Step 1: Import the Dataset and Perform EDA

1.1 Load the dataset and check its structure

```
import pandas as pd

# Load the dataset
file_path = 'bike_sharing.csv'
yulu_data = pd.read_csv(file_path)

# Display the first few rows of the dataset
print(yulu_data.head())

# Display summary statistics
print(yulu_data.describe())

# Display info about the dataset
print(yulu_data.info())
```

```
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
             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
<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
```

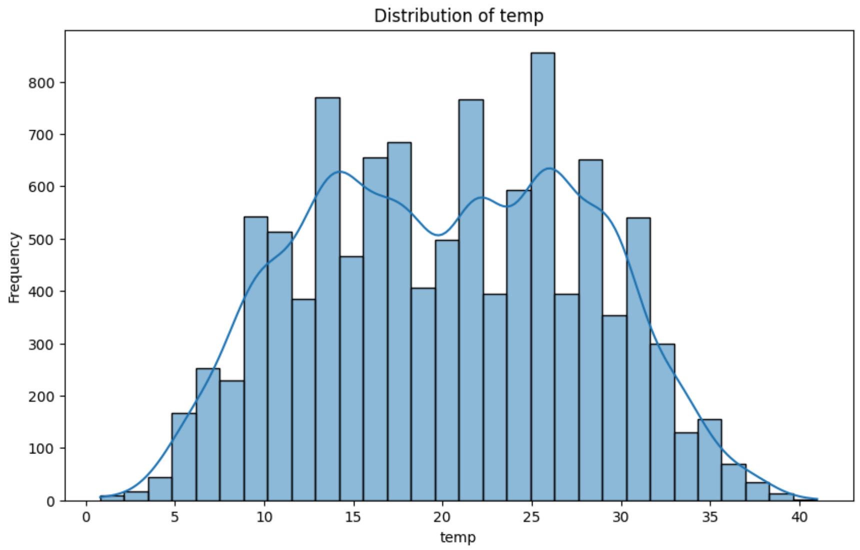
1.2 Check for missing values

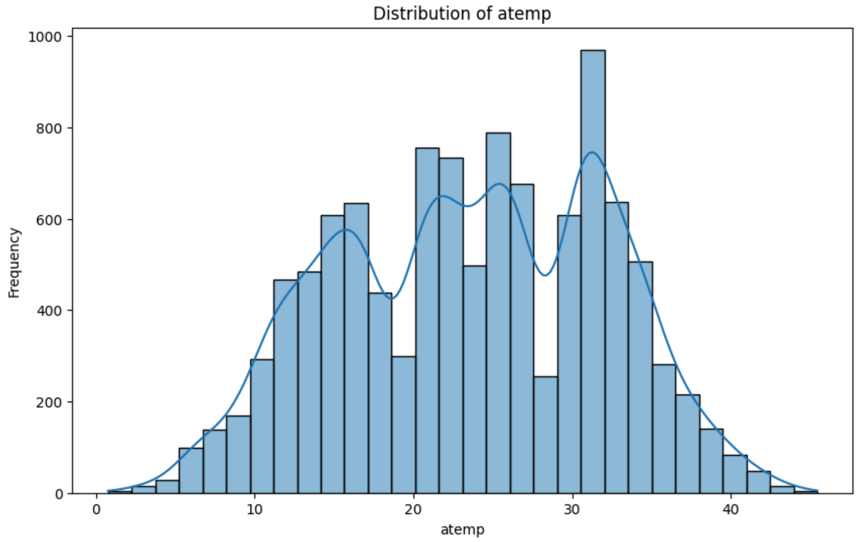
```
# Check for missing values
print(yulu_data.isnull().sum())
```

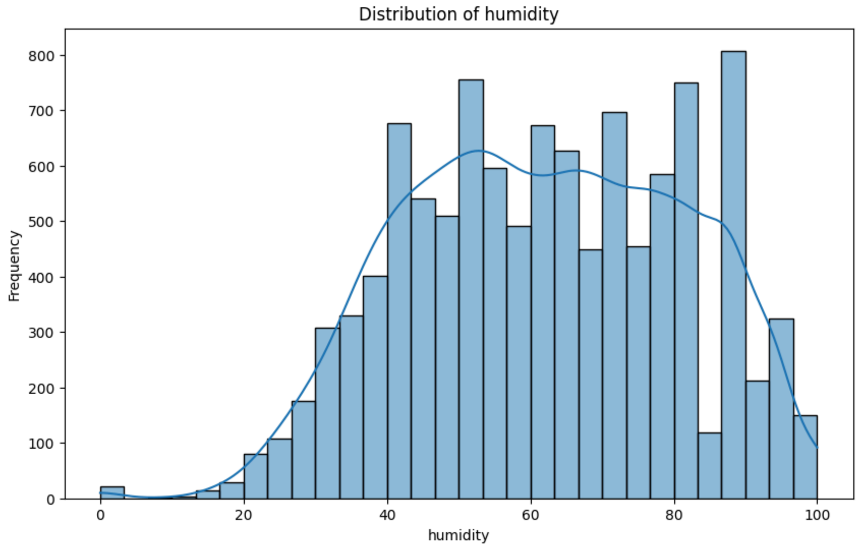
```
datetime      0
season        0
holiday       0
workingday    0
weather       0
temp          0
atemp         0
humidity      0
windspeed     0
casual        0
registered    0
count         0
dtype: int64
```

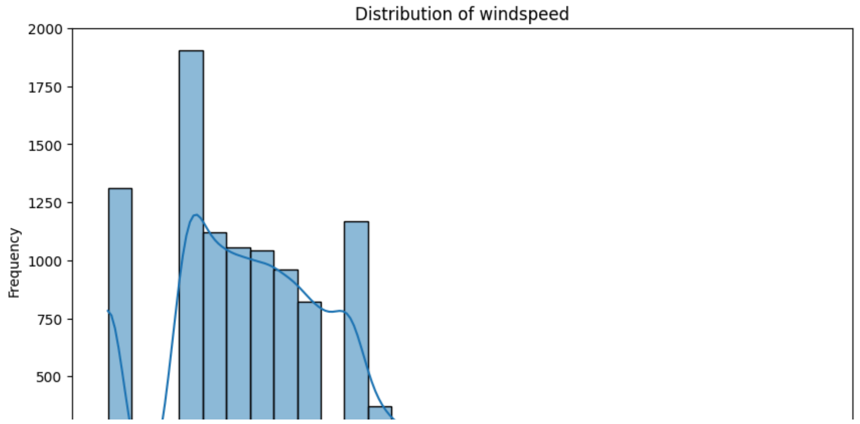
1.3 Univariate Analysis

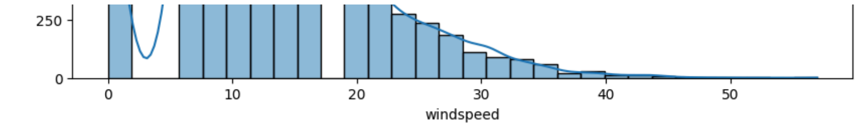
```
import matplotlib.pyplot as plt
import seaborn as sns

# Plot distribution of continuous variables
continuous_vars = ['temp', 'atemp', 'humidity', 'windspeed', 'casual', 'registered', 'count']
for var in continuous_vars:
    plt.figure(figsize=(10, 6))
    sns.histplot(yulu_data[var], bins=30, kde=True)
    plt.title(f'Distribution of {var}')
    plt.xlabel(var)
    plt.ylabel('Frequency')
    plt.show()

# Plot bar plots for categorical variables
categorical_vars = ['season', 'holiday', 'workingday', 'weather']
for var in categorical_vars:
    plt.figure(figsize=(10, 6))
    sns.countplot(data=yulu_data, x=var)
    plt.title(f'Countplot of {var}')
    plt.xlabel(var)
    plt.ylabel('Count')
    plt.show()
```

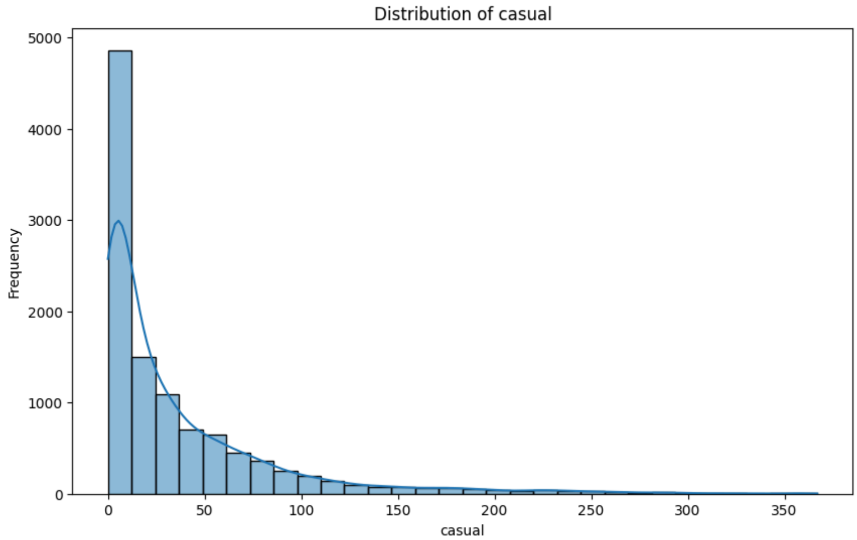

Distribution of temp
Frequency
temp
0
100
200
300
400
500
600
700
800
0
5
10
15
20
25
30
35
40

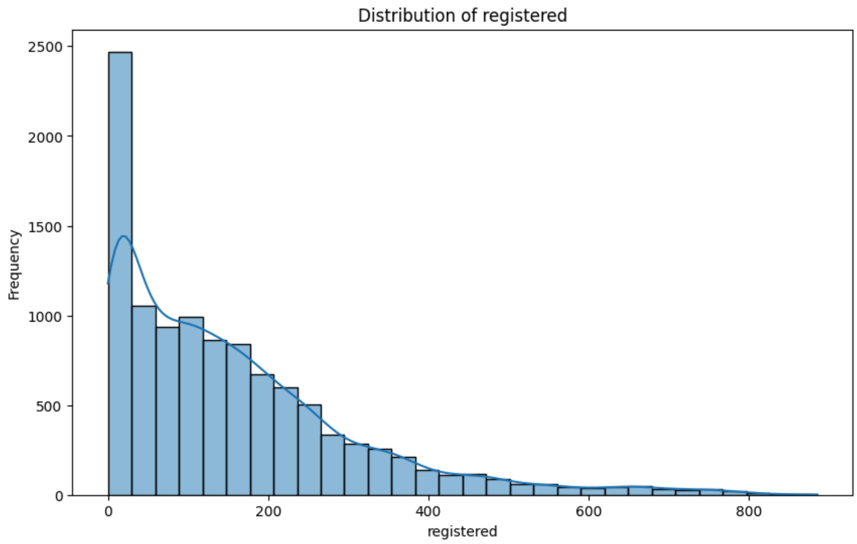

Distribution of atemp
Frequency
atemp
0
200
400
600
800
1000
0
10
20
30
40

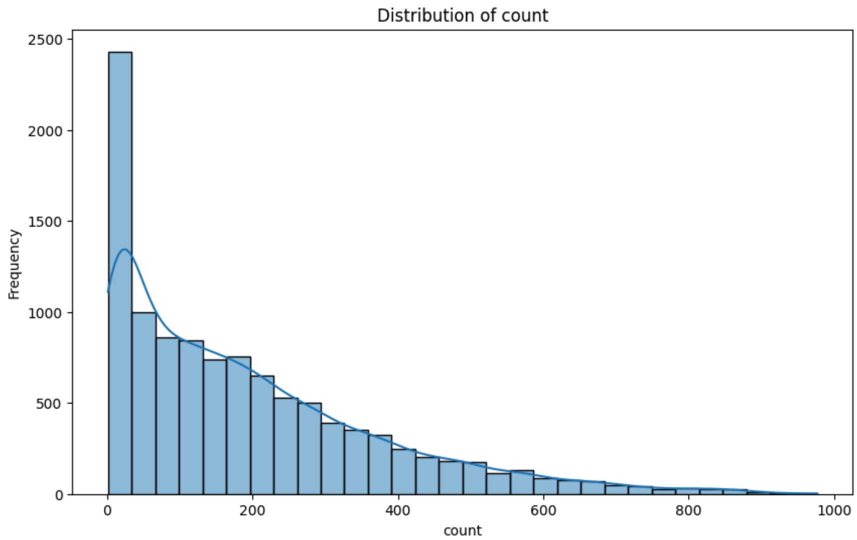

Distribution of humidity
Frequency
humidity
0
100
200
300
400
500
600
700
800
0
20
40
60
80
100

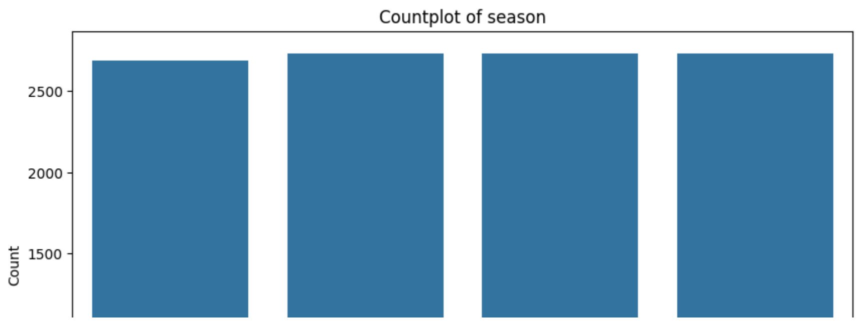

Distribution of windspeed
Frequency
500
750
1000
1250
1500
1750
2000

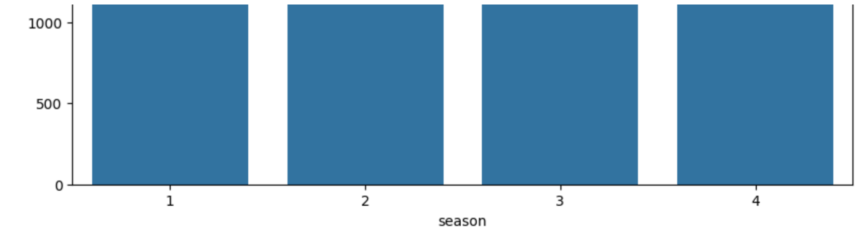

windspeed

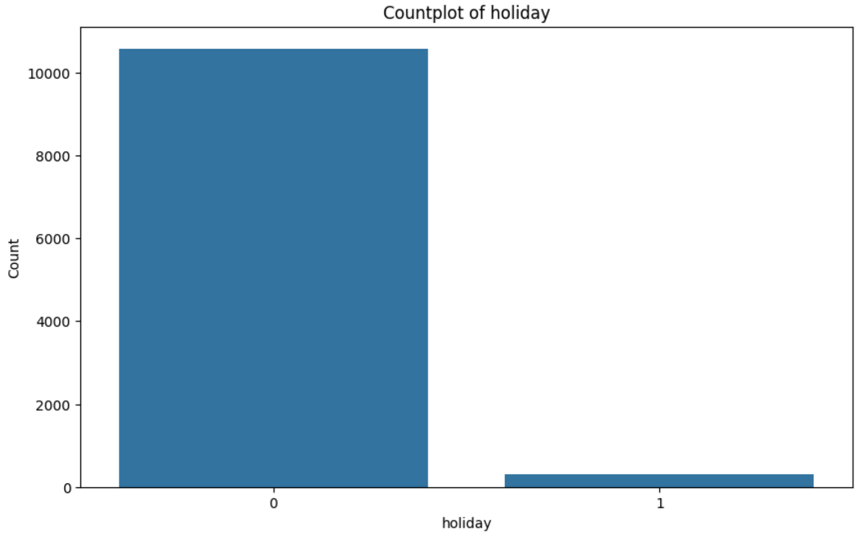

Distribution of casual
Frequency
casual

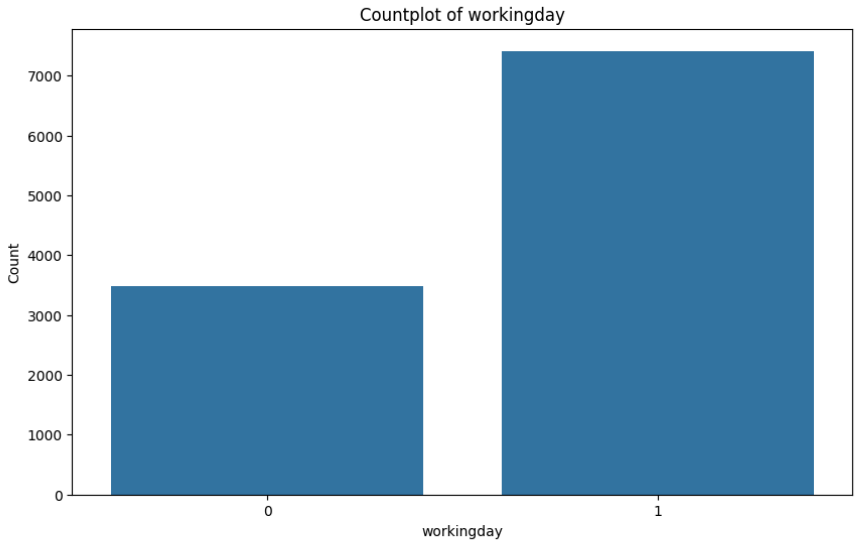

Distribution of registered
Frequency
registered

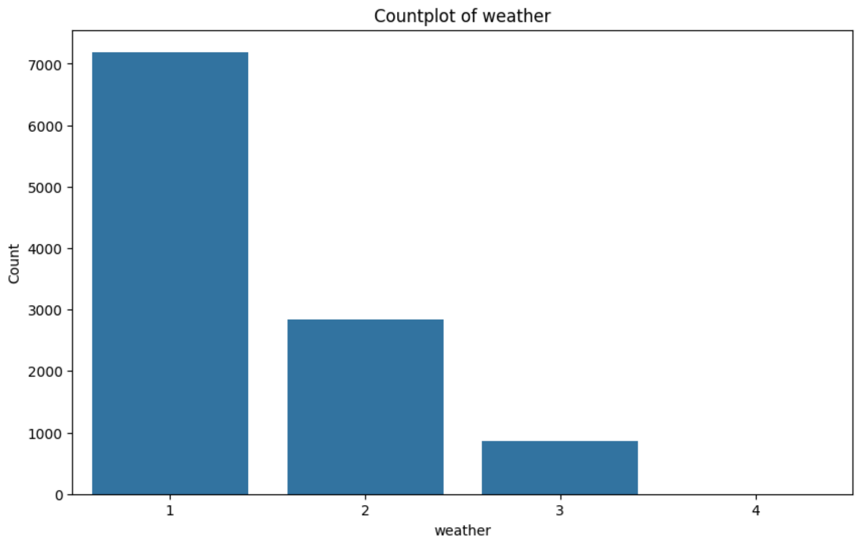

Distribution of count
Frequency
count


Countplot of season
Count


season

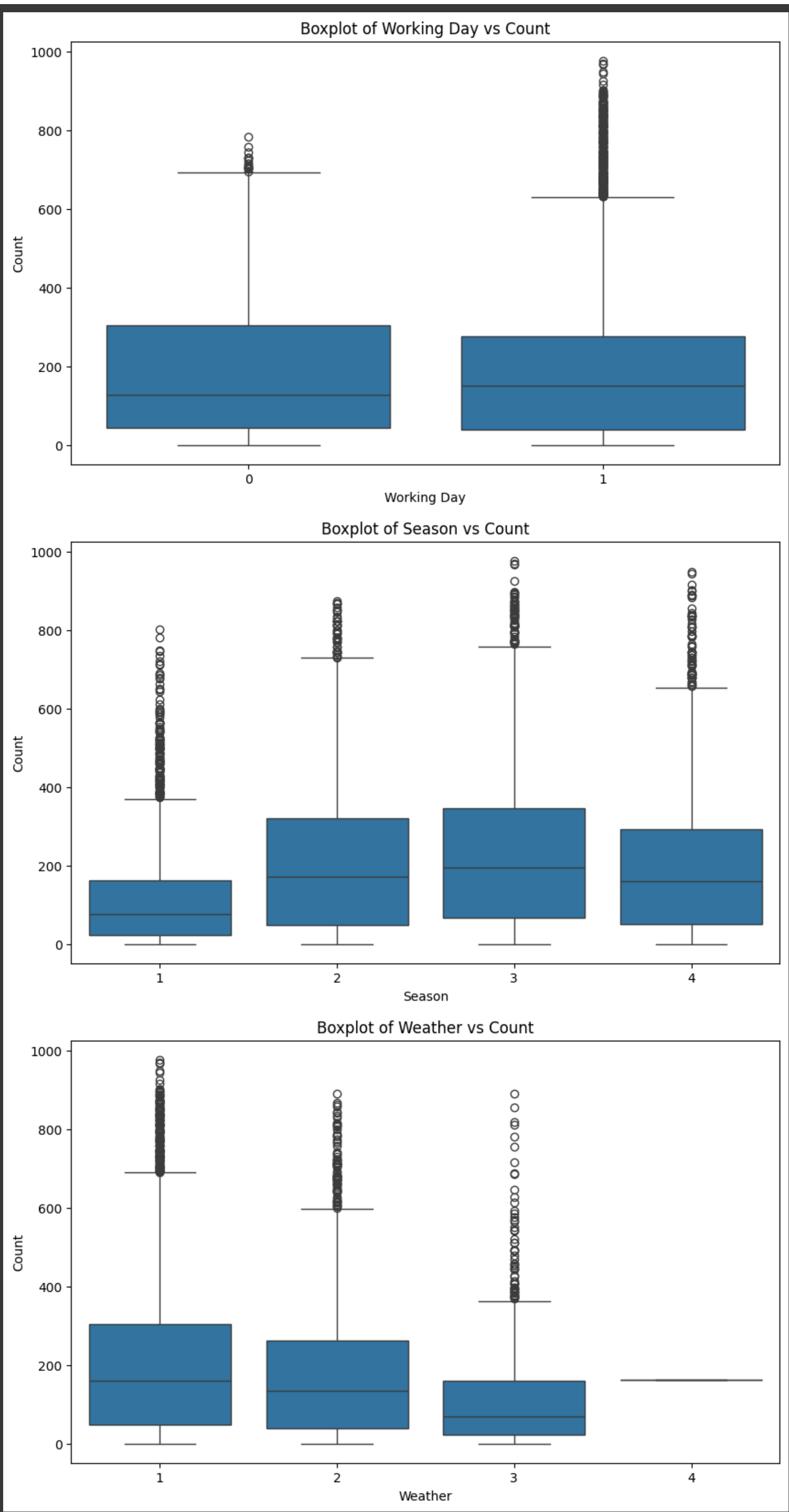

Countplot of holiday
Count
holiday


Countplot of workingday
Count
workingday


Countplot of weather
Count
weather

1.4 Bivariate Analysis

```
# Relationship between workingday and count
plt.figure(figsize=(10, 6))
sns.boxplot(data=yulu_data, x='workingday', y='count')
plt.title('Boxplot of Working Day vs Count')
plt.xlabel('Working Day')
plt.ylabel('Count')
plt.show()

# Relationship between season and count
plt.figure(figsize=(10, 6))
sns.boxplot(data=yulu_data, x='season', y='count')
plt.title('Boxplot of Season vs Count')
plt.xlabel('Season')
plt.ylabel('Count')
plt.show()

# Relationship between weather and count
plt.figure(figsize=(10, 6))
sns.boxplot(data=yulu_data, x='weather', y='count')
plt.title('Boxplot of Weather vs Count')
plt.xlabel('Weather')
plt.ylabel('Count')
plt.show()
```


Boxplot of Working Day vs Count
Count
Working Day
Boxplot of Season vs Count
Count
Season
Boxplot of Weather vs Count
Count
Weather

Step 2: Hypothesis Testing

2.1 Effect of Working Day on Number of Electric Cycles Rented Null Hypothesis (H_0): Working day has no effect on the number of electric cycles rented. Alternative Hypothesis (H_1): Working day has an effect on the number of electric cycles rented.

```
from scipy.stats import ttest_ind

# Separate data into working days and non-working days
workingday_data = yulu_data[yulu_data['workingday'] == 1]['count']
non_workingday_data = yulu_data[yulu_data['workingday'] == 0]['count']

# T-test
t_stat, p_value = ttest_ind(workingday_data, non_workingday_data)
print(f'T-test results: t-statistic = {t_stat}, p-value = {p_value}')
```

```
T-test results: t-statistic = 1.2096277376026694, p-value = 0.22644804226361348
```

2.2 Differences in Number of Cycles Rented Across Different Seasons Null Hypothesis (H_0): The number of cycles rented is similar across different seasons. Alternative Hypothesis (H_1): The number of cycles rented is different across different seasons.

```
from scipy.stats import f_oneway

# ANOVA for seasons
season_groups = [yulu_data[yulu_data['season'] == season]['count'] for season in yulu_data['season'].unique()]
f_stat_season, p_value_season = f_oneway(*season_groups)
print(f'ANOVA results for seasons: F-statistic = {f_stat_season}, p-value = {p_value_season}')
```

```
ANOVA results for seasons: F-statistic = 236.94671081032106, p-value = 6.164843386499654e-149
```

2.3 Differences in Number of Cycles Rented Across Different Weather Conditions Null Hypothesis (H_0): The number of cycles rented is similar across different weather conditions. Alternative Hypothesis (H_1): The number of cycles rented is different across different weather conditions.

```
# ANOVA for weather
weather_groups = [yulu_data[yulu_data['weather'] == weather]['count'] for weather in yulu_data['weather'].unique()]
f_stat_weather, p_value_weather = f_oneway(*weather_groups)
print(f'ANOVA results for weather: F-statistic = {f_stat_weather}, p-value = {p_value_weather}')
```

```
ANOVA results for weather: F-statistic = 65.53024112793271, p-value = 5.482069475935669e-42
```

2.4 Dependence of Weather on the Season Null Hypothesis (H_0): Weather is independent of the season. Alternative Hypothesis (H_1): Weather is dependent on the season.

```
from scipy.stats import chi2_contingency

# Chi-square test
weather_season_table = pd.crosstab(yulu_data['weather'], yulu_data['season'])
chi2_stat, p_value_chi2, dof, expected = chi2_contingency(weather_season_table)
print(f'Chi-square test results: chi2-statistic = {chi2_stat}, p-value = {p_value_chi2}')
```

```
Chi-square test results: chi2-statistic = 49.15865559689363, p-value = 1.5499250736864862e-07
```

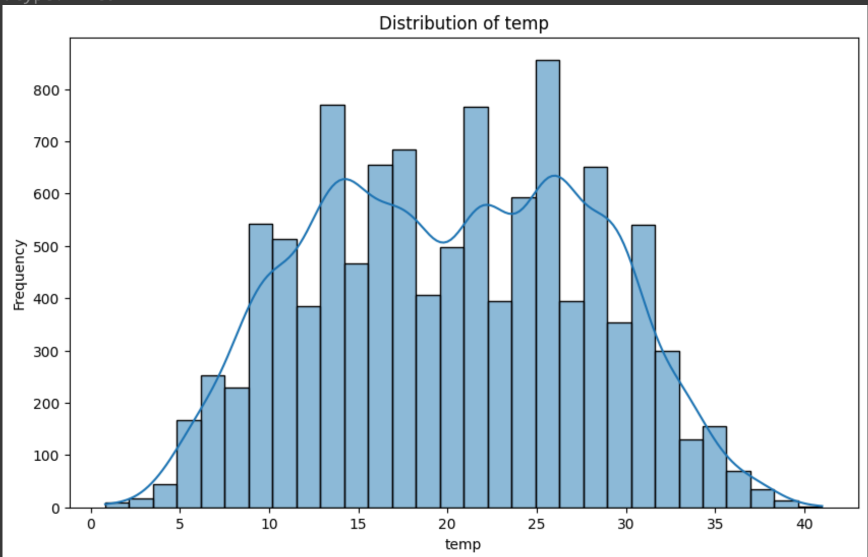
Step 3: Conclusion and Interpretation

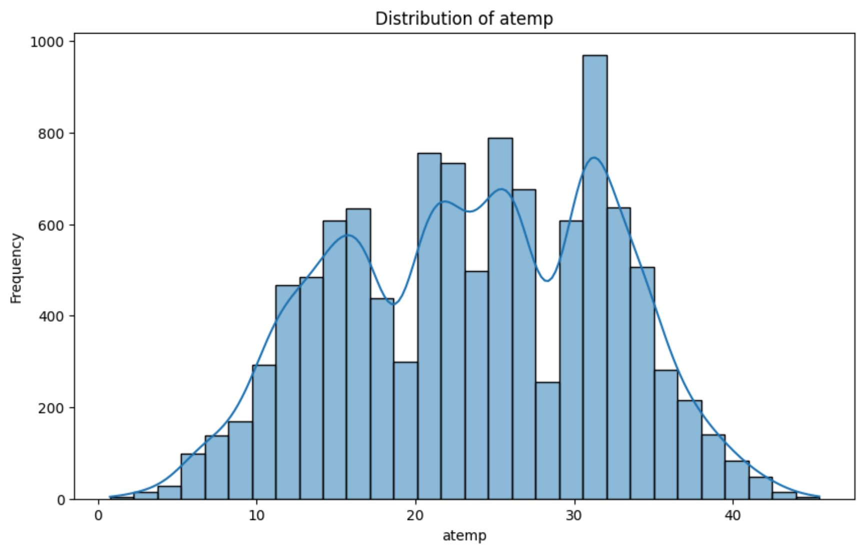
Interpret the p-values obtained from each hypothesis test and draw conclusions. For example:

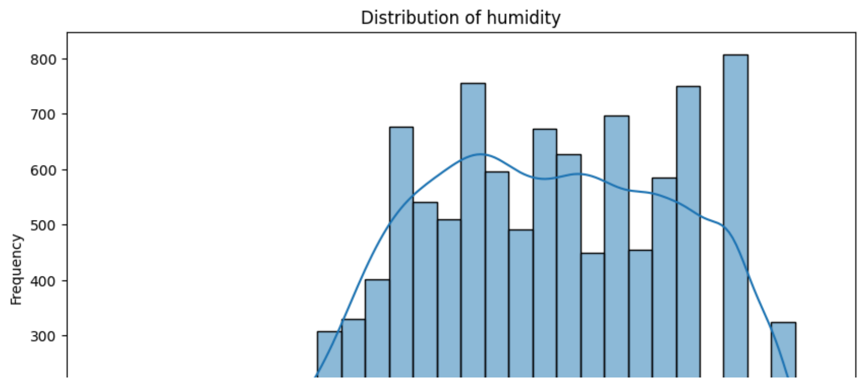
If the p-value < 0.05 for the t-test, reject the null hypothesis and conclude that working days significantly affect the number of electric cycles rented. If the p-value < 0.05 for ANOVA tests, reject the null hypothesis and conclude that there are significant differences in the number of cycles rented across different seasons and weather conditions. If the p-value < 0.05 for the Chi-square test, reject the null hypothesis and conclude that weather is dependent on the season.

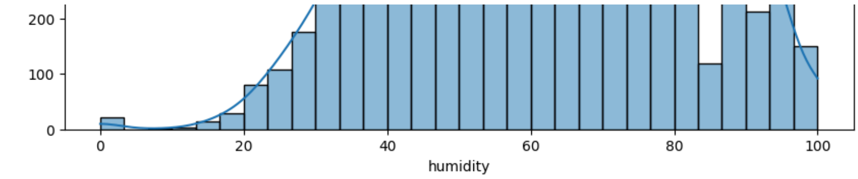
```
# Display the first few rows of the dataset
print(yulu_data.head())

# Check for missing values
print(yulu_data.isnull().sum())

# Univariate Analysis: Distribution plots for continuous variables
continuous_vars = ['temp', 'atemp', 'humidity', 'windspeed', 'casual', 'registered', 'count']
for var in continuous_vars:
    plt.figure(figsize=(10, 6))
    sns.histplot(yulu_data[var], bins=30, kde=True)
    plt.title(f'Distribution of {var}')
    plt.xlabel(var)
    plt.ylabel('Frequency')
    plt.show()

# Univariate Analysis: Bar plots for categorical variables
categorical_vars = ['season', 'holiday', 'workingday', 'weather']
for var in categorical_vars:
    plt.figure(figsize=(10, 6))
    sns.countplot(data=yulu_data, x=var)
    plt.title(f'Countplot of {var}')
    plt.xlabel(var)
    plt.ylabel('Count')
    plt.show()

# Bivariate Analysis: Relationship between workingday and count
plt.figure(figsize=(10, 6))
sns.boxplot(data=yulu_data, x='workingday', y='count')
plt.title('Boxplot of Working Day vs Count')
plt.xlabel('Working Day')
plt.ylabel('Count')
plt.show()

# Bivariate Analysis: Relationship between season and count
plt.figure(figsize=(10, 6))
sns.boxplot(data=yulu_data, x='season', y='count')
plt.title('Boxplot of Season vs Count')
plt.xlabel('Season')
plt.ylabel('Count')
plt.show()

# Bivariate Analysis: Relationship between weather and count
plt.figure(figsize=(10, 6))
sns.boxplot(data=yulu_data, x='weather', y='count')
plt.title('Boxplot of Weather vs Count')
plt.xlabel('Weather')
plt.ylabel('Count')
plt.show()

# Correlation matrix
plt.figure(figsize=(12, 8))
sns.heatmap(yulu_data.corr(numeric_only=True), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix')
plt.show()

# Hypothesis Testing: T-test for Working Day
workingday_data = yulu_data[yulu_data['workingday'] == 1]['count']
non_workingday_data = yulu_data[yulu_data['workingday'] == 0]['count']
t_stat, p_value = ttest_ind(workingday_data, non_workingday_data)
print(f'T-test results: t-statistic = {t_stat}, p-value = {p_value}')

# Hypothesis Testing: ANOVA for seasons
season_groups = [yulu_data[yulu_data['season'] == season]['count'] for season in yulu_data['season'].unique()]
f_stat_season, p_value_season = f_oneway(*season_groups)
print(f'ANOVA results for seasons: F-statistic = {f_stat_season}, p-value = {p_value_season}')

# Hypothesis Testing: ANOVA for weather
weather_groups = [yulu_data[yulu_data['weather'] == weather]['count'] for weather in yulu_data['weather'].unique()]
f_stat_weather, p_value_weather = f_oneway(*weather_groups)
print(f'ANOVA results for weather: F-statistic = {f_stat_weather}, p-value = {p_value_weather}')

# Hypothesis Testing: Chi-square test for weather and season
weather_season_table = pd.crosstab(yulu_data['weather'], yulu_data['season'])
chi2_stat, p_value_chi2, dof, expected = chi2_contingency(weather_season_table)
print(f'Chi-square test results: chi2-statistic = {chi2_stat}, p-value = {p_value_chi2}')
```

```
             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
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
```

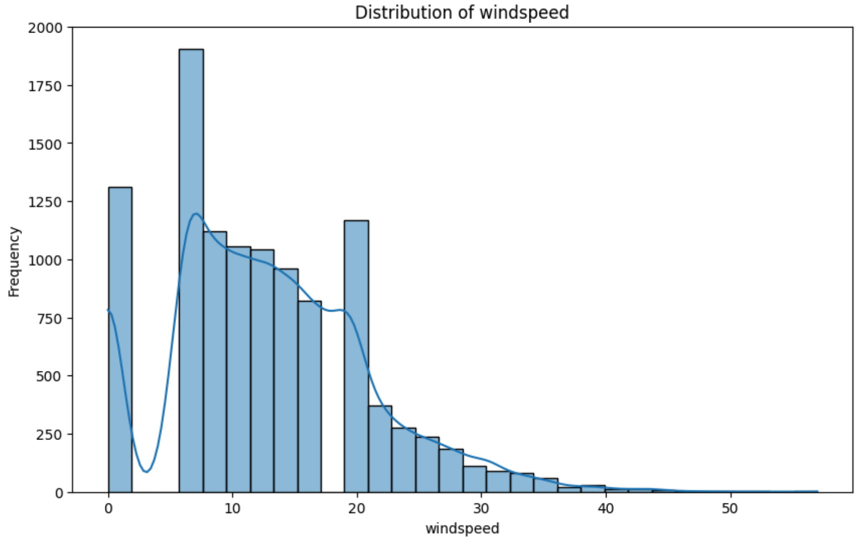




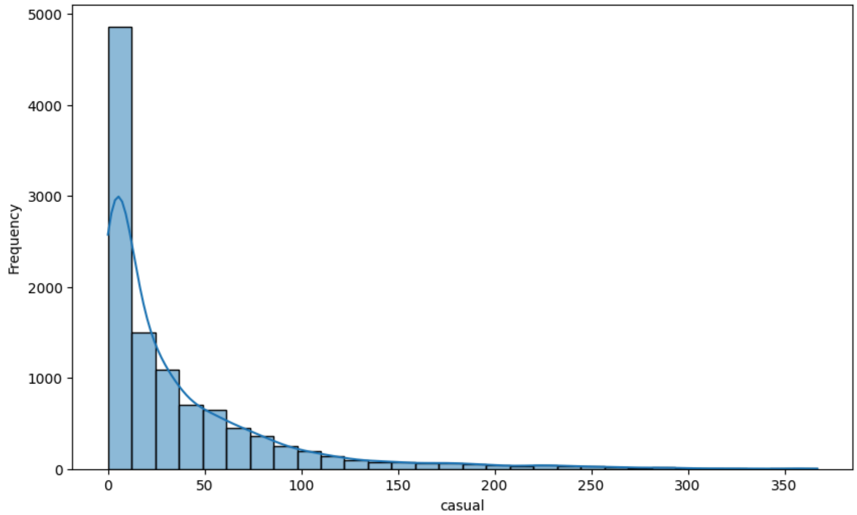




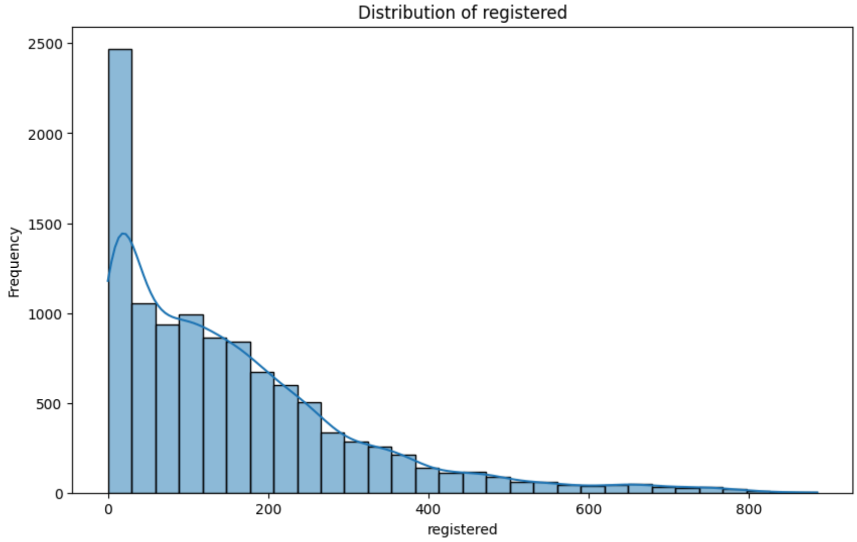
Distribution of windspeed



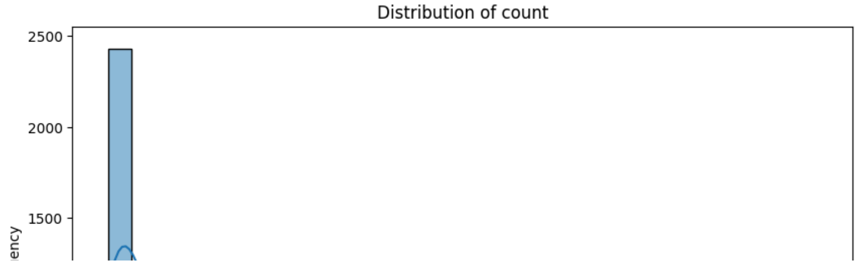
Distribution of casual

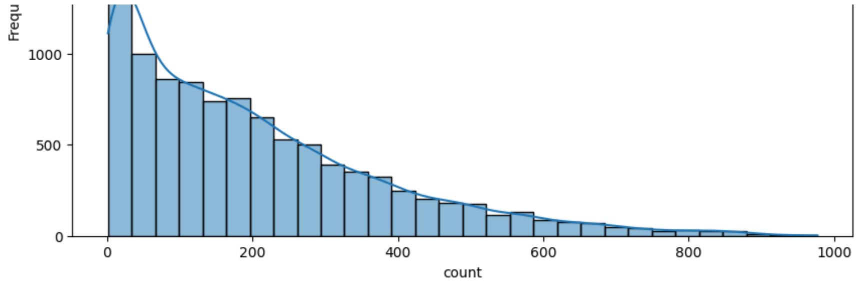


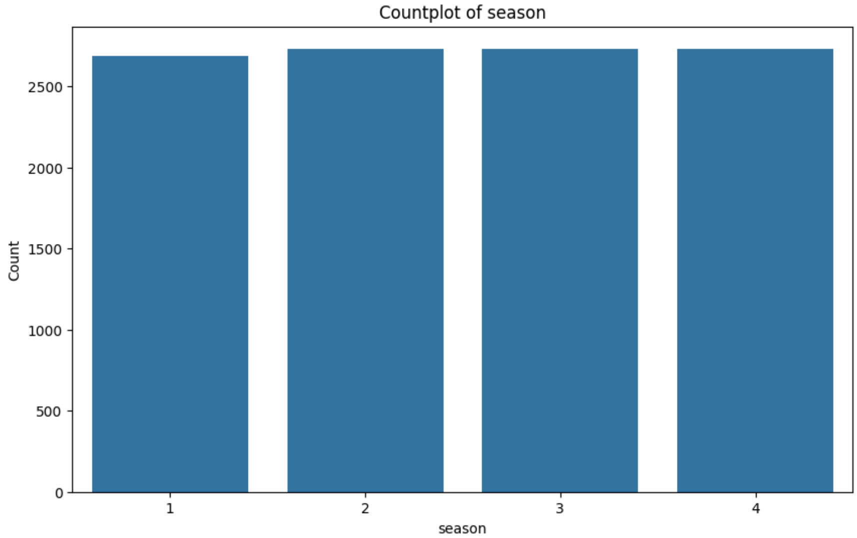
Distribution of registered

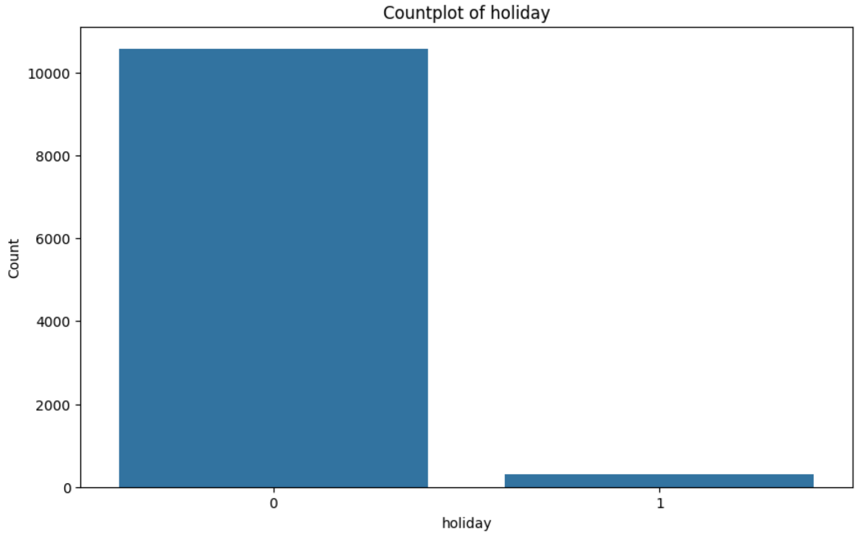


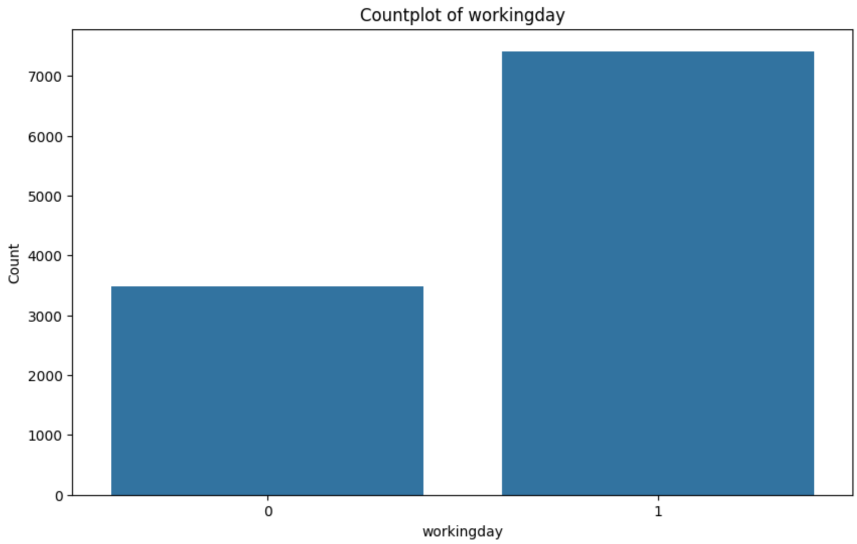
Distribution of count

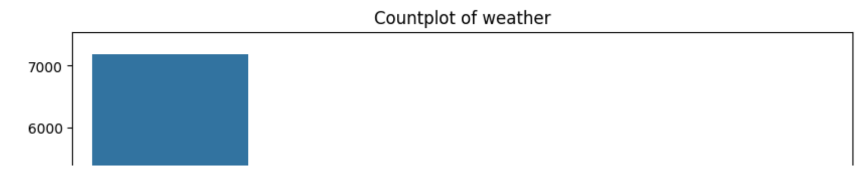


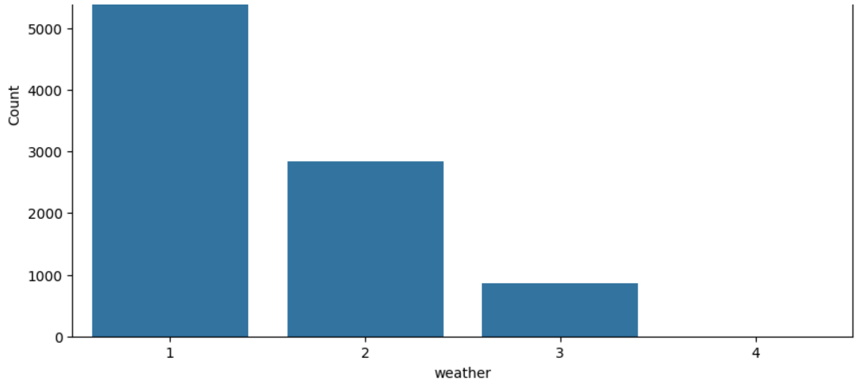

Frequ
1000
500
0
0
200
400
600
800
1000
count

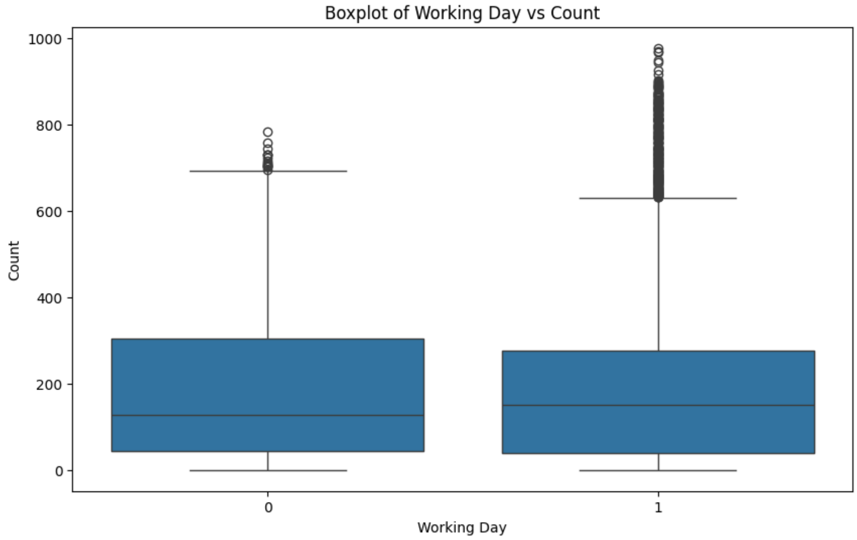

Countplot of season
2500
2000
1500
1000
500
0
Count
1
2
3
4
season

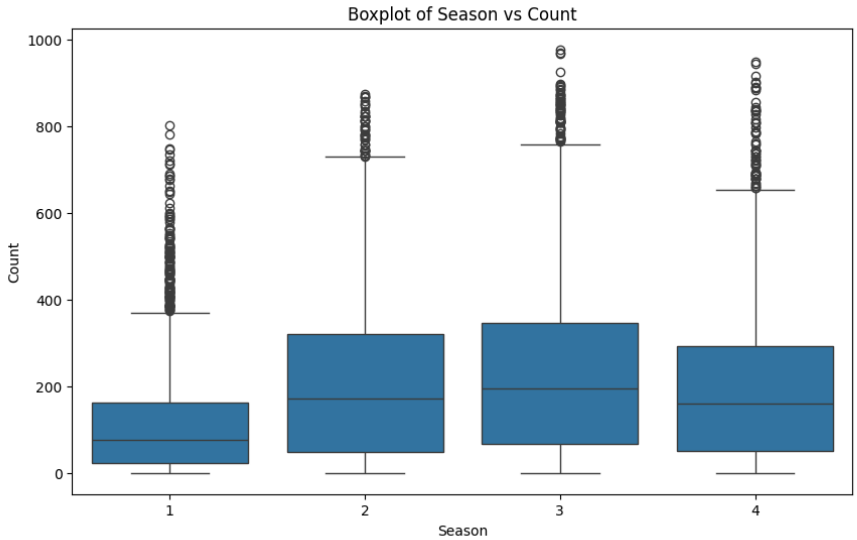

Countplot of holiday
10000
8000
6000
4000
2000
0
Count
0
1
holiday

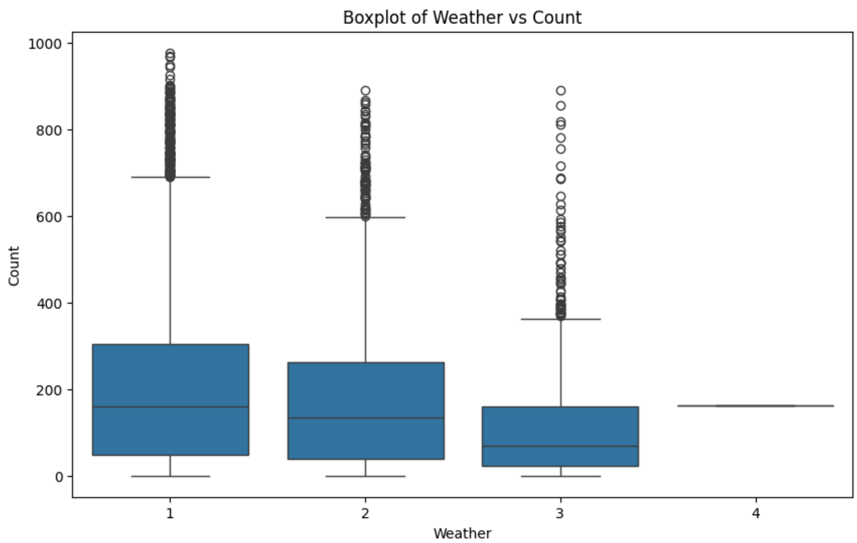

Countplot of workingday
7000
6000
5000
4000
3000
2000
1000
0
Count
0
1
workingday

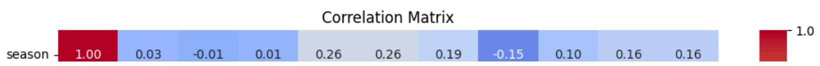

Countplot of weather
7000
6000

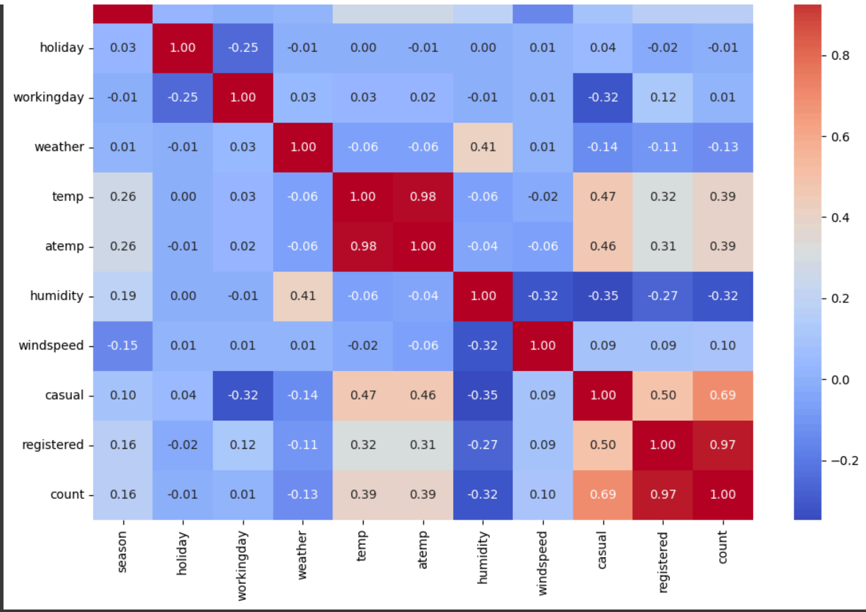

Count
weather
1
2
3
4


Boxplot of Working Day vs Count
Count
Working Day
0
1


Boxplot of Season vs Count
Count
Season
1
2
3
4


Boxplot of Weather vs Count
Count
Weather
1
2
3
4


Correlation Matrix
season
1.00
0.03
-0.01
0.01
0.26
0.26
0.19
-0.15
0.10
0.16
0.16
1.0




```
T-test results: t-statistic = 1.2096277376026694, p-value = 0.22644804226361348
ANOVA results for seasons: F-statistic = 236.94671081032106, p-value = 6.16484338649
ANOVA results for weather: F-statistic = 65.53024112793271, p-value = 5.48206947593
Chi-square test results: chi2-statistic = 49.15865559689363, p-value = 1.54992507368
```