yulu-hypothesis-testing

September 4, 2023

[1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
from scipy import stats
warnings.filterwarnings('ignore')
```

##Structure and Characteristics of the dataset

[2]:

```
df=pd.read_csv('yulu_data.csv')
df.head(2)
```

[2]:

```
              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

   humidity  windspeed  casual  registered  count
0        81        0.0       3          13     16
1        80        0.0       8          32     40
```

[3]:

```
df.shape
```

[3]:

```
(10886, 12)
```

[4]:

```
df.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
```

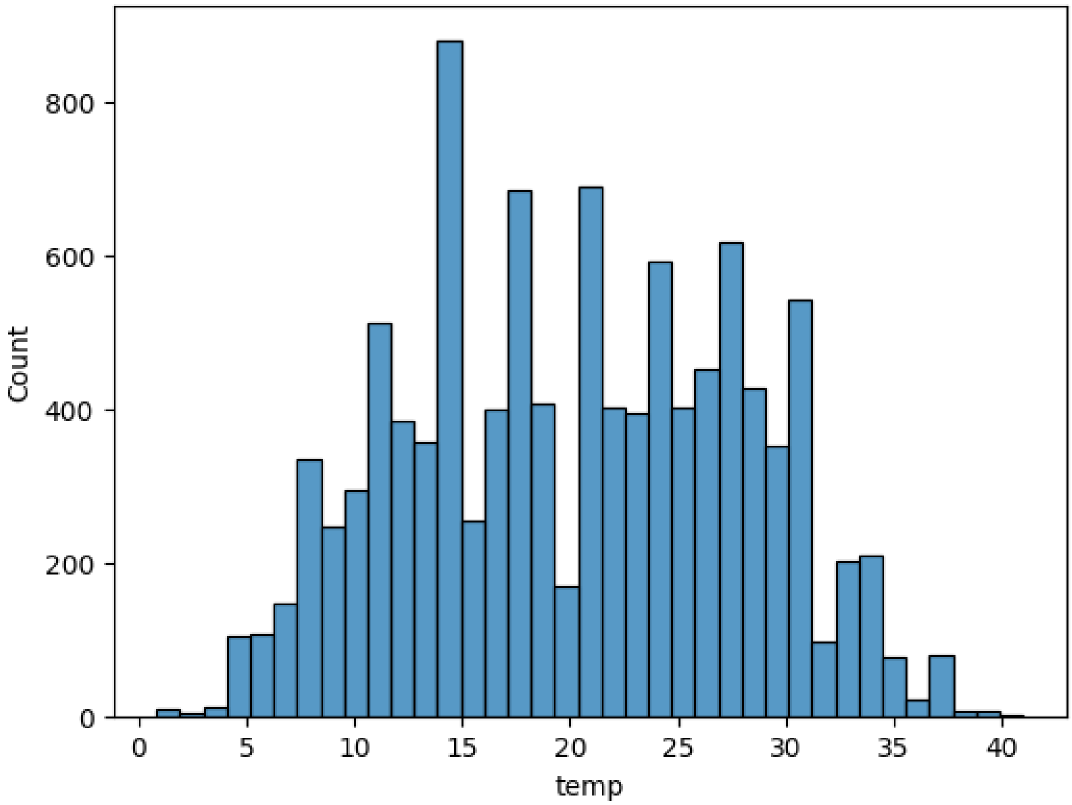
```
[ ]: df.describe()
```

