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	casual
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0	3
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0	8
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0	5
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0	3
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0	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.weather.value_counts()
```

Out[7]:

```
1    7192
2    2834
3     859
4       1
Name: weather, dtype: int64
```

In [8]:

```
df.workingday.value_counts()
```

Out[8]:

```
1    7412
0    3474
Name: workingday, dtype: int64
```

In [9]:

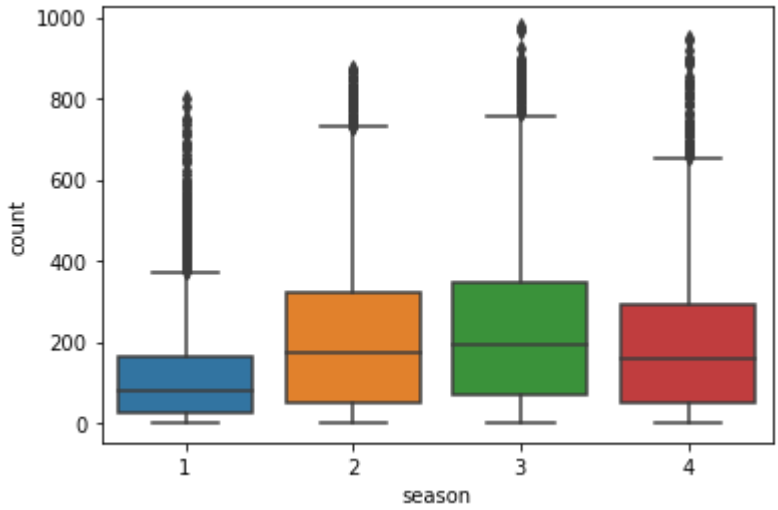
```
import seaborn as cat
```

In [11]:

```
cat.boxplot(y='count', x='season', data =df)
```

Out[11]:

```
<AxesSubplot:xlabel='season', ylabel='count'>
```



In [12]:

```
#iqr
q1=df['count'].quantile(0.25)
q3=df['count'].quantile(0.75)
iqr=q3-q1
```

In [13]:

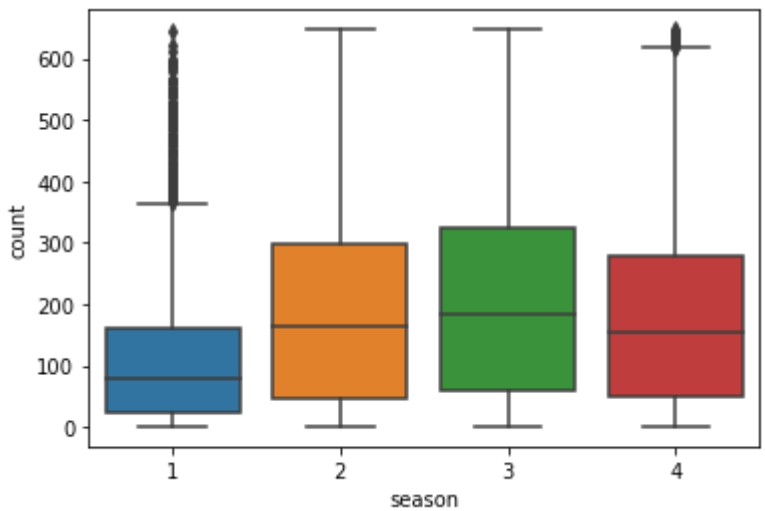
```
df_copy=df[(df['count']>=q1-1.5*iqr) & (df['count']<=q3+1.5*iqr)]
```

In [14]:

```
cat.boxplot(y='count', x='season', data =df_copy)
```

Out[14]:

