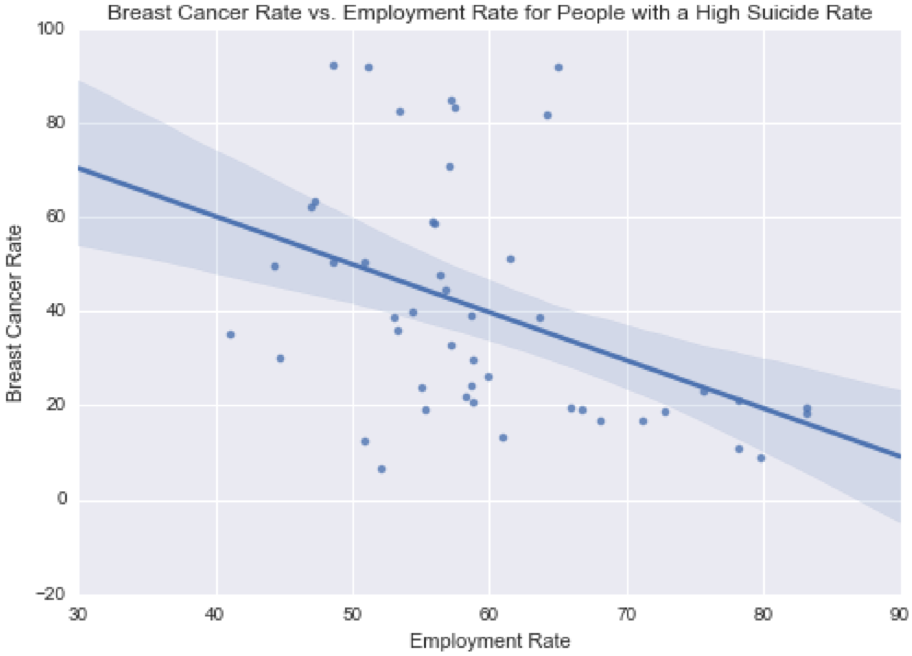
Output: Pearson coefficient, p value and r^2

Case 1: association between breast cancer rate and employment rate

r coefficient and p value
(-0.417536344446409588, 0.0034994561391527321)

r^2=17.43%

Pearson coefficient r=-0.42 shows the negative and moderate correlation between the breast cancer rate and employment rate. p value < 0.05 suggest a relationship between the breast cancer rate and employment rate. r^2=17.43% suggest a weak predictive power for this correlation.

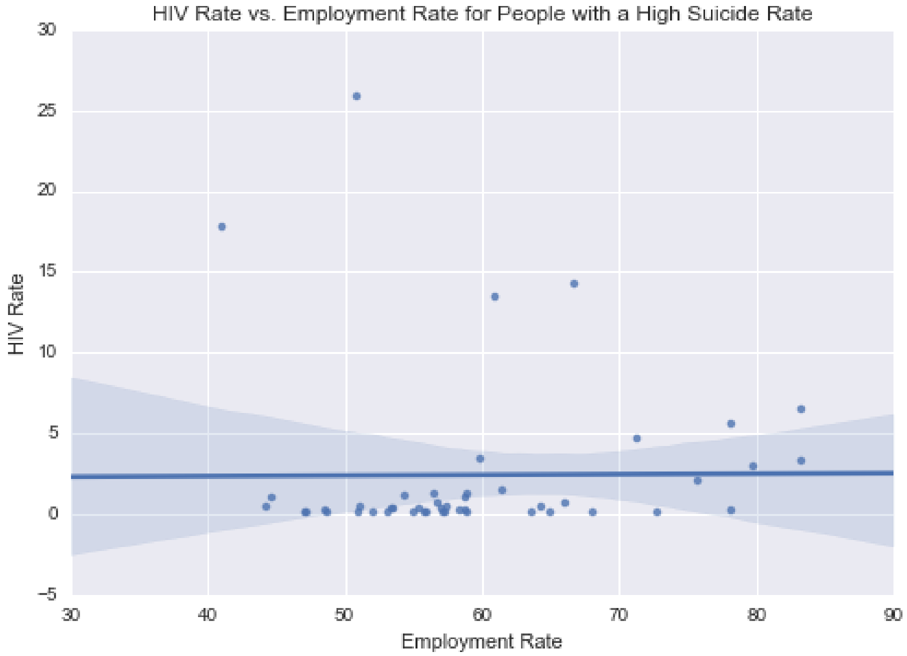


Case 2: association between HIV rate and employment rate

r coefficient and p value
(0.0076031649565114247, 0.95954722817525184)

r^2=0.006%

Pearson coefficient r=0.0078 shows there is no correlation between the HIV rate and employment rate. p value >> 0.05 suggest there is no relationship between the HIV rate and employment rate. r^2=0.006% suggest that there is no prediction can be made for this correlation.



Python Program

```
"""
Created on Tue Oct 27 2015

@author: violetgirl
"""
import pandas as pd
import numpy as np
import seaborn as sb
import scipy
import matplotlib.pyplot as plt

# load gapminder dataset
data = pd.read_csv('gapminder.csv',low_memory=False)
# lower-case all DataFrame column names
```

```
data.columns = map(str.lower, data.columns)
# bug fix for display formats to avoid run time errors
pd.set_option('display.float_format', lambda x:'%f'%x)

# setting variables to be numeric
data['suicideper100th'] =
data['suicideper100th'].convert_objects(convert_numeric=True)
data['breastcancerper100th'] =
data['breastcancerper100th'].convert_objects(convert_numeric=True)
data['hivrate'] = data['hivrate'].convert_objects(convert_numeric=True)
data['employrate'] = data['employrate'].convert_objects(convert_numeric=True)

# display summary statistics about the data
# print("Statistics for a Suicide Rate")
# print(data['suicideper100th'].describe())

# subset data for a high suicide rate based on summary statistics
sub = data[(data['suicideper100th']>12)]
# make a copy of my new subsetted data
sub_copy = sub.copy()
# remove missing values
sub_copy=sub_copy.dropna()

# Bivariate graph for association of breast cancer rate with HIV rate for people
with a high suicide rate
plt.figure(1)
sb.regplot(x="employrate",y="breastcancerper100th",fit_reg=True,data=sub_cop
y)
plt.xlabel('Employment Rate')
plt.ylabel('Breast Cancer Rate')
plt.title('Breast Cancer Rate vs. Employment Rate for People with a High Suicide
Rate')

plt.figure(2)
sb.regplot(x="employrate",y="hivrate",fit_reg=True,data=sub_copy)
plt.xlabel('Employment Rate')
plt.ylabel('HIV Rate')
plt.title('HIV Rate vs. Employment Rate for People with a High Suicide Rate')

print ('association between breast cancer rate and employment rate')
```

```
print('r coefficient and p value')
print
(scipy.stats.pearsonr(sub_copy['breastcancerper100th'],sub_copy['employrate']))
# r^2 = 0.1743 => 17.43%

print ('association between HIV rate and employment rate')
print('r coefficient and p value')
print (scipy.stats.pearsonr(sub_copy['hivrate'],sub_copy['employrate']))
# r^2 = 0.006%

# END
```