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.

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

```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objs as go
from scipy import stats
import plotly.subplots as sp
from scipy.stats import norm
```

```
import numpy as np, pandas as pd
import matplotlib.pyplot as plt
import matplotlib_inline
matplotlib_inline.backend_inline.set_matplotlib_formats('svg')

from IPython.display import display, HTML
display(HTML("<style>.container { width:100% !important; }</style>"))
pd.set_option('display.max_columns', None)
# pd.set_option('display.max_rows', None)
```

```
df=pd.read_csv(r'/content/drive/MyDrive/yulu.csv')
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
df
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	01-01-2011 00:00	1	0	0	1	9.84	14.395	81	0.0000	3	13	16
1	01-01-2011 01:00	1	0	0	1	9.02	13.635	80	0.0000	8	32	40
2	01-01-2011 02:00	1	0	0	1	9.02	13.635	80	0.0000	5	27	32
3	01-01-2011 03:00	1	0	0	1	9.84	14.395	75	0.0000	3	10	13
4	01-01-2011 04:00	1	0	0	1	9.84	14.395	75	0.0000	0	1	1

```
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.shape
```

```
(10886, 12)
```

```
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
```

There are no missing values in the dataset.

```
status_flag=['season','holiday','workingday','weather']
```

```
for i in status_flag:
    print(df[i].value_counts())
    print('')
```

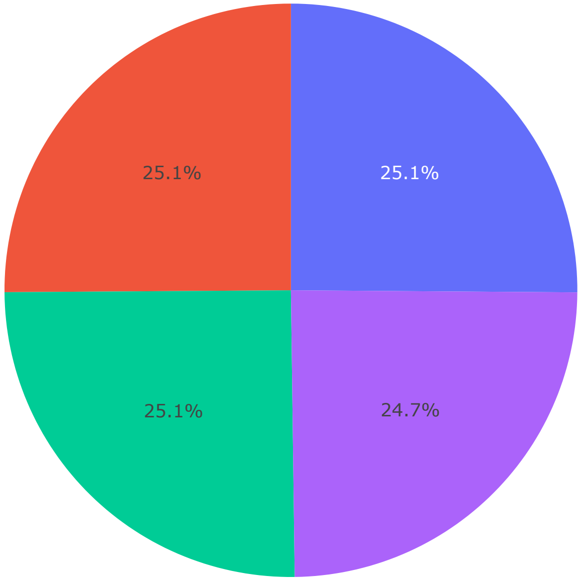
```
4    2734
2    2733
3    2733
1    2686
Name: season, dtype: int64

0    10575
1      311
Name: holiday, dtype: int64

1    7412
0    3474
Name: workingday, dtype: int64

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

```
for i in status_flag:
    print(round((df[i].value_counts(normalize=True)*100),2))
    print('')
```

```
4    25.11
2    25.11
3    25.11
1    24.67
Name: season, dtype: float64

0    97.14
1     2.86
Name: holiday, dtype: float64

1    68.09
0    31.91
Name: workingday, dtype: float64
```

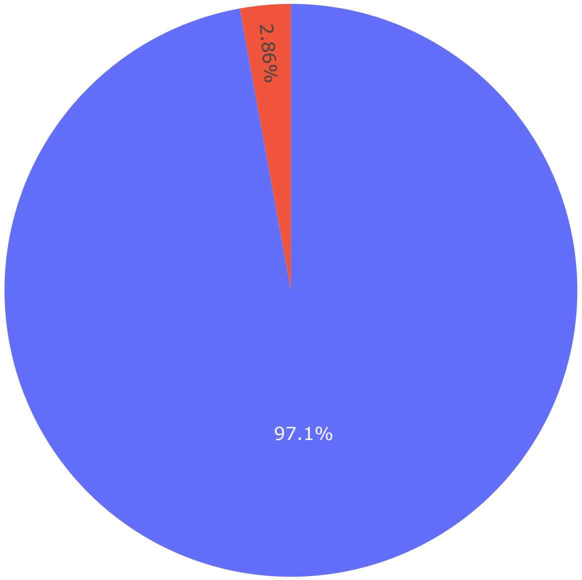
```
1    66.07
2    26.03
3     7.89
4     0.01
Name: weather, dtype: float64
```

```
import plotly.express as px
px.pie(df, names="season")
```



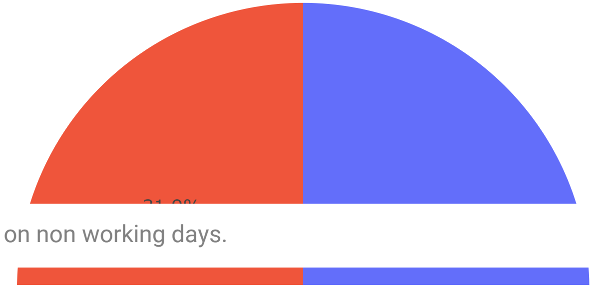
This shows that the number of booking is almost smiliar in different season

```
import plotly.express as px
px.pie(df, names="holiday")
```



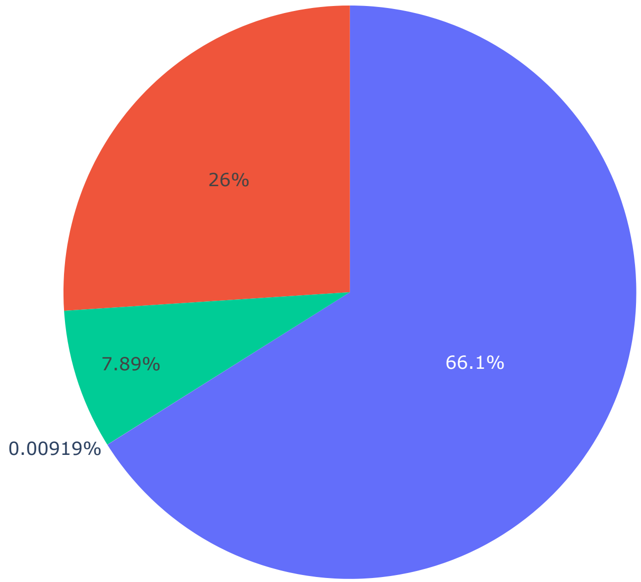
97% booking are during non-holidays

```
import plotly.express as px
px.pie(df, names="workingday")
```



Around 70% of the rides are booked on working days while 30% are booked on non working days.

```
import plotly.express as px
px.pie(df, names="weather")
```

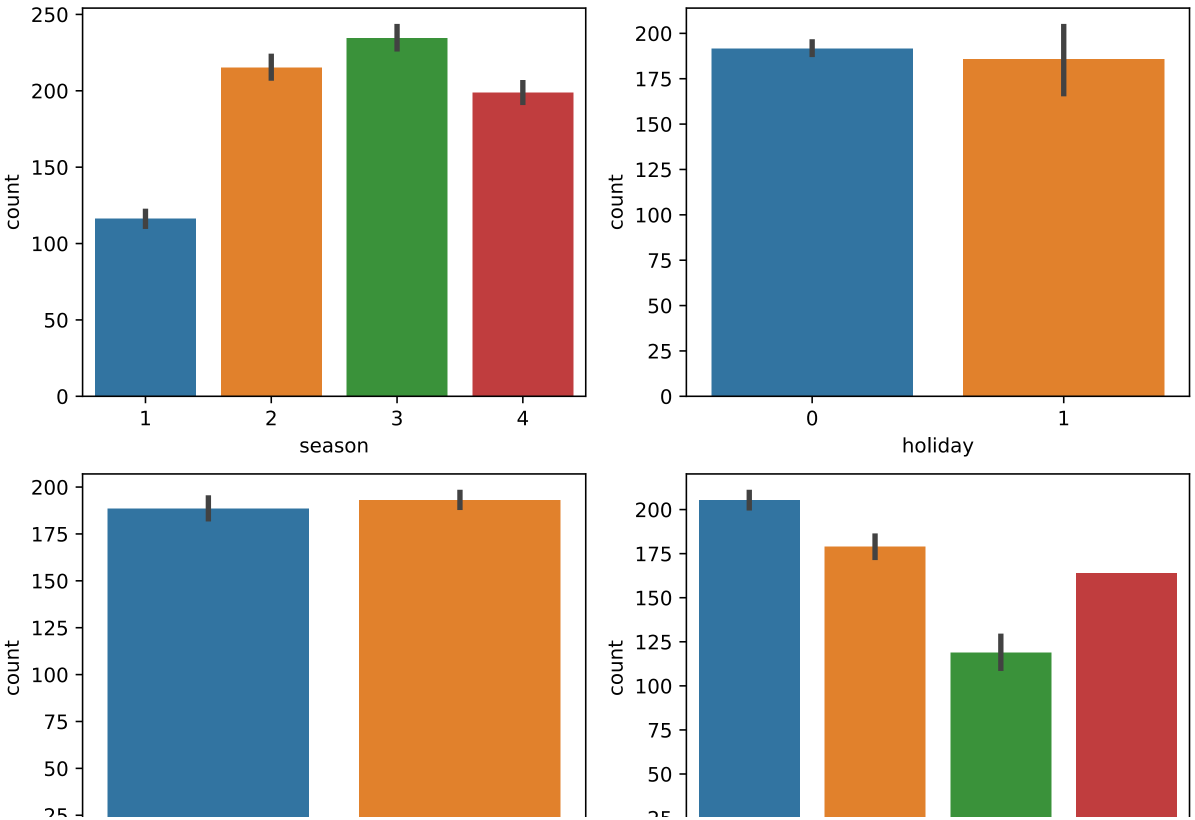


Most number of bookings were done in weather type 1, which is Clear, Few clouds, partly cloudy, partly cloudy And almost zero in weather type 4 , which is Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog

```
df["datetime"] = pd.to_datetime(df["datetime"])
df
df['dates'] = df['datetime'].dt.date
df["hour"] = df["datetime"].dt.hour
df["month"] = df["datetime"].dt.month
df["year"] = df["datetime"].dt.year
```

Bivariate Analysis

```
status_flag=['season','holiday','workingday','weather','hour','month']

fig,axes=plt.subplots(3,2,figsize=(10,12))

i = 0;j= 0
for column in status_flag:
    plot_df = sns.barplot(data=df,x=column,y='count',ax=axes[i,j])

    j=j+1
    if j%2==0:
        i=i+1;
        j=0
```



1.Fall season has highest number of bookings

2.peak hour for booking is from 4pm to 7 pm may be due to office hour rush

3.After 12 Am to 5 Am bookings are almost negligible

4.It shows that most of the rides are booked during office closing and opening hours.

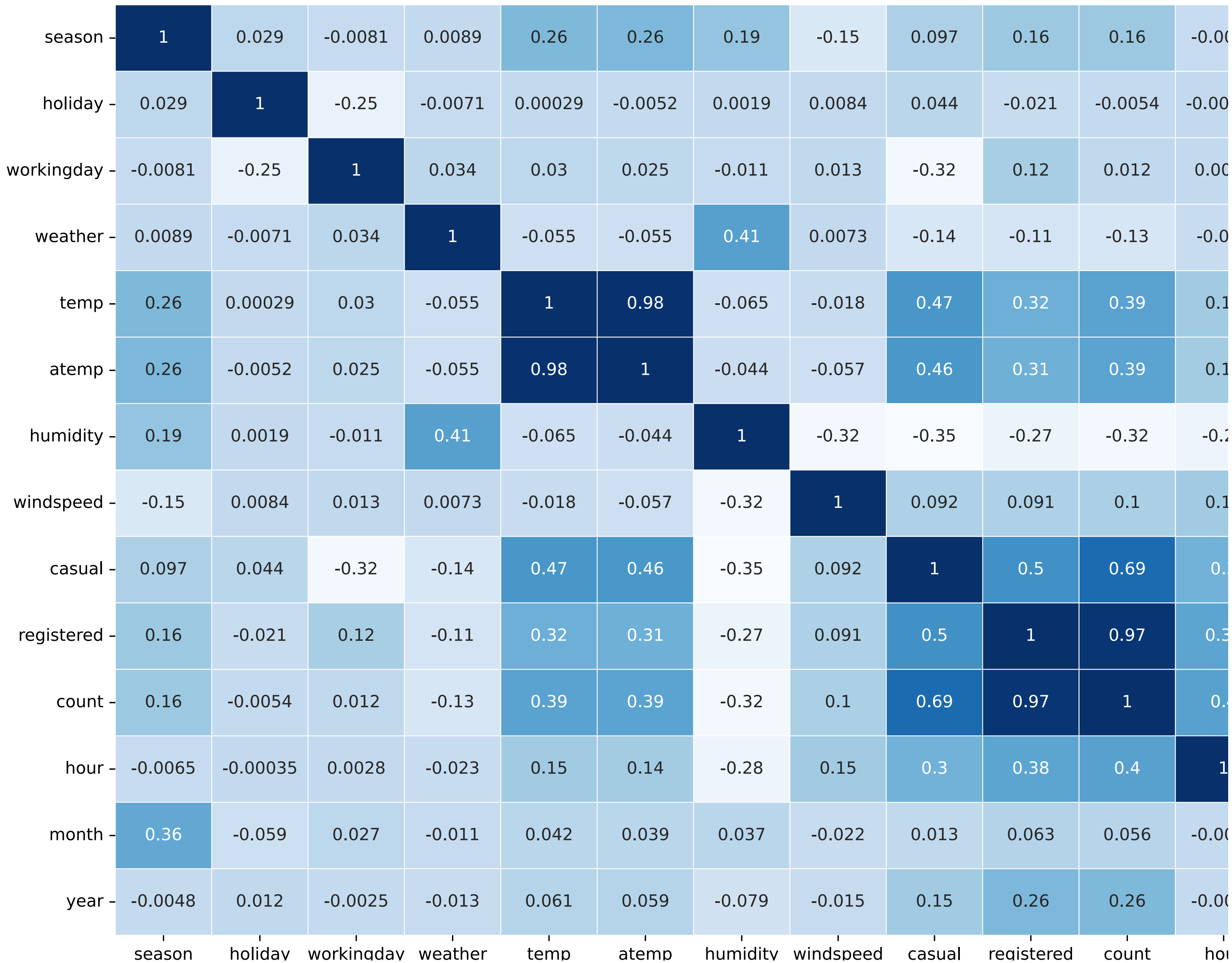
Hence most of our target audience is Working Class.

Double-click (or enter) to edit

Double-click (or enter) to edit

```
plt.figure(figsize=(18,10))
sns.heatmap(df.corr(),linewidths=0.5,cmap="Blues",annot=True)
```

<Axes: >



Here we are doing the analysis based on the count of bookings So, we calculate the corelation with count Here are some conclusions : 1.There is a strong correlation between casual and registered with count column which is more than 50% 2.There is a positive correlation between count and temparature 3.Humidity is negetively correlated

Hypothesis Testing

1. Checking if Working Day has effect on number of electric cycles rented

Null Hypothesis(Ho) : There is no effect of working day on the number of electric cycles rented . Alternate Hypothesis(Ha) : Work day has effect on the number of electric cycle rented .

For this let's take confidence interval to be 95% and signifance level to be 5%

```
workday=df[df['workingday']==1]
```

```
offday=df[df['workingday']==0]
```

we can perform t-test for this to check the p value

```
from scipy.stats import ttest_ind
ttest_ind(workday['count'],offday['count'])
```

```
TtestResult(statistic=1.2096277376026694, pvalue=0.22644804226361348, df=10884.0)
```

Since pvalue is 0.226 which greater than 0.05 so we cannot reject the Null hypothesis.

- conclusion : We don't have the sufficient evidence to say that working day has effect on the number of cycles being rented.

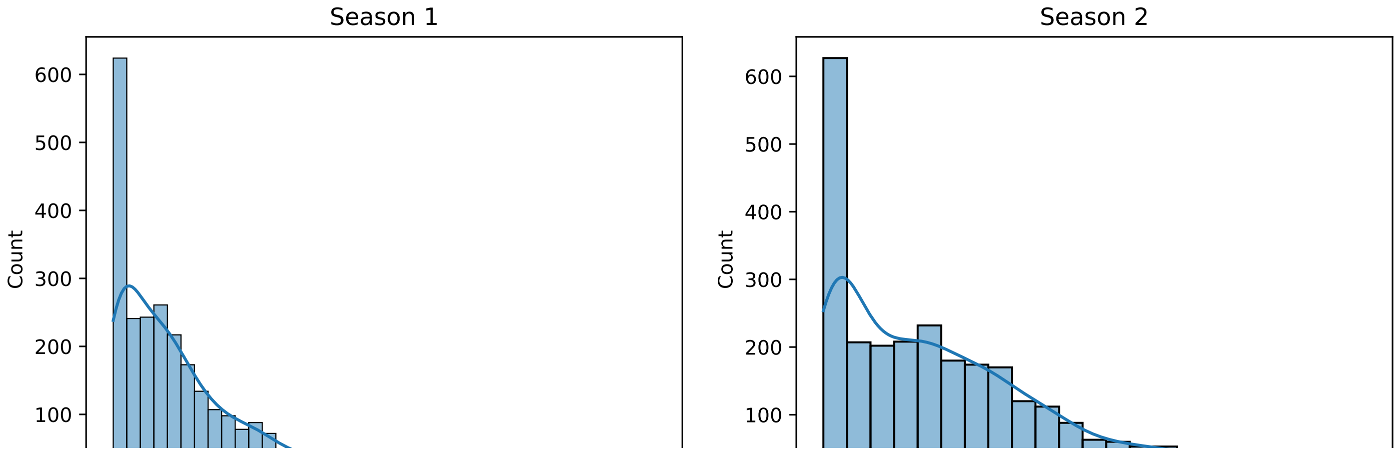
2. Checking if Season has any effect on number of electric cycles rented

Null hypothesis(Ho): Weather can not effect the number of bookings Alternate hypothesis(Ha): Weather can affect the numbre of rented cycles significance level > 0.05

Double-click (or enter) to edit

```
season_1 = df[df["season"]==1]
season_2 = df[df["season"]==2]
season_3 = df[df["season"]==3]
season_4 = df[df["season"]==4]

fig, axes = plt.subplots(2, 2, figsize=(10, 8))
```

```
# Plot 1: Season 1
sns.histplot(data=season_1, x="count", kde=True, ax=axes[0, 0])
axes[0, 0].set_title("Season 1")

# Plot 2: Season 2
sns.histplot(data=season_2, x="count", kde=True, ax=axes[0, 1])
axes[0, 1].set_title("Season 2")

# Plot 3: Season 3
sns.histplot(data=season_3, x="count", kde=True, ax=axes[1, 0])
axes[1, 0].set_title("Season 3")

# Plot 4: Season 4
sns.histplot(data=season_4, x="count", kde=True, ax=axes[1, 1])
axes[1, 1].set_title("Season 4")

# Adjust layout
plt.tight_layout()

# Show the plots
plt.show()
```



Since all the four plots are rightly skewed which shows that data is not normally distributed. Hence we cannot directly use Anova Test.

Hence we can use Kruskal - Wallis test here.

Kruskal Wallis Test

Ho: Season has no effect on number of rides book.

Ha: Season affects the number of rides book.

```
from scipy.stats import kruskal
kruskal(season_1['count'],season_2['count'],season_3['count'],season_4['count'])
```

```
KruskalResult(statistic=699.6668548181988, pvalue=2.479008372608633e-151)
```

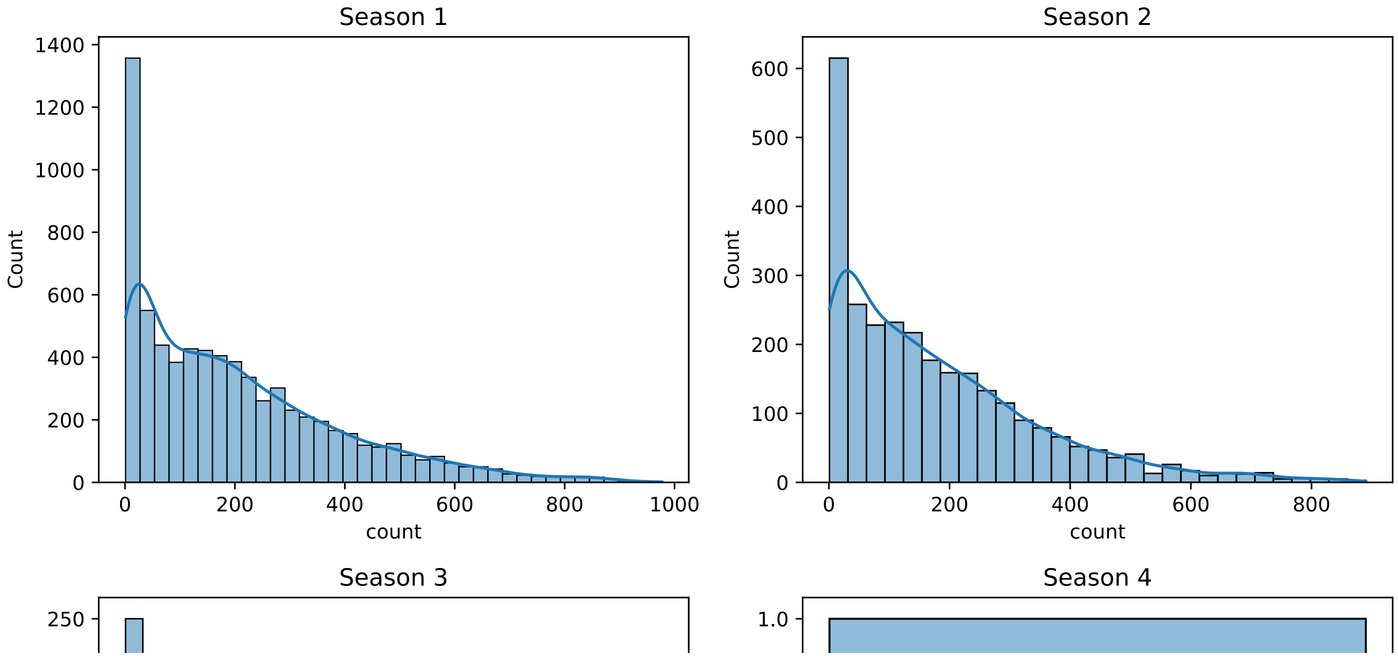
Here we can see that is very very less than the significance level.

conclusion:

We can reject the null hypothesis . So we can conclude that the season has effect on number of bookings

2. Checking if weather has any effect on number of electric cycles rented

```
weather_1 = df[df["weather"]==1]
weather_2 = df[df["weather"]==2]
weather_3 = df[df["weather"]==3]
weather_4 = df[df["weather"]==4]

fig, axes = plt.subplots(2, 2, figsize=(10, 8))

# Plot 1: Season 1
sns.histplot(data=weather_1, x="count", kde=True, ax=axes[0, 0])
axes[0, 0].set_title("Season 1")

# Plot 2: Season 2
sns.histplot(data=weather_2, x="count", kde=True, ax=axes[0, 1])
axes[0, 1].set_title("Season 2")

# Plot 3: Season 3
sns.histplot(data=weather_3, x="count", kde=True, ax=axes[1, 0])
axes[1, 0].set_title("Season 3")

# Plot 4: Season 4
sns.histplot(data=weather_4, x="count", kde=True, ax=axes[1, 1])
axes[1, 1].set_title("Season 4")

# Adjust layout
plt.tight_layout()

# Show the plots
plt.show()
```



Since all the four plots are rightly skewed which shows that data is not normally distributed. Hence we cannot directly use Anova Test.

Hence we can use Kruskal - Wallis test here.

Kruskal Wallis Test

Ho: Weather has no effect on number of rides book.

Ha: Weather affects the number of rides book.

```
kruskal(weather_1['count'], weather_2['count'], weather_3['count'], weather_4['count'])
```

```
KruskalResult(statistic=205.00216514479087, pvalue=3.501611300708679e-44)
```

Here we can see that is very very less than the significance level.

conclusion:

We can reject the null hypothesis . So we can conclude that the Weather has effect on number of bookings

4. Checking if season and weather is mutually dependent or not

Null Hypothesis (H0): Weather is independent of the season Alternate Hypothesis (H1): Weather is not independent of the season Significance level (alpha): 0.05

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

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

```
from scipy.stats import chi2_contingency

val = stats.chi2_contingency(dm)
print(val)
```

```
Chi2ContingencyResult(statistic=49.158655596893624, pvalue=1.549925073686492e-07, dof=9, expected_freq=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]]))
```

here p value is 1.549925073686492e-07 Since p-value is less than the alpha 0.05, We reject the Null Hypothesis.

Conclusion

Weather is dependent on the season.

Business Insights

1.The number of booking is almost smiliar in different season

2.97% booking are during non-holidays

3.Around 70% of the rides are booked on working days while 30% are booked on non working days.

4.Most number of bookings were done in weather type 1, which is Clear, Few clouds, partly cloudy, partly cloudy And almost zero in weather type 4 , which is Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog

5.peak hour for booking is from 4pm to 7 pm may be due to office hour rush

6.After 12 Am to 5 Am bookings are almost negligible

7.It shows that most of the rides are booked during office closing and opening hours.

Hence most of our target audience is Working Class.

8.There is a strong correlation between casual and registered with count column which is more than 50%

9.There is a positive correlation between count and temparature

Business Recommendations

1. Based on ride booking pattern we can see that working class are the biggest customers.They should be the primary focus customers.
2. The availability of vehicle around office areas should be more during office hours to reduce waiting time and improve customer experience .
3. Weather impact on number of bookings ,so based on the weather forecast ride numbers can be adjusted .
4. Since most of the rides are booked from June to October, it can be interpreted that people mostly tend to book Yulu during Summer Season.