Define Problem Statement and perform Exploratory Data Analysis

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import ttest_ind, f_oneway, chi2_contingency
```

In [ ]:
```
url='https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/428/origina
```

In [ ]:
```
df=pd.read_csv(url)
```

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

Out[ ]:

	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 [ ]:
```
df.columns
```

Out[ ]:
```
Index(['datetime', 'season', 'holiday', 'workingday', 'weather', 'temp',
       'atemp', 'humidity', 'windspeed', 'casual', 'registered', 'count'],
      dtype='object')
```

In [ ]:
```
# Initial exploration
print(df.info())
print(df.head())
```

```
<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
              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  
```

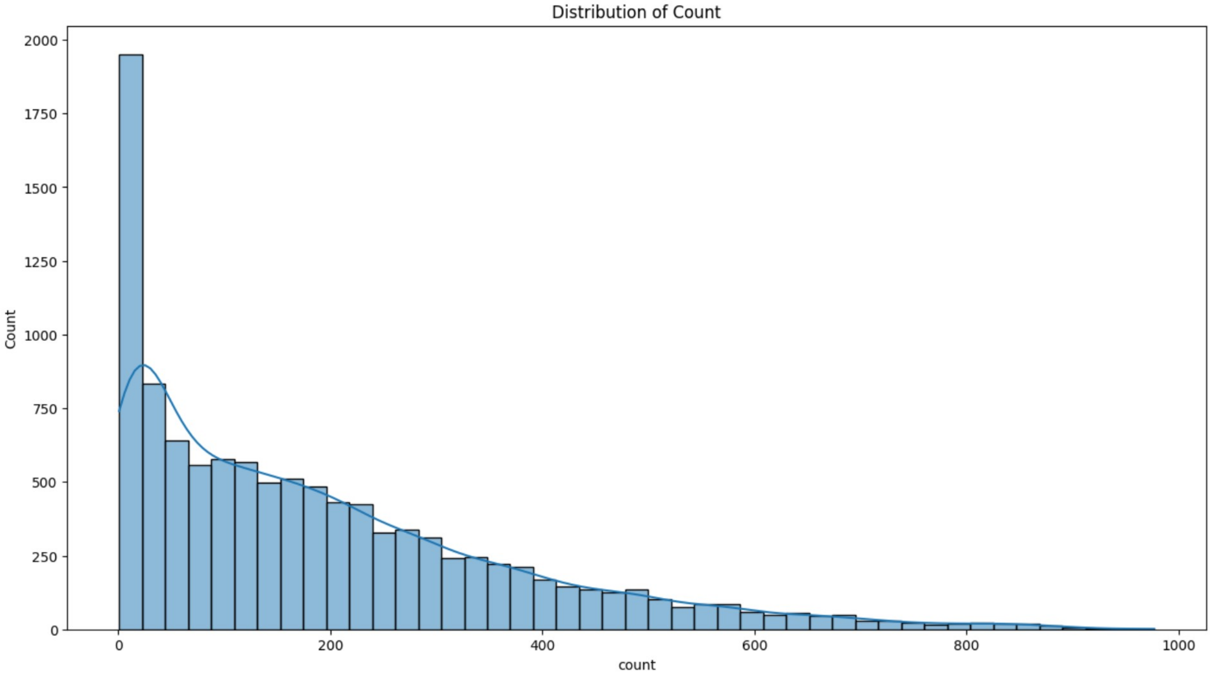
In [ ]:
```
# Check for missing values
print(df.isnull().sum())
```

```
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
```

In [ ]:
```
# Convert categorical variables to 'category' type
df['season'] = df['season'].astype('category')
df['holiday'] = df['holiday'].astype('category')
df['workingday'] = df['workingday'].astype('category')
df['weather'] = df['weather'].astype('category')
```

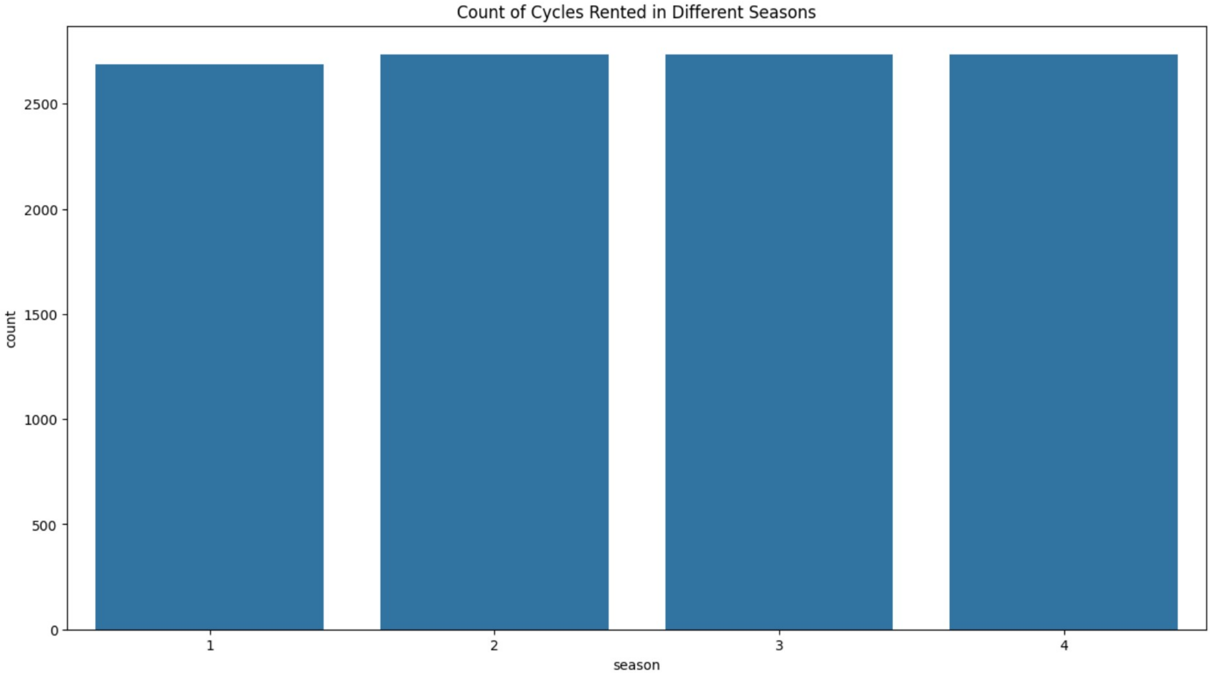
In [ ]:

```
# Univariate Analysis
# Distribution plots for continuous variables
plt.figure(figsize=(15, 8))
sns.histplot(df['count'], kde=True)
plt.title('Distribution of Count')
plt.show()
```



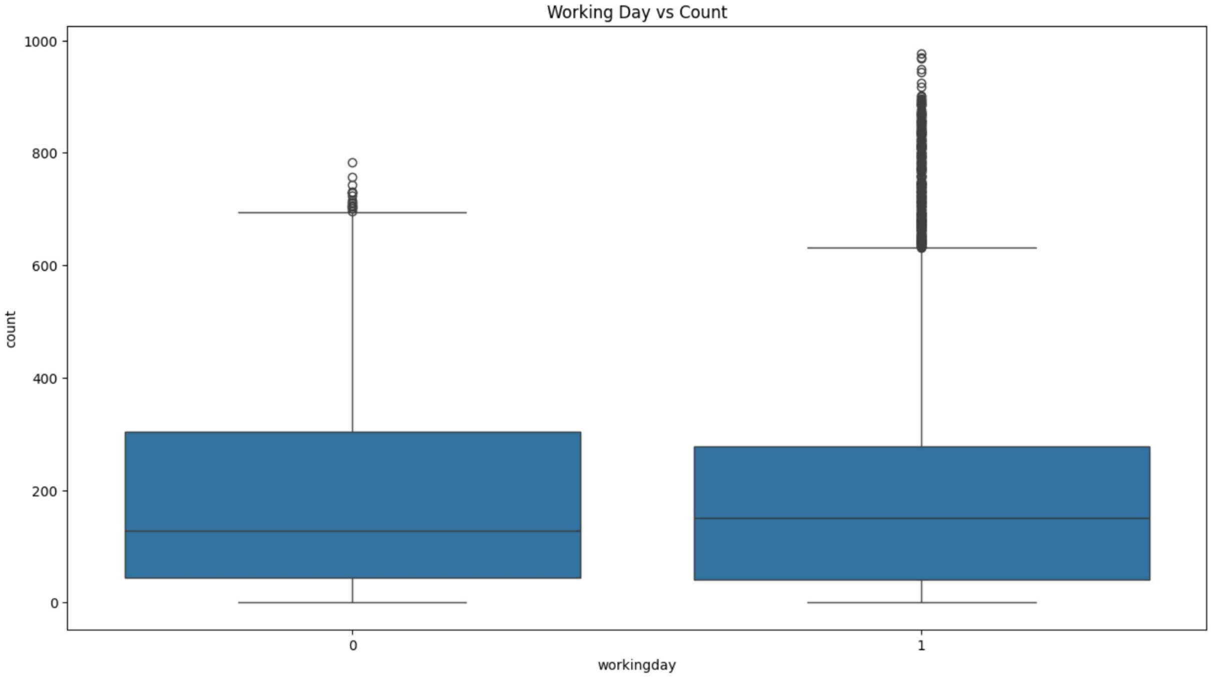
In [ ]:

```
# Bar plots for categorical variables
plt.figure(figsize=(15, 8))
sns.countplot(x='season', data=df)
plt.title('Count of Cycles Rented in Different Seasons')
plt.show()
```



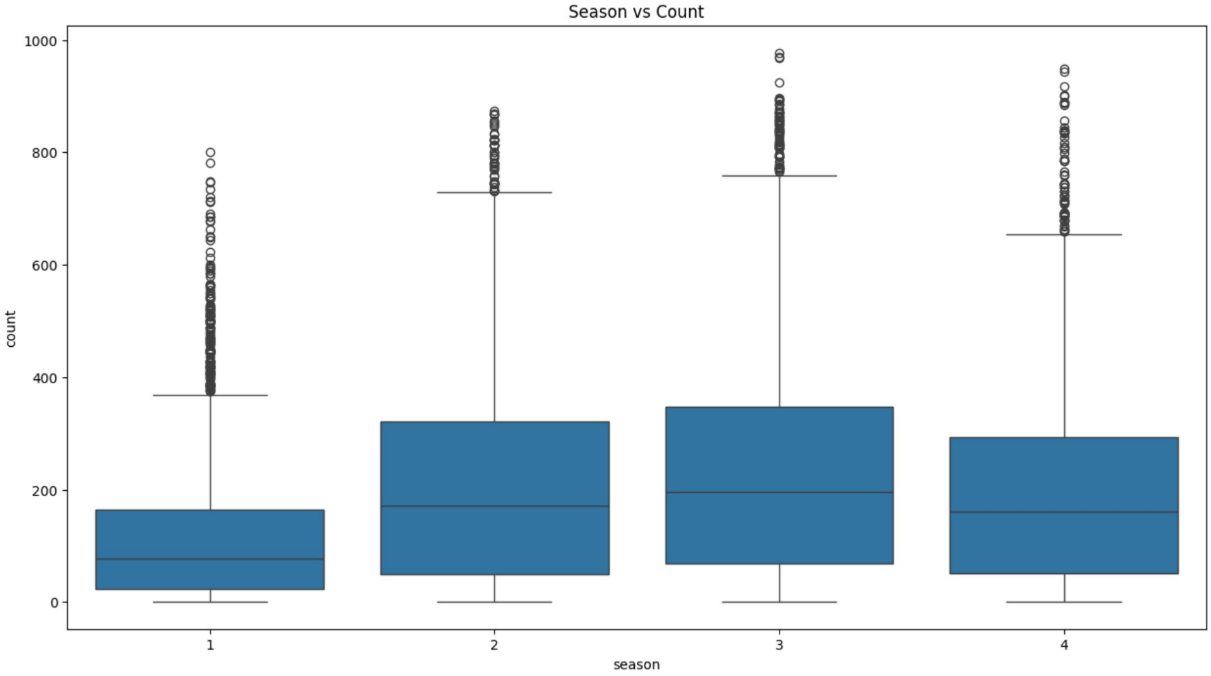
In [ ]:

```
# Bivariate Analysis
# Relationship between workingday and count
plt.figure(figsize=(15, 8))
sns.boxplot(x='workingday', y='count', data=df)
plt.title('Working Day vs Count')
plt.show()
```

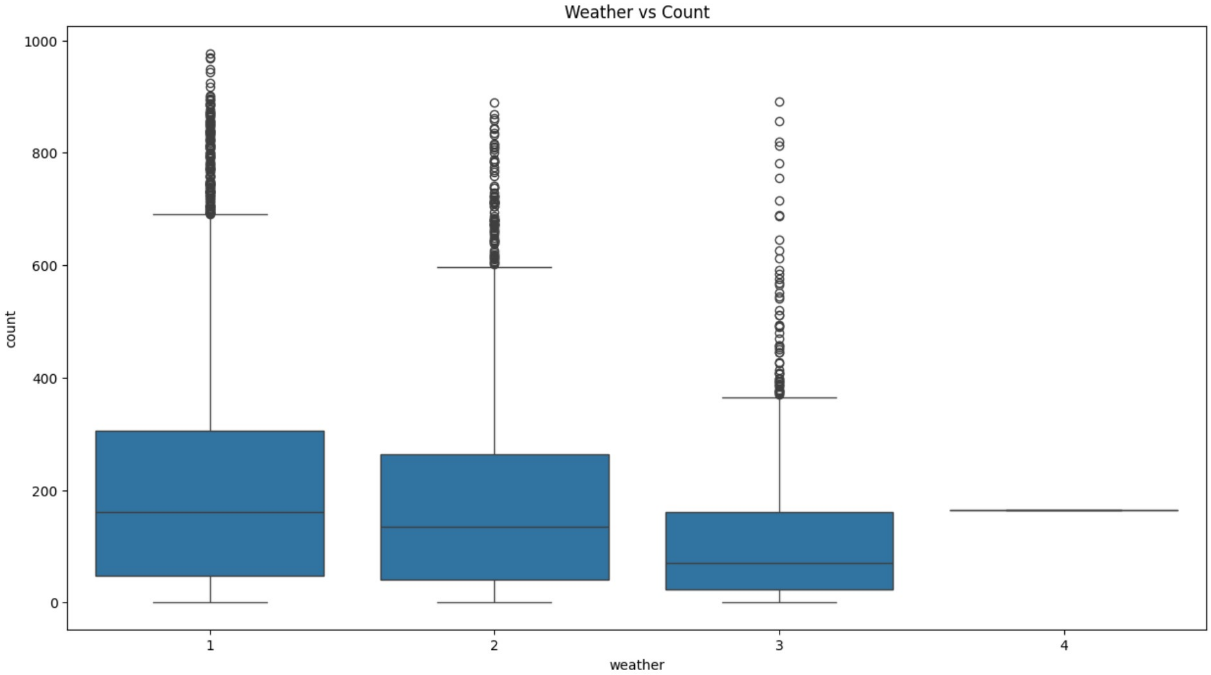


In [ ]:

```
# Relationship between season and count
plt.figure(figsize=(15, 8))
sns.boxplot(x='season', y='count', data=df)
plt.title('Season vs Count')
plt.show()
```



```
In [ ]: # Relationship between weather and count
        plt.figure(figsize=(15, 8))
        sns.boxplot(x='weather', y='count', data=df)
        plt.title('Weather vs Count')
        plt.show()
```



2. Hypothesis Testing (30 Points):

```
In [ ]: # Hypothesis Testing
        # 2-Sample T-Test for workingday and count
        workingday_yes = df[df['workingday'] == 1]['count']
        workingday_no = df[df['workingday'] == 0]['count']
        t_stat, p_value = ttest_ind(workingday_yes, workingday_no)
        print(f"2-Sample T-Test p-value: {p_value}")
```

```
2-Sample T-Test p-value: 0.22644804226361348
```

```
In [ ]: # ANOVA for weather and count, season and count
        weather_groups = [df['count'][df['weather'] == i] for i in df['weather'].unique()]
        f_stat_weather, p_value_weather = f_oneway(*weather_groups)
        print(f"ANOVA for Weather vs Count p-value: {p_value_weather}")
```

```
ANOVA for Weather vs Count p-value: 5.482069475935669e-42
```

```
In [ ]: season_groups = [df['count'][df['season'] == i] for i in df['season'].unique()]
        f_stat_season, p_value_season = f_oneway(*season_groups)
        print(f"ANOVA for Season vs Count p-value: {p_value_season}")
```

```
ANOVA for Season vs Count p-value: 6.164843386499654e-149
```

```
In [ ]: # Chi-square test for weather and season dependency
        contingency_table = pd.crosstab(df['weather'], df['season'])
        chi2_stat, p_value_chi2, _, _ = chi2_contingency(contingency_table)
        print(f"Chi-square test p-value: {p_value_chi2}")
```

```
Chi-square test p-value: 1.5499250736864862e-07
```