IMPACTS OF TRADE ON NATIONAL PRODUCTIVITY AND WELL-BEING

Team

3-11-2020

CORE MESSAGE & HYPOTHESIS OF THE PROJECT

- If the GDP of a nation rises, then the nation will be able to import more and experience improvements in well-being (life expectancy & infant mortality).
- However, the nation will export less as it becomes more productive, since it will be able to produce more of its own goods.

QUESTIONS& MOTIVATIONS

- We were most interested in first seeing the strength of the relationship between GDP per Capita and trade values (imports & exports), with the assumption that exports would decrease in correlation to GDP increase.
- Then we were interested in seeing how the GDP per Capita would effect national well-being (life expectancy & infant mortality).
- The main goal was to best identify how to improve the well-being of a nation.

SOURCES OF DATA

- Originally, we started by downloading data for both GDP and World Development Indicators on Kaggle as well as the World Bank website.
- We ran into the following issues:
  - We couldn't find columns to merge all the data sets.
  - The Kaggle data didn't have data for years past 1980.
  - There was a need to transpose the year rows in Kaggle data to get indicator value-years as columns.
  - This would exponentially increase the size of the merged data-frames, complicating merges and Git pushing.
- For this reason, we decided to exclusively use the World Bank website.

DATA CLEANUP PROCESS

- We merged the tables for GDP per Capita, trade values, and well-being indicators.
- In order to do that, we had to use the Add_Suffix function to ensure that each set of columns could be properly identified.
- To enable merging by Country Code column, we then used the Rename function to drop the suffix for those.

In [3]:

```
#Add Suffixes to Columns Except Country Code
#Code found on https://stackoverflow.com/questions/34049618/how-to-add-a-suffix-or-prefix-to-each-column-name
#Add a Suffix to Differentiate Columns For Merge
gdp_per_capita_df = gdp_per_capita_df.add_suffix("_GDP_Cap")
#Drop Suffix from 'CountryCode' column to Allow Merge
gdp_per_capita_df = gdp_per_capita_df.rename(columns = {"CountryCode_GDP_Cap":"CountryCode"})

#Add a Suffix to Differentiate Columns For Merge
life_expectancy_df = life_expectancy_df.add_suffix("_Life")
#Drop Suffix from 'CountryCode' column to Allow Merge
life_expectancy_df = life_expectancy_df.rename(columns = {"CountryCode_Life":"CountryCode"})

#Add a Suffix to Differentiate Columns For Merge
infant_mortality_df = infant_mortality_df.add_suffix("_Mortality")
#Drop Suffix from 'CountryCode' column to Allow Merge
infant_mortality_df = infant_mortality_df.rename(columns = {"CountryCode_Mortality":"CountryCode"})

#Add a Suffix to Differentiate Columns For Merge
exports_df = exports_df.add_suffix("_Exports")
#Drop Suffix from 'CountryCode' column to Allow Merge
exports_df = exports_df.rename(columns = {"CountryCode_Exports":"CountryCode"})

#Add a Suffix to Differentiate Columns For Merge
imports_df = imports_df.add_suffix("_Imports")
#Drop Suffix from 'CountryCode' column to Allow Merge
imports_df = imports_df.rename(columns = {"CountryCode_Imports":"CountryCode"})
```

DATA CLEANUP PROCESS (RESULTS)

	CountryName_GDP_Cap	CountryCode	IndicatorName_GDP_Cap	IndicatorCode_GDP_Cap	1960_GDP_Cap	1961_GDP_Cap	1962_GDP_Cap	1963_GDP_
0	Aruba	ABW	GDP per capita (current US$)	NY.GDP.PCAP.CD	NaN	NaN	NaN	
1	Afghanistan	AFG	GDP per capita (current US$)	NY.GDP.PCAP.CD	59.773194	59.860874	58.458015	78.706
2	Angola	AGO	GDP per capita (current US$)	NY.GDP.PCAP.CD	NaN	NaN	NaN	
3	Albania	ALB	GDP per capita (current US$)	NY.GDP.PCAP.CD	NaN	NaN	NaN	
4	Andorra	AND	GDP per capita (current US$)	NY.GDP.PCAP.CD	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	

DP_Cap	...	2010_Imports	2011_Imports	2012_Imports	2013_Imports	2014_Imports	2015_Imports	2016_Imports	2017_Imports	2018_Imports	2019_Imports
NaN	...	1.854190e+09	2.136313e+09	2.089385e+09	2.130726e+09	2.113408e+09	2.035754e+09	1.975419e+09	2.031844e+09	NaN	NaN
.108305	...	9.950050e+09	1.190656e+10	1.252506e+10	1.133990e+10	1.053200e+10	8.647160e+09	8.079197e+09	9.153344e+09	NaN	NaN
NaN	...	3.568226e+10	4.394739e+10	4.591971e+10	4.930420e+10	5.047164e+10	3.849951e+10	2.552939e+10	2.839711e+10	NaN	NaN
NaN	...	5.792202e+09	6.700290e+09	5.860856e+09	5.999409e+09	6.243051e+09	5.071012e+09	5.436291e+09	6.070288e+09	6.856193e+09	NaN
NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...

DATA ANALYSIS PROCESS – EXPORTS & IMPORTS

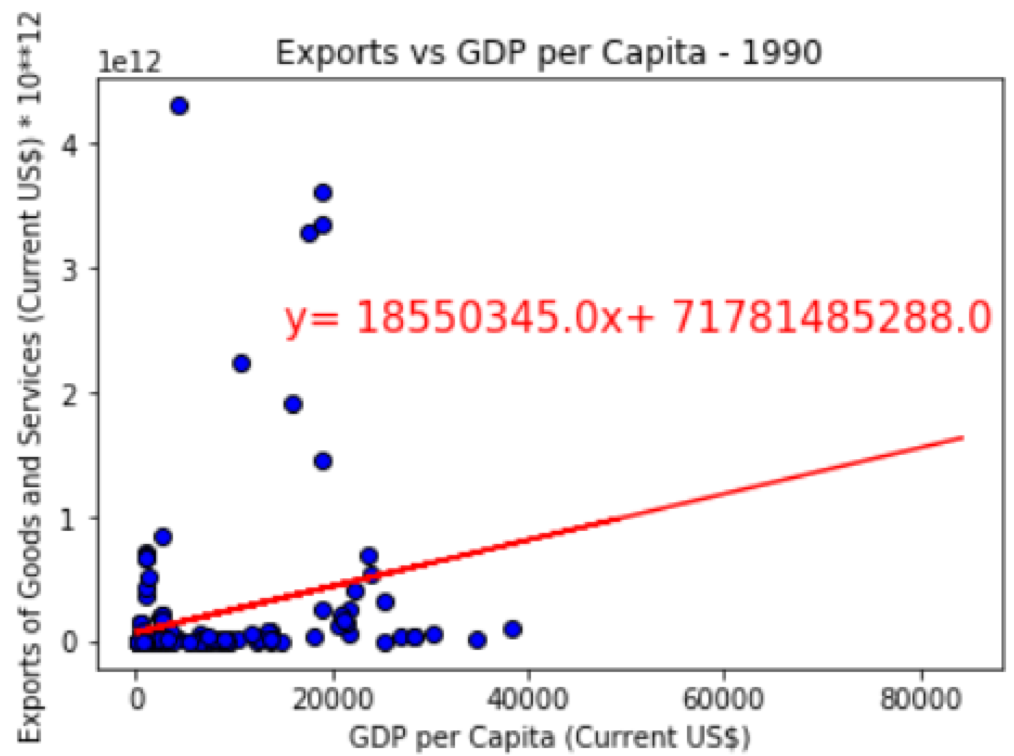
DATA ANALYSIS PROCESS – EXPORTS & IMPORTS

- We started by looking at scatter plots showing the relationship between the trade indicators and GDP per Capita for years 1990 & 2017.
- Linear regression couldn't initially be done, because there were NaN values in the data.
- The NumPy.IsNaN function was used to create a mask to filter out the NaN values to enable regression.

