In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from statsmodels.formula.api import ols
import statsmodels.api as sm

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

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

Out[1]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0

In [2]:
```
# Checking the structure of the dataset
data.info()

# Statistical summary of the dataset
data.describe()
```

```
<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
```

Out[2]:

	season	holiday	workingday	weather	temp	a
count	10886.000000	10886.000000	10886.000000	10886.000000	10886.00000	10886.00
mean	2.506614	0.028569	0.680875	1.418427	20.23086	23.65
std	1.116174	0.166599	0.466159	0.633839	7.79159	8.4
min	1.000000	0.000000	0.000000	1.000000	0.82000	0.76
25%	2.000000	0.000000	0.000000	1.000000	13.94000	16.66
50%	3.000000	0.000000	1.000000	1.000000	20.50000	24.24
75%	4.000000	0.000000	1.000000	2.000000	26.24000	31.06
max	4.000000	1.000000	1.000000	4.000000	41.00000	45.45

Here are some observations:

- The dataset contains 10,886 entries and 12 columns.
- The columns are: datetime, season, holiday, workingday, weather, temp, atemp, humidity, windspeed, casual, registered, and count.
- The datetime column is of object type and should be converted to datetime type for analysis.
- There are no missing values in the dataset.
- temp, atemp, humidity, and windspeed are continuous variables.
- season, holiday, workingday, and weather are categorical variables.
- The dependent variable is count.

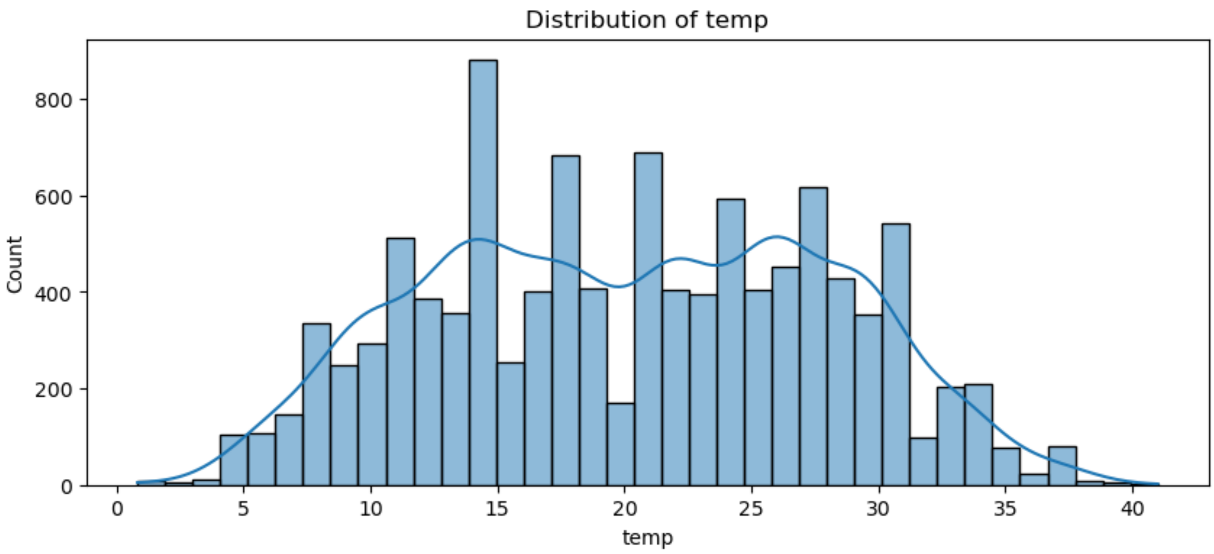
In [3]:

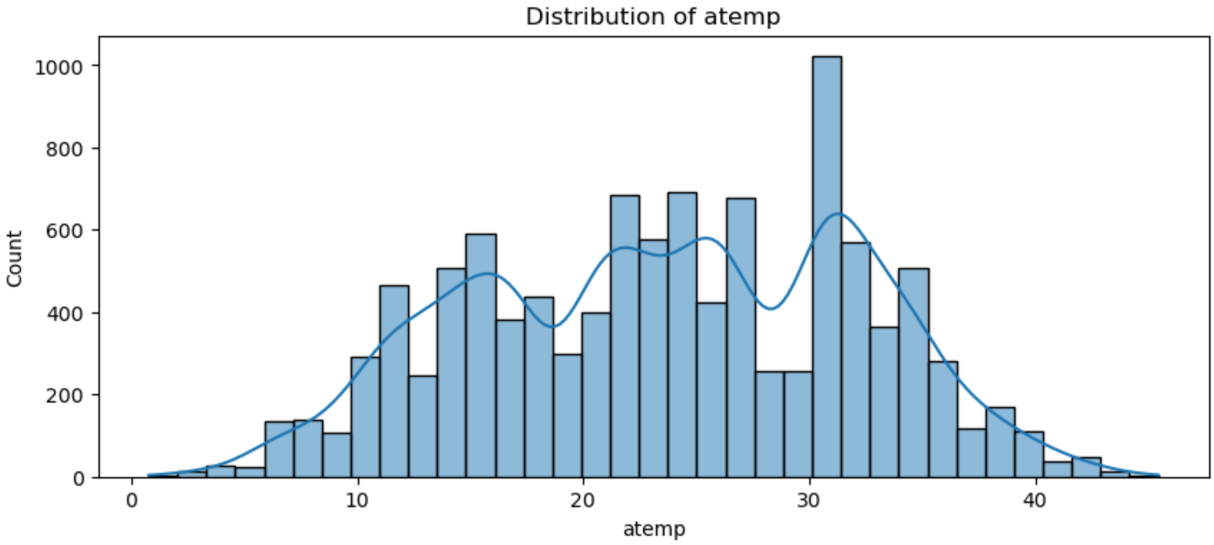
```
data.isna().sum()
```

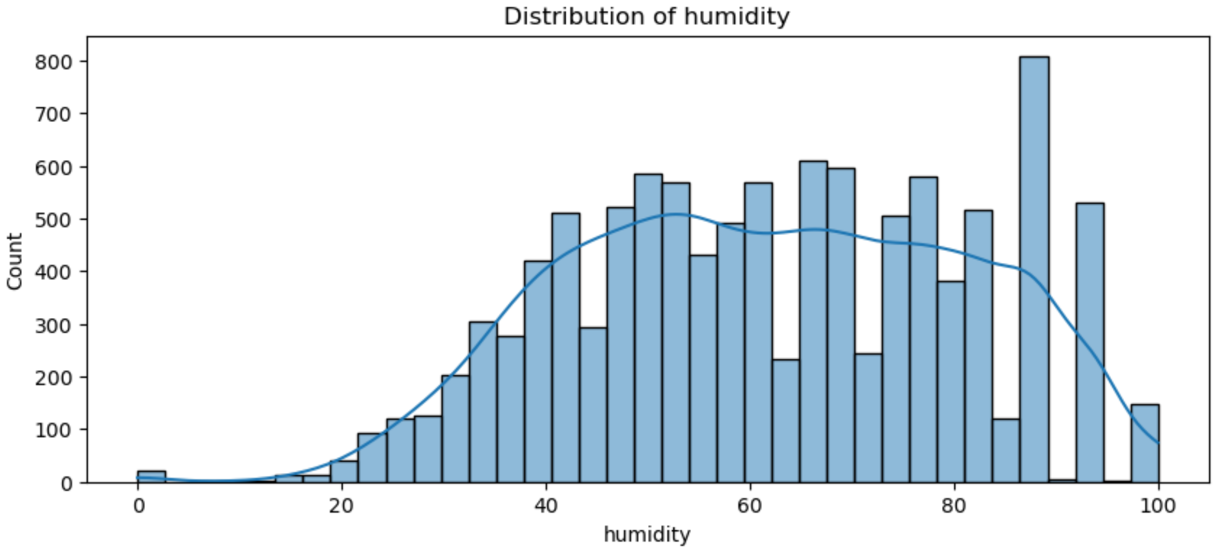
Out[3]:
```
datetime      0
season        0
holiday       0
workingday    0
weather       0
temp          0
atemp         0
humidity      0
windspeed     0
casual        0
registered    0
count         0
dtype: int64
```

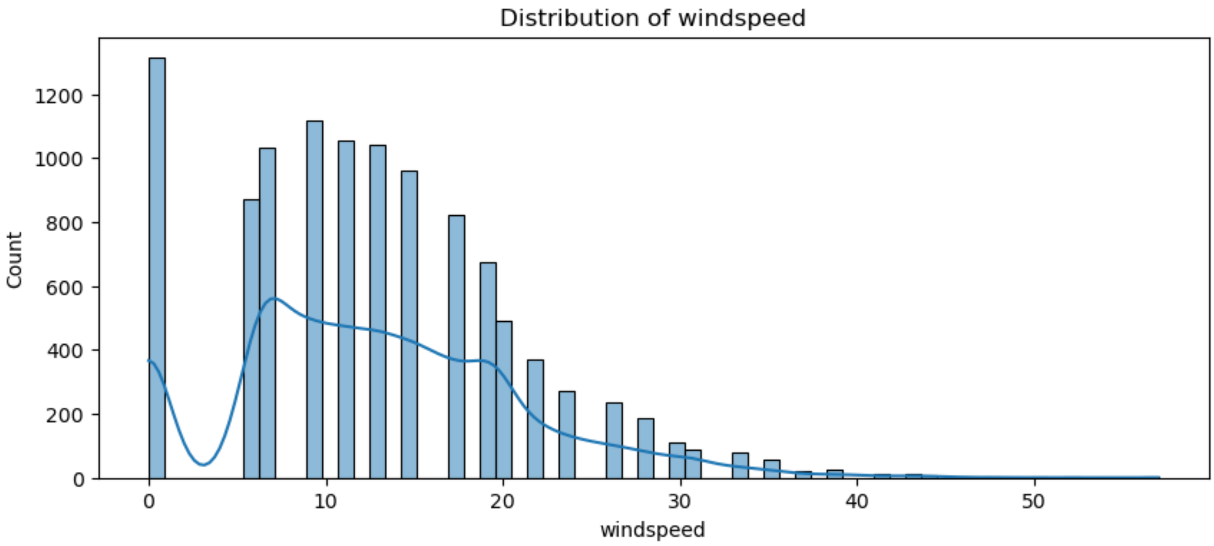
Univariate Analysis

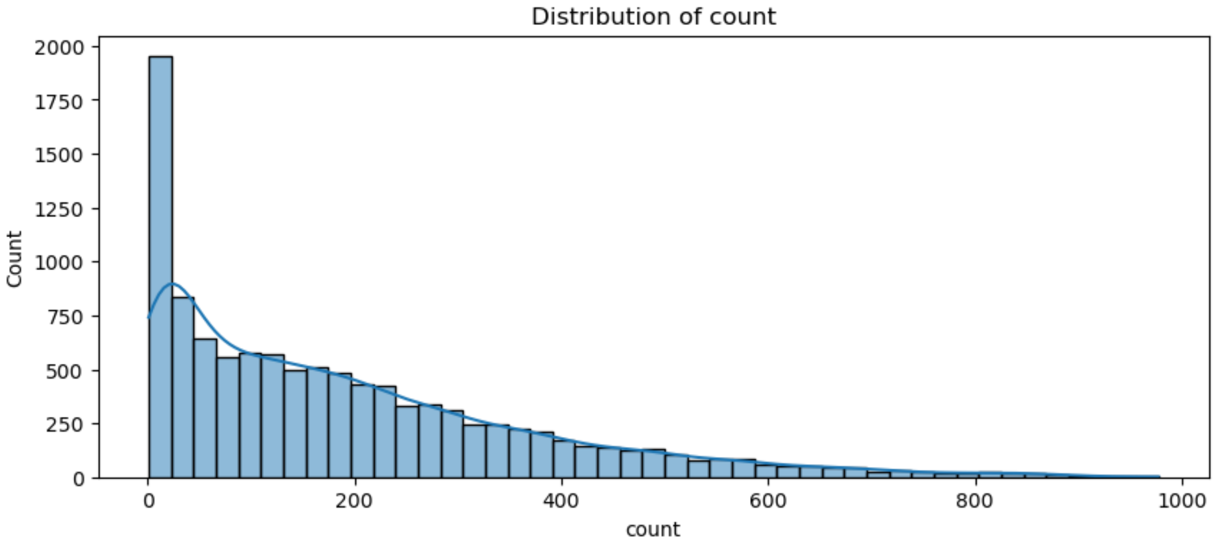
In [10]:
```
# Convert datetime column to datetime type
data['datetime'] = pd.to_datetime(data['datetime'])

# Distribution of continuous variables
continuous_vars = ['temp', 'atemp', 'humidity', 'windspeed', 'count']
for var in continuous_vars:
    plt.figure(figsize=(10, 4))
    sns.histplot(data[var], kde=True)
    plt.title(f'Distribution of {var}')
    plt.show()
```




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


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

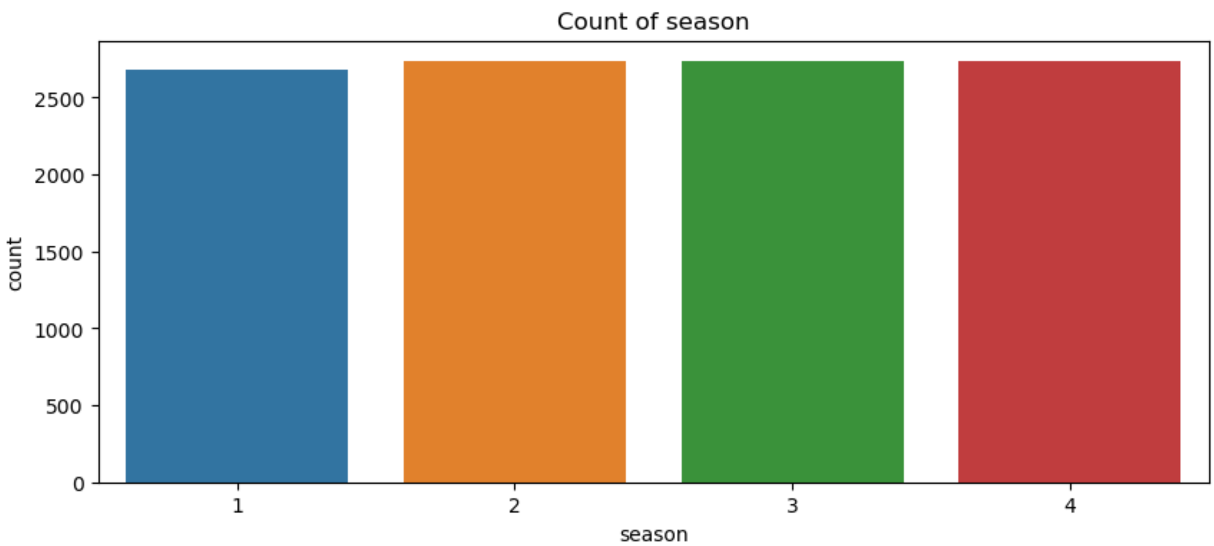

Distribution of windspeed
Count
1200
1000
800
600
400
200
0
0
10
20
30
40
50
windspeed

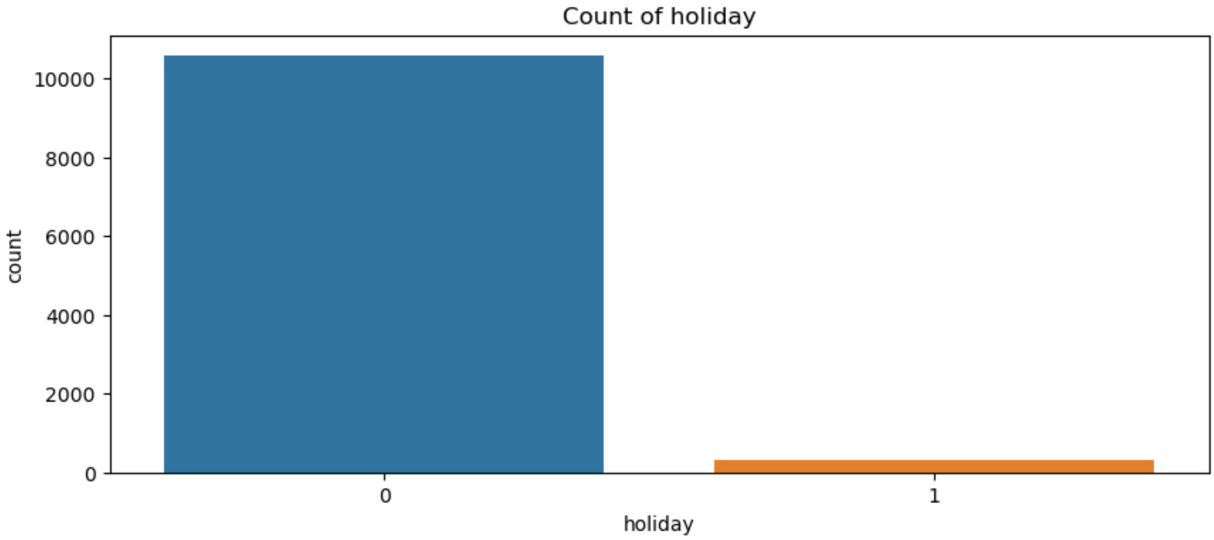


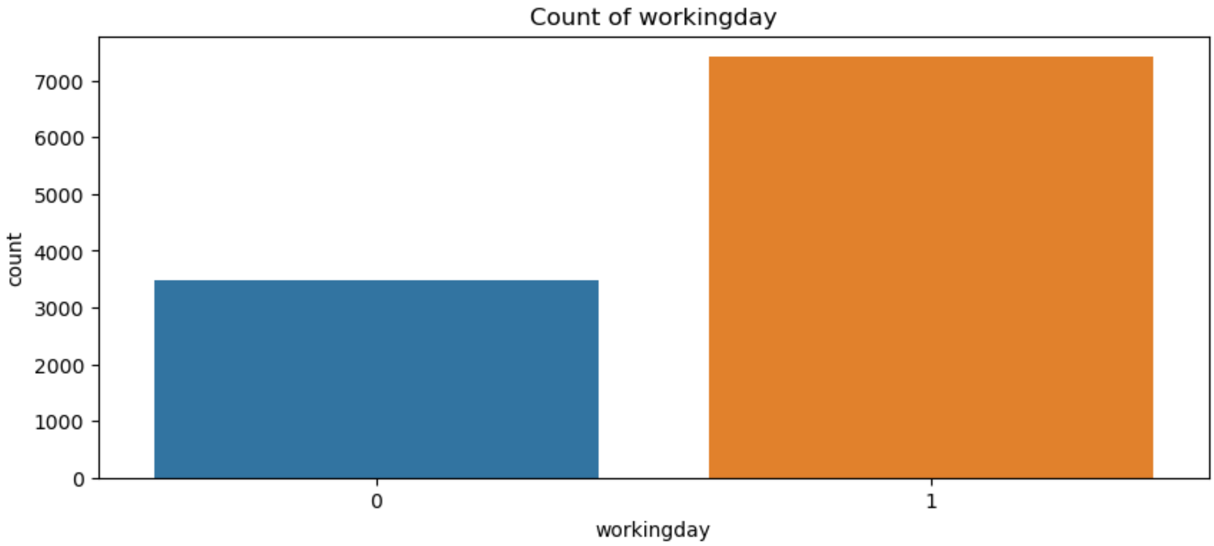
temp and atemp variables are almost normally distributed while (count and wind) are right skewed and humidity is somewhat right skewed.

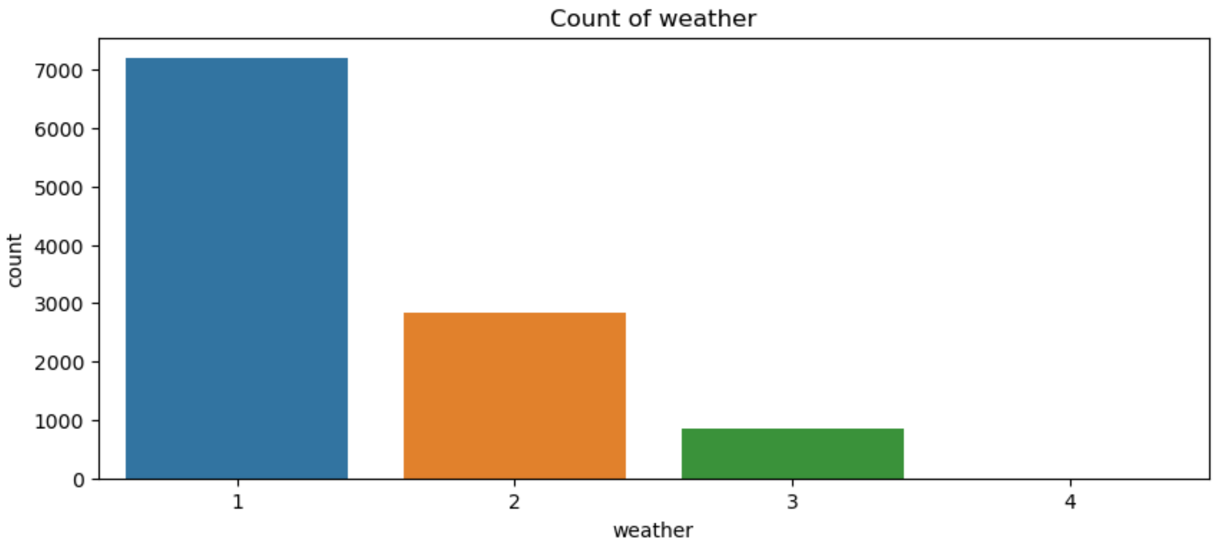
In [11]:

```
# Count plots for categorical variables
categorical_vars = ['season', 'holiday', 'workingday', 'weather']
for var in categorical_vars:
    plt.figure(figsize=(10, 4))
    sns.countplot(x=var, data=data)
    plt.title(f'Count of {var}')
    plt.show()
```




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


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


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

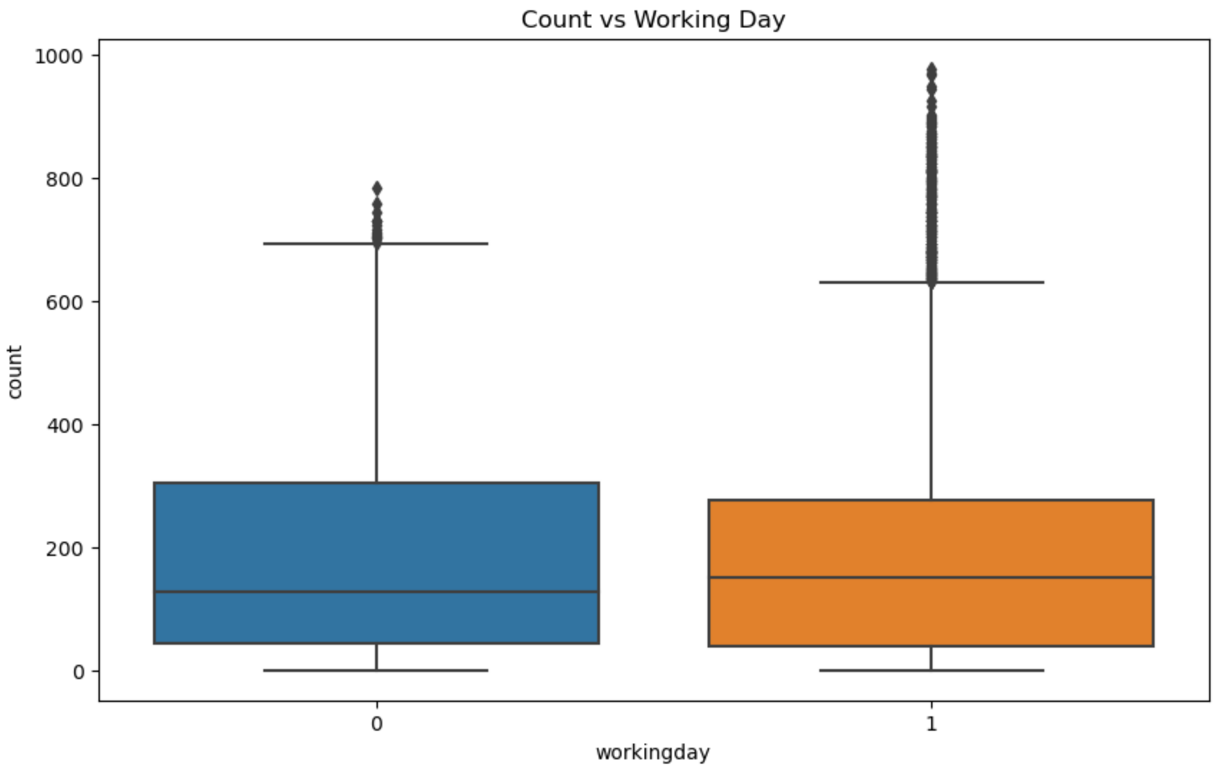
no count in winter season and max count in spring season when the temperatures are neither too hot neither too cold.

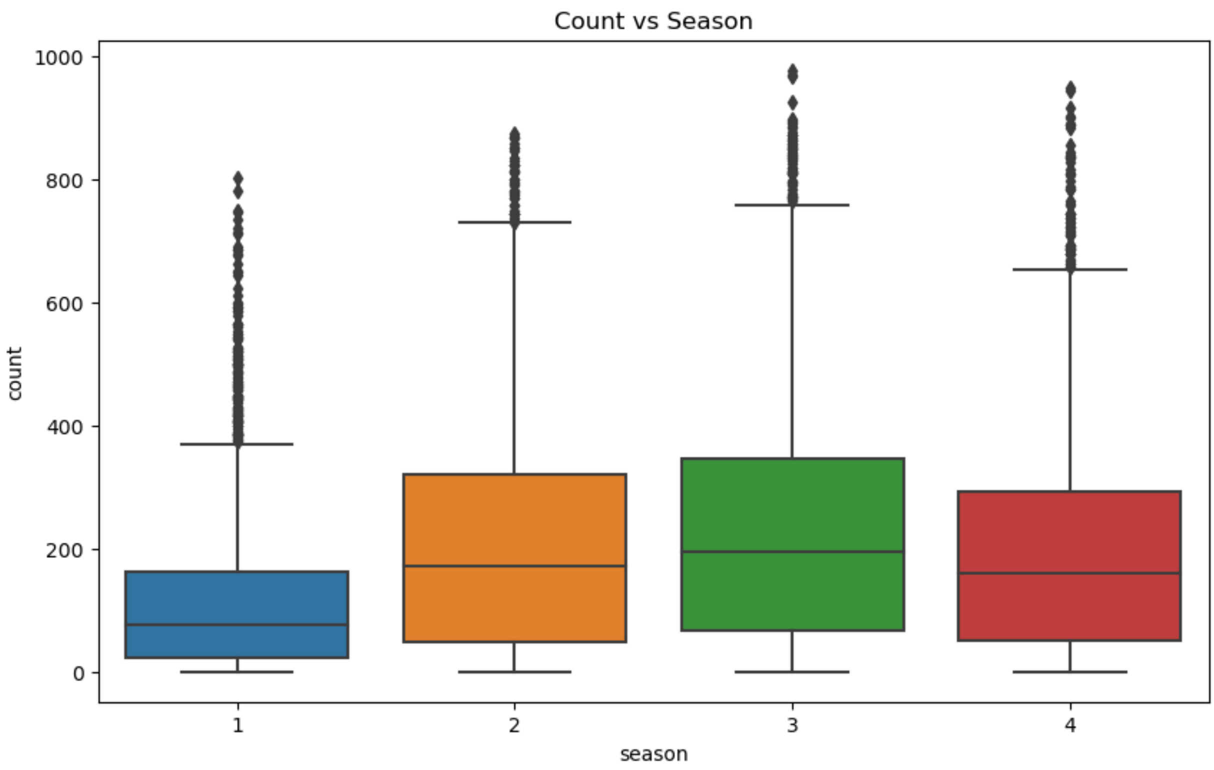
count is almost half on weekend or holidays.

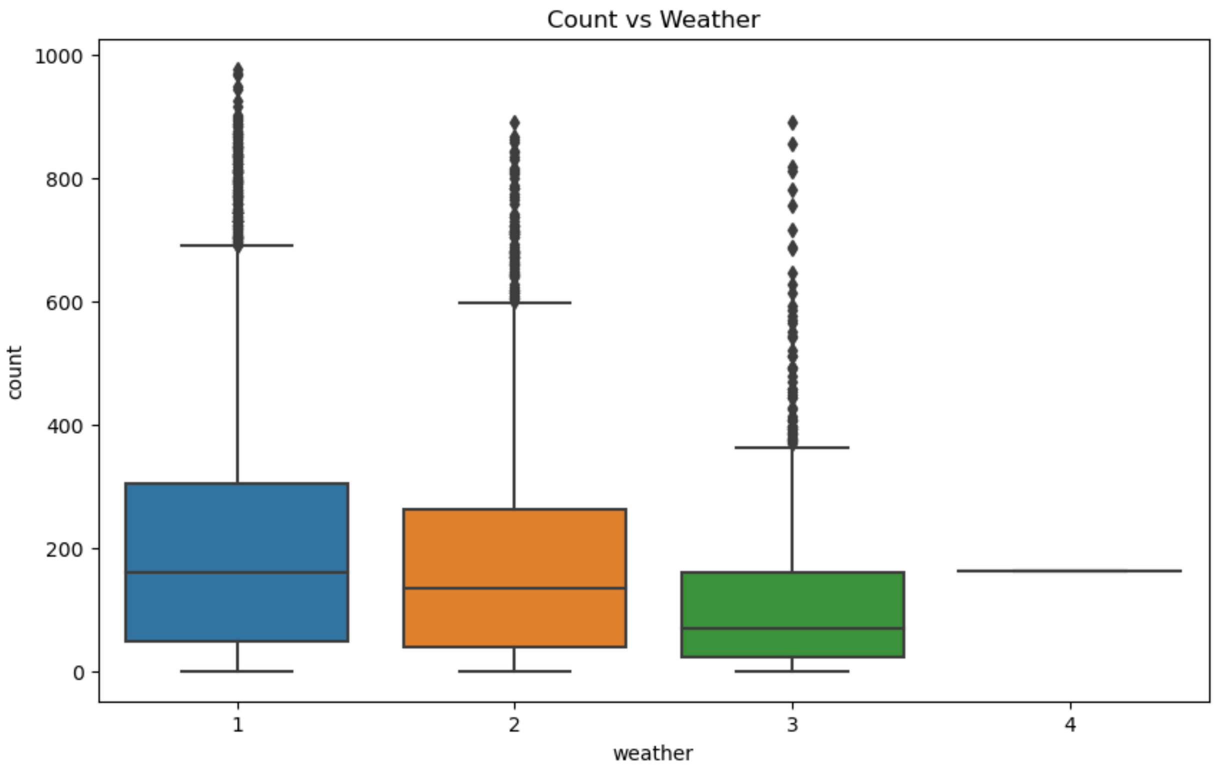
also, it is very less used on government declared holidays, maybe its used for day to day travels to office or shops, etc.

Bivariate Analysis

In [12]:
```
# Relationship between workingday and count
plt.figure(figsize=(10, 6))
sns.boxplot(x='workingday', y='count', data=data)
plt.title('Count vs Working Day')
plt.show()

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

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







Hypothesis Testing

based on above observations we can perform hypothesis testing for :

```
1. working days and non-working days
2. seasons
3. weathers
4. season and weather
```

2-Sample T-Test: Effect of Working Day on Number of Electric Cycles Rented

Null Hypothesis (H0): There is no significant difference in the number of cycles rented on working days and non-working days.

Alternate Hypothesis (H1): There is a significant difference in the number of cycles rented on working days and non-working days.

In [6]:
```
# Separate data into working days and non-working days
workingday_data = data[data['workingday'] == 1]['count']
non_workingday_data = data[data['workingday'] == 0]['count']

# Perform 2-sample t-test
t_stat, p_value = stats.ttest_ind(workingday_data, non_workingday_data)
print(f'T-statistic: {t_stat}, P-value: {p_value}')
```

```
T-statistic: 1.2096277376026694, P-value: 0.22644804226361348
```

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

# Perform 2-sample t-test
t_stat, p_value = stats.ttest_ind(workingday_data, non_workingday_data)
print(f'T-statistic: {t_stat}, P-value: {p_value}')
```

ANOVA: Number of Cycles Rented in Different Seasons and Weathers

For Seasons:

Null Hypothesis (H0): The mean number of cycles rented is the same across all seasons.

Alternate Hypothesis (H1): The mean number of cycles rented is different for at least one season.

In [7]:
```
# Perform ANOVA for seasons
anova_season = ols('count ~ C(season)', data=data).fit()
anova_table = sm.stats.anova_lm(anova_season, typ=2)
print(anova_table)
```

```
                 sum_sq       df           F         PR(>F)
C(season)  2.190083e+07      3.0  236.946711  6.164843e-149
Residual   3.352721e+08  10882.0         NaN            NaN
```

For Weathers:

Null Hypothesis (H0): The mean number of cycles rented is the same across all weathers.

Alternate Hypothesis (H1): The mean number of cycles rented is different for at least one weather type.

In [8]:
```
# Perform ANOVA for weathers
anova_weather = ols('count ~ C(weather)', data=data).fit()
anova_table_weather = sm.stats.anova_lm(anova_weather, typ=2)
print(anova_table_weather)
```

```
                  sum_sq       df          F        PR(>F)
C(weather)  6.338070e+06      3.0  65.530241  5.482069e-42
Residual    3.508348e+08  10882.0        NaN           NaN
```

For holiday:

Null Hypothesis (H0): The mean number of cycles rented is the same across all days.

Alternate Hypothesis (H1): The mean number of cycles rented is different on holidays.

In [14]:
```
# Perform ANOVA for holiday
anova_holiday = ols('count ~ C(holiday)', data=data).fit()
anova_table_holiday = sm.stats.anova_lm(anova_holiday, typ=2)
print(anova_table_holiday)
```

```
                  sum_sq       df         F    PR(>F)
C(holiday)  1.038812e+04      1.0  0.316563  0.573692
Residual    3.571625e+08  10884.0       NaN       NaN
```

For workingday:

Null Hypothesis (H0): The mean number of cycles rented is the same across all days.

Alternate Hypothesis (H1): The mean number of cycles rented is different on non workingday.

In [17]:
```
# Perform ANOVA for workingday
anova_workingday = ols('count ~ C(workingday)', data=data).fit()
anova_table_workingday = sm.stats.anova_lm(anova_workingday, typ=2)
print(anova_table_workingday)
```

```
                     sum_sq       df         F    PR(>F)
C(workingday)  4.801037e+04      1.0  1.463199  0.226448
Residual       3.571249e+08  10884.0       NaN       NaN
```

Chi-Square Test: Dependency of Weather on Season

Null Hypothesis (H0): Weather is independent of the season.

Alternate Hypothesis (H1): Weather is dependent on the season.

In [9]:
```
# Create a contingency table
contingency_table = pd.crosstab(data['season'], data['weather'])

# Perform Chi-Square test
chi2_stat, p_val, dof, ex = stats.chi2_contingency(contingency_table)
print(f'Chi-square statistic: {chi2_stat}, P-value: {p_val}')
```

```
Chi-square statistic: 49.158655596893624, P-value: 1.5499250736864920e-07
```

Interpretation of Results

Based on the p-values obtained from the hypothesis tests, we can make the following decisions:

1. 2-Sample T-Test (Working Day Effect): p-value ≥ 0.05, we fail to reject the null hypothesis, hence there is no significant difference in the number of cycles rented on working days and non-working days.

2a. ANOVA (Season Effects): p-value < 0.05 hence, we reject the null hypothesis and conclude that the mean number of cycles rented differs across different seasons types.

2b. ANOVA (Weather Effects): p-value < 0.05 hence, we reject the null hypothesis and conclude that the mean number of cycles rented differs across different weather types.

2b. ANOVA (holiday): p-value > 0.05 hence, we reject the null hypothesis and conclude that the mean number of cycles rented differs across holiday and non holidays.

2b. ANOVA (workingday): p-value > 0.05 hence, we reject the null hypothesis and conclude that the mean number of cycles rented differs across working and non working days.

3. Chi-Square Test (Dependency of Weather on Season): p-value < 0.05, we reject the null hypothesis and conclude that weather is dependent on the season.

Conclusion

This analysis provides insights into the factors affecting the demand for Yulu electric cycles, helping the company understand the impact of working days, seasons, and weather on rentals. Based on the results, Yulu can tailor its strategies to optimize operations and enhance customer satisfaction.

In [ ]: