```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import levene
from scipy.stats import shapiro
from scipy.stats import ttest_ind
from scipy.stats import f_oneway
from scipy.stats import chi2_contingency
```

Problem statement

What the company wants to find out ?

```
# 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
```

Understanding from the problem statement.

```
# Because of the revenue dips,
# Yulu wants to know all the factors(variables) that are important for the demand of electric cycles in India.
# We will be analyzing all those factors.
# We will also see how effective will these factors be for the demand of electric cycles.
```

Reading the data file.

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

```
data_yulu
```

	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

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

shape function to get the shape(no. of rows and no.of columns) from the data.

```
data_yulu.shape
```

```
(10886, 12)
```

Reading the first 5 rows of the data

```
data_yulu.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

Reading the last 5 rows of the data

```
data_yulu.tail()
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
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

Reading all the columns of the data

```
data_yulu.columns
```

```
Index(['datetime', 'season', 'holiday', 'workingday', 'weather', 'temp',
       'atemp', 'humidity', 'windspeed', 'casual', 'registered', 'count'],
      dtype='object')
```

info of the data

```
data_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
```

From the above we can observe that there are no null values in the data.

And also the info() function gives the information on data types of all the columns.

Using describe function which gives the statistical summary of all the columns from data.

```
data_yulu.describe()
```

	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
count	10886.000000	10886.000000	10886.000000	10886.000000	10886.00000	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000
mean	2.506614	0.028569	0.680875	1.418427	20.23086	23.655084	61.886460	12.799395	36.021955	155.552177	191.574132
std	1.116174	0.166599	0.466159	0.633839	7.79159	8.474601	19.245033	8.164537	49.960477	151.039033	181.144454
min	1.000000	0.000000	0.000000	1.000000	0.82000	0.760000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	2.000000	0.000000	0.000000	1.000000	13.94000	16.665000	47.000000	7.001500	4.000000	36.000000	42.000000
50%	3.000000	0.000000	1.000000	1.000000	20.50000	24.240000	62.000000	12.998000	17.000000	118.000000	145.000000
75%	4.000000	0.000000	1.000000	2.000000	26.24000	31.060000	77.000000	16.997900	49.000000	222.000000	284.000000
max	4.000000	1.000000	1.000000	4.000000	41.00000	45.455000	100.000000	56.996900	367.000000	886.000000	977.000000

Q)Whether working day have any effect on the number of cycles rented(count).

```
data_yulu.head(2)
```

	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

Here working day is categorical(0 or 1) and count is the total count of cycles rented on particular day(numerical).

As we are comparing "categorical" (working day or non working day) based on the count of cycles rented we can use the "ttest"

Count of electric cycles rented Season wise

```
data_yulu["season"].value_counts()
```

```
4    2734
2    2733
3    2733
1    2686
Name: season, dtype: int64
```

From the above with the help of value_counts() function we can get the count of electric cycles rented over the 4 seasons.

season 4 (winter) have the maximum count followed by season 2 (summer) and season 3 (fall)

season 1 (spring) have the lowest count of cycles rented.

```
# observing the season wise distribution in percentage
```

```
data_yulu["season"].value_counts(normalize = True) * 100
```

```
4    25.114826
2    25.105640
3    25.105640
1    24.673893
Name: season, dtype: float64
```

Count of electric cycles in different weathers.

```
data_yulu.head(2)
```

	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

```
data_yulu["weather"].value_counts()
```

```
1    7192
2    2834
3     859
4       1
Name: weather, dtype: int64
```

```
# 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
```

The highest rentals are when the weather category is 1 (Clear, Few clouds, partly cloudy, partly cloudy) followed by weather categor 2 and 3.

But there were almost no rentals when the weather category is 4 (Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog)

```
# Observing the weather wise distribution in the percentage format.
```

```
data_yulu["weather"].value_counts(normalize = True) * 100
```

```
1    66.066507
2    26.033437
3     7.890869
4     0.009186
Name: weather, dtype: float64
```

Count of cycles rented according to working day or non working day

```
data_yulu["workingday"].value_counts()
```

```
1    7412
0    3474
Name: workingday, dtype: int64
```

The count of electric cycles rented were high on the working day compared to the non working day. The reason might be lot of working class people might be using the cycles to commute to work.

```
# percentage distribution
```

```
data_yulu["workingday"].value_counts(normalize = True) * 100
```

```
1    68.087452
0    31.912548
Name: workingday, dtype: float64
```

Cleaning the data by removing the outliers from the data.

Applying the boxplot with respect to working day,season and weather and the count of cycles rented.

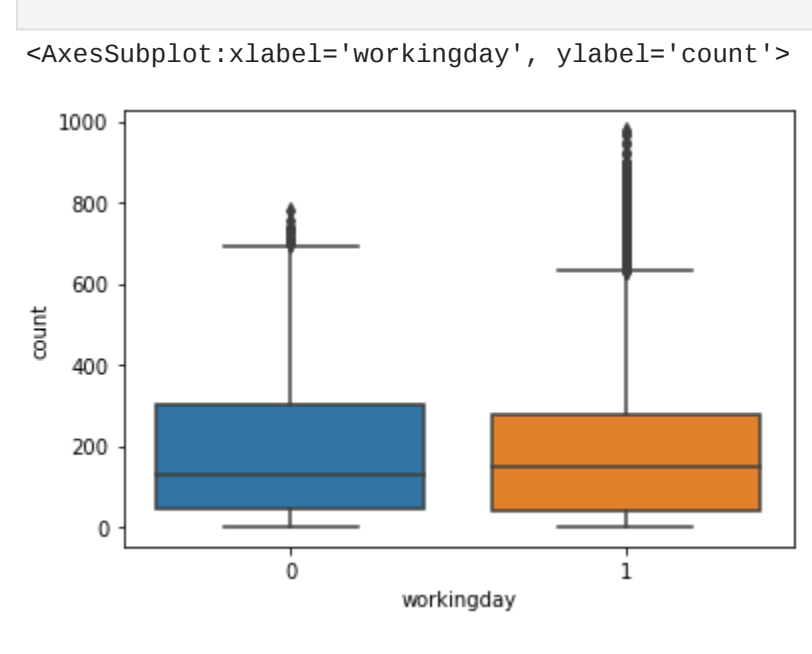
```
data_yulu.head(2)
```

	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

workingday

```
sns.boxplot(x = "workingday" , y = "count" , data = data_yulu)
```

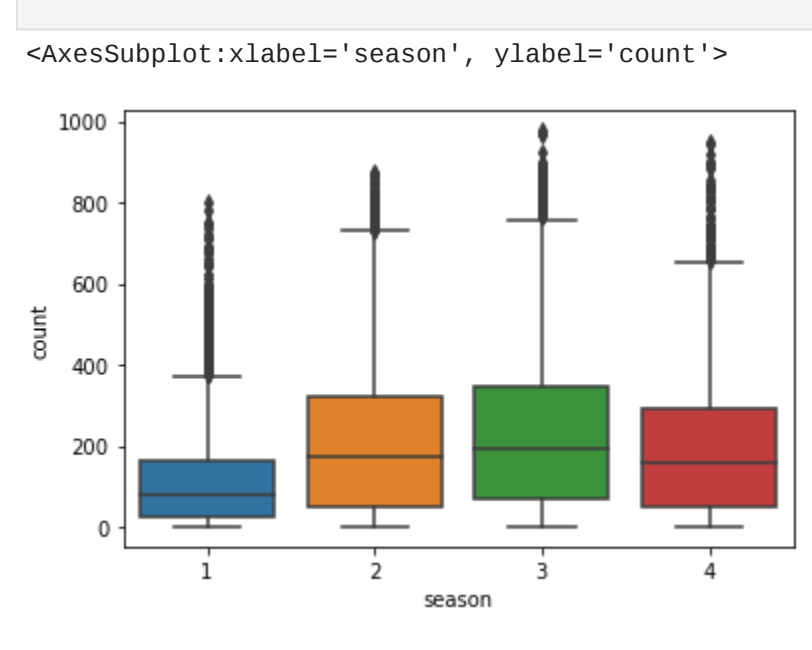
```
<AxesSubplot:xlabel='workingday', ylabel='count'>
```



season

```
sns.boxplot(x = "season" , y = "count" , data = data_yulu)
```

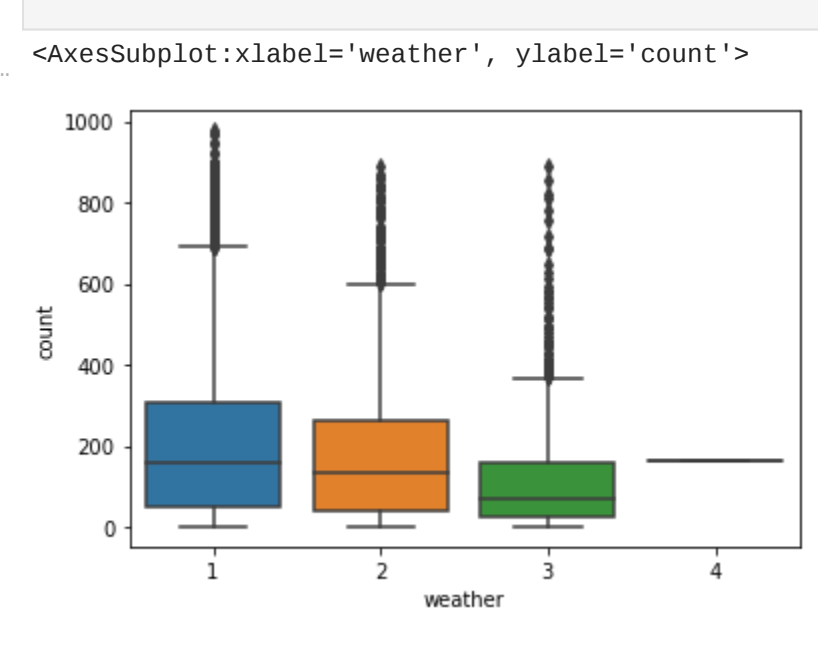
```
<AxesSubplot:xlabel='season', ylabel='count'>
```



weather

```
sns.boxplot(x = "weather" , y = "count" , data = data_yulu)
```

```
<AxesSubplot:xlabel='weather', ylabel='count'>
```



As we cannot interpret the effect of various factors (weather, workingday, season) with visual analysis, continue further using the hypothesis testing.

As the sample size is more than 30 and the standard deviation of the data is not given we use the ttest

1)working day has an effect on number of electric cycles rented.

null_hypothesis = count of rentals on weekday is equal to rentals on weekeneds alternate_hypothesis = count of rentals on weekday is greater than the rentals on weekend

```
data_yulu.head(2)
```

	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

```
workingday_rentals = data_yulu[data_yulu["workingday"] == 1]["count"].sample(3000)
nonworkingday_rentals = data_yulu[data_yulu["workingday"] == 1]["count"].sample(3000)
```

```
print(workingday_rentals)
print(nonworkingday_rentals)
```

```
3945     121
10882    241
4129       3
7262      95
5692     508
        ...
3245       6
2989     165
2947     514
4919      10
7080     330
Name: count, Length: 3000, dtype: int64
3928    599
595      25
6086    199
9345    977
5274    161
       ...
651     122
7873    249
8627    513
2019     20
474      75
Name: count, Length: 3000, dtype: int64
```

```
data_yulu["workingday"].value_counts()
```

```
1    7412
0    3474
Name: workingday, dtype: int64
```

```
# Calculating the p-value
```

```
stats , p_value = ttest_ind(workingday_rentals , nonworkingday_rentals , equal_var = False , alternative = "greater")
```

```
stats
```

```
0.2908063526234004
```

```
p_value
```

```
0.3856047409875341
```

-----As the p-value is higher we accept the null hypothesis(fail to reject the null hypothesis)

-----We can conclude that the count of electric cyles rented on weekdays is equal to the count on weekends.

-----Therefore the workingday have no effect on the number of electric cycles rented.

2) No. of cycles rented similar or different in different seasons.

```
data_yulu["season"].value_counts()
```

```
4    2734
2    2733
3    2733
1    2686
Name: season, dtype: int64
```

As we have to use the same sample size and 2500 would be the appropriate sample size as we can extract the sample number from all four season conditions.

```
season_1 = data_yulu[data_yulu["season"] == 1]["count"].sample(2500)
season_2 = data_yulu[data_yulu["season"] == 2]["count"].sample(2500)
season_3 = data_yulu[data_yulu["season"] == 3]["count"].sample(2500)
season_4 = data_yulu[data_yulu["season"] == 4]["count"].sample(2500)
```

```
# Null_hyp = count of electric cycle rentals are similar on all seasons
# alternate_hyp = count of electric cycle rentals are not similar in different seasons
# Here using Anova as we have more than 2 factors to compare with.
```

```
stats , p_value = f_oneway(season_1 , season_2 , season_3 , season_4)
```

```
p_value
```

```
3.940680121833304e-138
```

As the p-value is very (less than the alpha(0.05)) we reject the null hypothesis.

we conclude that count of electic cycle rentals are not same in different seasons.

3)No.of cycles rented similar or different in different weathers.

```
data_yulu["weather"].value_counts()
```

```
1    7192
2    2834
3     859
4       1
Name: weather, dtype: int64
```

As the weather type have only one rental we can exclude the weather type4.

As we have to use the same sample size and 850 would be the appropriate sample size as we have can extract the sample number from all three weather conditions.

creating samples

```
weather_1 = data_yulu[data_yulu["weather"] == 1]["count"].sample(850)
weather_2 = data_yulu[data_yulu["weather"] == 2]["count"].sample(850)
weather_3 = data_yulu[data_yulu["weather"] == 3]["count"].sample(850)
```

```
# Null_hyp = count of electric cycle rentals are similar on all weathers
# alternate_hyp = count of electric cycle rentals are not similar in different weathers
# Here using Anova as we have more than 2 factors to compare with.
```

```
stats , p_value = f_oneway(weather_1,weather_2,weather_3)
```

```
p_value
```

```
5.689563381955903e-24
```

As the p-value is very (less than the alpha(0.05)) we reject the null hypothesis.

we conclude that count of electic cycle rentals are not same in different weathers.

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

```
# null_hypothesis = weather is dependent on season
# alternate_hypothesis = weather is not dependent on season
```

```
stat , p_value , dof , expected = chi2_contingency(data_yulu["weather"] , data_yulu["season"])
```

```
p_value
```

```
1.0
```

As the p-value is very high (greater than alpha(0.05)) we fail to reject the null hypothesis.

we conclude that weather is actually not dependent on season in this case.