Analysis of the 2019 Index of Economic Freedom using Python

This project focuses on using data from the Heritage Foundation's *2019 Index of Economic Freedom* to come up with several conclusions on what indicators could possibly effect the economic freedom of a nation through the use of basic statistical analysis along with more advanced methods such as multivariate and logistical regressions.

While this dataset is robust, it does have the problem of using many arbitrary variables like the judicial effectiveness and government integrity scores.

In [2]:
```
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
```

In [3]:
```
pd
```

Out[3]:
```
<module 'pandas' from 'C:\\Users\\Woyte\\Anaconda3\\lib\\site-packages\\pandas\\__init__.py'>
```

In [4]:
```
df = pd.read_excel ('C:\\Users\\Woyte\\Documents\\Economic_Freedom.xlsx')
```

In [5]:
```
df = df.drop('WEBNAME', axis=1) #Dropping unnecessary and duplicated rows as well as setting a new unique i
ndex for the data.
```

In [6]:
```
df = df.drop('Country', axis=1)
```

In [7]:
```
df.set_index('CountryID', inplace=True)
```

In [8]:
```
df.head()
```

Out[8]:

	Country Name	Region	World Rank	Region Rank	2019 Score	Property Rights	Judicial Effectiveness	Government Integrity	Tax Burden	Gov't Spending	...	Gov't Expenditure % of GDP	Population (Millions)	(I
CountryID														
1.0	Afghanistan	Asia-Pacific	152.0	39.0	51.5	19.6	29.6	25.2	91.7	80.3	...	25.6	35.5	
2.0	Albania	Europe	52.0	27.0	66.5	54.8	30.6	40.4	86.3	73.9	...	29.5	2.9	
3.0	Algeria	Middle East and North Africa	171.0	14.0	46.2	31.6	36.2	28.9	76.4	48.7	...	41.4	41.5	
4.0	Angola	Sub-Saharan Africa	156.0	33.0	50.6	35.9	26.6	20.5	83.9	80.7	...	25.3	28.2	
5.0	Argentina	Americas	148.0	26.0	52.2	47.8	44.5	33.5	69.3	49.5	...	41.0	44.1	

5 rows × 31 columns

In [9]:
```
df.dropna(inplace=True) #Dropping countries with nulll values
```

In [10]:
```
df2=df.drop([179]) #Dropping Venezuela since it's inflation rate makes it an extreme outlier in the below r
egression
x = df2[['Unemployment (%)']+['Inflation (%)']]
y = df2['2019 Score'].values
x= sm.add_constant(x)
```

```
C:\Users\Woyte\Anaconda3\lib\site-packages\numpy\core\fromnumeric.py:2389: FutureWarning: Method .ptp is
deprecated and will be removed in a future version. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```

In [11]:
```
reg1 = sm.OLS(y,x, missing='drop')
```

In [12]:
```
results = reg1.fit()
```

In [13]:
```
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.145
Model:                            OLS   Adj. R-squared:                  0.135
Method:                 Least Squares   F-statistic:                     14.37
Date:                Thu, 17 Oct 2019   Prob (F-statistic):           1.72e-06
Time:                        01:19:20   Log-Likelihood:                -625.34
No. Observations:                 172   AIC:                             1257.
Df Residuals:                     169   BIC:                             1266.
Df Model:                           2
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const               65.2065      1.243     52.457      0.000      62.753      67.660
Unemployment (%)    -0.1187      0.125     -0.952      0.343      -0.365       0.128
Inflation (%)       -0.6033      0.116     -5.186      0.000      -0.833      -0.374
==============================================================================
Omnibus:                        0.681   Durbin-Watson:                   1.806
Prob(Omnibus):                  0.711   Jarque-Bera (JB):                0.353
Skew:                           0.013   Prob(JB):                        0.838
Kurtosis:                       3.220   Cond. No.                         18.7
==============================================================================

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

First Regression: Economic Freedom Score Versus the Unemployment and Inflation Rates