```
[ ]:              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
```

From the above data, we can get the statistical values of the dataset like Mean, Minimum, Maximum, Count and so on.

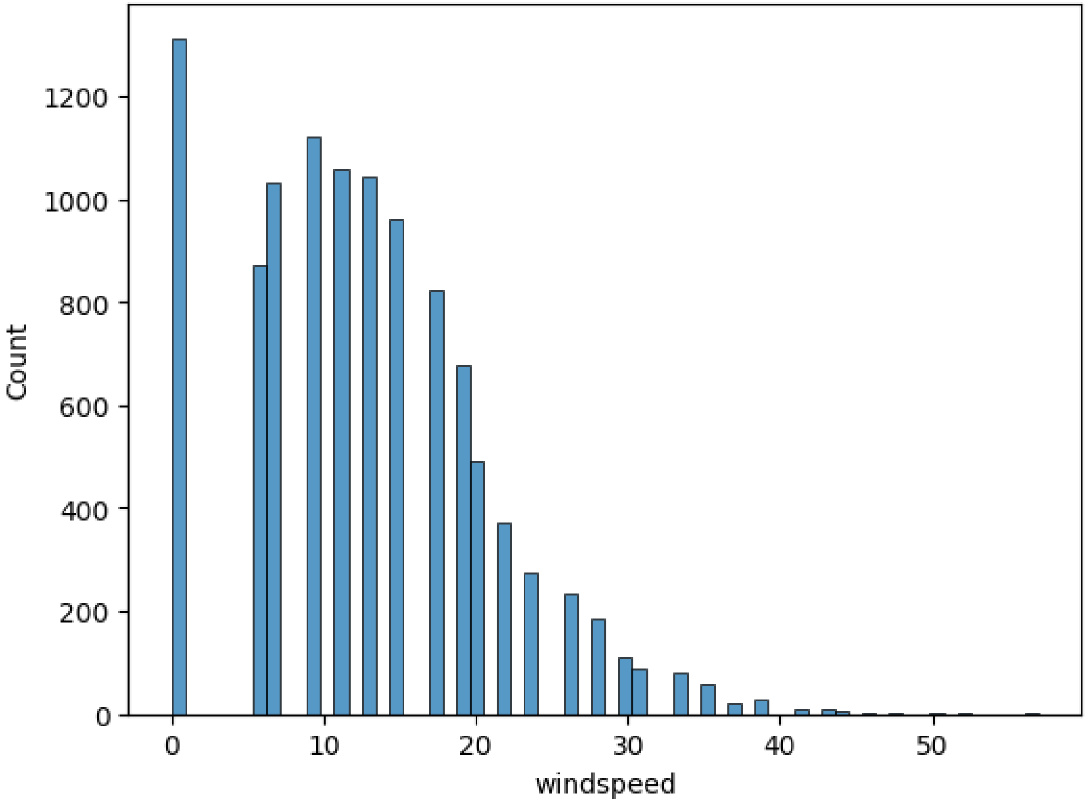
```
[15]: sns.histplot(data=df,x='temp')
      plt.plot()
```