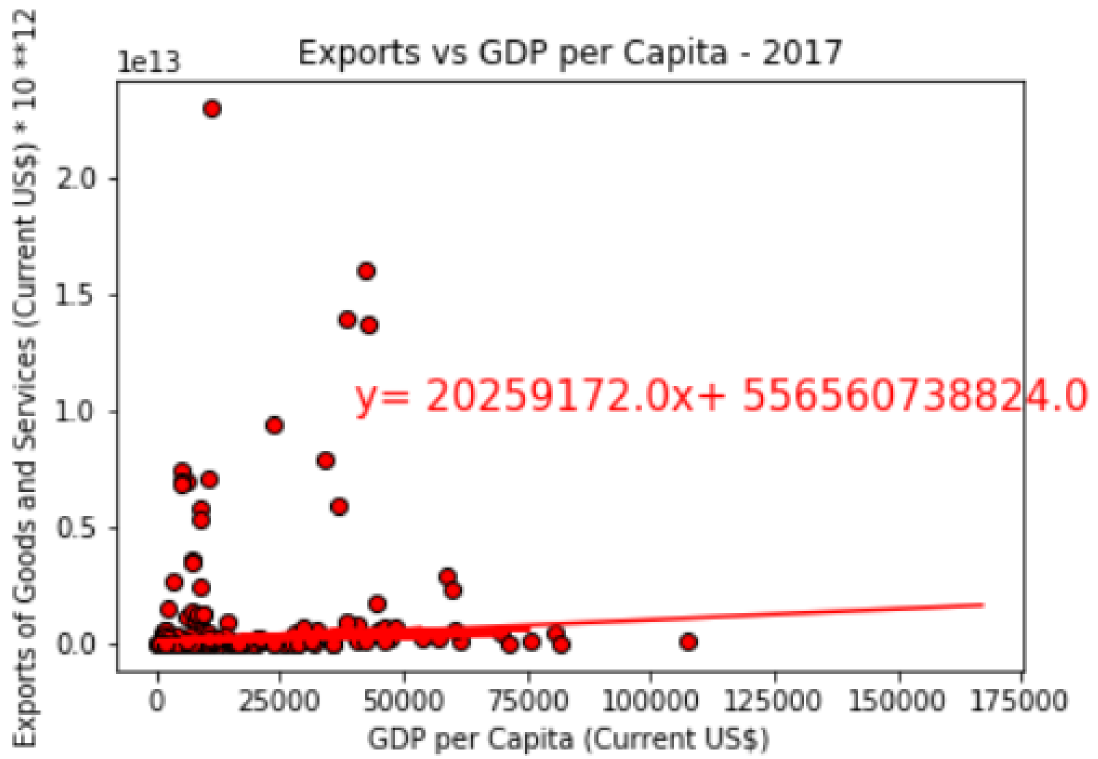
```
#Create a Scatter Plot of Exports vs GDP per Capita

#1990
#Apply Mask to Handle NaN Data
#Code Found on https://stackoverflow.com/questions/13643363/linear-regression-of-arrays-containing-nans-in-python-numpy
mask = ~np.isnan(summary_df['1990_GDP_Cap']) & ~np.isnan(summary_df['1990_Exports'])
#Make the Regression Parameters
(slope, intercept, rvalue, pvalue, stderr) = linregress(summary_df['1990_GDP_Cap'][mask], summary_df['1990_Exports'][mask])
#Calculate the Regress Values
regress_1990 = slope * summary_df['1990_GDP_Cap'] + intercept
#Create the Line Equation
line_eq_1990 = "y= " + str(round(slope,0)) + "x+ " + str(round(intercept,0))
#Plot the Export vs GDP per Capita Data
plt.scatter(summary_df['1990_GDP_Cap'], summary_df['1990_Exports'], c = "blue", edgecolors = "black")
#Plot the Regress Values
plt.plot(summary_df['1990_GDP_Cap'], regress_1990, c = "red")
#Annotate the Line Equation
plt.annotate(line_eq_1990, xy = (15000, 2.5*10**12), fontsize = 15, color = "red")
#Create the Labels
plt.title("Exports vs GDP per Capita - 1990")
plt.xlabel("GDP per Capita (Current US$)")
plt.ylabel("Exports of Goods and Services (Current US$) * 10**12")
#Display the R squared Value
print(f"The Rsquared value is {round(rvalue,2)}.")
#Save the Plot as a PNG
plt.savefig("Exports_vs_GDP_per_Capita_1990.png")
#Show the Plot
plt.show()
```

DATA ANALYSIS PROCESS – EXPORTS & IMPORTS

The Rsquared value is 0.06.



The Rsquared value is 0.02.

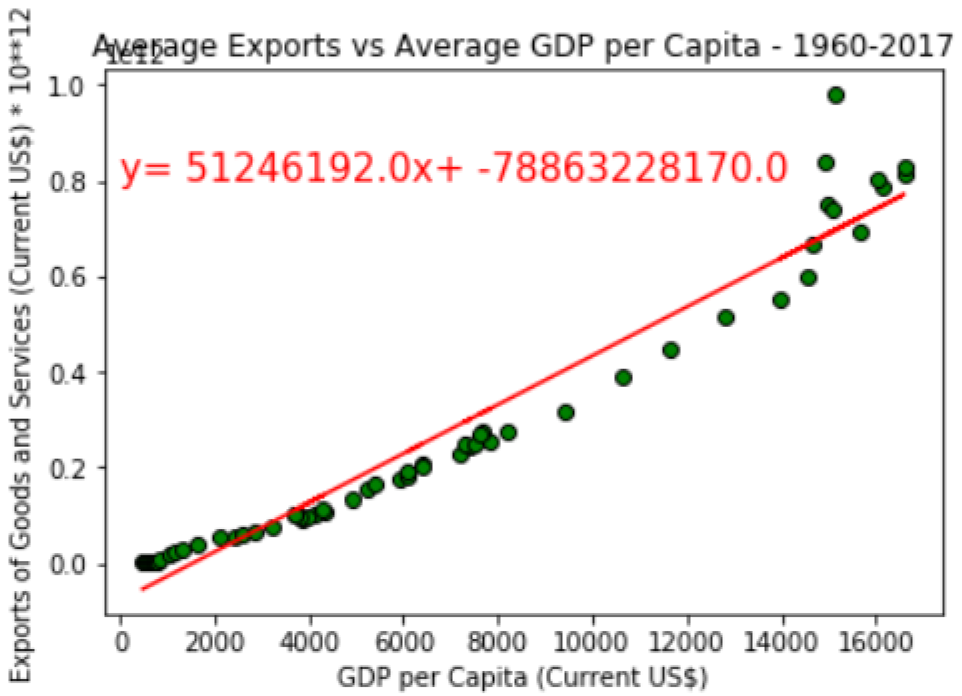


- The data showed that within any given year, the Exports didn't have a strong correlation with GDP per Capita, but this was because of outliers.
- The same held true for imports.
- Because the data was heavily skewed by outliers, a decision was made to further investigate this data.

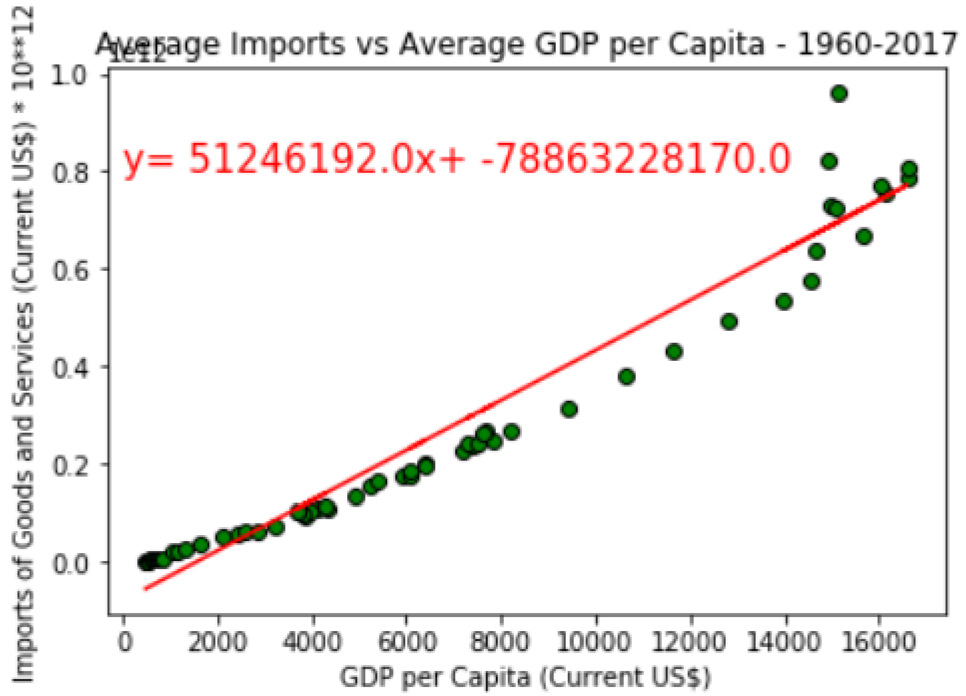
DATA ANALYSIS PROCESS – EXPORTS & IMPORTS (CONTINUED)

- To filter out the effects of the variation among countries, plots of average trade values vs average GDP per Capita were made.
- The mean values of individual year columns were calculated and put in lists.
- The results show that there is an extremely strong positive relationship between trade parameters and GDP per Capita.

The Rsquared value is 0.95.

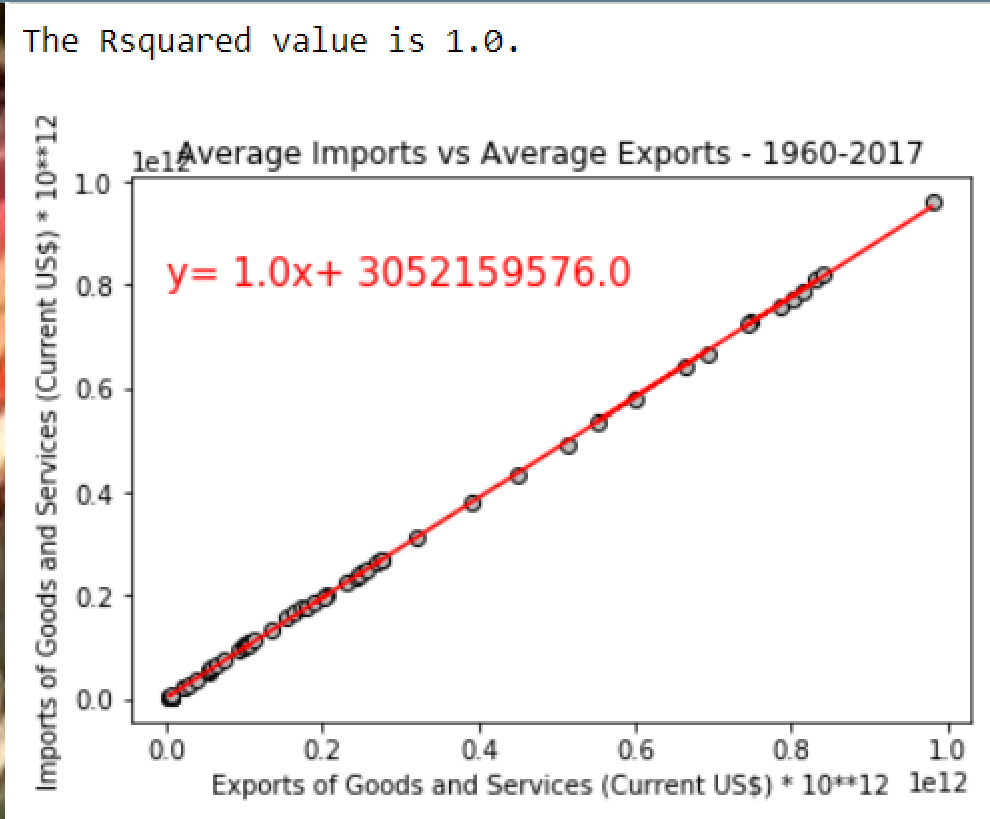


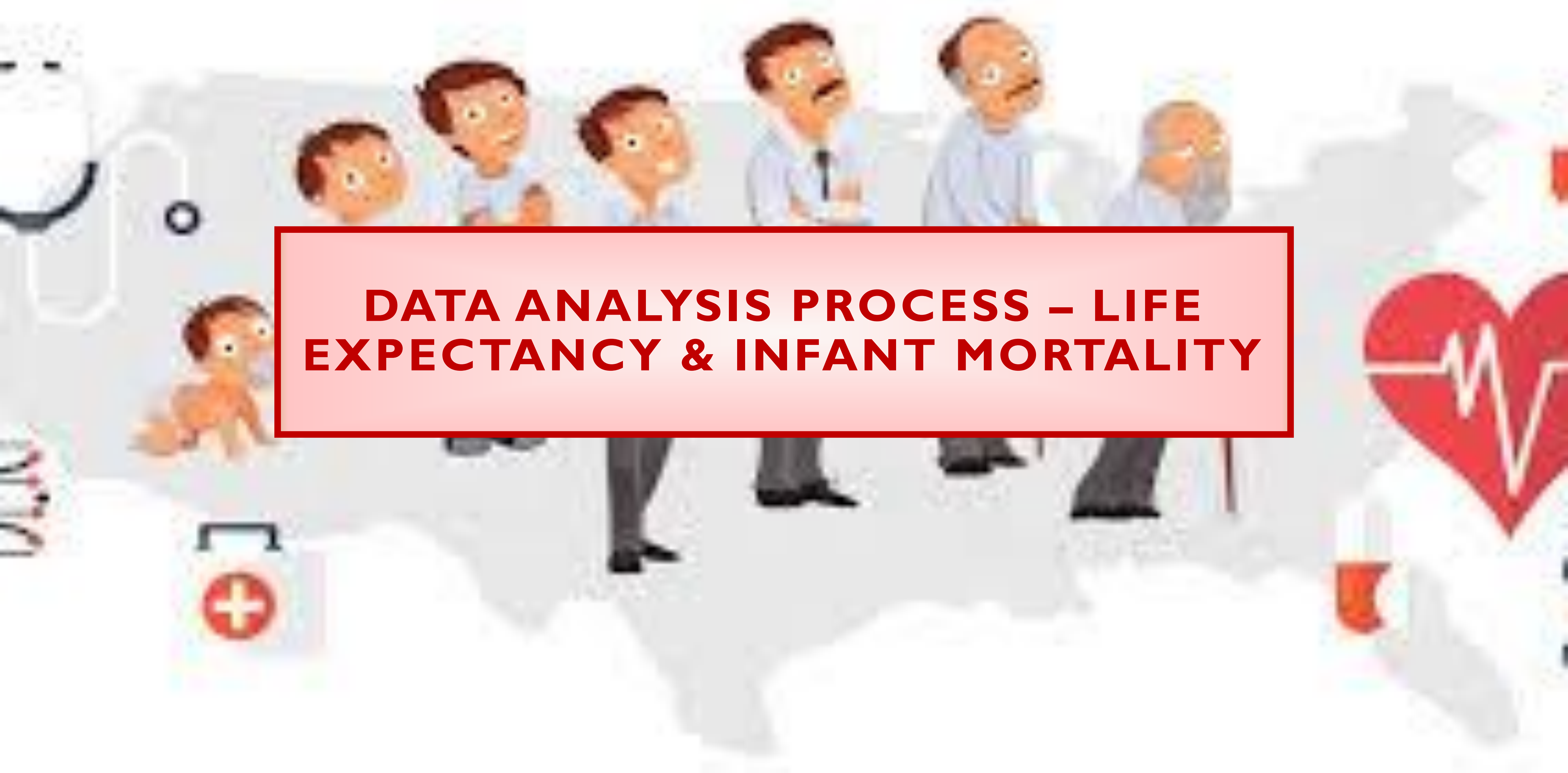
The Rsquared value is 0.95.



DATA ANALYSIS PROCESS – EXPORTS & IMPORTS (VALIDATION)

- As a means of validating the data from the World Bank, the average of Imports of all countries was compared to the average Exports of all countries for every available year.
- The data shows a virtually perfect linear correlation, indicating that the dollar value of each import was matched by the dollar amount of each export.
- This would tend to make the data more trustworthy.





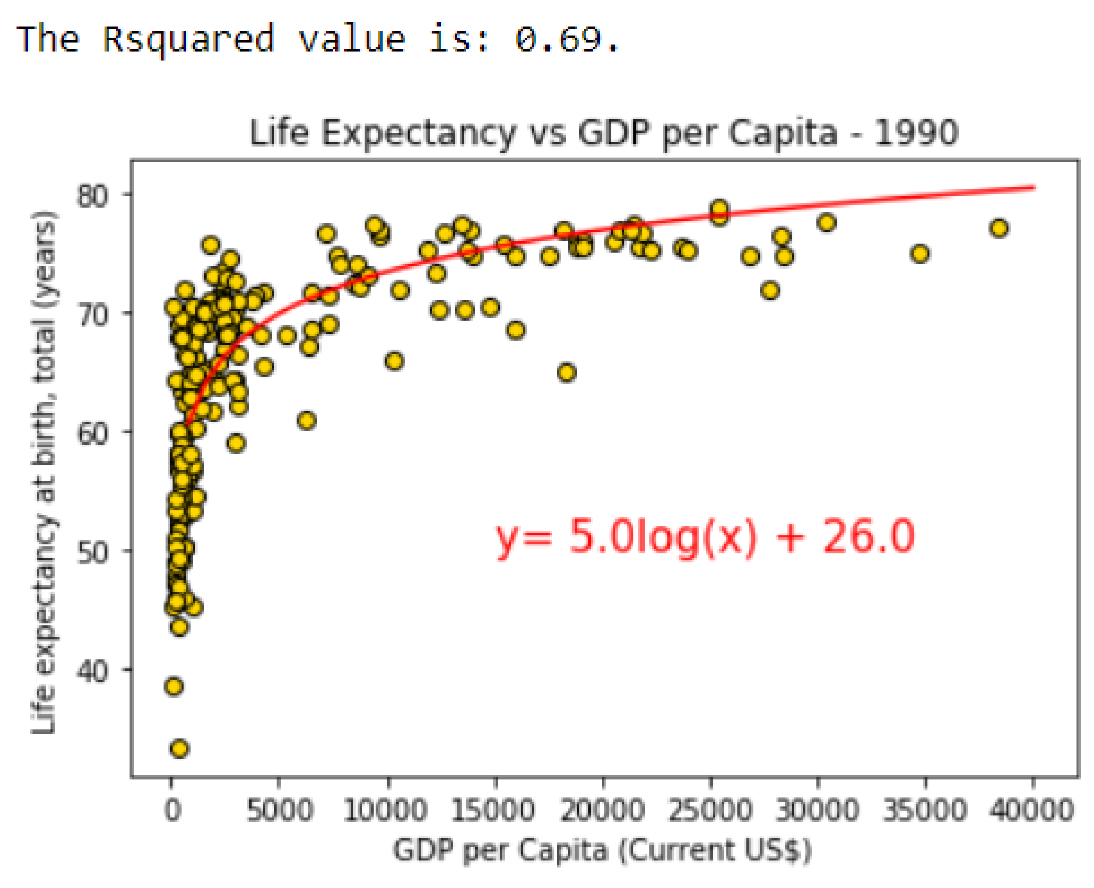
DATA ANALYSIS PROCESS – LIFE EXPECTANCY & INFANT MORTALITY

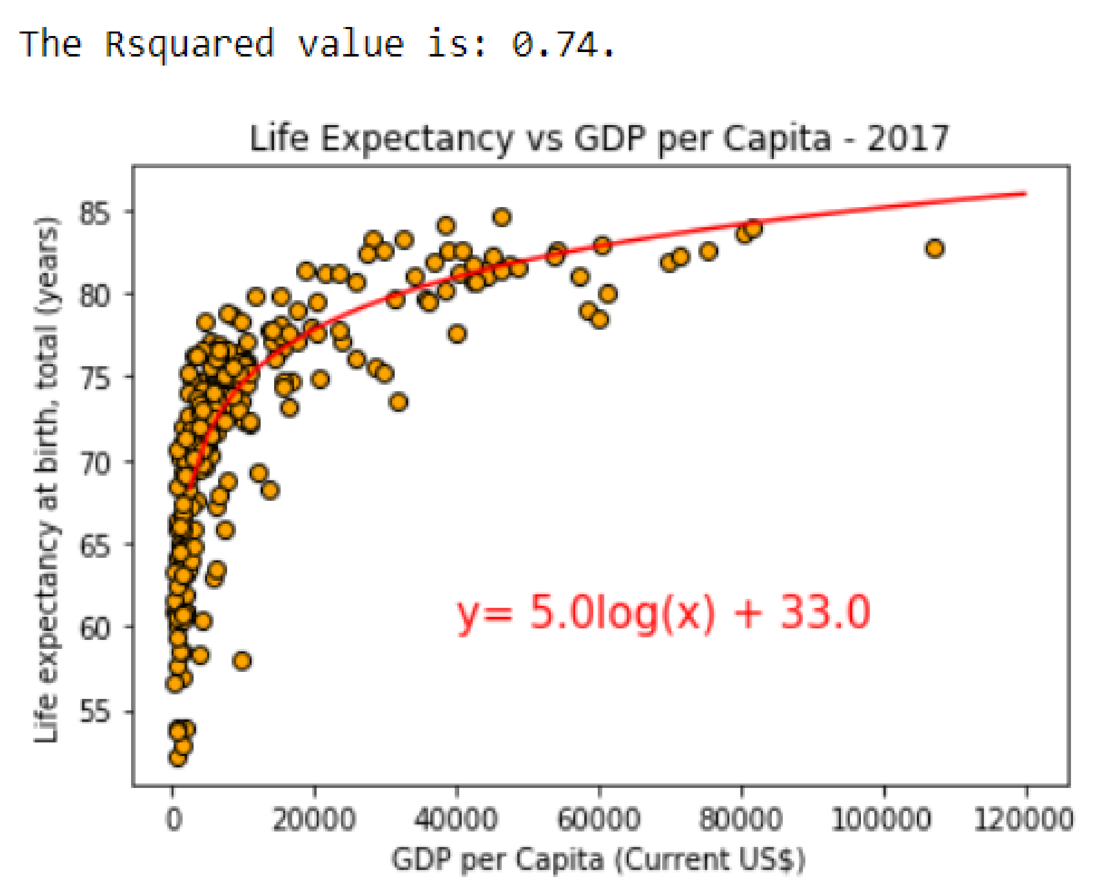
DATA ANALYSIS PROCESS – LIFE EXPECTANCY

- We started by looking at scatter plots showing the relationship between the life expectancy and GDP per Capita for years 1990 & 2017.
- The NumPy.IsNaN function was again used to create a mask to filter out the NaN values to enable regression.

```
#Create a Scatter Plot of Life Expectancy vs GDP per Capita

#1990
#Apply Mask to Handle NaN Data
#Code Found on https://stackoverflow.com/questions/13643363/linear-regression-of-arrays-containing-nans-in-python-numpy
mask = ~np.isnan(summary_df['1990_GDP_Cap']) & ~np.isnan(summary_df['1990_Life'])
#Make the Regression Parameters
(slope, intercept, rvalue, pvalue, stderr) = linregress(summary_df['1990_GDP_Cap'][mask], summary_df['1990_Life'][mask])
#Calculate the Regress Values
regress_1990 = slope * summary_df['1990_GDP_Cap'] + intercept
#Create the Line Equation
line_eq_1990 = "y= " + str(round(slope,0)) + "x+ " + str(round(intercept,0))
#Plot the Export vs GDP per Capita Data
plt.scatter(summary_df['1990_GDP_Cap'], summary_df['1990_Life'], c = "gold", edgecolors = "black")
#Plot the Regress Values
plt.plot(summary_df['1990_GDP_Cap'], regress_1990, c = "red")
#Annotate the Line Equation
plt.annotate(line_eq_1990, xy = (15000, 75), fontsize = 15, color = "red")
#Create the Labels
plt.title("Life Expectancy vs GDP per Capita - 1990")
plt.xlabel("GDP per Capita (Current US$)")
plt.ylabel("Life expectancy at birth, total (years)")
#Display the R squared Value
print(f"The Rsquared value is {round(rvalue,2)}.")
#Save the Plot as a PNG
plt.savefig("Life_Expectancy_vs_GDP_per_Capita_1990.png")
#Show the Plot
plt.show()
```

DATA ANALYSIS PROCESS – LIFE EXPECTANCY (CONTINUED)

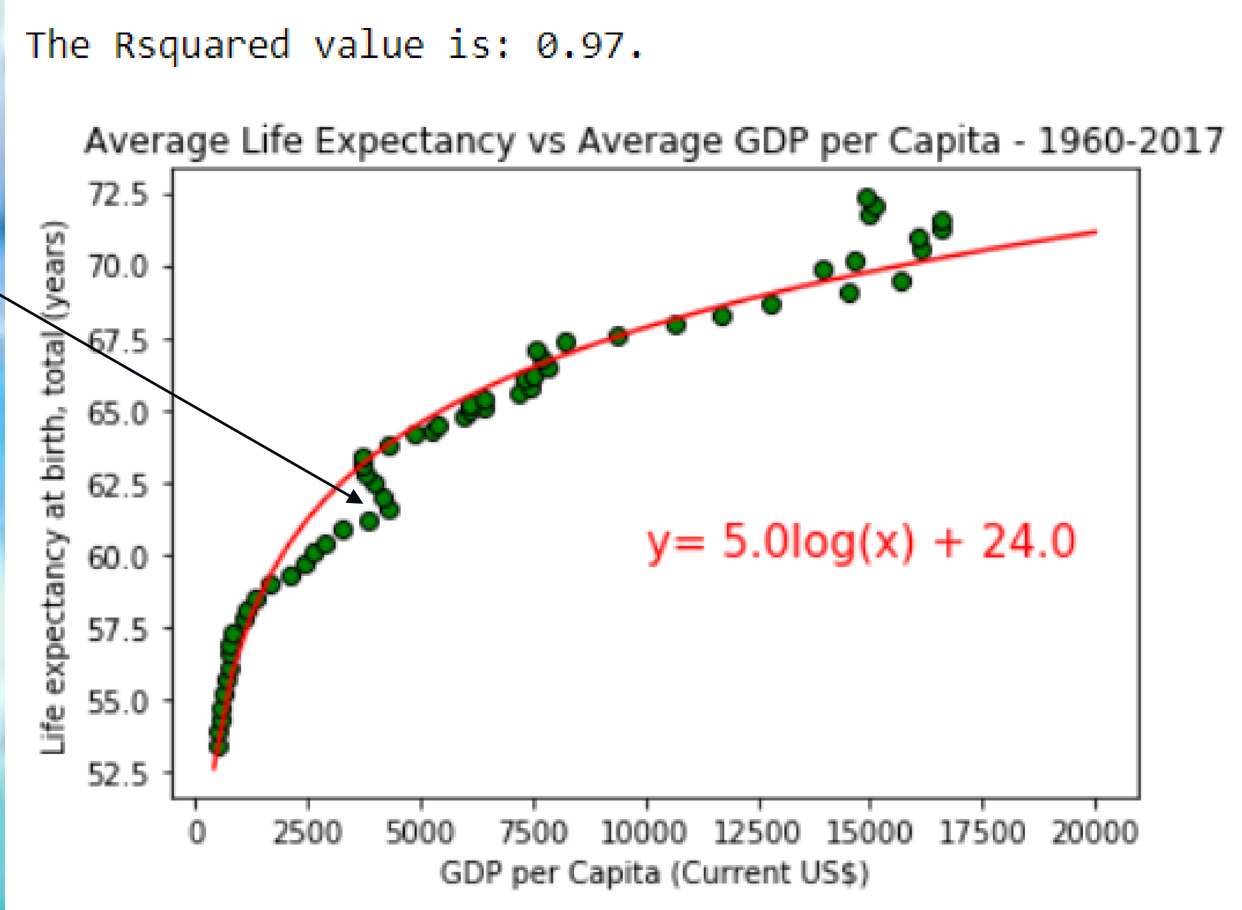




- The data showed that there wasn't a linear relationship between Life Expectancy and GDP per Capita.
- It has been shown that there is a strong logarithmic relationship between Life Expectancy and GDP per Capita.
- This means that Life Expectancy grows with GDP, but the rate of improvement slows as an economy advances.

DATA ANALYSIS PROCESS – LIFE EXPECTANCY (CONTINUED)

It is unclear what is responsible for the dip in the data here.



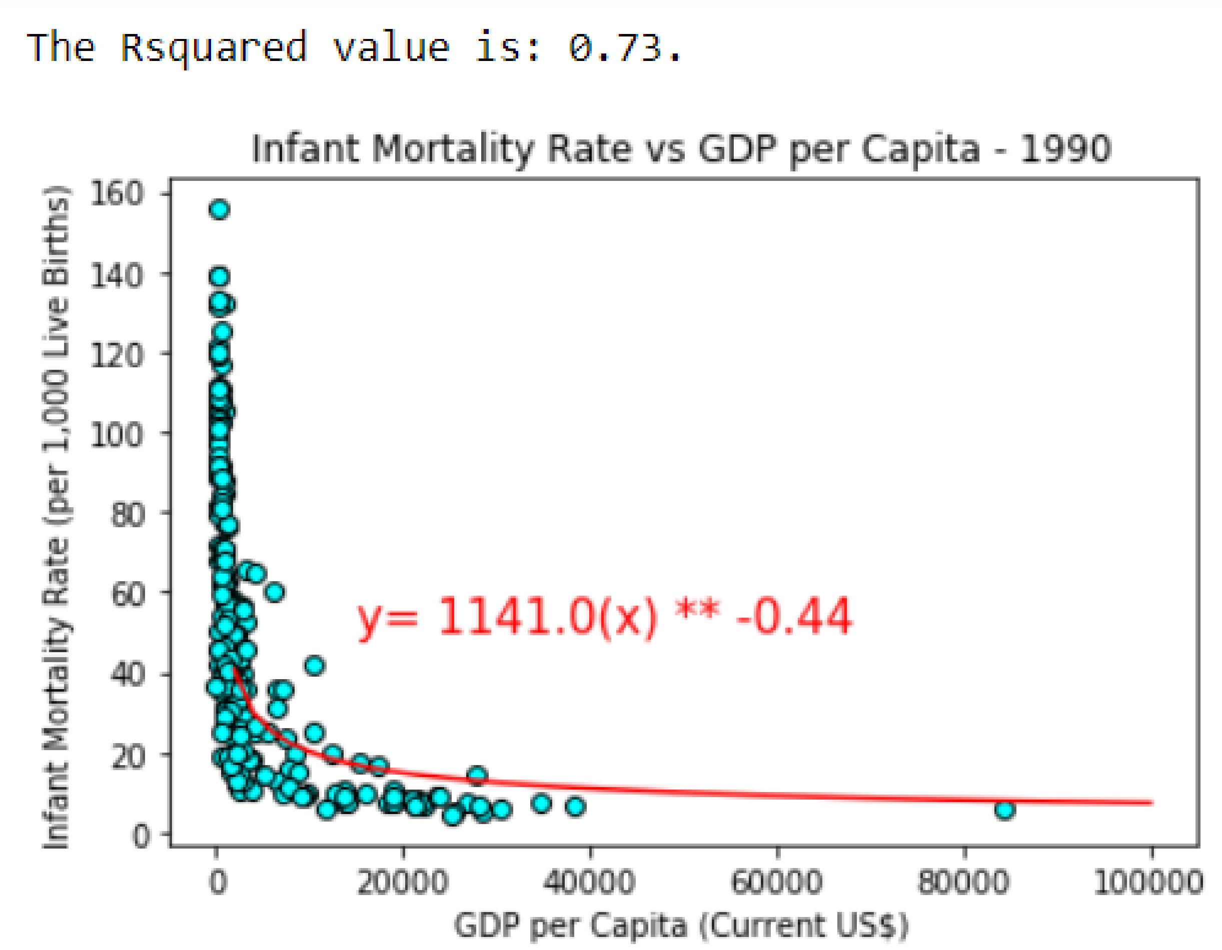
- To filter out the effects of the variation among countries, the plot of average Life Expectancy vs average GDP per Capita was made.
- The mean values of individual year columns were calculated and put in lists.
- The results show that there is an extremely strong positive relationship between trade parameters and GDP per Capita.

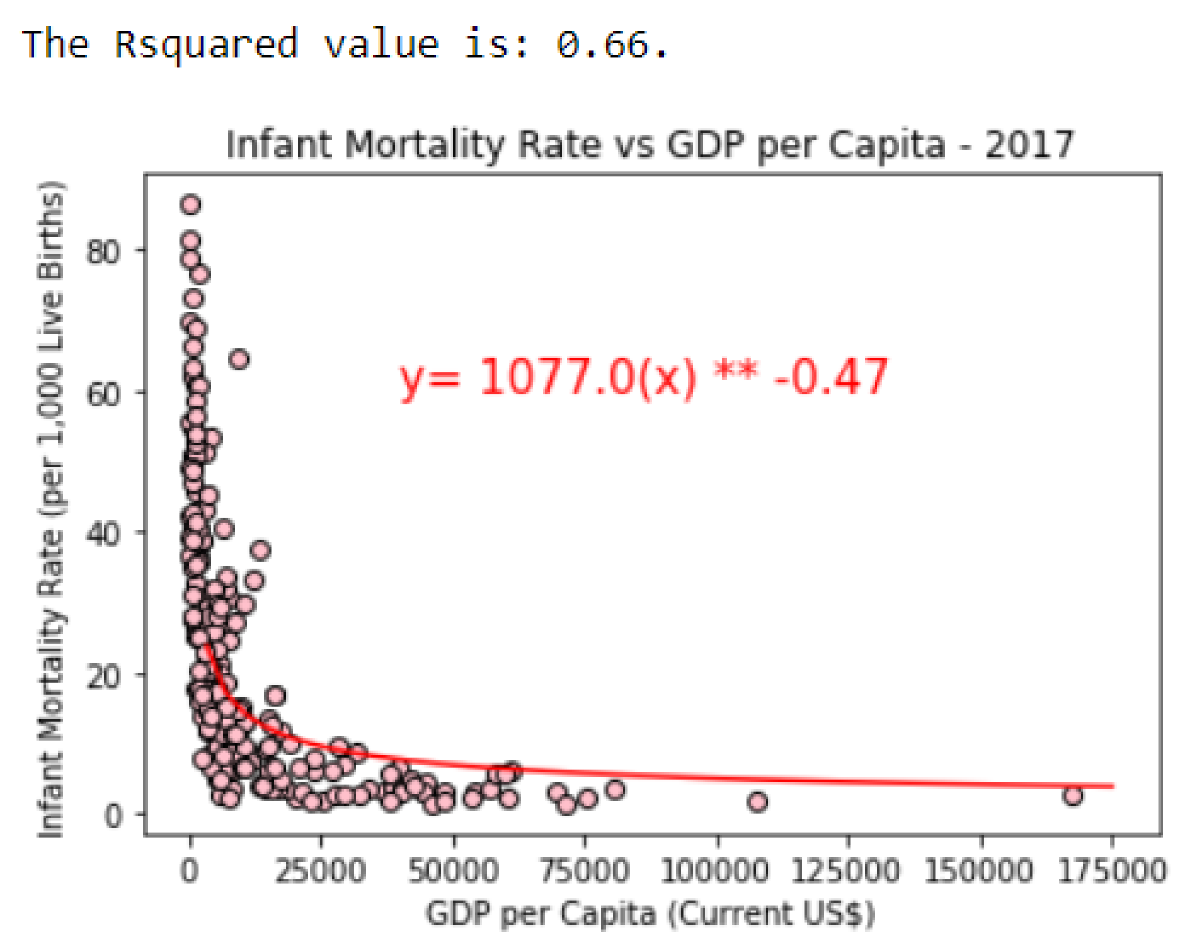
DATA ANALYSIS PROCESS – INFANT MORTALITY RATE

- We started by looking at scatter plots showing the relationship between infant mortality rate and GDP per Capita for years 1990 & 2017.
- The NumPy.IsNaN function was again used to create a mask to filter out the NaN values to enable regression.

```python
#Create a Scatter Plot of Infant Mortality Rate vs GDP per Capita

#1990
#Apply Mask to Handle NaN Data
#Code Found on https://stackoverflow.com/questions/13643363/linear-regression-of-arrays-containing-nans-in-python-numpy
mask = ~np.isnan(summary_df['1990_GDP_Cap']) & ~np.isnan(summary_df['1990_Mortality'])
#Make the Regression Parameters
(slope, intercept, rvalue, pvalue, stderr) = linregress(summary_df['1990_GDP_Cap'][mask], summary_df['1990_Mortality'][mask])
#Calculate the Regress Values
regress_1990 = slope * summary_df['1990_GDP_Cap'] + intercept
#Create the Line Equation
line_eq_1990 = "y= " + str(round(slope,0)) + "x+ " + str(round(intercept,0))
#Plot the Export vs GDP per Capita Data
plt.scatter(summary_df['1990_GDP_Cap'], summary_df['1990_Mortality'], c = "cyan", edgecolors = "black")
#Plot the Regress Values
plt.plot(summary_df['1990_GDP_Cap'], regress_1990, c = "red")
#Annotate the Line Equation
plt.annotate(line_eq_1990, xy = (15000, 75), fontsize = 15, color = "red")
#Create the Labels
plt.title("Infant Mortality Rate vs GDP per Capita - 1990")
plt.xlabel("GDP per Capita (Current US$)")
plt.ylabel("Infant Mortality Rate (per 1,000 Live Births)")
#Display the R squared Value
print(f"The Rsquared value is {round(rvalue**2,2)}.")
#Save the Plot as a PNG
plt.savefig("Infant_Mortality_vs_GDP_per_Capita_1990.png")
#Show the Plot
plt.show()
```

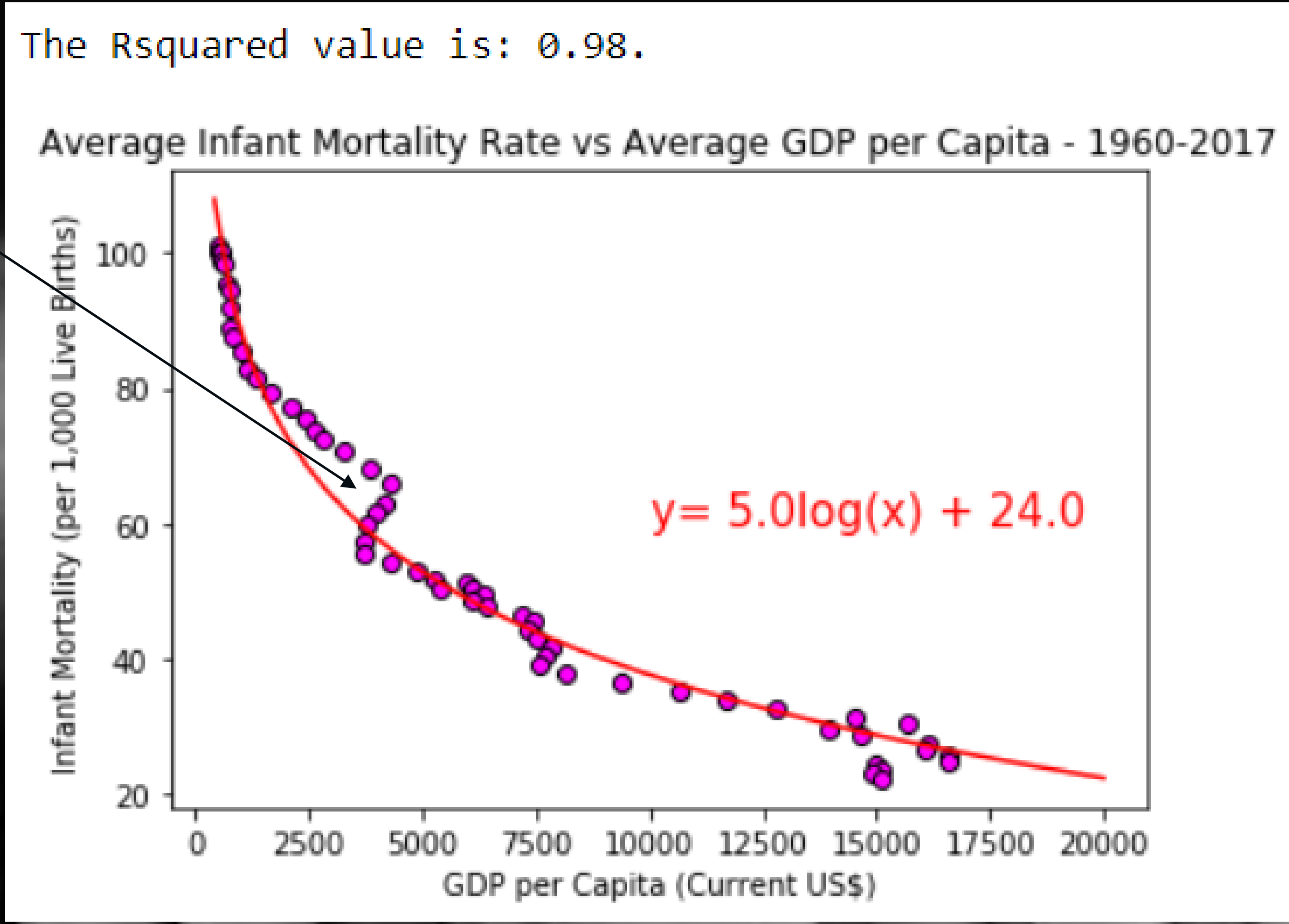
DATA ANALYSIS PROCESS – INFANT MORTALITY RATE (CONTINUED)





- The data showed that there wasn't a linear relationship between Infant Mortality Rate and GDP per Capita.
- It has been shown that there is a strong logarithmic relationship between Infant Mortality Rate and GDP per Capita.
- This means that Infant Mortality falls with GDP, but the rate of improvement slows as an economy advances.

DATA ANALYSIS PROCESS – INFANT MORTALITY RATE (CONTINUED)



It is unclear what is responsible for the dip in the data here.

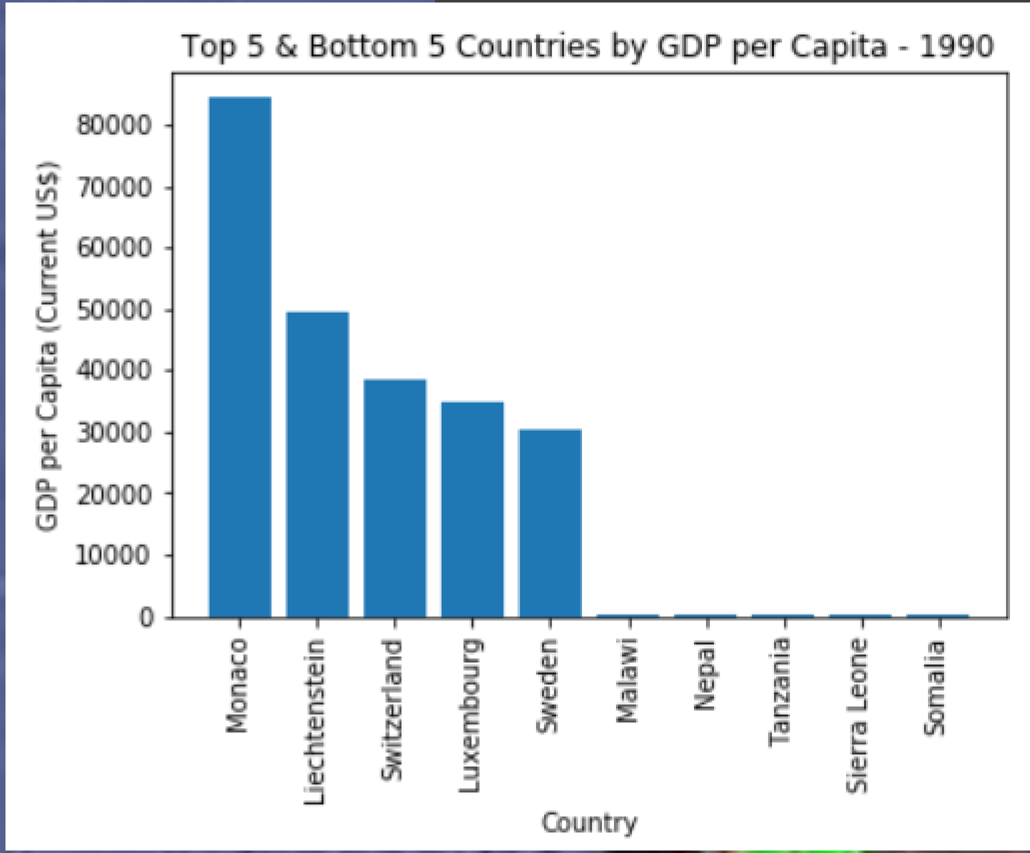
- To filter out the effects of the variation among countries, the plot of average Infant Mortality Rate vs average GDP per Capita was made.
- The mean values of individual year columns were calculated and put in lists.
- The results show that there is an extremely strong positive relationship between trade parameters and GDP per Capita.

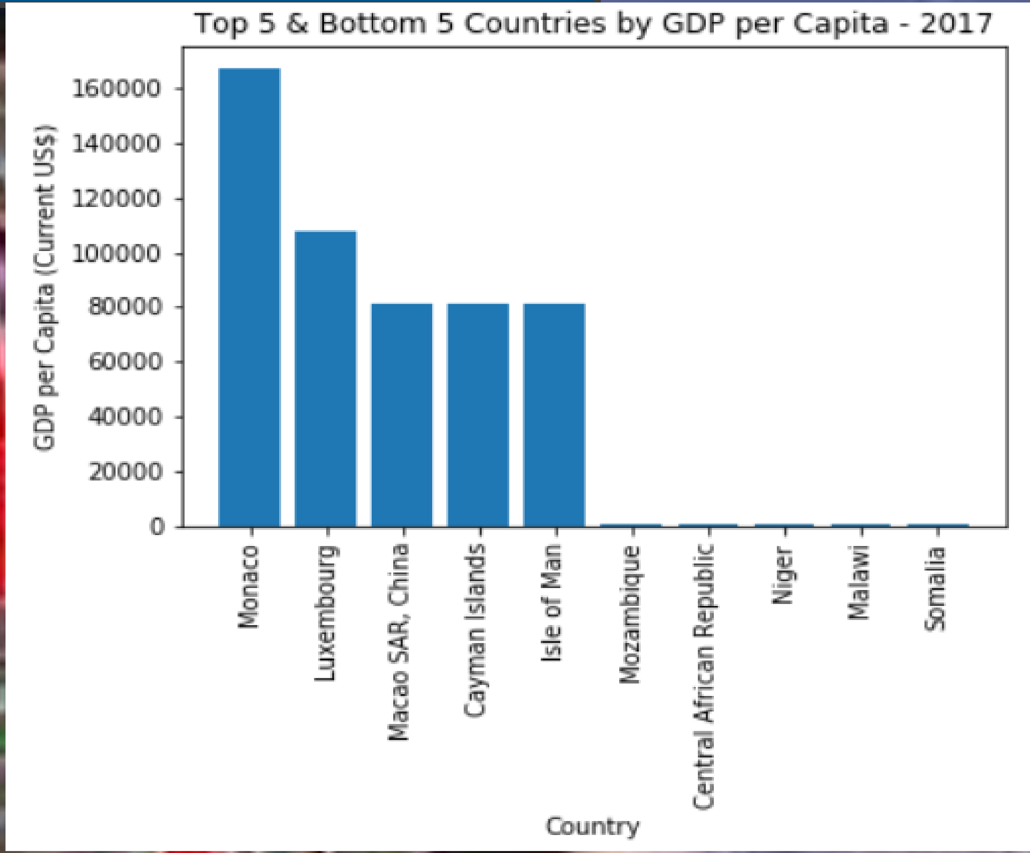
PREDICTIONS

PURPOSE

- We wanted to see the level of differences between the top 5 & bottom 5 countries in a couple of years.
- It was also desired to see whether any of the bottom 5 countries in one year “escaped” the bottom 5 and whether it was because of sharp increases in trade and/or whether it led to improvements in well-being values.

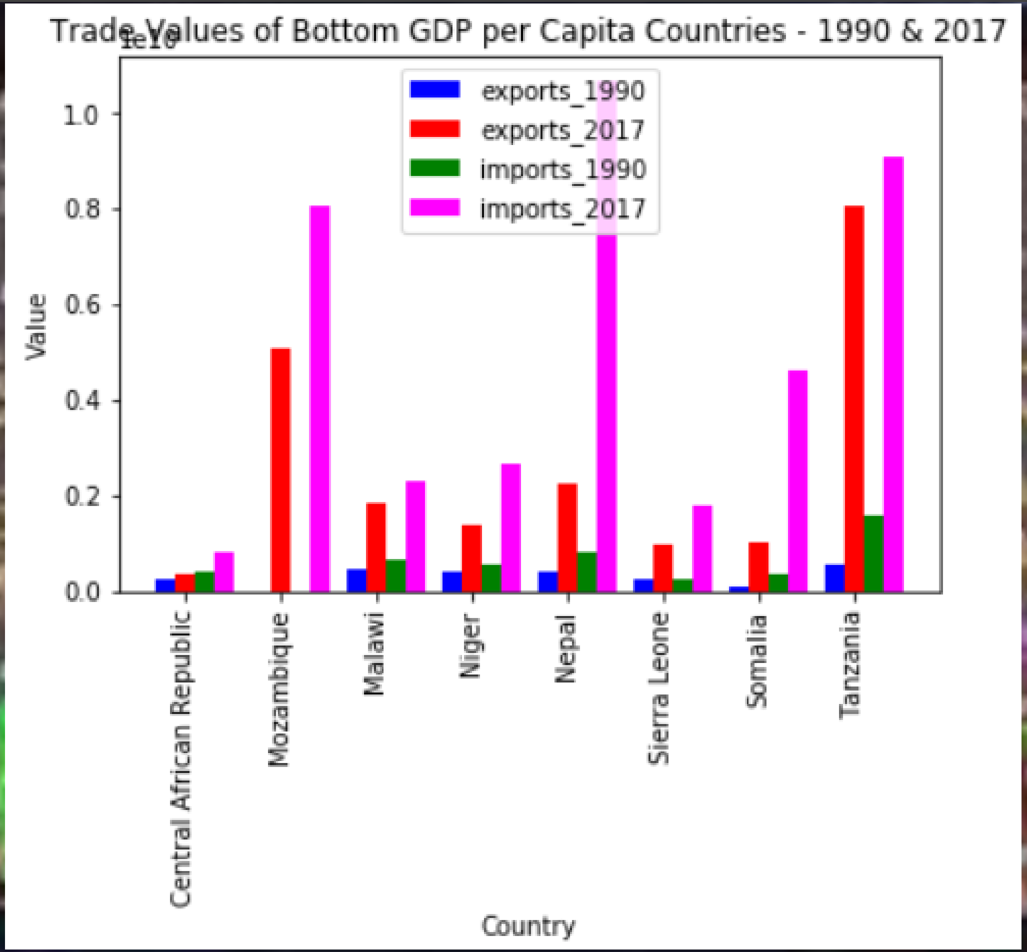
GDP PER CAPITA DIFFERENCES





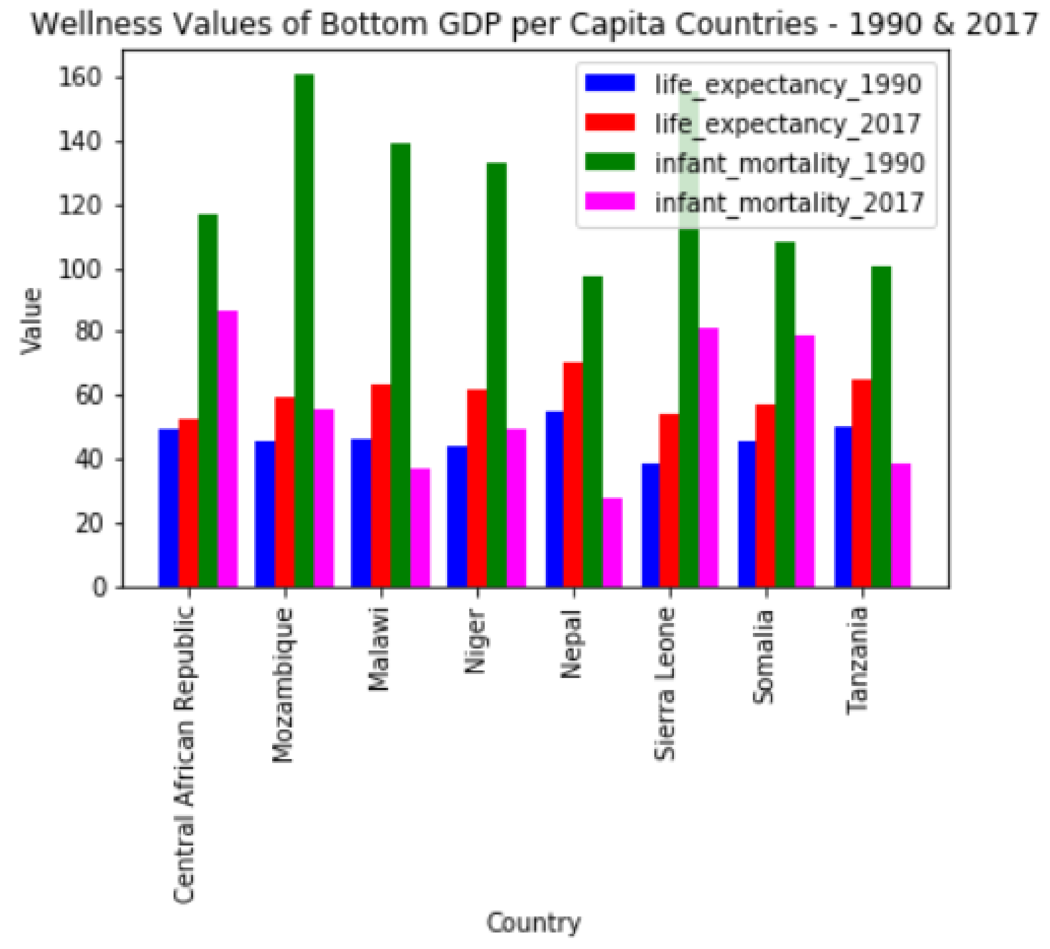
- There is a several order of magnitude difference in GDP per Capita between the top 5 & bottom 5 countries.
- The countries of Nepal, Tanzania, and Sierra Leone "escaped" the bottom 5.

EFFECT OF TRADE ON GDP PER CAPITA



- Of the countries of Nepal, Tanzania, and Sierra Leone that "escaped" the bottom 5, Nepal & Tanzania had the largest increases in trade, but Sierra Leone didn't.
- This would likely indicate that trade can improve the productivity of a country, but it is not a perfect predictor for individual countries between two separate years.

EFFECT OF GDP PER CAPITA ON WELL-BEING



- Of the countries of Nepal, Tanzania, and Sierra Leone that "escaped" the bottom 5, all had relatively large improvements in life expectancy & infant mortality rates, but not always the largest.
- This would likely indicate that productivity of a country can improve well-being, but it is not a perfect predictor for individual countries between two separate years.