'] for s in
data['weather'].unique()}

print("#"* 50)
alpha = 0.05
print(f"Set a significance level (alpha) {alpha}")
print("#"* 50)
print("Test for Assumptions")
print('Normality')
text ="QQ plot for count with weather=1"
check_normality( weatherdata[1],text)
text ="QQ plot for count with weather==2"
check_normality(weatherdata[2] ,text)
text ="QQ plot for count with weather==3"
check_normality(weatherdata[3] ,text)
text ="QQ plot for count with weather==4"
check_normality(weatherdata[4] ,text)
print("#"* 50)
```

```
print("Checking for equal variance using levene")
test =levene(weatherdata[1] ,weatherdata[2] , weatherdata[3] ,
weatherdata[4])
print(test)
if test[1] <alpha:
  print("Reject Ho and  data doesnt have equal variance")
else:
  print("Fail Reject Ho and  data  has equal variance")
```

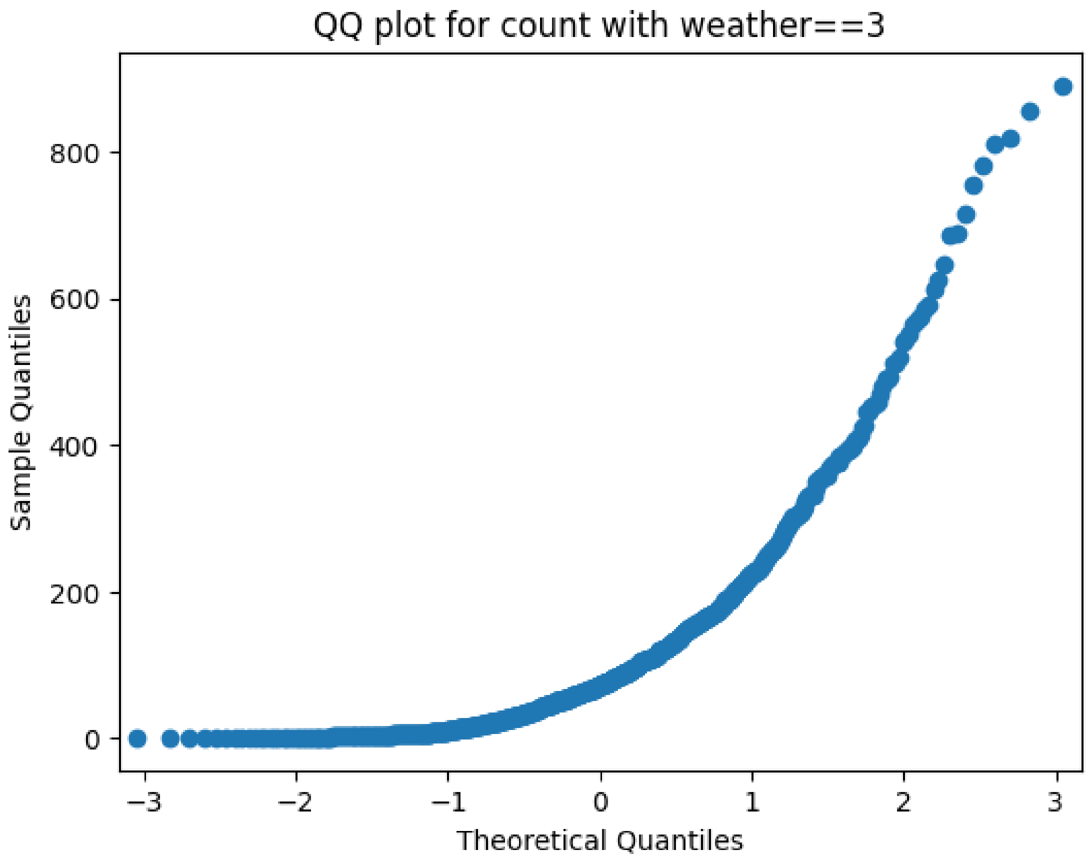
```
##################################################
Set a significance level (alpha) 0.05
##################################################
Test for Assumptions
Normality
```



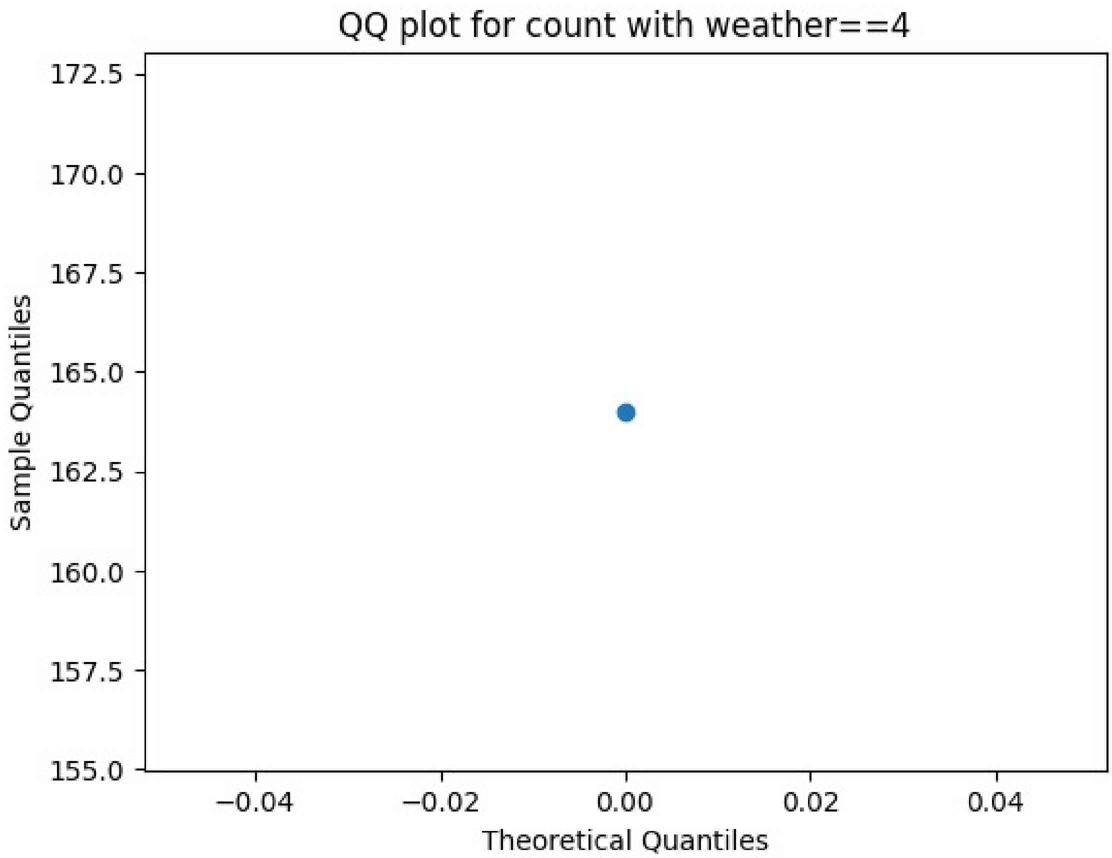

```
shapiro test
ShapiroResult(statistic=0.8585354089736938,
pvalue=2.4837834899926747e-08)
Reject Ho and  data is not normal
```




```
shapiro test
ShapiroResult(statistic=0.8388574719429016, pvalue=4.695187350023389e-09)
Reject Ho and  data is not normal
```




```
shapiro test
ShapiroResult(statistic=0.6883695721626282,
pvalue=2.9135399062969747e-13)
Reject Ho and  data is not normal
```




```
shapiro test
Data must be at least length 3.
##################################################
Checking for equal variance using levene
LeveneResult(statistic=54.85106195954556, pvalue=3.504937946833238e-
35)
Reject Ho and  data doesnt have equal variance
```

- We failed assumption of Anova -- we will ks test if group are similar
- I will also give Anova a try -- in real world meeting all the assumption would be hard
- Removing weather ==4 as it has only one datapoint while conducting test

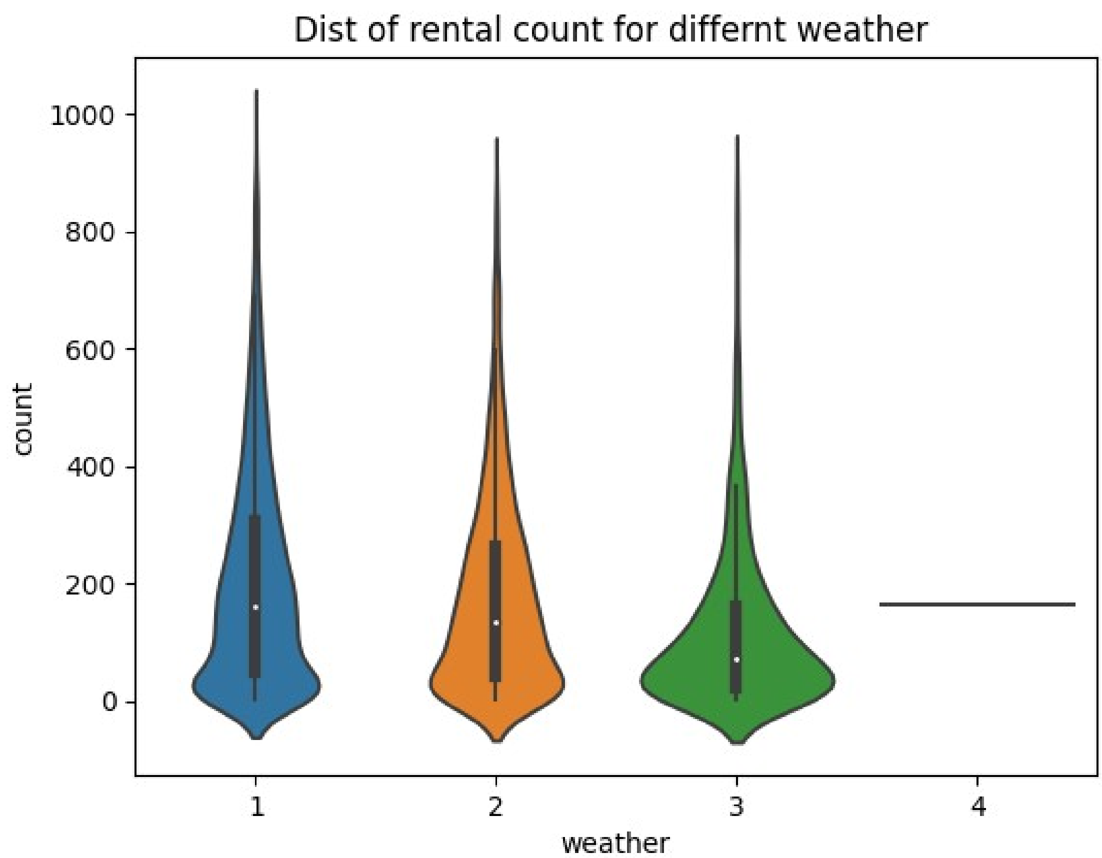
```
print("#"* 50)
print("Anova test:")
res =f_oneway(weatherdata[1] ,weatherdata[2] , weatherdata[3] )
print(res)
print("Decision to accept or reject null hypothesis")

if res[1] <alpha:
  print("Reject Ho :There is significant difference between rental
count across weather")
else:
  print("Fail Reject Ho: There is no significant difference between
```

```
rental count across weather")

print("#"* 50)
print("kruskal test:")
res =kruskal(weatherdata[1] ,weatherdata[2] , weatherdata[3] )
print(res)
print("Decision to accept or reject null hypothesis")

if res[1] <alpha:
  print("Reject Ho :There is significant difference between rental
count across weather")
else:
  print("Fail Reject Ho: There is no significant difference between
rental count across weather")
print("#"* 50)
```

```
##################################################
Anova test:
F_onewayResult(statistic=98.28356881946706, pvalue=4.976448509904196e-
43)
Decision to accept or reject null hypothesis
Reject Ho :There is significant difference between rental count across
weather
##################################################
kruskal test:
KruskalResult(statistic=204.95566833068537, pvalue=3.122066178659941e-
45)
Decision to accept or reject null hypothesis
Reject Ho :There is significant difference between rental count across
weather
##################################################
```

```
sns.violinplot(data , x ='weather' , y='count')
plt.title("Dist of rental count for differnt weather ")
```

```
Text(0.5, 1.0, 'Dist of rental count for differnt weather ')
```



summary

- we can conclude that weather does effect bike rental based on test
- From dist we can see that median of group are different weather to weather
- weather 1 is data is very spread compared to 2,3,4 , people usually take bike for short ride may they prefer to walk when weather is good

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

- H0 : Weather is independent of season
- HA : Weather is dependent of season
- Test Chisquare

```
from scipy.stats import chi2_contingency

print("Preparing data for chisquare test: ")
contegency =pd.crosstab(data['weather'],data['season'])
```

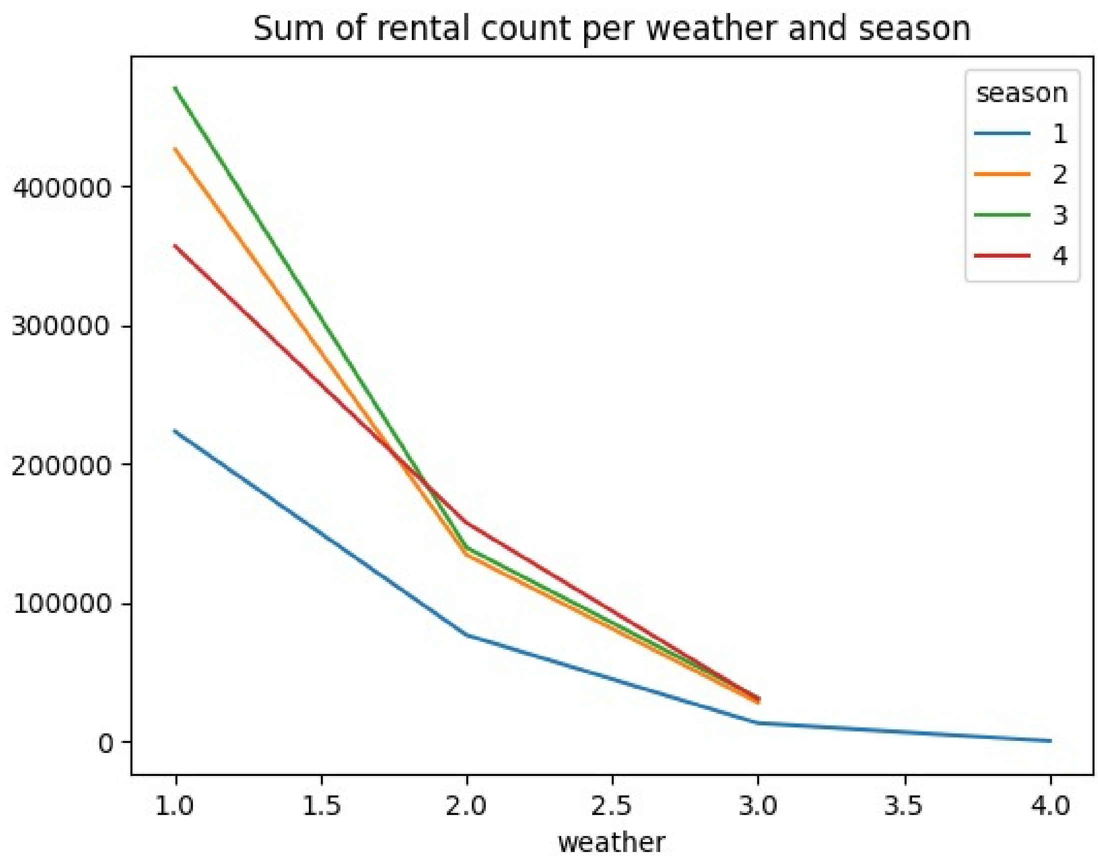
```
display(contegency)
alpha =0.05
print(f" defining alpha as {alpha} :")
print("#"* 50)
res =chi2_contingency(contegency)
print("Test stats: ")
display(res)
if res[1] <alpha:
  print("Reject Ho : Weather and season are dependent")
else:
  print("Fail Reject Ho: Weather and season are independent no effect
of weather on season ")
print("#"* 50)
```

```
Preparing data for chisquare test:

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

 defining alpha as 0.05 :
##################################################
Test stats:

Chi2ContingencyResult(statistic=49.15865559689363,
pvalue=1.5499250736864862e-07, dof=9,
expected_freq=array([[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]]))

Reject Ho : Weather and season are dependent
##################################################
```

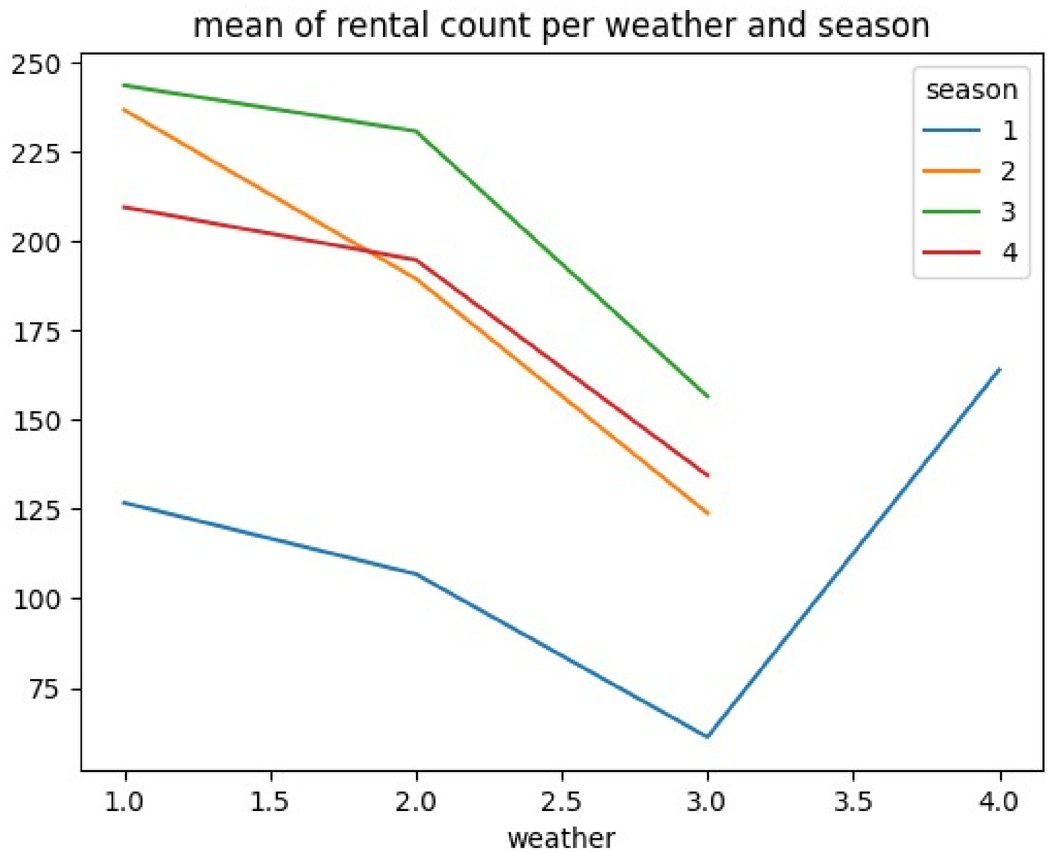
```
pivot =pd.pivot_table(data = data , values ='count', index
='weather' , columns='season' , aggfunc=np.sum)
pivot.plot()
plt.title("Sum of rental count per weather and season")
pivot
```

```
season          1         2         3         4
weather
1        223009.0  426350.0  470116.0  356588.0
2         76406.0  134177.0  139386.0  157191.0
```

```
3          12919.0   27755.0   31160.0   30255.0
4            164.0       NaN       NaN       NaN
```




```
pivot =pd.pivot_table(data = data , values ='count', index
='weather' , columns='season' , aggfunc=np.mean)
pivot.plot()
plt.title("mean of rental count per weather and season")
pivot

season            1           2           3           4
weather
1        126.781694  236.729595  243.583420  209.511163
2        106.861538  189.515537  230.771523  194.784387
3         61.227488  123.906250  156.582915  134.466667
4        164.000000         NaN         NaN         NaN
```



Conclusion:

- Weather and season are dependent as weather changes season changes

1. When weather is clear and season is spring overall rental is high
2. When weather is rain for all season rental is low
3. Avg rental count is highest for weather is clear and season is fall
4. Avg rental count is lowest for weather rain and season is spring

Insight/Recommendations

- There is no significant difference between working day and non working day
- There is significant difference between rental count across season
- we can conclude that Season does effect bike rental based on test
- From dist we can see that median of group are different season to season
- season 1 is data is not very spread compared to 2,3,4 , people usually take bike for short ride may they prefer to walk when weather is good
- There is significant difference between rental count across weather
- we can conclude that weather does effect bike rental based on test
- weather 1 is data is very spread compared to 2,3,4 , people usually take bike for short ride may they prefer to walk when weather is good
- Weather and season are dependent as weather changes season changes

- When weather is clear and season is spring overall rental is high
- When weather is rain for all season rental is low
- Avg rental count is highest for weather is clear and season is fall
- Avg rental count is lowest for weather rain and season is spring
- With increase on windspeed count tend to increase slightly , then it drops post 40 and then picks up again
- With increase in humidity till 25 count seems to increase but then it keeps droping as humidity increases
- As temp increase count seem to increase