```
import pandas as pd

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

data.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
```

```
data.head()
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0	3	13	16
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0	8	32	40
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0	5	27	32
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0	3	10	13
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0	0	1	1

```
data['count'].describe()
```

```
count    10886.000000
mean       191.574132
std        181.144454
min          1.000000
25%         42.000000
50%        145.000000
75%        284.000000
max        977.000000
Name: count, dtype: float64
```

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

```
# Extract hour, day, month, and year from datetime
data['hour'] = data['datetime'].dt.hour
data['day'] = data['datetime'].dt.day
data['month'] = data['datetime'].dt.month
data['year'] = data['datetime'].dt.year
```

```
data.head()
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count	hour	day	month	yea
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0	3	13	16	0	1	1	201
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0	8	32	40	1	1	1	201
	2011-01															

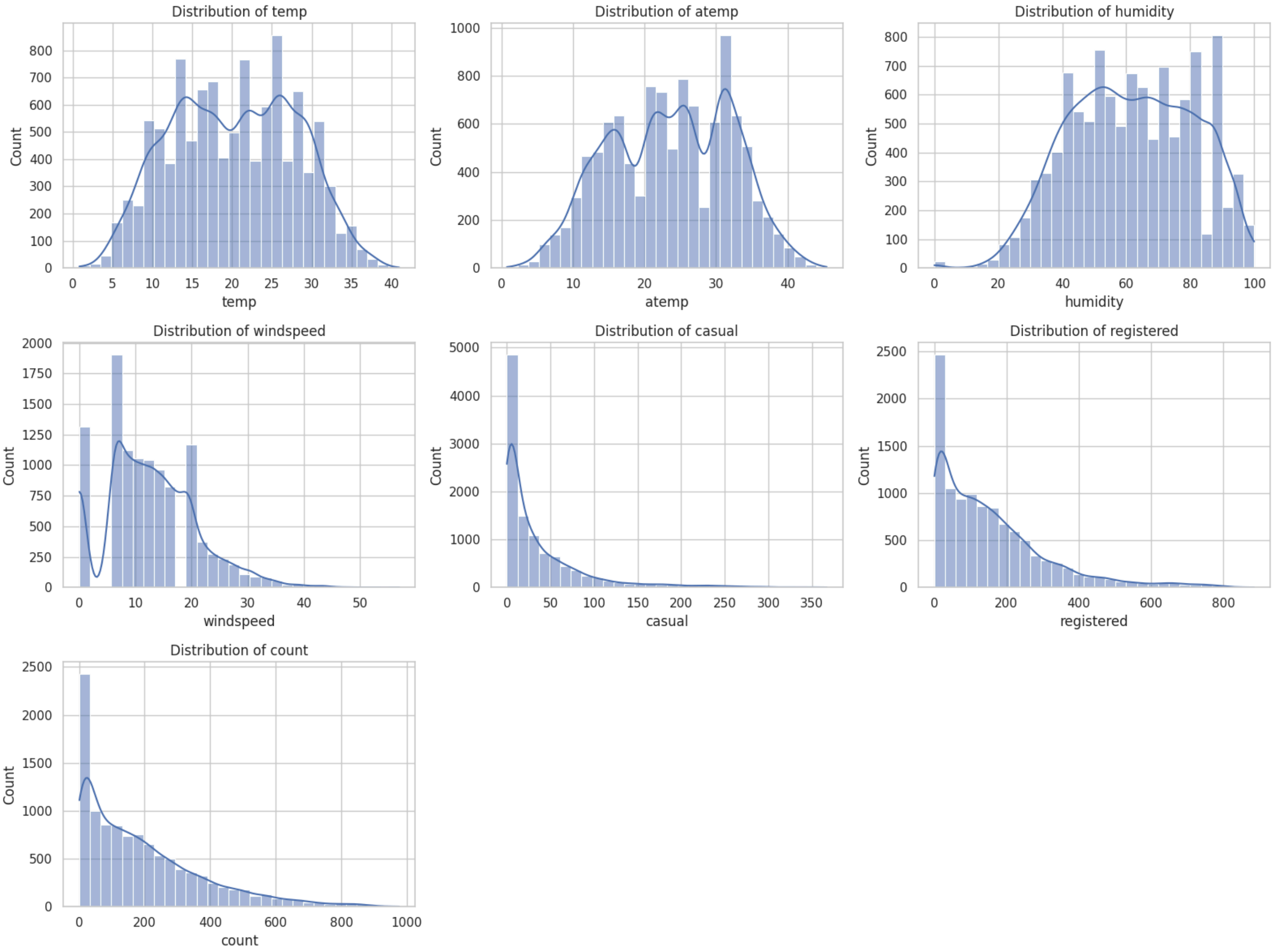
```
data["year"].unique()
```

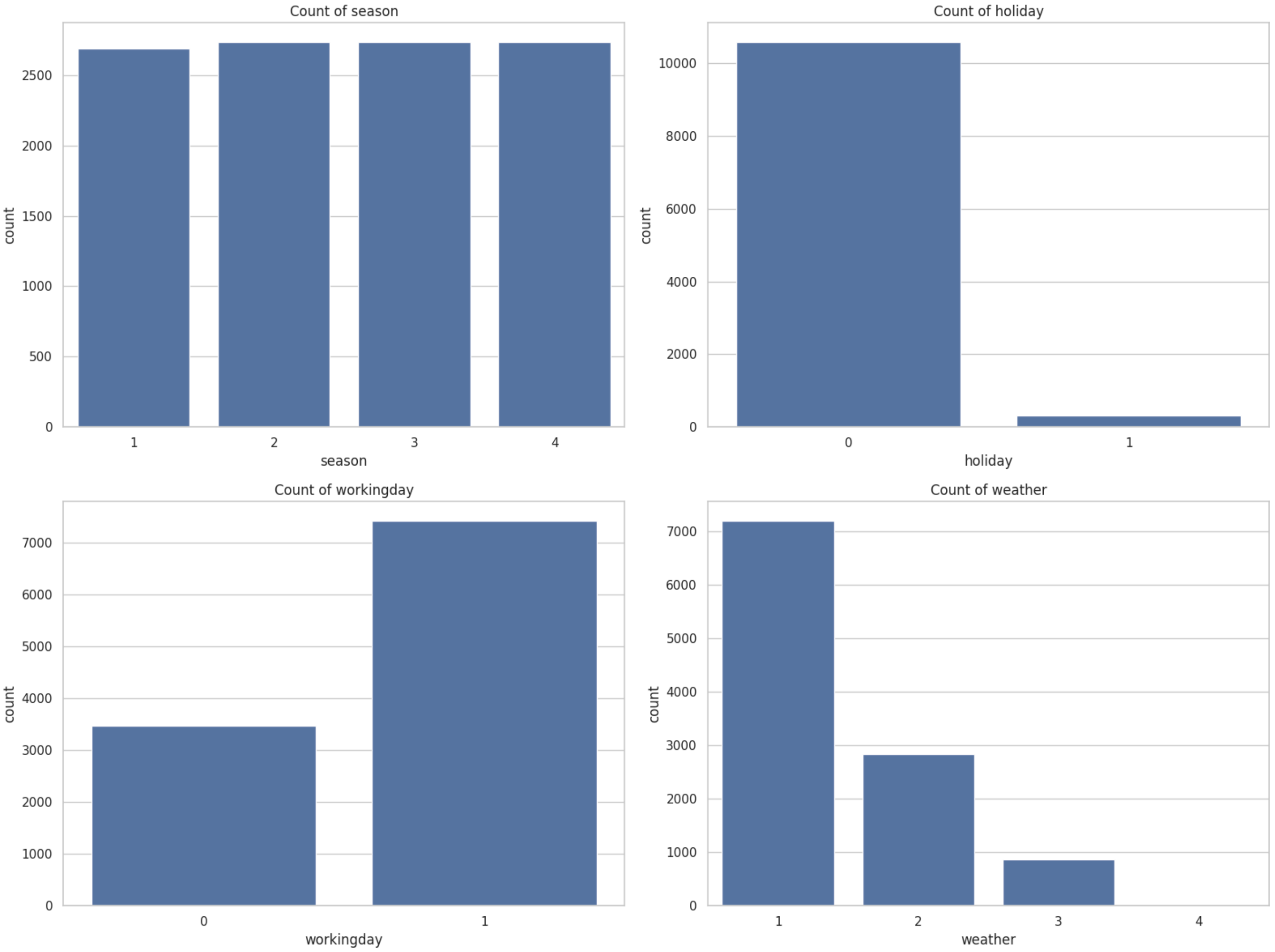
```
array([2011, 2012], dtype=int32)
```

```
# Convert categorical variables to 'category' data type
categorical_columns = ['season', 'holiday', 'workingday', 'weather']
for col in categorical_columns:
    data[col] = data[col].astype('category')


# Verify the conversion
print(data.info())
```

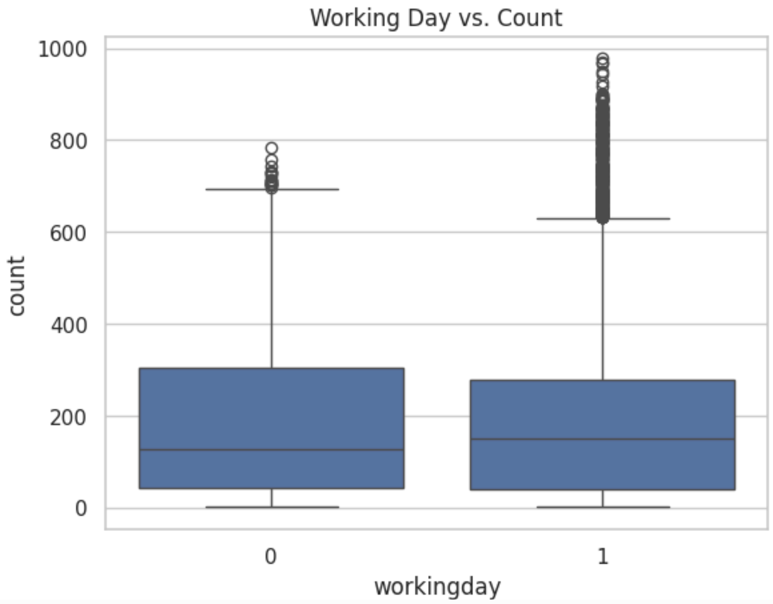
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   datetime    10886 non-null  datetime64[ns]
 1   season      10886 non-null  category
 2   holiday     10886 non-null  category
 3   workingday  10886 non-null  category
 4   weather     10886 non-null  category
 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
 12  hour        10886 non-null  int32
 13  day         10886 non-null  int32
 14  month       10886 non-null  int32
 15  year        10886 non-null  int32
dtypes: category(4), datetime64[ns](1), float64(3), int32(4), int64(4)
memory usage: 893.8 KB
None
```

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

# Set the style of the visualizations
sns.set(style="whitegrid")

# Continuous variables
continuous_vars = ['temp', 'atemp', 'humidity', 'windspeed', 'casual', 'registered', 'count']

# Plot distribution for continuous variables
plt.figure(figsize=(16, 12))
for i, var in enumerate(continuous_vars, 1):
    plt.subplot(3, 3, i)
    sns.histplot(data[var], kde=True, bins=30)
    plt.title(f'Distribution of {var}')
plt.tight_layout()
plt.savefig('histograms.png')
plt.show()

# Categorical variables
categorical_vars = ['season', 'holiday', 'workingday', 'weather']

# Plot bar plots for categorical variables
plt.figure(figsize=(16, 12))
for i, var in enumerate(categorical_vars, 1):
    plt.subplot(2, 2, i)
    sns.countplot(data=data, x=var)
    plt.title(f'Count of {var}')
plt.tight_layout()
plt.savefig('barplots.png')
plt.show()
```

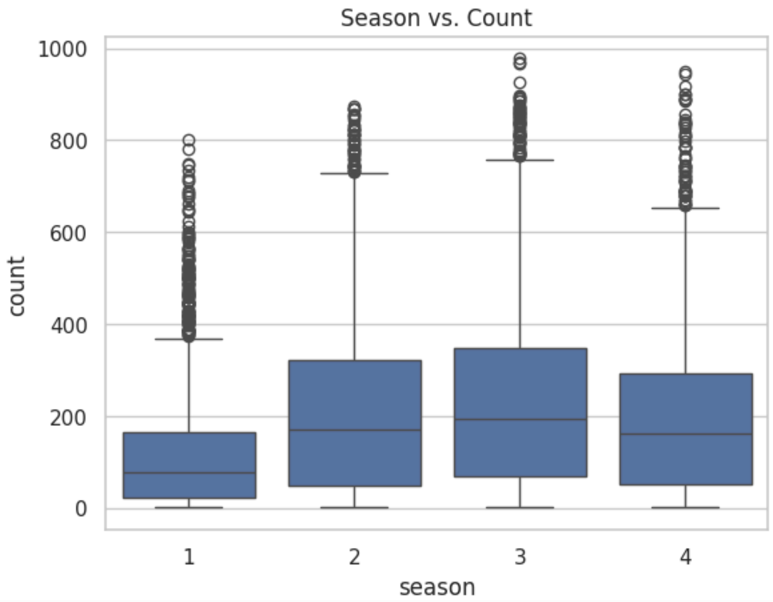

Distribution of temp
Count
temp
0 100 200 300 400 500 600 700 800
0 5 10 15 20 25 30 35 40
Distribution of atemp
Count
atemp
0 200 400 600 800 1000
0 10 20 30 40
Distribution of humidity
Count
humidity
0 100 200 300 400 500 600 700 800
0 20 40 60 80 100
Distribution of windspeed
Count
windspeed
0 250 500 750 1000 1250 1500 1750 2000
0 10 20 30 40 50
Distribution of casual
Count
casual
0 1000 2000 3000 4000 5000
0 50 100 150 200 250 300 350
Distribution of registered
Count
registered
0 500 1000 1500 2000 2500
0 200 400 600 800
Distribution of count
Count
count
0 500 1000 1500 2000 2500
0 200 400 600 800 1000


Count of season
count
season
0 500 1000 1500 2000 2500
1 2 3 4
Count of holiday
count
holiday
0 2000 4000 6000 8000 10000
0 1
Count of workingday
count
workingday
0 1000 2000 3000 4000 5000 6000 7000
0 1
Count of weather
count
weather
0 1000 2000 3000 4000 5000 6000 7000
1 2 3 4

```
sns.boxplot(x='workingday', y='count', data=data)
plt.title('Working Day vs. Count')
plt.savefig('WorkingdayVsCount.png')
plt.show()
```



```
sns.boxplot(x='season', y='count', data=data)
plt.title('Season vs. Count')
plt.savefig('SeasonvsCount.png')
plt.show()
```



```
sns.boxplot(x='weather', y='count', data=data)
plt.title('Weather vs. Count')
plt.savefig('WeatherVsCount.png')
plt.show()
```