In [1]:
```
import pandas as pd
```

In [3]:
```
df=pd.read_csv("/Users/suraaj/Downloads/bike_sharing.csv")
```

In [4]:
```
df.head()
```

Out[4]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casua
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 [5]:
```
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
```

In [6]:
```
df['season'].value_counts()
```

Out[6]:
```
4    2734
2    2733
3    2733
1    2686
Name: season, dtype: int64
```

In [7]:
```
df['workingday'].value_counts()
```

Out[7]:
```
1    7412
0    3474
Name: workingday, dtype: int64
```

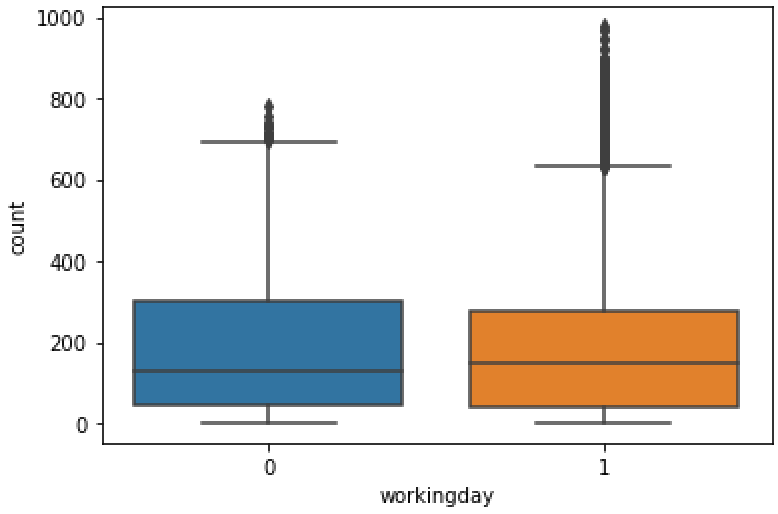
In [8]:
```
df['weather'].value_counts()
```

Out[8]:
```
1    7192
2    2834
3     859
4       1
Name: weather, dtype: int64
```

In [9]:
```
import seaborn as sbn
```

In [11]:
```
sbn.boxplot(x='workingday', y='count', data=df)
```

Out[11]:
```
<AxesSubplot:xlabel='workingday', ylabel='count'>
```

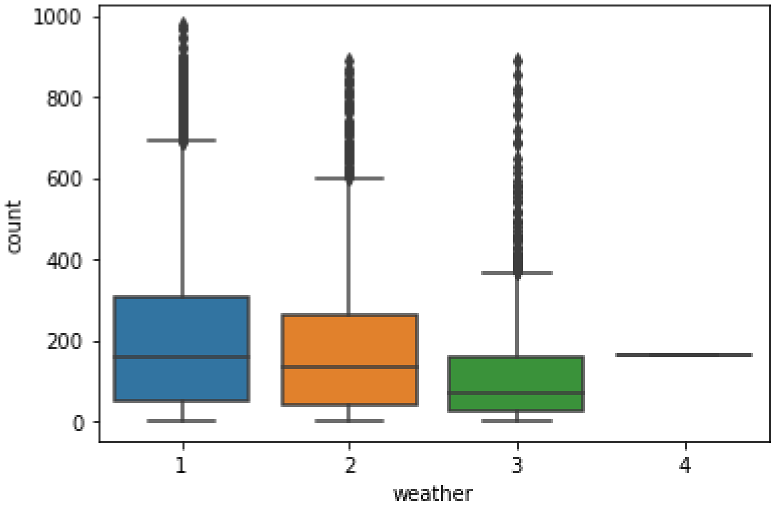


In [ ]:
```
for sample we got the that the count on working day is more than non wo
But this difference is not significant, therefore, we have to do hypoth
for wider audience
```

In [ ]:
```
#should you even remove outliers to check for wider group
#if no then why?
```

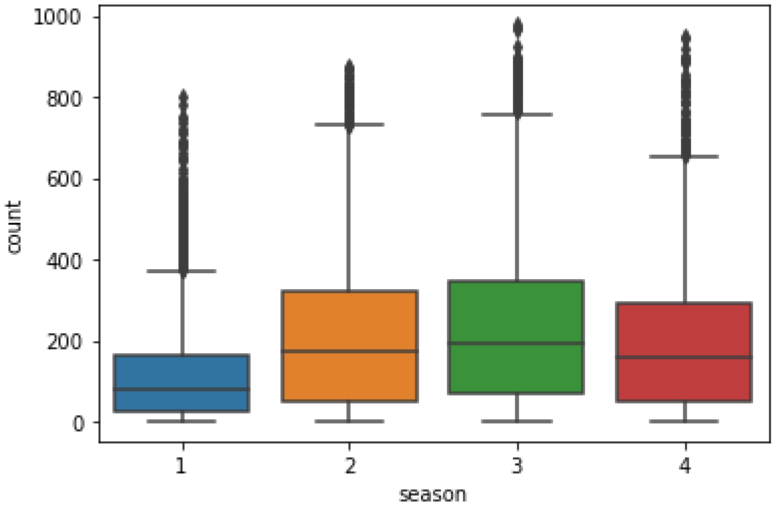
In [12]:
```
sbn.boxplot(x='weather', y='count', data=df)
```

Out[12]:
```
<AxesSubplot:xlabel='weather', ylabel='count'>
```



In [13]:
```
sbn.boxplot(x='season', y='count', data=df)
```

Out[13]:
```
<AxesSubplot:xlabel='season', ylabel='count'>
```



In [ ]:
```
#which test to be used for working day and count

Ho= The count of bikes on workingday <= the count on non working day
Ha=The count of bikes on workingday > the count on non working day

#t_test, sign=0.05
```

In [19]:
```
working= df[df['workingday']==1]['count'].sample(3400)
non_working=df[df['workingday']==0]['count'].sample(3400)
```

In [20]:
```
df.groupby('workingday')['count'].describe()
```

Out[20]:

workingday	count	mean	std	min	25%	50%	75%	max
0	3474.0	188.506621	173.724015	1.0	44.0	128.0	304.0	783.0
1	7412.0	193.011873	184.513659	1.0	41.0	151.0	277.0	977.0

In [21]:
```
from scipy.stats import ttest_ind

test_stats, p_val= ttest_ind(working, non_working, alternative='greate
```

In [22]:
```
p_val>0.05
```

Out[22]:
```
False
```

In [24]:
```
p_val
```

Out[24]:
```
0.03172239038282953
```

In [ ]:
```
######
```

In [25]:
```
df.groupby('weather')['count'].describe()
```

Out[25]:

weather	count	mean	std	min	25%	50%	75%	max
1	7192.0	205.236791	187.959566	1.0	48.0	161.0	305.0	977.0
2	2834.0	178.955540	168.366413	1.0	41.0	134.0	264.0	890.0
3	859.0	118.846333	138.581297	1.0	23.0	71.0	161.0	891.0
4	1.0	164.000000	NaN	164.0	164.0	164.0	164.0	164.0

In [26]:
```
#check the effect of weather
w1= df[df['weather']==1]['count'].sample(800)
w2= df[df['weather']==2]['count'].sample(800)
w3= df[df['weather']==3]['count'].sample(800)
```

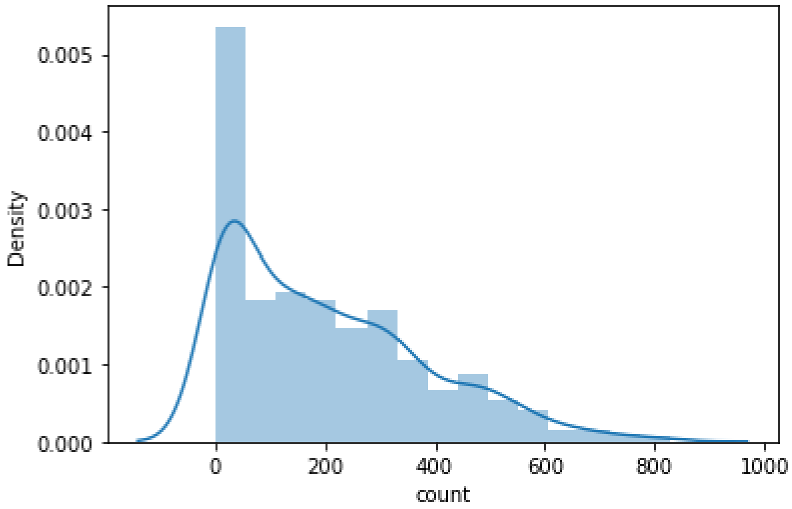
In [ ]:
```
#anova
Ho= the count of bikes are independent of weather
Ha= the count of bikes is affected by weather

# assumptions of anova
#1. Normal - QQPLOT, DISTPLOT, SHAPIRO
#2. should have equal variance -- No , DESCRIBE, LEVENE
```

In [27]:
```
import seaborn as sbn
```

```
In [29]: sbn.distplot(w1)
```

```
/Users/suraaj/opt/anaconda3/lib/python3.9/site-packages/seaborn/dist
ributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to
use either `displot` (a figure-level function with similar flexibili
ty) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

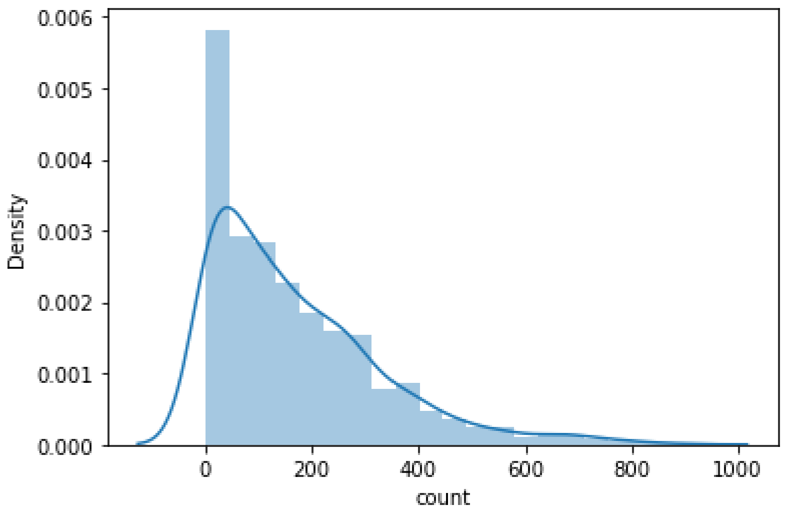
```
Out[29]: <AxesSubplot:xlabel='count', ylabel='Density'>
```



```
In [30]: sbn.distplot(w2)
```

```
/Users/suraaj/opt/anaconda3/lib/python3.9/site-packages/seaborn/dist
ributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to
use either `displot` (a figure-level function with similar flexibili
ty) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

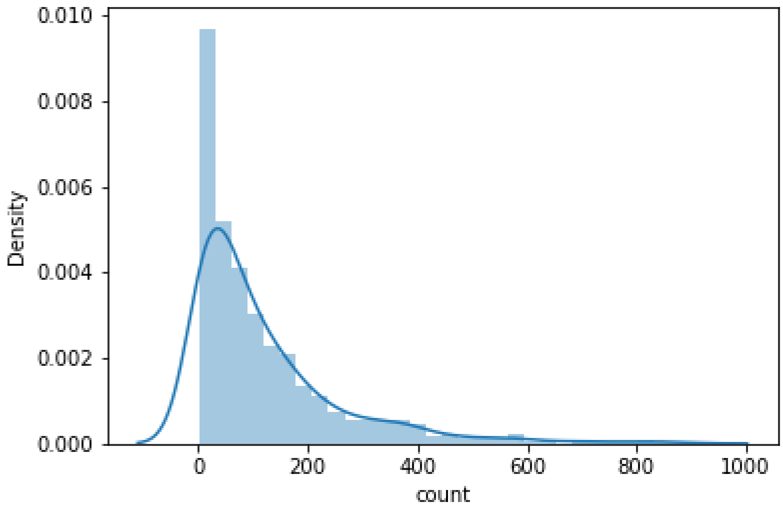
```
Out[30]: <AxesSubplot:xlabel='count', ylabel='Density'>
```



In [31]:
```
sbn.distplot(w3)
```

```
/Users/suraaj/opt/anaconda3/lib/python3.9/site-packages/seaborn/dist
ributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to
use either `displot` (a figure-level function with similar flexibili
ty) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

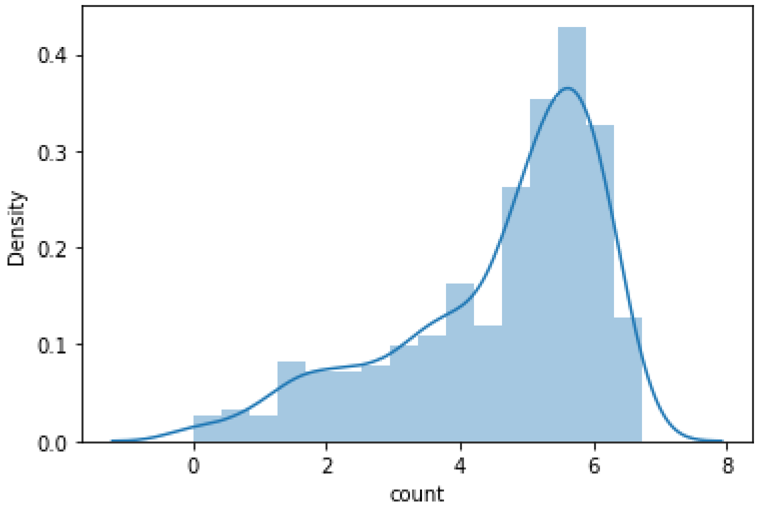
Out[31]:
```
<AxesSubplot:xlabel='count', ylabel='Density'>
```



In [32]:
```
import numpy as np
sbn.distplot(np.log(w1))
```

```
/Users/suraaj/opt/anaconda3/lib/python3.9/site-packages/seaborn/dist
ributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to
use either `displot` (a figure-level function with similar flexibili
ty) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

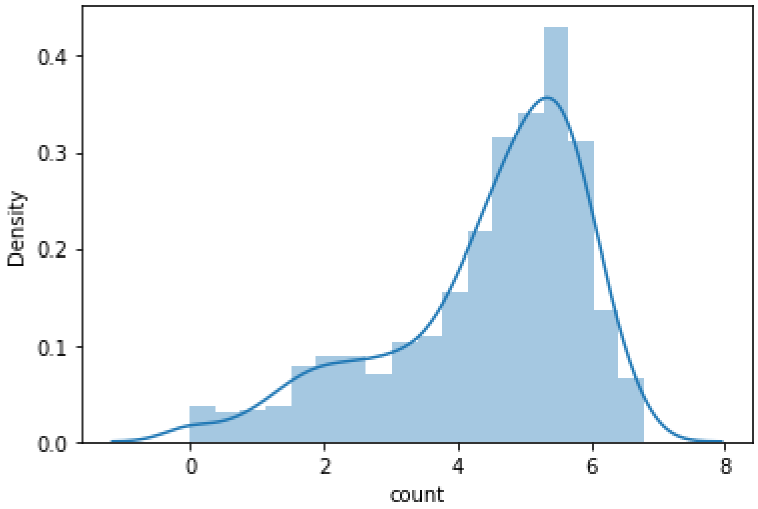
Out[32]:
```
<AxesSubplot:xlabel='count', ylabel='Density'>
```



In [33]:
```
sbn.distplot(np.log(w2))
```

```
/Users/suraaj/opt/anaconda3/lib/python3.9/site-packages/seaborn/dist
ributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to
use either `displot` (a figure-level function with similar flexibili
ty) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

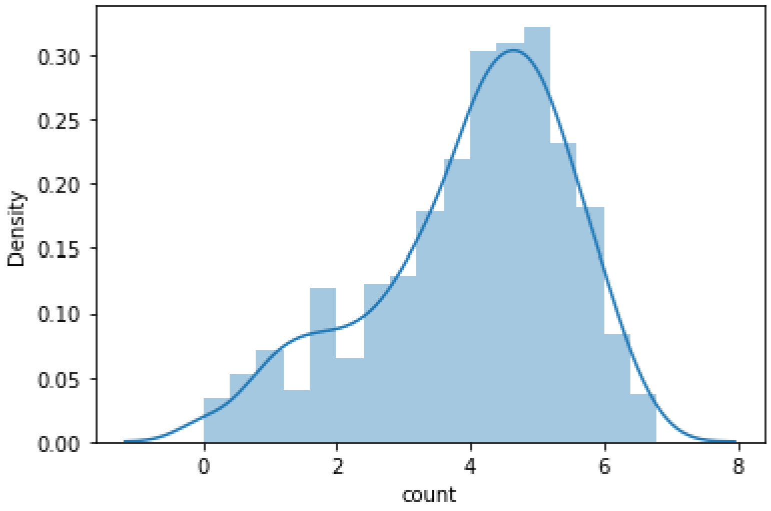
Out[33]:
```
<AxesSubplot:xlabel='count', ylabel='Density'>
```



In [34]:
```
sbn.distplot(np.log(w3))
```

```
/Users/suraaj/opt/anaconda3/lib/python3.9/site-packages/seaborn/dist
ributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to
use either `displot` (a figure-level function with similar flexibili
ty) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[34]:
```
<AxesSubplot:xlabel='count', ylabel='Density'>
```



In [35]:
```
from scipy.stats import shapiro

t_test, p_value= shapiro(w1)
```

In [37]:
```
p_value>0.05
```

Out[37]:
```
False
```

In [38]:
```
#normality condition is failing
```

In [39]:
```
from scipy.stats import levene

t_test, p_value= levene(w1,w2,w3)
```

In [41]:
```
p_value<0.05
```

Out[41]:
```
True
```

In [ ]:
```
#Kruskal wallis test

-- Appying code
https://machinelearningmastery.com/statistical-hypothesis-tests-in-pyt
```

In [42]:
```
from scipy.stats import f_oneway

t_test, p_value= f_oneway(w1,w2,w3)
```

In [44]:
```
p_value < 0.05
```

Out[44]:
```
True
```

In [ ]:
```
-----

#season dependency -- test
```