This is the first regression test run on the data. It was a multi-variate regression test to find to what degree the two economic targets of most central banks - that being the inflation and unemployment targets - have on a country's economic freedom score. As the results show, while the inflation rate has a statistically significant effect on the score, the unemployment rate was statistically insignificant with a p-value of .343, higher than the standard confidence cutoffs of .05 and .1. As unemployment is statistically insignificant from the model, it is probably more efficient to remove it. Fortuntely, the data in the model was of a normal distribution, with only a minor skew of .013

In [14]:
```
x2 = df2['Inflation (%)']
x3= sm.add_constant(x2)
reg2= sm.OLS(y,x3, missing='drop')
results2 = reg2.fit()
print(results2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.141
Model:                            OLS   Adj. R-squared:                  0.136
Method:                 Least Squares   F-statistic:                     27.86
Date:                Thu, 17 Oct 2019   Prob (F-statistic):           3.94e-07
Time:                        01:19:20   Log-Likelihood:                -625.80
No. Observations:                 172   AIC:                             1256.
Df Residuals:                     170   BIC:                             1262.
Df Model:                           1
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
const            64.3764      0.885     72.715      0.000      62.629      66.124
Inflation (%)    -0.6119      0.116     -5.278      0.000      -0.841      -0.383
==============================================================================
Omnibus:                        0.935   Durbin-Watson:                   1.799
Prob(Omnibus):                  0.626   Jarque-Bera (JB):                0.579
Skew:                           0.068   Prob(JB):                        0.749
Kurtosis:                       3.249   Cond. No.                         9.64
==============================================================================

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

Second Regression: Unemployment removed

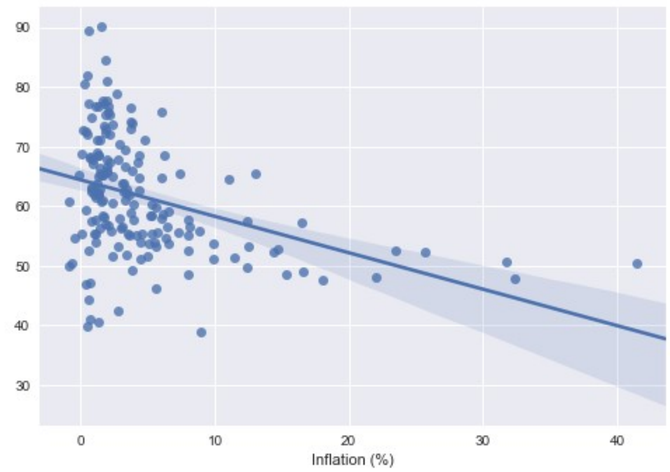
Turning the model into a simple linear regresion has changed the outputs in some interesting ways. For one, the R-squared value for the model has dropped by nearly half a percent. This means the model is slightly less capable of accounting for the variation within the data and will be less accurate. The AIC has also decreased by one, which is usually a good thing for a model, but, since AIC is supposed to penalize a model the more explanatory variables are being used, it makes sense to get a better value when removing a variable.

Final model: $2019Score = -.612(InflationPercentage) + 64.44 + Error$

In layman's terms, starting out at a value of 64.44, a country's 2019 score typically drops by .612 points for every percentage increase in inflation.

In [15]:
```
%matplotlib inline
plt.style.use('seaborn')
sns.regplot(x2,y)
```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x203b08a8d68>



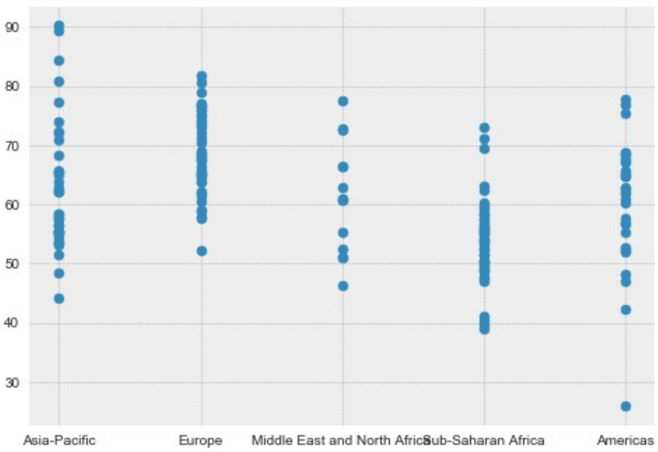
With that question answered, let's see which region of the world has the highest economic freedom on average

In [16]:
```
Region_Score=df['2019 Score'].groupby([df['Region']]).mean()
Region_Score
```

Out[16]:
```
Region
Americas                        60.526667
Asia-Pacific                    62.555000
Europe                          68.665116
Middle East and North Africa    61.257143
Sub-Saharan Africa              54.026087
Name: 2019 Score, dtype: float64
```

In [17]:
```
plt.style.use('bmh')
plt.scatter('Region', '2019 Score', data=df)
```

Out[17]: <matplotlib.collections.PathCollection at 0x203b10640f0>



Based on the data, Europe is the region with the highest amount of economic freedom and has the narrowest variation in that regard. Interestingly, the Americas have a lower economic freedom score on average than both Asia-Pacific and the Middle East and North African regions. Also Sub-Saharan Africa comes in dead last for economic freedom, only have about 79% of the economic freedom of an average european country.

Comparing the annual FDI and the Percentage Tariff and Corporate Tax Rates to the Unemployment Rate

In [18]:
```
x4 = df[['FDI Inflow (Millions)']+['Tariff Rate (%)']+['Corporate Tax Rate (%)']]
y2=df['Unemployment (%)'].values
x5= sm.add_constant(x4)
reg4= sm.OLS(y2,x5, missing='drop')
results3 = reg4.fit()
print(results3.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.012
Model:                            OLS   Adj. R-squared:                 -0.006
Method:                 Least Squares   F-statistic:                    0.6659
Date:                Thu, 17 Oct 2019   Prob (F-statistic):              0.574
Time:                        01:19:21   Log-Likelihood:                -544.40
No. Observations:                 173   AIC:                             1097.
Df Residuals:                     169   BIC:                             1109.
Df Model:                           3
Covariance Type:            nonrobust
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
const                      6.9560      1.281      5.430      0.000       4.427       9.485
FDI Inflow (Millions)  -2.166e-05   1.67e-05     -1.295      0.197   -5.47e-05    1.14e-05
Tariff Rate (%)           -0.0340      0.106     -0.321      0.748      -0.243       0.175
Corporate Tax Rate (%)     0.0312      0.051      0.607      0.544      -0.070       0.133
==============================================================================
Omnibus:                       56.552   Durbin-Watson:                   1.831
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              111.358
Skew:                           1.558   Prob(JB):                     6.59e-25
Kurtosis:                       5.395   Cond. No.                     8.20e+04
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 8.2e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

