About Yulu

Yulu is India's leading micro-mobility service provider, which offers unique vehicles for the daily commute. Starting off as a mission to eliminate traffic congestion in India, Yulu provides the safest commute solution through a user-friendly mobile app to enable shared, solo and sustainable commuting.

Yulu zones are located at all the appropriate locations (including metro stations, bus stands, office spaces, residential areas, corporate offices, etc) to make those first and last miles smooth, affordable, and convenient!

Yulu has recently suffered considerable dips in its revenues. They have contracted a consulting company to understand the factors on which the demand for these shared electric cycles depends. Specifically, they want to understand the factors affecting the demand for these shared electric cycles in the Indian market.

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:

Dataset Link: yulu_data.csv

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:
    - 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

Concept Used:

- Bi-Variate Analysis
- 2-sample t-test: testing for difference across populations
- ANNOVA
- Chi-square

```
yulu = pd.read_csv("yulu.txt")
yulu["datetime"] = pd.to_datetime(yulu.datetime)
yulu["date"] = yulu.datetime.dt.date

print(yulu.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        date
0        81        0.0       3          13     16  2011-01-01
1        80        0.0       8          32     40  2011-01-01
2        80        0.0       5          27     32  2011-01-01
3        75        0.0       3          10     13  2011-01-01
4        75        0.0       0           1      1  2011-01-01
```

```
print(yulu.info())
```

```
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   datetime    10886 non-null  datetime64[ns]
 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
 12  date        10886 non-null  object
```

```
yulu = yulu.astype({"season" : "object" , "holiday" : "object" ,
"workingday" : "object" , "weather" : "object"})
print(yulu.info())
```

```
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   datetime    10886 non-null  datetime64[ns]
 1   season      10886 non-null  object
 2   holiday     10886 non-null  object
 3   workingday  10886 non-null  object
 4   weather     10886 non-null  object
 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
 12  date        10886 non-null  object
 13  month       10886 non-null  int64
```

```
print(yulu.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
date          0
dtype: int64
```

```
print(yulu.describe())
```

```
              temp         atemp      humidity     windspeed        casual  \
count  10886.00000  10886.000000  10886.000000  10886.000000  10886.000000
mean      20.23086     23.655084     61.886460     12.799395     36.021955
std        7.79159      8.474601     19.245033      8.164537     49.960477
min        0.82000      0.760000      0.000000      0.000000      0.000000
25%       13.94000     16.665000     47.000000      7.001500      4.000000
50%       20.50000     24.240000     62.000000     12.998000     17.000000
75%       26.24000     31.060000     77.000000     16.997900     49.000000
max       41.00000     45.455000    100.000000     56.996900    367.000000

         registered         count         month
count  10886.000000  10886.000000  10886.000000
mean     155.552177    191.574132      6.521495
std      151.039033    181.144454      3.444373
min        0.000000      1.000000      1.000000
25%       36.000000     42.000000      4.000000
50%      118.000000    145.000000      7.000000
75%      222.000000    284.000000     10.000000
max      886.000000    977.000000     12.000000
```

```
shape = yulu.shape
print(f"Number of rows : {shape[0]} \nNumber of columns : {shape[1]}")
```

```
Number of rows : 10886
Number of columns : 13
```

Observation :
1. Data has no missing values.
2. Data has 10886 rows and 13 columns.
3. There is a huge difference between mean and median of the data and also the standard deviation is large so we can infer that there might be outliers present.

```
print(yulu.season.unique())
```

```
[1 2 3 4]
```

```
print(yulu.holiday.unique())
```

```
[0 1]
```

```
print(yulu.workingday.unique())
```

```
[0 1]
```

```
print(yulu.weather.unique())
```

```
[1 2 3 4]
```

```
print(yulu.season.value_counts())
4    2734
2    2733
3    2733
1    2686
Name: season, dtype: int64
```

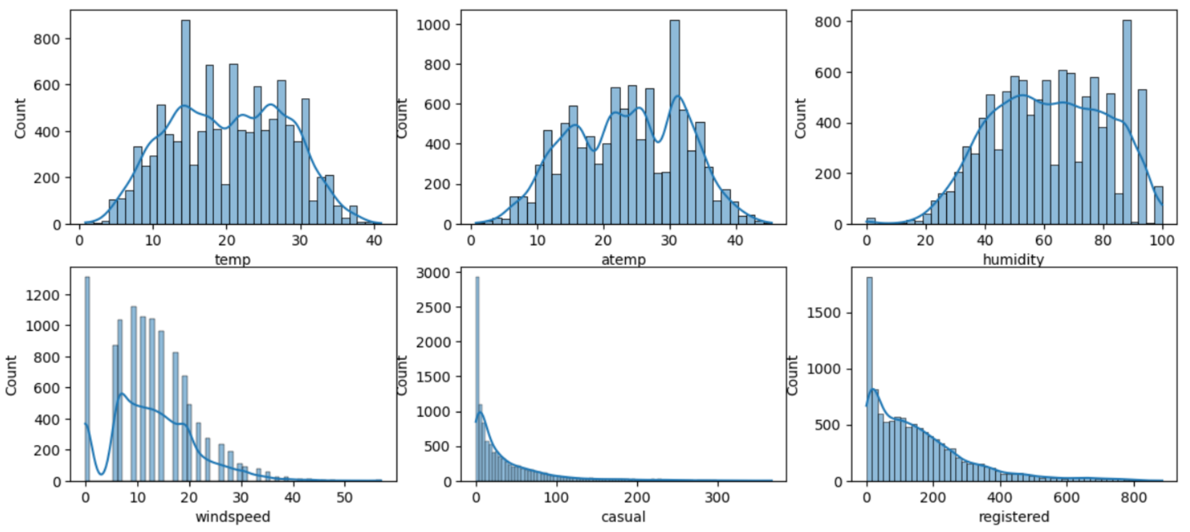
```
print(yulu.holiday.value_counts())
0    10575
1      311
Name: holiday, dtype: int64
```

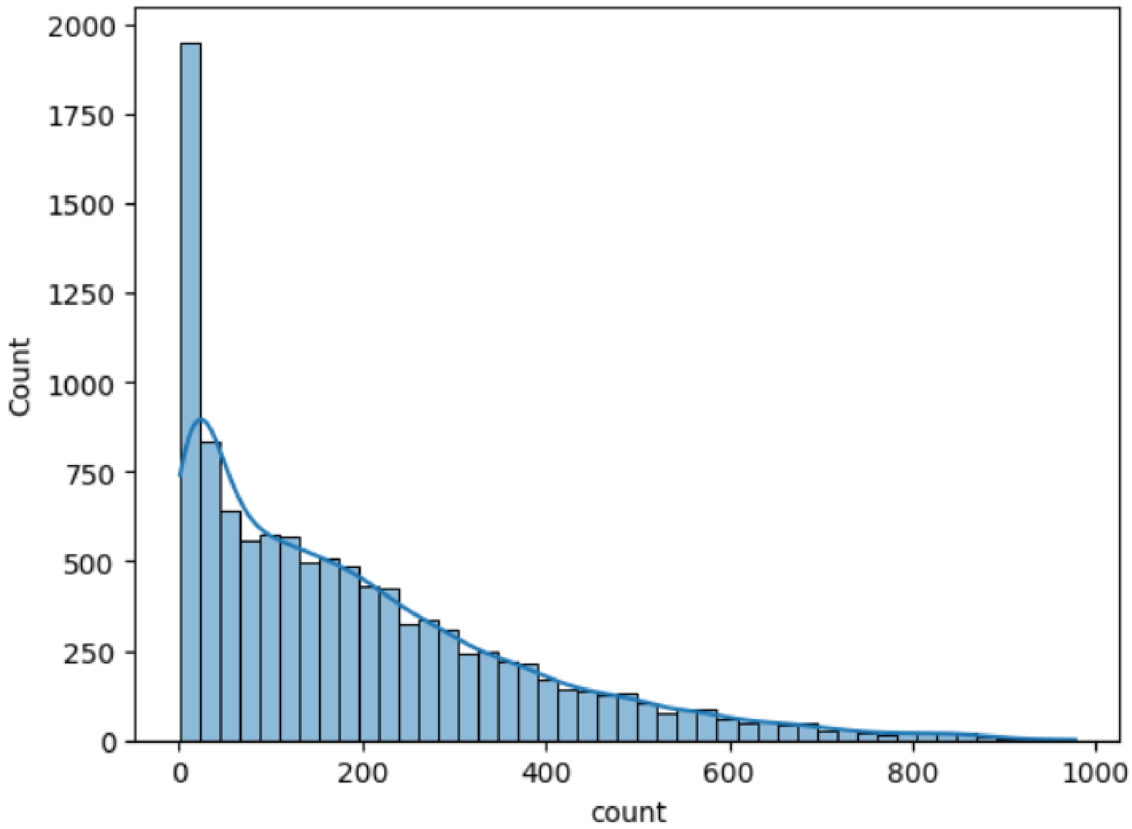
```
print(yulu.workingday.value_counts())
1    7412
0    3474
Name: workingday, dtype: int64
```

```
print(yulu.weather.value_counts())
1    7192
2    2834
3     859
4       1
Name: weather, dtype: int64
```

Uni-Variate Analysis :

```
num_cols = ["temp" , "atemp" , "humidity" , "windspeed" , "casual" ,
"registered" , "count"]
index = 0
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(14, 6))

for i in range(2) :
  for j in range(3) :
    sns.histplot(x = num_cols[index] , data = yulu , ax = axis[i, j] ,
kde = True )
    index = index + 1

plt.show()
sns.histplot(x = num_cols[-1] , data = yulu , kde = True)
plt.show()
```



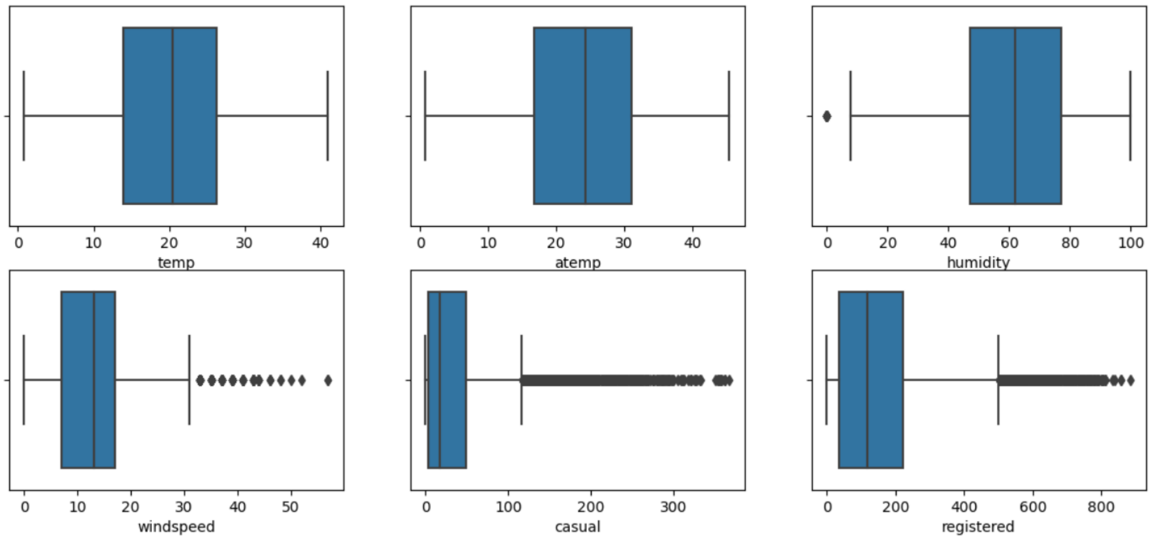


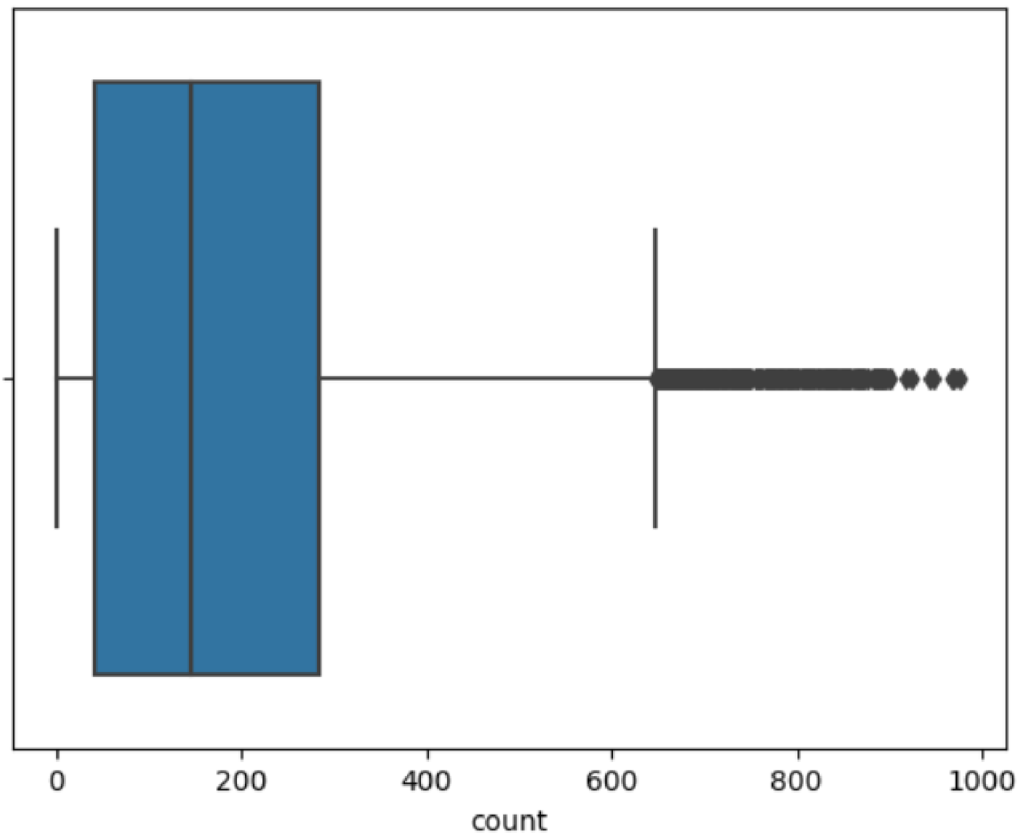
- **casual**, **registered** and **count** somewhat looks like **Log Normal Distribution.**
- **temp**, **atemp** and **humidity** looks like they follows the **Normal Distribution.**
- **windspeed** follows the **binomial distribution.**

Outlier Check :

```
num_cols = ["temp" , "atemp" , "humidity" , "windspeed" , "casual" ,
"registered" , "count"]
index = 0
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(14, 6))
```

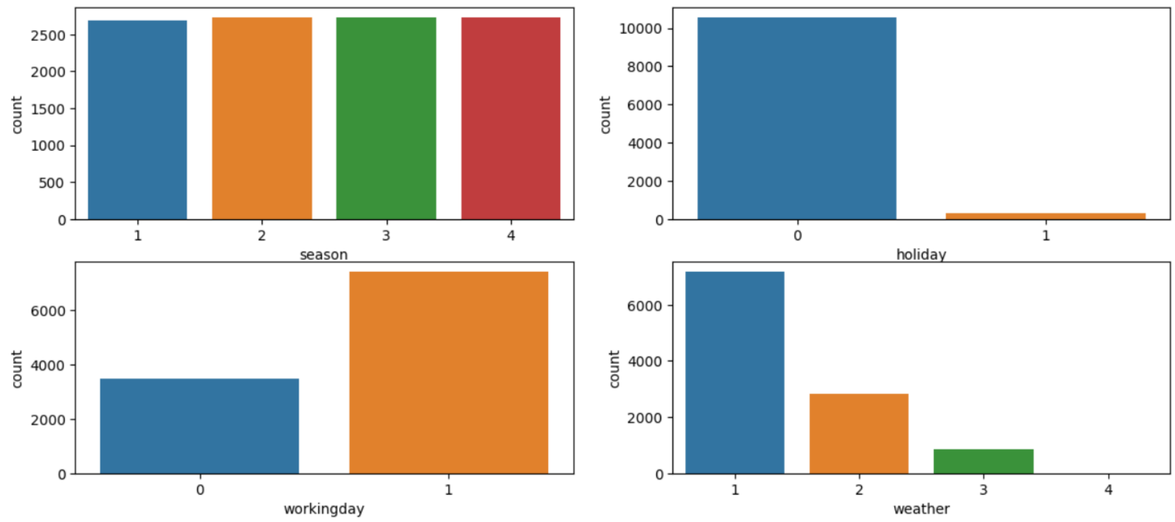
```python
for i in range(2) :
  for j in range(3) :
    sns.boxplot(x = num_cols[index] , data = yulu , ax = axis[i, j] )
    index = index + 1

plt.show()
sns.boxplot(x = num_cols[-1] , data = yulu)
plt.show()
```





- **Windspeed**, **casual**, **registered** and **count** have outliers in the data.

```
cat_cols = ["season" , "holiday" , "workingday" , "weather"]
index = 0
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(14, 6))

for i in range(2) :
  for j in range(2) :
    sns.countplot(x = cat_cols[index] , data = yulu , ax = axis[i, j]
)
    index = index + 1

plt.show()
```

