In [1]:
```
import rpy2.rinterface

import os
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib import pyplot
import statsmodels as sm
import statsmodels.formula.api as smf
from scipy.stats import t as tdist
from scipy import interpolate
from statsmodels.stats.outliers_influence import summary_table
```

In [58]:
```
%reload_ext rpy2.ipython
```

In [59]:
```
#Current working directory in Python
os.getcwd()
```

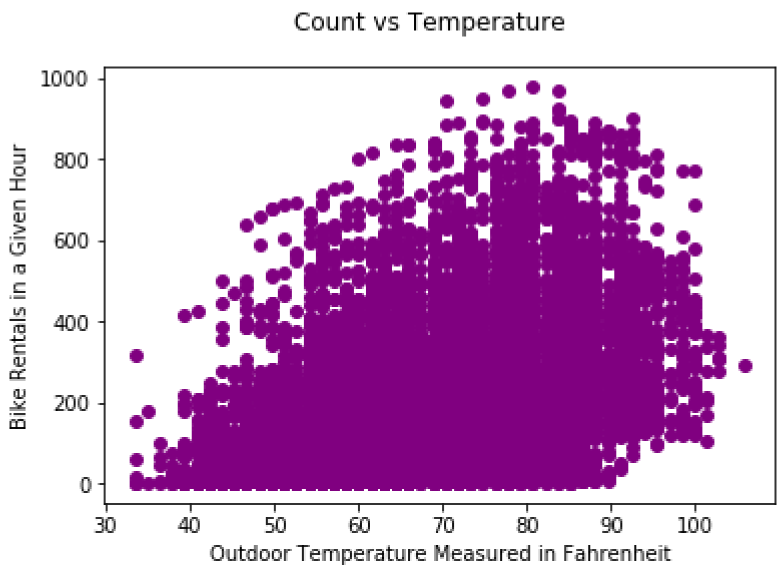
Out[59]:
```
'C:\\Users\\Erin Canada\\Documents\\USF\\Fall 2018\\Regression\\Lab'
```

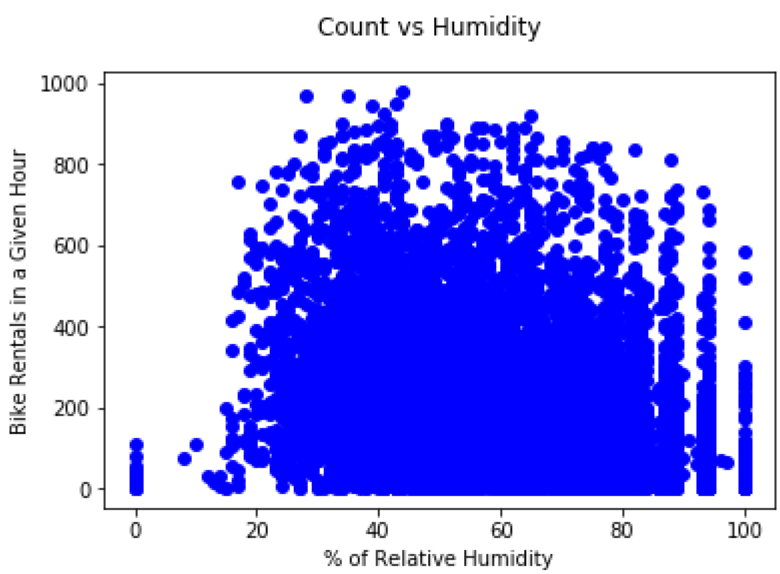
In [61]:
```
#Set working directory in Python
os.chdir('C:\\Users\\Erin Canada\\Documents\\USF\\Fall 2018\\Regression\\Lab')
```

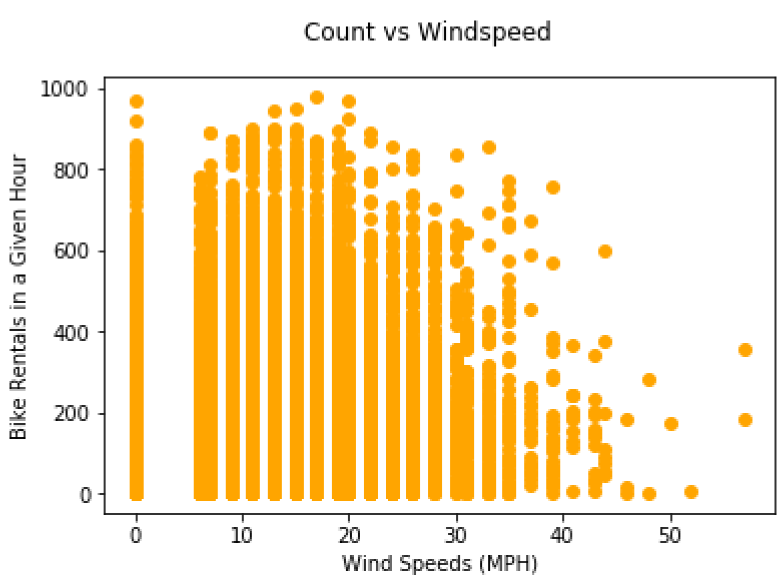
In [5]:
```
##Reading in relevant file in Python
bike = pd.read_csv('bike_share.csv')

## y = count, t = temp, h = humidity, w = windspeed
y = bike['count'] #the number of bike rentals in a given hourly period
t = bike['temp'] # outdoor temperature(measured in Fahrenheit)
h = bike['humidity'] #relative humidity(as a percentage)
w = bike['windspeed'] # wind speeds (miles per hour)
```

In [6]:
```
#Part A

##Constructing a scatterplot of these data in Python

#Count VS Temp
temp_fig = plt.figure()
plt.scatter(t,y, c = 'purple')
temp_fig.suptitle('Count vs Temperature')
plt.ylabel('Bike Rentals in a Given Hour')
plt.xlabel('Outdoor Temperature Measured in Fahrenheit')

#Count VS Humidity
humid_fig = plt.figure()
plt.scatter(h,y, c = 'blue')
humid_fig.suptitle('Count vs Humidity')
plt.ylabel('Bike Rentals in a Given Hour')
plt.xlabel('% of Relative Humidity')

#Count vs Windspeed
#Count VS Temp
wind_fig = plt.figure()
plt.scatter(w,y, c = 'orange')
wind_fig.suptitle('Count vs Windspeed')
plt.ylabel('Bike Rentals in a Given Hour')
plt.xlabel('Wind Speeds (MPH)')
```

```
Out[6]: Text(0.5,0,'Wind Speeds (MPH)')
```







In [7]:
```
## Calculation of the Correlation Coefficient in Python

#Count vs Temperature
temp_co = np.corrcoef(t,y)[0,1]
print temp_co

#Count vs Humidity
humid_co = np.corrcoef(h,y)[0,1]
print humid_co

#Count vs Wind Speed
wind_co = np.corrcoef(w,y)[0,1]
print wind_co
```

```
0.3944536449672491
-0.31737147887659445
0.10136947021033277
```

In [8]:
```
##Describe the Linear Relationship (Part A continued)

'''
Count vs Temperature have a strong positive relationship as seen by the correl
ation
coefficients.  It is moderately close to 1 and is in a positive direction.

Count vs Humidity have a strong negative relationship where the correlation
coefficients are again moderately close to 1 but in a negative direction.

Count vs Wind Speed have a weak postive relationship. The correlation
coefficients are very close to zero but in a positive direction.
```

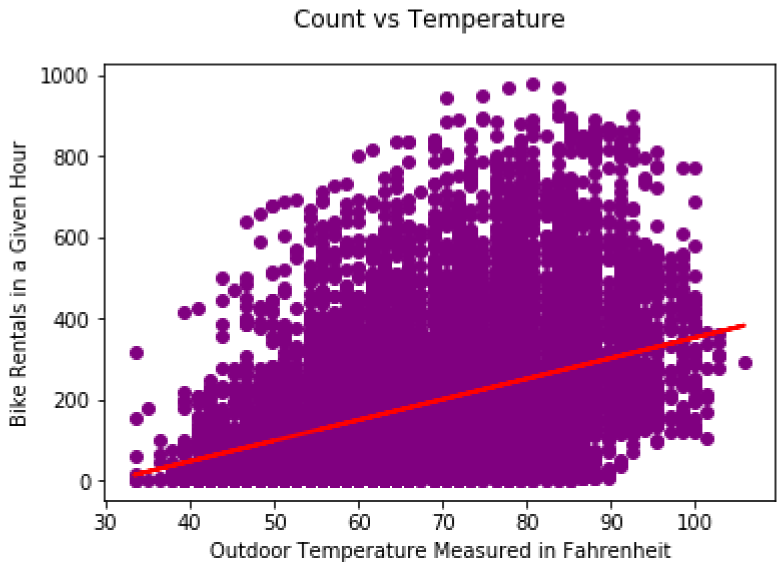
In [9]:
```
#Part B

#Calculate Beta Hat 0 and Beta Hat 1 in Python

#Count vs Temp
beta1_hat_temp = np.corrcoef(t,y)[0,1] * np.std(y) / np.std(t)
print 'Beta Hat 1--Temp'
print beta1_hat_temp
print

beta0_hat_temp = np.mean(y) - beta1_hat_temp * np.mean(t)
print 'Beta Hat 0 --Temp'
print beta0_hat_temp
print


#Checking answers
lm_t = smf.ols('y~t', data = bike)
model_t = lm_t.fit()
print model_t.summary()

#Count vs Humidity
beta1_hat_humid = np.corrcoef(h,y)[0,1] * np.std(y) / np.std(h)
print 'Beta Hat 1--Humidity'
print beta1_hat_humid
print

beta0_hat_humid = np.mean(y) - beta1_hat_humid * np.mean(h)
print 'Beta Hat 0 --Humidity'
print beta0_hat_humid
print


#Checking answers
lm_h = smf.ols('y~h', data = bike)
model_h = lm_h.fit()
print model_h.summary()


#Count vs Wind Speed
beta1_hat_wind = np.corrcoef(w,y)[0,1] * np.std(y) / np.std(w)
print 'Beta Hat 1--Wind Speed'
print beta1_hat_wind
print

beta0_hat_wind = np.mean(y) - beta1_hat_temp * np.mean(w)
print 'Beta Hat 0 --Wind Speed'
print beta0_hat_wind
print


#Checking answers
lm_w = smf.ols('y~w', data = bike)
model_w = lm_w.fit()
print model_w.summary()
```

```
Beta Hat 1--Temp
5.094744711903571

Beta Hat 0 --Temp
-156.98561782130787

                            OLS Regression Results                            
==============================================================================
=
Dep. Variable:                      y   R-squared:                       0.15
6
Model:                            OLS   Adj. R-squared:                  0.15
6
Method:                 Least Squares   F-statistic:                     200
6.
Date:                Tue, 11 Sep 2018   Prob (F-statistic):               0.0
0
Time:                        18:17:19   Log-Likelihood:                -7112
5.
No. Observations:               10886   AIC:                         1.423e+0
5
Df Residuals:                   10884   BIC:                         1.423e+0
5
Df Model:                           1                                         

Covariance Type:            nonrobust                                         

==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
Intercept   -156.9856      7.945    -19.759      0.000    -172.560    -141.41
2
t              5.0947      0.114     44.783      0.000       4.872       5.31
8
==============================================================================
=
Omnibus:                     1871.687   Durbin-Watson:                   0.36
9
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3221.96
6
Skew:                           1.123   Prob(JB):                         0.0
0
Kurtosis:                       4.434   Cond. No.                         34
8.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
Beta Hat 1--Humidity
-2.987268578534409
```

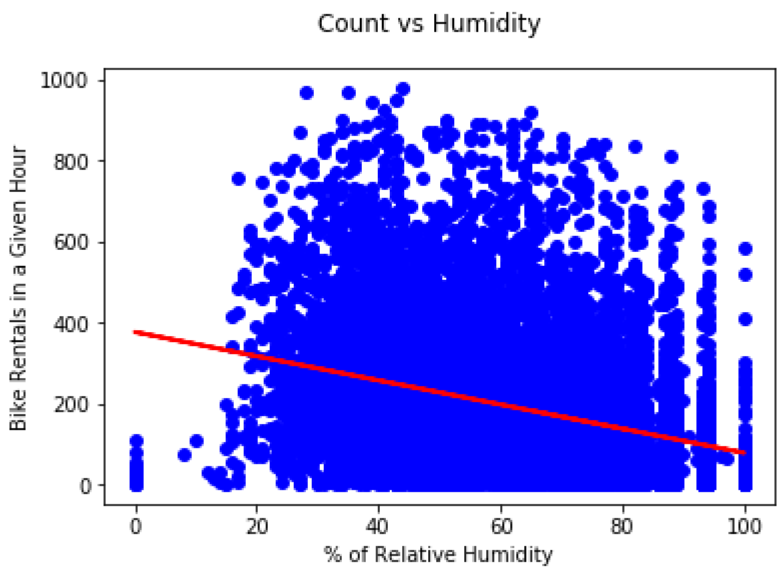
```
Beta Hat 0 --Humidity
376.44560833036167

                            OLS Regression Results
==============================================================================
=
Dep. Variable:                      y   R-squared:                       0.10
1
Model:                            OLS   Adj. R-squared:                  0.10
1
Method:                 Least Squares   F-statistic:                     121
9.
Date:                Tue, 11 Sep 2018   Prob (F-statistic):          2.92e-25
3
Time:                        18:17:19   Log-Likelihood:                -7146
8.
No. Observations:               10886   AIC:                         1.429e+0
5
Df Residuals:                   10884   BIC:                         1.430e+0
5
Df Model:                           1

Covariance Type:            nonrobust

==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
Intercept    376.4456      5.545     67.890      0.000     365.577     387.31
5
h             -2.9873      0.086    -34.915      0.000      -3.155      -2.82
0
==============================================================================
=
Omnibus:                     2068.515   Durbin-Watson:                   0.35
1
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3709.73
9
Skew:                           1.210   Prob(JB):                        0.0
0
Kurtosis:                       4.525   Cond. No.                        21
8.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
Beta Hat 1--Wind Speed
2.2490579173365712

Beta Hat 0 --Wind Speed
126.36447984745188
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                      y   R-squared:                       0.01
0
Model:                            OLS   Adj. R-squared:                  0.01
0
Method:                 Least Squares   F-statistic:                     113.
0
Date:                Tue, 11 Sep 2018   Prob (F-statistic):          2.90e-2
6
Time:                        18:17:19   Log-Likelihood:                -7198
9.
No. Observations:               10886   AIC:                         1.440e+0
5
Df Residuals:                   10884   BIC:                         1.440e+0
5
Df Model:                           1

Covariance Type:            nonrobust

==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
Intercept    162.7876      3.212     50.682      0.000     156.492     169.08
4
w              2.2491      0.212     10.630      0.000       1.834       2.66
4
==============================================================================
=
Omnibus:                     2086.612   Durbin-Watson:                   0.32
2
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3633.79
9
Skew:                           1.247   Prob(JB):                         0.0
0
Kurtosis:                       4.338   Cond. No.                         28.
3
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

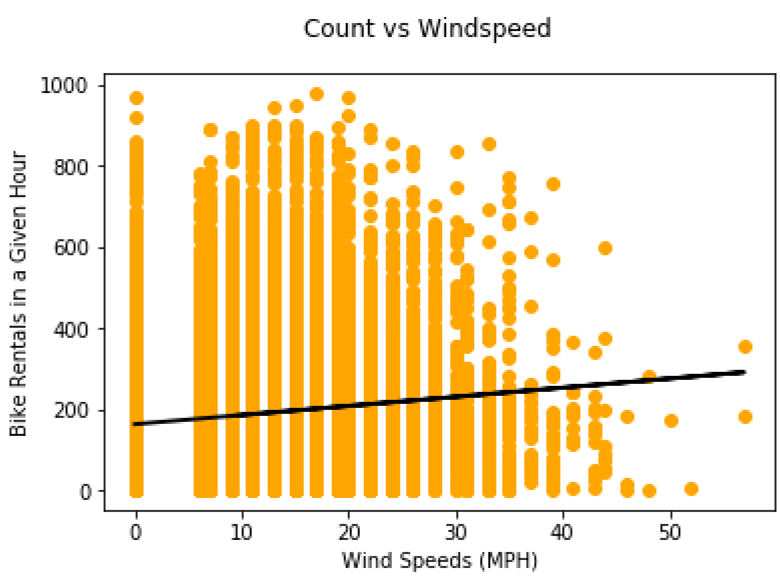
In [10]:
```
#Part C -- Add fitted regression lines

#Count VS Temp
temp_fig = plt.figure()
plt.scatter(t,y, c = 'purple')
temp_fig.suptitle('Count vs Temperature')
plt.ylabel('Bike Rentals in a Given Hour')
plt.xlabel('Outdoor Temperature Measured in Fahrenheit')
temp_fitted_line, = plt.plot(t, model_t.fittedvalues, '-', color = "red", line
width = 2, label = "Fitted Values")

#Count VS Humidity
humid_fig = plt.figure()
plt.scatter(h,y, c = 'blue')
humid_fig.suptitle('Count vs Humidity')
plt.ylabel('Bike Rentals in a Given Hour')
plt.xlabel('% of Relative Humidity')
humid_fitted_line, = plt.plot(h, model_h.fittedvalues, '-', color = "red", lin
ewidth = 2, label = "Fitted Values")

#Count vs Windspeed
wind_fig = plt.figure()
plt.scatter(w,y, c = 'orange')
wind_fig.suptitle('Count vs Windspeed')
plt.ylabel('Bike Rentals in a Given Hour')
plt.xlabel('Wind Speeds (MPH)')
wind_fitted_line, = plt.plot(w, model_w.fittedvalues, '-', color = "black", li
newidth = 2, label = "Fitted Values")
```



Count vs Temperature
Bike Rentals in a Given Hour
Outdoor Temperature Measured in Fahrenheit
0
200
400
600
800
1000
30
40
50
60
70
80
90
100



Count vs Humidity
Bike Rentals in a Given Hour
% of Relative Humidity
0
200
400
600
800
1000
0
20
40
60
80
100



Count vs Windspeed
Bike Rentals in a Given Hour
Wind Speeds (MPH)
0
200
400
600
800
1000
0
10
20
30
40
50

In [11]:
```
#Part D

'''
From the correlation coeffiecients to most weakly associated to most strongly
 associated:

Count vs Wind Speeds
0.10136947021033277

Count vs Humidity
-0.31737147887659445

Count vs Temperature
0.3944536449672491

Here we can see a pretty strong correlation between the number of bike rentals
 and the
temperature as wells as the humidity, but a  very weak correlation between the
 count and
wind speeds making it seem as though windspeed has no effect on the number
of bike rentals.
```

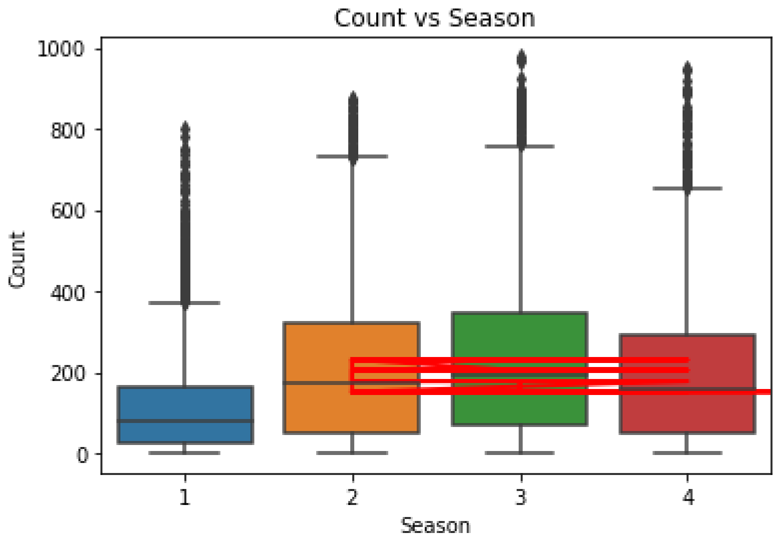
In [12]:
```
#Part E

## Temperature is 70 degrees
temp_70 = beta0_hat_temp + beta1_hat_temp*70
stt = "The expected number of bike rentals when the temperature is 70 degrees:
  "
print stt + '{:.2f}'.format(temp_70)

##Wind speed at 10 mph
wind_10 = beta0_hat_wind + beta1_hat_wind*10
stw = "The expected number of bike rentals when the wind speed is at 10 mph: "
print stw + '{:.2f}'.format(wind_10)

##Relative Humidity at 40%
humid_40 = beta0_hat_humid + beta1_hat_humid*40
sth = "The expected number of bike rentals when the humidity is 40%: "
print sth + '{:.2f}'.format(humid_40)
```

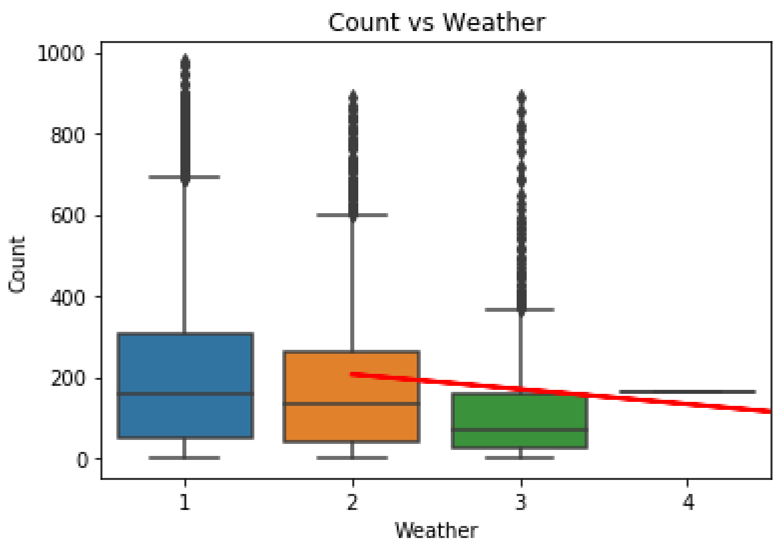
```
The expected number of bike rentals when the temperature is 70 degrees:  199.
65
The expected number of bike rentals when the wind speed is at 10 mph: 148.86
The expected number of bike rentals when the humidity is 40%: 256.95
```

In [ ]:
```
#Part F

'''
The risk when predicting the ouside the range of observed explanatory variable
s values
is that it may be assuming to much and be outside the scope of the prediction
 value.
It could produce very wrong or inconsistant results.

'''
```

In [43]:
```
#Part G & H

#Count vs Season boxplot
bplot = plt.figure()
cs = bike['season']
bplot_cs= sns.boxplot(cs,y,data = bike)
bplot_cs.axes.set_title("Count vs Season")
bplot_cs.set_xlabel("Season")
bplot_cs.set_ylabel("Count")
lm_cs = smf.ols('y~cs', data = bike)
model_cs = lm_cs.fit()
season_fitted_line, = plt.plot(cw, model_cs.fittedvalues, '-', color = "red",
linewidth = 2, label = "Fitted Values")

#Count vs Season correlation coefficient
count_season = np.corrcoef(cs,y)[0,1]
print count_season

#Count vs Weather boxplot
bplot_cw = plt.figure()
cw = bike['weather']
bplot_cw =sns.boxplot(cw,y,data=bike)
bplot_cw.axes.set_title("Count vs Weather")
bplot_cw.set_xlabel("Weather")
bplot_cw.set_ylabel("Count")
lm_cw = smf.ols('y~cw', data = bike)
model_cw = lm_cw.fit()
weather_fitted_line, = plt.plot(cw, model_cw.fittedvalues, '-', color = "red",
 linewidth = 2, label = "Fitted Values")

#Count vs Weather
count_weather = np.corrcoef(cw,y)[0,1]
print count_weather
```

```
0.16343901657636173
-0.1286552010385064
```





In [ ]:
```
##Part G and H continued and I

'''
From the boxplots, we can tell a little about the information. It seems
for season 3, that there are a higher average number of bike rentals, probably
 due
to that it is nicer outside meaning summer and summer vacation. If the season
is not so nice, there are less bike rentals which could be related to count vs
 weather.
Here there are higher bike rentals when there is the #1 type of weather(assumi
ng nice and sunny)
and lower to no number of bike rentals when the weather is like #4.

These interetations are not really useful because it just states the average f
or each
type of season or weather and can not really make any concrete predictions and

there is no real visual telling anything but just assumptions like I made abov
e.
As seen in the correlation coefficients, the relationships are very weak for b
oth boxplots and
there is no real evidence to continue with this data.

The linear regression in h is inappropriate because it establishes no relation
ship between the season variables but tells
us a singular patter between one season and count. The linear regression does
 not really exit because there
is a lack of relationship between the seasons in response to the count with a
 boxplot.
Perhaps a scatterplot of each individual season would be more appropriate.

'''
```