```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
df = pd.read_csv('yulu.csv')
```

```
# Display first five rows about the dataset
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

```
# Display basic information about the dataset
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
```

```
# Check for missing values
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
```

There are no missing values present in the dataset.

```
# 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')

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

```
df.describe(include='all')
```

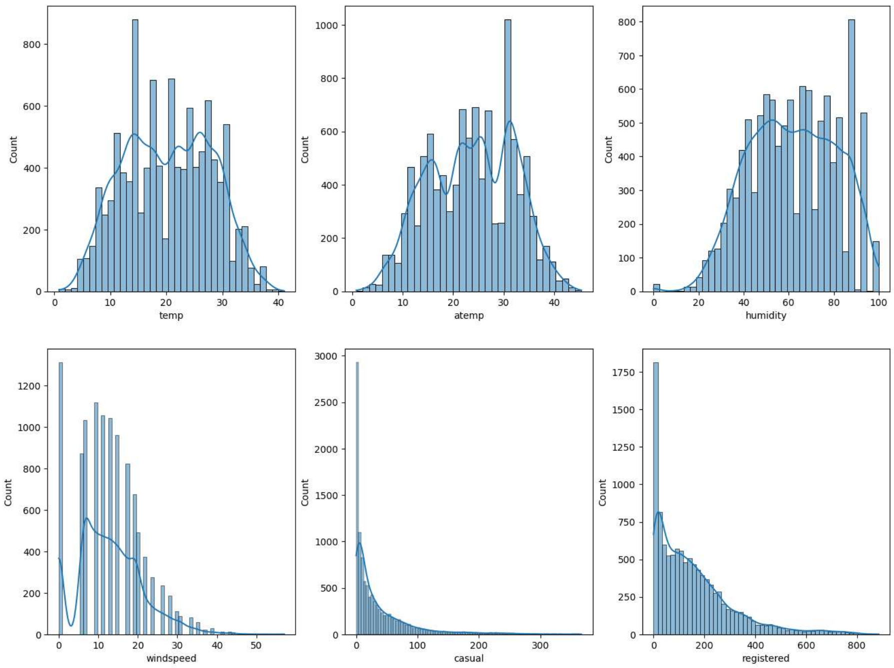
```
<ipython-input-6-174ba9bf1a5c>:1: FutureWarning: Treating datetime data as categorical r
  df.describe(include='all')
```

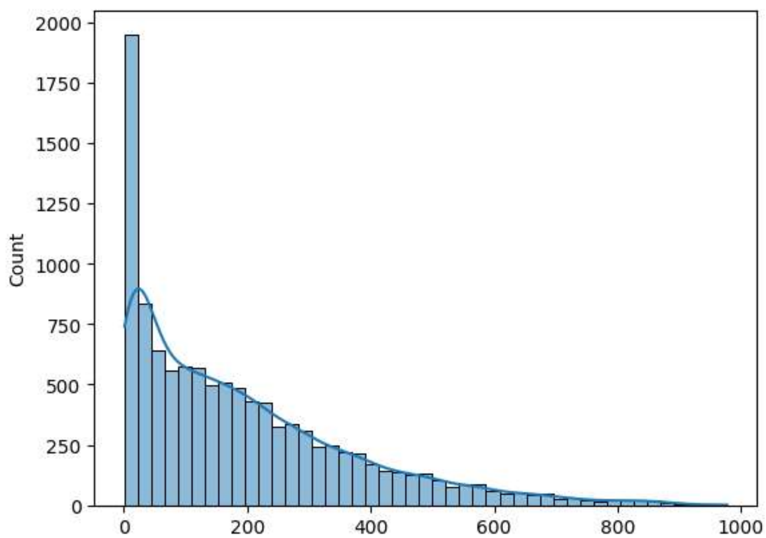
	datetime	season	holiday	workingday	weather	temp	atemp	hu
count	10886	10886.0	10886.0	10886.0	10886.0	10886.00000	10886.000000	10886
unique	10886	4.0	2.0	2.0	4.0	NaN	NaN	
top	2011-01-01 00:00:00	4.0	0.0	1.0	1.0	NaN	NaN	
freq	1	2734.0	10575.0	7412.0	7192.0	NaN	NaN	
first	2011-01-01 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN	
last	2012-12-19 23:00:00	NaN	NaN	NaN	NaN	NaN	NaN	
mean	NaN	NaN	NaN	NaN	NaN	20.23086	23.655084	61
std	NaN	NaN	NaN	NaN	NaN	7.79159	8.474601	19
min	NaN	NaN	NaN	NaN	NaN	0.82000	0.760000	0
25%	NaN	NaN	NaN	NaN	NaN	13.94000	16.665000	47

```
# minimum datetime and maximum datetime
df['datetime'].min(), df['datetime'].max()
```

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

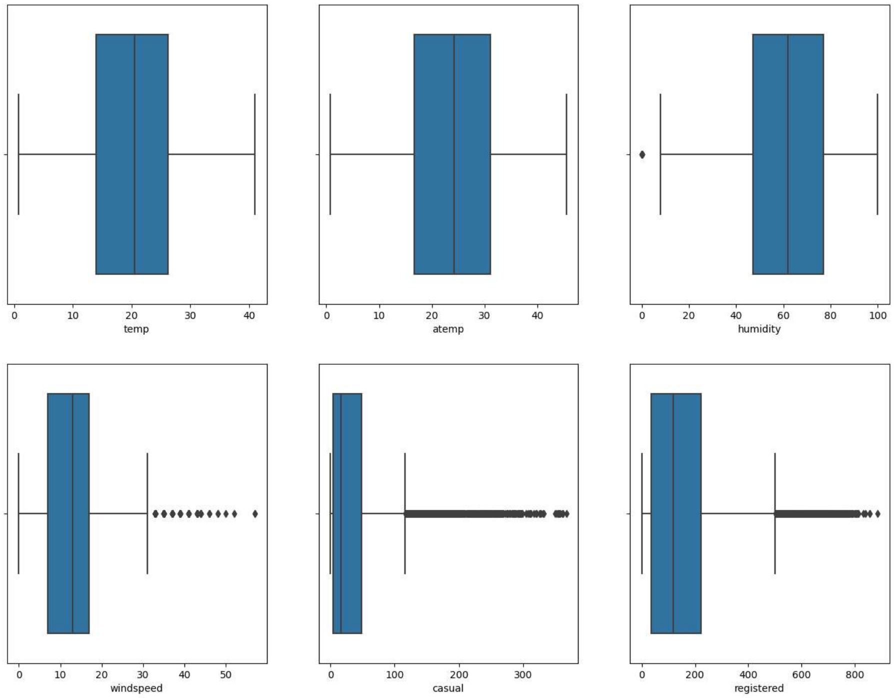
```
# Univariate Analysis

# 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()
```

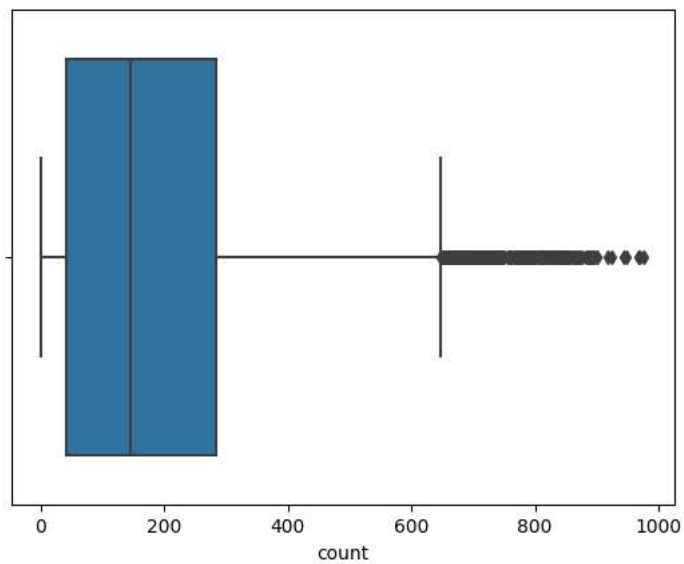




1. The distribution of casual, registered, and total bike counts appears to resemble a Log-Normal Distribution.
2. The temperature (temp), apparent temperature (atemp), and humidity exhibit characteristics indicative of a Normal Distribution.
3. Windspeed appears to adhere to a Binomial Distribution.

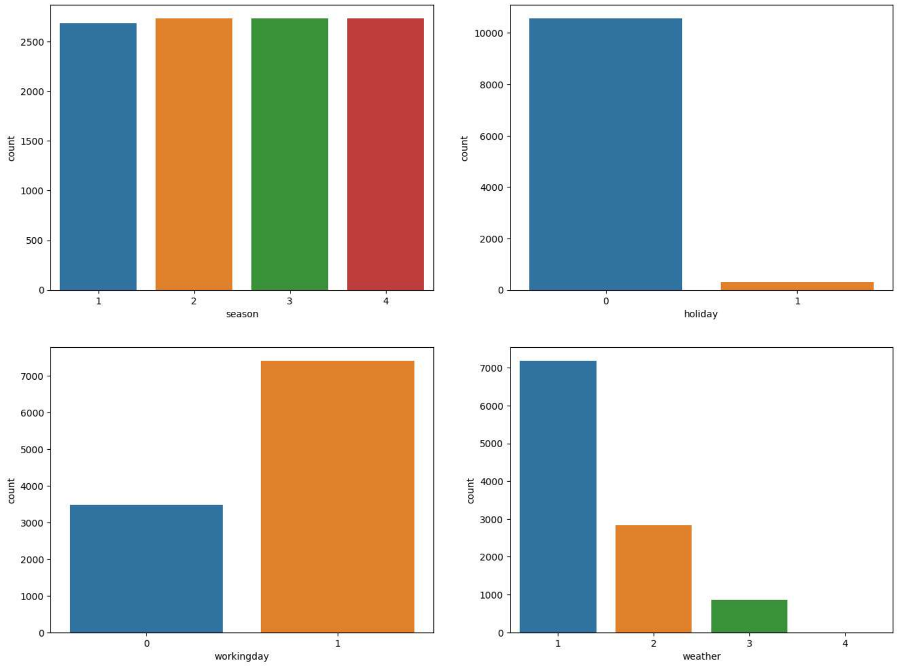
```
# plotting box plots to detect outliers in the data
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(16, 12))

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()
```





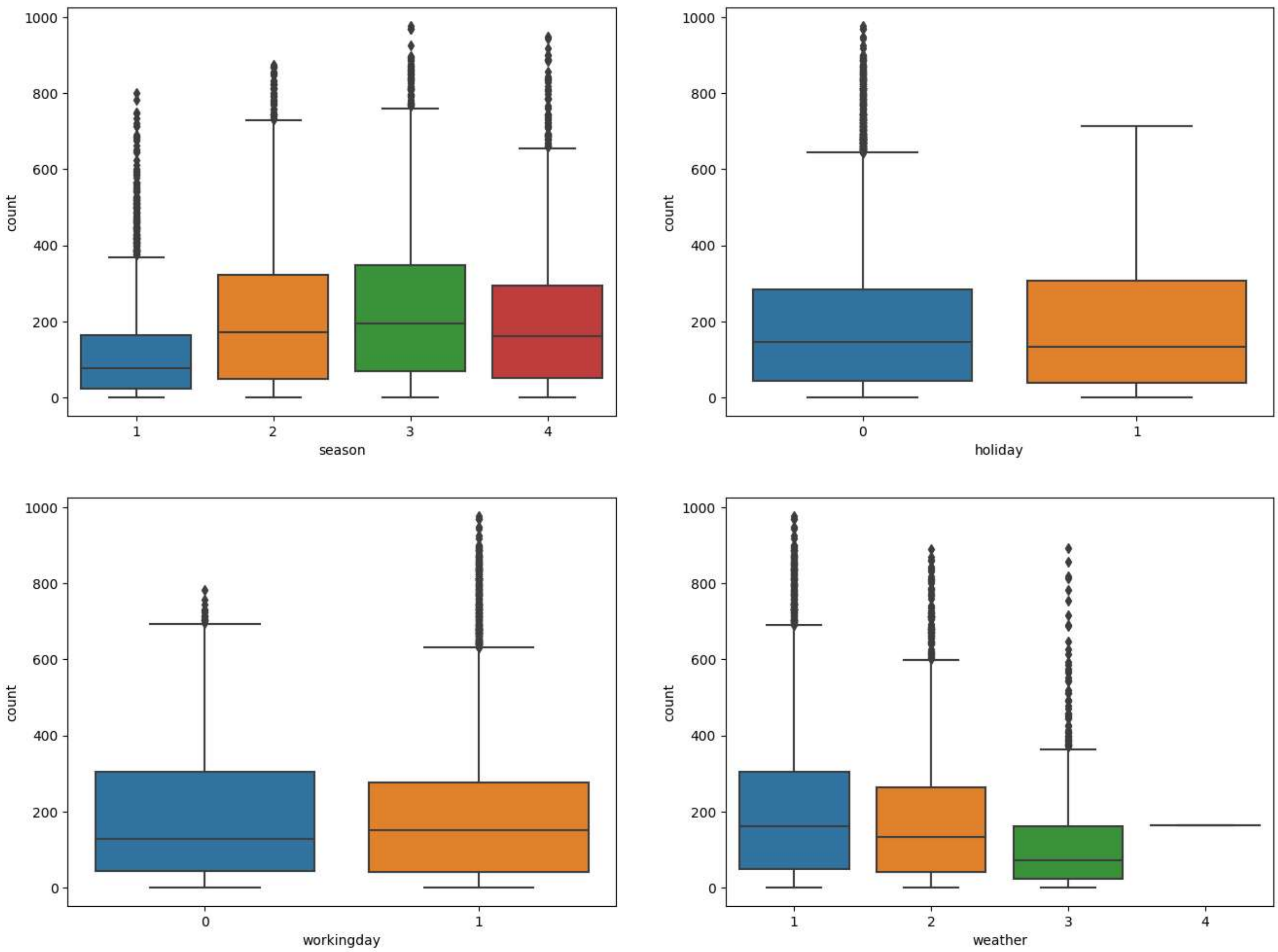
Looks like humidity, casual, registered and count have outliers in the data.

```
# countplot of each categorical column
cat_cols= ['season', 'holiday', 'workingday', 'weather']
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()
```



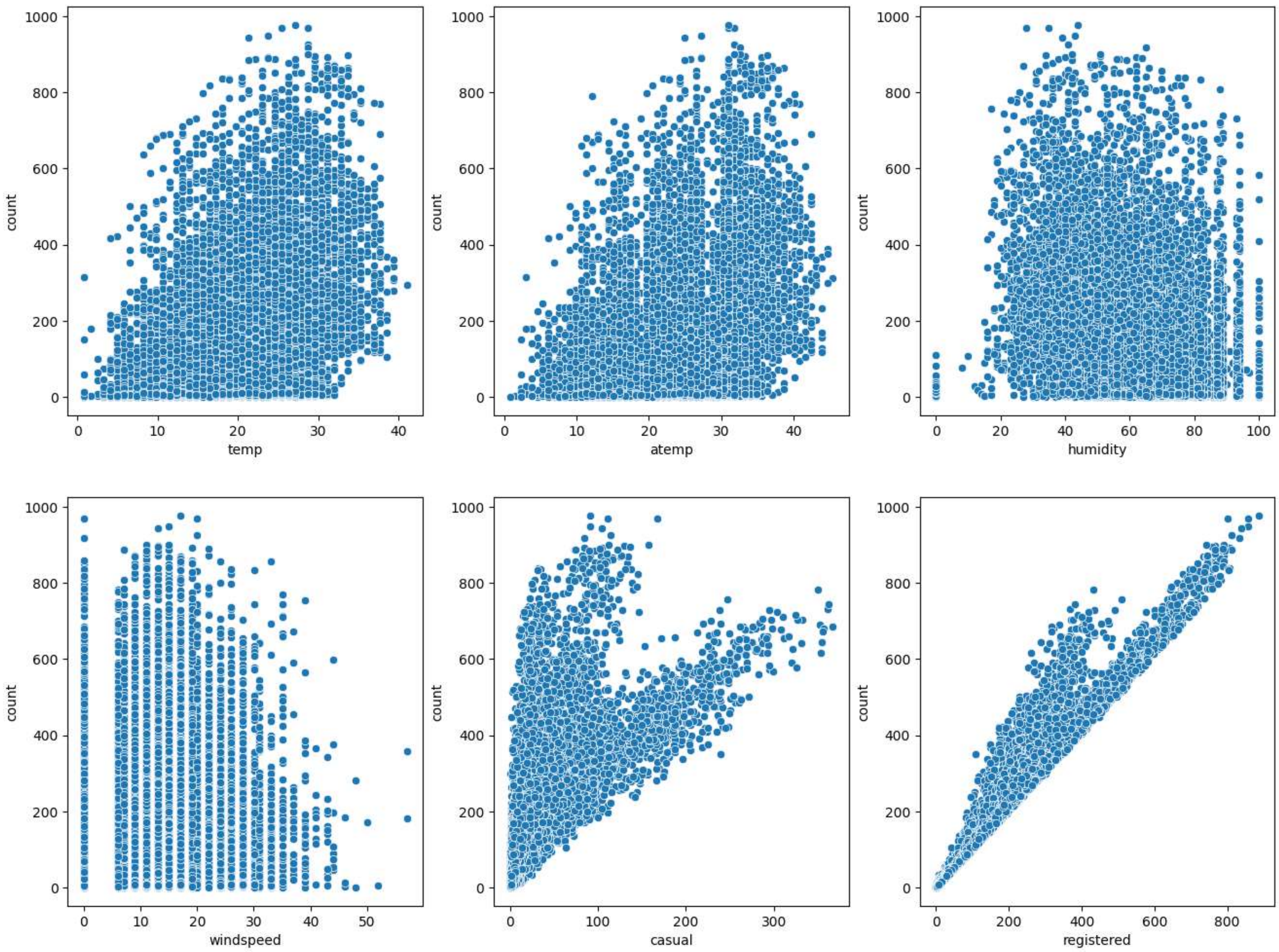
The dataset appears to be well-balanced, with an equal distribution of days across each season. Working days are more prevalent, and the predominant weather conditions include clear skies, a few clouds, and partly cloudy conditions.

```
# Bivariate Analysis

# 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. Higher bike rentals are observed during the summer and fall seasons compared to other seasons.
2. Increased bike rentals are noted on holidays.
3. Analysis of working days also indicates that slightly more bikes are rented on holidays or weekends.
4. Reduced bike rentals are observed during rainy, thunderstorm, snowy, or foggy conditions.

```
# 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. Whenever the humidity is less than 20, number of bikes rented is very low.
2. Whenever the temperature is less than 10, number of bikes rented is less.
3. Whenever the windspeed is greater than 35, number of bikes rented is less.

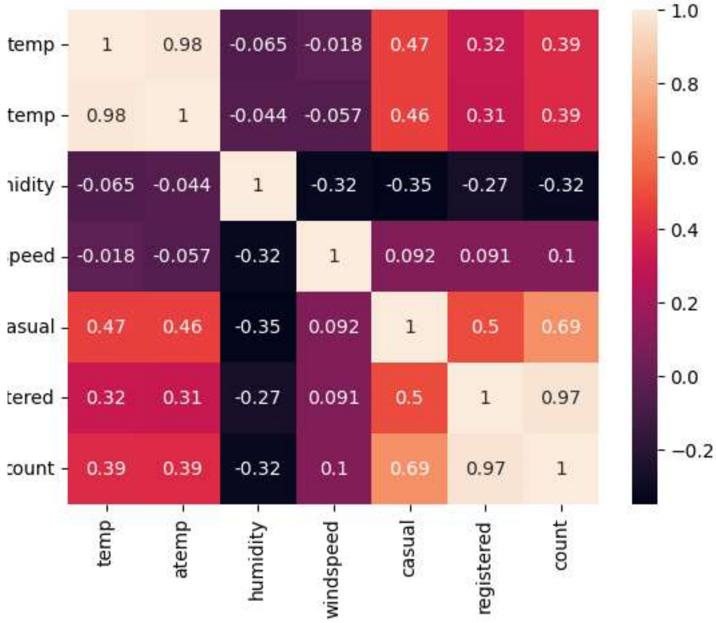
```
# understanding the correlation between count and numerical variables
df.corr()['count']
```

```
    3-d24438c865eb>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default
    nt']
    394454
    389784
```

```
317371
101369
690414
970948
000000
pe: float64
```

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

```
ɔn-input-14-6522c2b4e5f9>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it wi
neatmap(df.corr(), annot=True)
```



Hypothesis Testing:

1. 2-Sample T-Test:

Checking if working day has an effect on the number of electric cycles rented.

```
from scipy.stats import ttest_ind

# Define Null Hypothesis (H0) and Alternative Hypothesis (H1)
# H0: There is no significant difference in the number of cycles rented between working and non-working days.
# H1: There is a significant difference in the number of cycles rented between working and non-working days.

# Perform the t-test
t_stat, p_value = ttest_ind(df[df['workingday'] == 1]['count'],
                             df[df['workingday'] == 0]['count'])

# Set significance level (alpha)
alpha = 0.05

# Decision
if p_value < alpha:
    print("Reject H0: There is a significant difference.")
else:
    print("Fail to reject H0: There is no significant difference.")
```

```
Fail to reject H0: There is no significant difference.
```

2. ANOVA:

Checking if the number of cycles rented is similar or different in different weather and season conditions.

```
from scipy.stats import f_oneway

# H0: There is no significant difference in the number of cycles rented across different weather conditions.
# H1: There is a significant difference in the number of cycles rented across different weather conditions.

# Perform ANOVA for weather
weather_groups = [df[df['weather'] == i]['count'] for i in df['weather'].unique()]
f_stat_weather, p_value_weather = f_oneway(*weather_groups)

# Repeat for season
season_groups = [df[df['season'] == i]['count'] for i in df['season'].unique()]
f_stat_season, p_value_season = f_oneway(*season_groups)

# Decision
if p_value_weather < alpha:
    print("Reject H0: There is a significant difference in the number of cycles rented across different weather conditions.")
else:
    print("Fail to reject H0: There is no significant difference in the number of cycles rented across different weather conditions.")

# Repeat for season
# Decision
if p_value_season < alpha:
    print("Reject H0: There is a significant difference in the number of cycles rented across different seasons.")
else:
    print("Fail to reject H0: There is no significant difference in the number of cycles rented across different seasons.")
```

```
Reject H0: There is a significant difference in the number of cycles rented across different weather conditions.
Reject H0: There is a significant difference in the number of cycles rented across different seasons.
```

3. Chi-square Test:

Checking if weather is dependent on the season.

```
from scipy.stats import chi2_contingency

# H0: Weather and season are independent.
# H1: Weather and season are dependent.

# Create a contingency table
contingency_table = pd.crosstab(df['weather'], df['season'])

# Perform chi-square test
chi2_stat, p_value_chi2, dof, expected = chi2_contingency(contingency_table)

# Decision
if p_value_chi2 < alpha:
    print("Reject H0: Weather and season are dependent.")
else:
    print("Fail to reject H0: Weather and season are independent.")
```

```
Reject H0: Weather and season are dependent.
```

Insights

1. Increased bike rentals are observed during the summer and fall seasons compared to other times of the year.
2. Higher bike rental rates are evident on holidays.
3. Analysis of workingday data indicates that more bikes are rented on holidays or weekends.
4. Reduced bike rentals are observed during rainy, thunderstorm, snowy, or foggy conditions.
5. When humidity levels drop below 20, there is a notable decrease in the number of bikes rented.
6. Bike rentals tend to be lower when the temperature falls below 10.
7. Elevated windspeeds exceeding 35 are associated with a decline in the number of bikes rented.

Recommendations

1. During the summer and fall seasons, it is advisable for the company to increase its bike inventory to meet the higher demand experienced in these periods compared to other seasons.

2. At a significance level of 0.05, there is no observed effect of workingday on the number of bikes rented.
3. On days with very low humidity, it is recommended for the company to reduce its bike inventory available for rental.
4. In instances where the temperature falls below 10 degrees Celsius or during very cold days, the company should consider reducing the number of bikes available for rental.
5. When the windspeed exceeds 35 or during thunderstorms, it is advisable for the company to decrease its bike inventory to be rented.