```
<AxesSubplot:xlabel='season', ylabel='count'>
```

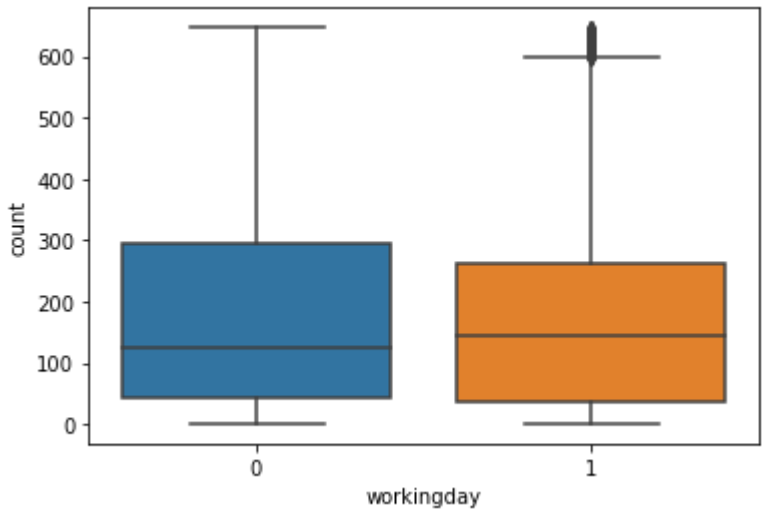


In [15]:

```
cat.boxplot(y='count', x='workingday', data =df_copy)
```

Out[15]:

```
<AxesSubplot:xlabel='workingday', ylabel='count'>
```

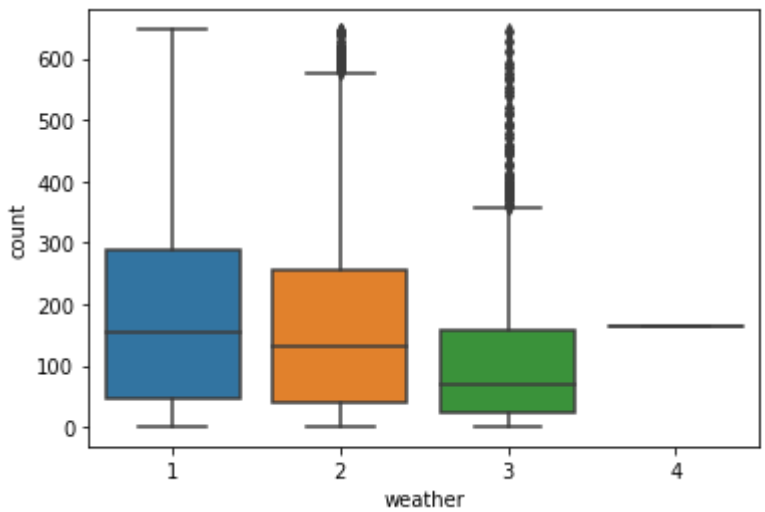


In [16]:

```
cat.boxplot(y='count', x='weather', data =df_copy)
```

Out[16]:

```
<AxesSubplot:xlabel='weather', ylabel='count'>
```



In [18]:

```
df_copy.shape
```

Out[18]:

```
(10586, 12)
```

hypothesis

Working Day has effect on number of electric cycles rented

Ho= Count of cycles rentend on working day is less than or equal to the count of cycles rented on a non working day

Ha= Count of cycles rentend on working day is greater to the count of cycles rented on a non working day

In [21]:

```
df_copy.workingday.value_counts()
```

Out[21]:

```
1    7161
0    3425
Name: workingday, dtype: int64
```

In [22]:

```
workingday=df_copy[df_copy['workingday']==1]['count'].sample(3425)
non_workingday=df_copy[df_copy['workingday']==0]['count'].sample(3425)
```

In [24]:

```
df_copy.groupby('workingday')['count'].describe()
```

Out[24]:

workingday	count	mean	std	min	25%	50%	75%	max
0	3425.0	181.373723	164.290054	1.0	43.0	125.0	296.0	647.0
1	7161.0	173.011591	152.358993	1.0	38.0	143.0	262.0	646.0

In [23]:

```
from scipy.stats import ttest_ind
```

In [25]:

```
test_stat, p_value=ttest_ind(workingday, non_workingday, equal_var=False, alternativ
```

In [26]:

```
p_value
```

Out[26]:

```
0.9645645904955159
```

In [34]:

```
test_stat, p_value=ttest_ind(np.log(workingday), np.log(non_workingday), equal_var=F
```

In [35]:

```
p_value
```

Out[35]:

```
0.9959902743980567
```

In [36]:

```
#weather
df_copy.groupby('weather')['count'].describe()
```

Out[36]:

weather	count	mean	std	min	25%	50%	75%	max
1	6965.0	187.329218	161.581066	1.0	45.0	153.0	287.0	647.0
2	2770.0	166.117690	146.992422	1.0	39.0	130.0	254.0	646.0
3	850.0	111.862353	121.233389	1.0	23.0	70.5	157.0	646.0
4	1.0	164.000000	NaN	164.0	164.0	164.0	164.0	164.0

In [37]:

```
weather_1=df_copy[df_copy['weather']==1]['count'].sample(850)
weather_2=df_copy[df_copy['weather']==2]['count'].sample(850)
weather_3=df_copy[df_copy['weather']==3]['count'].sample(850)
```

#assumption testing

1. normality - qq plot, displot, shapiro wilk test
2. equal variance - levene test, group by describe

In [38]:

```
from scipy.stats import shapiro
```

In [45]:

```
stats, p_value=shapiro(np.log(df_copy['count']).sample(4999))
```

In [46]:

```
p_value
```

Out[46]:

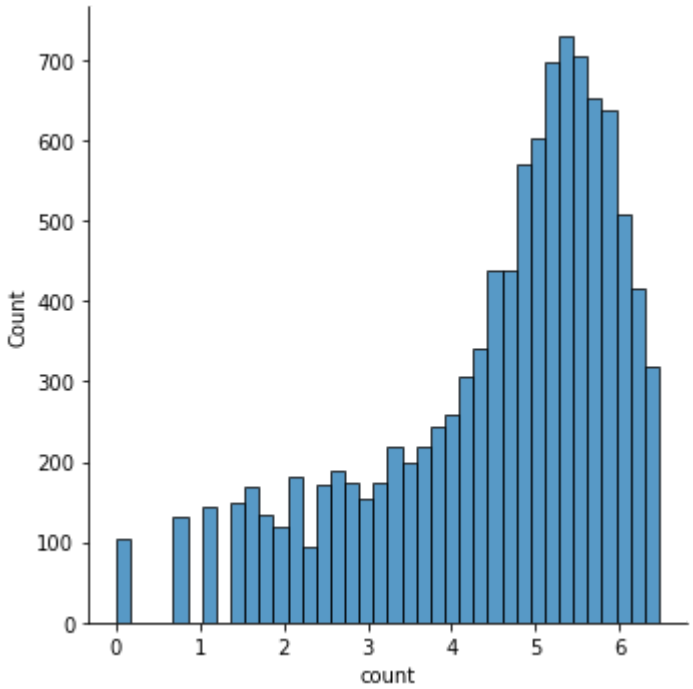
```
0.0
```

In [48]:

```
cat.displot(np.log(df_copy['count']))
```

Out[48]:

```
<seaborn.axisgrid.FacetGrid at 0x7fa6fa0e1f40>
```



In [49]:

```
from scipy.stats import levene
```

In [50]:

```
w, p_value=levene(weather_1,weather_2,weather_3)
```

In [51]:

```
p_value
```

Out[51]:

```
8.564371904987799e-21
```

In [53]:

```
#Ho=same count of bicycles rented
#HA= don't have same count
```

In [54]:

```
from scipy.stats import f_oneway
```

In [55]:

```
test_stats, p_value=f_oneway(weather_1,weather_2,weather_3)
```

In [56]:

```
p_value
```

Out[56]:

```
1.3580059758823905e-31
```

In [ ]:

```

```