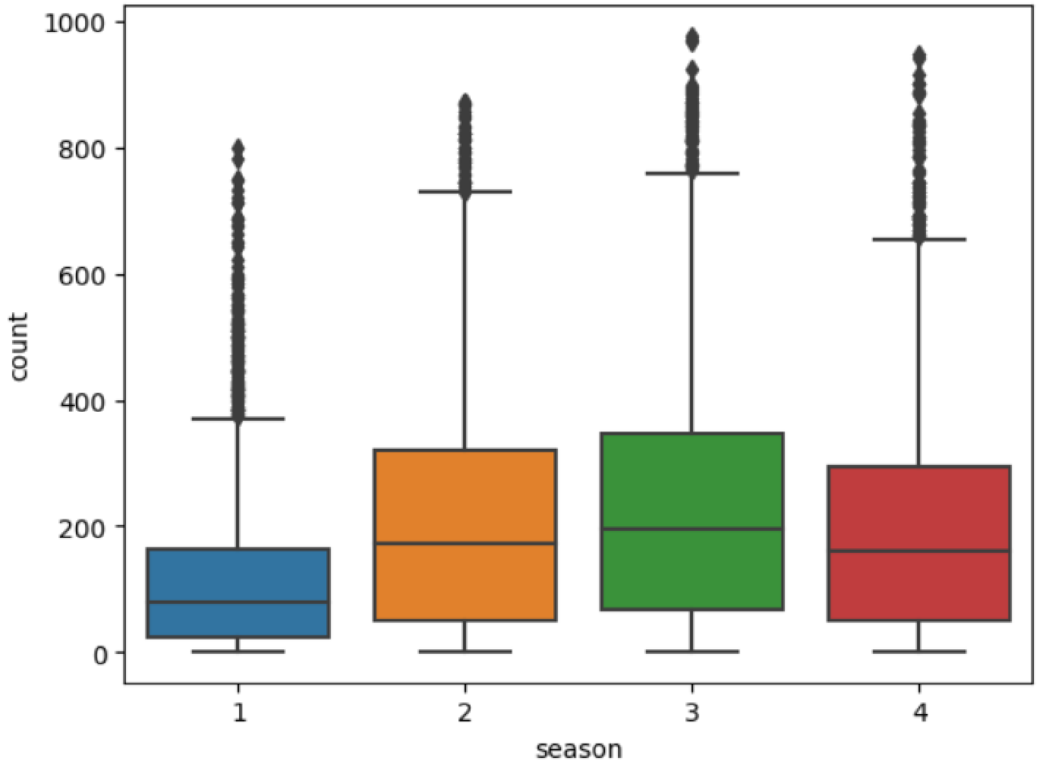


- Data looks common as it should be like equal number of days in each season, more working days and weather is mostly Clear, Few clouds, partly cloudy, partly cloudy.

Bi-Variate Analysis :

```
df_season = yulu.groupby("season")["count"].describe()
print(df_season)
         count        mean         std  min   25%    50%    75%    max
 season
 1      2686.0  116.343261  125.273974  1.0  24.0   78.0  164.0  801.0
 2      2733.0  215.251372  192.007843  1.0  49.0  172.0  321.0  873.0
 3      2733.0  234.417124  197.151001  1.0  68.0  195.0  347.0  977.0
 4      2734.0  198.988296  177.622409  1.0  51.0  161.0  294.0  948.0
```

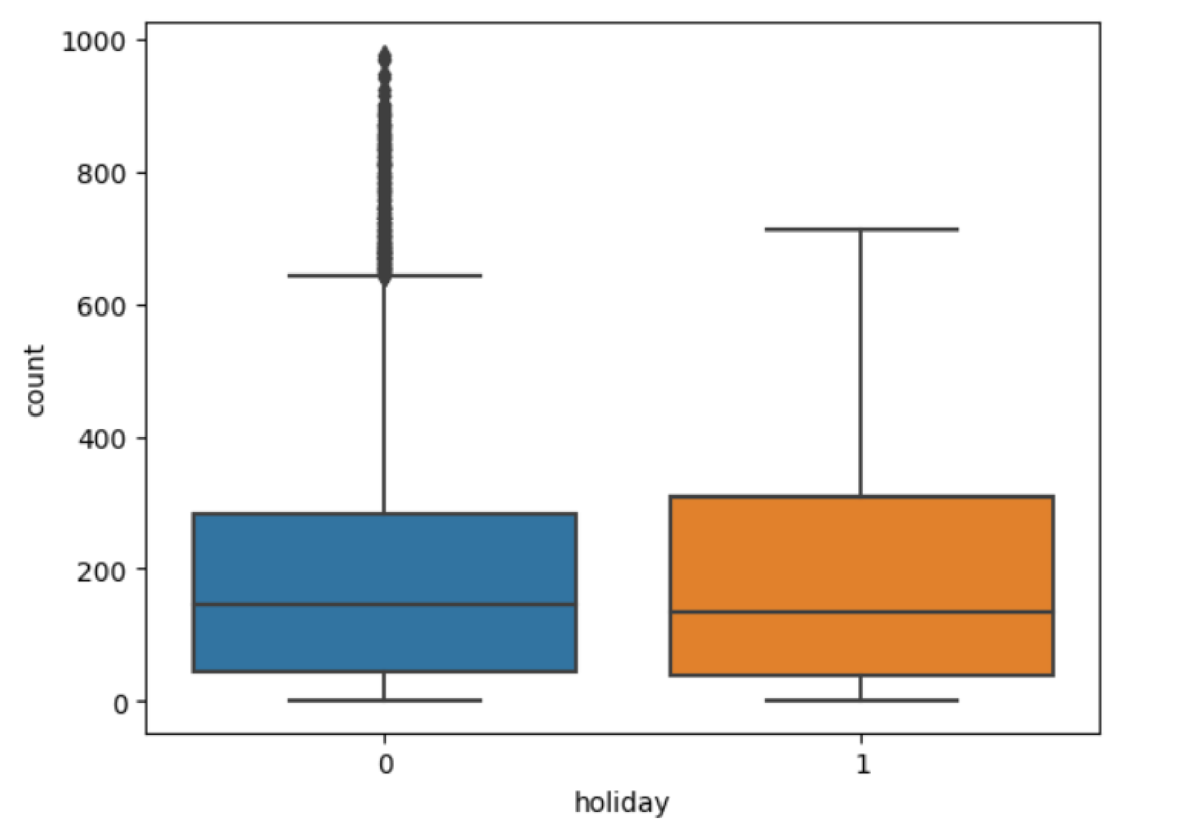
```
sns.boxplot(x = "season" , y = "count" , data = yulu)
plt.show()
```




```
holiday_describe = yulu.groupby("holiday")["count"].describe()
print(holiday_describe)
```

```
           count        mean         std  min   25%    50%    75%    max
holiday
0        10575.0  191.741655  181.513131  1.0  43.0  145.0  283.0  977.0
1          311.0  185.877814  168.300531  1.0  38.5  133.0  308.0  712.0
```

