```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
data=pd.read_csv('https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/428/original/bike_sharing.csv?1642089089')
```

```
data
```

| | 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

```
data.head()
```

| | 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 | 0.0 | 3 | 13 | 16 |
| **1** | 2011-01-01 01:00:00 | 1 | 0 | 0 | 1 | 9.02 | 13.635 | 80 | 0.0 | 8 | 32 | 40 |
| **2** | 2011-01-01 02:00:00 | 1 | 0 | 0 | 1 | 9.02 | 13.635 | 80 | 0.0 | 5 | 27 | 32 |
| **3** | 2011-01-01 03:00:00 | 1 | 0 | 0 | 1 | 9.84 | 14.395 | 75 | 0.0 | 3 | 10 | 13 |
| **4** | 2011-01-01 04:00:00 | 1 | 0 | 0 | 1 | 9.84 | 14.395 | 75 | 0.0 | 0 | 1 | 1 |

```
# finding missing values
data.isnull().sum()
```

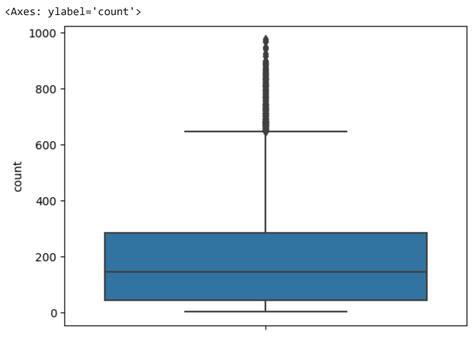
```
datetime      0
season        0
holiday       0
workingday    0
weather       0
temp          0
atemp         0
humidity      0
windspeed     0
casual        0
registered    0
count         0
dtype: int64
```

```
data.duplicated().any()
```

```
False
```

```
sns.boxplot(y=data['count']) # it is useful outliers detection
```

```
<Axes: ylabel='count'>
```




```
iqr=data['count'].quantile(0.75)-data['count'].quantile(0.25)
iqr
```

```
242.0
```

```
upper_limit=data['count'].quantile(0.75)+iqr
lower_limit=data['count'].quantile(0.25)-iqr
```

```
outliers=data[data['count']>upper_limit]
```

```
outliers
```

| | datetime | season | holiday | workingday | weather | temp | atemp | humidity | windspeed | casual | registered | count |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1819** | 2011-05-02 17:00:00 | 2 | 0 | 1 | 1 | 27.06 | 31.060 | 65 | 12.9980 | 65 | 472 | 537 |
| **1844** | 2011-05-03 18:00:00 | 2 | 0 | 1 | 1 | 28.70 | 32.575 | 48 | 27.9993 | 59 | 485 | 544 |
| **1891** | 2011-05-05 17:00:00 | 2 | 0 | 1 | 1 | 22.96 | 26.515 | 26 | 26.0027 | 66 | 467 | 533 |
| **1915** | 2011-05-06 17:00:00 | 2 | 0 | 1 | 1 | 23.78 | 27.275 | 40 | 23.9994 | 83 | 470 | 553 |
| **1987** | 2011-05-09 17:00:00 | 2 | 0 | 1 | 1 | 25.42 | 31.060 | 38 | 16.9979 | 59 | 539 | 598 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **10846** | 2012-12-18 08:00:00 | 4 | 0 | 1 | 1 | 15.58 | 19.695 | 94 | 0.0000 | 10 | 652 | 662 |
| **10855** | 2012-12-18 17:00:00 | 4 | 0 | 1 | 1 | 16.40 | 20.455 | 47 | 30.0026 | 39 | 533 | 572 |
| **10870** | 2012-12-19 08:00:00 | 4 | 0 | 1 | 1 | 9.84 | 12.880 | 87 | 7.0015 | 13 | 665 | 678 |
| **10879** | 2012-12-19 17:00:00 | 4 | 0 | 1 | 1 | 16.40 | 20.455 | 50 | 26.0027 | 26 | 536 | 562 |
| **10880** | 2012-12-19 18:00:00 | 4 | 0 | 1 | 1 | 15.58 | 19.695 | 50 | 23.9994 | 23 | 546 | 569 |