```
[15]: []
```



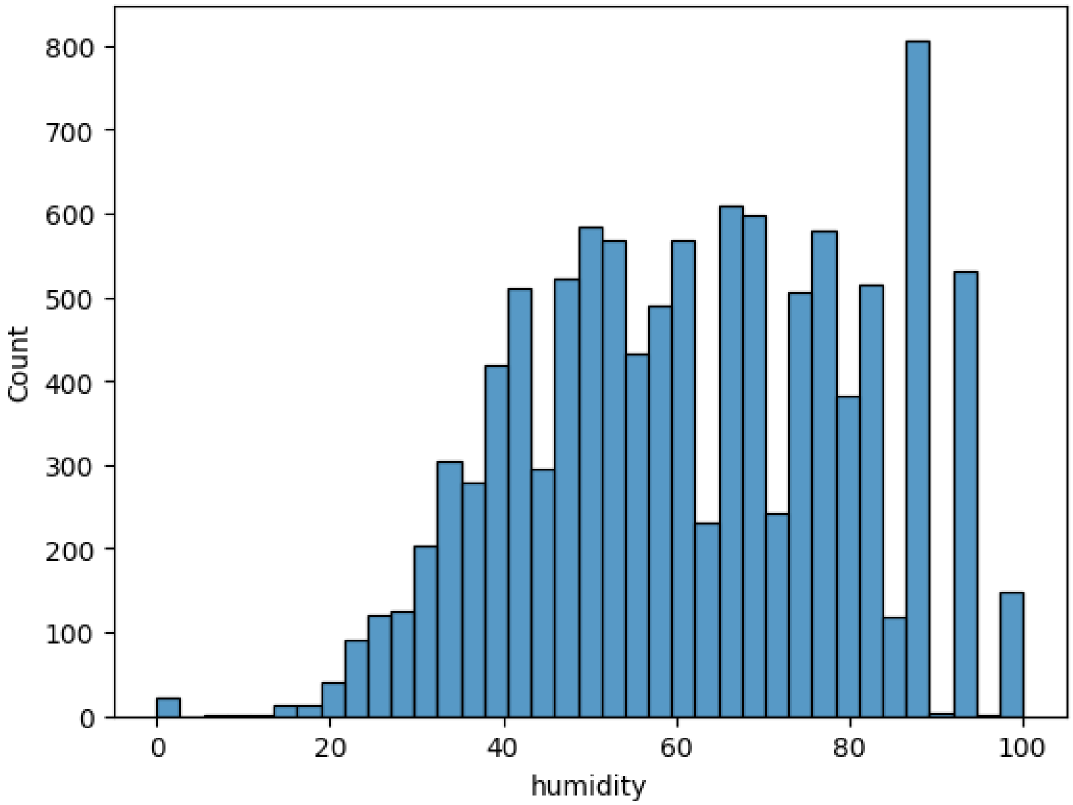
```
[16]: sns.histplot(data=df,x='windspeed')
      plt.plot()
```

```
[16]: []
```



[17]:
```
sns.histplot(data=df,x='humidity')
plt.plot()
```

[17]:
```
[]
```



[ ]:
```
df.isnull().sum()
```

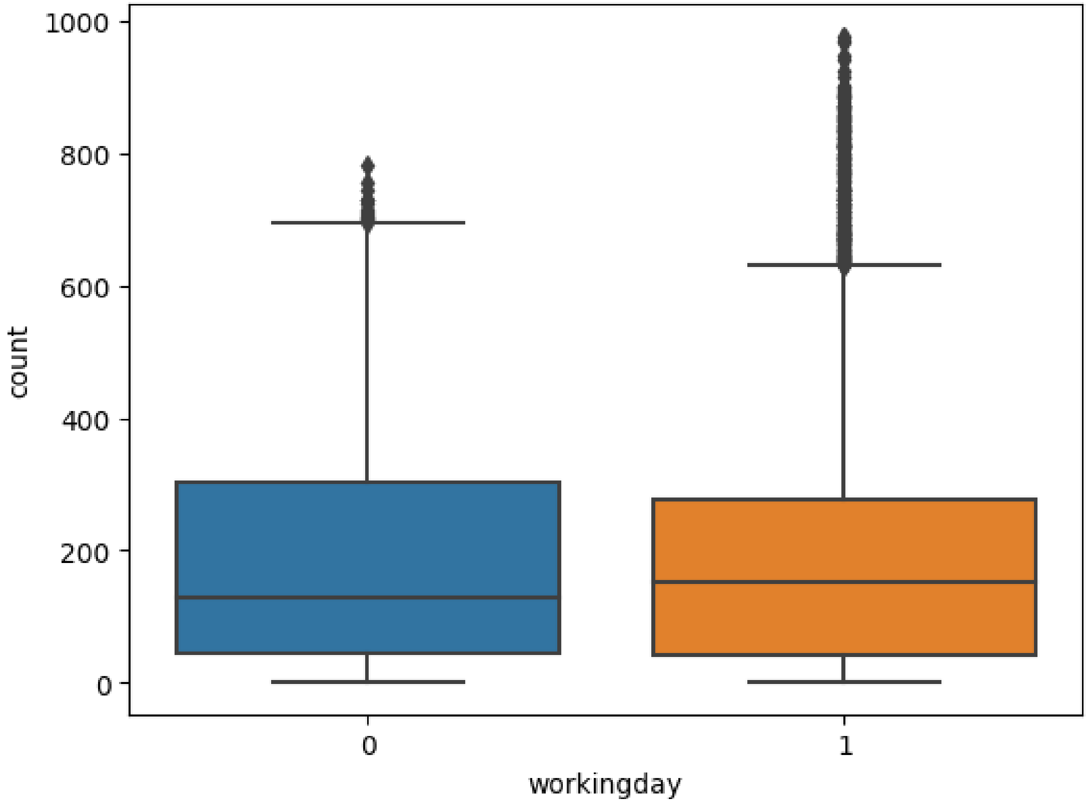
[ ]:
```
datetime      0
season        0
holiday       0
workingday    0
weather       0
temp          0
atemp         0
humidity      0
windspeed     0
casual        0
registered    0
count         0
dtype: int64
```

[ ]:
```
df['workingday'].unique()
```

[ ]:
```
array([0, 1])
```

[ ]:
```
sns.boxplot(x='workingday',y='count',data=df)
```

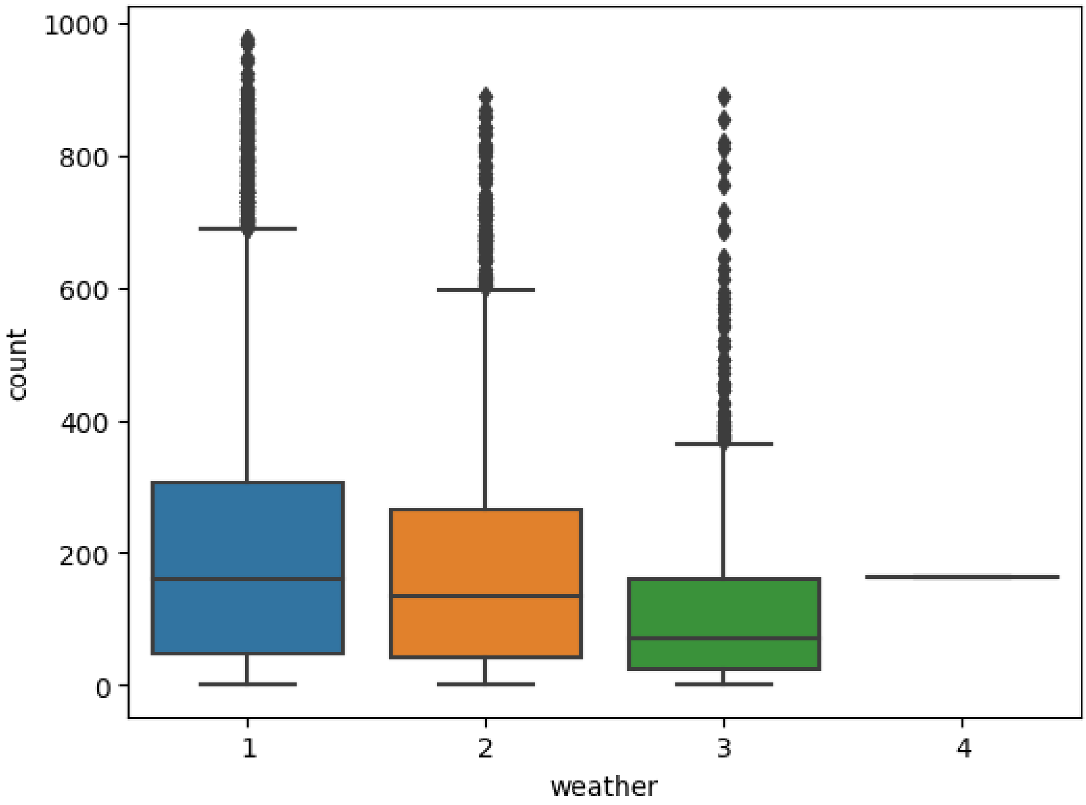
```
[ ]: <Axes: xlabel='workingday', ylabel='count'>
```



From the above data, it can be concluded that count does not have much dependence on working day.

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

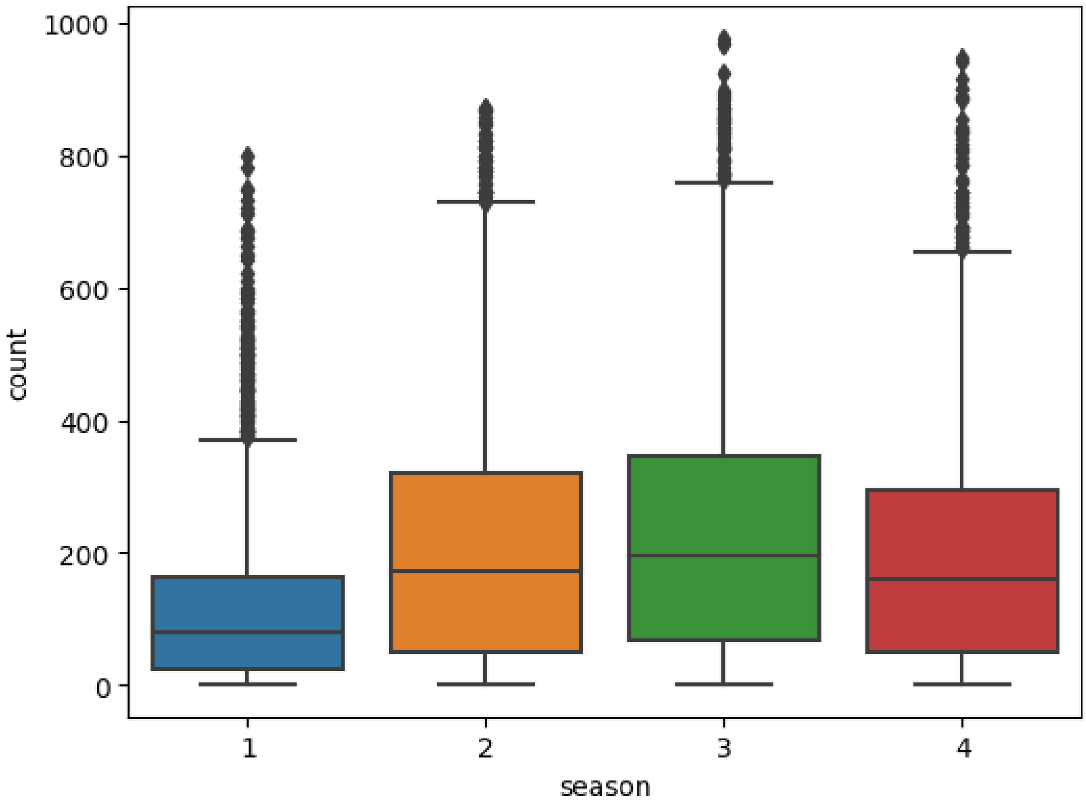
```
[ ]: <Axes: xlabel='weather', ylabel='count'>
```



From the above data, it can be concluded that in Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog, very less bikes are rented.

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

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



from the above data, it can be concluded that in summer anf fall season, more bikes are rented as compared to other remaining seasons.

##Hypothesis Testing

CASE-1: * H0=Working Day has no effect on number of electric cycles being rented

- Ha=Working Day has effect on number of electric cycles rented

we will use two sample T-test and will use significance value as 0.05

```
[ ]: df1= df[df['workingday']==0]['count'].values
     df2= df[df['workingday']==1]['count'].values
```

```
[ 16  40  32 … 106  89  33]
```

```
[ ]: np.var(df1)
```

```
[ ]: 30171.346098942427
```

```
[ ]: np.var(df2)
```

```
[ ]: 34040.69710674686
[ ]: np.var(df2)/np.var(df1)
```

```
[ ]: 1.1282458858519429
```

If the ratio of variance of larger data group to that of smaller data group is less than 4:1, then we consider both the data groups have equal variance.

```
[ ]: stats.ttest_ind(a=df1,b=df2,equal_var=True)
```

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

Since p_value is greater than 0.05, so we do not reject null hypothesis. So,we don't have enough evidence to say that working day has effect on number of electric cycles being rented.

CASE-2: * H0=No. of cycles rented is similar in different seasons

- Ha=No. of cycles rented is different in different seasons

we will use ANNOVA test and will use significance value as 0.05

```
[ ]: df1= df[df['season']==1]['count'].values
     df2= df[df['season']==2]['count'].values
     df3= df[df['season']==3]['count'].values
     df4= df[df['season']==4]['count'].values
```

```
[ ]: stats.f_oneway(df1,df2,df3,df4)
```

```
[ ]: F_onewayResult(statistic=236.94671081032106, pvalue=6.164843386499654e-149)
```

Since p_value is smaller than 0.05, so we reject null hypothesis. Hence, no. of cycles rented is different in different seasons

CASE-3:

H0=No. of cycles rented is similar in different weather

Ha=No. of cycles rented is different in different weather

we will use ANNOVA test and will use significance value as 0.05

```
[ ]: df1= df[df['weather']==1]['count'].values
     df2= df[df['weather']==2]['count'].values
     df3= df[df['weather']==3]['count'].values
     df4= df[df['weather']==4]['count'].values
```

```
[ ]: stats.f_oneway(df1,df2,df3,df4)
```

```
[ ]: F_onewayResult(statistic=65.53024112793271, pvalue=5.482069475935669e-42)
```

Since p_value is smaller than 0.05, so we reject null hypothesis. Hence, no. of cycles rented is different in different weather.

CASE-4:

H0=Weather is independent on season

Ha=Weather is not independent on season

we will use Chi-square test and will use significance value as 0.05

[5]:

```
data_table=pd.crosstab(df['season'],df['weather'])
data_table
```

[5]:

```
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
```

[7]:

```
value=stats.chi2_contingency(data_table)
expected_values=value[3]
expected_values
```

[7]:

```
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]])
```

[13]:

```
n_rows=4
n_columns=4
dof=(n_rows-1)*(n_columns-1)
print("degrees of freedom = ",dof)
alpha=0.05
print("alpha = ",alpha)

chi_square=sum([(o-e)**2/e for o, e in zip(data_table.values, expected_values)])
chi_square_statistic=chi_square[0]+chi_square[1]
print("chi_square test statistic = ",chi_square_statistic)

critical_value=stats.chi2.ppf(q=1-alpha,df=dof)
print("critical_value = ",critical_value)

p_value=1-stats.chi2.cdf(x=chi_square_statistic,df=dof)
print("p_value = ",p_value)

if p_value <= alpha:
  print("since p_value is less than alpha, we reject the null hypothesis means␣
 ↪weather is dependent on season")
else:
 print("since p_value is greater than alpha, we don't reject the null␣
↪hypothesis means weather is independent on season")
```

```
degrees of freedom =  9
alpha =  0.05
chi_square test statistic =  44.09441248632364
critical_value =  16.918977604620448
p_value =  1.3560001579371317e-06
since p_value is less than alpha, we reject the null hypothesis means weather is
dependent on season
```

##Inference

- Whenever there is a holiday, more bikes are rented at that time.
- In summer and fall season, more bikes are rented as compared to other seasons like rain, thunderstorm, snow or fog.
- It is also found that working day has no effect on number of bikes rented.