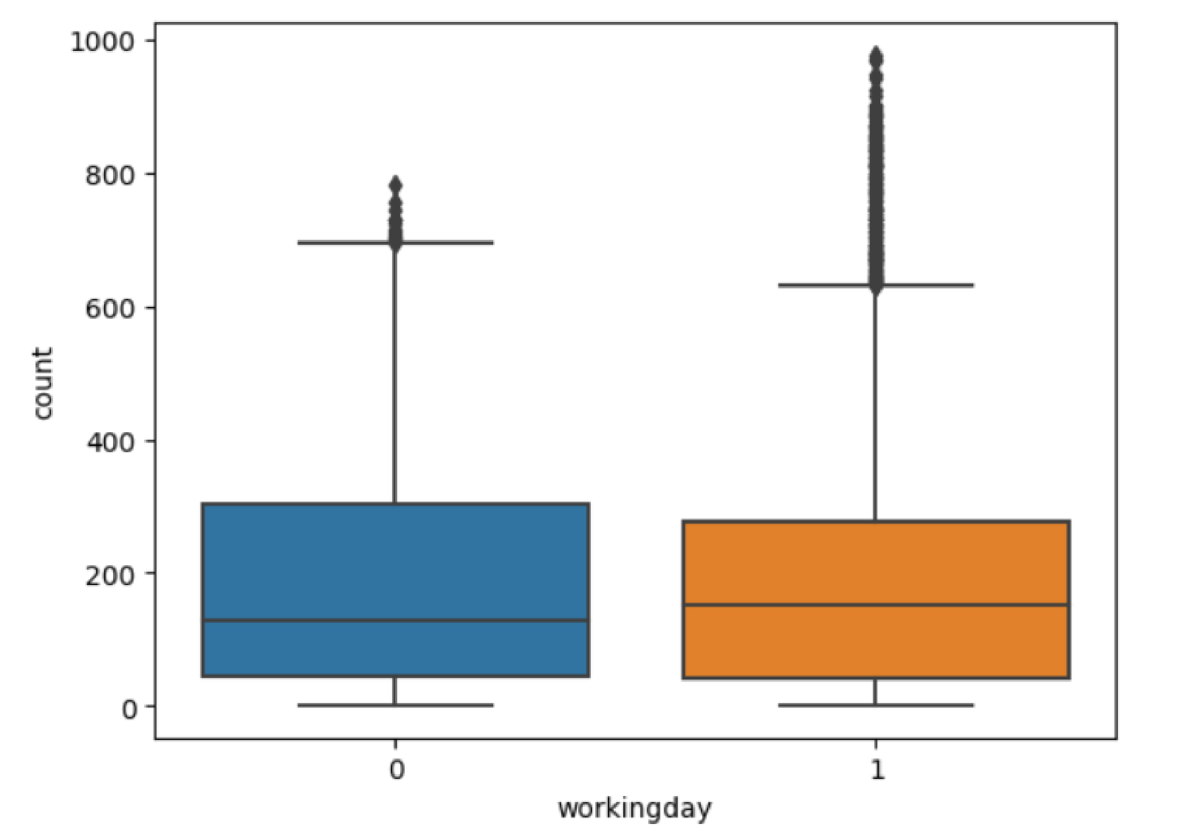
```
sns.boxplot(x = "holiday" , y = "count" , data = yulu)
plt.show()
```




```
workingday_describe = yulu.groupby("workingday")["count"].describe()
print(workingday_describe)
```

workingday	count	mean	std	min	25%	50%	75%	max
0	3474.0	188.506621	173.724015	1.0	44.0	128.0	304.0	783.0
1	7412.0	193.011873	184.513659	1.0	41.0	151.0	277.0	977.0

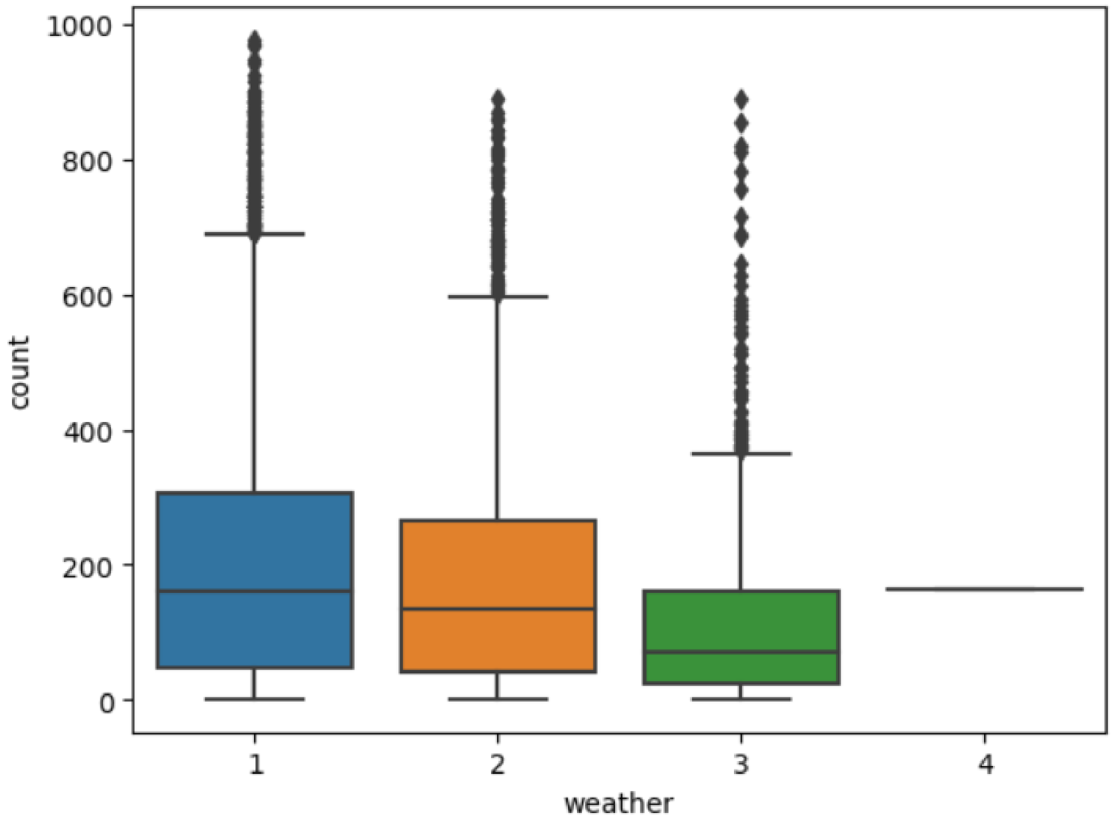
```
sns.boxplot(x = "workingday" , y = "count" , data = yulu)
plt.show()
```




```
weather_describe = yulu.groupby("weather")["count"].describe()
print(weather_describe)
```

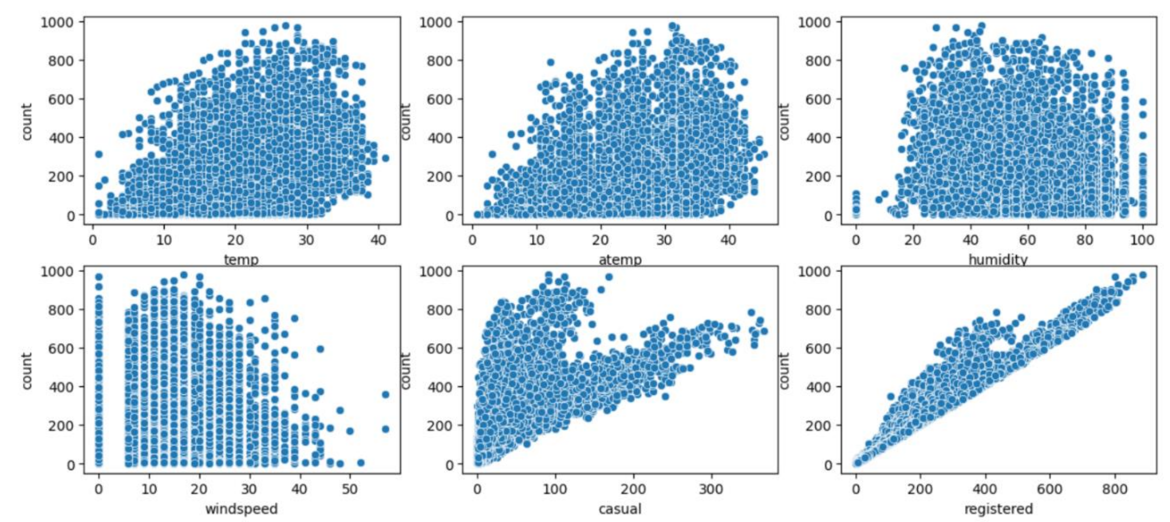
weather	count	mean	std	min	25%	50%	75%	max
1	7192.0	205.236791	187.959566	1.0	48.0	161.0	305.0	977.0
2	2834.0	178.955540	168.366413	1.0	41.0	134.0	264.0	890.0
3	859.0	118.846333	138.581297	1.0	23.0	71.0	161.0	891.0
4	1.0	164.000000	NaN	164.0	164.0	164.0	164.0	164.0

```
sns.boxplot(x = "weather" , y = "count" , data = yulu)
plt.show()
```



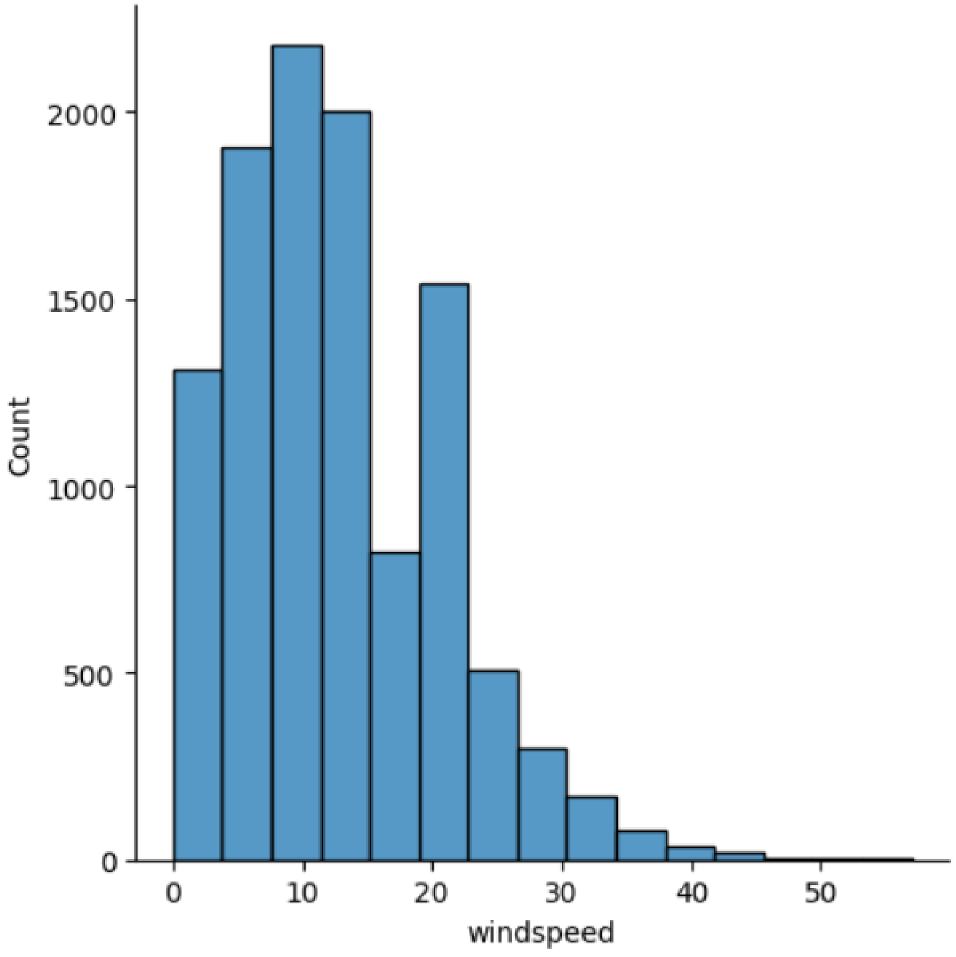
- In **summer** and **fall** seasons more bikes are rented as compared to other seasons.
- Whenever it's a **holiday** more bikes are rented.
- It is also clear from the working day also that whenever day is holiday or weekend, slightly more bikes were rented.
- Whenever there is **rain, thunderstorm, snow or fog**, there were less bikes were rented.

```
num_cols = ["temp" , "atemp" , "humidity" , "windspeed" , "casual" ,
"registered" , "count"]
index = 0
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(14, 6))

for i in range(2) :
  for j in range(3) :
    sns.scatterplot(x = num_cols[index] , y = "count" , data = yulu ,
ax = axis[i, j] )
    index = index + 1

plt.show()
```



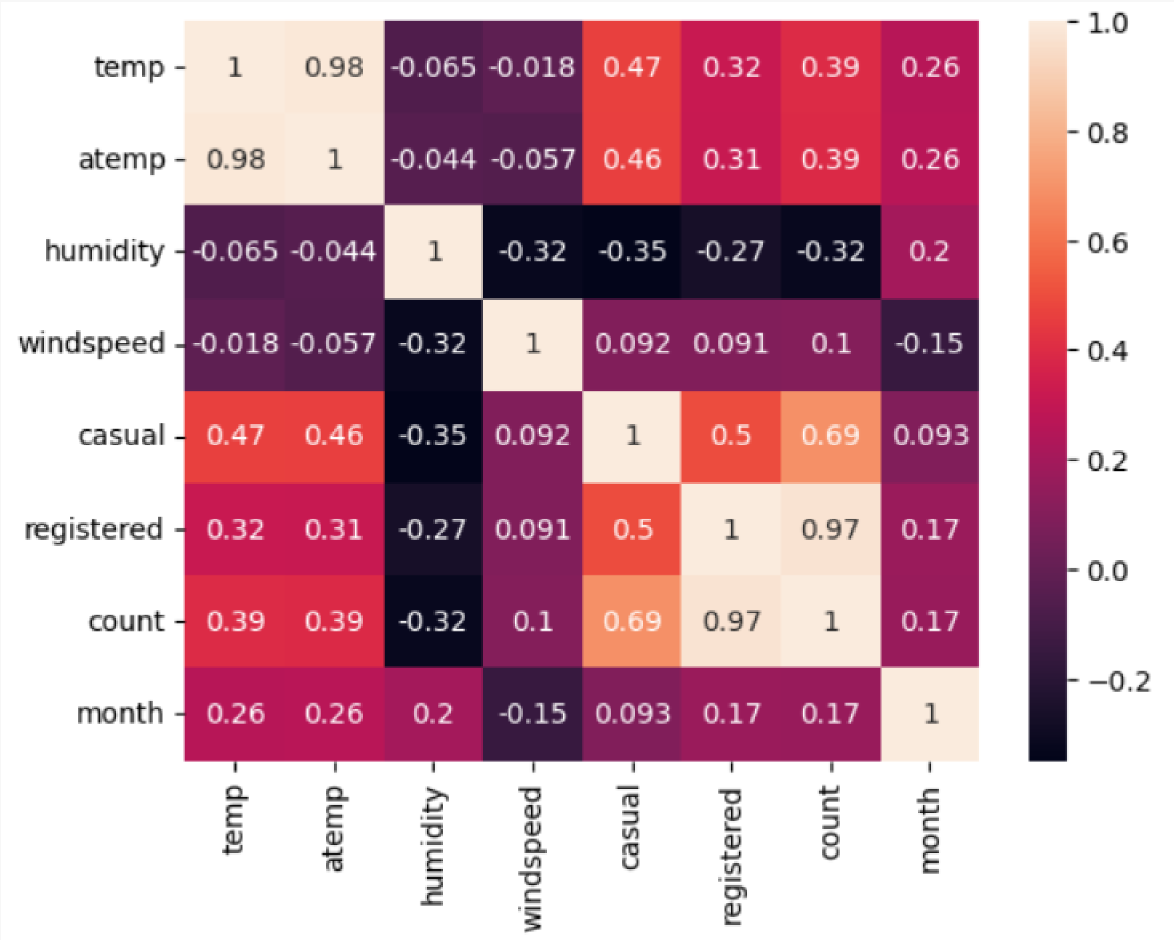
- Whenever the humidity is less than 20, number of bikes rented is very very low.
- Whenever the temperature is less than 10, number of bikes rented is less.
- Whenever the windspeed is greater than 35, number of bikes rented is less.

```
plt.figure(figsize = (15, 6))
sns.displot(x = "windspeed" , data = yulu , bins = 15)
plt.show()
```



Multi-Variate Analysis :

```
sns.heatmap(data = yulu.corr() , annot = True )
plt.show()
```



- Temperature and number of cycles rented are positively correlated.
- Humidity and number of cycles rented are negatively correlated.

Hypothesis Testing on Working Day and Electric cycles rented:

Null Hypothesis (H0) : Working Day has no effect on number of electric cycles rented.

Alternative Hypothesis (H1) : Working Day has an effect on number of electric cycles rented.

Significance Value (alpha) : 0.05

We will use T-test for this case as Working Day has 2 categories

```
t_stat , p_value = ttest_ind(yulu.loc[yulu.workingday == 1 , "count"]
, yulu.loc[yulu.workingday == 0 , "count"])

print(np.round(p_value , 2))
```

```
0.23
```

Since p_value is greater than significance values we can not reject null hypothesis. Hence we don't have enough evidence to conclude that working day has an effect on number of electric cycles rented.

Hypothesis Testing on Holiday and Electric cycles rented :

Null Hypothesis (H0) : Holiday has no effect on number of electric cycles rented.

Alternative Hypothesis (H1) : Holiday has an effect on number of electric cycles rented.

Significance Value (alpha) : 0.05

We will use T-test for this case as Holiday has 2 categories

```
t_stat , p_value = ttest_ind(yulu.loc[yulu.holiday == 1 , "count"] ,
yulu.loc[yulu.holiday == 0 , "count"])

print(np.round(p_value , 2))
```

```
0.57
```

Since p_value is greater than significance values we can not reject null hypothesis. Hence we don't have enough evidence to conclude that holiday has an effect on number of electric cycles rented.

Hypothesis Testing on Season and Electric cycles rented :

```
print(yulu.season.unique())
```

```
[1 2 3 4]
```

We will use ANOVA for this case as Season has more than 2 categories

Checking conditions of ANOVA :

1. Normal Distribution.
2. Categories should be independent of each other.
3. There should be equal variance between the categories.

If any of the above conditions fails we will not proceed with ANOVA for Hypothesis testing we will use KRUSKAL test.

Checking Normal Distribution for count column:
SHAPIRO TEST ->

Null Hypothesis : Data follow Normal Distribution.
Alternative Hypothesis : Data doesn't follow Normal Distribution.
Significance value = 0.05

```
# Taking sample size as 100
count_subset = yulu["count"].sample(100)
shapiro_stat , p_value2 = shapiro(count_subset)
print(p_value2)
```

```
2.114466290947803e-08
```

Since p_value is less than significance value we reject null hypothesis. Hence we don't have enough evidence to conclude that column count follows normal distribution.

Checking equal variance for the categories:
LEVENE TEST ->
Null Hypothesis : Variance is same for different categories.
Alternative Hypothesis : Variance is not same for different categories.
Significance value = 0.05

```
levene_stat , p_value = levene(yulu.loc[yulu.season == 1 , "count"] ,
yulu.loc[yulu.season == 2 , "count"], yulu.loc[yulu.season == 3 ,
"count"], yulu.loc[yulu.season == 4 , "count"])
print(p_value)
```

```
1.0147116860043298e-118
```

Since p_value is less than significance value we reject null hypothesis. Hence we don't have enough evidence to conclude that column season categories have equal variance.

-> As conditions of ANOVA is not satisfied we will proceed with KRUSKAL.

KRUSKAL TEST ->
Null Hypothesis : Weather has no effect on number of electric cycles rented.
Alternative Hypothesis : Weather has an effect on number of electric cycles rented.
Significance value = 0.05

```
kruskal_stat , p_value = kruskal(yulu.loc[yulu.season == 1 , "count"]
, yulu.loc[yulu.season == 2 , "count"], yulu.loc[yulu.season == 3 ,
"count"], yulu.loc[yulu.season == 4 , "count"])
print(p_value)
```

```
2.479008372608633e-151
```

Since p_value is less than significance value we can reject null hypothesis. Hence we have enough evidence to conclude that season has an effect on number of electric cycles rented.

If we use ANOVA :

ANOVA TEST ->

Null Hypothesis : Weather has no effect on number of electric cycles rented.

Alternative Hypothesis : Weather has an effect on number of electric cycles rented.

Significance value = 0.05

```
anova_stat , p_value = kruskal(yulu.loc[yulu.season == 1 , "count"] ,
yulu.loc[yulu.season == 2 , "count"], yulu.loc[yulu.season == 3 ,
"count"], yulu.loc[yulu.season == 4 , "count"])
print(p_value)
```

```
6.164843386499654e-149
```

Since p_value is less than significance value we can reject null hypothesis. Hence we do have enough evidence to conclude that season has an effect on number of electric cycles rented.

Hypothesis Testing on Weather and Electric cycles rented :

```
print(yulu.weather.unique())
```

```
[1 2 3 4]
```

We can use ANOVA for this case as Weather has more than 2 categories.

Checking conditions of ANOVA :

1. Normal Distribution.
2. Categories should be independent of each other.
3. There should be equal variance between the categories.

If any of the above conditions fails we will not proceed with ANOVA for Hypothesis testing we will use KRUSKAL.

Checking Normal Distribution for count column:
SHAPIRO TEST ->
Null Hypothesis : Data follow Normal Distribution.
Alternative Hypothesis : Data doesn't follow Normal Distribution.
Significance value = 0.05

```
# Taking sample size as 100
count_subset = yulu["count"].sample(100)
shapiro_stat , p_value2 = shapiro(count_subset)
print(p_value2)
```

```
7.298385895637693e-08
```

Since p_value is less than significance value we reject null hypothesis. Hence we don't have enough evidence to conclude that column count follows normal distribution.

Checking equal variance for the categories:
LEVENE TEST ->
Null Hypothesis : Variance is same for different categories.
Alternative Hypothesis : Variance is not same for different categories.
Significance value = 0.05

```
levene_stat , p_value = levene(yulu.loc[yulu.weather == 1 , "count"] ,
yulu.loc[yulu.weather == 2 , "count"], yulu.loc[yulu.weather == 3 ,
"count"], yulu.loc[yulu.weather == 4 , "count"])
```

```
print(p_value)
```

```
3.504937946833238e-35
```

Since p_value is less than significance value we reject null hypothesis. Hence we don't have enough evidence to conclude that column weather categories have equal variance.

-> As conditions of ANOVA is not satisfied we will proceed with KRUSKAL.

KRUSKAL TEST ->
Null Hypothesis : Weather has no effect on number of electric cycles rented.
Alternative Hypothesis : Weather has an effect on number of electric cycles rented.
Significance value = 0.05

```
kruskal_stat , p_value = kruskal(yulu.loc[yulu.weather == 1 , "count"]
, yulu.loc[yulu.weather == 2 , "count"], yulu.loc[yulu.weather == 3 ,
"count"], yulu.loc[yulu.weather == 4 , "count"])
print(p_value)
```

```
3.501611300708679e-44
```

Since p_value is less than significance value we can reject null hypothesis. Hence we have enough evidence to conclude that weather has an effect on number of electric cycles rented.

If we use ANOVA :
ANOVA TEST ->

Null Hypothesis : Weather has no effect on number of electric cycles rented.

Alternative Hypothesis : Weather has an effect on number of electric cycles rented.

Significance value = 0.05

```
anova_stat , p_value = f_oneway(yulu.loc[yulu.weather == 1 , "count"]
, yulu.loc[yulu.weather == 2 , "count"], yulu.loc[yulu.weather == 3 ,
"count"], yulu.loc[yulu.weather == 4 , "count"])
print(p_value)
```

```
5.482069475935669e-42
```

Since p_value is less than significance value we can reject null hypothesis. Hence we have enough evidence to conclude that weather has an effect on number of electric cycles rented.

Hypothesis Testing on Weather and Season:

Null Hypothesis (H0) : Weather and Season are independent on each other.

Alternative Hypothesis (H1) : Weather and Season are dependent on each other.

Significance Value (alpha) : 0.05

We will use Chi-Square test for this case as we are dealing with two individual categorical fields.

```
crosstab = pd.crosstab(yulu.weather , yulu.season)
print(crosstab)
```

```
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
```

```
chi2_stat , p_value, df, exp_value = chi2_contingency(crosstab)
print(p_value)
```

```
1.5499250736864862e-07
```

Since p_value is less than significance values we reject null hypothesis. Hence the column season and weather are dependent on each other.

Insights :

- In **summer** and **fall** seasons more bikes are rented as compared to other seasons.
- Whenever it’s a **holiday** more bikes are rented.
- It is also clear from the working day also that whenever day is holiday or weekend, slightly more bikes were rented.
- Whenever there is **rain, thunderstorm, snow or fog**, there were less bikes were rented.
- Whenever the humidity is less than 20, number of bikes rented is very very low.

- Whenever the temperature is less than 10, number of bikes rented is less.
- Whenever the windspeed is greater than 35, number of bikes rented is less.

Recommendations :

- In **summer** and **fall** seasons the company should have more bikes in stock to be rented. Because the demand in these seasons is higher as compared to other seasons.
- With a significance level of 0.05, working day and holiday has no effect on the number of bikes being rented.
- With a significance level of 0.05, Season and Weather has an effect on the number of bikes being rented.
- In very low humid days, company should have less bikes in the stock to be rented.
- Whenever temperature is less than 10 or in very cold days, company should have less bikes.
- Whenever the windspeed is greater than 35 or in thunderstorms, company should have less bikes in stock to be rented.