Business Case: Yulu - Hypothesis Testing

In [1]:

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

In [2]:

```
#Data extract
DF=pd.read_csv("C:/Users/srinj/Downloads/Business Case Yulu - Hypothesis Testing/bike_sharing.csv")
```

In [3]:

```
DF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 12 columns):
datetime      10886 non-null object
season        10886 non-null int64
holiday       10886 non-null int64
workingday    10886 non-null int64
weather       10886 non-null int64
temp          10886 non-null float64
atemp         10886 non-null float64
humidity      10886 non-null int64
windspeed     10886 non-null float64
casual        10886 non-null int64
registered    10886 non-null int64
count         10886 non-null int64
dtypes: float64(3), int64(8), object(1)
memory usage: 1020.6+ KB
```

In [4]:

```
#checking missing value
DF.isnull().sum()
#result: No missing value
```

Out[4]:

```
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 [5]:

```
DF.shape
```

Out[5]:

```
(10886, 12)
```

In [6]:

```
import plotly.express as px
import plotly.graph_objects as go
```

In [7]:

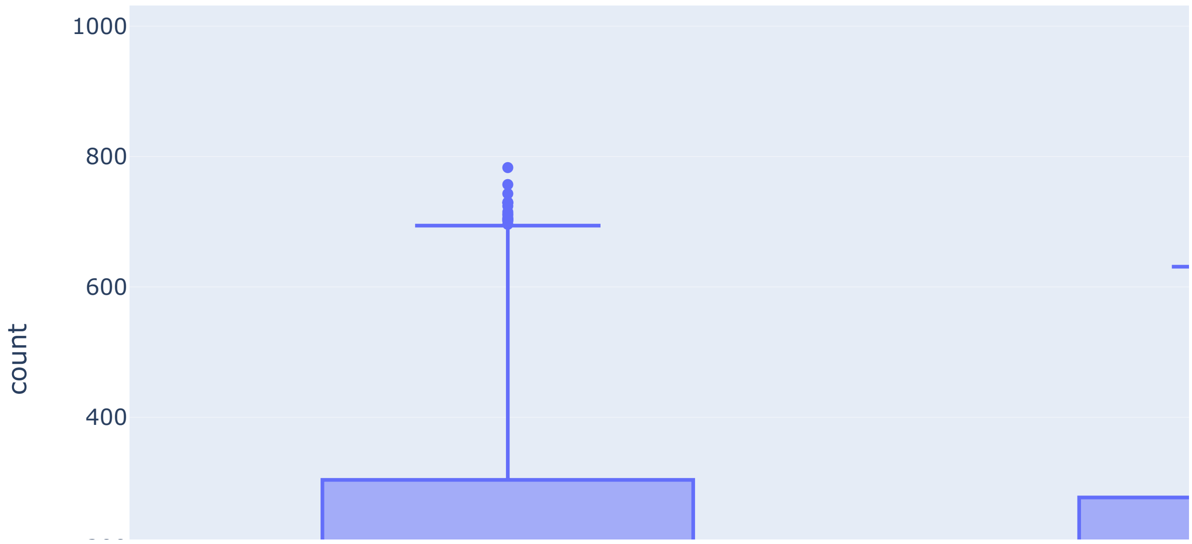
```
DF.workingday.unique()
```

Out[7]:

```
array([0, 1], dtype=int64)
```

In [8]:

```
px.box(DF,x='workingday',y='count')
```



In [9]:

```
DF.season.unique()
```

Out[9]:

```
array([1, 2, 3, 4], dtype=int64)
```

In [10]:

```
DF.weather.unique()
```

Out[10]:

```
array([1, 2, 3, 4], dtype=int64)
```

In [11]:

```
DF.holiday.unique()
```

Out[11]:

```
array([0, 1], dtype=int64)
```

In [12]:

```
DF = DF.astype({"season":'category',"holiday":'category',"workingday":'category',"weather":'categor
```

In [13]:

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

In [14]:

```
DF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 12 columns):
datetime      10886 non-null datetime64[ns]
season        10886 non-null category
holiday       10886 non-null category
workingday    10886 non-null category
weather       10886 non-null category
temp          10886 non-null float64
atemp         10886 non-null float64
humidity      10886 non-null int64
windspeed     10886 non-null float64
casual        10886 non-null int64
registered    10886 non-null int64
count         10886 non-null int64
dtypes: category(4), datetime64[ns](1), float64(3), int64(4)
memory usage: 723.5 KB
```

In [15]:

```
DF.max()
```

Out[15]:

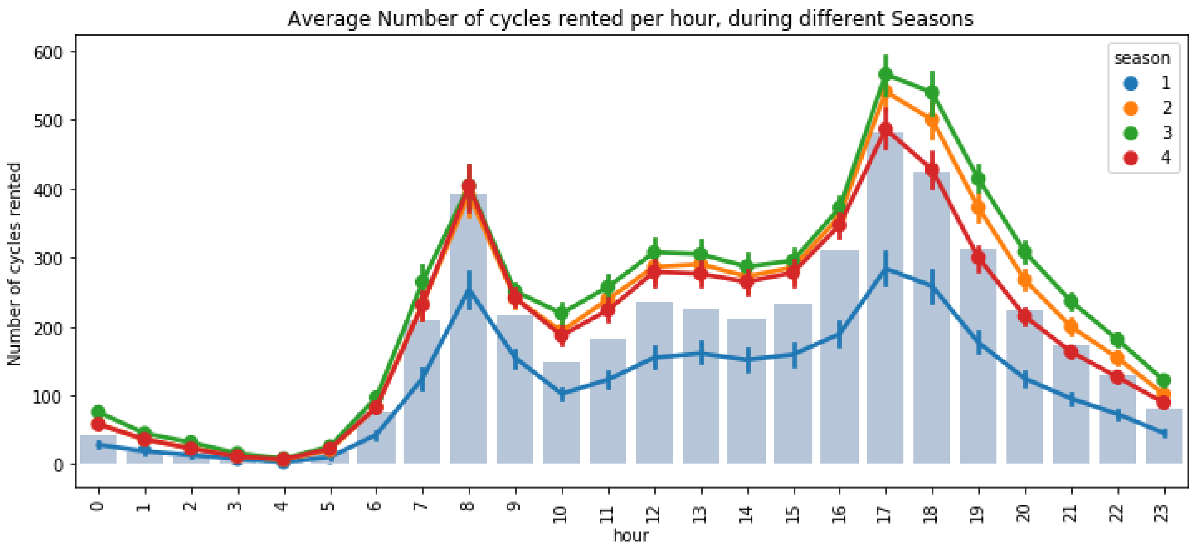
```
datetime      2012-12-19 23:00:00
season                          4
holiday                         1
workingday                      1
weather                         4
temp                           41
atemp                      45.455
humidity                      100
windspeed                 56.9969
casual                        367
registered                    886
count                         977
dtype: object
```

In [16]:

```
DF['hour']=DF['datetime'].dt.hour
```

In [17]:

```
plt.figure(figsize=(12,5))
sns.barplot(y = DF.groupby("hour")["count"].median(),
           x = DF.groupby("hour")["count"].median().index,
            color="lightsteelblue")
sns.pointplot(x = DF["hour"],
              y= DF["count"],
              hue=DF["season"],
              ci=85)
plt.title("Average Number of cycles rented per hour, during different Seasons")
plt.xticks(rotation = 90)
plt.ylabel("Number of cycles rented")
plt.show()
```



between 7-9am and 4pm to 7pm , the cycles rent counts is increasing as that is office hours

In [18]:

```
season_wise_cycle_rent_percentage = DF.groupby("season")["count"].sum()/np.sum(DF["count"])*100
```

In [19]:

```
season_wise_cycle_rent_percentage
```

Out[19]:

```
season
1    14.984493
2    28.208524
3    30.720181
4    26.086802
Name: count, dtype: float64
```

In the spring season , people rent less cycle

In [20]:

```
weather_wise_cycle_rent_percentage = DF.groupby("weather")["count"].sum()/np.sum(DF["count"])*100
```

In [21]:

```
weather_wise_cycle_rent_percentage
```

Out[21]:

```
weather
1    70.778230
2    24.318669
3     4.895237
4     0.007864
Name: count, dtype: float64
```

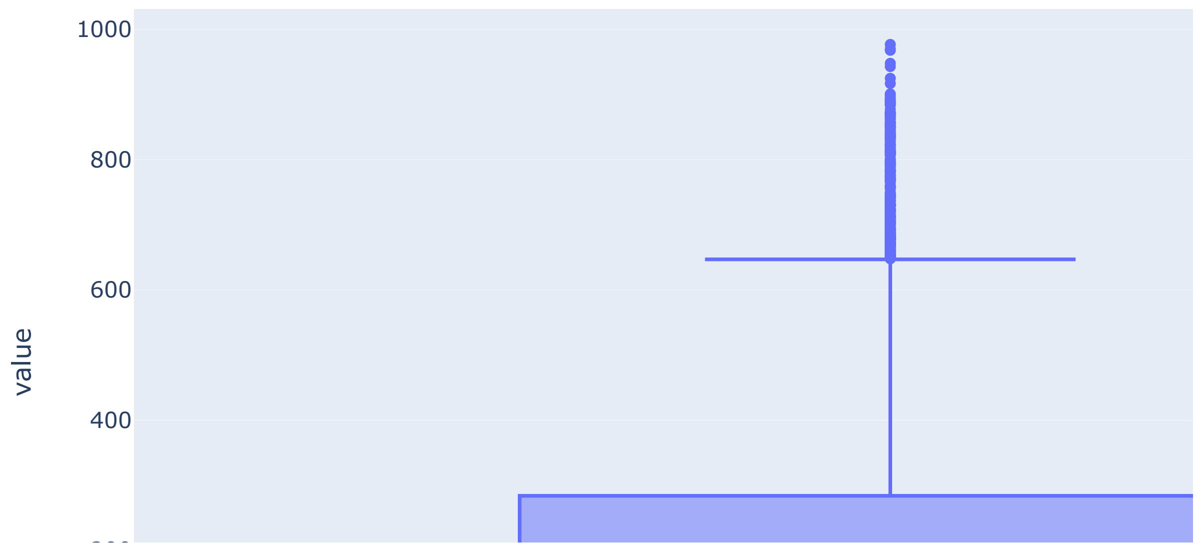
Cycles are mostly rented in Clear weather

In [22]:

```
import plotly.express as px
import plotly.graph_objects as go
```

In [23]:

```
px.box(DF['count'])
```



In [24]:

```
DF_with_outlier_count = DF.query('count > 647')
```

In [25]:

```
DF_with_outlier_count.head(5)
```

Out[25]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registe
6611	2012-03-12 18:00:00	1	0	1	2	24.60	31.06	43	12.9980	89	
6634	2012-03-13 17:00:00	1	0	1	1	28.70	31.82	37	7.0015	62	
6635	2012-03-13 18:00:00	1	0	1	1	28.70	31.82	34	19.9995	96	
6649	2012-03-14 08:00:00	1	0	1	1	18.04	21.97	82	0.0000	34	
6658	2012-03-14 17:00:00	1	0	1	1	28.70	31.82	28	6.0032	140	

In [26]:

```
DF_with_outlier_count.season.unique()
```

Out[26]:

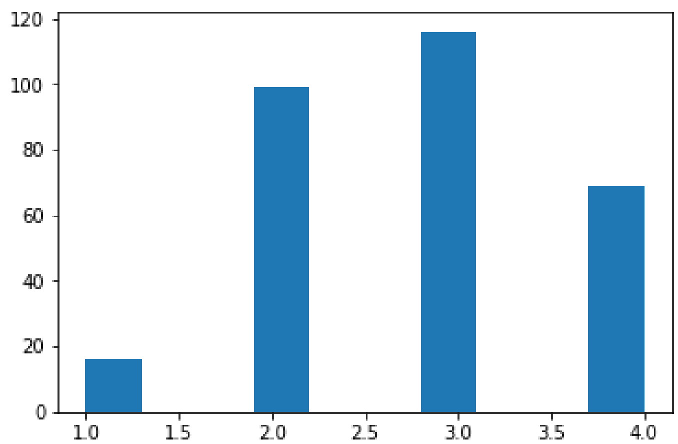
```
[1, 2, 3, 4]
Categories (4, int64): [1, 2, 3, 4]
```

In [27]:

```
plt.hist(DF_with_outlier_count.season)
```

Out[27]:

```
(array([ 16.,   0.,   0.,  99.,   0.,   0., 116.,   0.,   0.,  69.]),
 array([1. , 1.3, 1.6, 1.9, 2.2, 2.5, 2.8, 3.1, 3.4, 3.7, 4. ]),
 <a list of 10 Patch objects>)
```

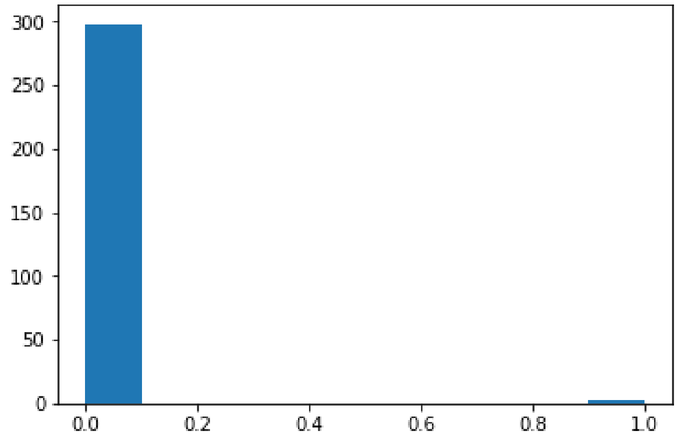


In [28]:

```
plt.hist(DF_with_outlier_count.holiday)
#extreme bike counts happened on non holiday mostly
```

Out[28]:

```
(array([298.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   2.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```

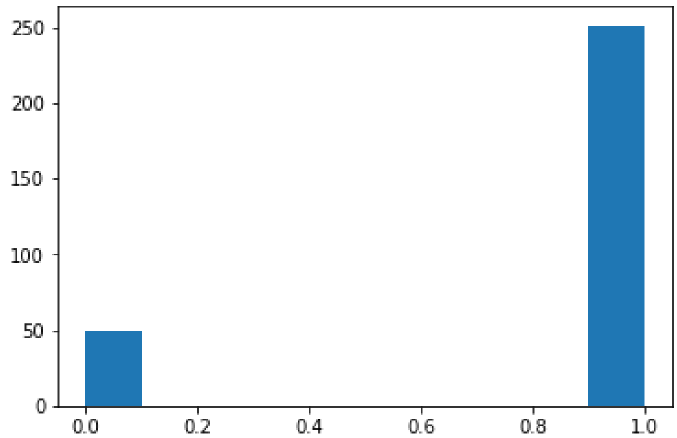


In [29]:

```
plt.hist(DF_with_outlier_count.workingday)
#extreme bike counts happened on working day mostly
```

Out[29]:

```
(array([ 49.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0., 251.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```

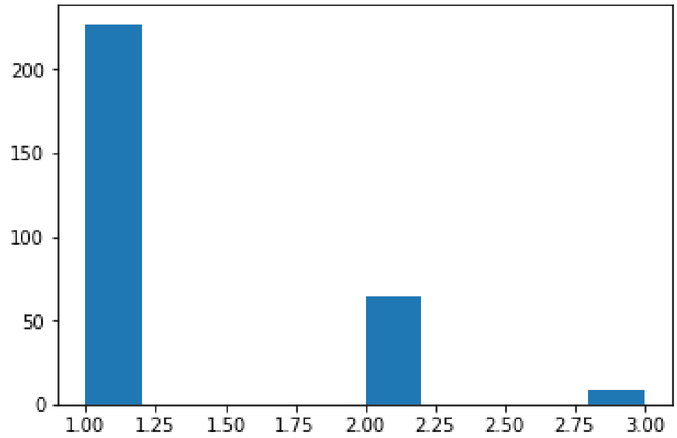


In [30]:

```
plt.hist(DF_with_outlier_count.weather)
#Clear, Few clouds, partly cloudy, partly cloudy has most extreme bike rents
```

Out[30]:

```
(array([227.,   0.,   0.,   0.,   0.,  64.,   0.,   0.,   0.,   9.]),
 array([1. , 1.2, 1.4, 1.6, 1.8, 2. , 2.2, 2.4, 2.6, 2.8, 3. ]),
 <a list of 10 Patch objects>)
```



In [31]:

```
DF_with_outlier_count.shape
```

Out[31]:

```
(300, 13)
```

In [32]:

```
DF.shape
```

Out[32]:

```
(10886, 13)
```

In [33]:

