BUSINESS CASE: YULU

Problem Definition: The goal is to analyse the demand for Yulu's shared electric cycles in the Indian market and identify significant variables that affect this demand. The dataset contains information on factors like season, holiday status, working day status, weather, temperature, windspeed, and cycle rentals. We aim to understand the relationships between these variables and how they influence the total rental count.

Data Structure Overview:

```
[1] import pandas as pd

[2] data = pd.read_csv("/content/bike_sharing.txt")

# Check the structure and data types of the dataset
print("Data Types and Shape of Data:")
print(data.info())
```

```
Data Types and Shape of Data:
<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
None
```

```
# Check for missing values
print("\nMissing Values:")
print(data.isnull().sum())
```

```
Missing Values:
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
```

```
# Statistical Summary for Continuous Variables
print("\nStatistical Summary:")
print(data.describe())
```

```
Statistical Summary:
             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
```

- Rows: 10,886
- Columns: 12
- Data Types:
    1. Datetime: datetime64
    2. Season, Holiday, Workingday, Weather: int64 (categorical variables)
    3. Temperature, Atemp, Humidity, Windspeed: float64 (continuous variables)
    4. Casual, Registered, Count: int64 (continuous variables)
- Missing Values: No missing values detected, which simplifies the analysis.

Converting Categorical Attributes to 'Category' Type (if required):

```
# Convert 'season', 'holiday', 'workingday', and 'weather' to category if necessary
data['season'] = data['season'].astype('category')
data['holiday'] = data['holiday'].astype('category')
data['workingday'] = data['workingday'].astype('category')
data['weather'] = data['weather'].astype('category')

# Verifying the conversion
print("\nData Types after conversion to category:")
print(data.dtypes)
```

```
Data Types after conversion to category:
datetime        object
season        category
holiday       category
workingday    category
weather       category
temp           float64
atemp          float64
humidity         int64
windspeed      float64
casual           int64
registered       int64
count            int64
dtype: object
```

Summary Statistics:

- Season: Values range from 1 to 4, representing the four seasons.
- Weather: Values range from 1 to 4, representing different weather conditions.

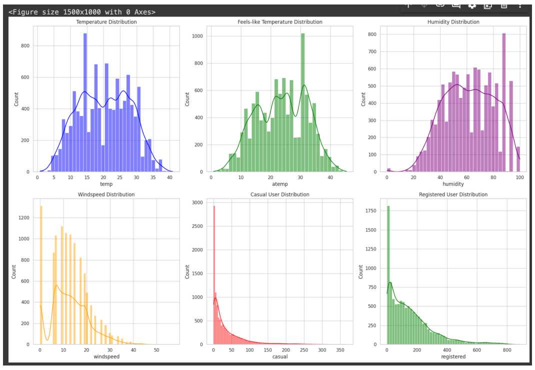
- Temperature: Ranges from 0.82°C to 41°C.
- Humidity: Varies from 0% to 100%.
- Windspeed: Ranges from 0 to 56.99, indicating significant variability in weather conditions.
- Count: Total rentals range from 1 to 977.

Univariate Analysis:

Continuous Variables:

- Temperature (Temp & Atemp): Checking how the temperature varies and how it might affect the rental count.
- Humidity: High variability could affect user comfort and demand.
- Windspeed: Likely to affect cycling behaviour.
- Casual, Registered, Count: Understanding the distribution of registered vs. casual users and overall demand.

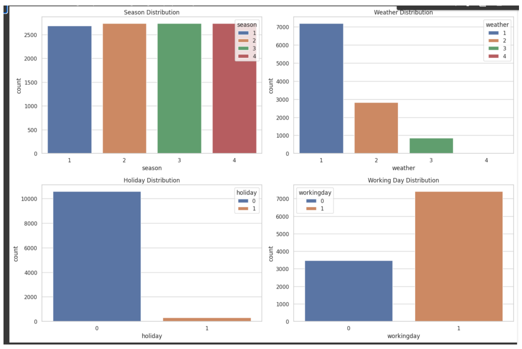
```python
import matplotlib.pyplot as plt
import seaborn as sns

# Set up the plot size and style
plt.figure(figsize=(15, 10))
sns.set(style="whitegrid")

# Plot continuous variables
fig, axes = plt.subplots(2, 3, figsize=(18, 12))

# Temperature
sns.histplot(data['temp'], ax=axes[0, 0], kde=True, color='blue').set(title='Temperature Distribution')
sns.histplot(data['atemp'], ax=axes[0, 1], kde=True, color='green').set(title='Feels-like Temperature Distribution')

# Humidity
sns.histplot(data['humidity'], ax=axes[0, 2], kde=True, color='purple').set(title='Humidity Distribution')

# Windspeed
sns.histplot(data['windspeed'], ax=axes[1, 0], kde=True, color='orange').set(title='Windspeed Distribution')

# Casual and Registered users
sns.histplot(data['casual'], ax=axes[1, 1], kde=True, color='red').set(title='Casual User Distribution')
sns.histplot(data['registered'], ax=axes[1, 2], kde=True, color='green').set(title='Registered User Distribution')

plt.tight_layout()
plt.show()
```



Categorical Variables:

- Season: Analysing the seasonal distribution.
- Weather: Exploring how weather conditions vary.
- Holiday, Workingday: Exploring the impact of holidays and working days.

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Barplots for Categorical Variables: Season, Weather, Holiday, Workingday
fig, axes = plt.subplots(2, 2, figsize=(15, 10))

# Season
sns.countplot(x='season', data=data, ax=axes[0, 0], hue='season').set(title='Season Distribution')

# Weather
sns.countplot(x='weather', data=data, ax=axes[0, 1], hue='weather').set(title='Weather Distribution')

# Holiday
sns.countplot(x='holiday', data=data, ax=axes[1, 0], hue='holiday').set(title='Holiday Distribution')

# Workingday
sns.countplot(x='workingday', data=data, ax=axes[1, 1], hue='workingday').set(title='Working Day Distribution')

plt.tight_layout()
plt.show()
```



Observations from Univariate Analysis:

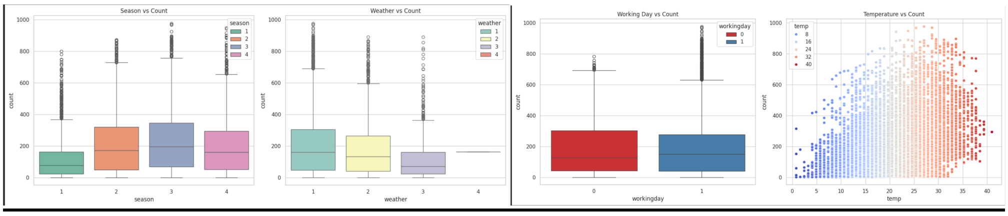
- Temperature & Feels-Like Temperature: Both distributions are approximately normal, with the majority of observations between 10°C and 30°C. This suggests that rentals are likely influenced by comfortable temperatures.
- Humidity: The distribution is right-skewed, with a significant number of observations above 50% humidity, which may affect user comfort.
- Windspeed: The majority of windspeed values are below 20, with very few instances of extreme wind. High windspeed could discourage cycling.

- Casual & Registered Users: Registered users show a higher count compared to casual users, suggesting that most of Yulu's demand comes from frequent or committed users rather than occasional users.
- Season Distribution: Data is evenly distributed across seasons, indicating no immediate seasonal bias in the dataset.
- Weather: Most observations are for clear weather conditions, followed by mist, while heavy rain and snow are less frequent.
- Holiday & Workingday: There are relatively fewer holidays compared to working days, which will be an important factor to test in the bivariate analysis.

Bivariate Analysis:

- Relationship between Important Variables and Count

```python
# Bivariate Analysis: Relationship between important variables and rental count
plt.figure(figsize=(15, 10))

# Set up subplots
fig, axes = plt.subplots(2, 2, figsize=(15, 12))

# Season vs. Count
sns.boxplot(x='season', y='count', data=data, ax=axes[0, 0], hue='season', palette='Set2').set(title='Season vs Count')

# Weather vs. Count
sns.boxplot(x='weather', y='count', data=data, ax=axes[0, 1], hue='weather', palette='Set3').set(title='Weather vs Count')

# Workingday vs. Count
sns.boxplot(x='workingday', y='count', data=data, ax=axes[1, 0], hue='workingday', palette='Set1').set(title='Working Day vs Count')

# Temperature vs. Count
sns.scatterplot(x='temp', y='count', data=data, ax=axes[1, 1], hue='temp', palette='coolwarm').set(title='Temperature vs Count')

plt.tight_layout()
plt.show()
```



Observations from Bivariate Analysis:

- Season vs. Count: The number of rentals varies across seasons, with the highest demand observed in summer and fall (seasons 2 and 3). Winter (season 4) sees the lowest demand, which could be due to lower temperatures and less favourable conditions for cycling.
- Weather vs. Count: Clear weather conditions correspond to higher rental counts, while mist and rainy weather show a decrease in demand. This confirms that weather significantly influences rental behaviour, as expected.
- Working Day vs. Count: Working days (coded as 1) generally show higher rental counts compared to non-working days, indicating that a large portion of Yulu's usage might be for commuting purposes.
- Temperature vs. Count: There appears to be a positive relationship between temperature and rental count up to a certain point, after which extreme temperatures (higher than 30°C) show diminishing returns, possibly due to discomfort from the heat.

Comments on Data Distribution and Relationships:

- Outliers**:** A few extreme outliers are observed in rental counts, particularly in the lower demand seasons and under unfavourable weather conditions.
- Variable Relationships**:** There is a strong relationship between rental count and weather-related variables (temperature, weather), as well as day-related factors (working day). Seasonality also plays an important role in influencing demand.

2-Sample T-Test: Does Working Day Affect the Number of Electric Cycles Rented:

The goal is to test whether the average number of cycles rented is significantly different between working days and non-working days.

Null Hypothesis (H_0): There is no significant difference in the number of electric cycles rented between working and non-working days.

Alternative Hypothesis (H_1): There is a significant difference in the number of electric cycles rented between working and non-working days.

Formula: The t-test formula for two independent samples is given by:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Where:

- $\bar{X}_1$ and $\bar{X}_2$ are the sample means for the two groups.
- s_1 and s_2 are the standard deviations.
- n_1 and n_2 are the sample sizes.

```
from scipy import stats

# Separate data into two groups: working day and non-working day
workingday_0 = data[data['workingday'] == 0]['count']
workingday_1 = data[data['workingday'] == 1]['count']

# Perform 2-sample T-test
t_statistic, p_value_ttest = stats.ttest_ind(workingday_0, workingday_1)

# Display results
print(f"2-Sample T-Test result:")
print(f"t-statistic: {t_statistic}")
print(f"p-value: {p_value_ttest}")
```

```
2-Sample T-Test result:
t-statistic: -1.2096277376026694
p-value: 0.22644804226361348
```

Conclusion (based on p-value):

- If the p-value < 0.05, we reject the null hypothesis and conclude that working days do affect the number of electric cycles rented.
- If p-value > 0.05, we fail to reject the null hypothesis and conclude that working days do not have a significant effect on rentals.

Here, p-value(0.226) > 0.05 so we fail to reject the null hypothesis. That concludes working days do not have a significant effect on rentals.

Insights and Recommendations:

- **Commuter vs. Leisure Usage**: Since there is no significant difference between working and non-working days, this suggests that Yulu's electric cycle rentals are not solely driven by commuters. Leisure or recreational riders may be just as important for demand as commuters.
- **Marketing Strategies**: Yulu should target **both commuting and non-commuting riders** equally. There is no need to emphasize working days over weekends. Promotions could be geared towards leisure and recreational riders, such as weekend discounts, as well as daily commuters.
- **Operational Flexibility**: Since the demand does not fluctuate significantly between working and non-working days, Yulu can **maintain consistent operational levels** (e.g., the number of bikes available) throughout the week without focusing too much on increasing supply on working days or reducing it on weekends.

Recommendation Summary:

- There is no significant difference in rental demand between working days and non-working days, meaning Yulu can focus on serving both commuter and leisure users without adjusting their operational strategies based on the day of the week.

ANOVA: Does the Number of Cycles Rented Differ Across Weather Conditions and Seasons:

The goal is to test whether the mean number of rentals is significantly different across various weather conditions and seasons.

Null Hypothesis (H_0): The mean number of cycles rented is the same across all weather conditions or seasons.

Alternative Hypothesis (H_1): There is a significant difference in the mean number of cycles rented across different weather conditions or seasons.

Formula: The ANOVA test statistic is calculated as:

$$F = \frac{\text{Between-group variance}}{\text{Within-group variance}}$$

Where:

- The **between-group variance** measures the variability between different groups (weather/seasons).
- The **within-group variance** measures the variability within the groups.

ANOVA for Weather:

```
# Perform ANOVA test for weather and count
anova_weather = stats.f_oneway(data[data['weather'] == 1]['count'],
                               data[data['weather'] == 2]['count'],
                               data[data['weather'] == 3]['count'],
                               data[data['weather'] == 4]['count'])

# Display results
print(f"ANOVA (Weather) result:")
print(f"F-statistic: {anova_weather.statistic}")
print(f"p-value: {anova_weather.pvalue}")
```

```
ANOVA (Weather) result:
F-statistic: 65.53024112793271
p-value: 5.482069475935669e-42
```

ANOVA for Weather (p-value = 5.48e-42):

- **Null Hypothesis (H_0)**: The mean number of cycles rented is the same across all weather conditions.
- **Alternative Hypothesis (H_1)**: The mean number of cycles rented is significantly different across different weather conditions.

Since the p-value (5.48e-42) is much smaller than the typical significance level of 0.05, we reject the null hypothesis**.** This means that the number of cycles rented differs significantly across different weather conditions. For instance, users may rent more bikes during clear or favourable weather and less during rainy or stormy weather.

ANOVA for Season:

```python
# Perform ANOVA test for season and count
anova_season = stats.f_oneway(data[data['season'] == 1]['count'],
                              data[data['season'] == 2]['count'],
                              data[data['season'] == 3]['count'],
                              data[data['season'] == 4]['count'])

# Display results
print(f"ANOVA (Season) result:")
print(f"F-statistic: {anova_season.statistic}")
print(f"p-value: {anova_season.pvalue}")
```

```
ANOVA (Season) result:
F-statistic: 236.94671081032106
p-value: 6.164843386499654e-149
```

ANOVA for Season (p-value = 6.16e-149):

- **Null Hypothesis (H_0)**: The mean number of cycles rented is the same across all seasons.
- **Alternative Hypothesis (H_1)**: The mean number of cycles rented is significantly different across different seasons.

Since the p-value (6.16e-149) is also far smaller than 0.05, we reject the null hypothesis. This suggests that the number of rentals varies significantly between different seasons. For example, Yulu might see higher demand in warmer months and lower demand during winter.

Conclusion Based on p-values: Both tests indicate that there are significant differences in the number of cycles rented depending on both the weather conditions and the season.

Insights and Recommendations:

- **Weather's Impact on Demand**: Since weather significantly impacts demand, Yulu can plan their fleet deployment and marketing strategies around weather forecasts. For example, during clear weather, more bikes could be made available at key locations, while during bad weather, they might reduce deployment or offer promotions to incentivize rentals.
- **Seasonal Impact**: Yulu's demand for cycles is also seasonally dependent. During peak seasons (likely spring, summer, or fall), Yulu can optimize pricing, introduce dynamic pricing, and ensure high availability of cycles. During the off-peak season (possibly winter), they could reduce fleet size in certain areas to cut costs and introduce promotional campaigns to maintain demand.

Actionable Strategies:

- Operational adjustments: Align bike availability and maintenance schedules with weather conditions and seasons to ensure optimal operations.
- Marketing campaigns: Promote the service more aggressively during favourable weather and peak seasons, while offering incentives like discounts during the low season or unfavourable weather to boost demand.
- Dynamic pricing: Introduce pricing strategies that account for weather and seasonality to attract more customers during off-peak periods.

Chi-Square Test: Is Weather Dependent on the Season:

The goal is to check if there is an association between weather and season. A chi-square test helps us determine whether two categorical variables are independent or not.

Null Hypothesis (H_0): Weather conditions are independent of the season.

Alternative Hypothesis (H_1): Weather conditions are dependent on the season.

Formula: The chi-square test statistic is calculated as:

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

Where:

- O is the observed frequency.
- E is the expected frequency.

```python
# Create a contingency table for season and weather
contingency_table = pd.crosstab(data['season'], data['weather'])

# Perform Chi-square test
chi2_stat, p_value_chi2, dof, ex = stats.chi2_contingency(contingency_table)

# Display results
print(f"Chi-Square Test result:")
print(f"Chi2-statistic: {chi2_stat}")
print(f"p-value: {p_value_chi2}")
print(f"Degrees of Freedom: {dof}")
```

```
Chi-Square Test result:
Chi2-statistic: 49.158655596893624
p-value: 1.549925073686492e-07
Degrees of Freedom: 9
```

Conclusion (based on p-value):

- If the **p-value** < 0.05, we reject the null hypothesis and conclude that weather is dependent on the season.
- If **p-value** > 0.05, we fail to reject the null hypothesis and conclude that weather is independent of the season.

Since the p-value is 1.55e-07, which is far smaller than the typical significance level of 0.05, we can reject the null hypothesis. This means that weather conditions are dependent on the season.

Insights and Recommendations:

- **Seasonal Weather Patterns**: Since weather is dependent on the season, Yulu can use this insight to plan their operations and fleet deployment. For example, if the rainy season is approaching, they can anticipate a decline in demand and adjust their marketing strategies or deploy fewer bikes in certain areas.
- **Predictive Planning**: Yulu can leverage seasonal weather patterns to forecast demand. By understanding how weather changes with seasons, Yulu could prepare better and optimize resources based on anticipated weather conditions.
- **Dynamic Adjustments**: Adjusting operations, pricing, and promotions based on both seasonal and weather patterns will help Yulu maximize efficiency and reduce operational costs. For example, during the rainy season, Yulu could offer rain gear or umbrellas for rental as part of a promotion, incentivizing customers to ride despite the weather.

Recommendation Summary:

- Weather conditions are dependent on seasons, so Yulu should anticipate changes in demand based on seasonal weather forecasts and align their operational strategies (fleet availability, pricing models, marketing campaigns) accordingly.