CSCI 3022 Project Spring 2018

Prediction of World Population and Contributing Factors

Anthony Olvera

In [1]:

```
# Import all libraries with data manipulation and statistical utilities that may be needed
%matplotlib inline
import numpy as np
import pandas as pd;
import scipy as sp
import scipy.stats as stats
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.formula.api as smf
import statsmodels.api as sm
import patsy
```

```
C:\Users\Tony\AppData\Local\Programs\Python\Python36-32\lib\site-packages\statsmodels\compat\pandas.py:56: Futu
reWarning: The pandas.core.datetools module is deprecated and will be removed in a future version. Please use t
he pandas.tseries module instead.
  from pandas.core import datetools
```

Problem Statement

It is a known fact backed up by data that the world’s population is currently increasing at an exponential rate. Many people worry that this could be a contributing factor to many global scale problems such as pollution, climate change, shortage of natural resources etc. My hypothesis is that even though population is increasing at the moment certain limiting factors will instead constrain population growth in such a way that the increase will begin to decelerate and eventually plateau in the next few decades. Such a growth is called logistic growth. The main limiting factors on population growth can include food, water, hospitable living space, and ability to recycle waste. However two major constraints on a country's population growth are its Development Index and Gross Domestic Product. These two factors and their effects on population growth will be analysed using data sets from https://www.gapminder.org/ (https://www.gapminder.org/). In my analysis I will conduct a regression study in order to cast a future prediction on population growth.

Summary of Factors

The total_population dataset contains the total population of countries around the world from the year 1800 through 2015. The data was collected via census and will be compared against all of the following factors. The invidual factors I will be analyzing will be Human Development Index, and Gross Domestic Product per capita. Other datasets I have obtained include birth rate and population growth percentage. These data sets a more so abstractions of the total population data and I will use them to indentify correlations they may exist with the other factors.

```
In [2]: # A preview of the raw total_population data set
        pop = pd.read_csv('total_population.csv')
        pop.head()
```

Out[2]:

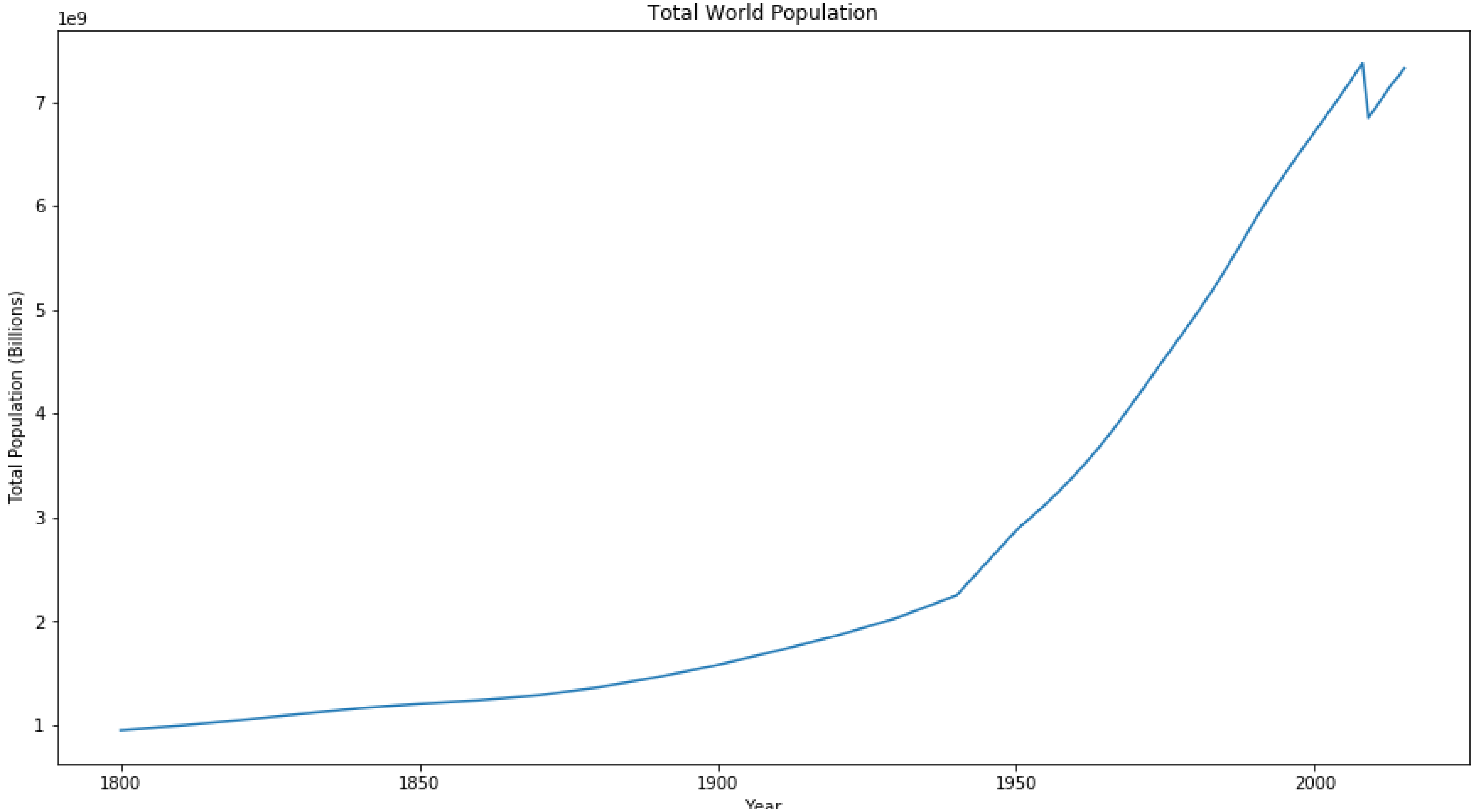
	Total population	1800	1810	1820	1830	1840	1850	1860	1870	1880	...	2006	2
0	Abkhazia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	
1	Afghanistan	3280000.0	3280000.0	3323519.0	3448982.0	3625022.0	3810047.0	3973968.0	4169690.0	4419695.0	...	25183615.0	258775
2	Akrotiri and Dhekelia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	15700.0	157
3	Albania	410445.0	423591.0	438671.0	457234.0	478227.0	506889.0	552800.0	610036.0	672544.0	...	3050741.0	30108
4	Algeria	2503218.0	2595056.0	2713079.0	2880355.0	3082721.0	3299305.0	3536468.0	3811028.0	4143163.0	...	33749328.0	342619

5 rows × 82 columns

Upon a visual analysis the following countries have no population data so those rows will be removed from the data frame. Abkhazia, Ngorno-Karabakh, Northern Cyprus, Somaliland, South Ossetia, Transnistria St. Martin (French part), Antarctica, Bouvet Island, British Indian Ocean Territory, Clipperton, French Southern and Antarctic Lands, Gaza Strip, Heard and McDonald Islands, Northern Marianas, South Georgia and the South Sandwich Islands, US Minor Outlying Islands, Virgin Islands, and West Bank.

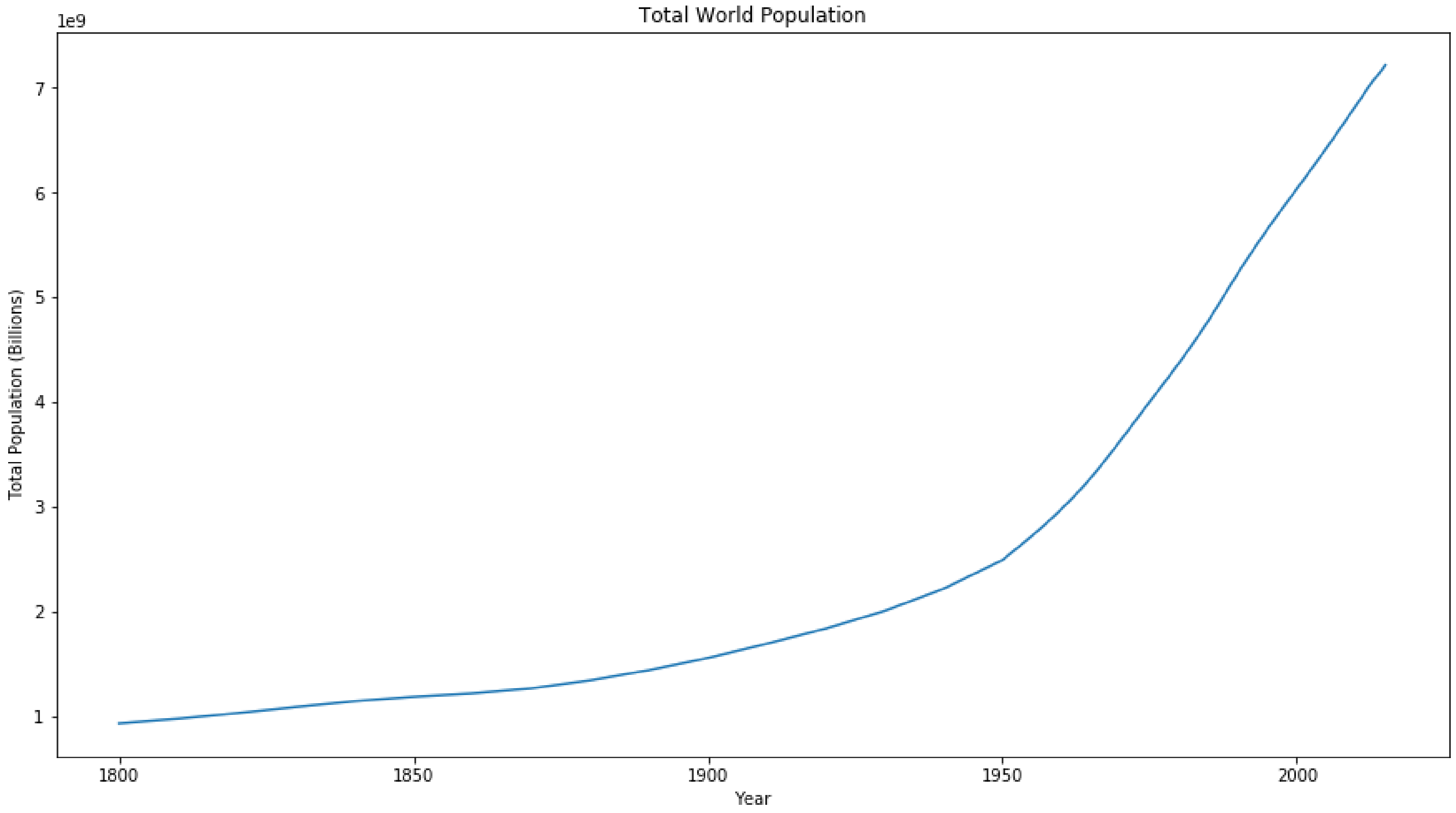
```
In [3]: # drop nan rows
        pop = pop.dropna(thresh = 2);

        # Convert to the proper data type.
        pop = pop.convert_objects(convert_numeric=True);

        # Sum over all countries and produce a plot of total population over time.
        years = [1800,1810,1820,1830,1840,1850,1860,1870,1880,1890,1900,1910,1920,1930,1940,1950,1951,1952,1953,1954,195
                 1958,1959,1960,1961,1962,1963,1964,1965,1966,1967,1968,1969,1970,1971,1972,1973,1974,1975,1976,1977,197
                 1981,1982,1983,1984,1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,200
                 2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015]

        total_pop = pop.sum(numeric_only=True);
        plt.figure(figsize=(15,8))
        plt.plot(years, total_pop);
        plt.title('Total World Population')
        plt.xlabel('Year')
        plt.ylabel('Total Population (Billions)');
```

```
C:\Users\Tony\AppData\Local\Programs\Python\Python36-32\lib\site-packages\ipykernel_launcher.py:5: FutureWarnin
g: convert_objects is deprecated.  To re-infer data dtypes for object columns, use DataFrame.infer_objects()
For all other conversions use the data-type specific converters pd.to_datetime, pd.to_timedelta and pd.to_numer
ic.
  """
```



It is clear that up to the year 2015 that the rate of population growth is still increasing. The dip at the year 2008 is because there is missing data for a number of countries after this year I will remove the following countries from the data in order to obtain a better curve. Akrotiri and Dhekelia, Albania, Czechoslovakia, East Germany, Eritrea and Ethiopia, Estonia, Guarnessy, Jersey, United Korea, Kosovo, St. Martain, Serbia and Montenegro, Serbia excluding Kosovo, Svalbard, Turkey, United Korea Former, USSR, Nort Yemen, South Yemen, West Germany and Yugoslavia.

```
In [4]: pop = pop.drop([2,3,58,63,69,70,91,111,116,118,119,191,200,201,216,232,242,248,252,253,255,258])
        total_pop = pop.sum(numeric_only=True);
        plt.figure(figsize=(15,8))
        plt.plot(years, total_pop);
        plt.title('Total World Population')
        plt.xlabel('Year')
        plt.ylabel('Total Population (Billions)');
```



Individual Factors

The first factor I'm expecting to have an effect on population growth is Human Development Index. This dataset came from The United

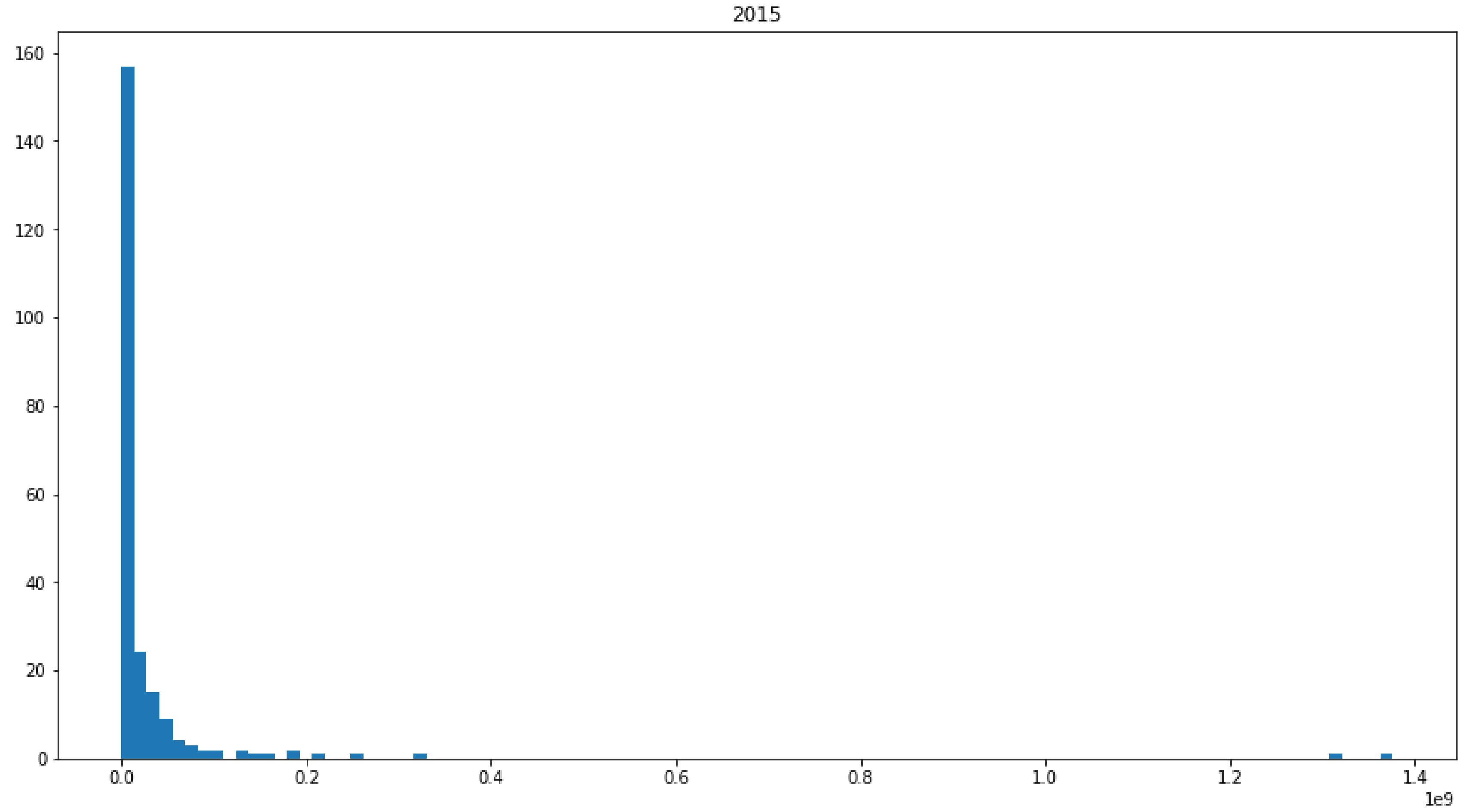
Nation Development Programme http://hdr.undp.org/en/data (http://hdr.undp.org/en/data) It contains the quantitative HDI indicator for many countries around the world from the year 1980. As HDI is a reletivley new statistc this data set is quite small similar to the GDP data set. In additon it does not contain data from every year up until the year 2005 so I will need to use interpolation to fill in the missing data. Like the other data sets it also contains a few NaN rows for various countries so it will require clever parsing and cleaning as well.

The second factor is Gross Domestic Product. I will use this data set as a measure of a country's industrial development. This dataset came from the world bank data base https://data.worldbank.org/indicator/NY.GDP.PCAP.KD.ZG (https://data.worldbank.org/indicator/NY.GDP.PCAP.KD.ZG). Im hypothesising that countries with higher GDP will have lower birth rates than countries with lower industrial development indicators, hence we should see a negative correlation. The data contains one categorical attribute (country) and two quantitative (GDP and year). This data set tends to be quite sparse and only dates back to 1960 so an obvious reduction will be done when analysing these factors.

Distribution Analysis

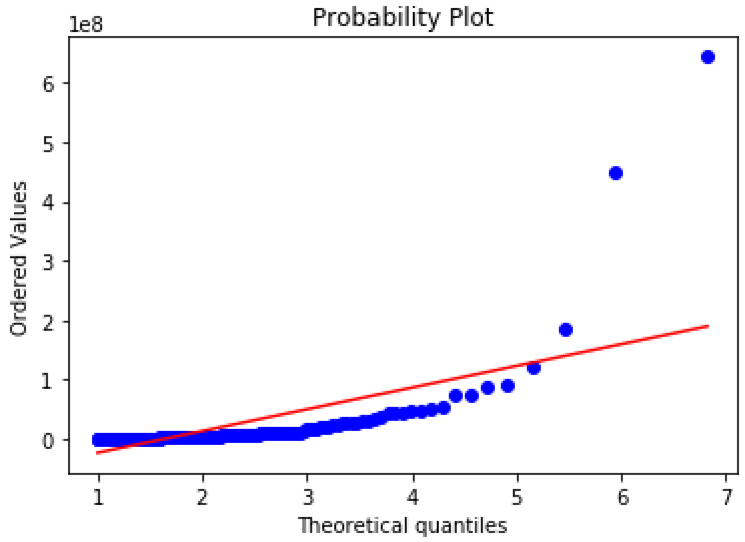
For any given year I would assume that population along with all factors that affect it will fit an exponetial distribution. I will plot a histogram and a quanitle-quantile plot for the year 2015 for the total population data.

```
In [5]: # Histogram of each country's population for the year 2015
        pop.dropna(subset = ['2015'])
        pop.hist(column = '2015', grid = False, figsize =(15,8), bins = 100);
```



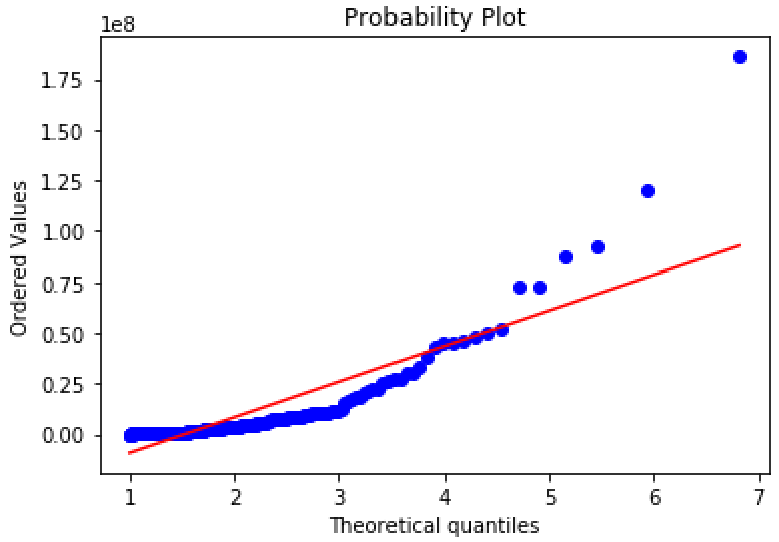
The data seems to be highley skewed to the right. Its dificult to see but there are two countries to the far right with very high populations.Im assuming these are China and india I will remove these outliers from the data.

In [6]:
```
stats.probplot(pop['1960'], dist = stats.expon(1), plot = plt);
```



With China and India removed.

In [7]:
```
pop = pop.drop([44, 101])
stats.probplot(pop['1960'], dist = stats.expon(1), plot = plt);
```



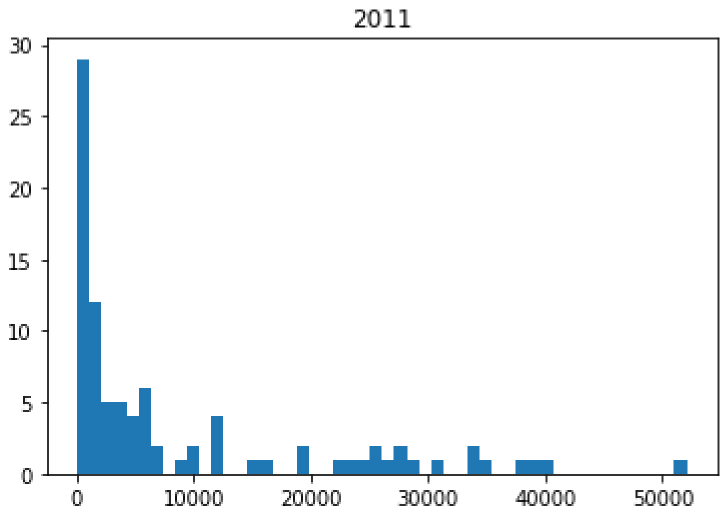
It's not a perfect fit however it is the closest probability distribution for the data.

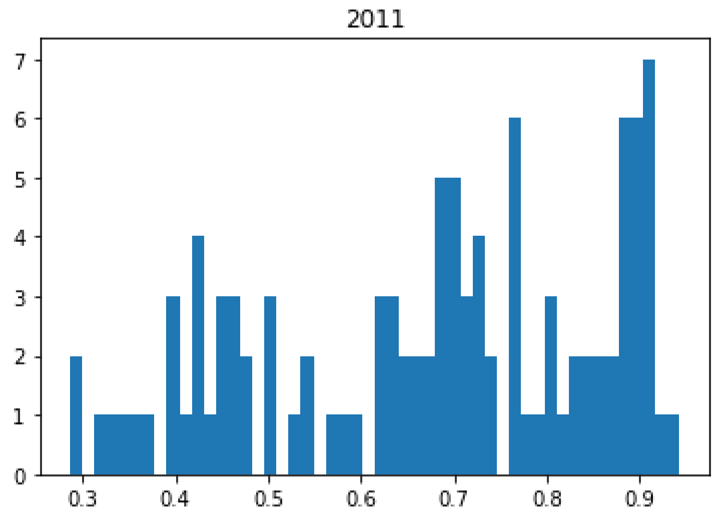
Next I will use histograms to analyse the distributions of my two individual factors HDI and GDP. I suspect to see the data normally distributed across each country.

In [8]:

```
# Read the data
hdi = pd.read_csv('HDI_indicator.csv')
# Drop NaN rows
hdi = hdi.dropna(thresh=10)
# Convert To proper data type
hdi = hdi.convert_objects(convert_numeric=True);
# Reindex
hdi = hdi.set_index('Country')
gdp = pd.read_csv('GDP_percapita.csv')
gdp = gdp.dropna(thresh=53)
gdp = gdp.convert_objects(convert_numeric = True);
gdp = gdp.set_index('Country')
# Plot Histograms
gdp.hist(column = '2011', grid = False, bins = 50);
hdi.hist(column = '2011', grid = False, bins = 50);
```

```
C:\Users\Tony\AppData\Local\Programs\Python\Python36-32\lib\site-packages\ipykernel_launcher.py:6: FutureWarnin
g: convert_objects is deprecated.  To re-infer data dtypes for object columns, use DataFrame.infer_objects()
For all other conversions use the data-type specific converters pd.to_datetime, pd.to_timedelta and pd.to_numer
ic.

C:\Users\Tony\AppData\Local\Programs\Python\Python36-32\lib\site-packages\ipykernel_launcher.py:11: FutureWarni
ng: convert_objects is deprecated.  To re-infer data dtypes for object columns, use DataFrame.infer_objects()
For all other conversions use the data-type specific converters pd.to_datetime, pd.to_timedelta and pd.to_numer
ic.
  # This is added back by InteractiveShellApp.init_path()
```





To my surprize GDP for each country appears to also fit an exponential distribution in the year 2011, however HDI appears to be random, not fitting any specific distribution.

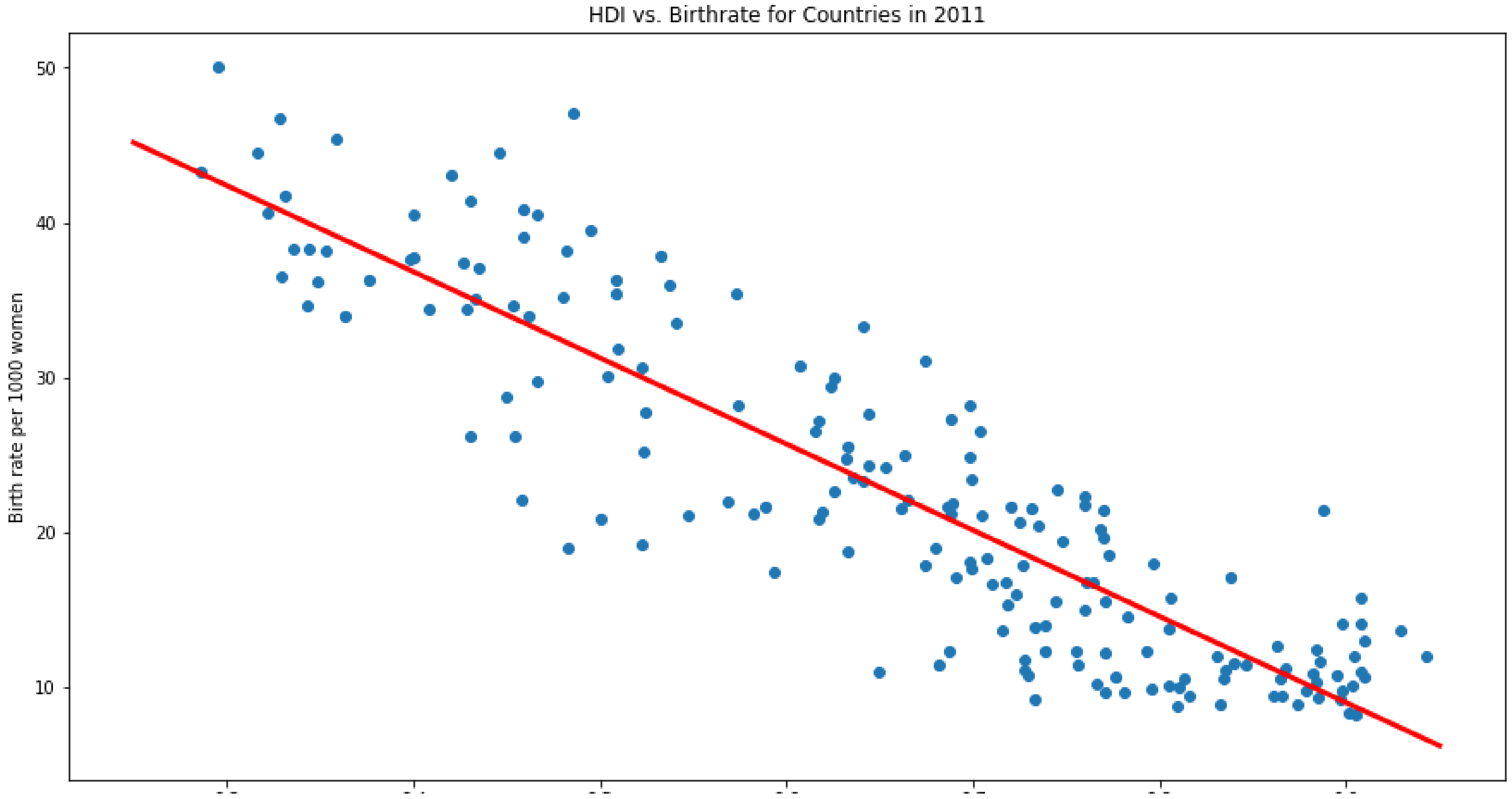
Correlation

The strongest correlation I would expect to see first is between HDI and birth rate. I would expect developing countries to have the highest birth rates therefore we should see a negative correlation between HDI and birthrate. Below I concatanate the birthrate and the HDI datasets in order to more easily model the data. I will create a scatter plot with a linear regression model to fit the data. Note the year 2010 is missing from the dataset.

In [9]:

```
# Merge HDI with Birthrate
xticks = np.linspace(0.25,0.95,10)
hdi = pd.read_csv('HDI_indicator.csv')
brate = pd.read_csv('birth_rate.csv')
brate=brate[['Country','1980','1990','2000','2005', '2006','2007', '2008', '2009', '2011']]
hdi = hdi.convert_objects(convert_numeric=True);
brate = brate.convert_objects(convert_numeric=True);
hdi = hdi.set_index('Country')
brate = brate.set_index('Country')
joined  = pd.concat([hdi,brate], axis =1, join = 'inner')
joined.columns = [ '1980h','1990h','2000h','2005h', '2006h','2007h', '2008h', '2009h',
               'h2011','1980b','1990b','2000b','2005b', '2006b','2007b', '2008b', '2009b', 'b2011']

# Create the linear Regression Model
lmodel = smf.ols(formula = 'b2011~h2011', data = joined).fit()
beta0,beta1 = lmodel.params

# Plot the model to overlay the scatter plot
plt.figure(figsize=(15,8))
plt.scatter(x= joined['h2011'], y = joined['b2011']);
plt.plot(xticks, beta0 + beta1*xticks, lw =3, c = 'r')
plt.title('HDI vs. Birthrate for Countries in 2011');
plt.xlabel('Human Development Index');
plt.ylabel('Birth rate per 1000 women');
```

```
C:\Users\Tony\AppData\Local\Programs\Python\Python36-32\lib\site-packages\ipykernel_launcher.py:6: FutureWarnin
g: convert_objects is deprecated.  To re-infer data dtypes for object columns, use DataFrame.infer_objects()
For all other conversions use the data-type specific converters pd.to_datetime, pd.to_timedelta and pd.to_numer
ic.

C:\Users\Tony\AppData\Local\Programs\Python\Python36-32\lib\site-packages\ipykernel_launcher.py:7: FutureWarnin
g: convert_objects is deprecated.  To re-infer data dtypes for object columns, use DataFrame.infer_objects()
For all other conversions use the data-type specific converters pd.to_datetime, pd.to_timedelta and pd.to_numer
ic.
  import sys
```



There is a clear negative correlation as expected. The data is fit with a linear regression model given by the following summary.

In [10]:
```
lmodel.summary()
```

Out[10]:

OLS Regression Results

Dep. Variable:	b2011	**R-squared:**	0.792
Model:	OLS	**Adj. R-squared:**	0.791
Method:	Least Squares	**F-statistic:**	684.6
Date:	Fri, 04 May 2018	**Prob (F-statistic):**	3.06e-63
Time:	14:51:52	**Log-Likelihood:**	-547.66
No. Observations:	182	**AIC:**	1099.
Df Residuals:	180	**BIC:**	1106.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	59.0621	1.448	40.795	0.000	56.205	61.919
h2011	-55.5912	2.125	-26.165	0.000	-59.784	-51.399

Omnibus:	0.108	**Durbin-Watson:**	2.093
Prob(Omnibus):	0.948	**Jarque-Bera (JB):**	0.026
Skew:	-0.029	**Prob(JB):**	0.987
Kurtosis:	3.011	**Cond. No.**	8.39

And the following formula.

$$Y \sim \beta_0 + \beta_1 X_1$$

$$Birthrate \sim \beta_0 + \beta_1 HDI$$

In [11]:
```
print('Our Model is:  Birthrate = ', beta0, '+', beta1, '*HDI')
```

```
Our Model is:  Birthrate =  59.06213578300286 + -55.591164270887646 *HDI
```

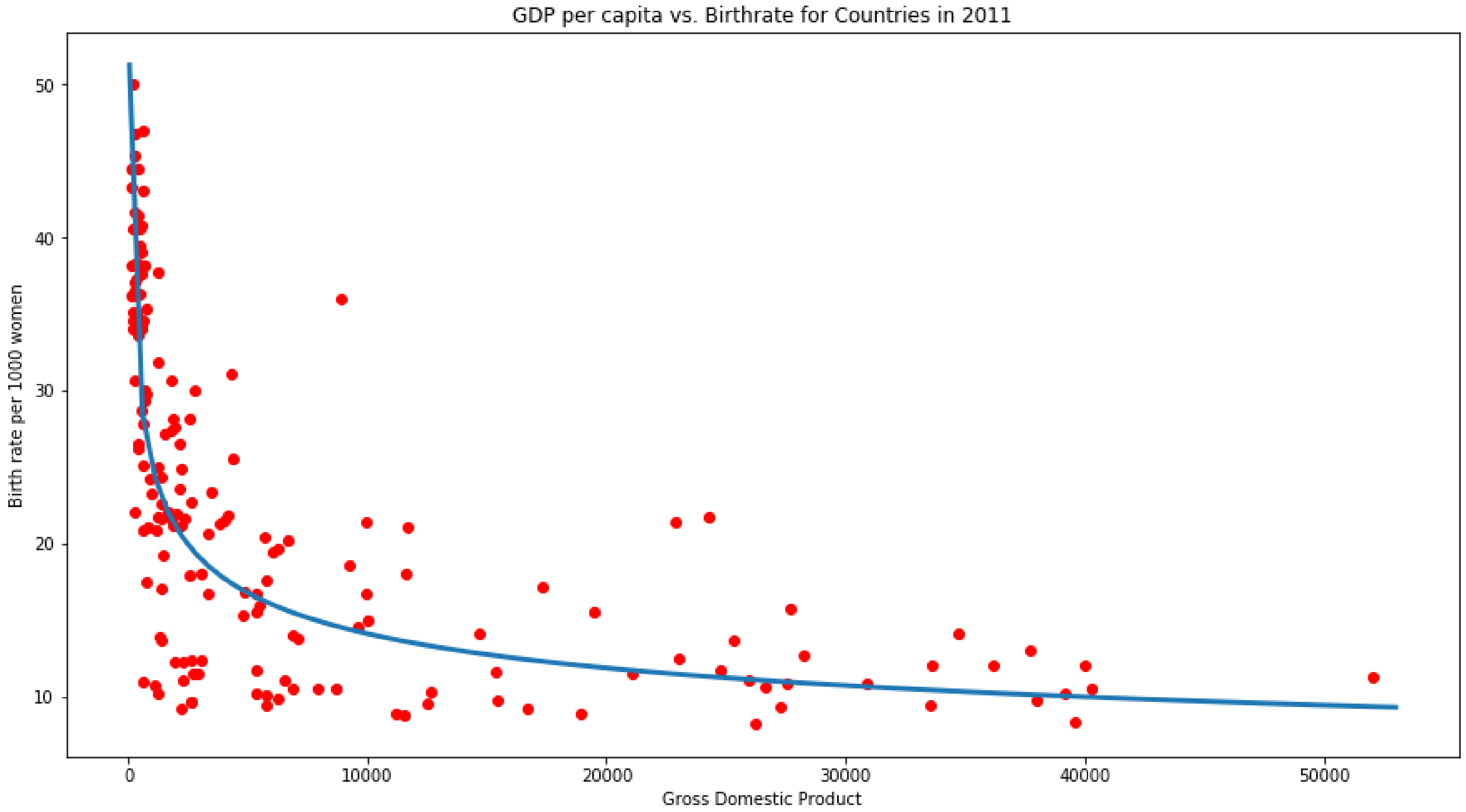
I will do the same for Gross domestic product however I dont expect as strong of a correlation. I will use a non linear model to fit the data

In [12]:

```
# Merge GDP and Birthrate
xticks = np.linspace(-1035,53000,100)
gdp = pd.read_csv('GDP_percapita.csv')
brate = pd.read_csv('birth_rate.csv')
brate.drop(brate.columns.to_series()["1800":"1959"], axis=1)
brate.drop(brate.columns.to_series()["2012":], axis=1)
gdp = gdp.convert_objects(convert_numeric = True);
gdp.columns = ['g' + str(col) for col in gdp.columns]
gdp.rename(columns={'gCountry':'Country'}, inplace=True)
brate = brate.convert_objects(convert_numeric = True);
brate.columns = ['b' + str(col) for col in brate.columns]
brate.rename(columns={'bCountry':'Country'}, inplace=True)
gdp = gdp.set_index('Country')
brate = brate.set_index('Country')
joined1  = pd.concat([gdp,brate], axis =1, join = 'inner')

# Quadratic model R-squared = 0.424
qmodel = smf.ols('b2011~g2011 + np.power(g2011, 2)', joined1).fit()

# Log-log model, best fit, R-squared = 0.615
lmodel = smf.ols('np.log(b2011) ~ np.log(g2011)', joined1).fit()

# Plot the data and the model
plt.figure(figsize=(15,8))
plt.scatter(x= joined1['g2011'], y = joined1['b2011'], c = 'r');
plt.plot(xticks,  np.exp(lmodel.params[0] + np.log(xticks) * lmodel.params[1]),  lw=3);
plt.title('GDP per capita vs. Birthrate for Countries in 2011');
plt.xlabel('Gross Domestic Product');
plt.ylabel('Birth rate per 1000 women');
```

```
C:\Users\Tony\AppData\Local\Programs\Python\Python36-32\lib\site-packages\ipykernel_launcher.py:7: FutureWar
ning: convert_objects is deprecated.  To re-infer data dtypes for object columns, use DataFrame.infer_object
s()
For all other conversions use the data-type specific converters pd.to_datetime, pd.to_timedelta and pd.to_nu
meric.
  import sys
C:\Users\Tony\AppData\Local\Programs\Python\Python36-32\lib\site-packages\ipykernel_launcher.py:10: FutureWa
rning: convert_objects is deprecated.  To re-infer data dtypes for object columns, use DataFrame.infer_objec
ts()
For all other conversions use the data-type specific converters pd.to_datetime, pd.to_timedelta and pd.to_nu
meric.
  # Remove the CWD from sys.path while we load stuff.
```

```
C:\Users\Tony\AppData\Local\Programs\Python\Python36-32\lib\site-packages\ipykernel_launcher.py:26: RuntimeW
arning: invalid value encountered in log
```



This correlation doesent appear to be linear but it does expose an evident trend. Ive found a log-log model to best fit the data. Below is the model summary and eqaution using the corelation coefficents for the data.

In [13]:
```
lmodel.summary()
```

Out[13]:

OLS Regression Results

Dep. Variable:	np.log(b2011)	**R-squared:**	0.615
Model:	OLS	**Adj. R-squared:**	0.613
Method:	Least Squares	**F-statistic:**	266.9
Date:	Fri, 04 May 2018	**Prob (F-statistic):**	1.86e-36
Time:	14:51:53	**Log-Likelihood:**	-42.906
No. Observations:	169	**AIC:**	89.81
Df Residuals:	167	**BIC:**	96.07
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	4.9424	0.122	40.409	0.000	4.701	5.184
np.log(g2011)	-0.2490	0.015	-16.337	0.000	-0.279	-0.219

Omnibus:	9.640	**Durbin-Watson:**	2.195
Prob(Omnibus):	0.008	**Jarque-Bera (JB):**	9.613
Skew:	-0.545	**Prob(JB):**	0.00818
Kurtosis:	3.422	**Cond. No.**	41.3

$$Birthrate = \exp(\beta_0 + \beta_1 \log(GDP)) = \exp(\beta_0) * \exp(\beta_1 \log(GDP))$$

In [14]:
```
print('Our Model is:  Birthrate = exp(',lmodel.params[0], ') * exp(',lmodel.params[1] ,'* log(GDP))')
```

```
Our Model is:  Birthrate = exp( 4.9424321269644045 ) * exp( -0.24904932291475362 * log(GDP))
```

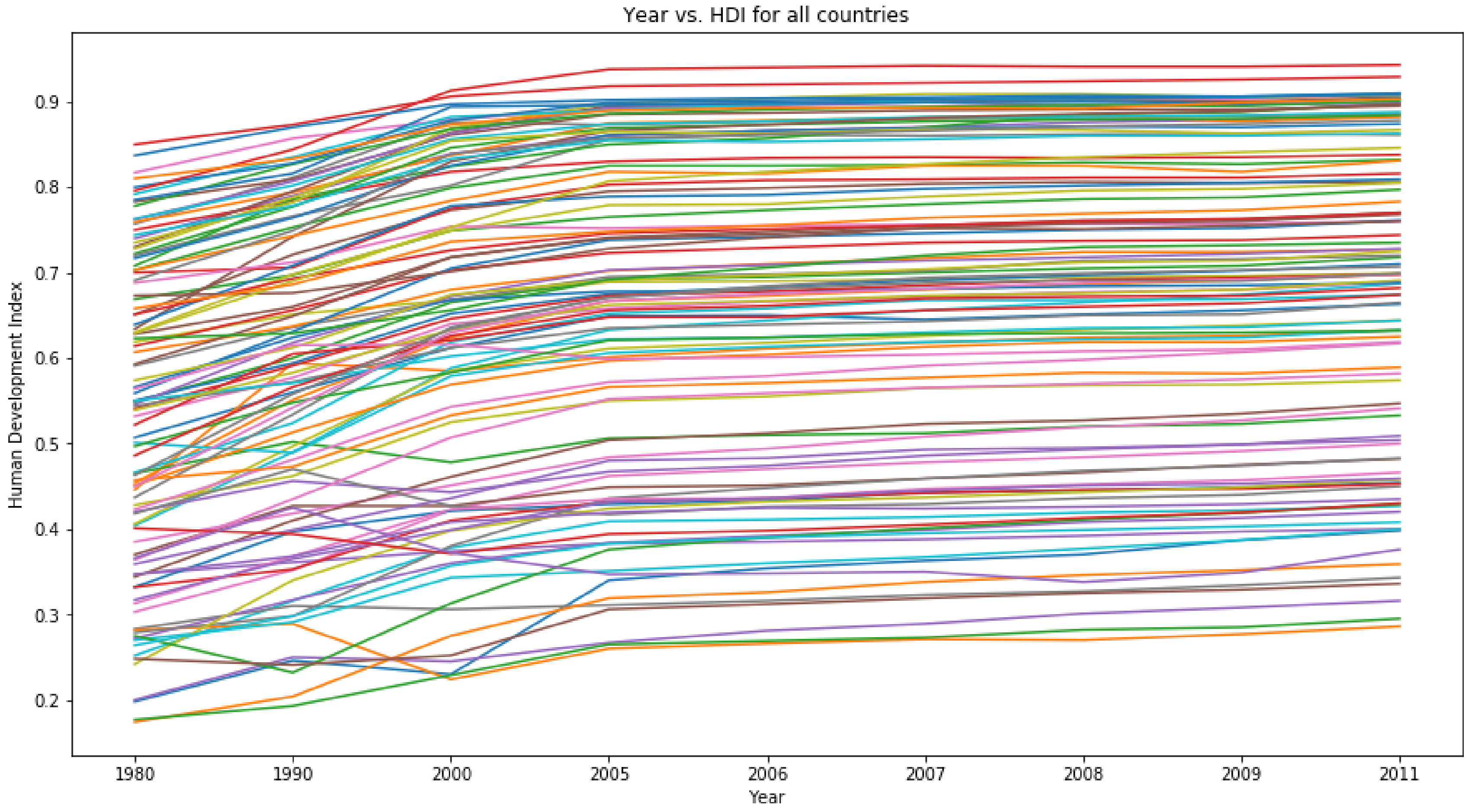
Seeing as we have negative correlations between birthrate and human development Index in addition to birthrate with gross domestic product this would mean that if the gross domestic product of developing countries is increasing, it would be reasonable to belive the populations are stabalizing as well.

```
In [15]: # Clean the data by slicing all rows containing any NaN values.
         hdi = hdi.dropna(thresh=9)
```

Upon visual inspection of the data it appears that for nearley all countries HDI is increasing as the years pass. Below I will produce a plot of all countries HDI from 1980 to 2011.

In [16]:

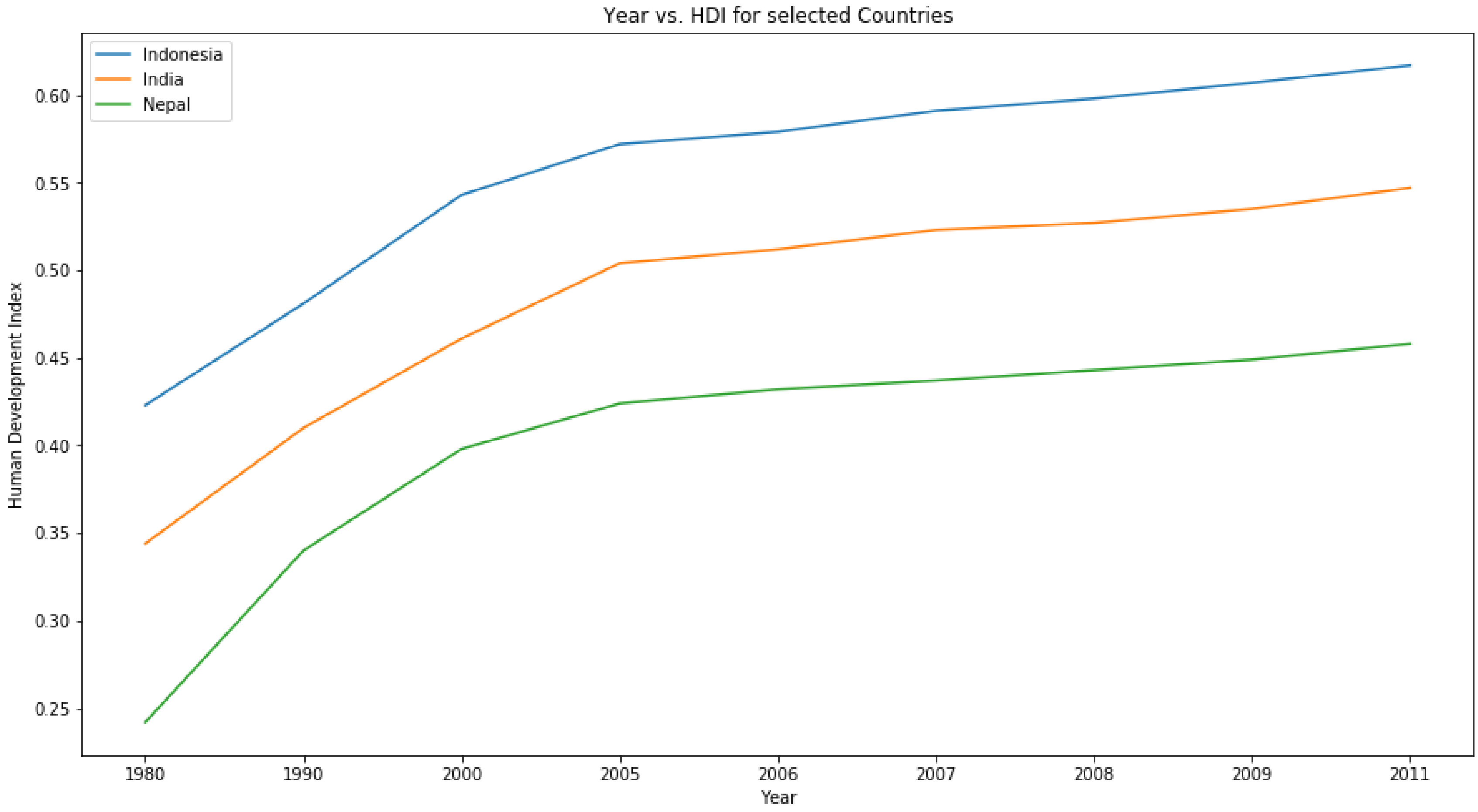
```
plt.figure(figsize=(15,8))
plt.title('Year vs. HDI for all countries')
plt.xlabel('Year')
plt.ylabel('Human Development Index')

# Plot the curve for each country in the dataframe
for i in range(len(hdi)):
    plt.plot(hdi.iloc[i])
```



For the most part, an upward trend can be seen for most countries, however the plot is slightley cluttered, In addition the plateau is due to the data points being more sparsely distributed in the earlier years. In reality we would see a linear increase. Below I will plot HDI only for a few countries with a low initial value. For these I've choosen Indonesia, India and Nepal.

In [17]:

```
plt.figure(figsize=(15,8))

# Produce plot for above three countries
plt.plot(hdi.loc['Indonesia']);
plt.plot(hdi.loc['India']);
plt.plot(hdi.loc['Nepal']);
plt.title('Year vs. HDI for selected Countries')
plt.xlabel('Year')
plt.ylabel('Human Development Index');
plt.legend();
```

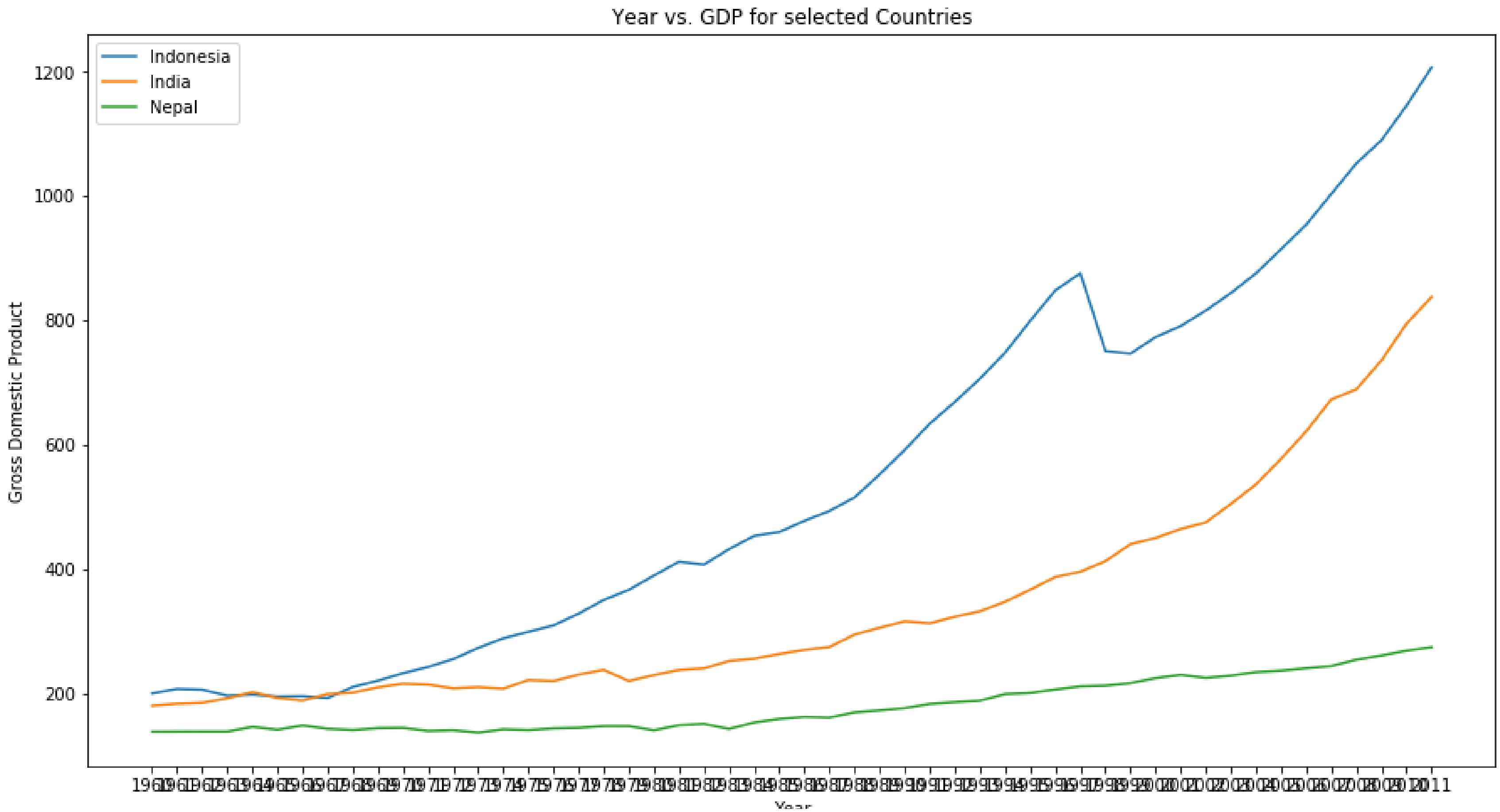


Now the GDP for the same three countries

In [18]:

```
# Re-read and clean the data
gdp = pd.read_csv('GDP_percapita.csv')
gdp = gdp.dropna(thresh=53)
gdp = gdp.convert_objects(convert_numeric = True);
gdp = gdp.set_index('Country')

# Produce plot
plt.figure(figsize=(15,8))
plt.plot(gdp.loc['Indonesia']);
plt.plot(gdp.loc['India']);
plt.plot(gdp.loc['Nepal']);
plt.title('Year vs. GDP for selected Countries')
plt.xlabel('Year')
plt.ylabel('Gross Domestic Product');
plt.legend();
```

```
C:\Users\Tony\AppData\Local\Programs\Python\Python36-32\lib\site-packages\ipykernel_launcher.py:4: FutureWarnin
g: convert_objects is deprecated.  To re-infer data dtypes for object columns, use DataFrame.infer_objects()
For all other conversions use the data-type specific converters pd.to_datetime, pd.to_timedelta and pd.to_numer
ic.
  after removing the cwd from sys.path.
```



Conclusion

We've known for a fact now that the world population has been increasing since the early 18th century, In addition we also know that at present the rate of population growth is also increasing. However two major factors, Human Development Index and Gross Domestic Product, have shown to be correlated with these trends. Because most countries are still in the early stages of development we continue to see world population grow. However in this analysis weve seen that as there are obvious negative correlations between birthrate and and HDI as well as birthrate and GDP, and furthermore GDP and HDI are increasing for most countries around the world. Therefore weve reached the conclusion that, if these trends continue, then world population growth must eventually slow and eventually stabilize in the comming years.