```
#Since we have very few outliers compared to total data, we will remove the outliers by IQR method
```

In [34]:

```
Q1 = DF['count'].quantile(0.25)
Q3 = DF['count'].quantile(0.75)
IQR = Q3 - Q1
ub = Q3 + (1.5*IQR)
lb = Q1 - (1.5*IQR)
```

In [35]:

```
ub
```

Out[35]:

```
647.0
```

In [36]:

```
lb
```

Out[36]:

```
-321.0
```

In [37]:

```
DF=DF[(DF['count']>lb) & (DF['count']<ub)]
```

In [38]:

```
DF.shape
```

Out[38]:

```
(10583, 13)
```

In [39]:

```
#Correlation between the dependent and independent variable (Dependent “Count” & Independent: Workin
DF.corr()
```

Out[39]:

	temp	atemp	humidity	windspeed	casual	registered	count	hour
temp	1.000000	0.985885	-0.050958	-0.022109	0.468881	0.304261	0.387816	0.133799
atemp	0.985885	1.000000	-0.030118	-0.062602	0.463878	0.301943	0.384432	0.129143
humidity	-0.050958	-0.030118	1.000000	-0.319592	-0.335204	-0.273894	-0.323054	-0.270702
windspeed	-0.022109	-0.062602	-0.319592	1.000000	0.088060	0.102536	0.109715	0.145105
casual	0.468881	0.463878	-0.335204	0.088060	1.000000	0.512966	0.716661	0.302234
registered	0.304261	0.301943	-0.273894	0.102536	0.512966	1.000000	0.966296	0.412975
count	0.387816	0.384432	-0.323054	0.109715	0.716661	0.966296	1.000000	0.426164
hour	0.133799	0.129143	-0.270702	0.145105	0.302234	0.412975	0.426164	1.000000

In [40]:

```
#extract hour mark from date to find the demand of cycle from different day time
DF['hour']=DF['datetime'].dt.hour
```

In [41]:

```
DF['weather'].replace({1:"Clear",2:"Misty/Cloudy",3:"Light snow/rain", 4:"Heavy Rain"},inplace=True
```

In [42]:

```
DF["season"].replace({1:"spring", 2:"summer", 3:"fall", 4:"winter"}, inplace=True)
```

In [43]:

```
DF['workingday'].replace({1:"Yes",0:"No"}, inplace=True)
```

In [44]:

```
DF['holiday'].replace({1:"Yes", 0:"No"},inplace = True)
```

In [45]:

```
DF['month name']=DF["datetime"].dt.month_name()
```

In [46]:

```
DF.head()
```

Out[46]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered
0	2011-01-01 00:00:00	spring	No	No	Clear	9.84	14.395	81	0.0	3	13
1	2011-01-01 01:00:00	spring	No	No	Clear	9.02	13.635	80	0.0	8	32
2	2011-01-01 02:00:00	spring	No	No	Clear	9.02	13.635	80	0.0	5	27
3	2011-01-01 03:00:00	spring	No	No	Clear	9.84	14.395	75	0.0	3	10
4	2011-01-01 04:00:00	spring	No	No	Clear	9.84	14.395	75	0.0	0	1

In [47]:

```
DF.corr()
```

Out[47]:

	temp	atemp	humidity	windspeed	casual	registered	count	hour
temp	1.000000	0.985885	-0.050958	-0.022109	0.468881	0.304261	0.387816	0.133799
atemp	0.985885	1.000000	-0.030118	-0.062602	0.463878	0.301943	0.384432	0.129143
humidity	-0.050958	-0.030118	1.000000	-0.319592	-0.335204	-0.273894	-0.323054	-0.270702
windspeed	-0.022109	-0.062602	-0.319592	1.000000	0.088060	0.102536	0.109715	0.145105
casual	0.468881	0.463878	-0.335204	0.088060	1.000000	0.512966	0.716661	0.302234
registered	0.304261	0.301943	-0.273894	0.102536	0.512966	1.000000	0.966296	0.412975
count	0.387816	0.384432	-0.323054	0.109715	0.716661	0.966296	1.000000	0.426164
hour	0.133799	0.129143	-0.270702	0.145105	0.302234	0.412975	0.426164	1.000000

Correlation between Temperature and Number of Cycles Rented for all customers : 0.39

Correlation between Temperature and Number of Cycles Rented for casual subscribers : 0.46

Correlation between Temperature and Number of Cycles Rented for registered subscribers : 0.30

Humidity has a negative correlation with the number of cycles rented which is -0.32

windspeed has very low correlation around 0.1 with number of bikes rented

In [48]:

```
DF.sum()
```

Out[48]:

```
temp          2.122890e+05
atemp         2.484259e+05
humidity      6.579470e+05
windspeed     1.351981e+05
casual        3.588910e+05
registered    1.499309e+06
count         1.858200e+06
hour          1.212410e+05
dtype: float64
```

In [49]:

```
(DF["casual"].sum()/DF["count"].sum())*100
```

Out[49]:

```
19.313905930470348
```

In [50]:

```
(DF["registered"].sum()/DF["count"].sum())*100
```

Out[50]:

```
80.68609406952966
```

Hypothesis Testing

check if Working Day has an effect on the number of electric cycles rented

Null Hypothesis (H0) = Working day does not have any effect on number of rented cycles

Alternative Hypothesis (H1) = Working day does have an effect on number of rented cycles

significance level = 0.05

In [51]:

```python
DF_with_workingday = DF[DF['workingday']=="Yes"]
DF_with_workingday.head(5)
```

Out[51]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered
47	2011-01-03 00:00:00	spring	No	Yes	Clear	9.02	9.850	44	23.9994	0	5
48	2011-01-03 01:00:00	spring	No	Yes	Clear	8.20	8.335	44	27.9993	0	2
49	2011-01-03 04:00:00	spring	No	Yes	Clear	6.56	6.820	47	26.0027	0	1
50	2011-01-03 05:00:00	spring	No	Yes	Clear	6.56	6.820	47	19.0012	0	3
51	2011-01-03 06:00:00	spring	No	Yes	Clear	5.74	5.305	50	26.0027	0	30

In [52]:

```python
DF_without_workingday = DF[DF['workingday']=="No"]
DF_without_workingday.head(5)
```

Out[52]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered
0	2011-01-01 00:00:00	spring	No	No	Clear	9.84	14.395	81	0.0	3	13
1	2011-01-01 01:00:00	spring	No	No	Clear	9.02	13.635	80	0.0	8	32
2	2011-01-01 02:00:00	spring	No	No	Clear	9.02	13.635	80	0.0	5	27
3	2011-01-01 03:00:00	spring	No	No	Clear	9.84	14.395	75	0.0	3	10
4	2011-01-01 04:00:00	spring	No	No	Clear	9.84	14.395	75	0.0	0	1

In [53]:

```python
import scipy.stats as stats
```

In [54]:

```
stats.ttest_ind(a=DF_with_workingday['count'],b=DF_without_workingday['count'], equal_var=True)
```

Out[54]:

```
Ttest_indResult(statistic=-2.4512041726795246, pvalue=0.014253976221734492)
```

We reject the null hypothesis as the p-value (0.01) is less than the significance level (0.05)

In [ ]:

```

```

check if No. of cycles rented is similar or different in different 1. weather 2. season

Null Hypothesis (H0) = Weather does not have any effect on number of rented cycles

Alternative Hypothesis (H1) = Weather does have an effect on number of rented cycles

significance level = 0.05

In [55]:

```
Clear = DF.loc[DF["weather"]=="Clear"]["count"]
Cloudy = DF.loc[DF["weather"]=="Misty/Cloudy"]["count"]
Little_Rain = DF.loc[DF["weather"]=="Light snow/rain"]["count"]
Heavy_Rain = DF.loc[DF["weather"]=="Heavy Rain"]["count"]
```

In [56]:

```
len(Clear),len(Cloudy),len(Little_Rain),len(Heavy_Rain)
```

Out[56]:

```
(6962, 2770, 850, 1)
```

In [57]:

```
Heavy_Rain
```

Out[57]:

```
5631    164
Name: count, dtype: int64
```

We will exclude heavy rain for having just one record

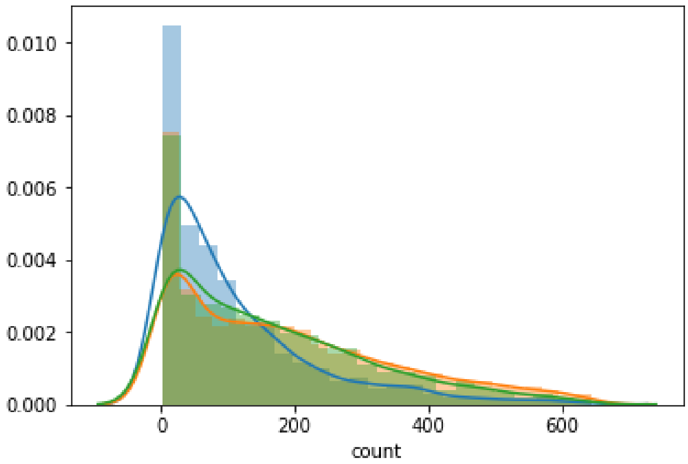
In [58]:

```
sns.distplot((Little_Rain))
sns.distplot((Clear))
sns.distplot((Cloudy))
```

```
C:\Users\srinj\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarnin
g:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

C:\Users\srinj\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarnin
g:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

C:\Users\srinj\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarnin
g:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
```

Out[58]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x289c407c4a8>
```



In [59]:

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

In [60]:

```
from scipy.stats import boxcox
```

In [81]:

```
from sklearn import preprocessing
```

In [84]:

```
Clear.mean()
```

Out[84]:

```
187.13114047687446
```

In [85]:

```
Clear.std()
```

Out[85]:

```
161.3337854491698
```

In [86]:

```
Clear=(Clear-Clear.mean())/Clear.std()
```

In [88]:

```
Cloudy=(Cloudy-Cloudy.mean())/Cloudy.std()
```

In [89]:

```
Little_Rain=(Little_Rain-Little_Rain.mean())/Little_Rain.std()
```

Data Normalization done by setting Z-Score

In [64]:

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

In [90]:

```
f_oneway(Clear, Cloudy, Little_Rain)
```

Out[90]:

```
F_onewayResult(statistic=1.010746972546296e-29, pvalue=1.0)
```

Here p-value is significantly greater than the level of significance, So we are unable to reject null hypothesis and conclude that weather does not have significant effect on rented cycle

checking the effect of season on cycle renting

In [ ]:

```
Null Hypothesis (H0) = Season does not have any effect on number of rented cycles

Alternative Hypothesis (H1) = Season does have an effect on number of rented cycles

significance level = 0.05
```

In [66]:

```
DF.head(5)
```

Out[66]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered
0	2011-01-01 00:00:00	spring	No	No	Clear	9.84	14.395	81	0.0	3	13
1	2011-01-01 01:00:00	spring	No	No	Clear	9.02	13.635	80	0.0	8	32
2	2011-01-01 02:00:00	spring	No	No	Clear	9.02	13.635	80	0.0	5	27
3	2011-01-01 03:00:00	spring	No	No	Clear	9.84	14.395	75	0.0	3	10
4	2011-01-01 04:00:00	spring	No	No	Clear	9.84	14.395	75	0.0	0	1

In [68]:

```
spring = DF.loc[DF["season"]=="spring"]["count"]
summer = DF.loc[DF["season"]=="summer"]["count"]
fall = DF.loc[DF["season"]=="fall"]["count"]
winter = DF.loc[DF["season"]=="winter"]["count"]
```

In [69]:

```
len(spring),len(summer),len(fall),len(winter)
```

Out[69]:

```
(2670, 2633, 2616, 2664)
```

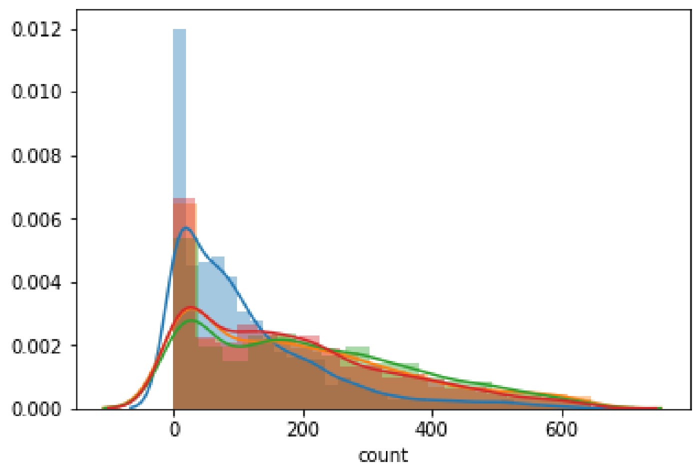
In [70]:

```
sns.distplot((spring))
sns.distplot((summer))
sns.distplot((fall))
sns.distplot((winter))
```

```
C:\Users\srinj\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarnin
g:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

C:\Users\srinj\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarnin
g:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

C:\Users\srinj\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarnin
g:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

C:\Users\srinj\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarnin
g:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
```

Out[70]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x289c5f9df98>
```



In [91]:

```
spring=(spring-spring.mean())/spring.std()
summer=(summer-summer.mean())/summer.std()
fall=(fall-fall.mean())/fall.std()
winter=(winter-winter.mean())/winter.std()
```

Data Normalization done by Z-score

In [92]:

```
f_oneway(spring, summer, fall,winter)
```

Out[92]:

```
F_onewayResult(statistic=5.064390999801078e-31, pvalue=1.0)
```

Here p-value is significantly greater than the level of significance, So we are unable to reject null hypothesis and conclude that weather does not have significant effect on rented cycle

In [ ]:

```

```

Chi-square test to check if Weather is dependent on the season

Null Hypothesis (H0) = Weather is not dependent on season

Alternative Hypothesis (H1) = Weather is dependent on season

significance level = 0.05

In [ ]:

```

```

In [94]:

```
DF.head(10)
```

Out[94]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	regist
0	2011-01-01 00:00:00	spring	No	No	Clear	9.84	14.395	81	0.0000	3	
1	2011-01-01 01:00:00	spring	No	No	Clear	9.02	13.635	80	0.0000	8	
2	2011-01-01 02:00:00	spring	No	No	Clear	9.02	13.635	80	0.0000	5	
3	2011-01-01 03:00:00	spring	No	No	Clear	9.84	14.395	75	0.0000	3	
4	2011-01-01 04:00:00	spring	No	No	Clear	9.84	14.395	75	0.0000	0	
5	2011-01-01 05:00:00	spring	No	No	Misty/Cloudy	9.84	12.880	75	6.0032	0	
6	2011-01-01 06:00:00	spring	No	No	Clear	9.02	13.635	80	0.0000	2	
7	2011-01-01 07:00:00	spring	No	No	Clear	8.20	12.880	86	0.0000	1	
8	2011-01-01 08:00:00	spring	No	No	Clear	9.84	14.395	75	0.0000	1	
9	2011-01-01 09:00:00	spring	No	No	Clear	13.12	17.425	76	0.0000	8	

In [100]:

```
from scipy.stats import chi2
```

In [101]:

```
from scipy.stats import chi2_contingency
```

In [102]:

```
contingency_tab=pd.crosstab(DF["weather"],DF["season"])
```

In [103]:

```
contingency_tab.head(5)
```

Out[103]:

season weather	fall	spring	summer	winter
Clear	1842	1744	1720	1656
Heavy Rain	0	1	0	0
Light snow/rain	195	211	223	221
Misty/Cloudy	579	714	690	787

In [104]:

```
chi2_stat, p_value, dof, expected = stats.chi2_contingency(contingency_tab)

print(chi2_stat)
print(p_value)
```

```
47.16590591959627
3.6550317439064896e-07
```

Here p-value is significantly less than the level of significance, So we reject null hypothesis and conclude that weather is dependent on season

Inference from the analysis

-There is a positive Correlation between Temperature and Number of cycles rented.

-Demand increases with the rise in the temperature from modate to not very high.

-between 7-9am and 4pm to 7pm , the cycles rent counts is increasing as that is office hours

-In the spring season , people rent less cycle

-registered customers are much higher than the casual customers. 81% customers are Registered and 19% only are casual riders

-Cycles are mostly rented in Clear weather

-As per hourly average number of cycles rented by registered and casual customer plots ,

-Registered Customers seems to be using rental cycles mostly for work commute purposes.

-demand on weekdays and off-days are similar

Conclusion from statistical tests

-Working day does have an effect on number of rented cycles

-Weather and season does not have any prominent effect on number of rented cycles

-weather and seasons are dependent.

```
In [ ]:
```