681 rows × 12 columns

```
data
```

| | 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

```
data['season'].unique()
data['season']=data['season'].map({1:'spring',2:'summer',3:'fall',4:'winter'})
#sunny
#cloudy
#mist
#rainy
data['weather']=data['weather'].map({1:'sunny',2:'cloudy',3:'mist',4:'rainy'})
data['holiday'].unique()
data['holiday']=data['holiday'].map({0:'No',1:'yes'})
data['workingday']=data['workingday'].map({0:'No',1:'yes'})
data
```

| | datetime | season | holiday | workingday | weather | temp | atemp | humidity | windspeed | casual | registered | count |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2011-01-01 00:00:00 | spring | No | No | sunny | 9.84 | 14.395 | 81 | 0.0000 | 3 | 13 | 16 |
| **1** | 2011-01-01 01:00:00 | spring | No | No | sunny | 9.02 | 13.635 | 80 | 0.0000 | 8 | 32 | 40 |
| **2** | 2011-01-01 02:00:00 | spring | No | No | sunny | 9.02 | 13.635 | 80 | 0.0000 | 5 | 27 | 32 |
| **3** | 2011-01-01 03:00:00 | spring | No | No | sunny | 9.84 | 14.395 | 75 | 0.0000 | 3 | 10 | 13 |
| **4** | 2011-01-01 04:00:00 | spring | No | No | sunny | 9.84 | 14.395 | 75 | 0.0000 | 0 | 1 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **10881** | 2012-12-19 19:00:00 | winter | No | yes | sunny | 15.58 | 19.695 | 50 | 26.0027 | 7 | 329 | 336 |
| **10882** | 2012-12-19 20:00:00 | winter | No | yes | sunny | 14.76 | 17.425 | 57 | 15.0013 | 10 | 231 | 241 |
| **10883** | 2012-12-19 21:00:00 | winter | No | yes | sunny | 13.94 | 15.910 | 61 | 15.0013 | 4 | 164 | 168 |
| **10884** | 2012-12-19 22:00:00 | winter | No | yes | sunny | 13.94 | 17.425 | 61 | 6.0032 | 12 | 117 | 129 |
| **10885** | 2012-12-19 23:00:00 | winter | No | yes | sunny | 13.12 | 16.665 | 66 | 8.9981 | 4 | 84 | 88 |

10886 rows × 12 columns

```
data
```

| | datetime | season | holiday | workingday | weather | temp | atemp | humidity | windspeed | casual | registered | count |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2011-01-01 00:00:00 | spring | No | No | sunny | 9.84 | 14.395 | 81 | 0.0000 | 3 | 13 | 16 |
| **1** | 2011-01-01 01:00:00 | spring | No | No | sunny | 9.02 | 13.635 | 80 | 0.0000 | 8 | 32 | 40 |
| **2** | 2011-01-01 02:00:00 | spring | No | No | sunny | 9.02 | 13.635 | 80 | 0.0000 | 5 | 27 | 32 |
| **3** | 2011-01-01 03:00:00 | spring | No | No | sunny | 9.84 | 14.395 | 75 | 0.0000 | 3 | 10 | 13 |
| **4** | 2011-01-01 04:00:00 | spring | No | No | sunny | 9.84 | 14.395 | 75 | 0.0000 | 0 | 1 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **10881** | 2012-12-19 19:00:00 | winter | No | yes | sunny | 15.58 | 19.695 | 50 | 26.0027 | 7 | 329 | 336 |
| **10882** | 2012-12-19 20:00:00 | winter | No | yes | sunny | 14.76 | 17.425 | 57 | 15.0013 | 10 | 231 | 241 |
| **10883** | 2012-12-19 21:00:00 | winter | No | yes | sunny | 13.94 | 15.910 | 61 | 15.0013 | 4 | 164 | 168 |
| **10884** | 2012-12-19 22:00:00 | winter | No | yes | sunny | 13.94 | 17.425 | 61 | 6.0032 | 12 | 117 | 129 |
| **10885** | 2012-12-19 23:00:00 | winter | No | yes | sunny | 13.12 | 16.665 | 66 | 8.9981 | 4 | 84 | 88 |

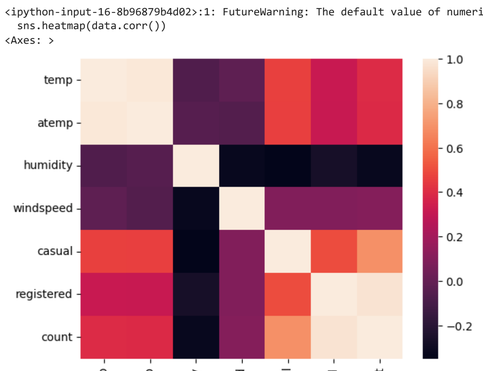
10886 rows × 12 columns

```
data
```

| | datetime | season | holiday | workingday | weather | temp | atemp | humidity | windspeed | casual | registered | count |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2011-01-01 00:00:00 | spring | No | No | sunny | 9.84 | 14.395 | 81 | 0.0000 | 3 | 13 | 16 |
| **1** | 2011-01-01 01:00:00 | spring | No | No | sunny | 9.02 | 13.635 | 80 | 0.0000 | 8 | 32 | 40 |
| **2** | 2011-01-01 02:00:00 | spring | No | No | sunny | 9.02 | 13.635 | 80 | 0.0000 | 5 | 27 | 32 |
| **3** | 2011-01-01 03:00:00 | spring | No | No | sunny | 9.84 | 14.395 | 75 | 0.0000 | 3 | 10 | 13 |
| **4** | 2011-01-01 04:00:00 | spring | No | No | sunny | 9.84 | 14.395 | 75 | 0.0000 | 0 | 1 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **10881** | 2012-12-19 19:00:00 | winter | No | yes | sunny | 15.58 | 19.695 | 50 | 26.0027 | 7 | 329 | 336 |
| **10882** | 2012-12-19 20:00:00 | winter | No | yes | sunny | 14.76 | 17.425 | 57 | 15.0013 | 10 | 231 | 241 |
| **10883** | 2012-12-19 21:00:00 | winter | No | yes | sunny | 13.94 | 15.910 | 61 | 15.0013 | 4 | 164 | 168 |
| **10884** | 2012-12-19 22:00:00 | winter | No | yes | sunny | 13.94 | 17.425 | 61 | 6.0032 | 12 | 117 | 129 |
| **10885** | 2012-12-19 23:00:00 | winter | No | yes | sunny | 13.12 | 16.665 | 66 | 8.9981 | 4 | 84 | 88 |

```
data['holiday'].value_counts()
sns.countplot(x='holiday',hue='weather',data=data)
# in sunny weather more holidays are  there compared to cloudy and mist weather
```

```
<Axes: xlabel='holiday', ylabel='count'>
```



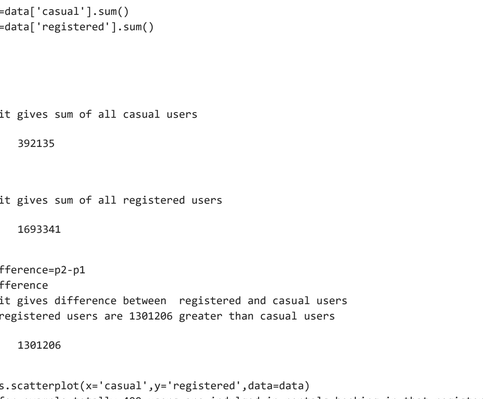

```
data.count
sns.countplot(x='workingday',hue='weather',data=data)
```

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


```
#sunny weather more workingdays are there
# but in mist less working days are there
```

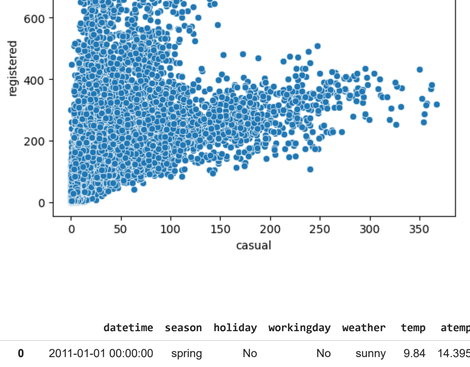
```
sns.heatmap(data.corr())
```

```
<ipython-input-16-8b96879b4d02>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  sns.heatmap(data.corr())
<Axes: >
```




```
data.head(2)
data
sns.barplot(x='season',y='count',data=data)
# in fall season more number of rentals are there compared to winter,suumer,spring,summer means more number of people are likely to take rentals in fall season
```

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




```
p1=data['casual'].sum()
p2=data['registered'].sum()
```

```
p1
# it gives sum of all casual users
```

```
392135
```

```
p2
# it gives sum of all registered users
```

```
1693341
```

```
difference=p2-p1
difference
# it gives difference between  registered and casual users
# registered users are 1301206 greater than casual users
```

```
1301206
```

```
sns.scatterplot(x='casual',y='registered',data=data)
# for example totally 400 users are indulged in rentals booking in that register type users are 170-180
```

```
<Axes: xlabel='casual', ylabel='registered'>
```




```
data
```

| | datetime | season | holiday | workingday | weather | temp | atemp | humidity | windspeed | casual | registered | count |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2011-01-01 00:00:00 | spring | No | No | sunny | 9.84 | 14.395 | 81 | 0.0000 | 3 | 13 | 16 |
| **1** | 2011-01-01 01:00:00 | spring | No | No | sunny | 9.02 | 13.635 | 80 | 0.0000 | 8 | 32 | 40 |
| **2** | 2011-01-01 02:00:00 | spring | No | No | sunny | 9.02 | 13.635 | 80 | 0.0000 | 5 | 27 | 32 |
| **3** | 2011-01-01 03:00:00 | spring | No | No | sunny | 9.84 | 14.395 | 75 | 0.0000 | 3 | 10 | 13 |
| **4** | 2011-01-01 04:00:00 | spring | No | No | sunny | 9.84 | 14.395 | 75 | 0.0000 | 0 | 1 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **10881** | 2012-12-19 19:00:00 | winter | No | yes | sunny | 15.58 | 19.695 | 50 | 26.0027 | 7 | 329 | 336 |
| **10882** | 2012-12-19 20:00:00 | winter | No | yes | sunny | 14.76 | 17.425 | 57 | 15.0013 | 10 | 231 | 241 |
| **10883** | 2012-12-19 21:00:00 | winter | No | yes | sunny | 13.94 | 15.910 | 61 | 15.0013 | 4 | 164 | 168 |
| **10884** | 2012-12-19 22:00:00 | winter | No | yes | sunny | 13.94 | 17.425 | 61 | 6.0032 | 12 | 117 | 129 |
| **10885** | 2012-12-19 23:00:00 | winter | No | yes | sunny | 13.12 | 16.665 | 66 | 8.9981 | 4 | 84 | 88 |

10886 rows × 12 columns

```
# hypothesis testing
# we are going to find out weather working day has an effect on number of working days effected
p=data[data['workingday']=='yes']['count'] # we should have to take two groups from based on working day count of people who are taking
q=data[data['workingday']=='No']['count']
p
q
```

```
0         16
1         40
2         32
3         13
4          1
        ...
10809    109
10810    122
10811    106
10812     89
10813     33
Name: count, Length: 3474, dtype: int64
```

```
from scipy.stats import ttest_ind
#h0:both the groups will have same mean values
#h1:both groups will have different mean values
```

```
l=ttest_ind(p,q, alternative='two-sided')
l
```

```
TtestResult(statistic=-1.0833361748914772, pvalue=0.27873443811874504, df=3473.0)
```

```
p=l[1] # i have taken significant level alpha as 5 percent
p
data['weather'].unique()
```

```
array(['sunny', 'cloudy', 'mist', 'rainy'], dtype=object)
```

```
if p>0.05:
  print("null hypothesis will be accepted")
else:
  print("null hypothesis will be rejected")
# null is accepted means both will have same mean values  if both of them are having same mean values
# then this working days and non working days will not effect the count of rentals of it
```

```
null hypothesis will be accepted
```

```
# for different weather categories the number of rentals are different
```

```
# anova test is useful for finding the difference in variance or spread of data among different groups if there is a considerable difference
# then null will be rejected hypothesis because null assumes that in each group the variance will be same means no of   rentals for each season will be same
# based on data only we should have to use anova or kruskal we have to  decide it based on if all categories are having
# same variance then  anova test is applicable otherwise if different variances are observed  for different categories then  kruskhal test will
#be appllicable
```

```
0        16
1        40
2        32
3        13
4         1
       ...
6780    549
6781    330
6782    223
6783    148
6784     54
Name: count, Length: 2686, dtype: int64
```

```
# levene test is useful for  finding equal variances are there or not for each and every category
# when variance is equal means spread of data will be same
# h0:null hypothesis assumes that all categories are having equal variances
#h1:alternative hypothesis is quite opposite to null hypothesis assumption such that all the categories will not have equal spread of data
from scipy.stats import levene
d,p8=levene(r,s,v,w)
if p8>0.05:
  print("null hypothesis will be accepted")
else:
  print("null  hypothesis will be rejected")
```

```
null  hypothesis will be rejected
```

```
# if null hypothesis is rejected means anova test will not be applicable krushkal test should be applied to it
from scipy.stats import kruskal
jl,p10=kruskal(r,s,v,w)
if p10>0.05:
  print("null hypothesis will be accepted ")
else:
  print("null hypothesis will be rejected")
```

```
null hypothesis will be rejected
```

```
# if null hypothesis is rejected means for different seasons  different means will be there it indicates that for different seasons
# the count of rentals will be different means in one season number of rentals will be more
# in another season it will be less
```

```
# for weather also  we are  going to find out for different weather conditions the number of rentals are same or not
data['weather'].unique()
y=data[data['weather']=='sunny']['count']
c=data[data['weather']=='cloudy']['count']
m=data[data['weather']=='mist']['count']
gh=data[data['weather']=='rainy']['count']
from scipy.stats import levene
#h0: null assumes equal variance
#h1:alternate opposes the null hypothesis
jk,p34=levene(y,c,m,gh)
if p34>0.05:
  print('null will be accepted ')
else:
  print("alternate hypothesis will be accepted ")
```

```
alternate hypothesis will be accepted
```

```
# null hypothesis is rejected so we shoul have to use kruskall test here
# because for anova test all categories should have equal variances
#if unequal variances are there means krushkall test should be used
from scipy.stats import kruskal

df,p45=kruskal(y,c,m,gh)
```

```
if p45>0.05:
  print('null hypothesis will be accepted')
else:
  print("null hypothesis will be rejected")
```

```
null hypothesis will be rejected
```

```
# for different weather conditions  count of rentals will vary


type(s.values)
```

```
numpy.ndarray
```

```
if p>0.05:
  print('both groups are having same mean values')
else:
  print('both groups are having different mean values')
```

```
both groups are having same mean values
```

```
# here p value is greater than significantlevel so both will have same mean values null hypothesis is failed to reject
# for each and every category of weather the count of rentals are same


#chi square
#(null hypothesis)h0:there is no relation between weather and season
#(alternative hypothesis)h1:there is a relation between weather and season
# assumptions regarding null hypothesis
# null hypothesis says that the  it is a default assumption that there is no relationship between two variables
# but there is a relationship between two variables here season  and weather based on season weather will change automatically
# here in this case if p value means probability that null hypothesis to be true
# if p value is greater than significant level then null will be accepted otherwise null will  be rejected


from scipy.stats import chi2_contingency
o=pd.crosstab(data['weather'],data['season'])
o
```

| season | fall | spring | summer | winter |
|---|---|---|---|---|
| **weather** | | | | |
| **cloudy** | 604 | 715 | 708 | 807 |
| **mist** | 199 | 211 | 224 | 225 |
| **rainy** | 0 | 1 | 0 | 0 |
| **sunny** | 1930 | 1759 | 1801 | 1702 |

```
stat,p,dof,expected=chi2_contingency(o)
print(stat,p,dof,expected)
```

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

```
p
alpha=0.05 # i have taken significant level  as 0.05
```

```
if p>alpha:
  print('there is no relation between weather and season')
else:
    print("there is a relation between weather and season")
```

```
there is a relation between weather and season
```

```
# here null hypothesis is rejected there is a  relationship between weather and season
```

```
# business insights
# in fall  season more number of rentals    and in spring season less no of rentals i have observed  so far
# in working days during sunny weather conditions more  no of rentals and less no of rentals in mist weather condition
# in nonworking days  the same thing repeated less number in mist and more  in sunny season are observed through visualizations
# the mean values of number of rentals for working and nonworking days both are  same
# so working days and nonworking days atre not showing effect on the number of rentals of it
# weather and seasons will show effect on the number of rentals
# weather and seasons are corelated to each other




#reccommendations
# a detailed analysis should be there why  in some weather conditions and seasons the count of rentals are decreasing
# the root cause analysis is required for it
# a new system is required in order to increase the rate of bookings where rebates in booking and offers will increase the count of it
# advertisements and referral programmes regarding yulu app will show drastic impact on bookings
#i will suggest yulu team to implement new thing that is whenever booking is completed with payment there should be an option to
# select the vehicle that is comfortable to the user and the yulu driver should pick the vehicle and deliver that vehicle to customer
# if the vehicle is not comfortable means there should be exchange of vehicle policy also should be there because of providing
# these features to the user which he will also feel comfortable to use the services of yulu
```