Problem statement: Yulu has recently suffered considerable dips in its revenues. They want to understand the factors affecting the demand for these shared electric cycles in the Indian market. With the help of Hypothesis Testing we are going to analyze the various factors affecting the usage of the their shared electric vehicles and what is their role in it. We are going to test various factors against the usage and see how statistically significant they are and how useful they can be in making business decisions. From the datatset, we can also check what factors contribute the most in making useful inferences.

In [1]:

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

In [2]:

```
from scipy.stats import ttest_ind # Numeric Vs categorical
from scipy.stats import chi2 # Distribution (cdf etc.)
from scipy.stats import chisquare # Statistical test (chistat, pvalue)

from scipy.stats import chi2_contingency # Categorical Vs Categorical
from scipy.stats import f_oneway # Numeric Vs categorical
```

In [3]:

```
yulu=pd.read_csv('C:/Prakruthi/DSML/YULU - Case study/bike_sharing.csv')
```

In [4]:

```
yulu.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]:

```
#Checking if there are missing values:
yulu.isnull().sum()
```

Out[5]:

```
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
```

In [6]:

```
yulu['workingday'].value_counts().sort_values(ascending=False)
```

Out[6]:

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

In [7]:

```
yulu.groupby('workingday')['count'].sum()
```

Out[7]:

```
workingday
0     654872
1    1430604
Name: count, dtype: int64
```

In [8]:

```
yulu['weather'].value_counts().sort_values(ascending=False)
```

Out[8]:

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

In [9]:

```
yulu.groupby('weather')['count'].sum()
```

Out[9]:

```
weather
1    1476063
2     507160
3     102089
4        164
Name: count, dtype: int64
```

In [10]:

```
yulu['season'].value_counts().sort_values(ascending=False)
```

Out[10]:

```
4    2734
2    2733
3    2733
1    2686
Name: season, dtype: int64
```

In [11]:

```
yulu.groupby('season')['count'].sum()
```

Out[11]:

```
season
1    312498
2    588282
3    640662
4    544034
Name: count, dtype: int64
```

```
Basic metrics: We have a total of 10886 rows which represent the data on an hourly basis
where each row represents the total no.of bikes rented in that given hours and 12
attributes/columns which are going to use for our hypothesis testing

Below are some of the metrics based on the given dataset

1. Since there are no missing values, no action required
2. There are 7412 rows for working days and 3474 rows for non-working days which includes
both weekends and holidays.
3. Also, there are total 1430604 rented vehicles on working days and 654872 rented
vehicles on non-working days
4. There are below number of rented vehicles for various weather conditions(note that
weather conditions can vary across the day unlike seasons)
    1: Clear, Few clouds, partly cloudy, partly cloudy - 1476063
    2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist - 507160
    3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered
clouds - 102089
    4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog - 164
5. There are below number of rented vehicles for various seasons
    1: spring - 312498
    2: summer - 588282
    3: fall - 640662
    4: winter - 544034
```

In [12]:

```
#converting working day, season and weather columns to object type from integer to procedd
yulu["holiday"] = yulu["holiday"].astype('object')
yulu["workingday"] = yulu["workingday"].astype('object')
yulu["weather"] = yulu["weather"].astype('object')
yulu["season"] = yulu["season"].astype('object')
```

In [13]:

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

Out[13]:

	datetime	season	holiday	workingday	weather	temp	atemp	humi
count	10886	10886.0	10886.0	10886.0	10886.0	10886.00000	10886.000000	10886.000
unique	10886	4.0	2.0	2.0	4.0	NaN	NaN	N
top	2011-01-03 11:00:00	4.0	0.0	1.0	1.0	NaN	NaN	N
freq	1	2734.0	10575.0	7412.0	7192.0	NaN	NaN	N
mean	NaN	NaN	NaN	NaN	NaN	20.23086	23.655084	61.886
std	NaN	NaN	NaN	NaN	NaN	7.79159	8.474601	19.245
min	NaN	NaN	NaN	NaN	NaN	0.82000	0.760000	0.000
25%	NaN	NaN	NaN	NaN	NaN	13.94000	16.665000	47.000
50%	NaN	NaN	NaN	NaN	NaN	20.50000	24.240000	62.000
75%	NaN	NaN	NaN	NaN	NaN	26.24000	31.060000	77.000
max	NaN	NaN	NaN	NaN	NaN	41.00000	45.455000	100.000

Some additional observations

1. By looking at the casual, registered and total count of rented vehicles we can see that are a outliers. We can check this further from plots
2. The mean is higher than the median which means the data is right skewed and there are outliers in the right side of the distribution

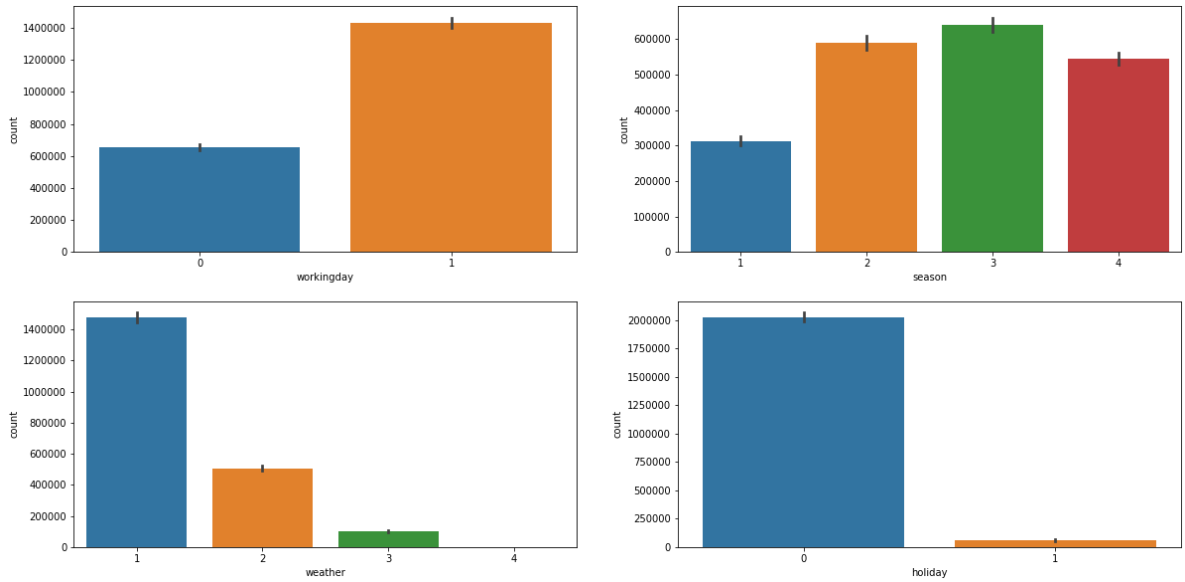
Univariate analysis

In [14]:

```
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(20, 10))
sns.barplot(x = 'workingday',y = 'count',data = yulu,estimator = np.sum,ax=axis[0,0])
sns.barplot(x = 'season',y = 'count',data = yulu,estimator = np.sum,ax=axis[0,1])
sns.barplot(x = 'weather',y = 'count',data = yulu,estimator = np.sum,ax=axis[1,0])
sns.barplot(x = 'holiday',y = 'count',data = yulu,estimator = np.sum,ax=axis[1,1])
```

Out[14]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x219413faef0>
```



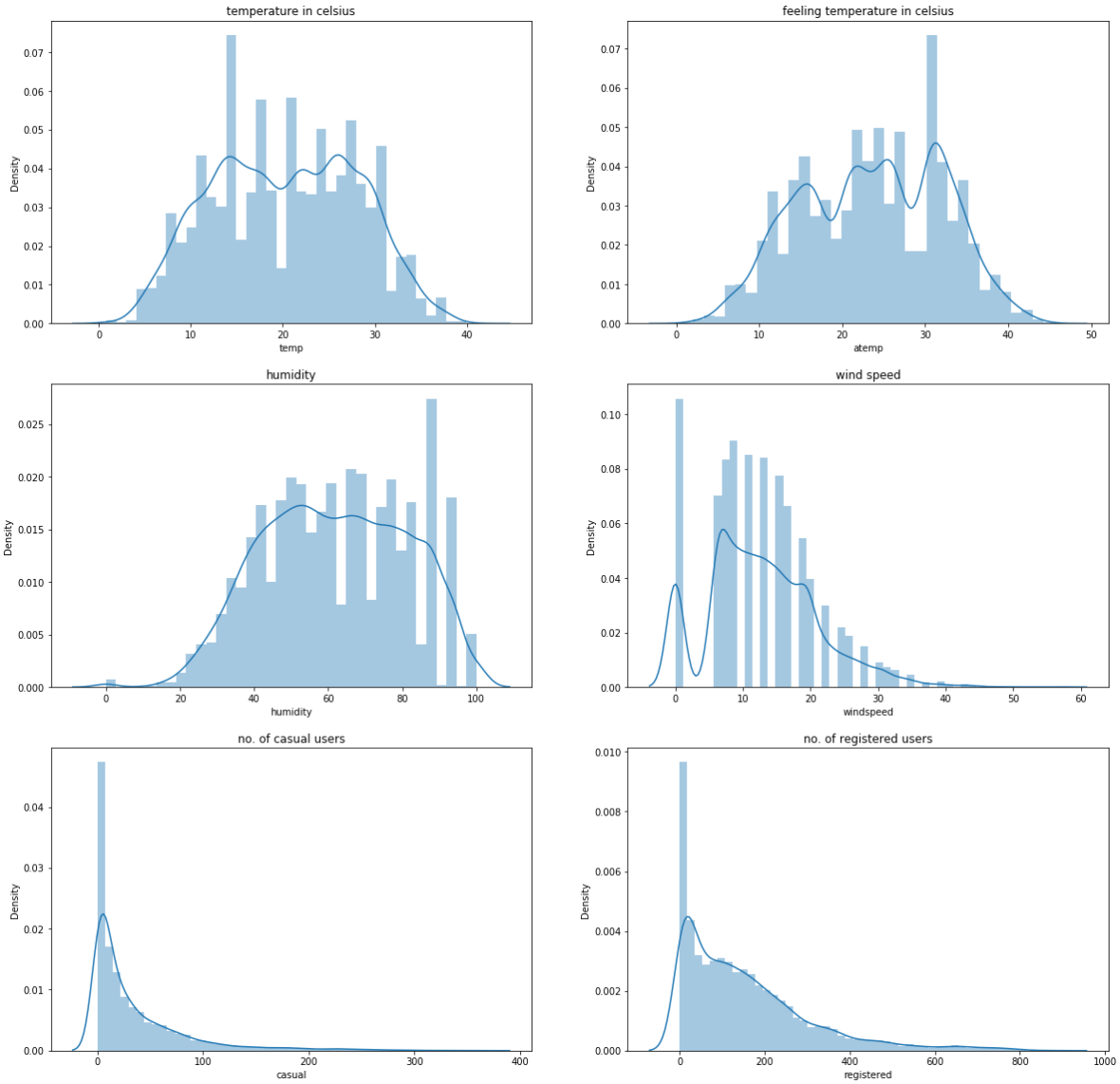
Below are the observations/analysis we can make based on the above plot

1. Working days have more than double the number of rented vehicles than the non-working days
2. Season 3(fall) has seen the highest number of rented vehicles count followed by 4(winter), 2(summer) and 1(spring)
3. As expected when weather is 1(Clear, Few clouds, partly cloudy, partly cloudy) the no of rented vehicles is the highest and it is more than the other 3 weather types combined.
4. The number of rented vehicles on a non-holiday is very high compared to a holiday. This could also be since the number of non-holidays will be much higher in a given dataset than the holidays. This attribute may not be very useful for our testing.

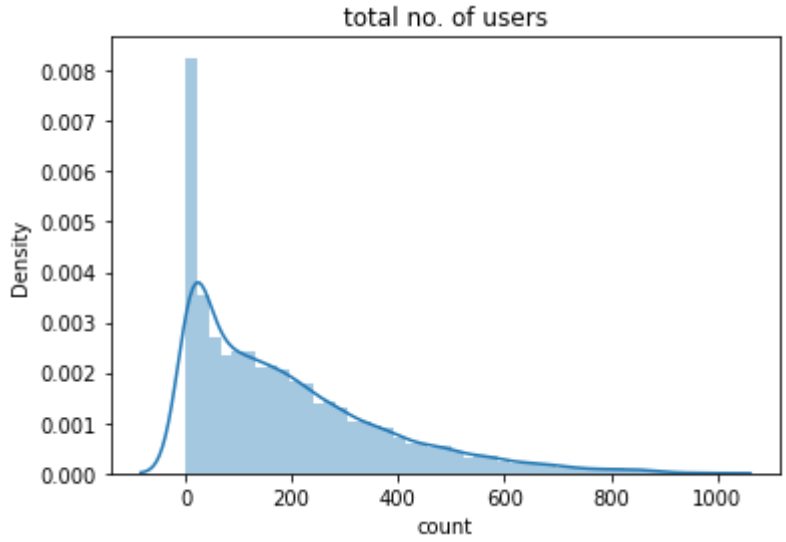
In [15]:

```
#data = pd.DataFrame(yulu['temp','atemp','humidity','windspeed','casual','registered'])
fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(20, 20))
sns.distplot(yulu['temp'],ax=axis[0,0]).set_title('temperature in celsius')
sns.distplot(yulu['atemp'],ax=axis[0,1]).set_title('feeling temperature in celsius')
sns.distplot(yulu['humidity'],ax=axis[1,0]).set_title('humidity')
sns.distplot(yulu['windspeed'],ax=axis[1,1]).set_title('wind speed')
sns.distplot(yulu['casual'],ax=axis[2,0]).set_title('no. of casual users')
sns.distplot(yulu['registered'],ax=axis[2,1]).set_title('no. of registered users')
plt.show()
sns.distplot(yulu['count']).set_title('total no. of users')
plt.show()
```

```
C:\Users\sanke\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\sanke\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\sanke\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\sanke\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\sanke\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\sanke\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```




```
C:\Users\sanke\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```



Observations from the above plots for continous variables
1. Both no. of casual users and no. of registered users follow a lognormal distribution. Hence the total no. of users also follow a lognormal distribution. We will further see how to use this data for our hypothesis testing
2. Other attributes as well do not follow a particular distribution. Since we will not be using the other variables, we do not need further analysis on these

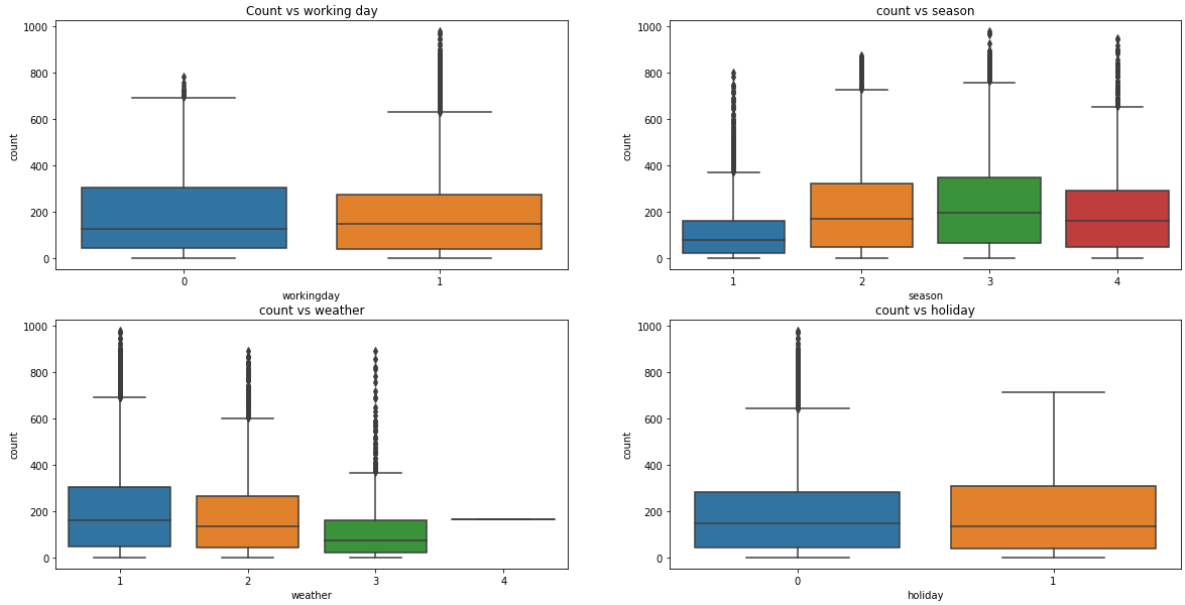
Bivariate analysis

In [16]:

```
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(20, 10))
sns.boxplot(data=yulu, y='count',x='workingday',ax=axis[0,0]).set_title('Count vs working d
sns.boxplot(data=yulu, y='count',x='season',ax=axis[0,1]).set_title('count vs season')
sns.boxplot(data=yulu, y='count',x='weather',ax=axis[1,0]).set_title('count vs weather')
sns.boxplot(data=yulu, y='count',x='holiday',ax=axis[1,1]).set_title('count vs holiday')
```

Out[16]:

```
Text(0.5, 1.0, 'count vs holiday')
```



Observations :

1. The median values and also the min,25th and 75th percentile values are almost the same for both working and non-working days. This means the no. of rented vehicles on an hourly basis seem to follow the same pattern for both. We are going to test this soon. Also, the outliers for working days are much more than non-working days. Hence it is difficult to conclude whether or not both follow the same pattern

2. Seasons 2 and 3 have almost the same min,25th and 75th percentile values with season 3 having more outliers than season 2. From this plot it is quite evident that Season 3 followed by season 2 has more no. of rented vehicles usage than the other 2 season.
3. As we can see Weather 1 has the highest no.of rented vehicles followed by weather 2 and 3. Weather 4 has almost no usage at all
4. From the plot, by looking at the median, 25ht and 75th percentile values it seems like both holidays and non-holidays have a similar usage but non-holidays have a lot of outliers which means they are not similar.

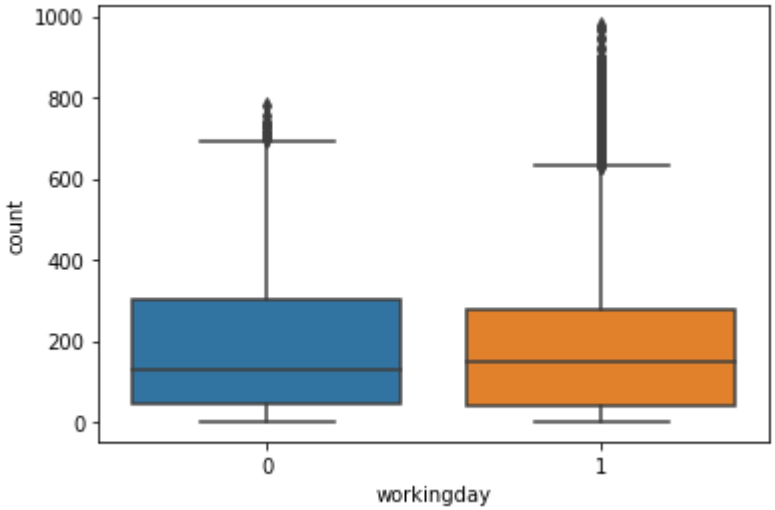
Hypothesis Testing

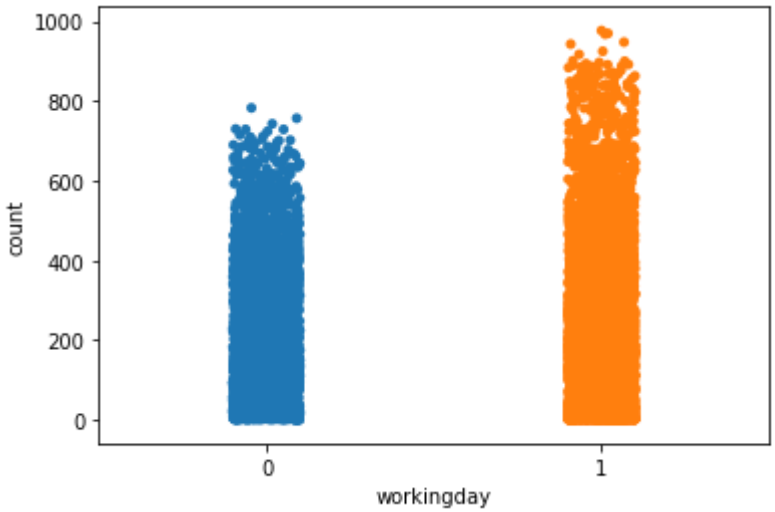
Testing whether Working Day has effect on number of electric cycles rented

Visual analysis

In [17]:

```
sns.boxplot(data=yulu, y='count',x='workingday')
plt.show()
sns.stripplot(y = yulu['count'], x = yulu['workingday'])
plt.show()
```





From the above plots we can seem to infer that the working days have more no. of rented vehicles being used than the non-working days. We can test this inference further using hypothesis testing

Hypothesis formulation

Setting up the null and alternate hypothesis
H0 : Working Day - whether its a working day or a non-working day does not have an effect on the number of electric cycles
rented
Ha : Working Day - whether its a working day or a non-working day has an effect on the number of electric cycles rented

To test the above we can use the 2-Sample T-Test since we are comparing two samples i.e working and non-working day

In [18]:

```
yulu.groupby(["workingday"])["count"].mean()
```

Out[18]:

```
workingday
0    188.506621
1    193.011873
Name: count, dtype: float64
```

In [19]:

```
#create 2 samples - working and non-working days
yulu_working = yulu[yulu["workingday"]==1]
yulu_non_working = yulu[yulu["workingday"]==0]
```

In [20]:

```
# H0 : Working Day - the number of electric cycles rented on a working day is the same as a
# Ha : Working Day - the number of electric cycles rented on a working day is not equal to
alpha = 0.05 # testing the null hypothesis at 95% confidence level
test_stat, p_value = ttest_ind(yulu_working["workingday"], yulu_non_working["workingday"],a
print("test statistic:",test_stat)
print("p value :",p_value)
if p_value < alpha:
    print("Reject H0")
else:
    print("Fail to reject H0")
```

```
test statistic: inf
p value : 0.0
Reject H0
```

Since the p-value is 0 from the above test we are going to Reject the null hypothesis
Hence we can very clearly conclude that the effect of working and non-working days are not the same for the no. of rented cycles

Further we are going to test if the no. of rented vehicles are more on working days compared to non-working days

In [21]:

```
# H0 : Working Day - the number of electric cycles rented on a working day is the same as a
# Ha : Working Day - the number of electric cycles rented on a working day is greater than
# non-working day
alpha = 0.05 # testing the null hypothesis at 95% confidence level
test_stat, p_value = ttest_ind(yulu_working["workingday"], yulu_non_working["workingday"],a
print("test statistic:",test_stat)
print("p value :",p_value)
if p_value < alpha:
    print("Reject H0")
else:
    print("Fail to reject H0")
```

```
test statistic: inf
p value : 0.0
Reject H0
```

```
Since the p-value is 0 from the above test we are going to Reject the null hypothesis
Hence we can very clearly conclude that the no. of electric vehicles rented is much more
on working days than non-working days
```

```
To be very sure, we are also going to test if it is possible that the no. of rented
cycles is more on non-working days compared to working days
```

In [22]:

```
# H0 : Working Day - the number of electric cycles rented on a working day is the same as a
# Ha : Working Day - the number of electric cycles rented on a working day is lesser than t
# non-working day
alpha = 0.05 # testing the null hypothesis at 95% confidence level
test_stat, p_value = ttest_ind(yulu_working["workingday"], yulu_non_working["workingday"],a
print("test statistic:",test_stat)
print("p value :",p_value)
if p_value < alpha:
    print("Reject H0")
else:
    print("Fail to reject H0")
```

```
test statistic: inf
p value : 1.0
Fail to reject H0
```

```
Since the p-value is 1.0 from the above test we fail to Reject the null hypothesis
Hence we can very clearly conclude that we cannot prove that the no. of electric vehicles
rented on non-working days are more than working days
```

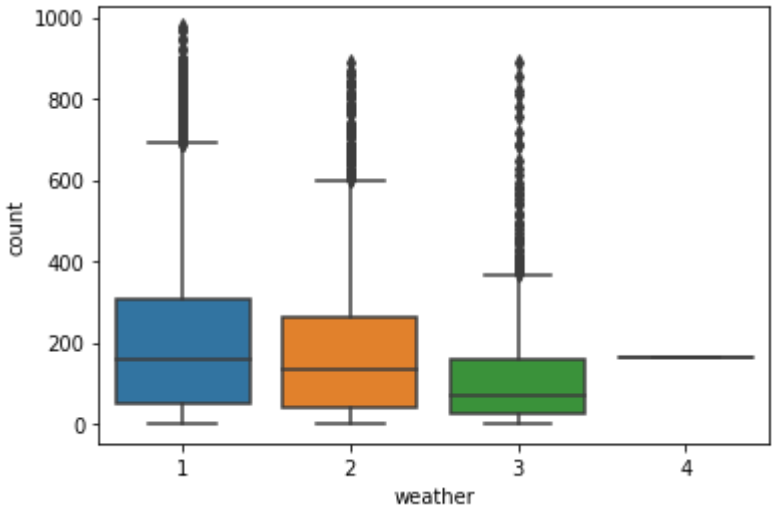
```
# Inference from the analysis
From the above test we can conclude that no. of rented vehicles are more on working days
compared to non-working days
```

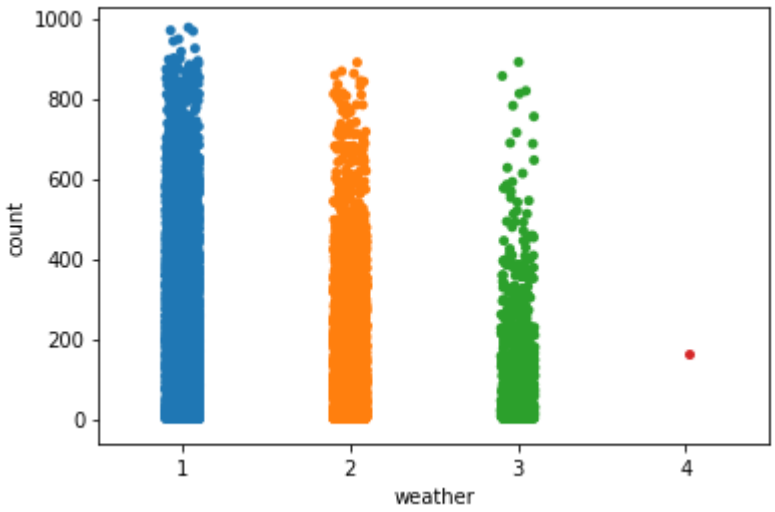
Testing whether No. of cycles rented is similar or different in different weather

```
Visual Analysis
```

In [23]:

```
sns.boxplot(data=yulu, y='count',x='weather')
plt.show()
sns.stripplot(y = yulu['count'], x = yulu['weather'])
plt.show()
```





From the above plots we can seem to infer that differrent weather types have diffrent effect on the no. of rented electric vehicles. We can test this inference further using hypothesis testing

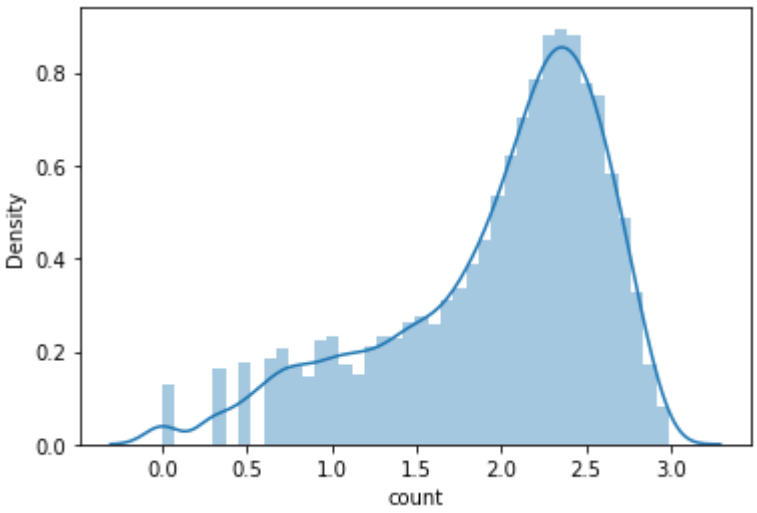
In [24]:

```
sample = np.log10(yulu['count'])
sns.distplot(sample)
```

```
C:\Users\sanke\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[24]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x219440acef0>
```



As we have seen before, the distribution of 'count' does not follow a normal distribution. Hence we have tried to transform it into a log normal distribution. Since even after the transformation the distribution still is not normal. hence we are going ahead with the hypothesis testing. Since ANNOVA is a robust test, the difference in distribution will not affect the efficiency of the test

To test the above we can use the ANNOVA test since we are comparing 4 different types of weathers against the count of rented vehicles for them. Since this is a numerical vs categorical testing, ANNOVA is the right fit

Hypothesis formulation

Setting up the null and alternate hypothesis
H0 : Weather does not have an impact on the No. of cycles rented
Ha : No. of cycles rented is dependent on the weather and the no. varies with change in weather

In [25]:

```
weather_1 = yulu[yulu["weather"]==1]["count"]
weather_2 = yulu[yulu["weather"]==2]["count"]
weather_3 = yulu[yulu["weather"]==3]["count"]
weather_4 = yulu[yulu["weather"]==4]["count"]
```

In [26]:

```
f_stat, p_value = f_oneway(weather_1,weather_2,weather_3,weather_4)
alpha = 0.05 # testing the null hypothesis at 95% confidence level
print("f_stat : ",f_stat)
print("p-value : ",p_value)
if p_value < alpha:
    print("Reject H0")
else:
    print("Fail to reject H0")
```

```
f_stat :  65.53024112793271
p-value :  5.482069475935669e-42
Reject H0
```

Since the p-value is very very low i.e 5.482069475935669e-42 from the above test we are going to Reject the null hypothesis.
Hence we can very clearly conclude that the no. of electric vehicles rented is different for different weather types

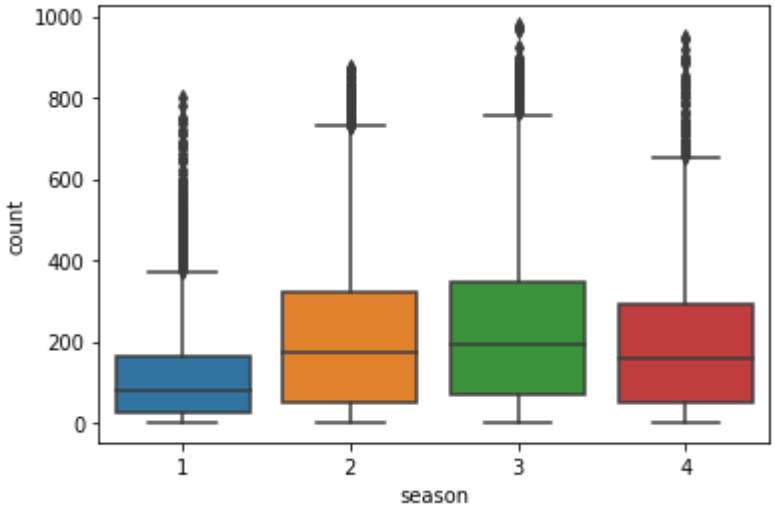
Inference from the analysis
We can very clearly conclude that the no. of electric vehicles rented is different for different weather types

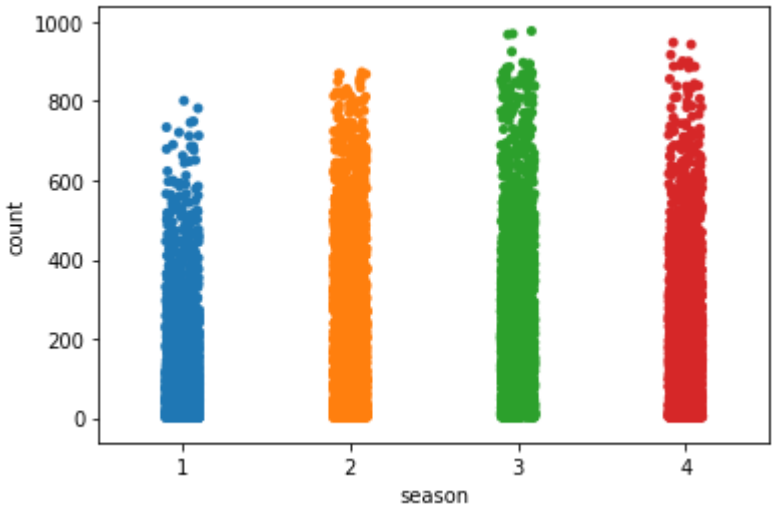
Testing whether No. of cycles rented is similar or different in different seasons

Visual Analysis

In [27]:

```
sns.boxplot(data=yulu, y='count',x='season')
plt.show()
sns.stripplot(y = yulu['count'], x = yulu['season'])
plt.show()
```





From the above plots we can seem to infer that differrent seasons have different effect on the no. of rented electric vehicles. We can test this inference further using hypothesis testing

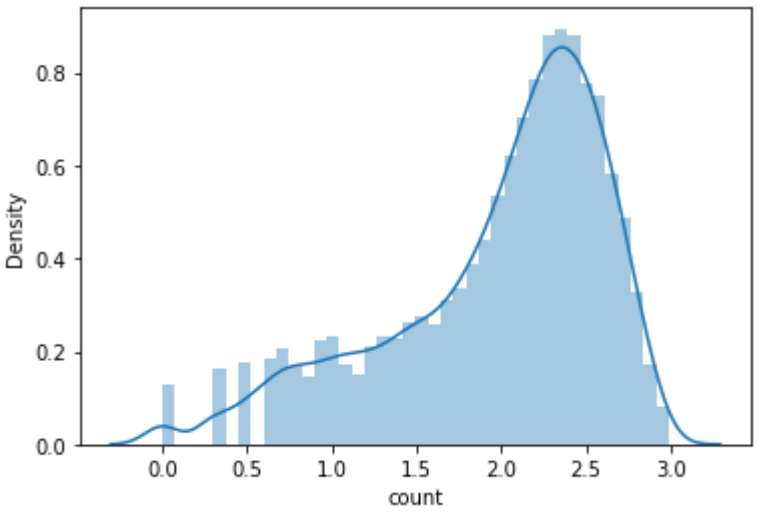
In [28]:

```
sample = np.log10(yulu['count'])
sns.distplot(sample)
```

```
C:\Users\sanke\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[28]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x219441d6208>
```



As we have seen before, the distribution of 'count' does not follow a normal distribution. Hence we have tried to transform it into a log normal distribution. Since even after the transformation the distribution still is not normal. hence we are going ahead with the hypothesis testing. Since ANNOVA is a robust test, the difference in distribution will not affect the efficiency of the test

To test the above we can use the ANNOVA test since we are comparing 4 different types of seasons against the count of rented vehicles for them. Since this is a numerical vs categorical testing, ANNOVA is the right fit

Hypothesis formulation

Setting up the null and alternate hypothesis
H0 : Seasons do not have an impact on the No. of cycles rented

Ha : No. of cycles rented is dependent on seasons and the no. varies with change in seasons

To test the above we can use the ANNOVA test since we are comparing 4 different types of seasons against the count of rented vehicles for them. Since this is a numerical vs categorical testing, ANNOVA is the right fit

In [29]:

```
season_1 = yulu[yulu["season"]==1]["count"]
season_2 = yulu[yulu["season"]==2]["count"]
season_3 = yulu[yulu["season"]==3]["count"]
season_4 = yulu[yulu["season"]==4]["count"]
```

In [30]:

```
f_stat, p_value = f_oneway(season_1,season_2,season_3,season_4)
alpha = 0.05 # testing the null hypothesis at 95% confidence level
print("f_stat : ",f_stat)
print("p-value : ",p_value)
if p_value < alpha:
    print("Reject H0")
else:
    print("Fail to reject H0")
```

```
f_stat :  236.94671081032106
p-value :  6.164843386499654e-149
Reject H0
```

Since the p-value is very very low i.e 6.164843386499654e-149 from the above test we are going to Reject the null hypothesis.
Hence we can very clearly conclude that the no. of electric vehicles rented is different for different seasons

Inference from the analysis
We can very clearly conclude that the no. of electric vehicles rented is different for different seasons

Testing if weather is dependent on the season

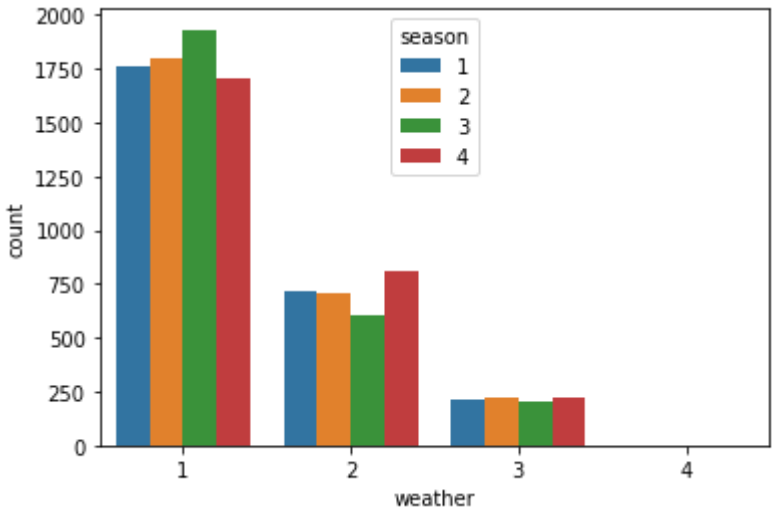
Visual analysis

In [31]:

```
sns.countplot(x='weather',data=yulu,hue='season')
```

Out[31]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21944440630>
```



From the above plots we can seem to infer that differrent seasons have different effect on diffrent weather types. We can test this inference further using hypothesis testing

Hypothesis formulation

Setting up the null and alternate hypothesis
H0 : Weather is independent of seasons
Ha : Seasons effect weather

Since we are testing categorical vs categorical variable, we can use CHI-SQUARE test for this

In [32]:

```
season_weather = pd.crosstab(index=yulu['season'],columns=yulu['weather'])
season_weather
```

Out[32]:

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

In [33]:

```
chi_stat, p_value, dof, expected = chi2_contingency(season_weather)
alpha = 0.05 # testing the null hypothesis at 95% confidence level
print("chi_stat:",chi_stat)
print("p-value:",p_value)
if p_value < alpha:
    print("Reject H0")
else:
    print("Fail to reject H0")
```

```
chi_stat: 49.158655596893624
p-value: 1.549925073686492e-07
Reject H0
```

```
Since the p-value is very low i.e 1.549925073686492e-07 from the above test we are going
to Reject the null hypothesis.
Hence we can very clearly conclude that the weather is dependent on the seasons
```

```
# Inference from the analysis
We can very clearly conclude that the weather is dependent on the seasons
```

In [ ]:

```

```