YULU BUSINESS CASE STUDY

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.

Dataset:

Dataset Link: yulu_data.csv

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

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)
- working day: 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

EXPLORATORY ANALYSIS

BASIC ANALYSIS ABOUT THE DATASET

Importing the required libraries and modules to perform analysis

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

```
df = pd.read_csv("yulu.txt")
df.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

```
df.shape
```

```
(10886, 12)
```

```
[ ] 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
```

INFERENCE:

- above code shows there are 10886 rows and 12 columns.
- Also, the data type of each column categorical and numerical columns can be identified by data types.
- All the columns are non –null … means there are no null values
- From the code we can also infer that which columns have categorical values and which has continuous values.

TO CHECK IF ANY NULL VALUES IN DATA SET

```
df.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
dtype: int64
```

There are no null values in the dataset.

ANALYSIS USING DESCRIBE FUNCTION

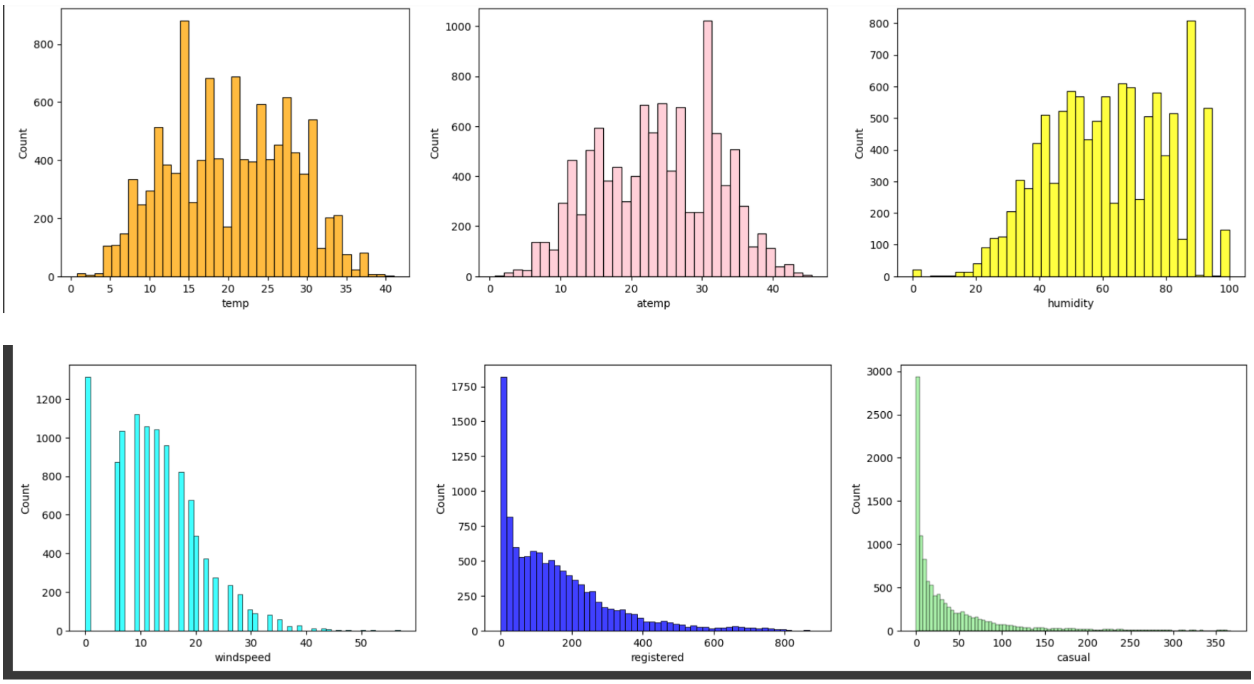
```
df.describe()
```

	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual
count	10886.000000	10886.000000	10886.000000	10886.000000	10886.00000	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
std	1.116174	0.166599	0.466159	0.633839	7.79159	8.474601	19.245033	8.164537	49.960477
min	1.000000	0.000000	0.000000	1.000000	0.82000	0.760000	0.000000	0.000000	0.000000
25%	2.000000	0.000000	0.000000	1.000000	13.94000	16.665000	47.000000	7.001500	4.000000
50%	3.000000	0.000000	1.000000	1.000000	20.50000	24.240000	62.000000	12.998000	17.000000
75%	4.000000	0.000000	1.000000	2.000000	26.24000	31.060000	77.000000	16.997900	49.000000
max	4.000000	1.000000	1.000000	4.000000	41.00000	45.455000	100.000000	56.996900	367.000000

INFERENCE: Inference using the above code that there are few outliers present in casual and registered categories.

UNIVARIATE ANALYSIS

```
fig, axis = plt.subplots(nrows = 2,ncols = 3,figsize=(20, 10))
sns.histplot(x = df["temp"], ax=axis[0,0], color = "orange")
sns.histplot(x = df["atemp"], ax=axis[0,1], color = "pink")
sns.histplot(x = df["humidity"], ax=axis[0,2], color = "yellow")
sns.histplot(x = df["windspeed"], ax=axis[1,0], color = "aqua")
sns.histplot(x = df["registered"], ax=axis[1,1], color = "blue")
sns.histplot(x = df["casual"], ax=axis[1,2], color = "lightgreen")
```



INFERENCE:

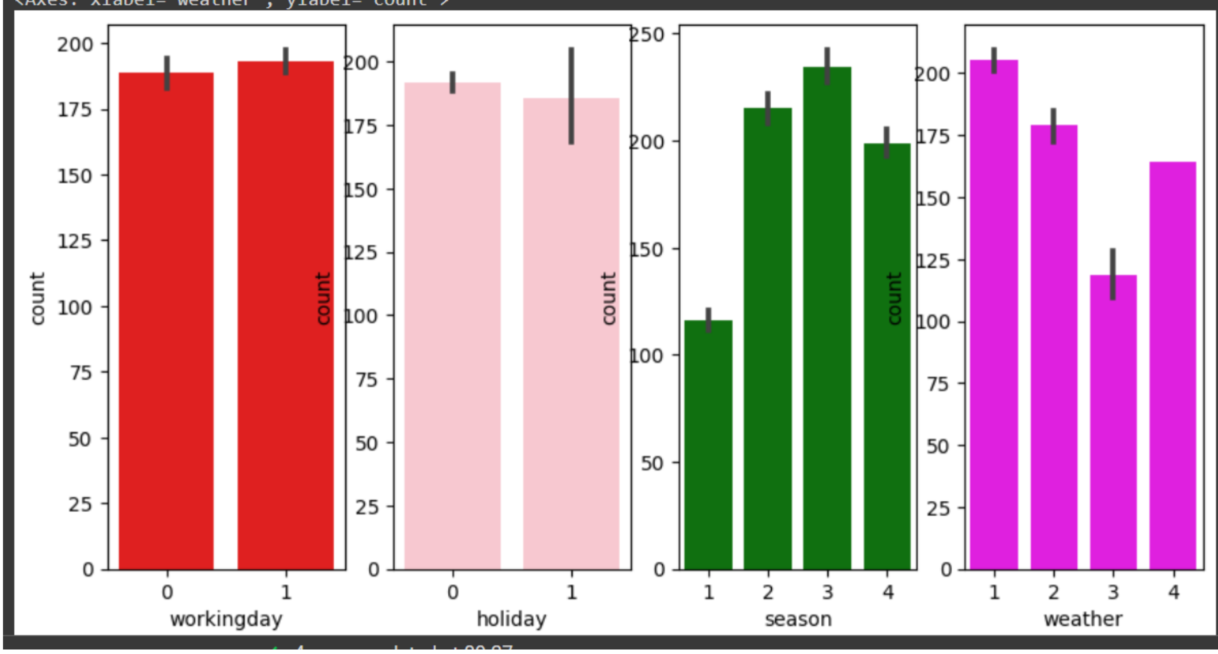
- From the above plots of various categories.. It is inferred that for temperatue: 14*c is the maximum temperature when bikes were rented.
- Whereas the maximum feeling temperature was 31*c.
- Bikes were purchased when the maximum humidity was approx. 90.
- Also when there was no wind speed. Most bikes were rented and renting significantly decreased as the wind speed increased .As , the bikers were well aware of the fact that it’s not safe to ride bikes during windy day.

BIVARIATE ANALYSIS

BIVARIATE ANALYSIS

```
fig, axis = plt.subplots(ncols = 4,figsize=(10, 5))
sns.barplot(x = df["workingday"],y = df["count"], ax=axis[0], color = "red")
sns.barplot(x = df["holiday"],y = df["count"],ax=axis[1],color = "pink")
sns.barplot(x = df["season"],y = df["count"],ax=axis[2], color = "green")
sns.barplot(x = df["weather"],y = df["count"],ax=axis[3],color = "magenta")
```

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



INFERENCE:

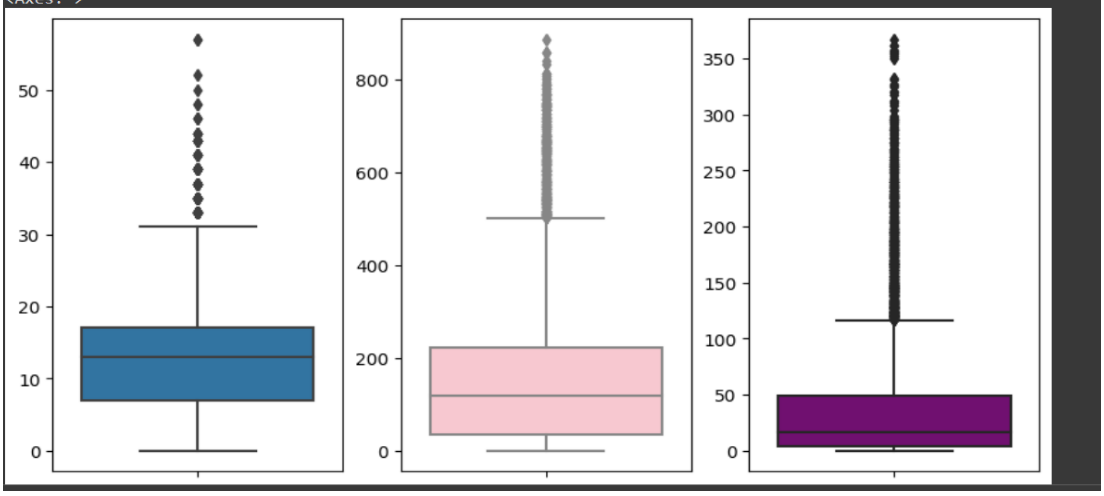
- Performing the analysis for all categories with count category.
- Workingday has no effect on count of bikes rented. As both the nonworking day as well as working day has almost same count.
- People tend to rent slightly more bikes on holidays.
- Most bikes were rented in fall season. Least bikes were rented during spring.
- Most bikes were rented when the weather was Clear, Few clouds, partly cloudy, partly cloudy.

- Least bikes were rented when there was Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds.

OUTLIERS DETECTION

```
fig, axis = plt.subplots(ncols = 3,figsize=(10, 5))
sns.boxplot(df["windspeed"],ax = axis[0])
sns.boxplot(df["registered"],ax = axis[1], color = "pink")
sns.boxplot(df["casual"],ax = axis[2], color = "purple")
```

<Axes: >



There are many outliers in the above categories

HYPOTHESIS TESTING

To check whether Working Day has effect on number of electric cycles rented.

- A sample of 500 records is taken from the population for better analysis.
- **Significance level is set to 0.05(95% confidence interval) for entire case study .**
- We have to perform analysis on working day and count of cycles. Hence data frame is filtered accordingly as a and b

```
[ ] # setting significance level (alpha) as 0.05

[ ] sample_df = df.sample(500)

[ ] new = sample_df[sample_df["workingday"]==0]
    a = new["count"]

[ ]

    new1 = sample_df[sample_df["workingday"]==1]
    b = new1["count"]
```

+ Code + Text

Null hypothesis (Ho): Working day has no effect on bike rentals

Alternate hypothesis (Ha): Working day has an effect on bike rentals

Test to be performed: ttest_ind : As two sample are given and we are comparing the means of two independant samples.

```
# H0: working day has no effect in number of vehicles rented
# Ha : working day has an effect in number of vehicles rented
tstatistic,pvalue = scipy.stats.ttest_ind(a,b)
print("tstatistic: ",tstatistic)
print("p value: ",pvalue)
alpha = 0.05
if pvalue < 0.05:
  print("reject ho : working day has an effect in number of vehicles rented")
else:
  print("failed to reject ho : working day has no effect in number of vehicles rented")
```

```
tstatistic:  -0.21011638809250208
p value:  0.8336627707965031
failed to reject ho : working day has no effect in number of vehicles rented
```

INFERENCE:

Here significance level is set to 005

Tstatistic value : -0.210116

The t-statistic is used to measure the difference between the means of the two samples The t-statistic is -0.210116, which means that the mean of the first sample is -0.210116 lower than the mean of the second sample.

Pvalue : 0.833662

Final result: failed to reject null hypothesis. As p value > significance level.

Final verdict: Working day has no effect in number of vehicles rented.

To check whether No. of cycles rented similar or different in different seasons.

```
[ ] sample_df["season"].unique()   #total seasons

array([2, 1, 3, 4])

+ Code   + Text

[ ] season1 = sample_df[sample_df["season"]==1]
    season2 = sample_df[sample_df["season"]==2]
    season3 = sample_df[sample_df["season"]==3]
    season4 = sample_df[sample_df["season"]==4]
```

INFERENCE:

Total number of unique seasons in dataset are 4.

Sample of each season taken respectively.

Null Hypothesis: No. of cycles rented are same in different seasons

Alternate Hypothesis: No. of cycles rented are different in different seasons.

Test to be performed : Here as samples are more than two then ANOVA test is performed.

```
# Ho: number of cylces are same in diff seasons
# ha: number of cycles are different in different sesons

f_statistic,pvalue = f_oneway(season1["count"],season2["count"],season3["count"],season4["count"])
print("f_statistic: ", f_statistic)
print("p_value:",pvalue)
alpha = 0.05
if pvalue <0.05:
  print("reject ho : number of cycles are different in different sesons")
else:
  print("failed to reject ho :number of cycles are same in different sesons")
```

```
f_statistic:  15.647588662195693
p_value: 9.730460125936882e-10
reject ho : number of cycles are different in different sesons
```

INFERENCE:

Significance level taken is 0.05

Value of f-statistic = 15.647588 . that is the difference between the means of the groups.

Value of p-value = 9.7304 x e^{-10} the probability of getting the observed results if the null hypothesis is true.

Final result : The pvalue is way too less than alpha hence we reject the null hypothesis.

Final verdict : Number of cycles rented are different in different seasons.

To check No. of cycles rented similar or different in different weather.

```
sample_df["weather"].unique()  #total weathers
```

```
array([1, 2, 3])
```

```
weather1= sample_df[sample_df["weather"]==1]
weather2 = sample_df[sample_df["weather"]==2]
weather3 = sample_df[sample_df["weather"]==3]
```

Total unique weather 3:

Sample of each season taken respectively.

Null Hypothesis: No. of cycles rented are same in different weather

Alternate Hypothesis: No. of cycles rented are different in different weather.

Test to be performed : Here as samples are more than two then ANOVA test is performed.

```
[ ] # Ho: number of cylces are same in diff weathers
    # ha: number of cycles are different in different weather

    f_stat,pvalue = f_oneway(weather1["count"],weather2["count"],weather3["count"])
    print("p_value:",pvalue)
    alpha = 0.05
    if pvalue <0.05:
      print("reject ho : number of cycles are different in different weather")
    else:
      print("failed to reject ho :number of cycles are same in different weather")

    p_value: 0.021078554174076246
    reject ho : number of cycles are different in different weather
```

INFERENCE:

Significance level(alpha) taken is 0.05

Value of p-value = 9.7304 x e^{-10} the probability of getting the observed results if the null hypothesis is true.

Final result : The pvalue is way too less than alpha hence we reject the null hypothesis.

Final verdict : Number of cycles rented are different in different seasons.

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

```
contingency_table = pd.crosstab(sample_df["weather"], sample_df["season"])
contingency_table
```

season	1	2	3	4
weather				
1	98	94	77	61
2	33	34	21	43
3	8	11	6	14

To check the dependency . A crosstab was created between count of vehicles in weather and season columns.

Null Hypothesis: weather is not dependent on season

Alternate Hypothesis: weather is dependent on season

Test to be performed : As the analysis to be performed between two categorical values hence Chi square test of independence to be performed.

```
# Ho: weather is not dependent on season
# Ha: weather is dependent on season
chi2_contingency(contingency_table)
print("chi_Statistics:" ,statistic)
print("p value: ",pvalue)
if 0.016112623198587965 < 0.05:
  print("reject ho : weather is dependent on season")
else:
  print("failed to reject ho : weather is not dependent on season")
```

```
chi_Statistics: -0.21011638809250208
p value:  9.730460125936882e-10
reject ho : weather is dependent on season
```

INFERENCE:

Chi_stat value = -0.210116 . That is the difference between the means of the groups.

P value : 9.7304 * e^{-10} .

Final result: As the value of p is way too less than alpha(0.05). Hence we reject the null hypothesis.

Final Verdict: Weather conditions are dependent on season.

INSIGHTS

- From the complete analysis following insights can be drawn.
- People rent almost same number of bikes on working or non working day.
- People tend to rent more bikes on holidays.
- People rent most bikes during Fall season and least during spring. Hence Yulu can provide offers during this season to attract more customers. Reduced prices can help.
- Monthly travel pass scheme can be initiated to get fixed amount of customers.
- People tend to rent less bikes during windy day or when the weather conditions are bad. Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
- For this Yulu can manufacture bikes with roof and windshield and wipers.So that people can commute with ease even when weather is bad or windy.
- Various campaigns , racing competitions can be held for promotion.

```
df["registered"].nunique()
```

```
731
```

```
df["casual"].nunique()
```

```
309
```

- There are 731 unique registered customers and 309 casual customers.
- Hence to convert casual customers to registered customers .Offers like free extra hour of rentals if registered can be initiated.

- Reference benefits can be given to registered customers. If they help in converting casual customers to registered ones.
- Home pick up and drop off of vehicles can be done.
- Business should also be focussed in tier2,tier3 cities.

COLAB LINK 

https://colab.research.google.com/drive/1H-SYbZdVU69E18EatJsHFJt3xtaTozxl?usp=sharing