Business Case: Yulu - Hypothesis Testing

- ***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.***

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sn
import scipy.stats as stats
```

In [2]:
```
# Importing the data
yulu = pd.read_csv("D://bike_sharing.txt")
```

In [3]:
```
yulu
```

Out[3]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0000	3	13	16
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0000	8	32	40
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0000	5	27	32
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0000	3	10	13
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0000	0	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...
10881	2012-12-19 19:00:00	4	0	1	1	15.58	19.695	50	26.0027	7	329	336
10882	2012-12-19 20:00:00	4	0	1	1	14.76	17.425	57	15.0013	10	231	241
10883	2012-12-19 21:00:00	4	0	1	1	13.94	15.910	61	15.0013	4	164	168
10884	2012-12-19 22:00:00	4	0	1	1	13.94	17.425	61	6.0032	12	117	129
10885	2012-12-19 23:00:00	4	0	1	1	13.12	16.665	66	8.9981	4	84	88

10886 rows × 12 columns

In [4]:
```
#checking the datatypes
yulu.dtypes
```

Out[4]:
```
datetime       object
season          int64
holiday         int64
workingday      int64
weather         int64
temp          float64
atemp         float64
humidity        int64
windspeed     float64
casual          int64
registered      int64
count           int64
dtype: object
```

In [5]:
```
yulu.isna().sum()
```

Out[5]:
```
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
```

In [6]:
```
yulu.duplicated().sum()
```

Out[6]:
```
0
```

In [7]:
```
yulu.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
```

Datatype of following attributes needs to changed to proper data type

- datetime - to datetime
- season - to categorical
- holiday - to categorical
- workingday - to categorical
- weather - to categorical

In [8]:
```
yulu['datetime'] = pd.to_datetime(yulu['datetime'])

cate_cl= ['season', 'holiday', 'workingday', 'weather']
for cl in cate_cl:
    yulu[cl] = yulu[cl].astype('object')
```

In [9]:
```
yulu.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  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
dtypes: datetime64[ns](1), float64(3), int64(4), object(4)
memory usage: 1020.7+ KB
```

In [10]:
```
yulu.isnull().sum()
```

Out[10]:
```
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
```

- There are no missing values in the dataset.
- casual and registered attributes might have outliers because their mean and median are very far away to one another and the value of standard deviation is also high which tells us that there is high variance in the data of these attributes.

In [ ]:

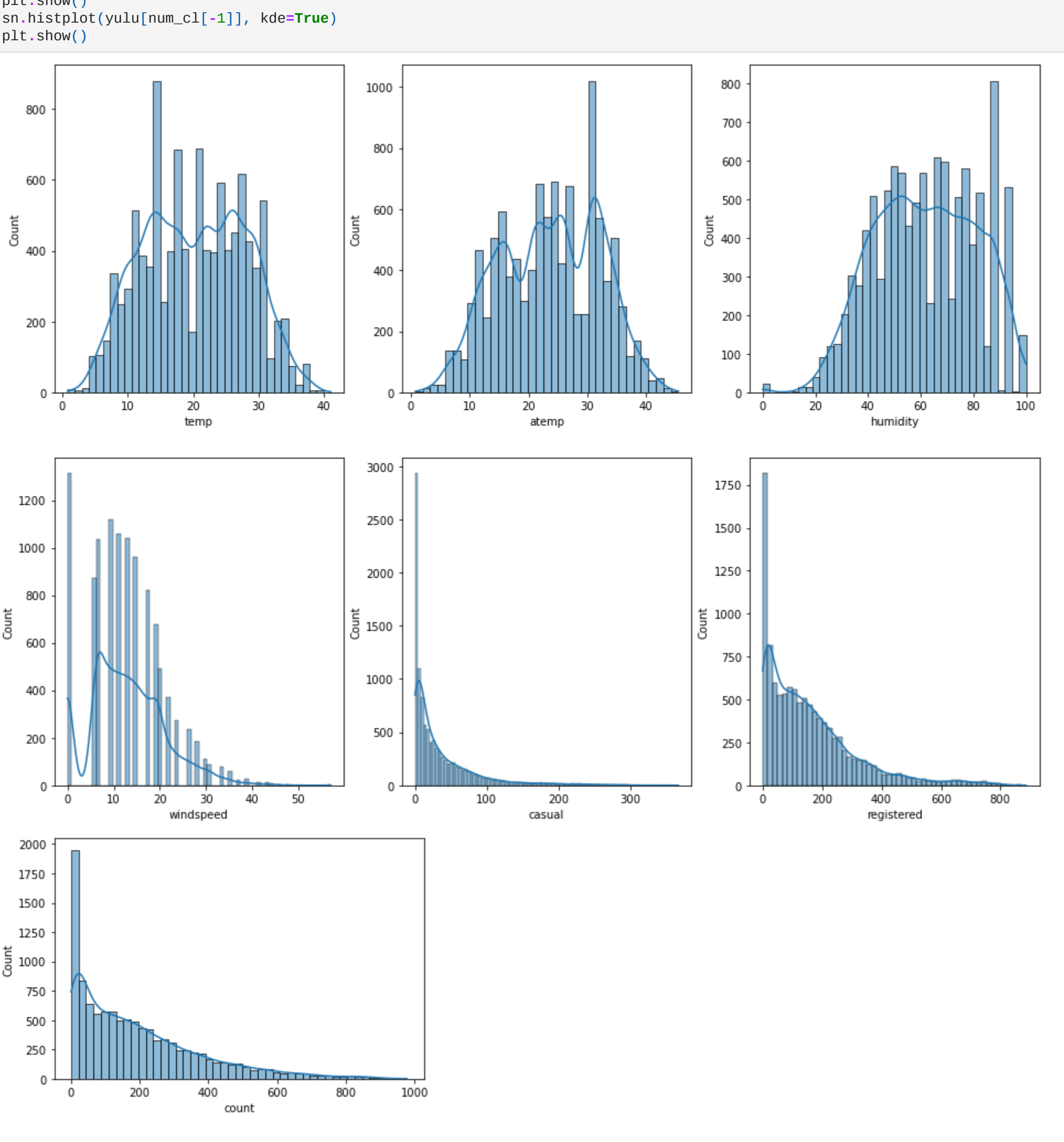
In [11]:
```
# No of unique values in each categorical columns
yulu[cate_cl].melt().groupby(['variable', 'value'])[['value']].count()
```

Out[11]:

variable	value	value
holiday	0	10575
	1	311
season	1	2686
	2	2733
	3	2733
	4	2734
weather	1	7192
	2	2834
	3	859
	4	1
workingday	0	3474
	1	7412

Univariate Analysis

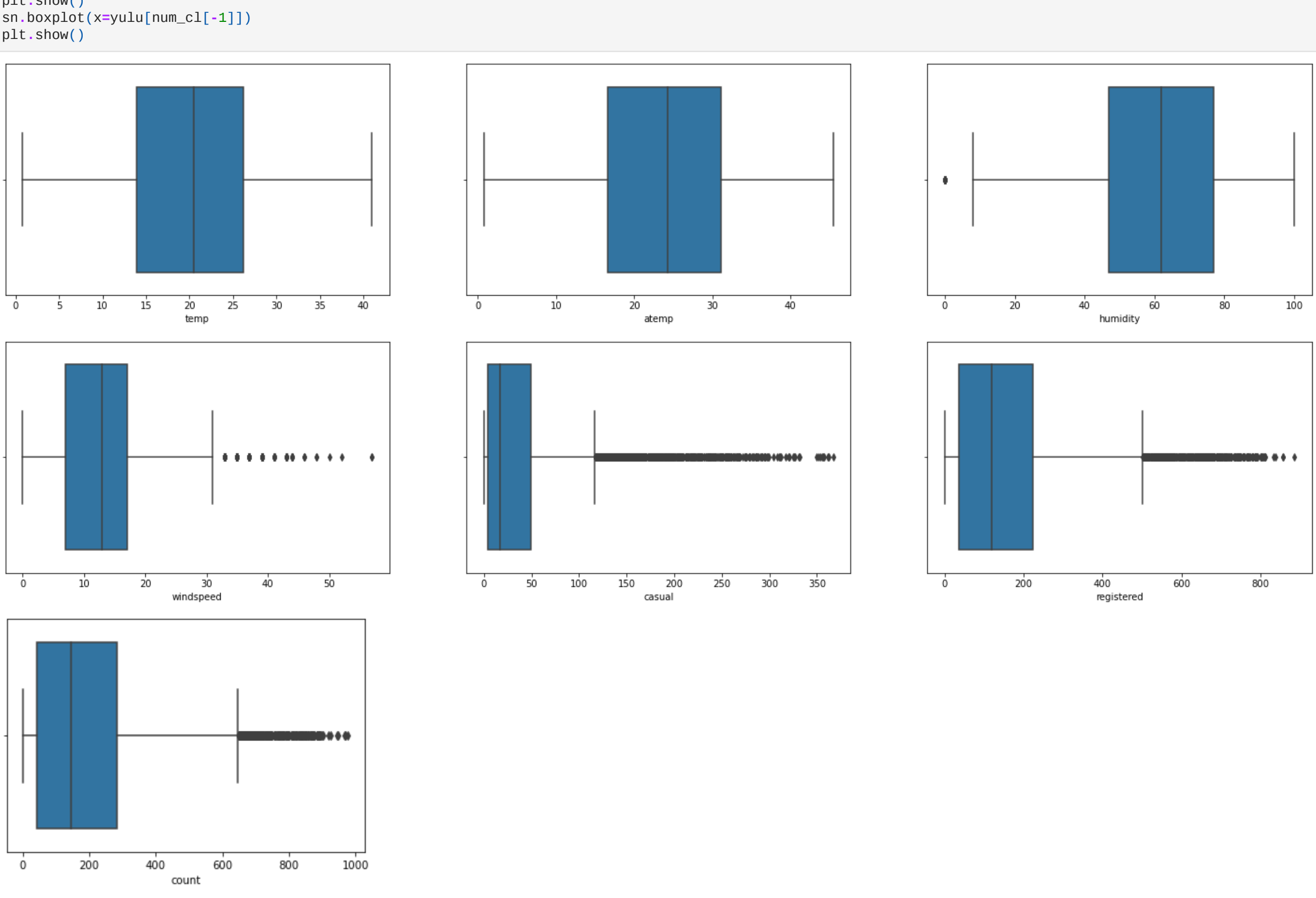
In [12]:
```
num_cl = ['temp', 'atemp', 'humidity', 'windspeed', 'casual', 'registered','count']

fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(16, 12))

index = 0
for row in range(2):
    for cl in range(3):
        sn.histplot(yulu[num_cl[index]], ax=axis[row, cl], kde=True)
        index += 1

plt.show()
sn.histplot(yulu[num_cl[-1]], kde=True)
plt.show()
```



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

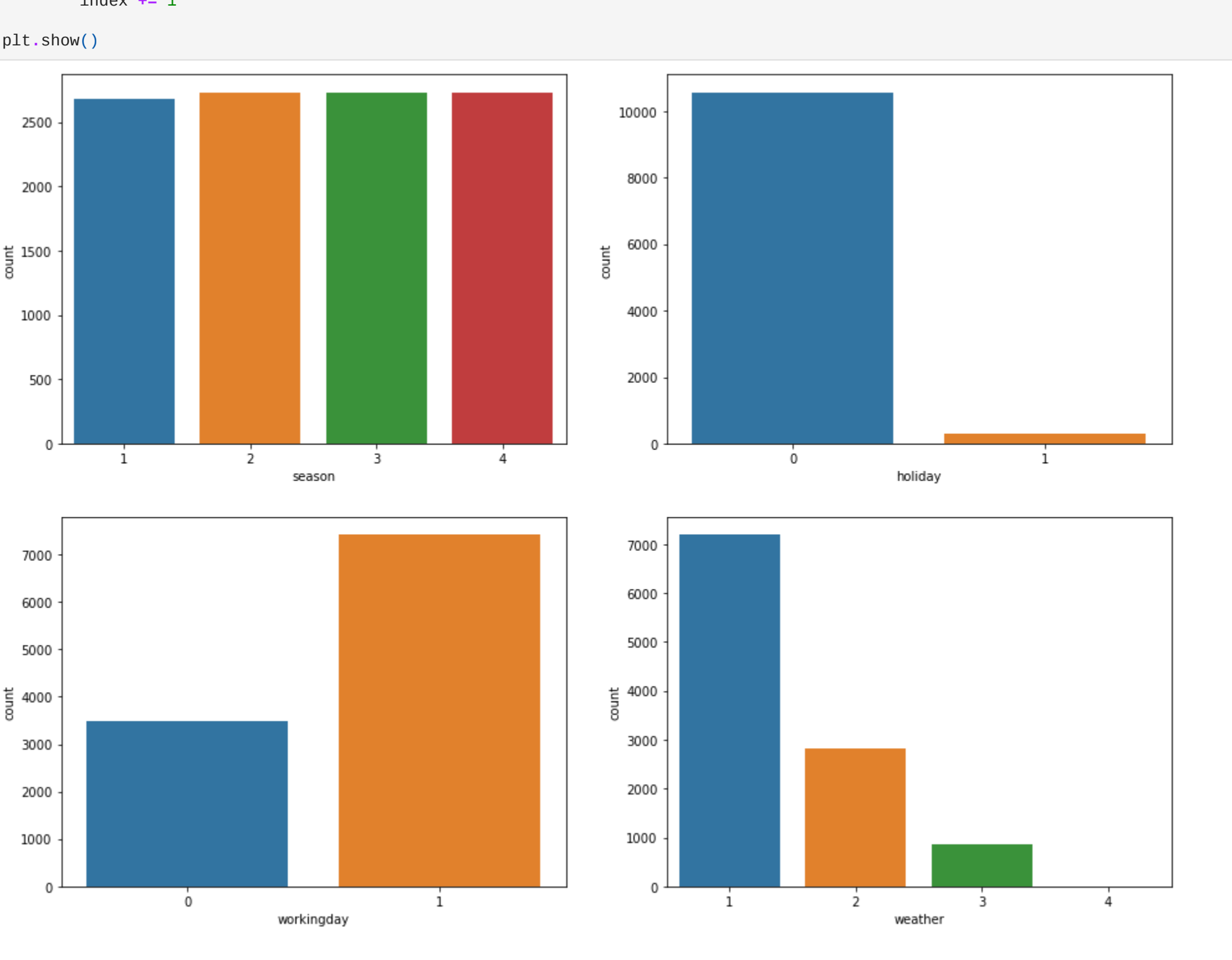
Detect outliers in the data

In [13]:
```
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(25, 10))

index = 0
for row in range(2):
    for cl in range(3):
        sn.boxplot(x=yulu[num_cl[index]], ax=axis[row, cl])
        index += 1

plt.show()
sn.boxplot(x=yulu[num_cl[-1]])
plt.show()
```



- Looks like humidity, casual, registered and count have outliers in the data.

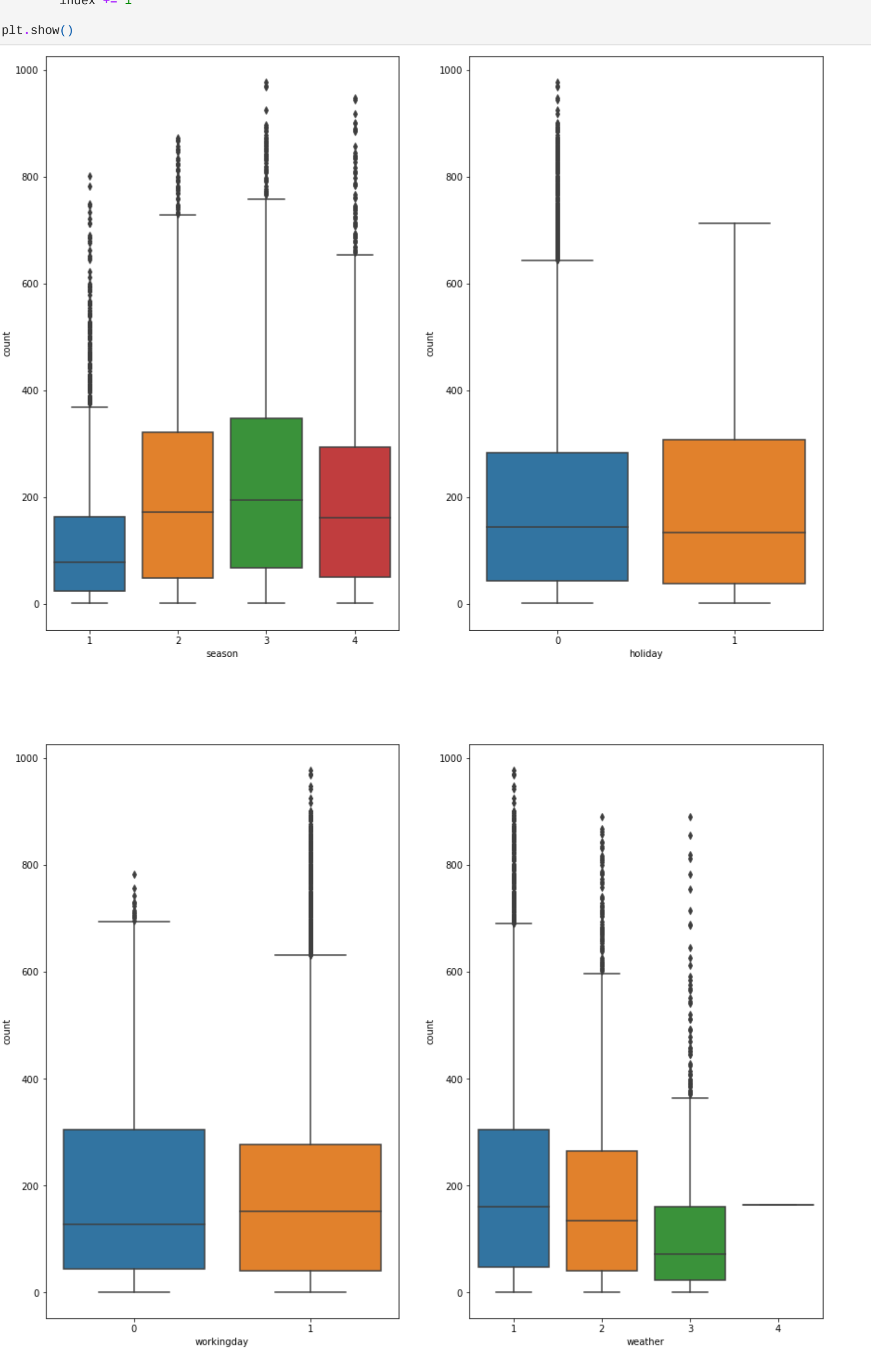
In [14]:
```
# countplot of each categorical column
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(16, 12))

index = 0
for row in range(2):
    for cl in range(2):
        sn.countplot(data=yulu, x=cate_cl[index], ax=axis[row, cl])
        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

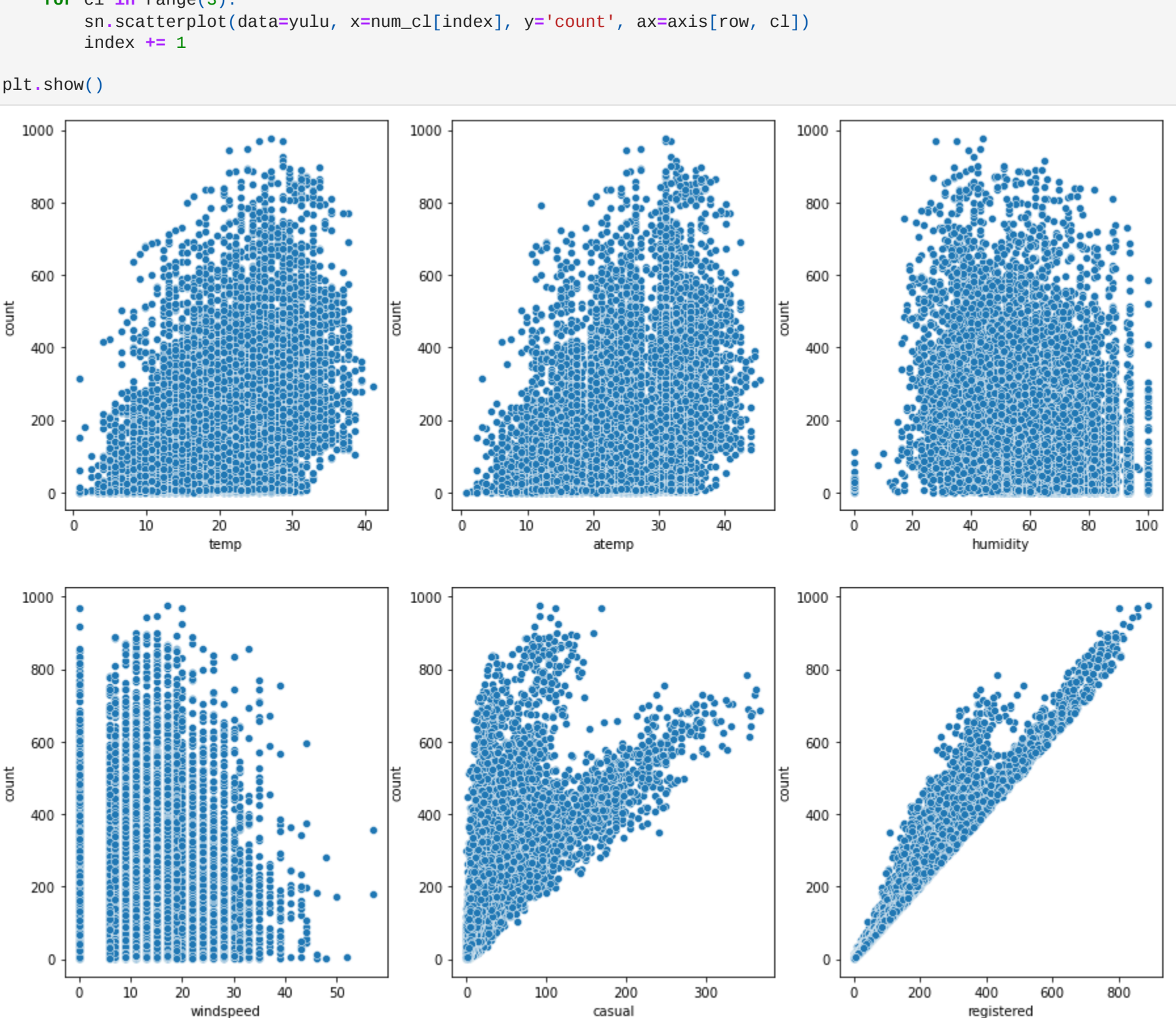
In [15]:
```
# plotting categorical variables againt count using boxplots
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(15, 25))

index = 0
for row in range(2):
    for cl in range(2):
        sn.boxplot(data=yulu, x=cate_cl[index], y='count', ax=axis[row, cl])
        index += 1

plt.show()
```



The insights based on EDA

- In summer and fall seasons more bikes are rented as compared to other seasons.
- Whenever its a holiday more bikes are rented.
- It is also clear from the workingday 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

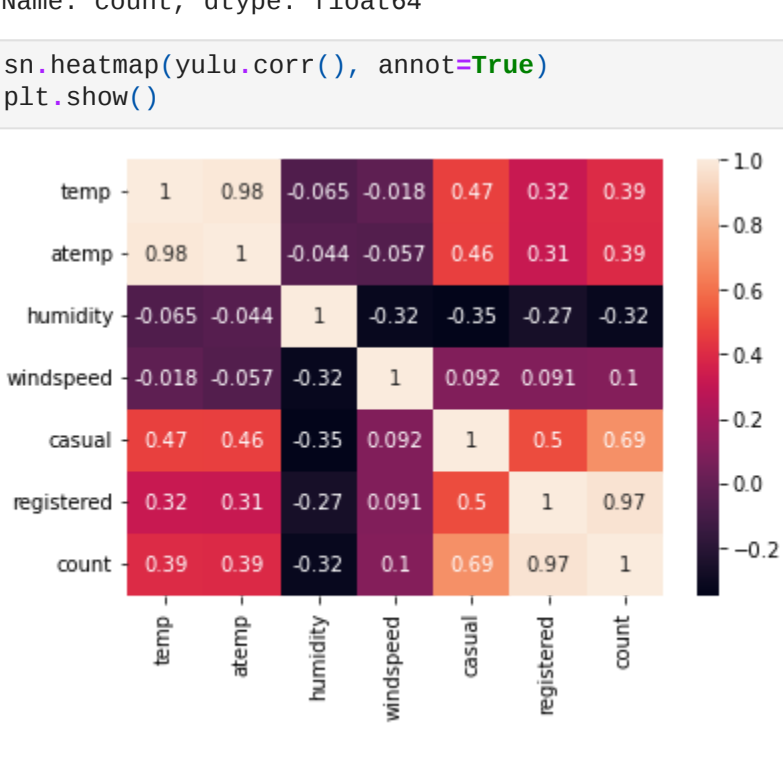
In [16]:
```
# plotting numerical variables againt count using scatterplot
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(15, 12))

index = 0
for row in range(2):
    for cl in range(3):
        sn.scatterplot(data=yulu, x=num_cl[index], y='count', ax=axis[row, cl])
        index += 1

plt.show()
```



In [17]:
```
# understanding the correlation between count and numerical variables
yulu.corr()['count']
```

Out[17]:
```
temp          0.394454
atemp         0.389784
humidity     -0.317371
windspeed     0.101369
casual        0.690414
registered    0.970948
count         1.000000
Name: count, dtype: float64
```

In [18]:
```
sn.heatmap(yulu.corr(), annot=True)
plt.show()
```



Hypothesis Testing - 1

- *Null Hypothesis: Working day has no effect on the number of cycles being rented.*
- *Alternate Hypothesis: Working day has effect on the number of cycles being rented.*
- *Significance level (alpha): 0.05 (default)*
- *We will use the 2-Sample T-Test to test the hypothess defined above*

- Here, the ratio is 34040.69 / 30171.34 which is less than 4:1

In [19]:
```
# 2- Sample T-Test
Batch_1 = yulu[yulu['workingday']==0]['count'].values
Batch_2 = yulu[yulu['workingday']==1]['count'].values

np.var(Batch_1), np.var(Batch_2)
```

Out[19]:
```
(30171.346098942427, 34040.69710674686)
```

In [20]:
```
stats.ttest_ind(a=Batch_1, b=Batch_2, equal_var=True)
```

Out[20]:
```
Ttest_indResult(statistic=-1.2096277376026694, pvalue=0.22644804226361348)
```

Since p-value is greater than 0.05 so we can not reject the Null hypothesis. We don't have the sufficient evidence to say that working day has effect on the number of cycles being rented

Hypothesis Testing - 2

- **Null Hypothesis:** *Number of cycles rented is similar in different weather and season.*
- **Alternate Hypothesis:** *Number of cycles rented is not similar in different weather and season.*
- **Significance level (alpha):** *0.05*
- *Here, we will use the ANOVA to test the hypothess defined above*

In [21]:
```
# defining the data groups for the ANOVA

B1 = yulu[yulu['weather']==1]['count'].values
B2 = yulu[yulu['weather']==2]['count'].values
B3 = yulu[yulu['weather']==3]['count'].values
B4 = yulu[yulu['weather']==4]['count'].values

B5 = yulu[yulu['season']==1]['count'].values
B6 = yulu[yulu['season']==2]['count'].values
B7 = yulu[yulu['season']==3]['count'].values
B8 = yulu[yulu['season']==4]['count'].values

# conduct the one-way anova
stats.f_oneway(B1, B2, B3, B4, B5, B6, B7, B8)
```

Out[21]:
```
F_onewayResult(statistic=127.96661249562491, pvalue=2.8074771742434642e-185)
```

Since p-value is less than 0.05, we reject the null hypothesis. This implies that Number of cycles rented is not similar in different weather and season conditions

Hypothesis Testing - 3

- **Null Hypothesis (H0):** Weather is independent of the season
- **Alternate Hypothesis (H1):** Weather is not independent of the season
- **Significance level (alpha):** 0.05
- We will use chi-square test to test hypyothesis

In [22]:
```
data_table = pd.crosstab(yulu['season'], yulu['weather'])
print("Observed values:")
data_table
```

Observed values:

Out[22]:

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

In [23]:
```
val = stats.chi2_contingency(data_table)
expected_values = val[3]
expected_values
```

Out[23]:
```
array([[1.77454639e+03, 6.99258130e+02, 2.11948742e+02, 2.46738931e-01],
       [1.80559765e+03, 7.11493845e+02, 2.15657450e+02, 2.51056403e-01],
       [1.80559765e+03, 7.11493845e+02, 2.15657450e+02, 2.51056403e-01],
       [1.80625831e+03, 7.11754180e+02, 2.15736359e+02, 2.51148264e-01]])
```

In [24]:
```
nrows, ncols = 4, 4
dof = (nrows-1)*(ncols-1)
print("degrees of freedom: ", dof)
alpha = 0.05


chi_sqr = sum([(o-e)**2/e for o, e in zip(data_table.values, expected_values)])
chi_sqr_statistic = chi_sqr[0] + chi_sqr[1]
print("chi-square test statistic: ", chi_sqr_statistic)

critical_val = stats.chi2.ppf(q=1-alpha, df=dof)
print(f"critical value: {critical_val}")

p_val = 1-stats.chi2.cdf(x=chi_sqr_statistic, df=dof)
print(f"p-value: {p_val}")

if p_val <= alpha:
    print("\nSince p-value is less than the alpha 0.05, We reject the Null Hypothesis. Meaning that\
    Weather is dependent on the season.")
else:
    print("Since p-value is greater than the alpha 0.05, We do not reject the Null Hypothesis")
```

```
degrees of freedom:  9
chi-square test statistic:  44.09441248632364
critical value: 16.918977604620448
p-value: 1.3560001579371317e-06

Since p-value is less than the alpha 0.05, We reject the Null Hypothesis. Meaning that    Weather is dependent on the season.
```

Insights

- In summer and fall seasons more bikes are rented as compared to other seasons.
- Whenever its a holiday more bikes are rented.
- It is also clear from the workingday 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, workingday has no 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 temprature 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

In [ ]: