Business Case Study: Micro-Mobility Service Provider

Context:

This business case focuses on the operations of a leading micro-mobility service provider which offers bike sharing as safest commute solution through a user-friendly mobile app to enable shared, solo and sustainable commuting. This case study aims to analyze the factors affecting the demand for shared electric cycles in the Indian market, provide data driven insights and actionable business recommendations to help the business target specific Customer base with more interesting products.

This case study report contains the solutions to the problem statements (as Python queries by employing data visualisation, descriptive Statistics & Probability), sample output of the queries, followed by insights and recommendations. As part of the confidentiality agreement, the name of the service provider, the actual dataset and problem statements are not included in this report.

[Google Colab Notebook pdf]() - This Python project involves exploratory data analysis of a dataset from this service provider. The code is importing necessary libraries such as numpy, pandas, seaborn, scipy.stats and matplotlib.

Importing libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency, levene, shapiro, ttest_ind, f_oneway
from scipy.stats import kruskal, kstest
from statsmodels.graphics.gofplots import qqplot
```

Loading Data

```
data = pd.read_csv("bike_sharing.csv")
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	0.0	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

Shape of the dataset and Column DataTypes

```
data.shape
data.info()
```

```
data.shape
```

```
(10886, 12)
```

```
data.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
```

Insights: There are a total of 10,886 rows (data points) and 12 columns. Following columns have integer datatype – season, holiday, workingday, weather, humidity, casual, registered and count. Remaining columns such as datetime, temp, atemp, windspeed have object datatype.

Null/Missing Values and Duplicate Values Detection

```
data.isna().sum()
data.duplicated().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 or duplicate values in the dataset

Updating Categorical Columns

```
updated_data = data.copy()
updated_data["season"] = updated_data["season"].apply(lambda x: "Spring" if x==1
else "Summer" if x==2 else "Fall" if x==3 else "Winter")

updated_data["holiday"] = updated_data["holiday"].apply(lambda x: "Not a holiday"
if x==0 else "Holiday")

updated_data["workingday"] = updated_data["workingday"].apply(lambda x:
"Weekend/Holiday" if x==0 else "Working day")

updated_data["Day_Type"] = updated_data.apply(lambda x: "Weekend" if
x["workingday"] == "Weekend/Holiday" and x["holiday"] == "Not a holiday" else
"Holiday" if x["workingday"] == "Weekend/Holiday" and x["holiday"] == "Holiday"
else "Working day", axis=1)

updated_data["weather"] = updated_data["weather"].apply(lambda x: "Clear+Few
clouds" if x==1 else "Mist+Cloudy" if x==2 else "Light Snow+Light Rain" if x==3
else "Heavy Rain")
```

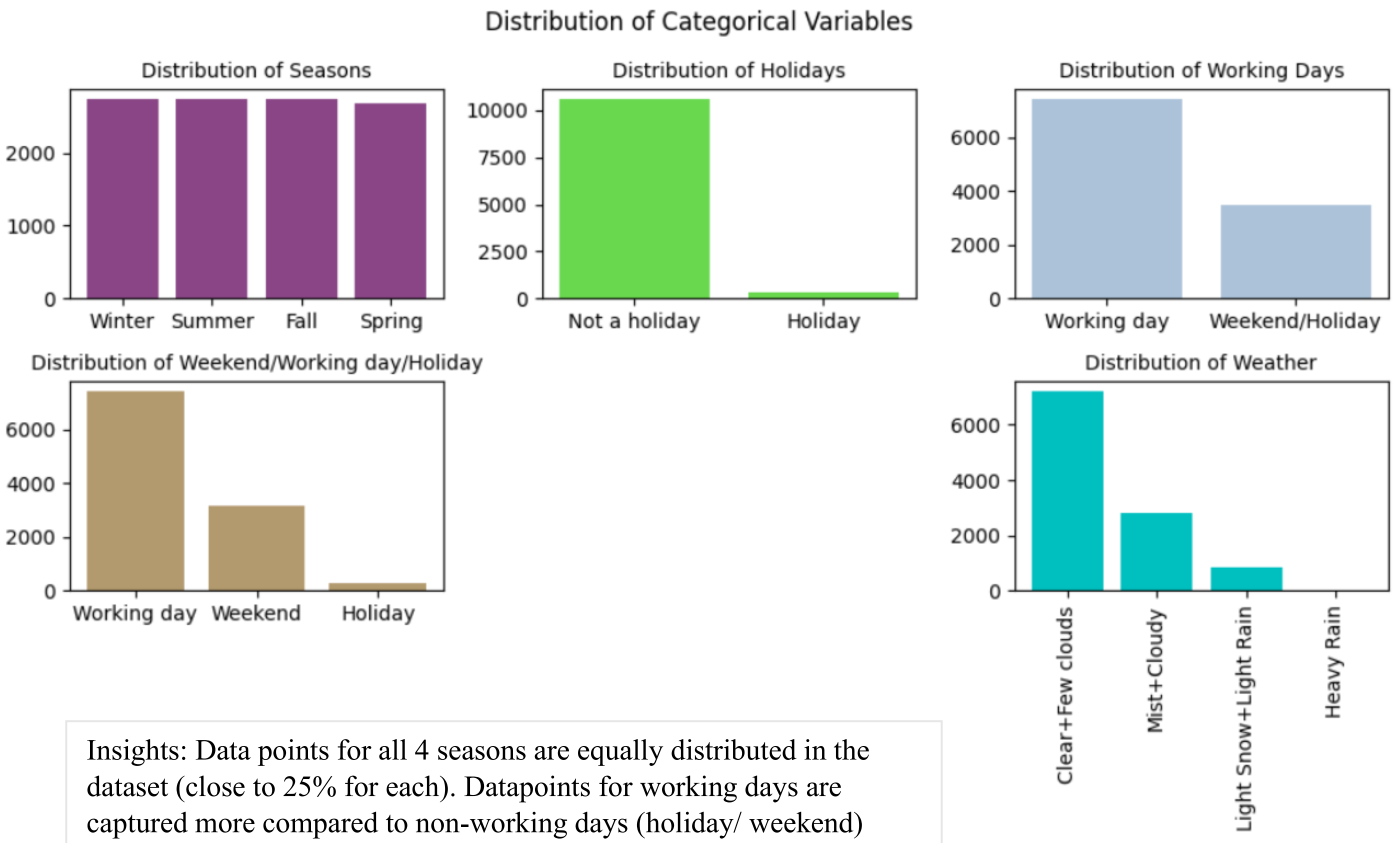
	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count	Day_Type
0	2011-01-01 00:00:00	Spring	Not a holiday	Weekend/Holiday	Clear+Few clouds	9.84	14.395	81	0.0000	3	13	16	Weekend
1	2011-01-01 01:00:00	Spring	Not a holiday	Weekend/Holiday	Clear+Few clouds	9.02	13.635	80	0.0000	8	32	40	Weekend
2	2011-01-01 02:00:00	Spring	Not a holiday	Weekend/Holiday	Clear+Few clouds	9.02	13.635	80	0.0000	5	27	32	Weekend
3	2011-01-01 03:00:00	Spring	Not a holiday	Weekend/Holiday	Clear+Few clouds	9.84	14.395	75	0.0000	3	10	13	Weekend
4	2011-01-01 04:00:00	Spring	Not a holiday	Weekend/Holiday	Clear+Few clouds	9.84	14.395	75	0.0000	0	1	1	Weekend
...	...	...	...	...	...	...	...	...	...	...	...	...	...
10881	2012-12-19 19:00:00	Winter	Not a holiday	Working day	Clear+Few clouds	15.58	19.695	50	26.0027	7	329	336	Working day
10882	2012-12-19 20:00:00	Winter	Not a holiday	Working day	Clear+Few clouds	14.76	17.425	57	15.0013	10	231	241	Working day
10883	2012-12-19 21:00:00	Winter	Not a holiday	Working day	Clear+Few clouds	13.94	15.910	61	15.0013	4	164	168	Working day
10884	2012-12-19 22:00:00	Winter	Not a holiday	Working day	Clear+Few clouds	13.94	17.425	61	6.0032	12	117	129	Working day
10885	2012-12-19 23:00:00	Winter	Not a holiday	Working day	Clear+Few clouds	13.12	16.665	66	8.9981	4	84	88	Working day

10886 rows × 13 columns

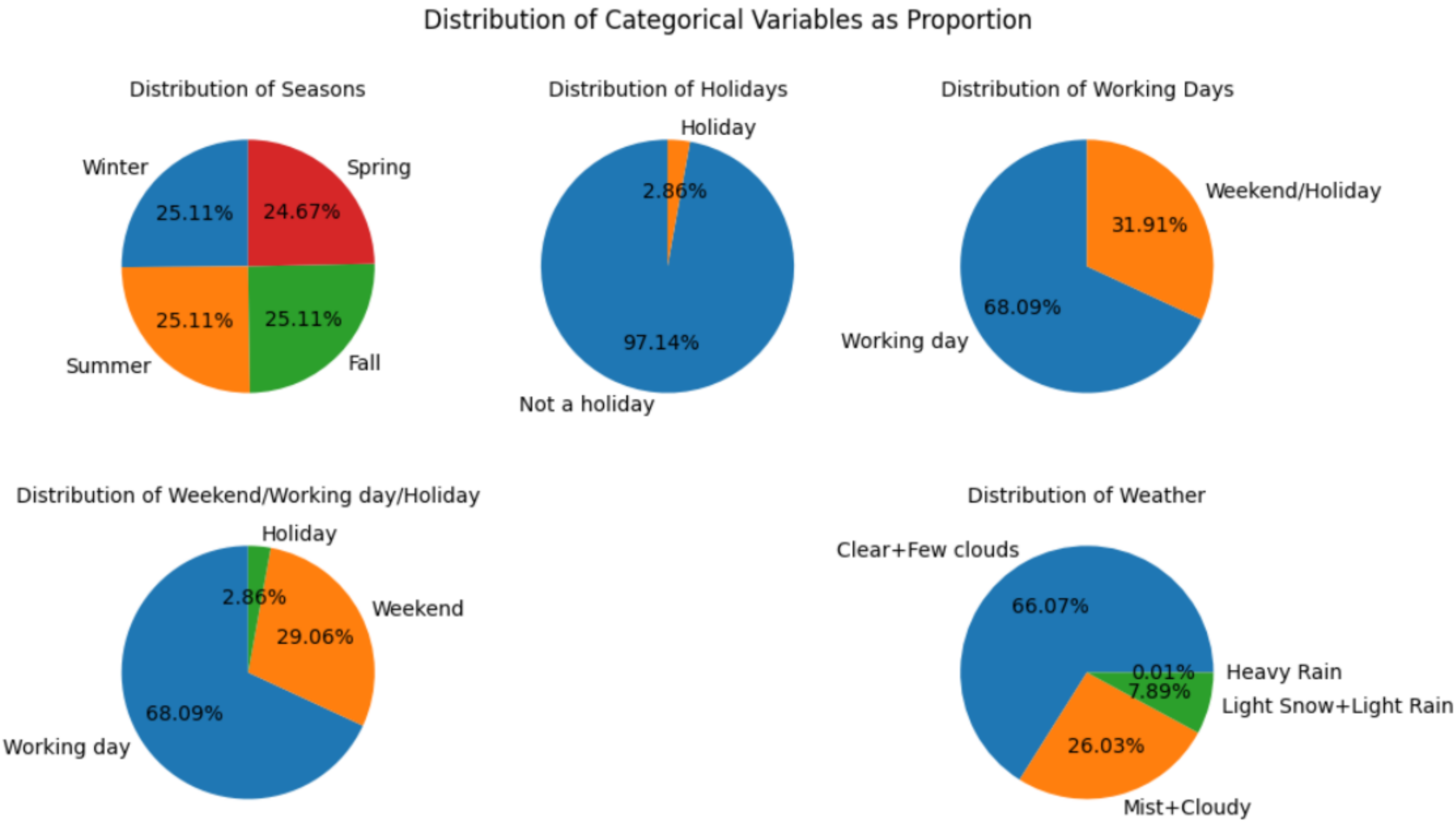
Univariate Analysis

Distribution of Categorical Variables

```
plt.figure(figsize = (10, 6)).suptitle("Distribution of Categorical Variables")

plt.subplot(2,3,1)
plt.bar(updated_data["season"].value_counts().index, updated_data["season"].
value_counts(), color = "#894585")
plt.title("Distribution of Seasons", fontsize = 10)

plt.subplot(2,3,2)
plt.bar(updated_data["holiday"].value_counts().index, updated_data["holiday"].
value_counts(), color = "#69d84f")
plt.title("Distribution of Holidays", fontsize = 10)

plt.subplot(2,3,3)
plt.bar(updated_data["workingday"].value_counts().index, updated_data["workingday"]
.value_counts(), color = "#acc2d9")
plt.title("Distribution of Working Days", fontsize = 10)

plt.subplot(2,3,4)
plt.bar(updated_data["Day_Type"].value_counts().index, updated_data["Day_Type"].
value_counts(), color = "#b2996e")
plt.title("Distribution of Weekend/Working day/Holiday", fontsize = 10)

plt.subplot(2,3,6)
plt.bar(updated_data["weather"].value_counts().index, updated_data["weather"].
value_counts(), color = "c")
plt.xticks(rotation = 90)
plt.title("Distribution of Weather", fontsize = 10)

plt.tight_layout()
```



Insights: Data points for all 4 seasons are equally distributed in the dataset (close to 25% for each). Datapoints for working days are captured more compared to non-working days (holiday/ weekend)



Insights:

1. 68% (7412/10886) records are of working days & remaining are of Weekend/Holiday (32%)
2. When we split the records for Weekend/Holiday, out of the 32% of records of it, 29.06% are of weekends and rest are of holidays.
3. Out of total records, 66% (7192 / 10886) of the rows are of weather – Clear+ Few clouds, followed by for Mist+Cloudy (26%), Light snow+rain (8%). There is only one data point where weather – Heavy Rain was recorded.

Detect Outliers and Skewness (using boxplot, histogram, "describe" method by checking the difference between mean and median)

```
updated_data.describe()
```

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

Get the difference between mean and median for purchase column to identify outliers:

```
Updated_data.describe().loc["mean"] - updated_data.describe().loc["50%"]
```

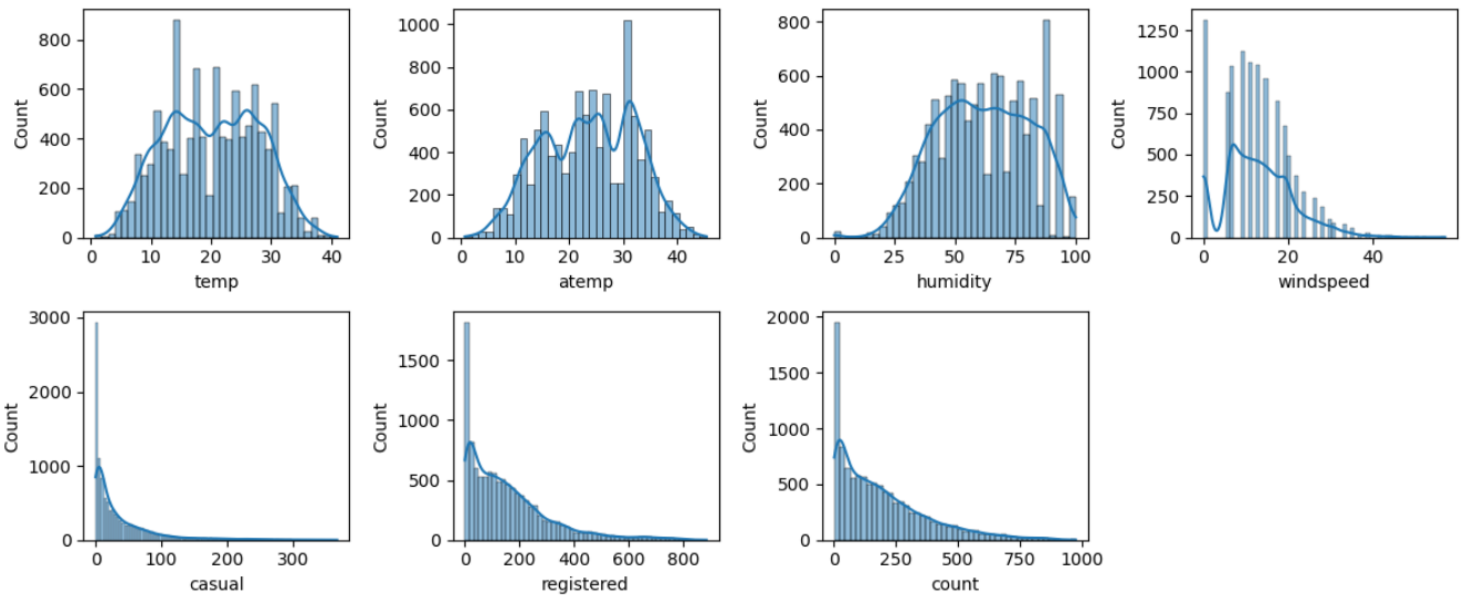
	0
temp	-0.269140
atemp	-0.584916
humidity	-0.113540
windspeed	-0.198605
casual	19.021955
registered	37.552177
count	46.574132

dtype: float64

Insights: Difference between mean and median of following columns –> casual, registered, and count suggests that they are right skewed. The mean is higher than the median.

Univariate Analysis: Distribution of Continuous Variables – To identify skewness

```
plt.figure(figsize = (12,5))

plt.subplot(2,4,1)
sns.histplot(updated_data["temp"], kde = True)

plt.subplot(2,4,2)
sns.histplot(updated_data["atemp"], kde = True)

plt.subplot(2,4,3)
sns.histplot(updated_data["humidity"], kde = True)

plt.subplot(2,4,4)
sns.histplot(updated_data["windspeed"], kde = True)

plt.subplot(2,4,5)
sns.histplot(updated_data["casual"], kde = True)

plt.subplot(2,4,6)
sns.histplot(updated_data["registered"], kde = True)

plt.subplot(2,4,7)
sns.histplot(updated_data["count"], kde = True)

plt.tight_layout()
```

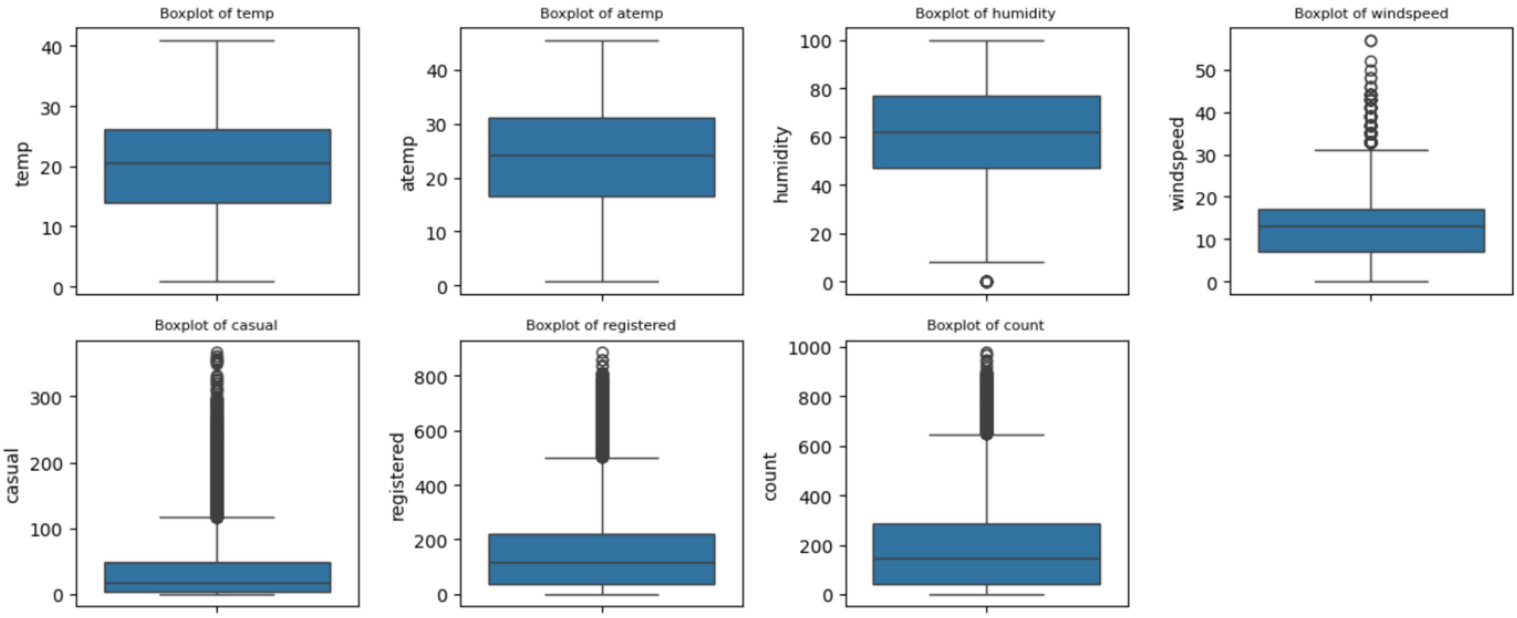


Insights:

1. The histograms for temp and atemp show a roughly symmetric distribution, with very slight skewness. These are close to a normal distribution.
2. The tail of the humidity distribution slightly extends to the left indicating slight negative skewness, but it still resembles a normal distribution.
3. Windspeed is positively skewed, with a tail extending to the right, indicating that lower wind speeds are more common, but there are some instances of higher wind speeds.
4. All three variables – casual, registered and count are heavily positively skewed with a concentration of data on the lower end and long tails extending to the right. They could have outliers with extreme values to the right. The value of standard deviation is also high which tells us that there is high variance in the data of these attribute

Identifying outliers using Boxplots

```
continuous_var = ["temp", "atemp", "humidity", "windspeed", "casual", "registered",
"count"]

plt.figure(figsize = (12,5))
for i,var in enumerate(continuous_var, 1):
  plt.subplot(2,4,i)
  sns.boxplot(y = updated_data[var])
  plt.title(f"Boxplot of {var}", fontsize = 8)
plt.tight_layout()
```



Insights: Following variables – windspeed, casual, registered and count seem to have many outliers above the upper whisker. It was found earlier in an above plot that all 4 also have right skewness.

Identifying IQR, lower and upper whisker for continuous variables

```
def whisker_limit(data, column):
  Q1 =  np.percentile(data[column], 25)
  Q3 =  np.percentile(data[column], 75)
  IQR = Q3 - Q1
  lower_whisker = Q1 - 1.5*IQR
  upper_whisker= Q3 + 1.5*IQR
```

```
  print(f"For {column} - IQR ->{IQR: .2f}, lower_whisker-> {lower_whisker: .2f},
upper_whisker-> {upper_whisker: .2f}")
for i in continuous_var:
  whisker_limit(updated_data, [i])
```

```
For ['temp'] - IQR -> 12.30, lower_whisker-> -4.51, upper_whisker->  44.69
For ['atemp'] - IQR -> 14.39, lower_whisker-> -4.93, upper_whisker->  52.65
For ['humidity'] - IQR -> 30.00, lower_whisker->  2.00, upper_whisker->  122.00
For ['windspeed'] - IQR -> 10.00, lower_whisker-> -7.99, upper_whisker->  31.99
For ['casual'] - IQR -> 45.00, lower_whisker-> -63.50, upper_whisker->  116.50
For ['registered'] - IQR -> 186.00, lower_whisker-> -243.00, upper_whisker->  501.00
For ['count'] - IQR -> 242.00, lower_whisker-> -321.00, upper_whisker->  647.00
```

Insights:

- The values above the upper whisker (Q3+ 1.5*IQR) and below the lower whisker (Q1 - 1.5*IQR) are considered outliers.

Now, clip the data between the 5 percentile and 95 percentiles by retaining all rows. This allows to set lower and upper bounds for the values in the DataFrame. i.e. it sets the values that are below the 5th percentile to the 5th percentile value, and those above the 95th percentile to the 95th percentile value.

```
def clip_outliers(data, columns):
  clipped_data = updated_data.copy()
  for column in columns:
    lower_bound = round(np.percentile(data[column], 5))
    upper_bound = round(np.percentile(data[column], 95))
    clipped_daSta[column] = data[column].clip(lower = lower_bound, upper =
upper_bound)
  return clipped_data

clipped_data = clip_outliers(updated_data, ["temp", "atemp", "humidity",
"windspeed", "casual", "registered", "count"])
print("Clipped DataFrame:")
clipped_data.head()
```

Results:

Clipped DataFrame:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count	Day_Type
0	2011-01-01 00:00:00	Spring	Not a holiday	Weekend/Holiday	Clear+Few clouds	9.84	14.395	81	0.0000	3	13	16	Weekend
1	2011-01-01 01:00:00	Spring	Not a holiday	Weekend/Holiday	Clear+Few clouds	9.02	13.635	80	0.0000	8	32	40	Weekend
2	2011-01-01 02:00:00	Spring	Not a holiday	Weekend/Holiday	Clear+Few clouds	9.02	13.635	80	0.0000	5	27	32	Weekend
3	2011-01-01 03:00:00	Spring	Not a holiday	Weekend/Holiday	Clear+Few clouds	9.84	14.395	75	0.0000	3	10	13	Weekend
4	2011-01-01 04:00:00	Spring	Not a holiday	Weekend/Holiday	Clear+Few clouds	9.84	14.395	75	0.0000	0	4	5	Weekend
...	...	...	...	...	...	...	...	...	...	...	...	...	...
10881	2012-12-19 19:00:00	Winter	Not a holiday	Working day	Clear+Few clouds	15.58	19.695	50	26.0027	7	329	336	Working day
10882	2012-12-19 20:00:00	Winter	Not a holiday	Working day	Clear+Few clouds	14.76	17.425	57	15.0013	10	231	241	Working day
10883	2012-12-19 21:00:00	Winter	Not a holiday	Working day	Clear+Few clouds	13.94	15.910	61	15.0013	4	164	168	Working day
10884	2012-12-19 22:00:00	Winter	Not a holiday	Working day	Clear+Few clouds	13.94	17.425	61	6.0032	12	117	129	Working day
10885	2012-12-19 23:00:00	Winter	Not a holiday	Working day	Clear+Few clouds	13.12	16.665	66	8.9981	4	84	88	Working day

10886 rows × 13 columns

Note: We will use this clipped_data for all further analysis

Multivariate Analysis

```
clipped_data.corr(numeric_only = True)
```

	temp	atemp	humidity	windspeed	casual	registered	count
temp	1.000000	0.984909	-0.059048	-0.013655	0.523487	0.332107	0.402680
atemp	0.984909	1.000000	-0.038466	-0.047598	0.516440	0.328008	0.397931
humidity	-0.059048	-0.038466	1.000000	-0.320708	-0.376588	-0.293735	-0.334440
windspeed	-0.013655	-0.047598	-0.320708	1.000000	0.109438	0.107766	0.114688
casual	0.523487	0.516440	-0.376588	0.109438	1.000000	0.589091	0.744404
registered	0.332107	0.328008	-0.293735	0.107766	0.589091	1.000000	0.973644
count	0.402680	0.397931	-0.334440	0.114688	0.744404	0.973644	1.000000

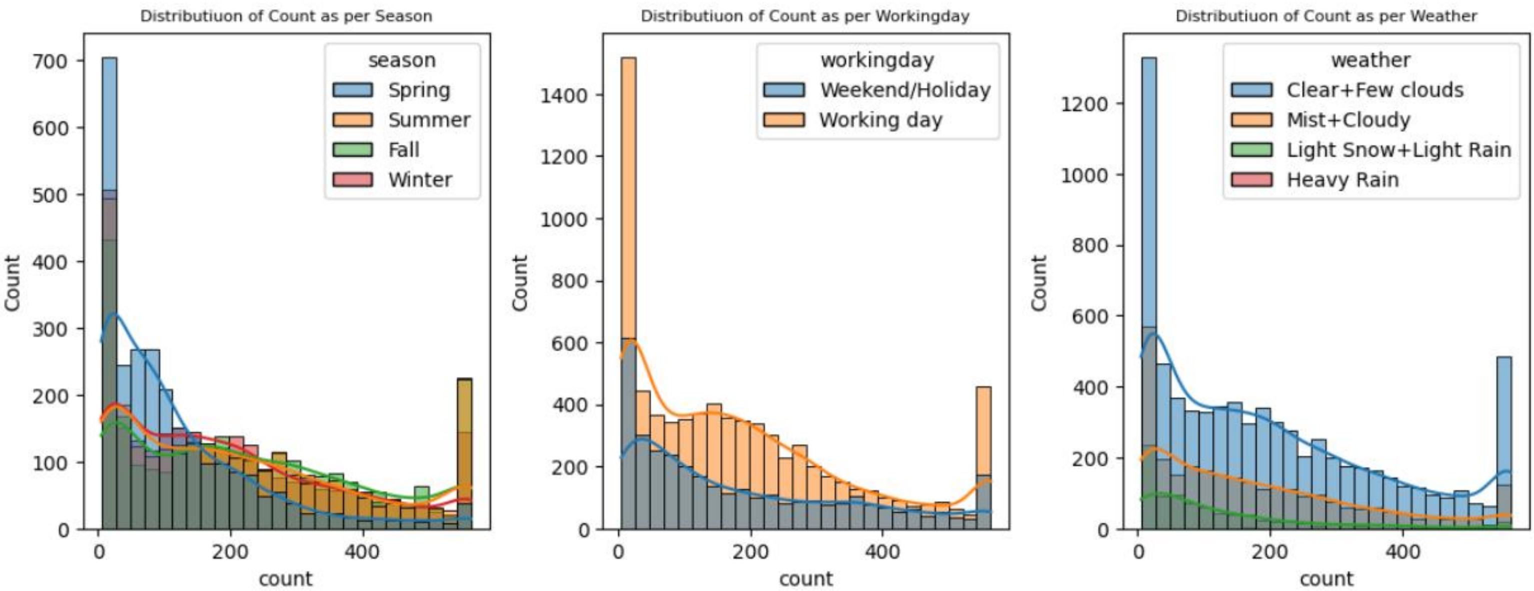
Insights:

- There is a moderate positive correlation between temperature and the total count of rides. This indicates that higher temperatures are associated with more bike rentals.
- There is a very weak positive correlation between windspeed and the total count of rides. This suggests that wind speed has little effect on the total number of rides.
- There is a very weak negative correlation between humidity and count. While high humidity might reduce bike rentals slightly, it's not the sole or dominant factor affecting the number of rentals.

Bivariate Analysis

Relationships between season and count, workday and count, weather and count (count of bikes)

```
sns.histplot(data=clipped_data, x = "count", hue = "season", kde = True)
sns.histplot(data=clipped_data, x = "count", hue = "workingday", kde = True)
sns.histplot(data=clipped_data, x = "count", hue = "weather", kde = True)
```

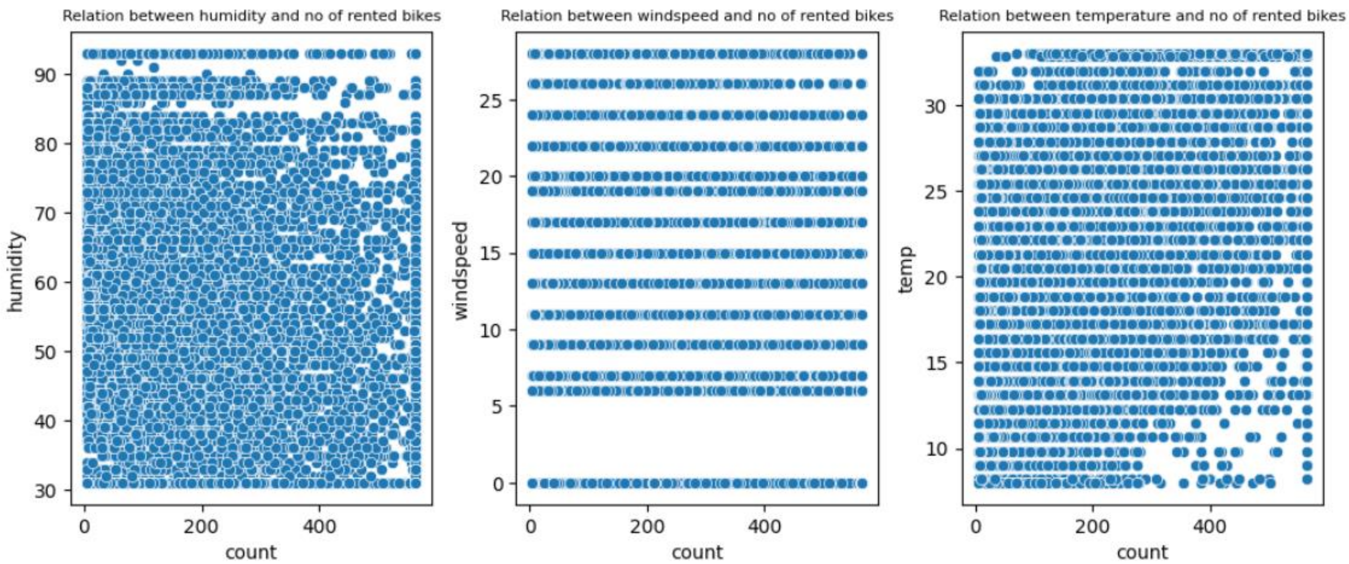


Insights:

- In summer and fall seasons, more bikes are rented as compared to other seasons. This suggests bike usage is most popular in warmer, milder weather.
- Clear or slightly cloudy weather strongly correlates with higher bike rental counts. This makes sense as people are more likely to cycle in pleasant weather.
- The weekend/holiday distribution has a lower overall count, this could be because the number of datapoints (rows) for weekend/holiday is only 32% (3474 / 10886) of total. At the same time, weekend/holiday also shows some high peaks like working days, suggesting concentrated periods of high rental activity. This suggests bike rentals are mostly equally popular during weekends/holidays and working days.
- Skewed distributions: All three histograms show right-skewed distributions, with a high frequency of lower counts and a long tail towards higher counts.

Relationship between humidity and count, windspeed and count, temperature and count

```
plt.figure(figsize = (10,10))

plt.subplot(2,3,1)
sns.scatterplot(data = clipped_data, x = "count", y = "humidity")
plt.title("Relation between humidity and number of rented bikes", fontsize = 8)

plt.subplot(2,3,2)
sns.scatterplot(data = clipped_data, x = "count", y = "windspeed")
plt.title("Relation between humidity and number of rented bikes", fontsize = 8)

plt.subplot(2,3,3)
sns.scatterplot(data = clipped_data, x = "count", y = "temp")
plt.title("Relation between humidity and number of rented bikes", fontsize = 8)

plt.tight_layout()
```



Insights:

- There appears to be a slight positive relationship between temperature and count, with higher counts generally associated with moderate temperatures (20°C to 30°C).
- The distribution of "count" does not show a clear trend with windspeed. Windspeed might not have a strong linear relationship with the number of rentals.
- Most of the bike rentals seem to occur when the humidity is within the 40-80% range, but there are instances of rentals across the entire spectrum of humidity values.

Hypothesis Testing

Test 1: Effect of Working Day on the number of electric cycles rented

	workingday	count
0	Working day	7412
1	Weekend/Holiday	3474

- First, filter data based on different categories within Working day Column to create 2 set of samples – One for Working_day and the other for Weekend/Holiday

```
Weekend_Holiday = clipped_data[clipped_data["workingday"] == "Weekend/Holiday"]
["count"]
Working_day = clipped_data[clipped_data["workingday"] == "Working day"]["count"]
```

- Formulate Null Hypothesis (H0) and Alternate Hypothesis (H1)

Null Hypothesis (Ho): Both samples come from same population. There is no difference in the distributions of number of rented cycles between 2 groups. Number of rented cycles is INDEPENDENT of Working Day (mu1 = mu2)

Alternate Hypothesis (Ha): Both samples come from different population. There is difference in the distributions of number of rented cycles between 2 groups. Number of rented cycles is DEPENDENT on Working Day (mu1 != mu2)

- Set a significance level and calculate the test Statistics / p-value.

At 5% significance level, check if both samples come from same population using **2 sample t-test**

```
ttest_ind(Weekend_Holiday, Working_day, alternative = "two-sided")
```

This will be a two-tailed test as we are checking if there is significant difference in the population mu of both samples (mu != mu)

Result: TtestResult(statistic=0.080747870855918 93, p-value=0.9356439502967323, df=10884.0)

Insights: **P-value of 0.935 is higher than the significance level of 0.05. Hence, we do not reject the Null Hypothesis.** ***The number of electric cycles rented is independent of Working day. We don't have the sufficient evidence to say that working day has effect on the number of cycles being rented.***

Test 2: Effect of seasons on the demand of electric cycles rented

	season	count
0	Winter	2734
1	Summer	2733
2	Fall	2733
3	Spring	2686

- First, filter data based on different categories within Working day Column to create samples

```
Spring = clipped_data[clipped_data["season"] == "Spring"]["count"]
Summer = clipped_data[clipped_data["season"] == "Summer"]["count"]
Fall   = clipped_data[clipped_data["season"] == "Fall"]["count"]
Winter = clipped_data[clipped_data["season"] == "Winter"]["count"]
```

- Formulate Null Hypothesis (H0) and Alternate Hypothesis (H1)

Null Hypothesis (Ho): All 4 samples have identical population means. Number of rented cycles is INDEPENDENT of seasons (mu1 = mu2 = mu3 = mu4)

Alternate Hypothesis (Ha): At least one group has a different population mean. Number of rented cycles is DEPENDENT on season

- Check assumptions of the test

Test of Normality – Shapiro Test – Check if all samples were drawn from a normal distribution

```
print(shapiro(Spring))
print(shapiro(Summer))
print(shapiro(Fall))
print(shapiro(Winter))
```

Results:

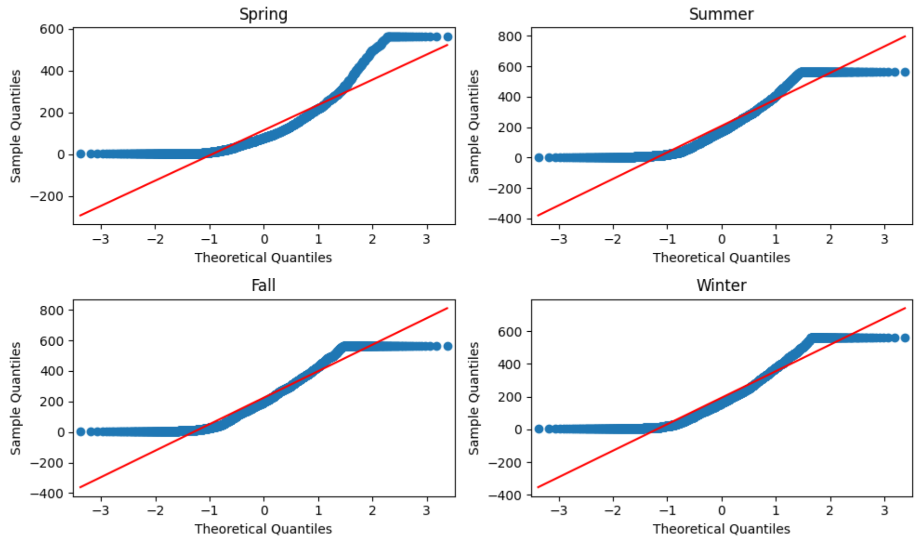
```
ShapiroResult(statistic=0.8185522376447556, pvalue=6.346630787920424e-48)
ShapiroResult(statistic=0.9038195320908844, pvalue=1.8953419034471773e-38)
ShapiroResult(statistic=0.9241542038454934, pvalue=4.4039542903799794e-35)
ShapiroResult(statistic=0.9092254797725821, pvalue=1.27845173286568e-37)
```

P-value is very low for all 4 samples. Hence, can conclude that none of them are normally distributed

QQ plot for visualisation of this normality

```
plt.figure(figsize=(15,6))

plt.subplot(2, 4, 1)
qqplot(Spring, line='s', ax=plt.gca())
plt.title('Spring')

plt.subplot(2, 4, 2)
qqplot(Summer, line='s', ax=plt.gca())
plt.title('Summer')

plt.subplot(2, 4, 3)
qqplot(Fall, line='s', ax=plt.gca())
plt.title('Fall')
plt.subplot(2, 4, 4)
```

```python
qqplot(Winter, line='s', ax=plt.gca())
plt.title('Winter')

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



Insights: All the quantile points would lie along the red line if data was normally distributed. There are major deviations in all 4 plots showing they are not normally distributed.

Test of Equal Variances – Levene – Test to check if all input samples are from populations with equal variances

```python
levene(Spring, Summer, Fall, Winter)
```

Results: LeveneResult(statistic=209.53866770018433, pvalue=3.841611052826981e-132)

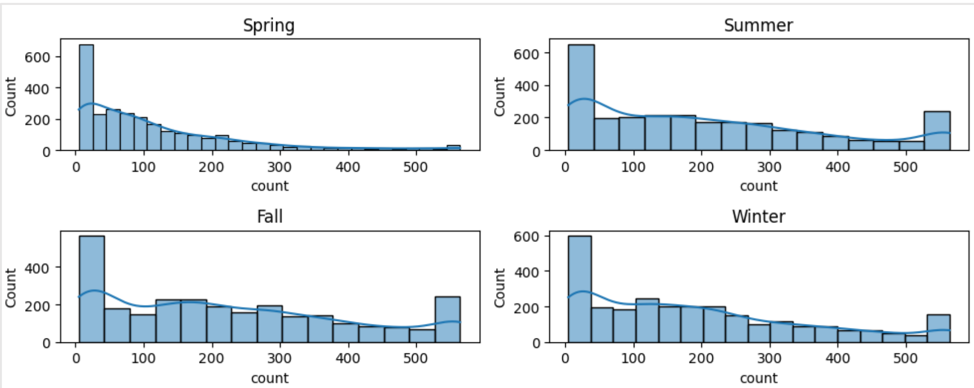
P-value is as low as 3.841611052826981e-132. Hence there is no equal variances.

- Check Skewness

```python
print(f"Skewness of Spring-> {Spring.skew() :.3f}, right skewed because the metric is > 0")
print(f"Skewness of Summer-> {Summer.skew() :.3f}, right skewed because the metric is > 0")
print(f"Skewness of Fall  -> {Fall.skew() :.3f}, right skewed because the metric is > 0")
print(f"Skewness of Winter-> {Winter.skew() :.3f}, right skewed because the metric is > 0")
```

Results:

```
Skewness of Spring-> 1.665, right skewed because the metric is > 0
Skewness of Summer-> 0.662, right skewed because the metric is > 0
Skewness of Fall  -> 0.496, right skewed because the metric is > 0
Skewness of Winter-> 0.760, right skewed because the metric is > 0
```



- Check Presence of Outliers

```
print(f"Kurtosis of Spring-> {Spring.kurt() :.3f},  for k < 3, it is called a
Platykurtic distribution (shows lack of outliers)")
print(f"Kurtosis of Summer-> {Summer.kurt() :.3f}, for k < 3, it is called a
Platykurtic distribution (shows lack of outliers)")
print(f"Kurtosis of Fall  -> {Fall.kurt() :.3f}, for k < 3, it is called a
Platykurtic distribution (shows lack of outliers)")
print(f"Kurtosis of Winter-> {Winter.kurt() :.3f}, for k < 3, it is called a
Platykurtic distribution (shows lack of outliers)")
```

Results:

```
Kurtosis of Spring-> 2.770,  for k < 3, it is called a Platykurtic distribution (shows lack of outliers)
Kurtosis of Summer-> -0.665, for k < 3, it is called a Platykurtic distribution (shows lack of outliers)
Kurtosis of Fall  -> -0.853, for k < 3, it is called a Platykurtic distribution (shows lack of outliers)
Kurtosis of Winter-> -0.372, for k < 3, it is called a Platykurtic distribution (shows lack of outliers)
```

- Set a significance level and calculate the test Statistics / p-value.

If the assumptions of normality or equal variances are violated, consider using a non-parametric alternative, such as the Kruskal-Wallis H test, which does not assume normality or equal variances. In this case, test of normality and equal variance have failed, hence we proceed with Kruskal test

At a 5% significance level, can we conclude all 4 seasons have different population means?

Kruskal:

```
kruskal(Spring, Summer, Fall, Winter)
```

Results: KruskalResult(statistic=690.4515233888959, pvalue=2.4688288437668016e-149)

Insights: P-value is very low. Hence, we reject the Null Hypothesis.

One Way ANOVA:

If we continue doing the analysis using one way ANOVA, even if some assumptions fail (Levene's test or Shapiro-wilk test), the test results look like this:

```
f_oneway(Spring, Summer, Fall, Winter)
```

Results: F_onewayResult(statistic=247.7072540561225, pvalue=1.690591355211833e-155)

Insights: **P-value is as low as 2.4688288437668016e-149 and 1.690591355211833e-155 for both Kruskal and ANOVA test respectively. This is lower than the 0.05 significance level. Hence, we reject the Null Hypothesis. All 4 samples come from different population. Demand of bicycles on rent is different for different Seasons. *Seasons have an effect on the number of electric cycles rented***

Test 3: Effect of weather on the demand of electric cycles rented

	weather	count
0	Clear+Few clouds	7192
1	Mist+Cloudy	2834
2	Light Snow+Light Rain	859
3	Heavy Rain	1

- First, filter data based on different categories within Working day Column to create samples

```
Clear = clipped_data[clipped_data["weather"] == "Clear+Few clouds"] ["count"]
Mist_Cloudy = clipped_data[clipped_data["weather"] == "Mist+Cloudy"] ["count"]
Light_Snow_Rain = clipped_data[clipped_data["weather"] == "Light Snow+Light
Rain"]["count"]
Heavy_Rain = clipped_data[clipped_data["weather"] == "Heavy Rain"] ["count"]
```

- Formulate Null Hypothesis (H0) and Alternate Hypothesis (H1)

Null Hypothesis (Ho): All 4 samples have identical population means. Number of rented cycles is INDEPENDENT of weather (mu1 = mu2 = mu3 = mu4)

Alternate Hypothesis (Ha): At least one group has a different population mean. Number of rented cycles is DEPENDENT on weather

- Check assumptions of the test

Test of Normality – Shapiro Test – Check if all samples were drawn from a normal distribution

```
print(shapiro(Clear))
print(shapiro(Mist_Cloudy))
print(shapiro(Light_Snow_Rain))
```

Results:

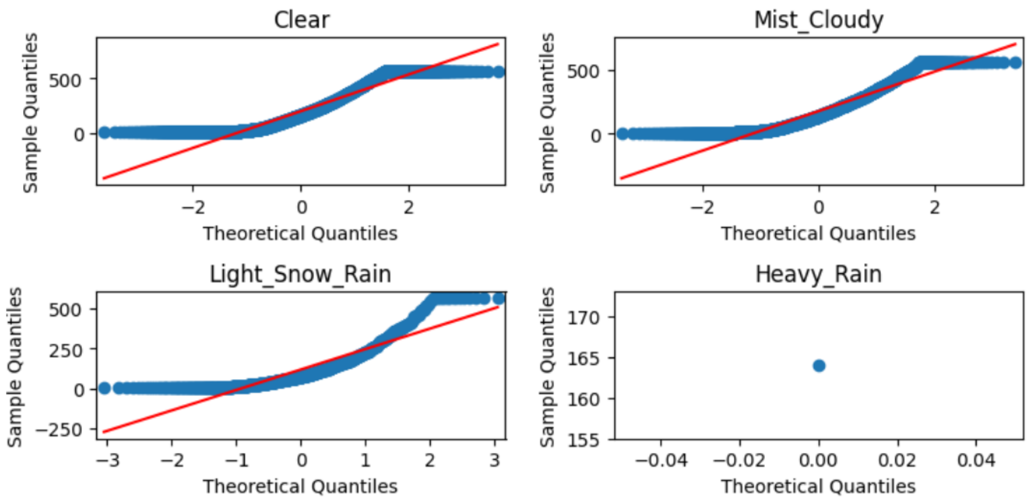
```
ShapiroResult(statistic=0.9005734320255468, pvalue=7.899821142408151e-56)
ShapiroResult(statistic=0.8921240894731988, pvalue=1.018044035961073e-40)
ShapiroResult(statistic=0.7977795250999241, pvalue=2.2842706810516476e-31)
```

P-value is very low for all 4 samples. Hence, can conclude that none of them are normally distributed. Since, data point for sample - Heavy Rain is only 1, we cannot perform Shapiro test on it.

QQ plot for visualisation of this normality

```
plt.figure(figsize=(8,4))

plt.subplot(2, 2, 1)
qqplot(Clear, line='s', ax=plt.gca())
plt.title('Clear')

plt.subplot(2, 2, 2)
qqplot(Mist_Cloudy, line='s', ax=plt.gca())
plt.title('Mist_Cloudy')
plt.subplot(2, 2, 3)
```

```
qqplot(Light_Snow_Rain, line='s', ax=plt.gca())
plt.title('Light_Snow_Rain')

plt.subplot(2, 2, 4)
qqplot(Heavy_Rain, line='s', ax=plt.gca())
plt.title('Heavy_Rain')

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



Insights: All the quantile points would lie along the red line if data was normally distributed. There are major deviations the plots showing they are not normally distributed.

Test of Equal Variances – Levene – Test to check if all input samples are from populations with equal variances

```
levene(Clear, Mist_Cloudy, Light_Snow_Rain, Heavy_Rain
```

Results: LeveneResult(statistic=63.54916328885072, pvalue=1.0010568660842785e-40)

P-value is as low as 1.0010568660842785e-40. Hence there is no equal variances.

- Check Skewness

```
print(f"Skewness of Clear -> {Clear.skew() :.3f}, right skewed because the metric is > 0")
print(f"Skewness of Mist_Cloudy -> {Mist_Cloudy.skew() :.3f}, right skewed because the metric is > 0")
print(f"Skewness of Light_Snow_Rain -> {Light_Snow_Rain.skew() :.3f}, right skewed because the metric is > 0")
print(f"Skewness of Heavy_Rain-> {Heavy_Rain.skew() :.3f}, only one datapoint available")
```

Results:

```
Skewness of Clear           -> 0.737, right skewed because the metric is > 0
Skewness of Mist_Cloudy     -> 0.930, right skewed because the metric is > 0
Skewness of Light_Snow_Rain -> 1.702, right skewed because the metric is > 0
Skewness of Heavy_Rain      -> nan, only one datapoint available
```

- Check Presence of Outliers

```
print(f"Kurtosis of Clear ->{Clear.kurt() :.3f}, for k < 3, it is called a Platykurtic distribution (shows lack of outliers)")
print(f"Kurtosis of Mist_Cloudy -> {Mist_Cloudy.kurt() :.3f}, for k < 3, it is called a Platykurtic distribution (shows lack of outliers)")
print(f"Kurtosis of Light_Snow_Rain -> {Light_Snow_Rain .kurt() :.3f}, for k < 3, it is called a Platykurtic distribution (shows lack of outliers)")
print(f"Kurtosis of Heavy_Rain -> {Heavy_Rain.kurt() :.3f}, only one datapoint available")
```

Results:

```
Kurtosis of Clear           ->-0.525, for k < 3, it is Platykurtic distribution (shows lack of outliers)
Kurtosis of Mist_Cloudy     -> 0.012, for k < 3, it is Platykurtic distribution (shows lack of outliers)
Kurtosis of Light_Snow_Rain -> 2.652, for k < 3, it is Platykurtic distribution (shows lack of outliers)
Kurtosis of Heavy_Rain      -> nan, only one datapoint available
```

- Set a significance level and calculate the test Statistics / p-value.

If the assumptions of normality or equal variances are violated, consider using a non-parametric alternative, such as the Kruskal-Wallis H test, which does not assume normality or equal variances. In this case, test of normality and equal variance have failed, hence we proceed with Kruskal test

At a 5% significance level, can we conclude all 4 weathers have different population means?

Kruskal:

```
kruskal(Clear, Mist_Cloudy, Light_Snow_Rain, Heavy_Rain)
```

Results: KruskalResult(statistic=204.7853967605586, pvalue=3.900417263983396e-44)

Insights: P-value is very low. Hence, we reject the Null Hypothesis.

One Way ANOVA:

If we continue doing the analysis using one way ANOVA, even if some assumptions fail (Levene's test or Shapiro-wilk test), the test results look like this:

```
f_oneway(Clear, Mist_Cloudy, Light_Snow_Rain, Heavy_Rain)
```

Results: F_onewayResult(statistic=70.16727781577517, pvalue=6.138757242589214e-45)

Insights: **P-value is as low as 3.900417263983396e-44 and 6.138757242589214e-45 for both Kruskal and ANOVA test respectively. This is lower than the 0.05 significance level. Hence, we reject the Null Hypothesis. All 4 samples come from different population. Demand of bicycles on rent is different for different weather**. ***Weather has an effect on the number of electric cycles rented***

Test 4: Check if Weather is dependent on season

- Formulate Null Hypothesis (H0) and Alternate Hypothesis (H1)

Null Hypothesis (Ho): Weather and Season are independent / Weather is not associated with seasons

Alternate Hypothesis (Ha): Weather and Season are dependent / Weather is associated with seasons

- Create a Contingency Table or Crosstab against 'Weather' & 'Season' columns

```
season_vs_weather = pd.crosstab(clipped_data["season"], clipped_data["weather"])
```

weather	Clear+Few clouds	Heavy Rain	Light Snow+Light Rain	Mist+Cloudy
season				
Fall	1930	0	199	604
Spring	1759	1	211	715
Summer	1801	0	224	708
Winter	1702	0	225	807

- Set a significance level and calculate the test Statistics / p-value.

At a 5% significance level, can you conclude that Weather is dependent on season using chi-square test of independence?

```
chi2_contingency(season_vs_weather)
```

Results:

```
Chi2ContingencyResult(statistic=49.158655596893624, pvalue=1.549925073686492e-07, dof=9,
7.11493845e+02],
       [1.77454639e+03, 2.46738931e-01, 2.11948742e+02, 6.99258130e+02],
       [1.80559765e+03, 2.51056403e-01, 2.15657450e+02, 7.11493845e+02],
       [1.80625831e+03, 2.51148264e-01, 2.15736359e+02, 7.11754180e+02]]))
```

Insights: **P-value of 1.549925073686492e-07 is lower than 0.05 significance level. Hence, we reject the Null Hypothesis. *Weather and Season are dependent / Weather is associated with seasons.***

Actionable Business Recommendations

To help business regain profitability based on the analysis of factors affecting the demand for shared electric cycles, here are some insights and recommendations:

- **Diversify Revenue Streams Based on Weather and Seasonality**:
  - Since both weather and season have a significant impact on demand, the service provider could consider introducing seasonal pricing or promotions to encourage rentals during low-demand periods (e.g., heavy rain or winter). This could involve discounts or special offers on rainy days to encourage ridership.
- **Diversify Revenue Streams Based on Weather and Seasonality:**
  - Since both weather and season have a significant impact on demand, business could consider introducing seasonal pricing or promotions to encourage rentals during low-demand periods (e.g., heavy rain or winter). This could involve discounts or special offers on rainy days to encourage ridership.
- **Enhanced Marketing During Favourable Weather**:
  - Highlight the convenience and eco-friendliness of using bikes from this company on pleasant days attracting more riders and boosting rides during favourable climate.
- **Weather-proofing Bikes:**
  - Ensure that the company's bikes are equipped with weather-resistant features (e.g., rain covers, anti-slip tires) to make them more appealing during light rain or misty conditions. This could help mitigate the drop in demand during these weather conditions.
- **Weather is Dependent on Season:**
  - Since weather patterns are dependent on seasons, the strategy needs to consider both factors simultaneously for better demand forecasting.
- **Working Days vs. Weekends:**
  - The demand does not differ significantly between working days and weekends, indicating consistent usage patterns across the week.
- **Enhance User Experience:**
  - Improve the app experience by providing real-time weather updates and suggesting optimal riding times based on current weather and traffic conditions.