08puv2xtg

March 7, 2023

[97]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[128]:
```python
from scipy.stats import chi2_contingency
from scipy.stats import ttest_1samp, ttest_ind
from scipy.stats import f_oneway
from statsmodels.graphics.gofplots import qqplot
from scipy.stats import shapiro, kstest
from scipy.stats import levene
from scipy.stats import norm
```

0.0.1 Problem Statement

Company want's to understand the factors on which the demand for shared electric cycles depends. Specifically, they want to understand the factors affecting the demand for these shared electric cycles in the Indian market.

[99]:
```python
df=pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/
↪001/428/original/bike_sharing.csv?1642089089")
```

[100]:
```python
df.head()
```

[100]:
```
              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
```

[68]:
```python
df.shape
```

[68]:
```
(10886, 12)
```

[69]:
```python
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
```

[70]:
```python
# no missing value found in the dataset
df.isnull().sum()
```

[70]:
```
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
```

[71]:
```python
#Datatype of following attributes needs to changed to proper data type
#datetime - to datetime
#season - to categorical
#holiday - to categorical
```

```python
#workingday - to categorical
#weather - to categorical
df['datetime'] = pd.to_datetime(df['datetime'])

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

[72]:
```python
# casual and registered might have outliers because their mean vs median has␣
↪high difference
df.describe()
```

[72]:
```
               temp         atemp      humidity     windspeed        casual  \
count  10886.00000  10886.000000  10886.000000  10886.000000  10886.000000
mean      20.23086     23.655084     61.886460     12.799395     36.021955
std        7.79159      8.474601     19.245033      8.164537     49.960477
min        0.82000      0.760000      0.000000      0.000000      0.000000
25%       13.94000     16.665000     47.000000      7.001500      4.000000
50%       20.50000     24.240000     62.000000     12.998000     17.000000
75%       26.24000     31.060000     77.000000     16.997900     49.000000
max       41.00000     45.455000    100.000000     56.996900    367.000000

         registered         count
count  10886.000000  10886.000000
mean     155.552177    191.574132
std      151.039033    181.144454
min        0.000000      1.000000
25%       36.000000     42.000000
50%      118.000000    145.000000
75%      222.000000    284.000000
max      886.000000    977.000000
```

[73]:
```python
df.columns
```

[73]:
```
Index(['datetime', 'season', 'holiday', 'workingday', 'weather', 'temp',
       'atemp', 'humidity', 'windspeed', 'casual', 'registered', 'count'],
      dtype='object')
```

[74]:
```python
#four types of season all fours have almost same probability of 25%
df['season'].value_counts(normalize=True)
```

[74]:
```
4    0.251148
2    0.251056
3    0.251056
1    0.246739
Name: season, dtype: float64
```

[75]:
```
df['holiday'].value_counts(normalize=True)
```

[75]:
```
0    0.971431
1    0.028569
Name: holiday, dtype: float64
```

[76]:
```
df['workingday'].value_counts(normalize=True)
```

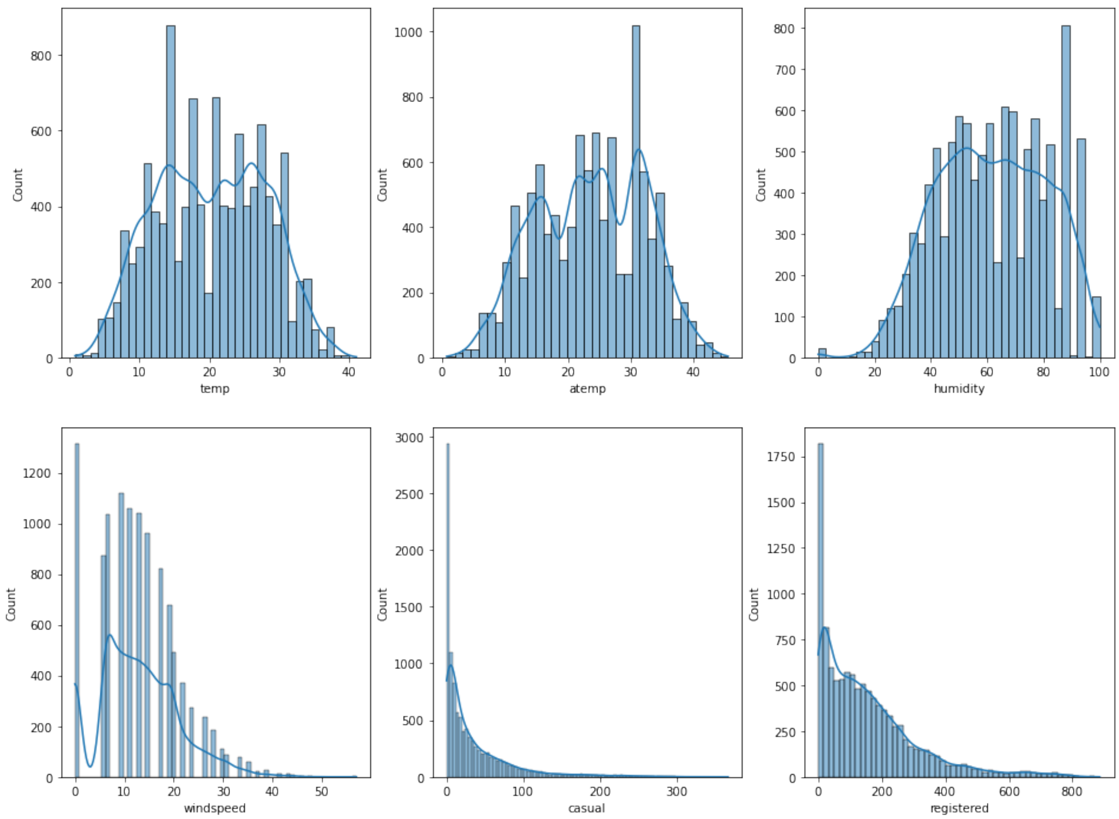
[76]:
```
1    0.680875
0    0.319125
Name: workingday, dtype: float64
```

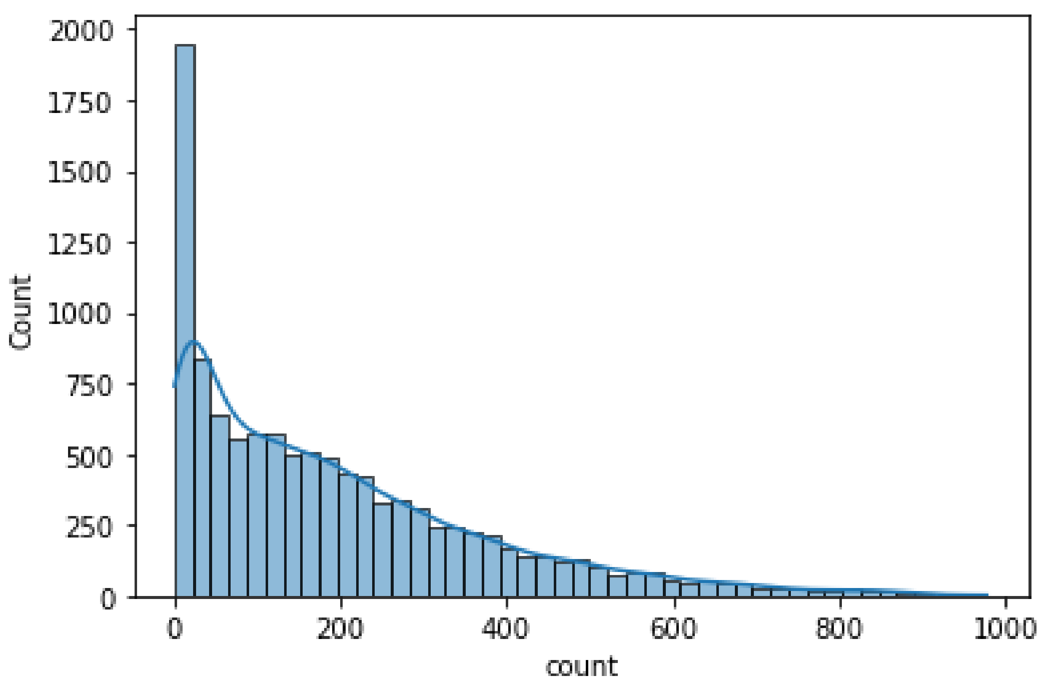
[77]:
```
# understanding the distribution for numerical variables
num_cols = ['temp', 'atemp', 'humidity', 'windspeed', 'casual',␣
 ↪'registered','count']

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()
```


Count
temp
atemp
humidity
windspeed
casual
registered

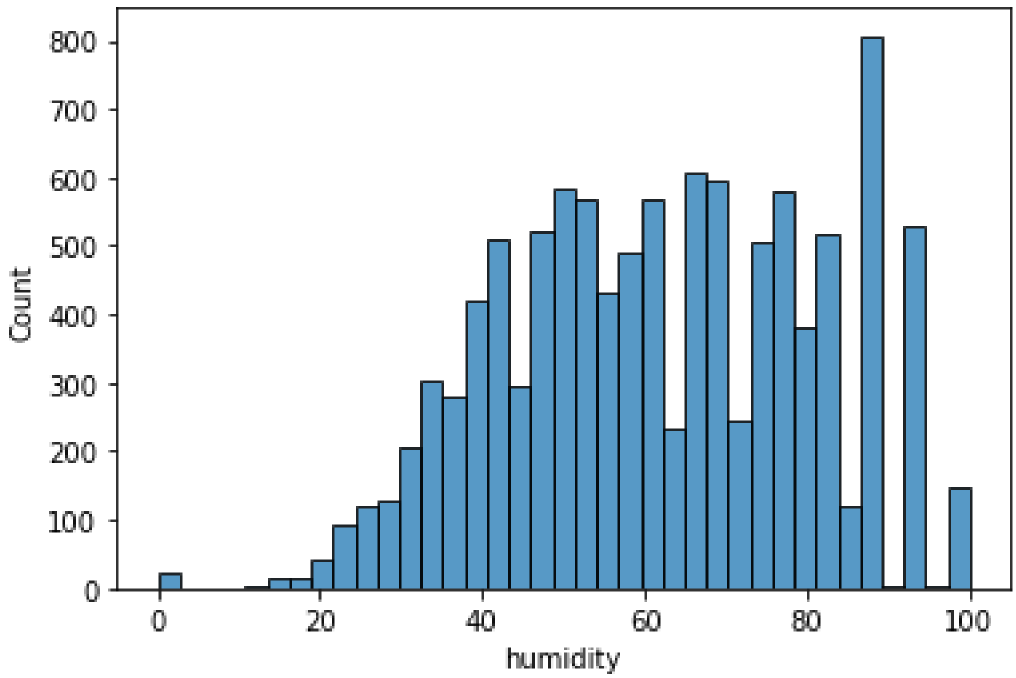

Count
count

TESTS for normality

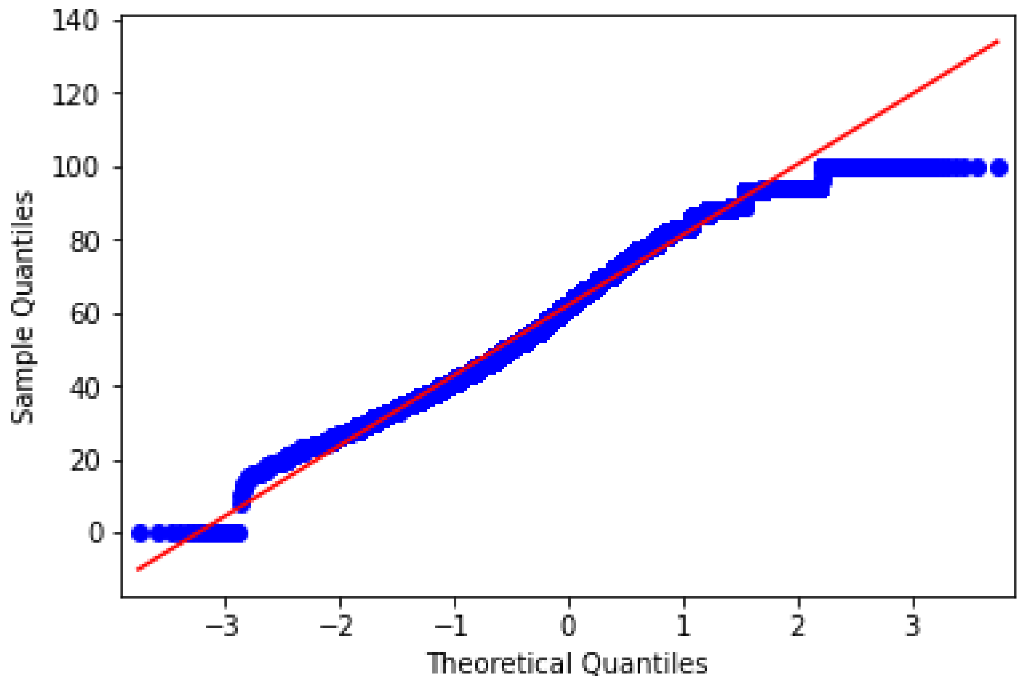
[116]:
```
a1 = df["humidity"]
```

[118]:
```
sns.histplot(a1)
```

[118]: <AxesSubplot:xlabel='humidity', ylabel='Count'>



[120]:
```
# QQ plot
qqplot(a1, line="s")
plt.show()
```



[122]:
```
a1.shape
```

[122]: (10886,)

[125]:
```
# Shapiro and Kolmogrov-Smirnoff test (KSTest) work best in 50 to 200
# They break down for large sample size
a1_subset = a1.sample(100)
```

[126]:
```
# Shapiro
# H0: Data is Gaussian
# Ha: Data is not Gaussian
test_stat, p_value = shapiro(a1_subset)
if p_value < 0.05:
    print("Reject H0")
    print("Data is not Gaussian")
else:
    print("Fail to reject H0")
    print("Data is Gaussian")
```

```
Fail to reject H0
Data is Gaussian
```

[129]:
```
# KSTest
# H0: Data is Gaussian
# Ha: Data is not Gaussian
```

```
test_stat, p_value = kstest(
    a1_subset,
    norm.cdf,
    args=(a1_subset.mean(), a1_subset.std())
)
if p_value < 0.05:
    print("Reject H0")
    print("Data is not Gaussian")
else:
    print("Fail to reject H0")
    print("Data is Gaussian")
```

```
Fail to reject H0
Data is Gaussian
```

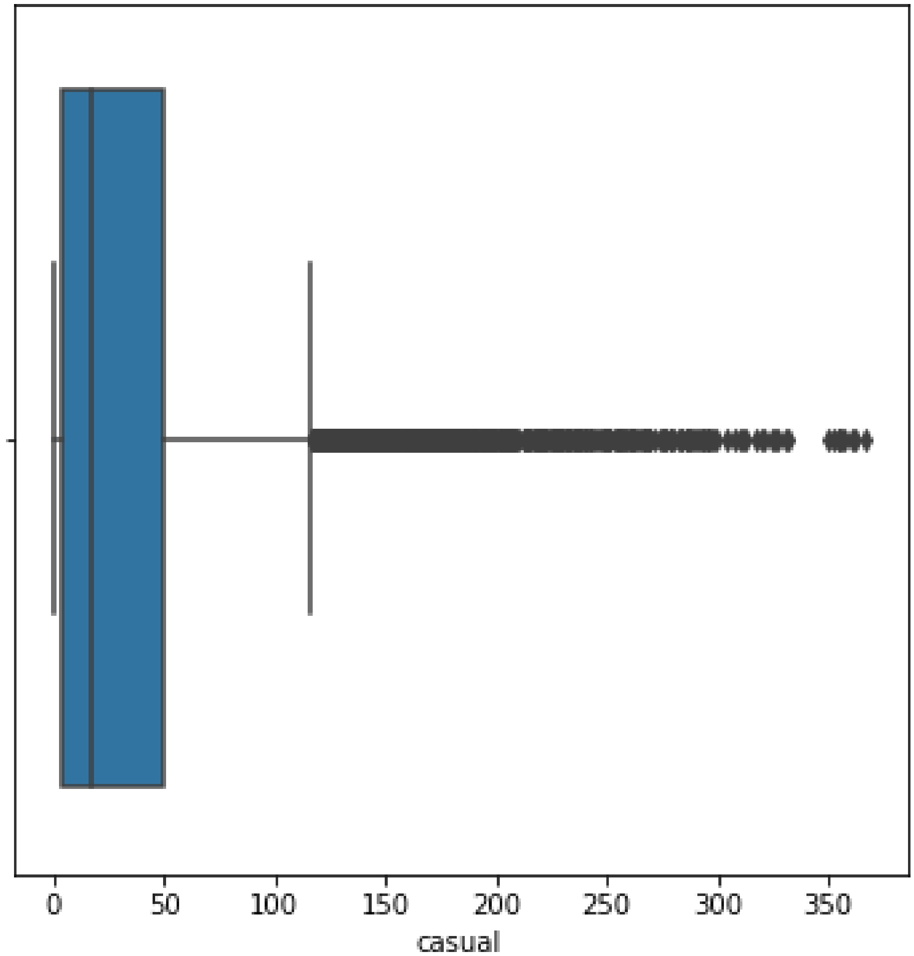
1) casual, registered and count somewhat looks like Log Normal Distrinution.

2) temp, atemp and humidity looks like they follows the Normal Distribution

3)windspeed follows the binomial distribution

0.0.2 Outliers

[78]:
```
#outlier in casual columns
plt.figure(figsize=(6,6))
sns.boxplot(data=df,
            x='casual')
plt.show()
```



[79]:
```
p_25 = df["casual"].quantile(0.25) # Q1 or p_25
p_50 = df["casual"].quantile(0.5)  # Q2 or p_50 or median
p_75 = df["casual"].quantile(0.75) # Q3 or p_75
print(p_25, p_50, p_75)
```

```
4.0 17.0 49.0
```

Boxplot state that meadian values lies near 17

[80]:
```
iqr = p_75 - p_25
lower = max(p_25 - 1.5*iqr, 0)
upper = p_75 + 1.5*iqr
print(lower, upper)
print(iqr)
```

```
0 116.5
45.0
```

[81]:
```
casual_outlier = df[df["casual"] > upper]
len(casual_outlier)
```

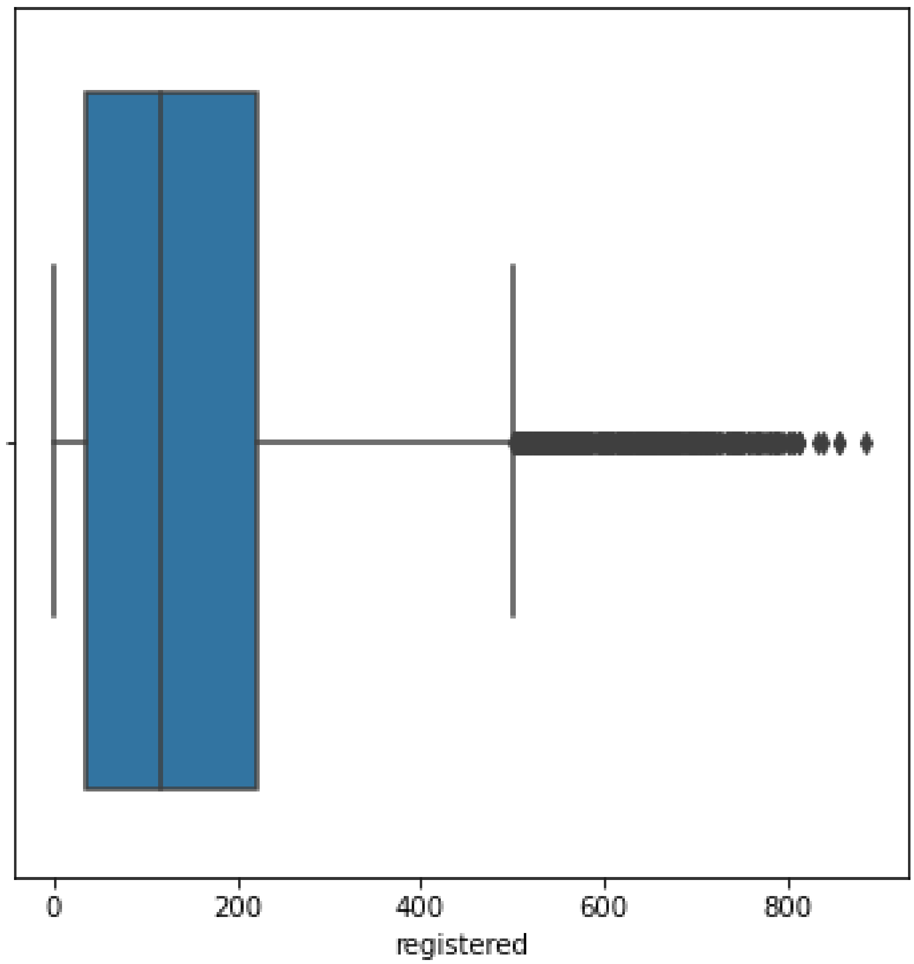
[81]: 749

approx 6 % outliers are present in casual columns

[82]:
```
len(casual_outlier) / len(df)
```

[82]: 0.06880396839977954

[83]:
```
#outlier in registered columns
plt.figure(figsize=(6,6))
sns.boxplot(data=df,
            x='registered')
plt.show()
```



[84]:
```
p_25 = df["registered"].quantile(0.25) # Q1 or p_25
p_50 = df["registered"].quantile(0.5)  # Q2 or p_50 or median
p_75 = df["registered"].quantile(0.75) # Q3 or p_75
print(p_25, p_50, p_75)
```

```
36.0 118.0 222.0
```

median value is 118 for registered

[85]:
```
iqr = p_75 - p_25
lower = max(p_25 - 1.5*iqr, 0)
upper = p_75 + 1.5*iqr
print(lower, upper)
print(iqr)
```

```
0 501.0
186.0
```

[86]:
```
reg_outlier = df[df["registered"] > upper]
len(reg_outlier)
```

[86]: 423

approx 3% outlier in registerd column

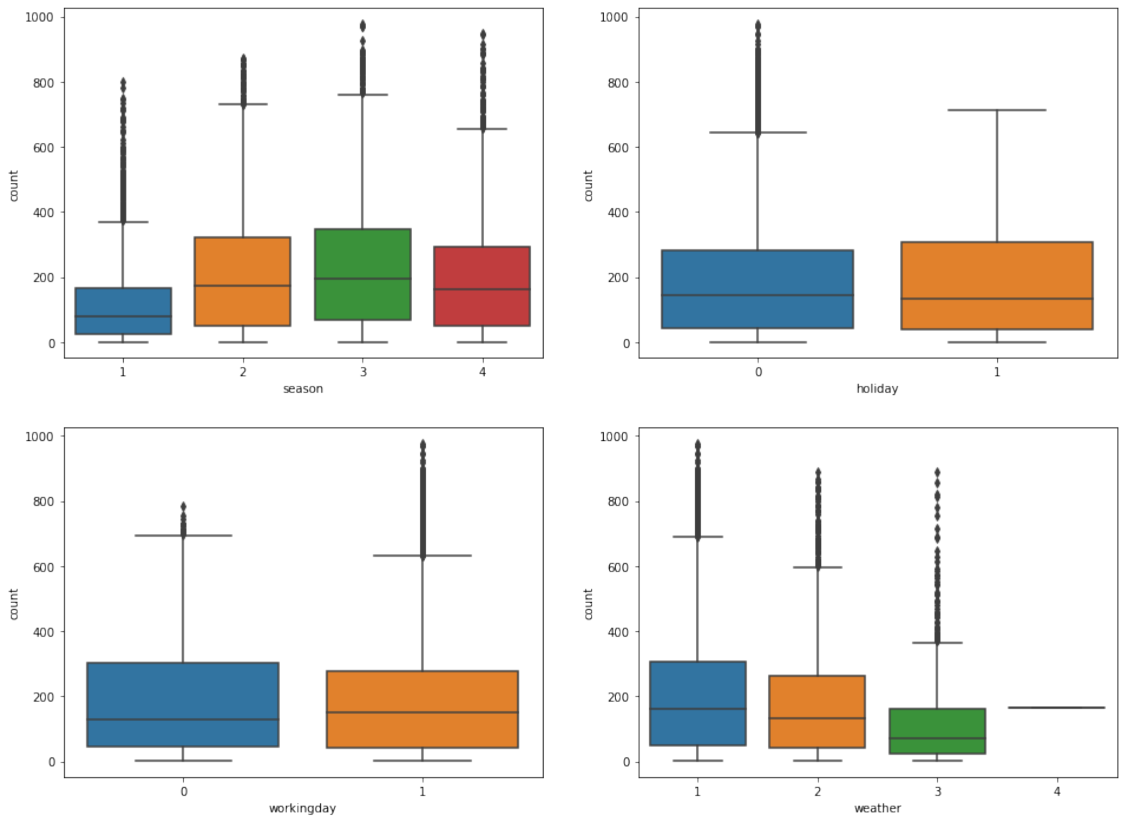
[87]:
```
len(reg_outlier) / len(df)
```

[87]: 0.03885724784126401

[88]:
```
# 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()
```



In summer and fall seasons more bikes are rented as compared to other seasons.

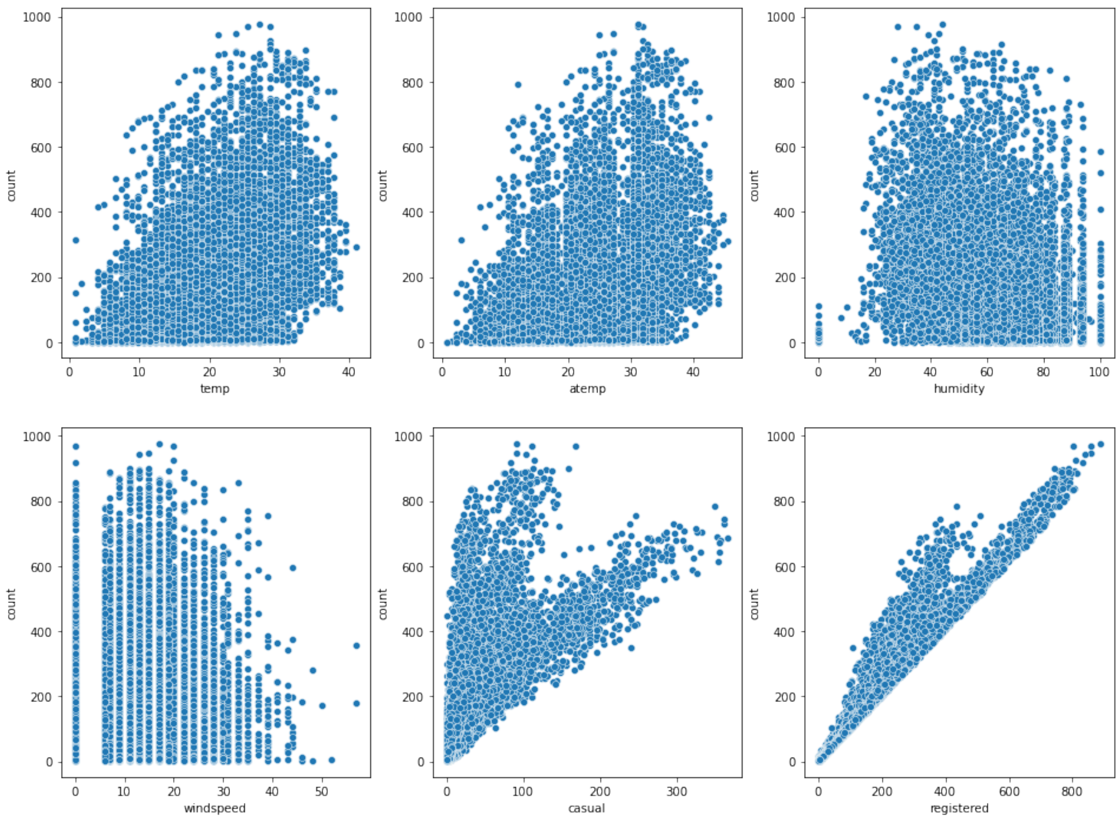
Whenever its a holiday more bikes are rented.

Whenever there is rain, thunderstorm, snow or fog, there were less bikes were rented.

[89]:
```
# 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()
```



Whenever the humidity is less than 20, number of bikes rented is very very low.

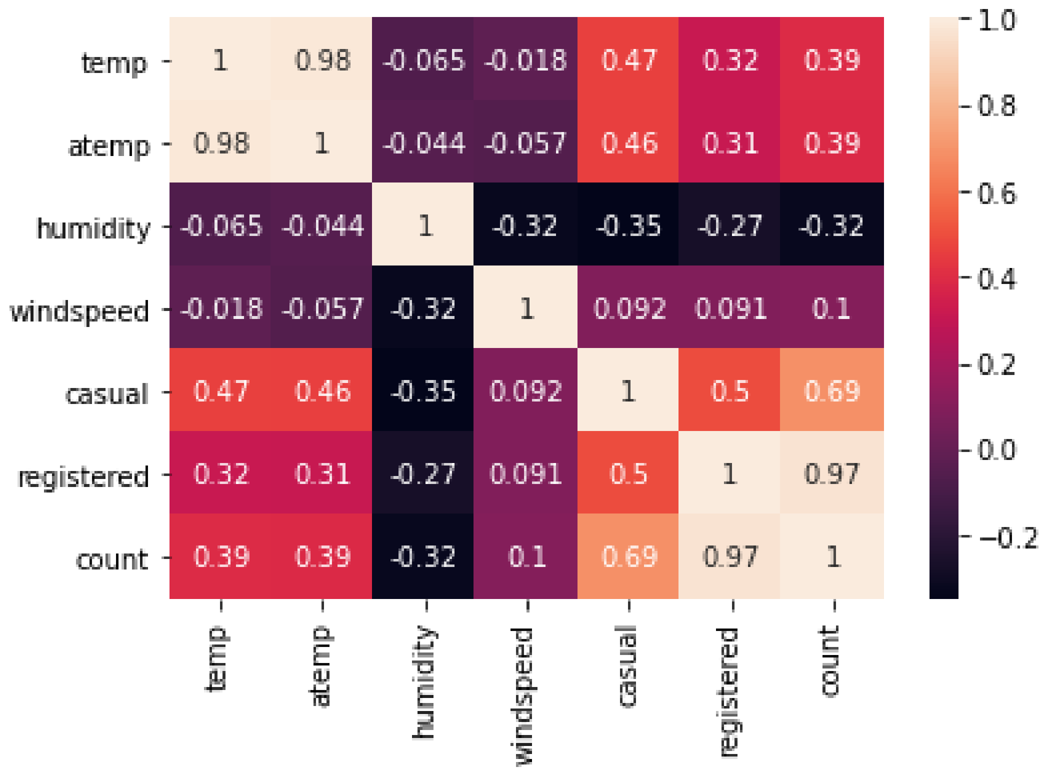
Whenever the temperature is less than 10, number of bikes rented is less.

Whenever the windspeed is greater than 35, number of bikes rented is less

[90]:
```python
# understanding the correlation between count and numerical variables
df.corr()['count']
```

[90]:
```
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
```

[91]:
```python
sns.heatmap(df.corr(), annot=True)
plt.show()
```



0.1 Hypothesis testing 1

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

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

Significance level (alpha): 0.05

we have two categorical variable so we will use chi2 test

[92]:
```
observed = pd.crosstab(df['season'], df['weather'])
print("Observed values:")
data_table
```

```
Observed values:
```

[92]:

weather	1	2	3	4
season				
1	1759	715	211	1
2	1801	708	224	0
3	1930	604	199	0
4	1702	807	225	0

[93]:
```
chi_stat, p_value, df1, exp_freq = chi2_contingency(observed)
```

[94]:
```
if p_value < 0.05:
    print("Reject H0")
else:
    print("Fail to reject H0")
```

```
Reject H0
```

we have rejected the H0 means Weather is not independent of the season

0.2 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

[101]:
```
data_group1 = df[df['workingday']==0]['count'].values
data_group2 = df[df['workingday']==1]['count'].values
```

[106]:
```
np.var(data_group1), np.var(data_group2)
```

[106]:
```
(30171.346098942427, 34040.69710674686)
```

[107]:
```
t_stat, p_value=ttest_ind(a=data_group1, b=data_group2, equal_var=True)
```

[108]:
```
if p_value < 0.05:
    print("Reject H0")
else:
    print("Fail to reject H0")
```

```
Fail to reject H0
```

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.

0.3 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

[112]:
```
# 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
```

[113]:
```
# conduct the one-way anova
f_stat, p_value=f_oneway(gp1, gp2, gp3, gp4, gp5, gp6, gp7, gp8)
```

[114]:
```
if p_value < 0.05:
    print("Reject H0")
else:
    print("Fail to reject H0")
```

```
Reject H0
```

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

0.4 INSIGHTS

In summer and fall seasons more bikes are rented as compared to other seasons.

Whenever its a holiday more bikes are rented.

Whenever there is rain, thunderstorm, snow or fog, there were less bikes were rented.

Whenever the humidity is less than 20, number of bikes rented is very very low

Whenever the temperature is less than 10, number of bikes rented is less

Whenever the windspeed is greater than 35, number of bikes rented is less

0.5 RECOMMENDATIONS

To meet the higher demand during summer and fall, the company should increase its stock of bikes available for rent, as these seasons experience more demand compared to other seasons.

During days with very low humidity, the company should consider reducing the number of bikes available for rent in its stock, as there may be lower demand for bike rentals during such conditions.

Whenever temprature is less than 10 or in very cold days, company should have less bikes.

Whenever the windspeed is greater than 35 or in thunderstorms, company should have less bikes in stock to be rented.

[ ]:

```

```