C:\Users\Woyte\Anaconda3\lib\site-packages\numpy\core\fromnumeric.py:2389: FutureWarning: Method .ptp is
deprecated and will be removed in a future version. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```

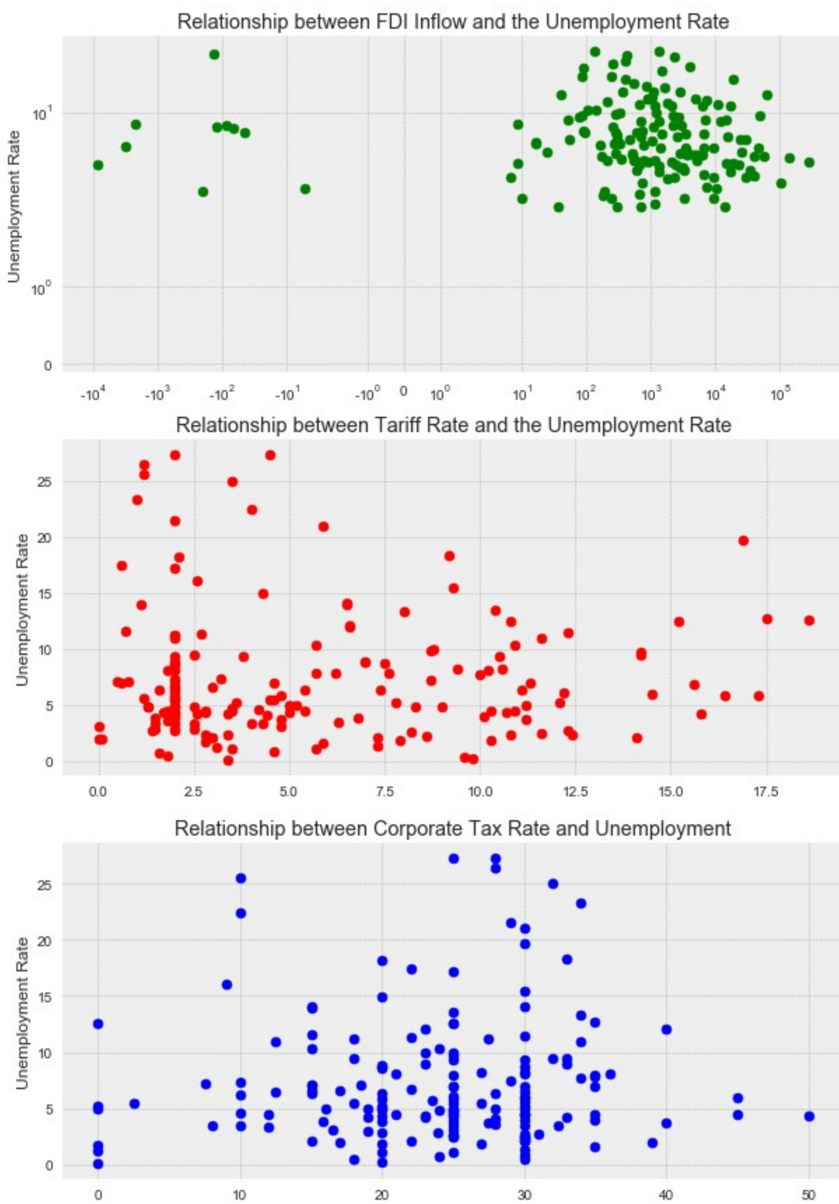
As the data for this shows, none of the inflows to either the government or private sector have any statistical significance to the inflation rate as their p-values are all below the golden theshold of .05, and the lesser used threshold of .1. Moreover, the R-squared value is only .012, so even if the values did have some statistical significance, this model would have much predictive power. Overall the model is not very useful in the analysis.

In [19]:

```
fig, (ax1, ax2, ax3) = plt.subplots(figsize=(10,15), nrows=3, ncols=1)
ax1.scatter(df['FDI Inflow (Millions)'],y2, c='g')
ax1.set_title('Relationship between FDI Inflow and the Unemployment Rate')
ax1.set_ylabel('Unemployment Rate')
ax1.set_xscale('symlog')
ax1.set_yscale('symlog')
ax2.scatter(df['Tariff Rate (%)'],y2, c='red')
ax2.set_title('Relationship between Tariff Rate and the Unemployment Rate')
ax2.set_ylabel('Unemployment Rate')
ax3.scatter(df['Corporate Tax Rate (%)'],y2, c='blue')
ax3.set_title('Relationship between Corporate Tax Rate and Unemployment')
ax3.set_ylabel('Unemployment Rate')
```

Out[19]:

```
Text(0, 0.5, 'Unemployment Rate')
```



These scatterplots are meant to show the relationship between the unemployment rate and the three variables run in the regression. As the FDI scatterplot had an extreme outlier in it, I decided to run the logged version of their relationship to better visualize it. These charts also show next to no relationship between the three x variables and the unemployment rate, agreeing with the findings of the regression analysis run above.

Determining the Probability of a Country being in the Top 10% of Countries Ranked on the Index Based on their Government Integrity, Tax Burden, and Public Debt

In [20]:
```
# Step 1: Create a new binary variable to determine which countries are in the top 10%.
df['Top 10%'] = np.where(df['World Rank']<=18, 1, 0)
df.head()
```

Out[20]:

	Country Name	Region	World Rank	Region Rank	2019 Score	Property Rights	Judicial Effectiveness	Government Integrity	Tax Burden	Gov't Spending	...	Population (Millions)	GDP (Billions, PPP)	G Grov R
CountryID														
1.0	Afghanistan	Asia-Pacific	152.0	39.0	51.5	19.6	29.6	25.2	91.7	80.3	...	35.5	69.6	
2.0	Albania	Europe	52.0	27.0	66.5	54.8	30.6	40.4	86.3	73.9	...	2.9	36.0	
3.0	Algeria	Middle East and North Africa	171.0	14.0	46.2	31.6	36.2	28.9	76.4	48.7	...	41.5	632.9	
4.0	Angola	Sub-Saharan Africa	156.0	33.0	50.6	35.9	26.6	20.5	83.9	80.7	...	28.2	190.3	
5.0	Argentina	Americas	148.0	26.0	52.2	47.8	44.5	33.5	69.3	49.5	...	44.1	920.2	

5 rows × 32 columns

In [22]:
```
#Define new x and y, add a constant, and run the logistical regression
x5 = df[['Government Integrity']+['Tax Burden']+['Public Debt (% of GDP)']]
y3=df['Top 10%'].values
x6= sm.add_constant(x5)
logit1 = sm.Logit(y3, x6, missing='drop')
results4 = logit1.fit().params
results4
```

```
Optimization terminated successfully.
         Current function value: 0.116421
         Iterations 9
```

Out[22]:
```
const                    -14.611586
Government Integrity       0.153614
Tax Burden                 0.055177
Public Debt (% of GDP)    -0.019435
dtype: float64
```

In [ ]:
```
#Plot the individual curves
plt.style.use('bmh')
sns.lmplot(x= 'Government Integrity', y= 'Top 10%', data=df, logistic=True, y_jitter=.01)
sns.lmplot(x= 'Tax Burden', y= 'Top 10%', data=df, logistic=True, y_jitter=.01)
sns.lmplot(x= 'Public Debt (% of GDP)', logistic=True, y= 'Top 10%', data=df, y_jitter=.01)
```

This test also proved to be statistically insignificant. While government integrity by itself proves to be a good predictor of if a country is in the top 10% on the *2019 Index of Economic Freedom*, the other independent variables were not nearly as good predictors, so the whole model was insignificant. With a p-value of around .12, this model fails to meet the 95% or 90% confidence thresholds required for statisticaly significance.

Determining the Relationship Between all the Relevant Independent Variable's and a Country's Economic Freedom Score in 2019

Null hypothesis: There is no statistically significant relationship between any of the independent variables and a country's Economic Freedom Score in 2019

Alternative hypothesis: There is a statistically significant relationship between at least one of the indepedent variables and a country's Economic Freedom Score in 2019

In [ ]:
```
x7= df.drop(['Country Name', 'World Rank', 'Region Rank', 'Region', '2019 Score', 'Top 10%'], axis=1)
x8=sm.add_constant(x7)
y4= df['2019 Score']
reg5= sm.OLS(y4,x8, missing='drop')
results4 = reg5.fit()
print(results4.summary())
```

While the model proved to be statistically significant, many of the variables within the model proved not to be so. Ironically, it was mostly the semi-arbitary numbers that proved to have a statisticaly significant relationship with the 2019 score. These numbers are the ones created by the Heritage Foundation, like a country's trade freedom, labor freedom, and business freedom. The hard data like the income tax rate, corporate tax rate, and GDP per capita were all deemed stastically insignificant. In terms of accounting for the noise and variation in the data though, this model is perfect, getting an R-squared value of 1.0.

The null hypothesis is therefore rejected and the alternative is not rejected.

Granted, with all the statistically insignificant independent variables, this model could be made better, so I'm going to clean it up a bit.

In [ ]:
```
# While I remove all the insignicant columns in two steps here, I did check the columns independently to ma
ke sure they were still statistically insignificant when I removed the columns with the highest p-value.
x9=x7.drop(['Public Debt (% of GDP)', 'GDP per Capita (PPP)', 'GDP (Billions, PPP)', 'Income Tax Rate (%)',
'Unemployment (%)', 'GDP Growth Rate (%)', 'Population (Millions)', '5 Year GDP Growth Rate (%)','FDI Inflo
w (Millions)', 'Tariff Rate (%)', 'Corporate Tax Rate (%)', 'Inflation (%)', 'Tax Burden % of GDP'], axis=
1)
x10=x9.drop(x9.columns[-1], axis=1)
reg6= sm.OLS(y4,x10, missing='drop')
results5 = reg6.fit()
print(results5.summary())
```

This model is a lot better and all the independent variables in it are statistically significant. The AIC for the model has also improved over the AIC for the last matter, which is a good sign. The final model concludes that each single point increase in any of the indicators created by the Heritage Foundation will increase a country's Economic Freedom Score by about .083 points with no statistically significant constant.

Principle Component Analysis for Several of the Above Indicators by Regions

In [ ]:
```
features = ['Property Rights', 'Judical Effectiveness', 'Government Integrity', 'Tax Burden']
x11 = df.loc[:, features].values
y5 = df.loc[:,['Region']].values
x11 = StandardScaler().fit_transform(x11)
#Standardizing the data to make sure change in variance due to scaling differences doesn't effect future PC
A
```

In [ ]:
```
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(x11)
df3 = pd.DataFrame(data = principalComponents,columns = ['principal component 1', 'principal component 2'])
pcadf = pd.concat([df3, df[['Region']]], axis=1)
```

In [ ]:
```
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot(1,1,1)
ax.set_xlabel('Principal Component 1', fontsize = 16)
ax.set_ylabel('Principal Component 2', fontsize = 16)
ax.set_title('Principle Component Relations', fontsize = 20)
targets = ['Americas', 'Europe', 'Sub-Saharan Africa']
colors = ['r', 'g', 'b']
for region, color in zip(targets,colors):
    indicesToKeep = pcadf['Region'] == region
    ax.scatter(pcadf.loc[indicesToKeep, 'principal component 1']
               , pcadf.loc[indicesToKeep, 'principal component 2']
               , c = color
               , s = 50)
ax.legend(targets)

ax.grid()
```

In [ ]:
```
pca.explained_variance_ratio_
```

Results:

As the chart shows and information below it show, the first principle component accounts for 71% of the total variation in the data, and the second principle component accounts for another 22%. In total the two principle components accounts for 93% of the total variation in the data.

Linear Discriminant Analysis

In [ ]:
```
#resusing the data for x and y in the PCA scenario for LDA
x11
y5
lda = LDA(n_components=2)
x11 = lda.fit_transform(x11,y5)
```

In [ ]:
```
df['PC1'] = x11[:,0]
df['PC2'] = x11[:,1]
ldaplot = sns.lmplot(data = df[['PC1','PC2','Region']], x = 'PC1', y = 'PC2',fit_reg=False, hue = 'Region')
```

In [ ]:
```
sns.pairplot(df[['PC1','PC2', 'Region']], hue='Region')
```

In [ ]:
```
lda.explained_variance_ratio_
```

Results:

As the linear discriminant analysis above was used on a different, much larger section of the data that the principal component analysis, not much can really be said on the accuracy of the LDA over PCA. What can be said is that for the data above, the LDA was able to account for 78% of the data with the first component, and another 29% with the second. The analysis as a whole was therefore able to predict the variation within the data to 97%, an extremely good outcome for the model.