YULU - Hypothesis Testing

Business Problem:

> Yulu, a popular micro-mobility service in India, has seen a decline in revenue recently. They've hired a consulting firm to figure out why. Essentially, they want to know what influences people to use their shared electric cycles. This means looking at factors like where people live, work, and travel, as well as what makes them choose Yulu over other options. By understanding these factors, Yulu hopes to boost demand and regain its momentum in the market.

In [177]:
```
#Importing all the required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy
import warnings
warnings.filterwarnings('ignore')
```

In [178]:
```
df = pd.read_csv('yulu.csv')  #import dataset using pandas
```

In [179]:
```
df.sample(5)    # sample of dataset
```

Out[179]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	cas
7010	2012-04-10 10:00:00	2	0	1	1	18.86	22.725	47	11.0014	
6127	2012-02-11 13:00:00	1	0	0	2	10.66	12.120	81	19.0012	
3805	2011-09-09 13:00:00	3	0	1	1	28.70	33.335	79	6.0032	
10185	2012-11-09 19:00:00	4	0	1	1	13.94	17.425	71	6.0032	
361	2011-01-16 13:00:00	1	0	0	1	10.66	11.365	35	19.9995	

In [180]:
```
df.shape    #there are 10886 rows and 12 columns
```

Out[180]:
```
(10886, 12)
```

In [181]:
```
df.info()   #overall characteristics of a dataset
```

```
<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
```

In [182]:
```
df.describe() #descriptive stats for numerical column
```

Out[182]:

	season	holiday	workingday	weather	temp	atemp	
count	10886.000000	10886.000000	10886.000000	10886.000000	10886.00000	10886.000000	1088
mean	2.506614	0.028569	0.680875	1.418427	20.23086	23.655084	6
std	1.116174	0.166599	0.466159	0.633839	7.79159	8.474601	1
min	1.000000	0.000000	0.000000	1.000000	0.82000	0.760000	
25%	2.000000	0.000000	0.000000	1.000000	13.94000	16.665000	4
50%	3.000000	0.000000	1.000000	1.000000	20.50000	24.240000	6
75%	4.000000	0.000000	1.000000	2.000000	26.24000	31.060000	7
max	4.000000	1.000000	1.000000	4.000000	41.00000	45.455000	10

In [114]:
```
df.isnull().sum()     #we can see there are no missing values in the dataset
```

Out[114]:
```
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
```

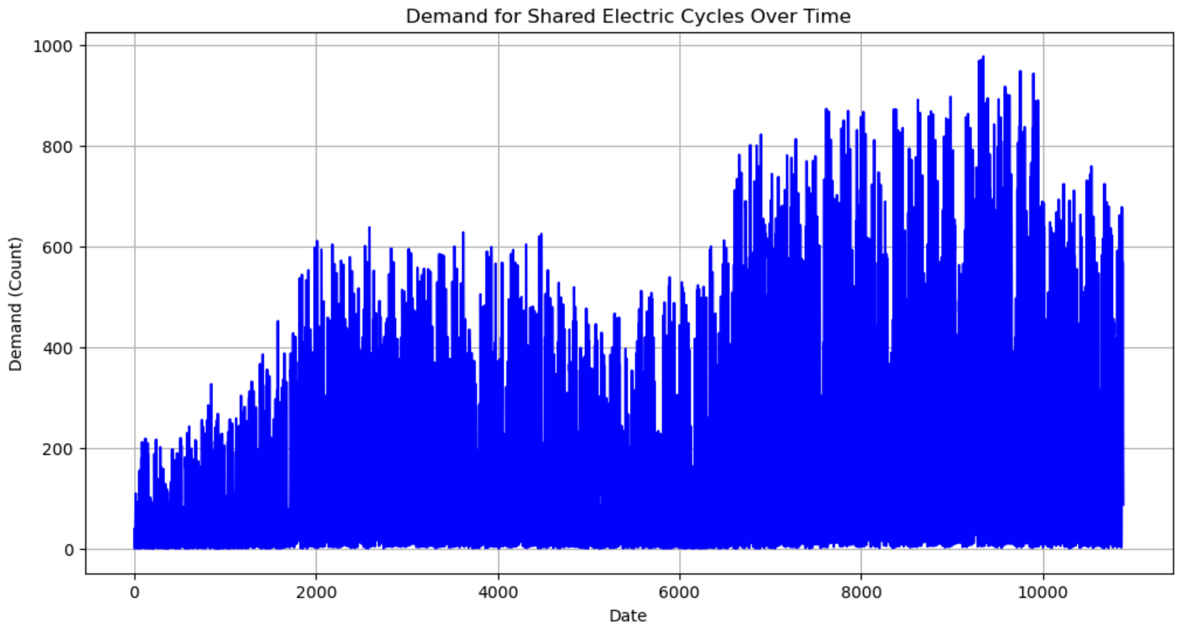
In [115]:
```
df.duplicated().value_counts()   #there are no duplicated records
```

Out[115]:
```
False    10886
dtype: int64
```

In [116]:
```
#We need to convert the datetime column to datetime datatype
df['datetime'] = pd.to_datetime(df['datetime'])
df.info()
```

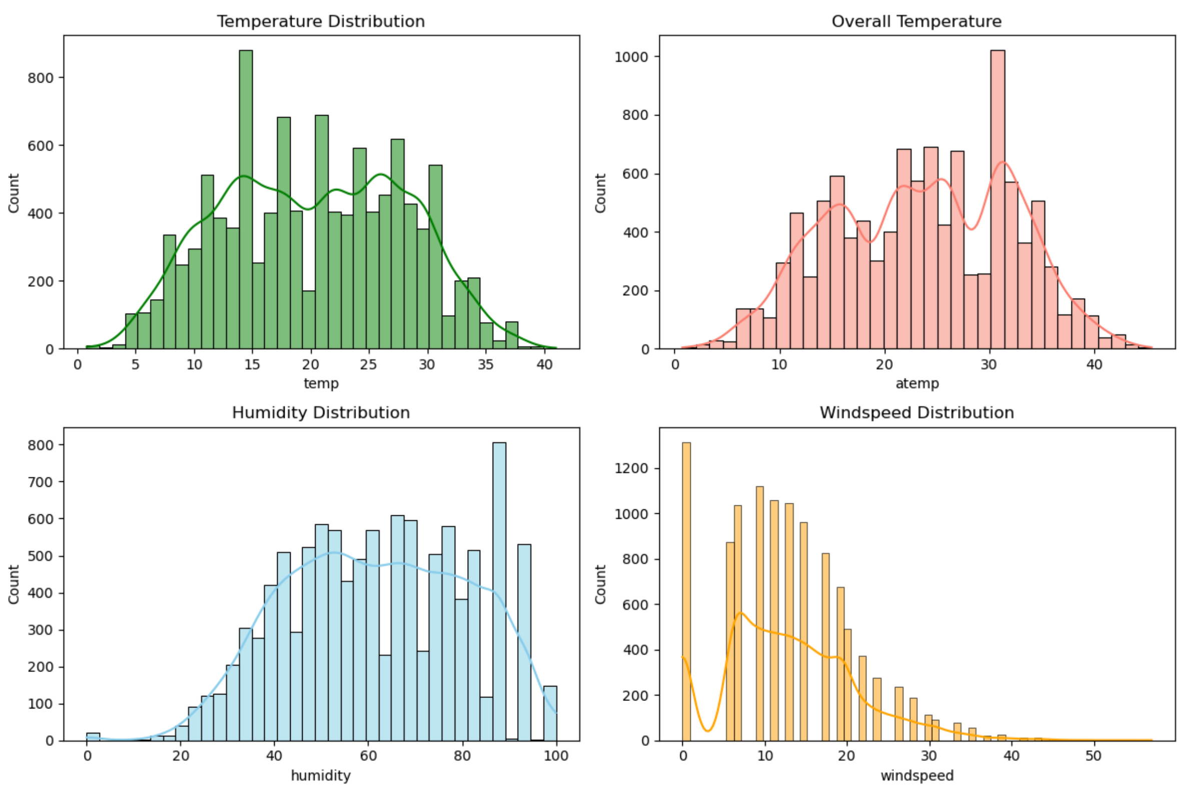
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   datetime    10886 non-null  datetime64[ns]
 1   season      10886 non-null  int64
 2   holiday     10886 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: datetime64[ns](1), float64(3), int64(8)
memory usage: 1020.7 KB
```

In [183]:
```
df.set_index('datetime')

# Plot the demand for shared electric cycles over time
plt.figure(figsize=(12, 6))
plt.plot(df['count'], color='blue', linestyle='-')
plt.title('Demand for Shared Electric Cycles Over Time')
plt.xlabel('Date')
plt.ylabel('Demand (Count)')
plt.grid(True)
plt.show()
```



In [117]:

```
# histograms for numerical variables
plt.figure(figsize=(12,8))
plt.subplot(2,2,1)
sns.histplot(data=df,x='temp',kde=True,color='green')
plt.title("Temperature Distribution")

plt.subplot(2,2,2)
sns.histplot(data=df,x='atemp',color='salmon',kde=True)
plt.title('Overall Temperature')

plt.subplot(2,2,3)
sns.histplot(data=df,x='humidity',kde=True,color='skyblue')
plt.title('Humidity Distribution')

plt.subplot(2,2,4)
sns.histplot(data=df,x='windspeed',kde=True,color='orange')
plt.title('Windspeed Distribution')

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



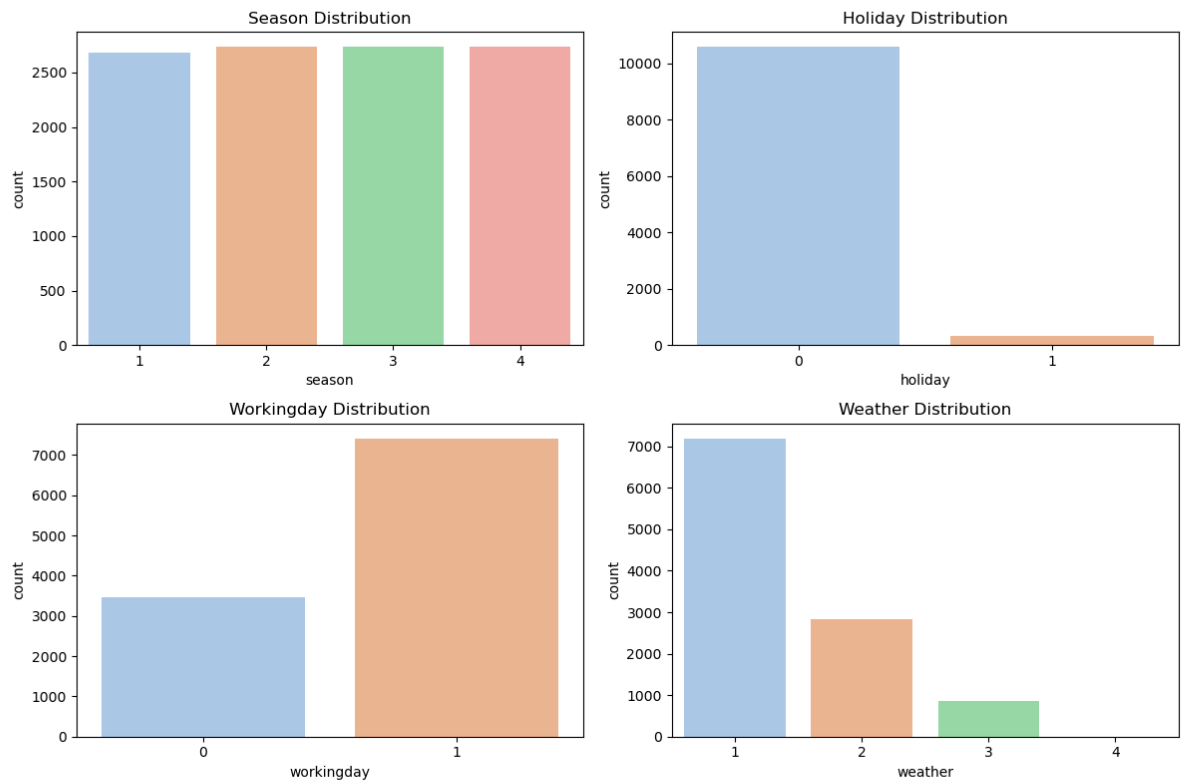
Humidity (Left-Skewed):

Left-skewed distribution suggests that most of the data points are concentrated on the higher end of the humidity scale.

Windspeed (Right-Skewed):

Right-skewed distribution suggests that most of the data points are concentrated on the lower end of the windspeed scale.

In [118]:
```
# Countplot For Categorical Variables
plt.figure(figsize=(12,8))
plt.subplot(2,2,1)
sns.countplot(data=df,x='season',palette='pastel')
plt.title("Season Distribution")

plt.subplot(2,2,2)
sns.countplot(data=df,x='holiday',palette='pastel')
plt.title('Holiday Distribution')

plt.subplot(2,2,3)
sns.countplot(data=df,x='workingday',palette='pastel')
plt.title('Workingday Distribution')

plt.subplot(2,2,4)
sns.countplot(data=df,x='weather',palette='pastel')
plt.title("Weather Distribution")

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



The countplot for the 'season' variable shows that the distribution of counts across different seasons appears to be relatively similar.

The countplot for the 'weather' variable indicates that weather condition

1. '1' (Clear, Few clouds, partly cloudy) has the highest count, followed by weather condition
2. '2' (Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist), and then weather condition
3. '3' (Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds).

4. '4' (Heavy Rain + Ice Pellets + Thunderstorm + Mist, Snow + Fog) has almost no counts, suggesting that such extreme weather conditions are less conducive to cycling and, therefore, result in fewer rentals.

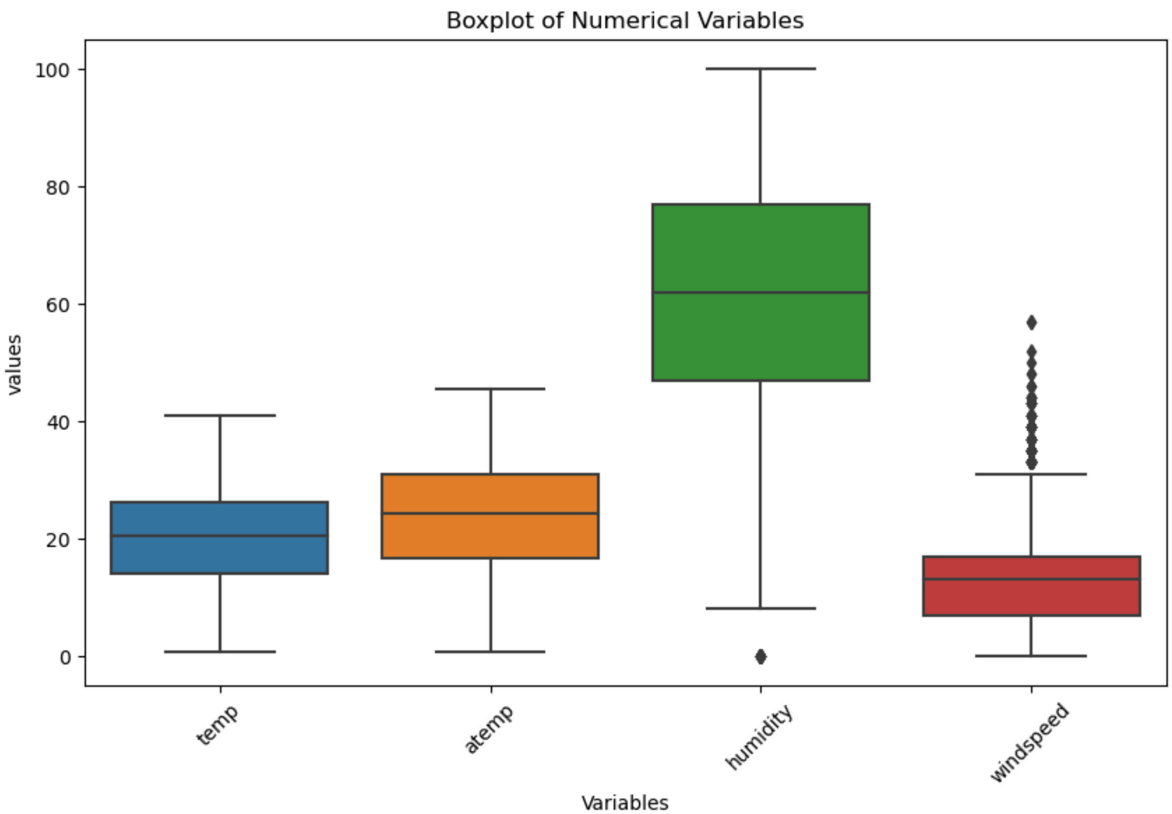
In [119]:
```
df['workingday'].value_counts() #there are 7412 working days and 3474 non work
```

Out[119]:
```
1    7412
0    3474
Name: workingday, dtype: int64
```

In [120]:
```
df['season'].value_counts()
```

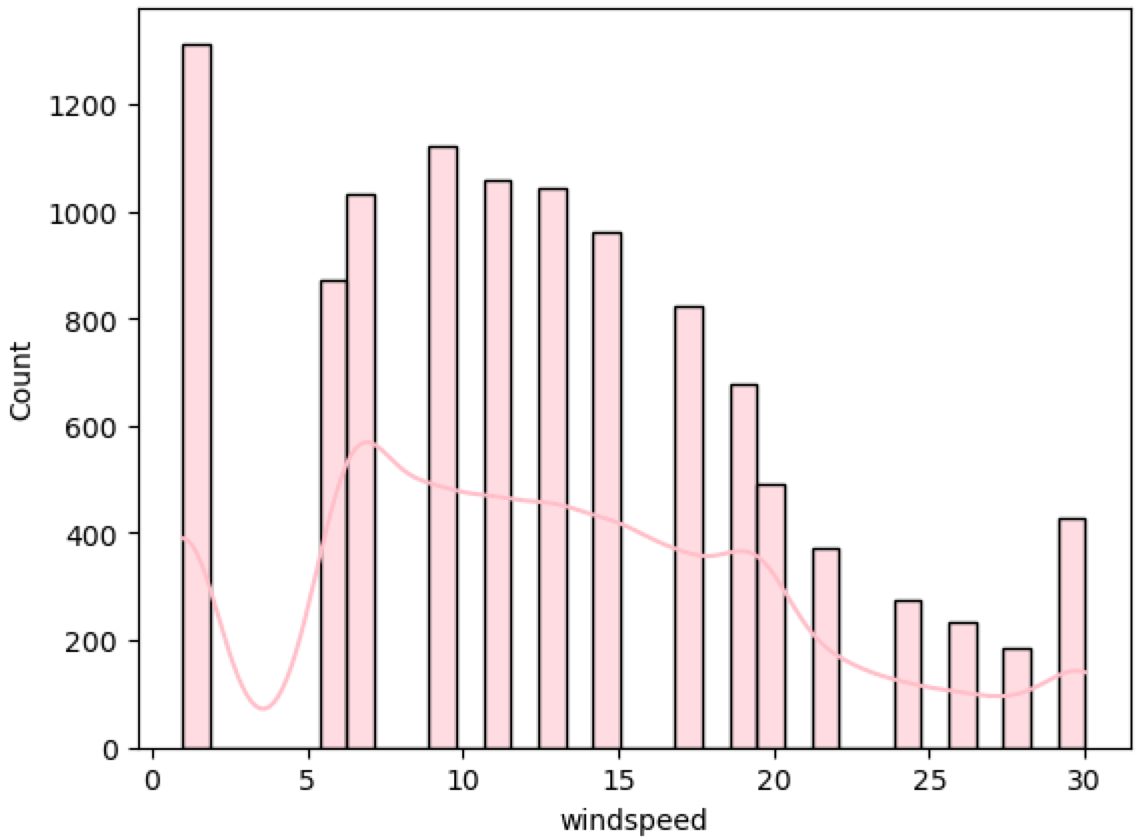
Out[120]:
```
4    2734
2    2733
3    2733
1    2686
Name: season, dtype: int64
```

In [121]:
```
numerical_vars=['temp','atemp','humidity','windspeed']

plt.figure(figsize=(10,6))
sns.boxplot(data=df[numerical_vars])
plt.title('Boxplot of Numerical Variables')
plt.xlabel('Variables')
plt.ylabel('values')
plt.xticks(rotation=45)
plt.show()
```

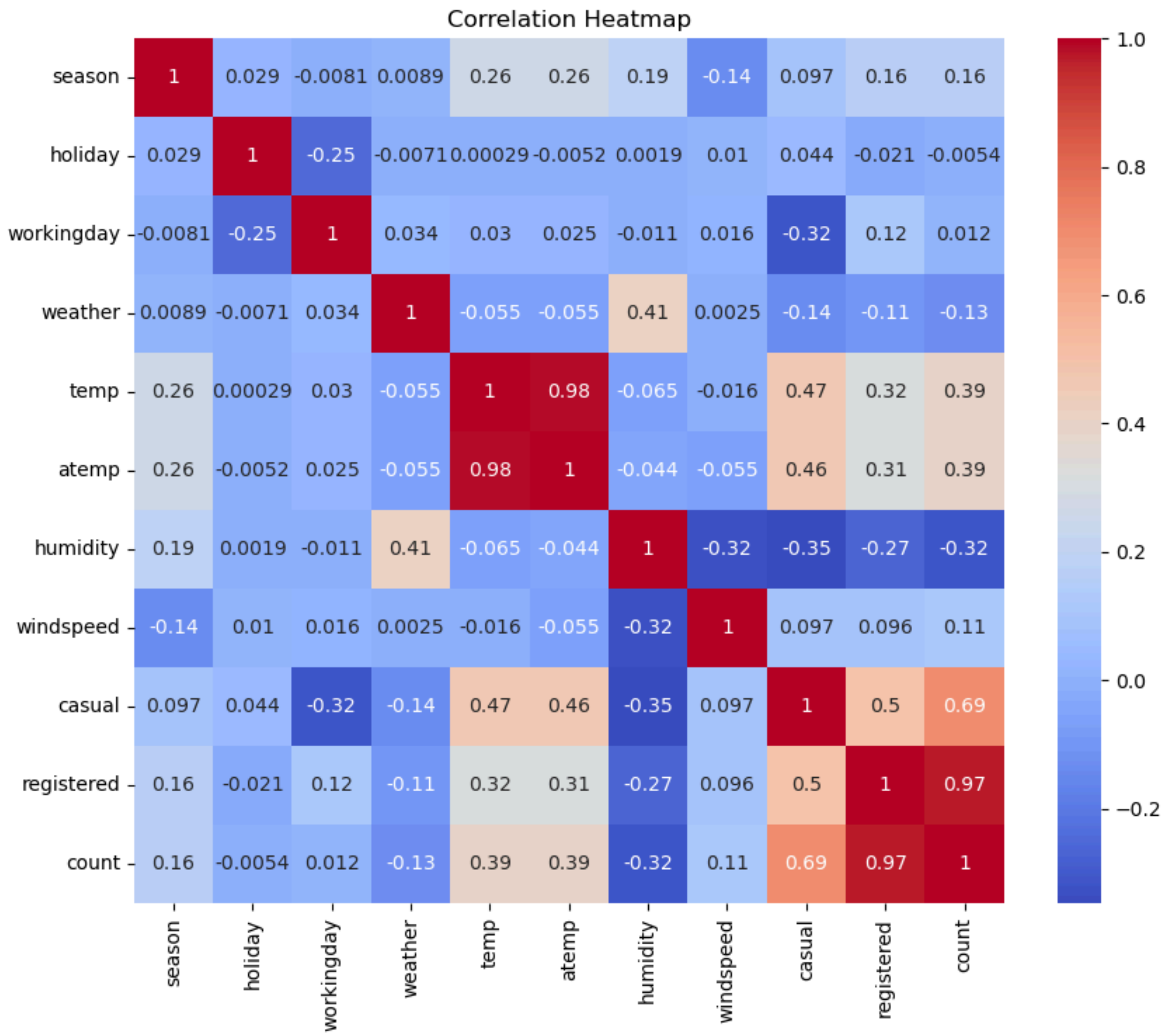


The windspeed variable in the dataset contains outliers, indicating potential extreme values that deviate from the majority of data points. (But should not be ignored)

In [122]:
```
lower_bound=1
upper_bound=30

df['windspeed'] = df['windspeed'].clip(lower=lower_bound,upper=upper_bound)

#Rechecking
plt.figure()
sns.histplot(data=df,x='windspeed',kde=True,color='pink')
plt.show()
```



In [123]:
```
#Correlation between Data Points
correlation_matrix =df.corr()
plt.figure(figsize=(10,8))
sns.heatmap(correlation_matrix,annot=True,cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



In [124]:
```
# we came to know that temp & atemp are highly corelated and then Registered &
df=df.drop(columns=['atemp','registered'])
```

In [125]:
```
df.sample(5)
```

Out[125]:

	datetime	season	holiday	workingday	weather	temp	humidity	windspeed	casual	co
5830	2012-01-18 03:00:00	1	0	1	2	18.04	51	30.0000	1	
9155	2012-09-04 20:00:00	3	0	1	1	30.34	70	16.9979	47	:
7988	2012-06-13 05:00:00	2	0	1	1	24.60	83	12.9980	2	
10151	2012-11-08 09:00:00	4	0	1	1	13.12	31	23.9994	22	2
4533	2011-11-01 23:00:00	4	0	1	1	13.94	87	1.0000	11	

Is there any significant difference between the no. of bike rides on Weekdays and Weekends?

Formulate Null Hypothesis and Alternate Hypothesis

H0(Null Hypothesis):There is no significant difference in the number of bike rides between weekdays and weekends.

HA(Alternate Hypothesis):There is a significant difference in the number of bike rides between weekdays and weekends.

In [133]:
```
from scipy.stats import ttest_ind
weekdays=df[df['workingday']==1]['count']
weekends=df[df['workingday']==0]['count']

t_stats,p_val =ttest_ind(weekdays,weekends)
print("test statistics",t_stats)
print("p_value :",p_val)
```

```
test statistics 1.2096277376026694
p_value : 0.22644804226361348
```

In [136]:
```
alpha=0.05
if p_val <= alpha:
    print("reject the null hypothesis Their is signifiant difference in the nu
else:
    print("fail to reject the null hypothesis. There is no significant differe
```

```
fail to reject the null hypothesis. There is no significant difference in the
number of bike rides between weekdays and weekends.
```

1. This implies that there is no significant difference in the number of bike rides between weekdays and weekends.
2. The analysis suggests that the demand for bike rides remains relatively consistent between weekdays and weekends.
3. Yulu may not need to adjust its operational strategies significantly based on whether it's a weekday or weekend.
4. Yulu can focus on expanding its services and coverage areas to meet the consistent demand observed across all days of the week.

Is their demand of bicycles on rent is the same for different Weather conditions?

Formulate the null hypothesis and alternative hypothesis

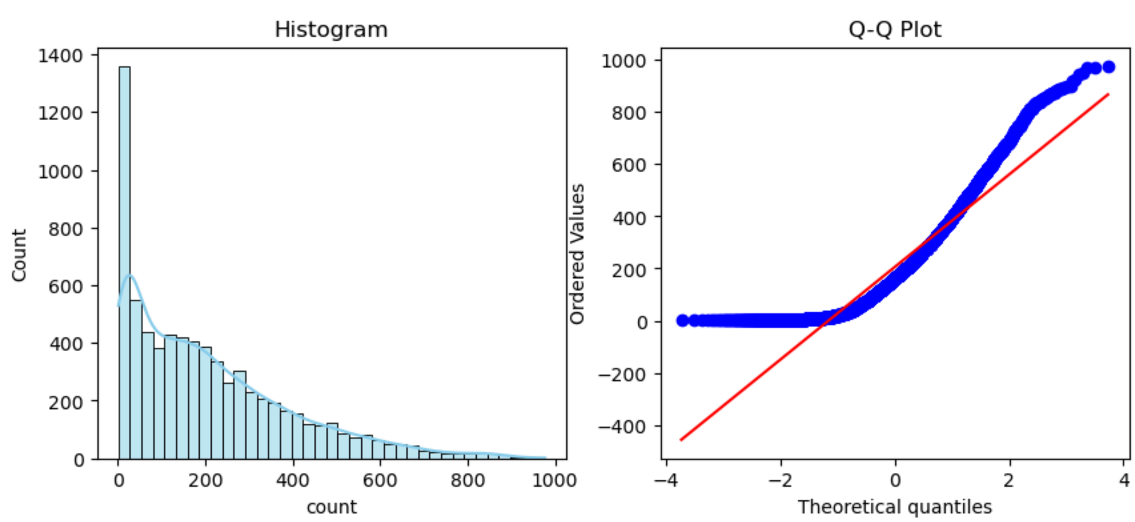
Null Hypothesis H0: There is no difference in the demand for bicycles on rent across different weather

Alternate Hypothesis HA: There is significant difference in the demand for bicycles on rent across different weather conditions.

> We need to use annova here but anova has assumption of normality and equality of variance, using qq plot shapiro test we check normality and for equality of variance we use levene test.

In [156]:
```
from scipy import stats

grouped_data = df.groupby('weather')['count']

for weather, data in grouped_data:
    print(f"--- Weather Condition: {weather} ---")

    # distribution using histogram and Q-Q plot
    plt.figure(figsize=(10, 4))
    plt.subplot(1, 2, 1)
    sns.histplot(data, kde=True, color='skyblue')
    plt.title('Histogram')
    plt.subplot(1, 2, 2)
    stats.probplot(data, dist="norm", plot=plt)
    plt.title('Q-Q Plot')
    plt.show()

    # Calculate skewness and kurtosis
    skewness = stats.skew(data)
    kurtosis = stats.kurtosis(data)
    print(f"Skewness: {skewness}")
    print(f"Kurtosis: {kurtosis}")

    # Shapiro-Wilk's test for normality
    if len(data) >= 3:
        shapiro_test = stats.shapiro(data)
        print(f"Shapiro-Wilk's Test p-value: {shapiro_test[1]}")

        # Print normality assessment
        if shapiro_test[1] > 0.05:
            print("Data is normally distributed.")
        else:
            print("Data is not normally distributed.")
    else:
        print("Insufficient data points for Shapiro-Wilk's test.")
    print("\n")
```

```
--- Weather Condition: 1 ---
```



In [152]:
```
from scipy.stats import levene
weather_groups = [data for weather, data in grouped_data]

# Perform Levene's test for equality of variances
levene_test = levene(*weather_groups)

# Print the test statistic and p-value
print("Levene's Test Statistic:", levene_test.statistic)
print("Levene's Test p-value:", levene_test.pvalue)

# Interpret the results
if levene_test.pvalue > 0.05:
    print("The variances are approximately equal across different weather cond
else:
    print("The variances are not equal across different weather conditions.")
```

```
Levene's Test Statistic: 54.85106195954556
Levene's Test p-value: 3.504937946833238e-35
The variances are not equal across different weather conditions.
```

Assumptions are not true still we will check for anova test

In [158]:
```
#1. H0: There is no difference in the demand for bicycles on rent
#across different weather
#2. HA: There is significant difference in the demand for
#bicycles on rent across different weather conditions.

from scipy.stats import f_oneway

# data for each weather condition
weather_groups = [data for weather, data in grouped_data]

anova_result = f_oneway(*weather_groups)

print("ANOVA Test Statistic:", anova_result.statistic)
print("ANOVA Test p-value:", anova_result.pvalue)

# results
if anova_result.pvalue < 0.05:
    print("Reject the null hypothesis, There is a significant difference in th
else:
    print("Fail to reject the null hypothesis. There is no significant differe
```

```
ANOVA Test Statistic: 65.53024112793271
ANOVA Test p-value: 5.482069475935669e-42
Reject the null hypothesis, There is a significant difference in the demand o
f bicycles on rent for different weather conditions.
```

Is their demand of bicycles on rent is the same for different Seasons?

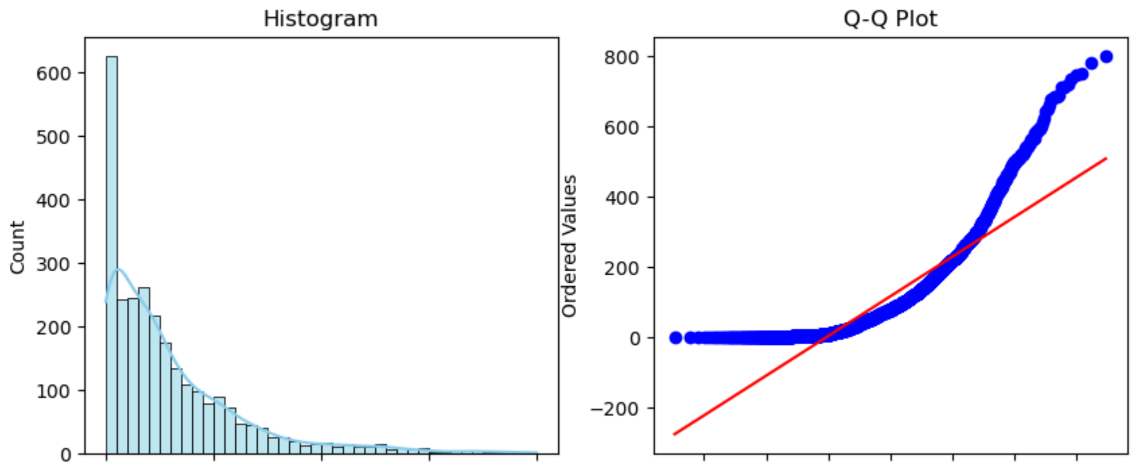
Formulate Null Hypothesis (H0) and Alternate Hypothesis (HA)

Null Hypothesis (H0): The demand for bicycles on rent is the same across different seasons.

Alternate Hypothesis (HA): The demand for bicycles on rent is different across different seasons.

In [159]:
```
print('''Null Hypothesis (H0): The demand for bicycles on rent is the same acr
Alternate Hypothesis (H1): The demand for bicycles on rent is different across

from scipy.stats import shapiro, levene, f_oneway

# data for each season
season_groups = [data for season, data in df.groupby('season')['count']]

for season, data in zip(range(1, 5), season_groups):
    print(f"--- Season: {season} ---")

    # Visualize distribution using histogram and Q-Q plot
    plt.figure(figsize=(10, 4))
    plt.subplot(1, 2, 1)
    sns.histplot(data, kde=True, color='skyblue')
    plt.title('Histogram')
    plt.subplot(1, 2, 2)
    stats.probplot(data, dist="norm", plot=plt)
    plt.title('Q-Q Plot')
    plt.show()

    #skewness and kurtosis
    skewness = stats.skew(data)
    kurtosis = stats.kurtosis(data)
    print(f"Skewness: {skewness}")
    print(f"Kurtosis: {kurtosis}")

    # Shapiro-Wilk's test for normality
    shapiro_test = shapiro(data)
    print(f"Shapiro-Wilk's Test p-value: {shapiro_test[1]}")
    if shapiro_test[1] > 0.05:
        print("Data appears to be normally distributed.")
    else:
        print("Data does not appear to be normally distributed.")

    # Levene's test for equality of variance
    if season != 1:  # Skip Levene's test for the first season (no comparison
        levene_test = levene(season_groups[0], data)
        print(f"Levene's Test p-value: {levene_test.pvalue}")
        if levene_test.pvalue > 0.05:
            print("Variances are approximately equal across seasons.")
        else:
            print("Variances are not equal across seasons.")

# One-way ANOVA test
anova_result = f_oneway(*season_groups)
print("\n--- One-way ANOVA Test ---")
print("ANOVA Test Statistic:", anova_result.statistic)
print("ANOVA Test p-value:", anova_result.pvalue)

if anova_result.pvalue < 0.05:
    print("Reject the null hypothesis. There is a significant difference in th
else:
    print("Fail to reject the null hypothesis. There is no significant differe
```

```
Null Hypothesis (H0): The demand for bicycles on rent is the same across d
ifferent seasons.
Alternate Hypothesis (H1): The demand for bicycles on rent varies across d
ifferent seasons.
--- Season: 1 ---
```



Normality Assumption:

> For all seasons, the data does not appear to be normally distributed based on Shapiro-Wilk's test, as all p-values are less than 0.05.

Equality of Variance Assumption:

> Levene's test indicates that variances are not equal across seasons, as all p-values are less than 0.05.

ANOVA Test:

> The one-way ANOVA test statistic is 236.95 with a p-value of approximately 6.16e-149. Since the p-value is much less than the significance level (alpha=0.05), we reject the null hypothesis. Therefore, There is a significant difference in the demand of bicycles on rent across different seasons.

If the Weather conditions are significantly different during different Seasons?

Formulate Null Hypothesis (H0) and Alternate Hypothesis (HA)

> (H0): The distribution of weather conditions is the same across different seasons.

> (H1): The distribution of weather conditions varies across different seasons.

we can use the chi-square test for independence.

In [174]:
```
print('''(H0): The distribution of weather conditions is the same across diffe
(H1): The distribution of weather conditions varies across different seasons.
''')
from scipy.stats import chi2_contingency

# Create a contingency table (cross-tabulation) for 'Weather' and 'Season'
contingency_table = pd.crosstab(df['weather'], df['season'])
print(contingency_table )
# Perform chi-square test for independence
chi2, p_value, _, _ = chi2_contingency(contingency_table)

# Set the significance level (alpha)
alpha = 0.05

# Print the test statistic and p-value
print("\nChi-square Test Statistic:", chi2)
print("Chi-square Test p-value:", p_value)

# Decide whether to accept or reject the Null Hypothesis
if p_value < alpha:
    print("Reject the null hypothesis. The distribution of weather conditions
else:
    print("Fail to reject the null hypothesis. There is no significant differe
```

```
(H0): The distribution of weather conditions is the same across different sea
sons.
(H1): The distribution of weather conditions varies across different seasons.

season      1     2     3     4
weather
1        1759  1801  1930  1702
2         715   708   604   807
3         211   224   199   225
4           1     0     0     0

Chi-square Test Statistic: 49.15865559689363
Chi-square Test p-value: 1.5499250736864862e-07
Reject the null hypothesis. The distribution of weather conditions is signifi
cantly different during different seasons.
```

The p-value is much less than the significance level (alpha=0.05), we reject the null hypothesis. Therefore, we conclude that the distribution of weather conditions is significantly different during different seasons.

key conclusions:

Demand Factors: The analysis identified significant factors influencing the demand for shared electric cycles in the Indian market. These factors can include weather conditions, seasons, working days, and holidays.

Revenue Recovery Strategies: Understanding the factors affecting demand allows Yulu to make informed adjustments to their services and strategies to recover from recent revenue setbacks. They can tailor their offerings based on seasonal variations and other demand drivers.

Market Insights: The analysis provides valuable insights into the Indian market's micro-mobility landscape. Yulu can use this information to optimize their operations, expand into new areas, and target specific customer segments effectively.

Strategic Expansion: Yulu's decision to enter the Indian market aligns with their mission to provide sustainable commute solutions. With a deeper understanding of demand factors, they can strategically expand their presence and enhance their services to meet the evolving needs of commuters.

Data-Driven Decision Making: The case study demonstrates the importance of data-driven decision-making in addressing real-world business challenges. By leveraging data analytics techniques, Yulu can continuously optimize their operations and offerings to stay competitive and fulfill their mission of reducing traffic congestion.

Consulting Skills Development: Learners engaging with this case study can develop essential consulting skills by applying machine learning and statistical analysis techniques to solve complex business problems. This experience prepares them to provide valuable insights and recommendations to organizations facing similar challenges in various industries.

In [ ]: