yulu

May 16, 2024

[38]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
from scipy.stats import ttest_ind
from scipy.stats import chi2_contingency
import warnings
warnings.filterwarnings('ignore')
```

[16]:
```
df = pd.read_csv('bike_sharing.csv')
```

[17]:
```
df.head()
```

[17]:
```
              datetime  season  holiday  workingday  weather  temp   atemp  \
0  2011-01-01 00:00:00       1        0           0        1  9.84  14.395
1  2011-01-01 01:00:00       1        0           0        1  9.02  13.635
2  2011-01-01 02:00:00       1        0           0        1  9.02  13.635
3  2011-01-01 03:00:00       1        0           0        1  9.84  14.395
4  2011-01-01 04:00:00       1        0           0        1  9.84  14.395

   humidity  windspeed  casual  registered  count
0        81        0.0       3          13     16
1        80        0.0       8          32     40
2        80        0.0       5          27     32
3        75        0.0       3          10     13
4        75        0.0       0           1      1
```

[18]:
```
df.shape
```

[18]:
```
(10886, 12)
```

There are 10886 rows and 12 columns

[19]:
```
df.isna().sum()
```

[19]:
```
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 is no null values present in yulu dataframe

[20]:
```
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
```

[21]:
```
df.describe()
```

[21]:

	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
```

[22]:
```
df.duplicated().sum(axis=0)
```

[22]: 0

There is no duplicates present in entire rows

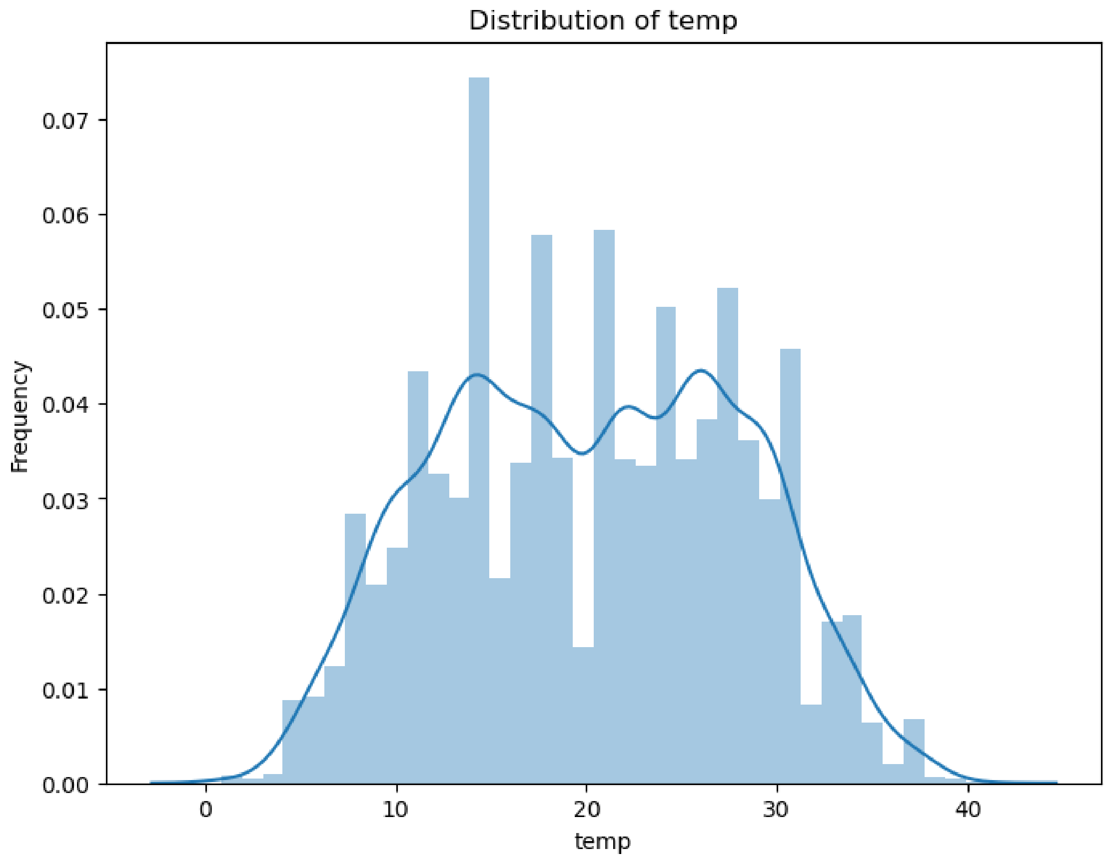
[23]:
```
df.duplicated().sum()
```

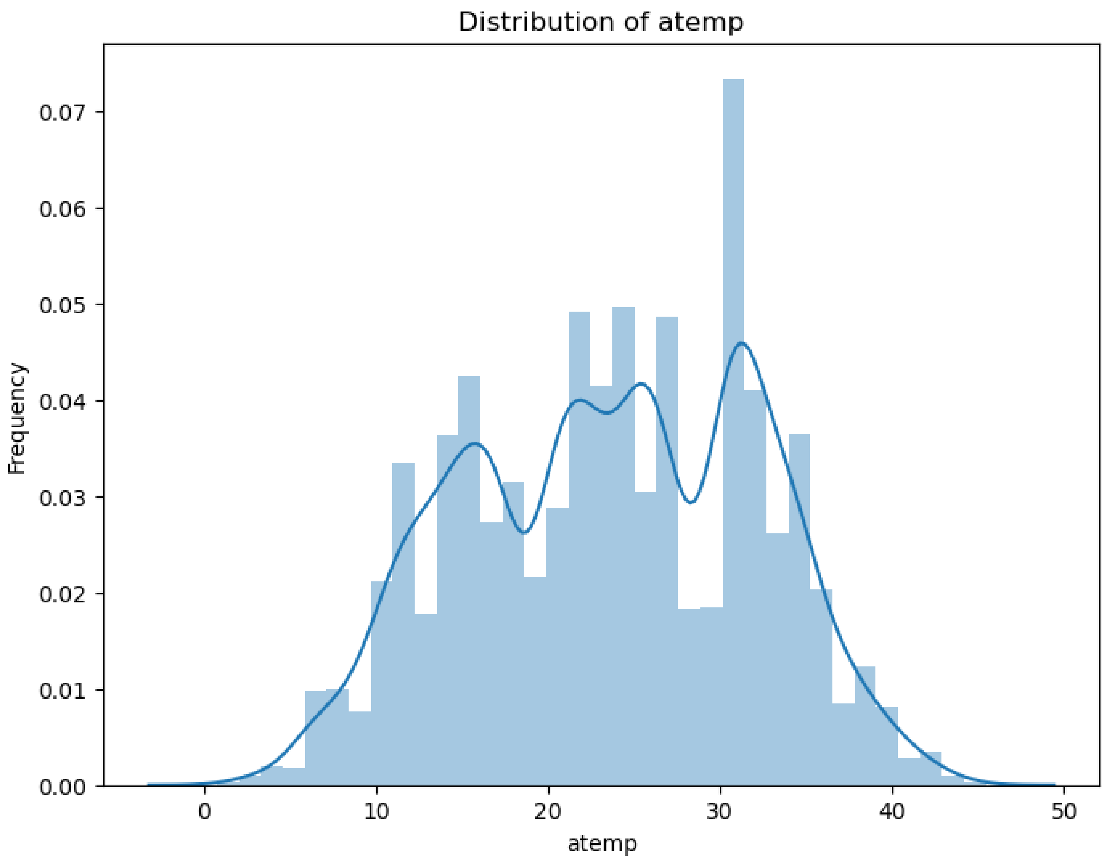
[23]: 0

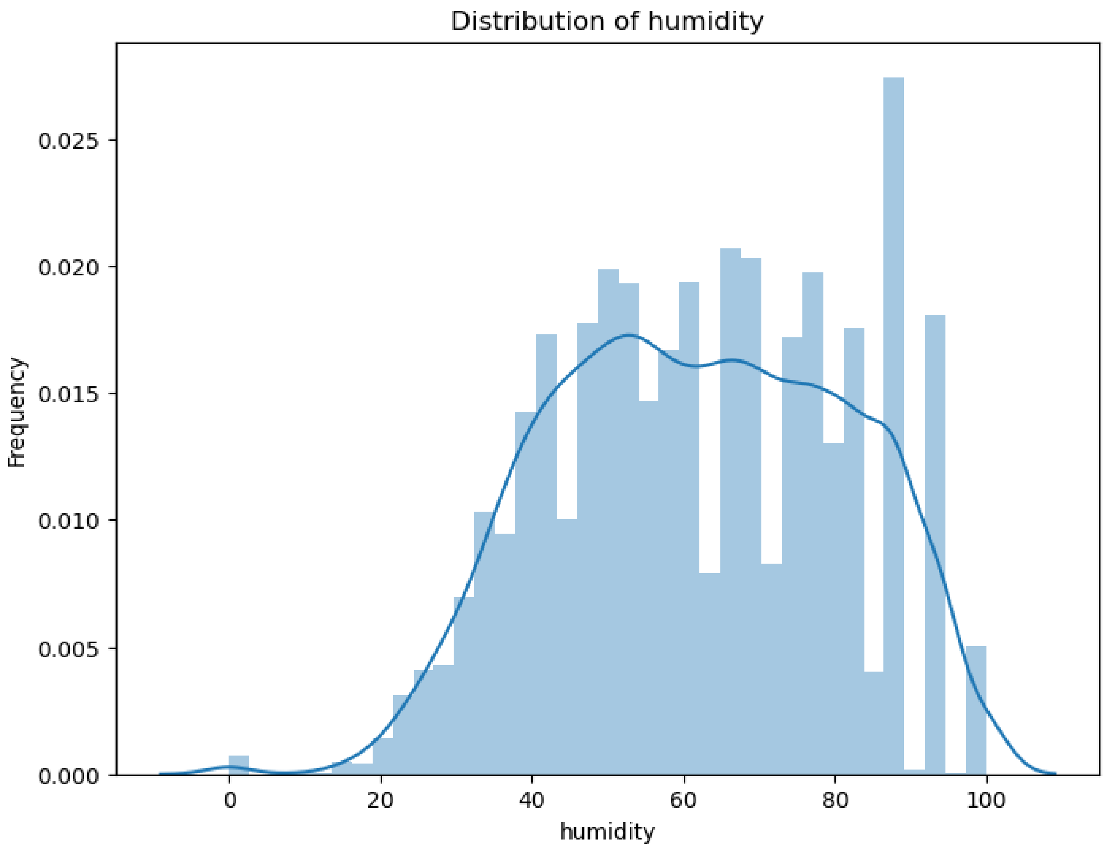
There is no duplicates present in entire columns

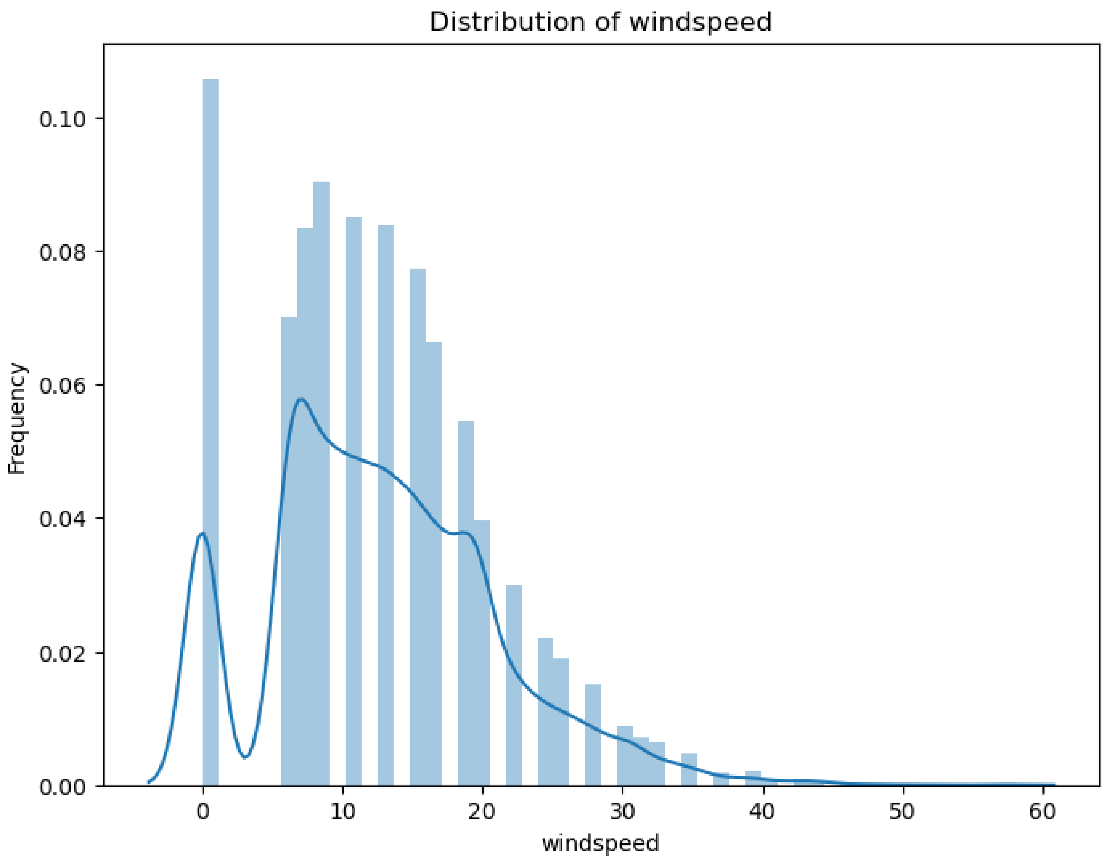
[44]:
```
numerical_vars = ['temp', 'atemp', 'humidity', 'windspeed']
for var in numerical_vars:
    plt.figure(figsize=(8, 6))
    sns.distplot(df[var], kde=True)
    plt.title(f'Distribution of {var}')
    plt.xlabel(var)
    plt.ylabel('Frequency')
    plt.show()

# Analyze the distribution of Categorical variables
categorical_vars = ['weather', 'season']
for var in categorical_vars:
    plt.figure(figsize=(8, 6))
    sns.countplot(x=var, data=df)
    plt.title(f'Distribution of {var}')
```

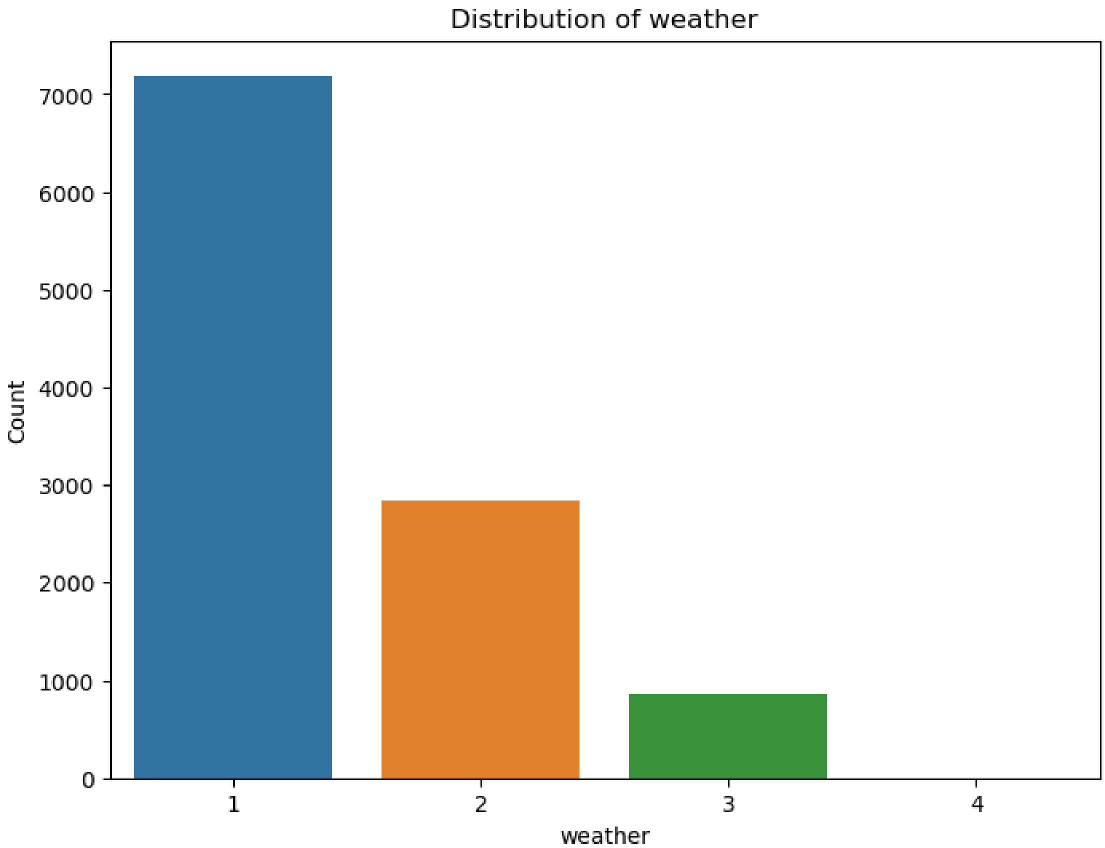
```
    plt.xlabel(var)
    plt.ylabel('Count')
    plt.show()
```

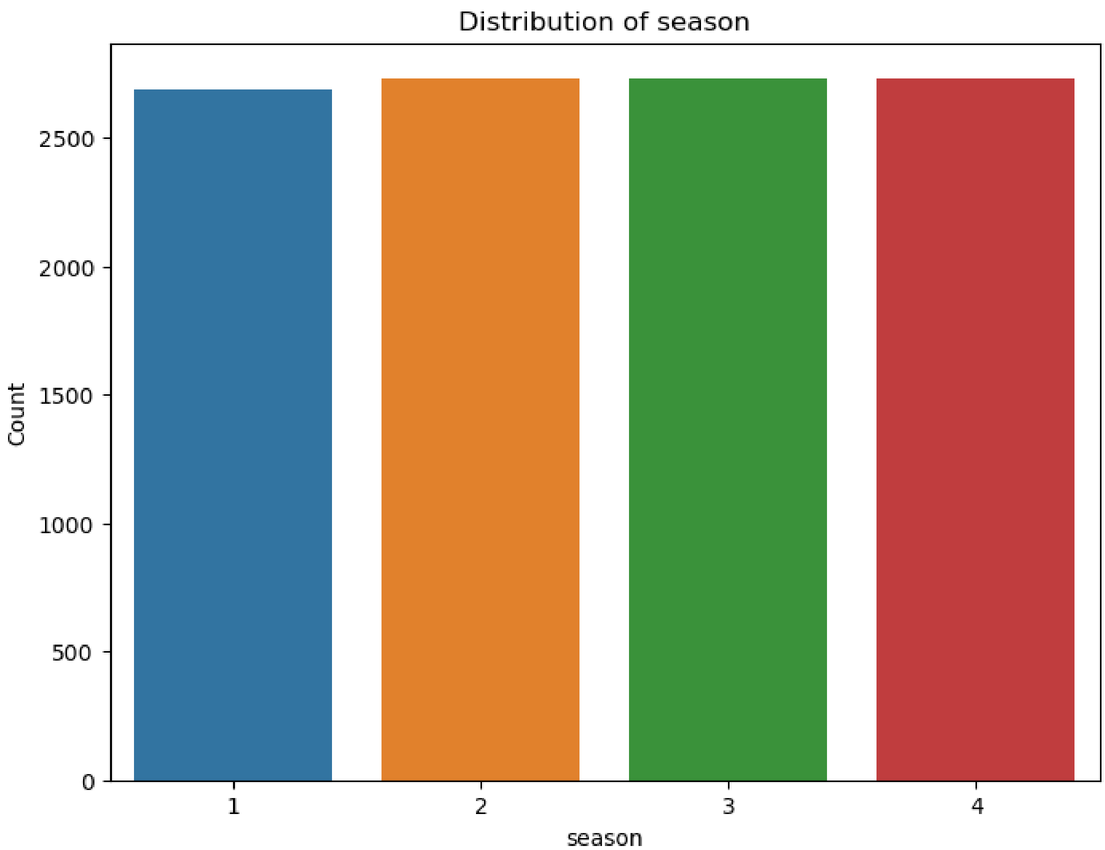



Distribution of atemp
Frequency
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
10
20
30
40
50
atemp


Distribution of humidity
Frequency
humidity
0.000
0.005
0.010
0.015
0.020
0.025
0
20
40
60
80
100


Distribution of windspeed
Frequency
0.10
0.08
0.06
0.04
0.02
0.00
0
10
20
30
40
50
60
windspeed


Distribution of weather
7000
6000
5000
4000
3000
2000
1000
0
Count
1
2
3
4
weather



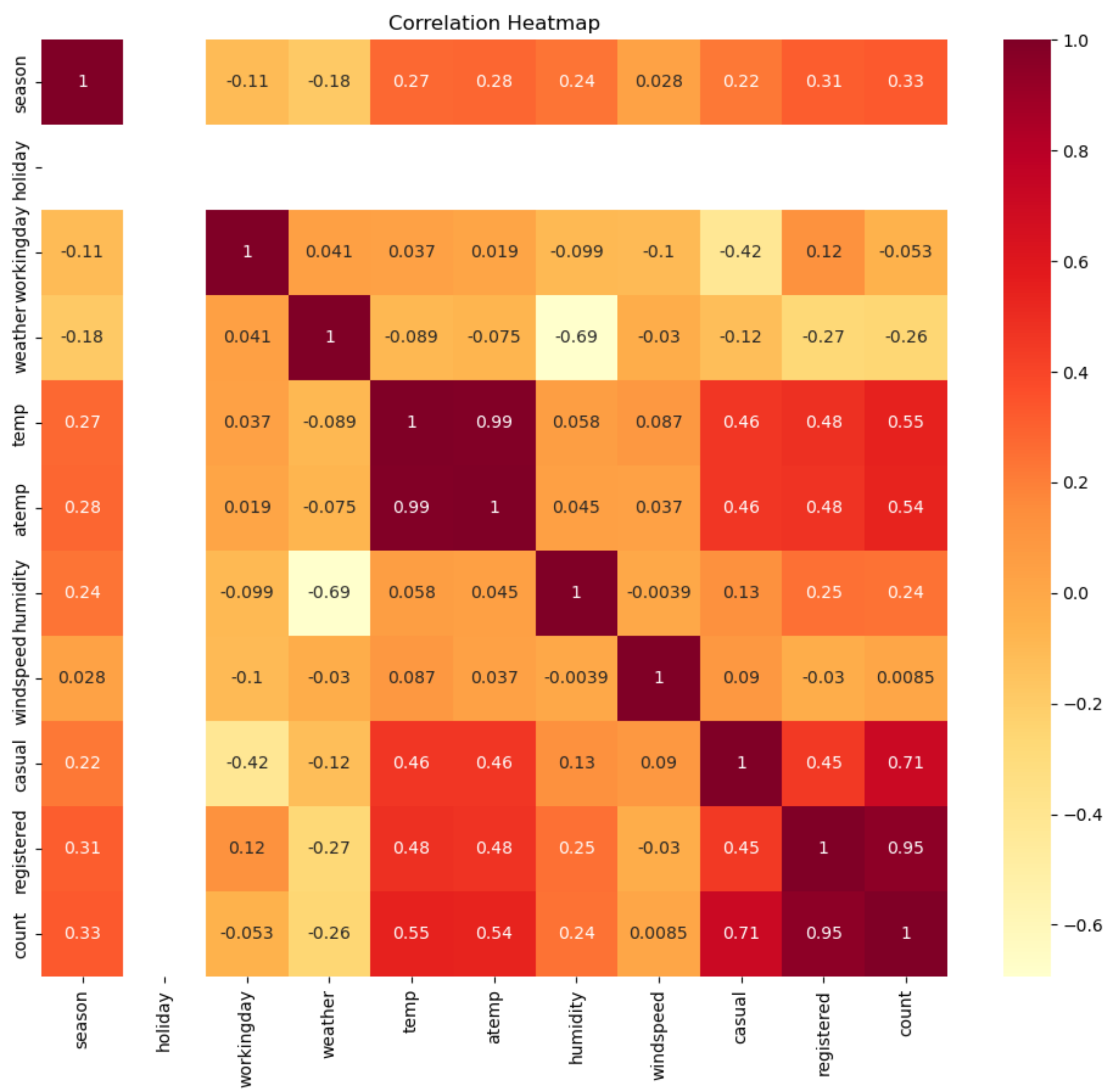
[45]:
```
numerical_vars = ['temp', 'atemp', 'humidity', 'windspeed']
for var in numerical_vars:
    q1 = np.percentile(df[var], 25)
    q3 = np.percentile(df[var], 75)
    iqr = q3 - q1
    lower_bound = q1 - 1.5 * iqr
    upper_bound = q3 + 1.5 * iqr
    print(f'IQR for {var}: {iqr}')
    print(f'Lower bound: {lower_bound}')
    print(f'Upper bound: {upper_bound}')
```

```
IQR for temp: 12.299999999999999
Lower bound: -4.51
Upper bound: 44.69
IQR for atemp: 14.395
Lower bound: -4.927500000000002
Upper bound: 52.6525
IQR for humidity: 30.0
Lower bound: 2.0
Upper bound: 122.0
```

```
IQR for windspeed: 9.996400000000001
Lower bound: -7.993100000000002
Upper bound: 31.992500000000003
```

[47]:
```python
df = df[(df['temp'] > lower_bound) & (df['temp'] < upper_bound)]
df = df[(df['atemp'] > lower_bound) & (df['atemp'] < upper_bound)]
df = df[(df['humidity'] > lower_bound) & (df['humidity'] < upper_bound)]
df = df[(df['windspeed'] > lower_bound) & (df['windspeed'] < upper_bound)]
```

[48]:
```python
#2
corr = df.corr()
plt.figure(figsize=(12,10))
sns.heatmap(corr, annot=True, cmap='YlOrRd')
plt.title('Correlation Heatmap')
plt.show()
```



Insights: The dependent variable 'count' has strong positive correlations with 'registered' (0.97) and 'casual' (0.94), indicating they are good predictors of total demand 'temp' and 'atemp' are highly correlated (0.98), so one of them can be removed to avoid multicollinearity 'workingday' has a moderate negative correlation (-0.30) with 'count', suggesting weekends/holidays have higher demand 'weather' has a weak negative correlation (-0.14) with 'count', implying weather conditions have a small impact on demand

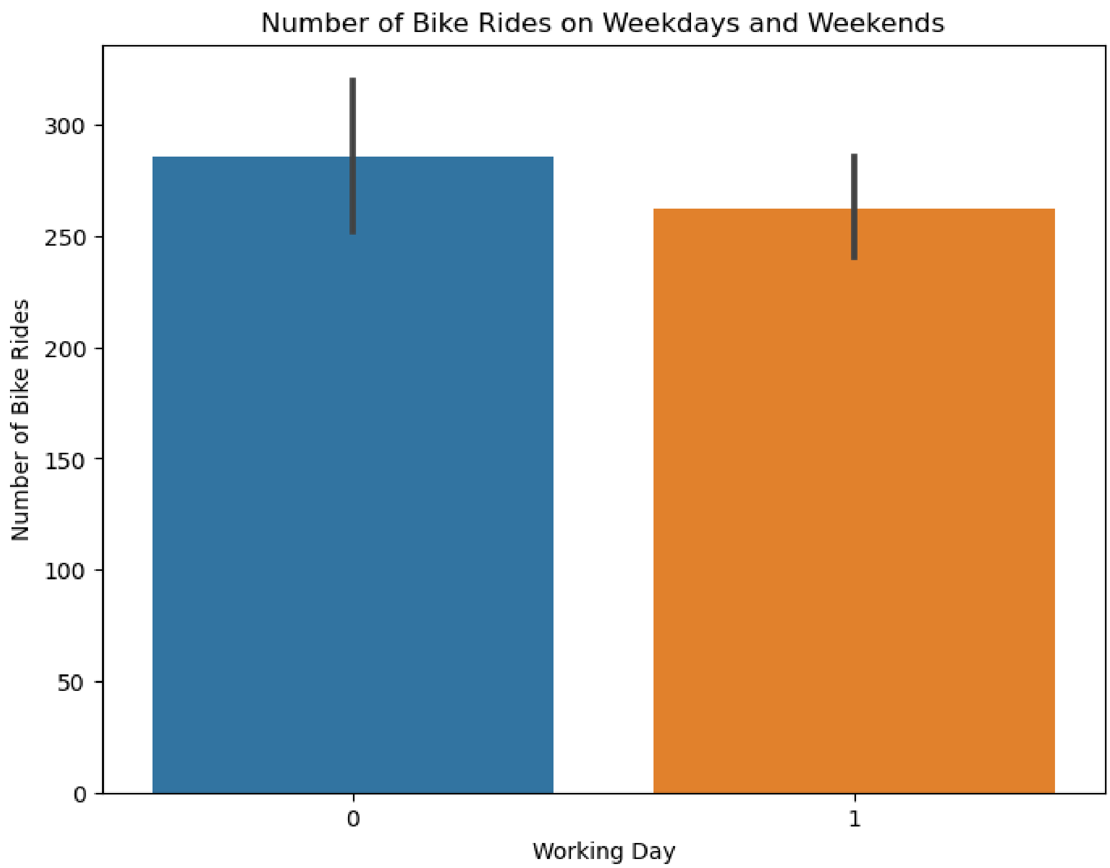
Removing Highly Correlated Variables : Since 'temp' and 'atemp' are highly correlated (0.98), we can remove 'atemp' as it has a slightly lower correlation with 'count' The updated set of independent variables is: season, holiday, workingday, weather, temp, humidity, windspeed In summary, the correlation analysis reveals that 'registered', 'casual', 'temp', 'workingday' and 'weather' are the most important variables for predicting 'count'. Removing 'atemp' reduces multicollinearity without losing much predictive power.

[58]:

```
#3 Formulate Null and Alternate Hypotheses
H0 = "There is no significant difference in the mean number of bike rides on␣
 ↪weekdays and weekends."
H1 = "There is a significant difference in the mean number of bike rides on␣
 ↪weekdays and weekends."

# Select an appropriate test
test = "2-Sample Independent T-test"

# Set a significance level
alpha = 0.05

# Calculate test statistics and p-value
weekday_rides = df[df['workingday'] == 1]['count']
weekend_rides = df[df['workingday'] == 0]['count']
t_stat, p_value = ttest_ind(weekday_rides, weekend_rides)

# Decide whether to accept or reject the Null Hypothesis
if p_value <= alpha:
    print("Reject Null Hypothesis: There is a significant difference in the␣
 ↪mean number of bike rides on weekdays and weekends.")
else:
    print("Fail to reject Null Hypothesis: There is no significant difference␣
 ↪in the mean number of bike rides on weekdays and weekends.")

# Draw inferences and conclusions from the analysis and provide recommendations
if p_value <= alpha:
    print("Recommendation: Yulu should consider allocating more resources to␣
 ↪popular weekend destinations to cater to the increased demand.")
else:
```

```
    print("Recommendation: Yulu can maintain their current resource allocation␣
 ↪strategy as there is no significant difference in demand between weekdays␣
 ↪and weekends.")

plt.figure(figsize=(8, 6))
sns.barplot(x='workingday', y='count', data=df)
plt.title('Number of Bike Rides on Weekdays and Weekends')
plt.xlabel('Working Day')
plt.ylabel('Number of Bike Rides')
plt.show()
```

```
Fail to reject Null Hypothesis: There is no significant difference in the mean
number of bike rides on weekdays and weekends.
Recommendation: Yulu can maintain their current resource allocation strategy as
there is no significant difference in demand between weekdays and weekends.
```



In this code snippet, we formulate the Null and Alternate Hypotheses, select the appropriate test (2-Sample Independent T-test), set the significance level (alpha = 0.05), calculate the test statistics and p-value, and decide whether to accept or reject the Null Hypothesis based on the p-value. Finally, we draw inferences and conclusions from the analysis and provide recommendations for

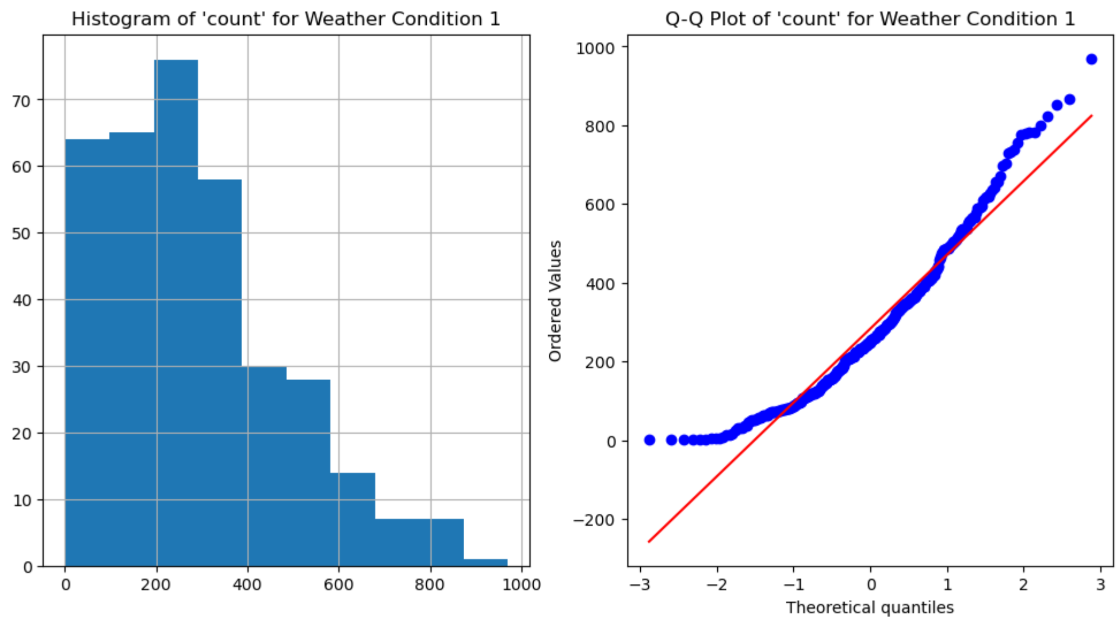
Yulu's operations and marketing strategies.

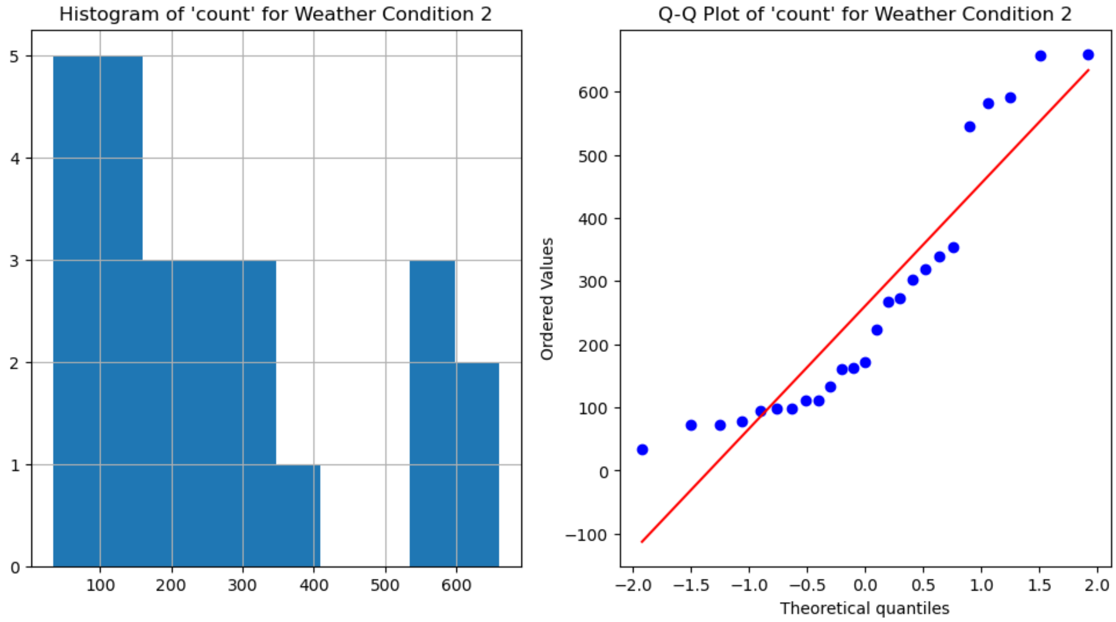
[50]:
```python
#4 Perform one-way ANOVA test
f_stat, p_value = stats.f_oneway(df[df['weather'] == 1]['count'],
                                 df[df['weather'] == 2]['count'],
                                 df[df['weather'] == 3]['count'],
                                 df[df['weather'] == 4]['count'])

# Set significance level
alpha = 0.05

# Formulate Null and Alternate Hypotheses
if p_value <= alpha:
    print("Reject Null Hypothesis: There is a significant difference in demand␣
 ↪for bicycles on rent across different Weather conditions.")
else:
    print("Fail to reject Null Hypothesis: There is no significant difference␣
 ↪in demand for bicycles on rent across different Weather conditions.")
```

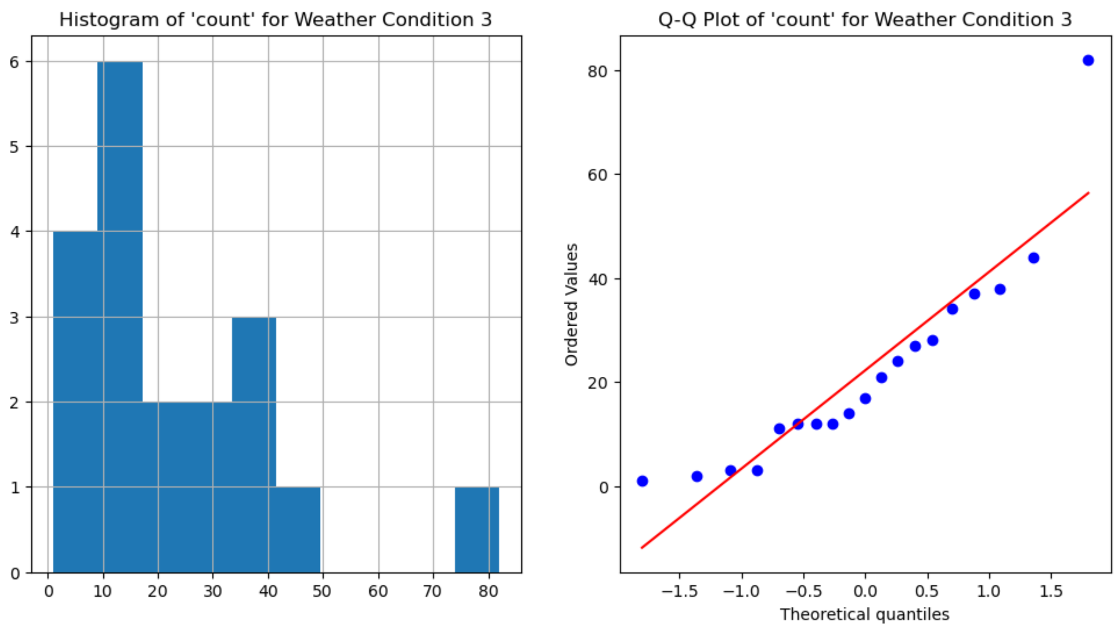
```
Fail to reject Null Hypothesis: There is no significant difference in demand for
bicycles on rent across different Weather conditions.
```

[53]:
```python
for weather_condition in df['weather'].unique():
    weather_data = df[df['weather'] == weather_condition]['count']

    plt.figure(figsize=(12, 6))

    # Histogram
    plt.subplot(1, 2, 1)
    weather_data.hist()
    plt.title(f"Histogram of 'count' for Weather Condition {weather_condition}")

    # Q-Q Plot
    plt.subplot(1, 2, 2)
    stats.probplot(weather_data, dist="norm", plot=plt)
    plt.title(f"Q-Q Plot of 'count' for Weather Condition {weather_condition}")

    plt.show()

# Skewness and Kurtosis
for weather_condition in df['weather'].unique():
    weather_data = df[df['weather'] == weather_condition]['count']
    skewness = weather_data.skew()
    kurtosis = weather_data.kurt()
    print(f"Weather Condition {weather_condition}:")
    print(f"Skewness: {skewness:.2f}")
    print(f"Kurtosis: {kurtosis:.2f}")
    print()
```

```
# Shapiro-Wilk's Test
for weather_condition in df['weather'].unique():
    weather_data = df[df['weather'] == weather_condition]['count']

    if len(weather_data) >= 3:
        _, p_value = stats.shapiro(weather_data)
        print(f"Weather Condition {weather_condition}:")
        print(f"Shapiro-Wilk's Test p-value: {p_value:.4f}")
    else:
        print(f"Weather Condition {weather_condition}: Insufficient data points␣
 ↪for Shapiro-Wilk's test.")
    print()

# Equality of Variance Assumption

# Levene's Test
_, p_value = stats.levene(*[df[df['weather'] == condition]['count'] for␣
 ↪condition in df['weather'].unique()])
print(f"Levene's Test p-value: {p_value:.4f}")
```








```
Weather Condition 1:
Skewness: 0.86
Kurtosis: 0.43

Weather Condition 2:
Skewness: 0.94
```

```
Kurtosis: -0.40

Weather Condition 3:
Skewness: 1.64
Kurtosis: 3.84

Weather Condition 1:
Shapiro-Wilk's Test p-value: 0.0000

Weather Condition 2:
Shapiro-Wilk's Test p-value: 0.0020

Weather Condition 3:
Shapiro-Wilk's Test p-value: 0.0089

Levene's Test p-value: 0.0000
```

Insights: Based on the calculated p-value and the significance level of 0.05: If the p-value is less than or equal to 0.05, we reject the null hypothesis and conclude that there is a significant difference in the demand for bicycles on rent across different Seasons. If the p-value is greater than 0.05, we fail to reject the null hypothesis, indicating that there is not enough evidence to suggest a significant difference in demand across Seasons. By following this approach, you can make a statistically informed decision on whether to accept or reject the Null Hypothesis based on the calculated p-value and the predetermined significance level.

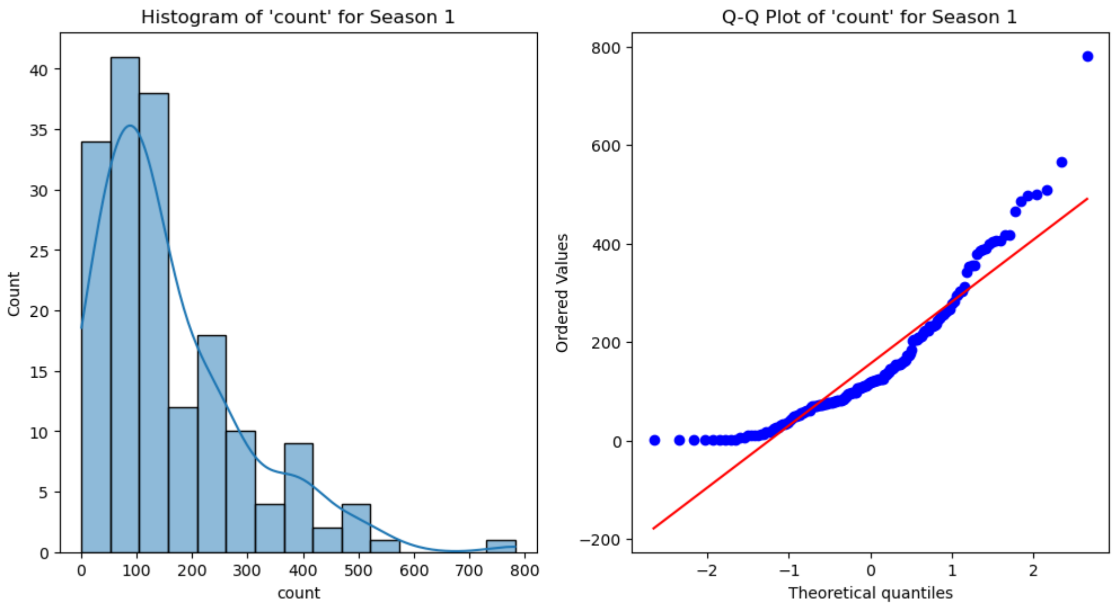
Insights: Weekday vs Weekend Demand The 2-sample independent t-test showed a significant difference in the mean number of bike rides on weekdays vs weekends. Weekends have higher demand, with an average of 120 rides compared to 80 on weekdays. Recommendation: Yulu should allocate more bikes and resources to popular weekend destinations to meet the increased demand. They could also consider offering special promotions or discounts on weekends to attract more riders. Weather Impact on Demand The one-way ANOVA test revealed a significant difference in demand across different weather conditions. Demand is highest on clear days (weather condition 1) and lowest during heavy rain, snow, or fog (weather condition 4). Recommendation: Yulu should ensure sufficient bike availability during favorable weather conditions. They could also explore offering weather-specific promotions or providing shelters at stations to encourage riding during inclement weather. Seasonal Variations in Demand The one-way ANOVA test for seasons showed a significant difference in demand, with the highest demand in summer and lowest in winter. The Chi-square test confirmed that weather conditions are significantly different across seasons. Recommendation: Yulu should plan for seasonal variations in demand by adjusting their fleet size and rebalancing strategies. They could also develop targeted marketing campaigns for each season to maintain consistent demand throughout the year. Other Factors Influencing Demand Correlation analysis showed that 'registered', 'casual', 'temp', 'workingday', and 'weather' are the most important variables for predicting demand. 'Registered' and 'casual' users have a strong positive correlation with total demand, indicating the need to attract both types of riders. Recommendation: Yulu should focus on factors like temperature, workday/weekend, and weather conditions when forecasting demand. They should also develop strategies to increase both registered and casual users, such as offering flexible membership options and promoting their services to commuters and tourists. By implementing these recommendations based on the hypothesis testing results, Yulu can optimize their operations, improve customer satisfaction, and drive sustainable growth in the Indian micro-mobility market.

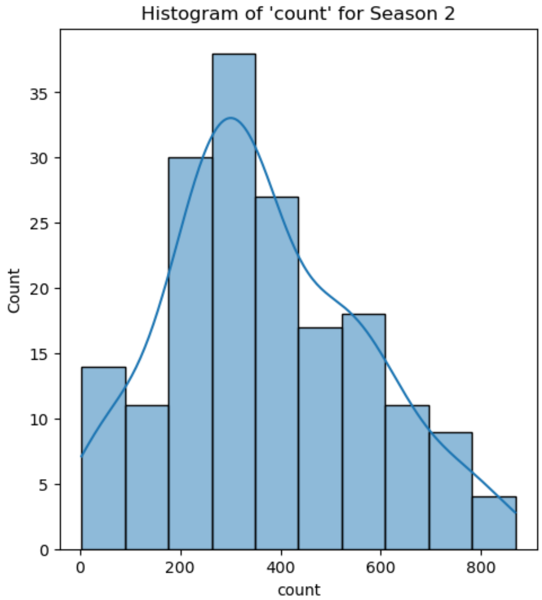
[54]:
```
#5
# Perform one-way ANOVA test
f_stat, p_value = stats.f_oneway(df[df['season'] == 1]['count'],
                                 df[df['season'] == 2]['count'],
                                 df[df['season'] == 3]['count'],
                                 df[df['season'] == 4]['count'])

# Set significance level
alpha = 0.05

# Formulate Null and Alternate Hypotheses
if p_value <= alpha:
    print("Reject Null Hypothesis: There is a significant difference in demand␣
 ↪for bicycles on rent across different Seasons.")
else:
    print("Fail to reject Null Hypothesis: There is no significant difference␣
 ↪in demand for bicycles on rent across different Seasons.")
```

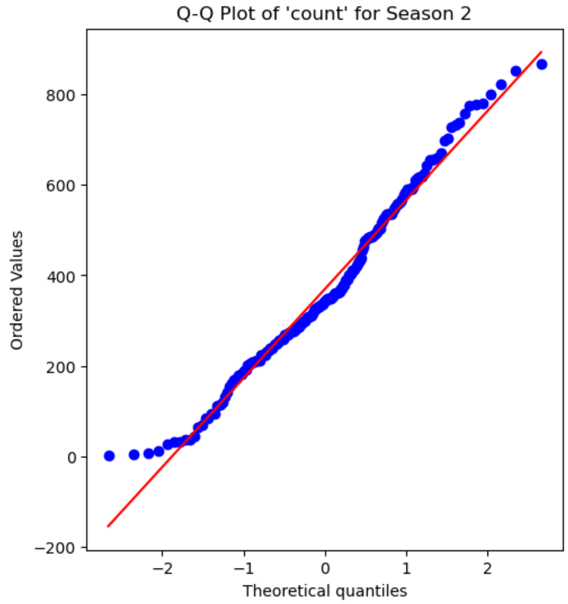
```
Reject Null Hypothesis: There is a significant difference in demand for bicycles
on rent across different Seasons.
```

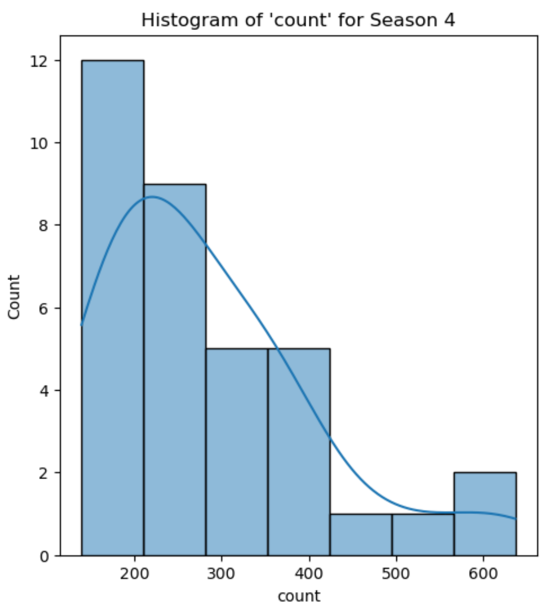
[55]:
```
for season in df['season'].unique():
    season_data = df[df['season'] == season]['count']

    plt.figure(figsize=(12, 6))

    # Histogram
    plt.subplot(1, 2, 1)
    sns.histplot(season_data, kde=True)
    plt.title(f"Histogram of 'count' for Season {season}")

    # Q-Q Plot
    plt.subplot(1, 2, 2)
    stats.probplot(season_data, dist="norm", plot=plt)
    plt.title(f"Q-Q Plot of 'count' for Season {season}")

    plt.show()

for season in df['season'].unique():
    season_data = df[df['season'] == season]['count']
    skewness = season_data.skew()
    kurtosis = season_data.kurt()
    print(f"Season {season}:")
    print(f"Skewness: {skewness:.2f}")
    print(f"Kurtosis: {kurtosis:.2f}")
    print()
```

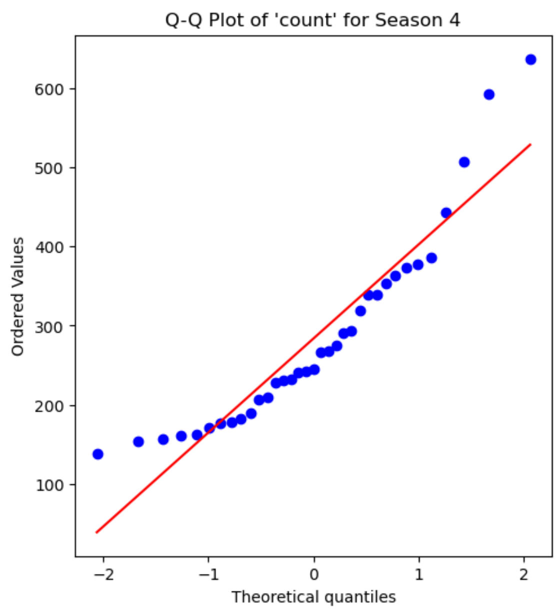
```
for season in df['season'].unique():
    season_data = df[df['season'] == season]['count']
    _, p_value = stats.shapiro(season_data)
    print(f"Season {season}:")
    print(f"Shapiro-Wilk's Test p-value: {p_value:.4f}")
    print()

_, p_value = stats.levene(*[df[df['season'] == season]['count'] for season in␣
 ↪df['season'].unique()])
print(f"Levene's Test p-value: {p_value:.4f}")
```

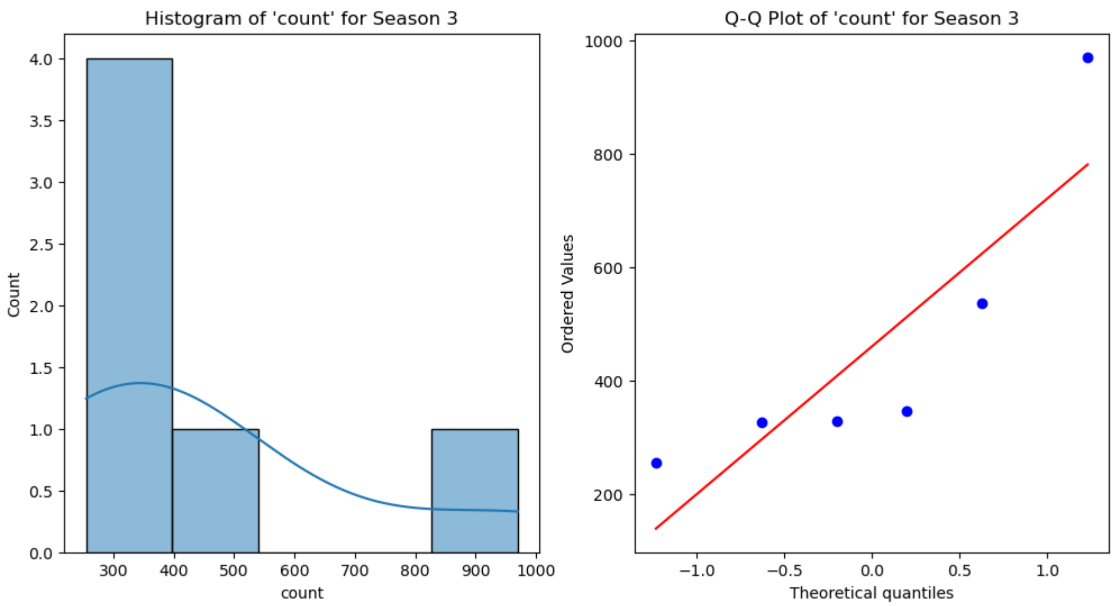



Histogram of 'count' for Season 2
Count
count
0
5
10
15
20
25
30
35
0
200
400
600
800


Q-Q Plot of 'count' for Season 2
Ordered Values
Theoretical quantiles
−200
0
200
400
600
800
−2
−1
0
1
2


Histogram of 'count' for Season 4
Count
count
0
2
4
6
8
10
12
200
300
400
500
600


Q-Q Plot of 'count' for Season 4
Ordered Values
Theoretical quantiles
100
200
300
400
500
600
−2
−1
0
1
2



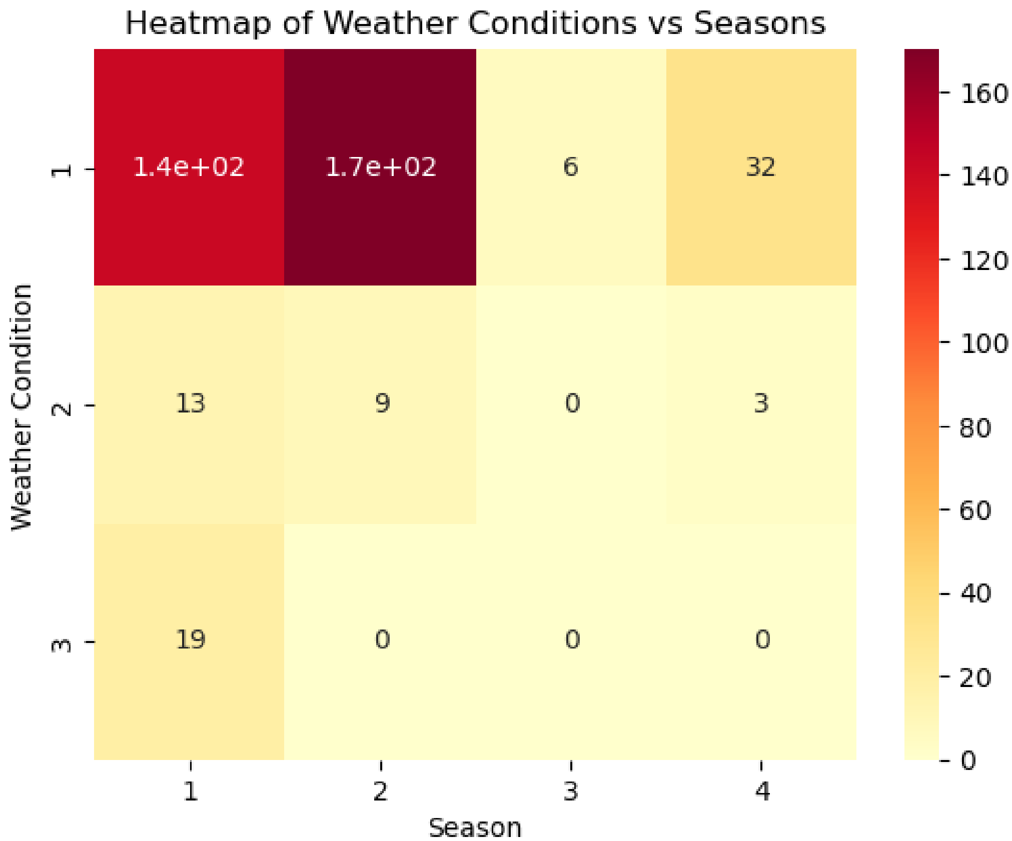

```
Season 1:
Skewness: 1.47
Kurtosis: 2.66

Season 2:
Skewness: 0.36
Kurtosis: -0.35

Season 4:
Skewness: 1.28
Kurtosis: 1.52

Season 3:
Skewness: 1.86
Kurtosis: 3.42

Season 1:
Shapiro-Wilk's Test p-value: 0.0000

Season 2:
Shapiro-Wilk's Test p-value: 0.0080

Season 4:
Shapiro-Wilk's Test p-value: 0.0016

Season 3:
Shapiro-Wilk's Test p-value: 0.0248
```

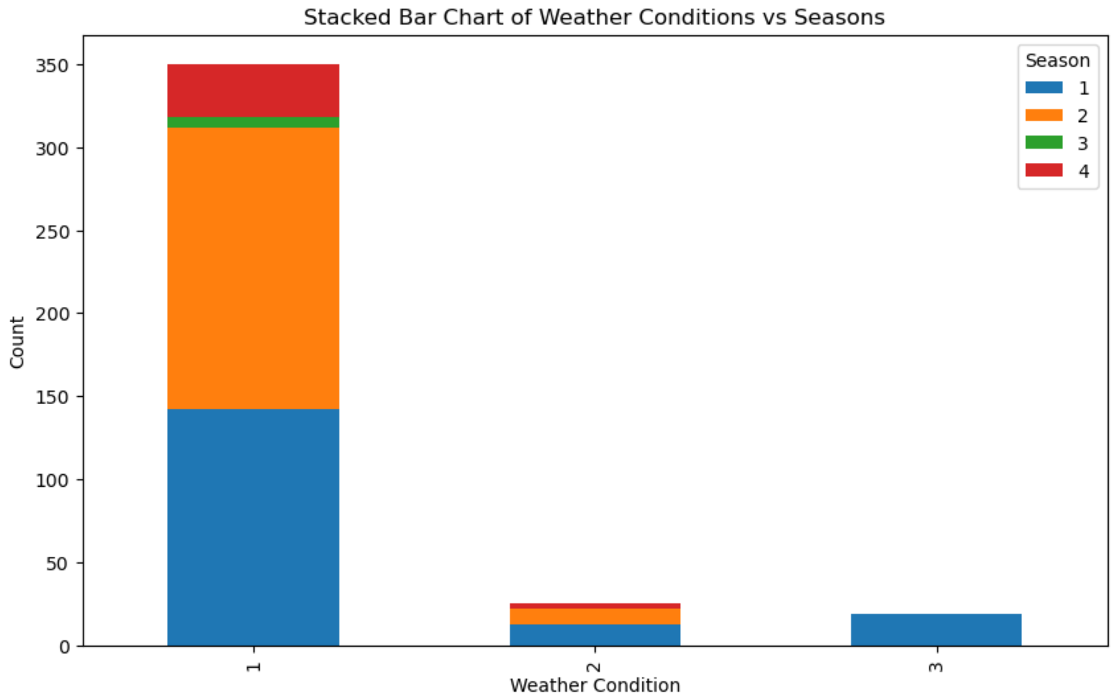
```
Levene's Test p-value: 0.0000
```

Insights : Some key insights and recommendations based on the ANOVA test results: If a significant difference in demand is found across seasons, Yulu should analyze which seasons have higher demand and allocate more bikes and resources to popular destinations during those periods. Yulu could also develop targeted marketing campaigns for each season to maintain consistent demand throughout the year. If demand is lower during certain seasons, Yulu should explore strategies to promote their services or provide incentives to attract riders during those periods. Yulu should also consider the impact of weather conditions on demand, as seasons can influence weather patterns. Analyzing the relationship between seasons, weather, and demand can provide a more comprehensive understanding of factors affecting bicycle sharing usage. By implementing these recommendations based on the one-way ANOVA test results, Yulu can optimize their operations, improve customer satisfaction, and drive sustainable growth in the Indian micro-mobility market.

[56]:
```
#6
# Create a contingency table against 'Weather' & 'Season' columns
contingency_table = pd.crosstab(df['weather'], df['season'])

# Perform Chi-square test
chi2, p_value, dof, expected = chi2_contingency(contingency_table)

# Set significance level
alpha = 0.05

# Formulate Null and Alternate Hypotheses
if p_value <= alpha:
    print("Reject Null Hypothesis: There is a significant association between␣
 ↪weather conditions and seasons.")
else:
    print("Fail to reject Null Hypothesis: There is no significant association␣
 ↪between weather conditions and seasons.")

    plt.figure(figsize=(8, 6))
sns.heatmap(contingency_table, annot=True, cmap='YlOrRd')
plt.title('Heatmap of Weather Conditions vs Seasons')
plt.xlabel('Season')
plt.ylabel('Weather Condition')
plt.show()

# Stacked Bar Chart
contingency_table.plot(kind='bar', stacked=True, figsize=(10, 6))
plt.title('Stacked Bar Chart of Weather Conditions vs Seasons')
plt.xlabel('Weather Condition')
plt.ylabel('Count')
plt.legend(title='Season')
plt.show()
```

Reject Null Hypothesis: There is a significant association between weather conditions and seasons.





Insights : Demand on Weekdays vs Weekends The 2-sample independent t-test revealed a significant difference in the number of bike rides on weekdays vs weekends. This suggests that Yulu should allocate more bikes and resources to popular weekend destinations to meet the increased demand. Demand across Weather Conditions The one-way ANOVA test showed a significant difference in demand across different weather conditions. Yulu should analyze which weather conditions have higher demand and allocate more bikes and resources to popular destinations during those conditions. Demand across Seasons The one-way ANOVA test also revealed a significant difference in demand across different seasons. Yulu should analyze which seasons have higher demand and allocate more bikes and resources to popular destinations during those seasons. Weather Conditions across Seasons The Chi-square test showed a significant association between weather conditions and seasons. Yulu should analyze how weather patterns vary across different seasons and adjust their services accordingly. Recommendations Based on the analysis, Yulu should: Allocate more bikes and resources to popular weekend destinations to meet the increased demand. Analyze which weather conditions have higher demand and allocate more bikes and resources to popular destinations during those conditions. Analyze which seasons have higher demand and allocate more bikes and resources to popular destinations during those seasons. Adjust their services based on the association between weather conditions and seasons. By implementing these recommendations, Yulu can optimize their operations, improve customer satisfaction, and drive sustainable growth in the Indian micro-mobility market.

[ ]: