```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
```

```
! gdown "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/428/original/bike_sharing.csv?1642089089"
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/428/original/bike_sharing.csv?1642089089
To: /content/bike_sharing.csv?1642089089
100% 648k/648k [00:00<00:00, 1.47MB/s]
```

```
df = pd.read_csv("bike_sharing.csv?1642089089")
```

```
# Problem Statement
'''
1. Which variables are significant in predicting the demand for shared electric cycles in the Indian market?
2. How well those variables describe the electric cycle demands?
'''
```

```
'\n1. Which variables are significant in predicting the demand for shared electric cycles in the Indian market?\n2.
How well those variables describe the electric cycle demands?\n'
```

```
df.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

```
df.describe()
```

	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
count	10886.000000	10886.000000	10886.000000	10886.000000	10886.00000	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000
mean	2.506614	0.028569	0.680875	1.418427	20.23086	23.655084	61.886460	12.799395	36.021955	155.552177	191.574132
std	1.116174	0.166599	0.466159	0.633839	7.79159	8.474601	19.245033	8.164537	49.960477	151.039033	181.144454
min	1.000000	0.000000	0.000000	1.000000	0.82000	0.760000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	2.000000	0.000000	0.000000	1.000000	13.94000	16.665000	47.000000	7.001500	4.000000	36.000000	42.000000
50%	3.000000	0.000000	1.000000	1.000000	20.50000	24.240000	62.000000	12.998000	17.000000	118.000000	145.000000

```
print(f"# rows: {df.shape[0]} \n# columns: {df.shape[1]}")
```

```
# rows: 10886
# columns: 12
```

```
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
```

```
df['datetime'] = pd.to_datetime(df['datetime'])
```

```
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  datetime64[ns]
 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: datetime64[ns](1), float64(3), int64(8)
memory usage: 1020.7 KB
```

```
cat_cols= ['season', 'holiday', 'workingday', 'weather']
for col in cat_cols:
    df[col] = df[col].astype('object')
```

```
df.iloc[:, 1:].describe(include='all')
```

	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
count	10886.0	10886.0	10886.0	10886.0	10886.00000	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000
unique	4.0	2.0	2.0	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	4.0	0.0	1.0	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freq	2734.0	10575.0	7412.0	7192.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	20.23086	23.655084	61.886460	12.799395	36.021955	155.552177	191.574132
std	NaN	NaN	NaN	NaN	7.79159	8.474601	19.245033	8.164537	49.960477	151.039033	181.144454
min	NaN	NaN	NaN	NaN	0.82000	0.760000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	NaN	NaN	NaN	NaN	13.94000	16.665000	47.000000	7.001500	4.000000	36.000000	42.000000
50%	NaN	NaN	NaN	NaN	20.50000	24.240000	62.000000	12.998000	17.000000	118.000000	145.000000
75%	NaN	NaN	NaN	NaN	26.24000	31.060000	77.000000	16.997900	49.000000	222.000000	284.000000
max	NaN	NaN	NaN	NaN	41.00000	45.455000	100.000000	56.996900	367.000000	886.000000	977.000000

```
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
```

```
df["datetime"].min(), df['datetime'].max()
```

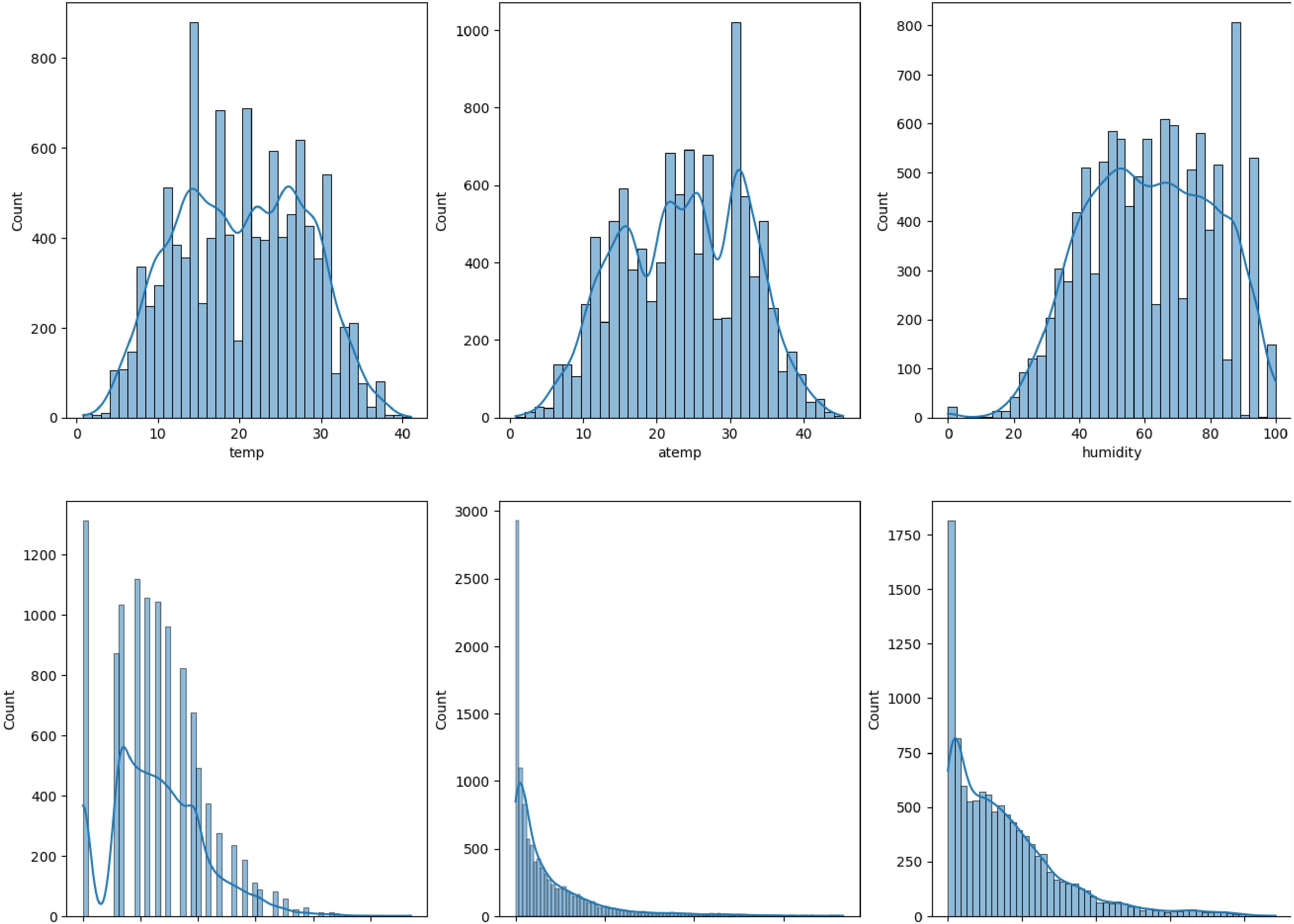
```
(Timestamp('2011-01-01 00:00:00'), Timestamp('2012-12-19 23:00:00'))
```

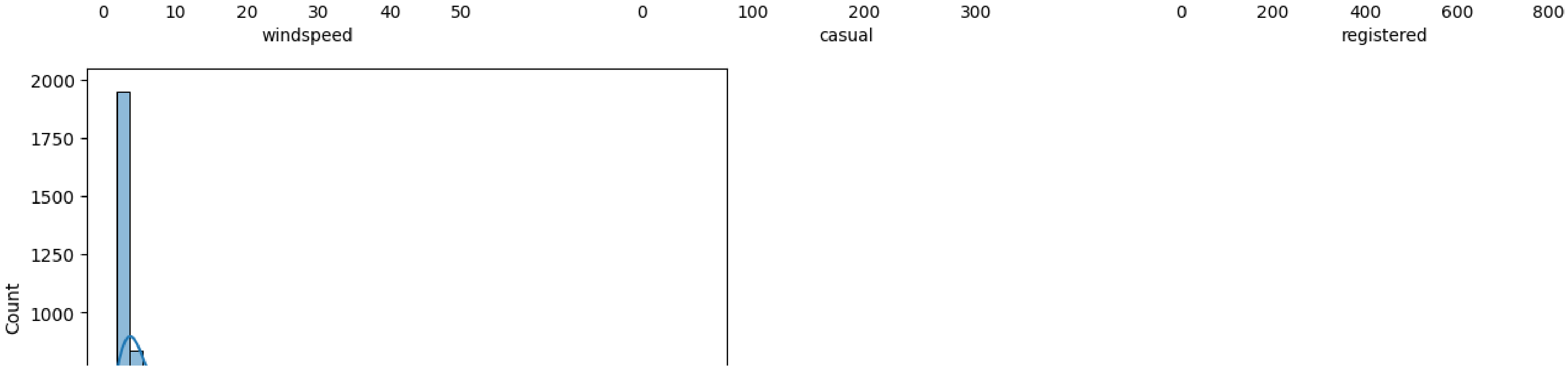
```
df[cat_cols].melt().groupby(['variable', 'value'])[['value']].count()
```

variable	value	value
holiday	0	10575
	1	311
season	1	2686
	2	2733
	3	2733
	4	2734
weather	1	7192
	2	2834
	3	859
	4	1
workingday	0	3474
	1	7412

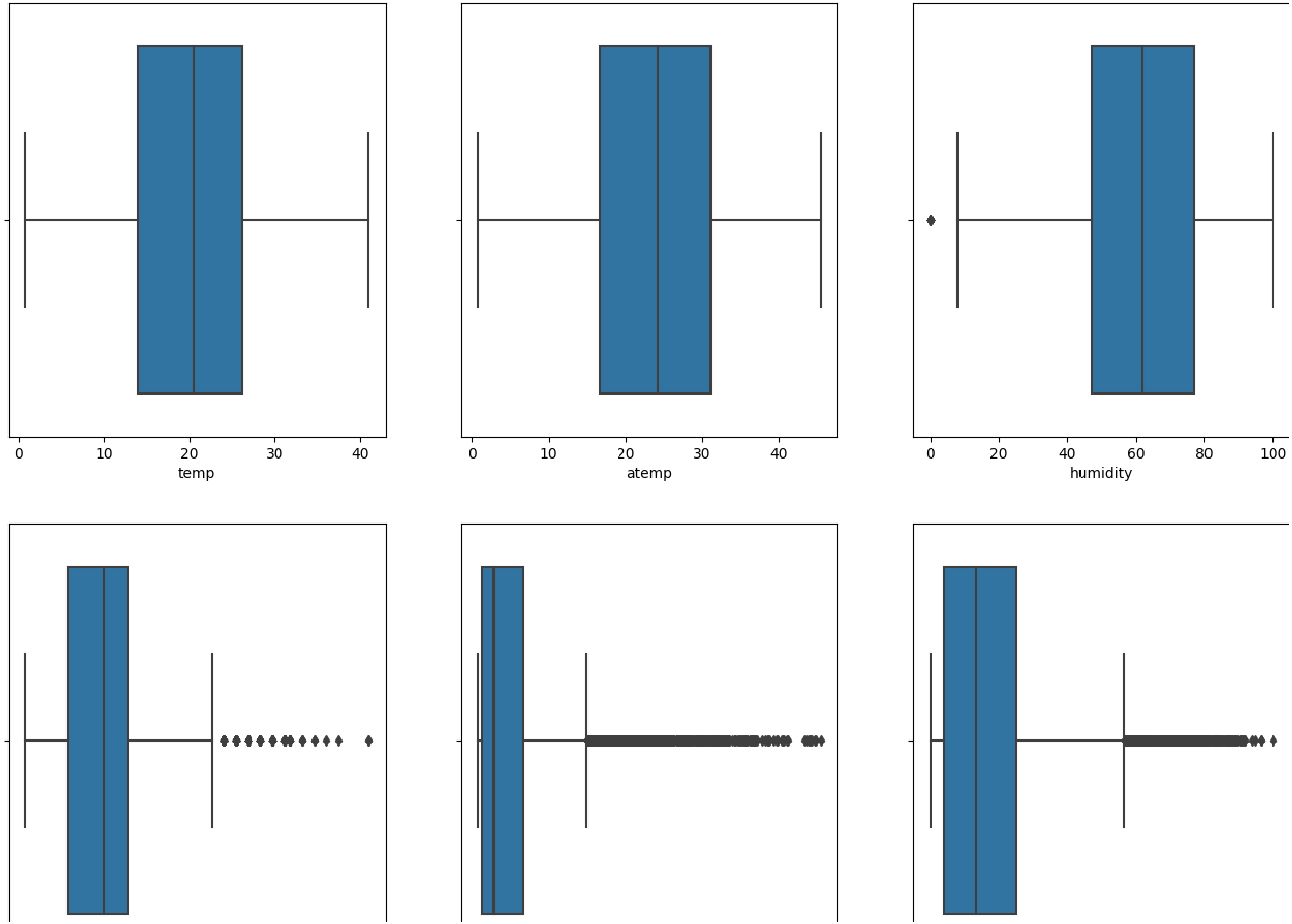
```
# understanding the distribution for numerical variables
num_cols = ['temp', 'atemp', 'humidity', 'windspeed', 'casual', 'registered','count']

fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(16, 12))
```

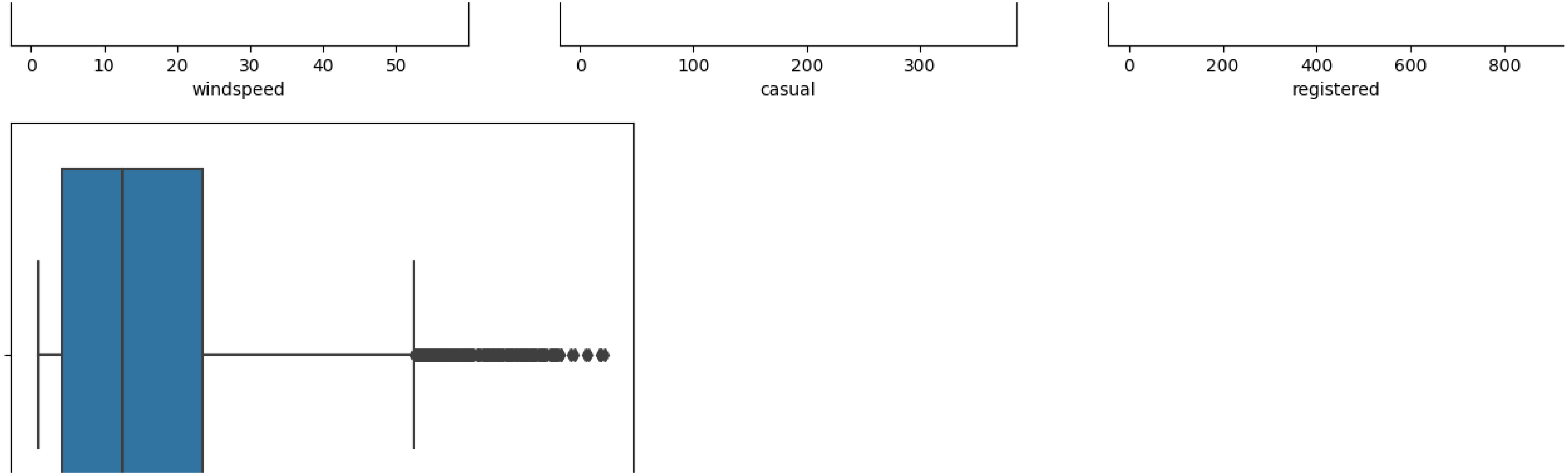
```
index = 0
for row in range(2):
    for col in range(3):
        sns.histplot(df[num_cols[index]], ax=axis[row, col], kde=True)
        index += 1

plt.show()
sns.histplot(df[num_cols[-1]], kde=True)
plt.show()

'''
1. Casual, Registered, and Count resembling Log Normal Distribution: This means that the data for these variables might exhibit a
distribution pattern similar to the log-normal distribution. In a log-normal distribution, the logarithm of the data values is normally
 distributed. This suggests that the data points might be skewed towards higher values, with a longer tail on the right side.

2. Temp, Atemp, and Humidity following Normal Distribution: This implies that the data for temperature, apparent temperature ("atemp"),
and humidity might be distributed in a way that resembles the normal distribution. The normal distribution, also known as the Gaussian
distribution, is characterized by its bell-shaped curve and is commonly observed in many natural phenomena.

3.Windspeed following Binomial Distribution: This suggests that the data for windspeed might exhibit a distribution similar to the
 binomial distribution. The binomial distribution is often associated with the number of successes in a fixed number of independent
 Bernoulli trials, which can translate to events with two possible outcomes

4. Correlation between Temperature and "Temp": This indicates that there is a noticeable relationship between the variables temperature
and "temp." The strong correlation suggests that changes in one variable are closely associated with corresponding changes in the other,
possibly indicating a linear relationship between the two
'''
```


Count
temp
0 10 20 30 40
0 200 400 600 800
Count
atemp
0 10 20 30 40
0 200 400 600 800 1000
Count
humidity
0 20 40 60 80 100
0 100 200 300 400 500 600 700 800
Count
0 200 400 600 800 1000 1200
Count
0 500 1000 1500 2000 2500 3000
Count
0 250 500 750 1000 1250 1500 1750

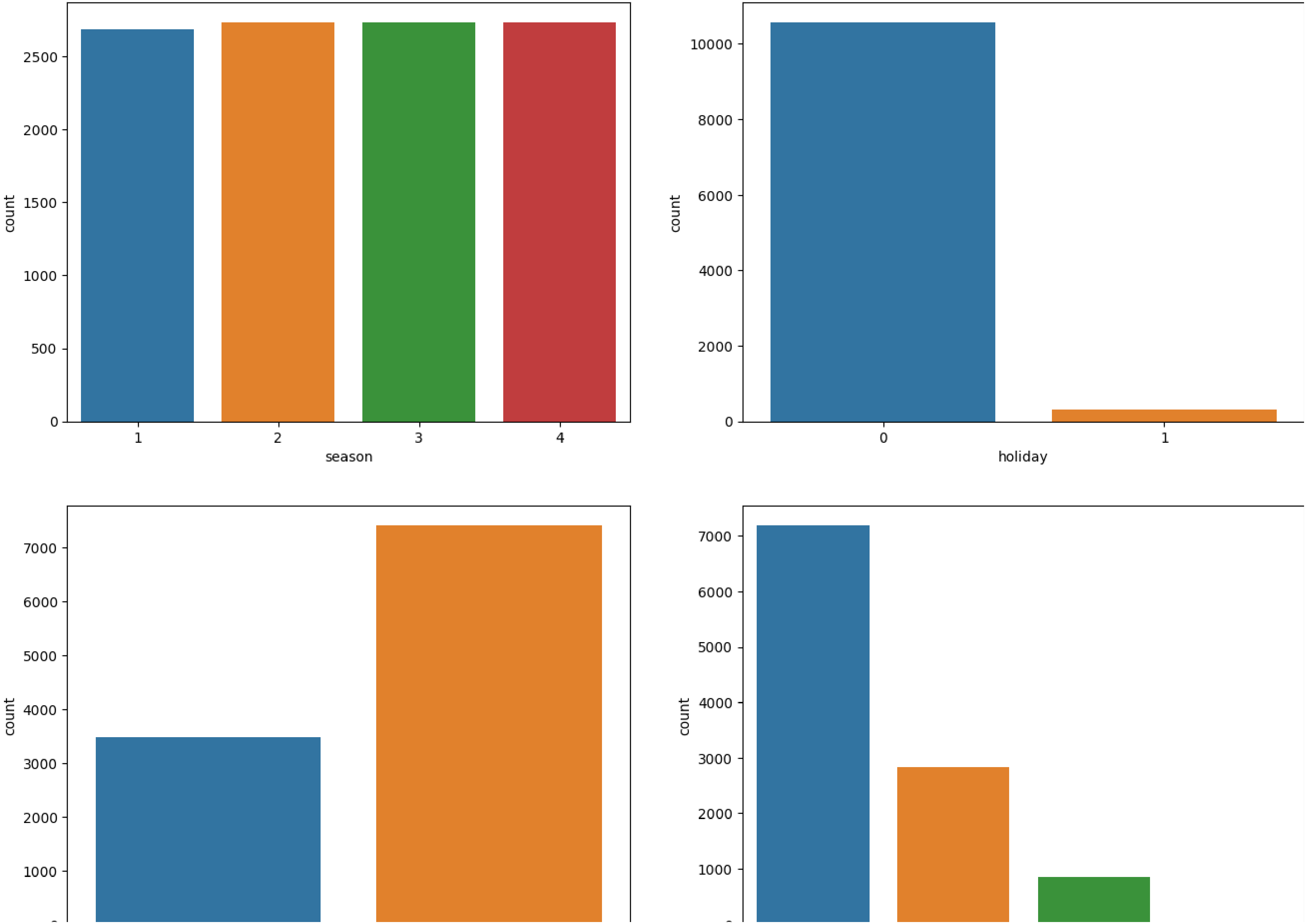



```
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(16, 12))

index = 0
for row in range(2):
    for col in range(3):
        sns.boxplot(x=df[num_cols[index]], ax=axis[row, col])
        index += 1

plt.show()
sns.boxplot(x=df[num_cols[-1]])
plt.show()

'''
The variables humidity, casual, registered, and count appear to have outliers within their data. Outliers are data points that deviate
significantly from the overall pattern of the dataset. In the context of your analysis, this suggests that there are values for humidity,
as well as the variables casual, registered, and count, that are unusually high or low compared to the majority of the data points.
These outliers can potentially impact the statistical analyses and interpretations of these variables, and it's important to consider
their presence when drawing conclusions from the data
'''
```

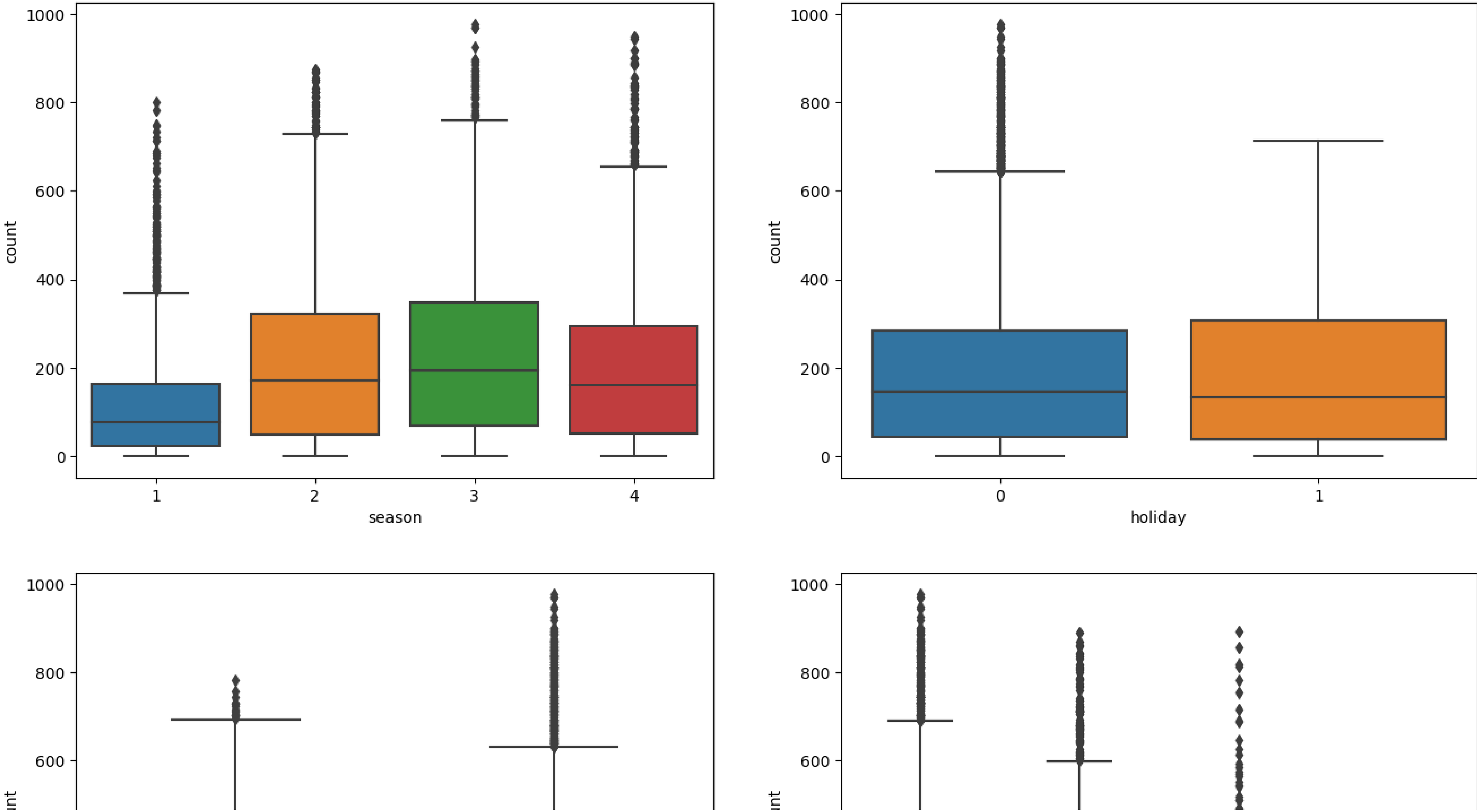

0
10
20
30
40
temp
0
10
20
30
40
atemp
0
20
40
60
80
100
humidity




```
# countplot of each categorical column
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(16, 12))

index = 0
for row in range(2):
    for col in range(2):
        sns.countplot(data=df, x=cat_cols[index], ax=axis[row, col])
        index += 1

plt.show()

'''
The dataset appears to exhibit a balanced distribution across the seasons, with an equal number of days represented in each season.
Additionally, there is a higher frequency of working days in the dataset. The weather conditions predominantly consist of clear skies,
a few clouds, and partly cloudy conditions. This common distribution suggests that the dataset reflects a typical and expected pattern
in terms of seasonal distribution, working days, and prevalent weather conditions.
'''
```

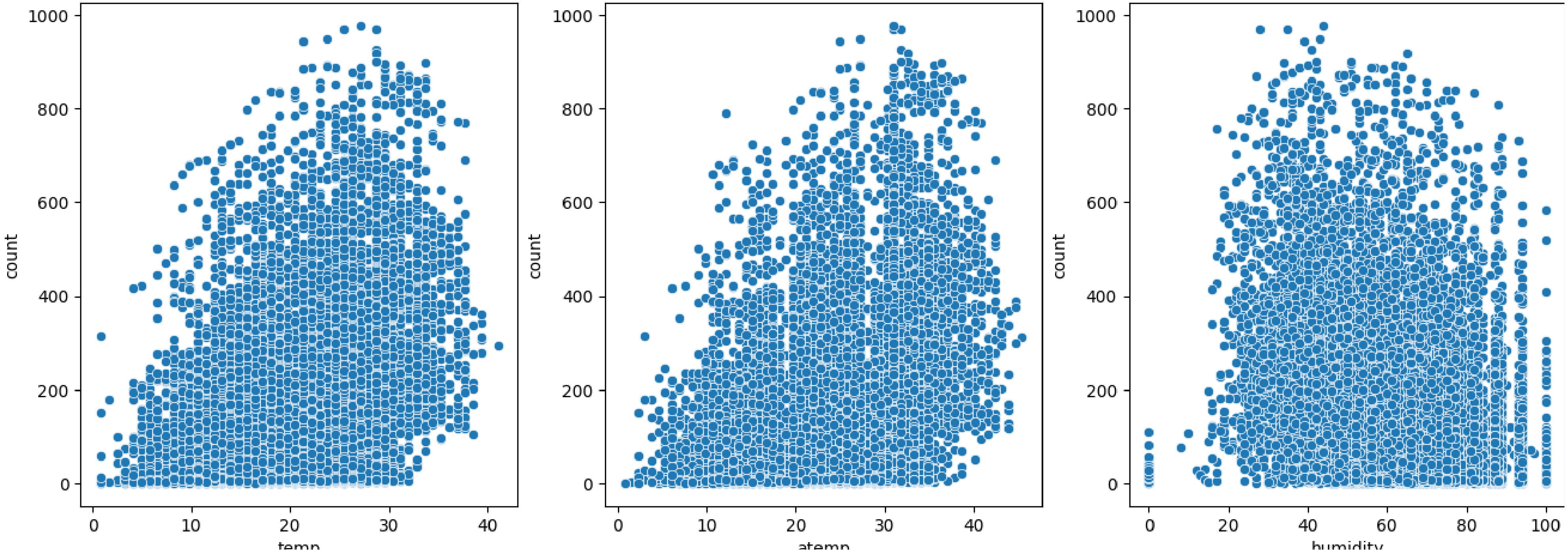

count
season
1
2
3
4
2500
2000
1500
1000
500
0
count
holiday
0
1
10000
8000
6000
4000
2000
0
count
7000
6000
5000
4000
3000
2000
1000
count
7000
6000
5000
4000
3000
2000
1000

```
# plotting categorical variables againt count using boxplots
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(16, 12))

index = 0
for row in range(2):
    for col in range(2):
        sns.boxplot(data=df, x=cat_cols[index], y='count', ax=axis[row, col])
        index += 1

plt.show()

'''
1. Seasonal Bike Rentals: During the summer and fall seasons, there is a higher demand for bike rentals compared to other seasons.
This trend could be attributed to more favorable weather conditions and outdoor activities during these seasons.

2. Holiday Bike Rentals: On holidays, there is an increased number of bike rentals. This could be due to people having more free
time and leisure on holidays, leading to a higher demand for bike rides and outdoor activities

3. Working Day Impact: The data indicates that bike rentals are slightly higher on holidays and weekends, suggesting that people
 are more likely to rent bikes when they have time off from work or their regular routines

4. Weather Conditions and Bike Rentals: Days with adverse weather conditions such as rain, thunderstorms, snow, or fog tend to
have fewer bike rentals. This is likely because such weather conditions can discourage outdoor activities and make biking less appealing or practical
'''
```




```
# plotting numerical variables againt count using scatterplot
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(16, 12))

index = 0
for row in range(2):
    for col in range(3):
        sns.scatterplot(data=df, x=num_cols[index], y='count', ax=axis[row, col])
        index += 1

plt.show()
```

```
'''
1. Low Humidity and Bike Rentals: When the humidity level drops below 20, the number of bikes rented is significantly reduced.
This could be because low humidity is often associated with dry and potentially uncomfortable conditions, which might discourage
people from engaging in outdoor activities like biking

2. Low Temperature Impact: Days with temperatures below 10 degrees Celsius see a decrease in the number of bike rentals.
Colder temperatures might make biking less appealing or practical due to the discomfort and potential challenges posed by the weather

3. High Windspeed and Bike Rentals: Whenever the windspeed exceeds 35 units, the number of bikes rented is lower. High winds
can make biking more challenging and less enjoyable, which likely contributes to the decrease in bike rentals on such days
'''
```

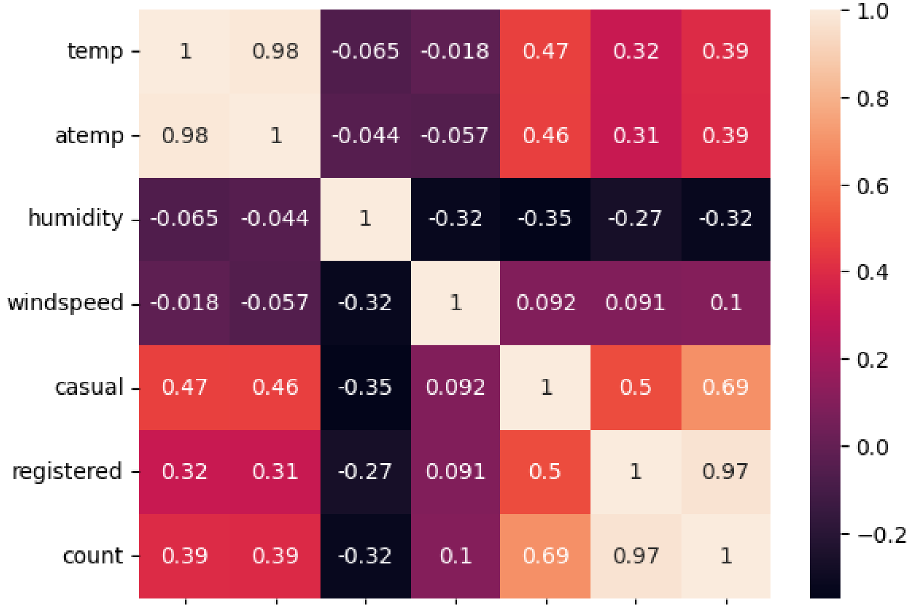



```
# understanding the correlation between count and numerical variables
df.corr()['count']
```

```
<ipython-input-22-85b774de02c3>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will def
  df.corr()['count']
temp          0.394454
atemp         0.389784
humidity     -0.317371
windspeed     0.101369
casual        0.690414
registered    0.970948
count         1.000000
Name: count, dtype: float64
```

```
sns.heatmap(df.corr(), annot=True)
plt.show()
```

```
<ipython-input-23-6522c2b4e5f9>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will def
  sns.heatmap(df.corr(), annot=True)
```




```
'''
Hypothesis Testing - 1
Null Hypothesis (H0): Weather is independent of the season

Alternate Hypothesis (H1): Weather is not independent of the season

Significance level (alpha): 0.05

We will use chi-square test to test hypyothesis defined above
'''


data_table = pd.crosstab(df['season'], df['weather'])
print("Observed values:")
data_table
```

```
Observed values:
 weather    1    2    3  4
```

```
val = stats.chi2_contingency(data_table)
expected_values = val[3]
expected_values
```

```
array([[1.77454639e+03, 6.99258130e+02, 2.11948742e+02, 2.46738931e-01],
       [1.80559765e+03, 7.11493845e+02, 2.15657450e+02, 2.51056403e-01],
       [1.80559765e+03, 7.11493845e+02, 2.15657450e+02, 2.51056403e-01],
       [1.80625831e+03, 7.11754180e+02, 2.15736359e+02, 2.51148264e-01]])
```

```
nrows, ncols = 4, 4
dof = (nrows-1)*(ncols-1)
print("degrees of freedom: ", dof)
alpha = 0.05

chi_sqr = sum([(o-e)**2/e for o, e in zip(data_table.values, expected_values)])
chi_sqr_statistic = chi_sqr[0] + chi_sqr[1]
print("chi-square test statistic: ", chi_sqr_statistic)

critical_val = stats.chi2.ppf(q=1-alpha, df=dof)
print(f"critical value: {critical_val}")

p_val = 1-stats.chi2.cdf(x=chi_sqr_statistic, df=dof)
print(f"p-value: {p_val}")

if p_val <= alpha:
    print("\nSince p-value is less than the alpha 0.05, We reject the Null Hypothesis. Meaning that\
    Weather is dependent on the season.")
else:
    print("Since p-value is greater than the alpha 0.05, We do not reject the Null Hypothesis")
```

```
degrees of freedom:  9
chi-square test statistic:  44.09441248632364
critical value: 16.918977604620448
p-value: 1.3560001579371317e-06

Since p-value is less than the alpha 0.05, We reject the Null Hypothesis. Meaning that    Weather is dependent on the season.
```

```
'''
Hypothesis Testing - 2
Null Hypothesis: Working day has no effect on the number of cycles being rented.

Alternate Hypothesis: Working day has effect on the number of cycles being rented.
```

```
Significance level (alpha): 0.05

We will use the 2-Sample T-Test to test the hypothess defined above
'''


data_group1 = df[df['workingday']==0]['count'].values
data_group2 = df[df['workingday']==1]['count'].values

np.var(data_group1), np.var(data_group2)
```

```
(30171.346098942427, 34040.69710674686)
```

```
'''
Checking the homogeneity of variances between the two data groups is crucial before proceeding with a two-sample T-Test,
and if the ratio of the larger group's size to the smaller group's size is less than 4:1, it indicates a reasonable assumption of equal variances

Here, the ratio is 34040.70 / 30171.35 which is less than 4:1
'''



stats.ttest_ind(a=data_group1, b=data_group2, equal_var=True)
```

```
Ttest_indResult(statistic=-1.2096277376026694, pvalue=0.22644804226361348)
```

```
'''
Since pvalue is greater than 0.05 so we can not reject the Null hypothesis. We don't have the sufficient evidence to say that
working day has effect on the number of cycles being rented
'''


'''
Hypothesis Testing - 3
Null Hypothesis: Number of cycles rented is similar in different weather and season.

Alternate Hypothesis: Number of cycles rented is not similar in different weather and season.

Significance level (alpha): 0.05

Here, we will use the ANOVA to test the hypothess defined above
'''


# defining the data groups for the ANOVA

gp1 = df[df['weather']==1]['count'].values
```

```
gp2 = df[df['weather']==2]['count'].values
gp3 = df[df['weather']==3]['count'].values
gp4 = df[df['weather']==4]['count'].values

gp5 = df[df['season']==1]['count'].values
gp6 = df[df['season']==2]['count'].values
gp7 = df[df['season']==3]['count'].values
gp8 = df[df['season']==4]['count'].values

# conduct the one-way anova
stats.f_oneway(gp1, gp2, gp3, gp4, gp5, gp6, gp7, gp8)
```

```
F_onewayResult(statistic=127.96661249562491, pvalue=2.8074771742434642e-185)
```

```
'''
Since p-value is less than 0.05, we reject the null hypothesis. This implies that Number of cycles rented is not similar in
different weather and season conditions
'''
```

```
##Insights
```

```
'''
1. Seasonal Trend: Bike rentals are higher during the summer and fall seasons compared to other seasons, possibly due
to more favorable weather conditions for outdoor activities.

2. Holiday Impact: Increased bike rentals are observed on holidays, indicating that leisure time on holidays leads to more demand for bike rides

3.Working Days and Weekends: Bike rentals are slightly elevated on holidays and weekends, suggesting people are more inclined
to rent bikes when they have time off

4. Weather Conditions: Fewer bikes are rented on days with adverse weather like rain, thunderstorms, snow, or fog, likely
due to decreased outdoor activity.

5. Low Humidity: Days with humidity levels below 20 see a significantly low number of bike rentals, possibly because dry
conditions discourage outdoor activities

6. Low Temperature: Bike rentals decrease when temperatures are below 10 degrees Celsius, likely due to discomfort and weather challenges

7. High Windspeed: Elevated windspeeds above 35 result in fewer bike rentals, possibly due to safety concerns and less enjoyable biking conditions
'''
```

```
#Recommendations based on insights
```

```
'''
1. Seasonal Demand: To meet higher demand during the summer and fall, the company should ensure an increased stock
of bikes available for rent during these seasons
```