Investigate_a_Dataset

January 30, 2018

1 Project Title : Investigating the Correlation and Influence of Democracy Score On Country GDP Per Capita, Food Consumption and Life Expectancy.

1.1 Table of Contents

Introduction
Data Wrangling
Exploratory Data Analysis
Conclusions
Introduction

For this project the dataset selected for investigation is the **Gapminder Dataset** which contain various type of global data on a number of different indicators , tracked across the years. In this project, **Democracy Score** is anchored as the dependent variable against three independent variables namely Life Expectancy at Birth ,GDP per Capita and Food Consumption .The assumption undertaken to classify as well as the choice for dependent and independent variables are soley a matter of preference of the Data Analyst .

The data is first downloaded as an excel file and then converted to csv file before uploading it to the Python Jupyter Notebook .To get a good scope and also to ensure the cleaning and exploration is performed appropraitely with in the available time ,the data sets of two countries namely *United States* and *Ethiopia* ,and the *Global mean* of the varaiables selected , are used for exploration and analysis. In addition to the time constraint ,the time frame for this analysis is also limited to approximately to the past 30 years(beginning from 1980),inorder that the most recent trends, changes and correlations could be investigated among the variables .

The questions posed for this investigation are as follows :-

1: How was Democracy Score of the two Countries over the years? How about as compared to the World ?

2: Are there any noticeable trends of the independent variables , as Democracy Score changes over the years ?

3: What kind of correlations and influences observed between dependent and independent Variables ?

In [35]:
```
# Importing the necessary Packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
% matplotlib inline
```

Data Wrangling

The activities planned and performed in this section include loading the selected data sets as csv file , checking for cleanliness, and then triming and cleaning the dataset for analysis.

1.1.1 General Properties

In [36]:
```
# importing the data  for Democracy score,Food consumption,GDP per capita,Life expentan
df_demo= pd .read_csv('democracy_score.csv')
df_food= pd .read_csv('food_consumption.csv')
df_GDP= pd .read_csv('GDPpercapita_with_projections.csv')
df_life= pd .read_csv('life_expectancy_at_birth.csv')
```

In [37]:
```
#Examining the data content
df_demo.head(1)
df_demo.tail()
```

Out[37]:

	Democracy, based on PolityIV	1800	1801	1802	1803	1804	1805	1806	\
270	Two Sicilies	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
271	United Province CA	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
272	Vietnam North	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
273	Vietnam South	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
274	Wuerttemburg	-7.0	-7.0	-7.0	-7.0	-7.0	-7.0	-7.0	

	1807	1808	...	2002	2003	2004	2005	2006	2007	2008	2009	2010	\
270	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
271	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
272	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
273	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
274	-7.0	-7.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

	2011
270	NaN
271	NaN
272	NaN
273	NaN
274	NaN

[5 rows x 213 columns]

```
In [38]: #exploring Food Consumption data
         df_food.head(1)
```

```
Out[38]:   Unnamed: 0  1961  1962  1963  1964  1965  1966  1967  1968  1969  ...   \
         0   Abkhazia   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN  ...

            1998  1999  2000  2001  2002  2003  2004  2005  2006  2007
         0   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN

         [1 rows x 48 columns]
```

```
In [39]: #exploring GDP Per Capita data
         df_GDP.head(1)
```

```
Out[39]:   GDP per capita PPP, with projections  1764  1765  1766  1767  1768  1769  \
         0                             Abkhazia   NaN   NaN   NaN   NaN   NaN   NaN

            1770  1771  1772  ...   2009  2010  2011  2012  2013  2014  2015  2016  \
         0   NaN   NaN   NaN  ...    NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN

            2017  2018
         0   NaN   NaN

         [1 rows x 256 columns]
```

```
In [40]: #Exploring Life Expectancy at Birth data
         df_life.head(1)
```

```
Out[40]:   Life expectancy  1800  1801  1802  1803  1804  1805  1806  1807  1808  ...   \
         0        Abkhazia   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN  ...

            2007  2008  2009  2010  2011  2012  2013  2014  2015  2016
         0   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN

         [1 rows x 218 columns]
```

```
In [41]: df_demo.shape
         #examining the shape of the table .
         df_demo.info()
         #info on data type ,shape , data size
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 275 entries, 0 to 274
Columns: 213 entries, Democracy, based on PolityIV to 2011
dtypes: float64(212), object(1)
memory usage: 457.7+ KB
```

```
In [42]: df_demo.describe ()
```

Out[42]:

	1800	1801	1802	1803	1804	1805	\
count	22.000000	22.000000	22.000000	22.000000	22.000000	22.000000	
mean	-7.363636	-7.363636	-7.363636	-7.363636	-7.363636	-7.363636	
std	3.910248	3.910248	3.910248	3.910248	3.910248	3.910248	
min	-10.000000	-10.000000	-10.000000	-10.000000	-10.000000	-10.000000	
25%	-10.000000	-10.000000	-10.000000	-10.000000	-10.000000	-10.000000	
50%	-10.000000	-10.000000	-10.000000	-10.000000	-10.000000	-10.000000	
75%	-6.000000	-6.000000	-6.000000	-6.000000	-6.000000	-6.000000	
max	4.000000	4.000000	4.000000	4.000000	4.000000	4.000000	

	1806	1807	1808	1809	...	2002	\
count	22.000000	22.000000	22.000000	22.000000	...	163.000000	
mean	-7.363636	-7.363636	-7.363636	-7.090909	...	3.276074	
std	3.910248	3.910248	3.910248	4.648698	...	6.534135	
min	-10.000000	-10.000000	-10.000000	-10.000000	...	-10.000000	
25%	-10.000000	-10.000000	-10.000000	-10.000000	...	-2.000000	
50%	-10.000000	-10.000000	-10.000000	-9.500000	...	6.000000	
75%	-6.000000	-6.000000	-6.000000	-6.000000	...	9.000000	
max	4.000000	4.000000	4.000000	9.000000	...	10.000000	

	2003	2004	2005	2006	2007	2008	\
count	163.000000	163.000000	163.000000	163.000000	162.000000	163.000000	
mean	3.233129	3.361963	3.619632	3.680982	3.660494	3.785276	
std	6.508845	6.570437	6.450389	6.463292	6.425071	6.375825	
min	-10.000000	-10.000000	-10.000000	-10.000000	-10.000000	-10.000000	
25%	-2.500000	-2.500000	-2.000000	-2.500000	-2.000000	-2.000000	
50%	6.000000	6.000000	6.000000	7.000000	6.500000	7.000000	
75%	9.000000	9.000000	9.000000	9.000000	9.000000	9.000000	
max	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	

	2009	2010	2011
count	163.000000	163.000000	163.000000
mean	3.785276	3.883436	4.036810
std	6.307689	6.234504	6.159294
min	-10.000000	-10.000000	-10.000000
25%	-2.000000	-1.500000	-1.000000
50%	7.000000	6.000000	7.000000
75%	9.000000	9.000000	9.000000
max	10.000000	10.000000	10.000000

[8 rows x 212 columns]

Findings:

The entries for Democracy Score are integers in the range of -10 to 10 .

Checking how the columns and rows are arrenged :275 rows and 213 Columns.

Null values are prevalent.However ,since the aim of this analysis is foused in country specific , the mean world values ,and the period starting from 1980,the null values assumed to not have

any effect on the result .

Data for all variables is within the period from 1800- 2011 .

212 columns have float64 data type and 1 column string object entries .

The 1st columns of the table doesn't have the correct name.The more appropraite one would be "Country" rather than the varaible name .

** The same procedure is followed to examin the general properties of the other 3 variables Data Sets.**

** The following steps are planned to tackle the above mentioned issues identified in the DataSets .**

1:Drop columns found before 1980 and change the 1st column header into "Country" and then assign it as index for the dataframe.

2: Separet or Extract the data for the selected countries (America and Ethiopia) and the World mean. Arrenge a new dataframe for ech data in such a way that the year and variable values will be set as a distinct columns

3: Change 1st and 2nd column names for the extracted data_frame to "Year"and "Variable Name".

4: After the data quality and structure of the variable dataframes are found satisfactory ,outer join the datasets into their respective catagory using Year as index.

5: Finally , Merge all catagories into a single dataframe for the sake of simplicity ,for an easy analysis and beter preservation and data storage usage.

1.1.2 Data Cleaning and Trimming Procedures

```
In [43]: # Drop columns in the n_years range
         #Changes the 1st  column header into country and then place it as index


         def drop_column(filename,n_years):
             """ drops columns idexes in the n_year range,
             renames 1st column into Country  and set it as index """
            # Create list of index
             ranges=[]
             for i in range (n_years):
                 ranges.append(i+1)
                 # drops with columns gaving an index in the n_years range
                 df_drop=filename.drop(filename.columns[ranges], axis=1)
                 # renames the first column into country
                 df_drop.rename(columns={df_drop.columns[0]:'Country'}, inplace=True )
                 # set country as index
                 df=df_drop.set_index('Country')
             return df

         # Extract the data for each variable begining from the year 1980 .
```

```
df_demo_cl1=drop_column(df_demo,179)
df_food_cl1=drop_column(df_food,18)
df_GDP_cl1=drop_column(df_GDP,215)
df_life_cl1=drop_column(df_life,179)

df_food_cl1.head(1)
```

Out[43]:

Country	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	...	\
Abkhazia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	

Country	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007
Abkhazia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

[1 rows x 29 columns]

In [44]:
```
# Extract  data for a given country

def data_extract (filename,Country):
    """Given the country name ,extracts the country data from the file """
    df=filename.loc[Country]
    return df

#Extract and saves the cleaned  Democracy Score data
(data_extract (df_demo_cl1,'United States')).to_csv('df_clnd_demo_US')
(data_extract (df_demo_cl1,'Ethiopia')).to_csv('df_clnd_demo_Eth')
(np.mean( df_demo_cl1, axis=0)).to_csv('df_mean_demo_world') # world mean democracy sco

#Extracting cleaned  Food Consumption data
(data_extract (df_food_cl1,'United States')).to_csv('df_clnd_food_US')
(data_extract (df_food_cl1,'Ethiopia')).to_csv('df_clnd_food_Eth')
(np.mean( df_food_cl1, axis=0)).to_csv('df_mean_food_world') #world mean Food Consumpti

#Extracting life Expectancy at birth  data
(data_extract (df_life_cl1,'United States')).to_csv( 'df_clnd_life_US')
(data_extract (df_life_cl1,'Ethiopia')).to_csv('df_clnd_life_Eth')
(np.mean( df_life_cl1, axis=0)).to_csv('df_mean_life_world')  #world mean life Expectan

#Extracting GDP per capita  data
(data_extract (df_GDP_cl1,'United States')).to_csv('df_clnd_GDP_US')
(data_extract (df_GDP_cl1,'Ethiopia')).to_csv('df_clnd_GDP_Eth')
(np.mean( df_GDP_cl1, axis=0)).to_csv('df_mean_GDP_world') #world mean GDP
```

In [45]:
```
#explore each newly created data frames
df_L=pd.read_csv('df_mean_GDP_world')
df_L.head(1)
```

Out[45]:

	1979	9375.563534527146
0	1980	9291.774245

```
In [46]: # change column names for the cleaned data .#Year & data_name
         def col_rename (filename,data_name):
             df=pd.read_csv(filename )
             df.columns=['Year',data_name]
             df=df.set_index('Year')
             return df
         # democracy

         (col_rename('df_clnd_demo_US','Democracy_Score_US')).to_csv('cleaned_D_US')
         (col_rename('df_clnd_demo_Eth','Democracy_Score_ETH')).to_csv('cleaned_D_Eth')
         (col_rename('df_mean_demo_world','Democracy_Score_World')).to_csv('cleaned_D_World')

         #Food Consumption
         (col_rename('df_clnd_food_US','Food_consumption_US')).to_csv('cleaned_F_US')
         (col_rename('df_clnd_food_Eth','Food_consumption_ETH')).to_csv('cleaned_F_Eth')
         (col_rename('df_mean_food_world','Food_consumption_World')).to_csv('cleaned_F_World')

         #life Expectancy
         (col_rename('df_clnd_life_US','Life_Expectancy_US')).to_csv('cleaned_L_US')
         (col_rename('df_clnd_life_Eth','Life_Expectancy_ETH')).to_csv('cleaned_L_Eth')
         (col_rename('df_mean_life_world','Life_Expectancy_World')).to_csv('cleaned_L_World')

         #GDP

         (col_rename('df_clnd_GDP_US','GDP_PerCapita_US')).to_csv('cleaned_G_US')
         (col_rename('df_clnd_GDP_Eth','GDP_PerCapita_ETH')).to_csv('cleaned_G_Eth')
         (col_rename('df_mean_GDP_world','GDP_PerCapita_World')).to_csv('cleaned_G_World')

In [47]: df=pd.read_csv('cleaned_G_World')
         df['GDP_PerCapita_World'].head(1)

Out[47]: 0    9291.774245
         Name: GDP_PerCapita_World, dtype: float64

In [48]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39 entries, 0 to 38
Data columns (total 2 columns):
Year                  39 non-null int64
GDP_PerCapita_World   39 non-null float64
dtypes: float64(1), int64(1)
memory usage: 704.0 bytes


In [49]: # Merge datasets into their respective catagory
         def merge_data(file1,file2,file3,given_name):
             """ Merge the data Set and saves as csv with the given name"""
             df1=pd.read_csv(file1)
```

```
    df2=pd.read_csv(file2)
    df3=pd.read_csv(file3)
    df4 = df1.merge(df2, on='Year', how='outer')
    df5=df4.merge (df3,on='Year',how= 'outer')
    return df5.to_csv(given_name,index=False)

merge_data('cleaned_D_US','cleaned_D_Eth','cleaned_D_World', 'democracy_csv')
merge_data('cleaned_F_US','cleaned_F_Eth','cleaned_F_World', 'Food_Cons_csv')
merge_data('cleaned_L_US','cleaned_L_Eth','cleaned_L_World', 'Life_Exp_csv')
merge_data('cleaned_G_US','cleaned_G_Eth','cleaned_G_World', 'GDP_Per_csv')
```

In [50]:
```
df_catagory =pd.read_csv('democracy_csv')
df_catagory.nunique()
```

Out[50]:
```
Year                     32
Democracy_Score_US        1
Democracy_Score_ETH       5
Democracy_Score_World    31
dtype: int64
```

In [51]:
```
# Merge all into single data drame
def merge_dataframe(file1,file2,file3,file4,given_name):
    """ Merge the data Sets into a single table  and saves as csv with the given name"'
    df1=pd.read_csv(file1)
    df2=pd.read_csv(file2)
    df3=pd.read_csv(file3)
    df4=pd.read_csv(file4)
    df6 = df1.merge(df2, on='Year', how='outer')
    df7=df6.merge(df3,on='Year', how='outer')
    df8=df7.merge(df4,on='Year', how='outer')

    return df8.to_csv(given_name,index=False)

merge_dataframe('democracy_csv','Food_Cons_csv','Life_Exp_csv','GDP_Per_csv', 'Project_
```

In [52]:
```
#Examine the newly created  data frame

df=pd.read_csv('Project_Cleaned_df.csv')
df.head()
```

Out[52]:

	Year	Democracy_Score_US	Democracy_Score_ETH	Democracy_Score_World	\
0	1980	10.0	-7.0	-2.568862	
1	1981	10.0	-7.0	-2.574850	
2	1982	10.0	-7.0	-2.502994	
3	1983	10.0	-7.0	-2.329341	
4	1984	10.0	-8.0	-2.383234	

	Food_consumption_US	Food_consumption_ETH	Food_consumption_World	\
0	3187.86	1842.25	2528.093333	

```
1              3230.36               1757.25              2535.653072
2              3204.68               1772.23              2530.604706
3              3245.92               1783.61              2526.914902
4              3292.55               1572.56              2538.046536

   Life_Expectancy_US  Life_Expectancy_ETH  Life_Expectancy_World  \
0               73.93                42.80              63.220446
1               74.36                42.87              63.657772
2               74.65                42.93              64.083614
3               74.71                42.50              64.445644
4               74.81                39.46              64.814455

   GDP_PerCapita_US  GDP_PerCapita_ETH  GDP_PerCapita_World
0       27838.10795         588.379057          9291.774245
1       28160.13515         584.643511          9199.681971
2       27243.46779         577.860747          9041.686813
3       28119.68684         590.638490          9030.555571
4       29785.25419         565.816810          9159.042420
```

In [53]:
```
df.describe()
```

Out[53]:
```
              Year  Democracy_Score_US  Democracy_Score_ETH  \
count    39.000000                32.0            32.000000
mean   1999.000000                10.0            -2.046875
std      11.401754                 0.0             4.127718
min    1980.000000                10.0            -8.000000
25%    1989.500000                10.0            -7.000000
50%    1999.000000                10.0             1.000000
75%    2008.500000                10.0             1.000000
max    2018.000000                10.0             1.000000

       Democracy_Score_World  Food_consumption_US  Food_consumption_ETH  \
count              32.000000            28.000000             28.000000
mean                1.270728          3536.256786           1716.457500
std                 2.472360           188.407697            142.463583
min                -2.574850          3187.860000           1516.480000
25%                -1.769461          3424.995000           1579.085000
50%                 2.248466          3547.105000           1703.775000
75%                 3.297546          3695.775000           1843.520000
max                 4.036810          3795.800000           1979.680000

       Food_consumption_World  Life_Expectancy_US  Life_Expectancy_ETH  \
count               28.000000           37.000000            37.000000
mean              2607.282431           76.679730            51.852703
std                 68.836139            1.628594             8.283103
min               2526.914902           73.930000            35.430000
25%               2557.607239           75.100000            44.820000
50%               2579.833250           76.800000            50.600000
```

```
75%              2648.598835           78.100000            58.600000
max              2756.139831           79.100000            65.700000

       Life_Expectancy_World  GDP_PerCapita_US  GDP_PerCapita_ETH  \
count              37.000000         39.000000          39.000000
mean               68.077811      37469.349379         667.730517
std                 2.596790       5709.389477         228.331179
min                63.220446      27243.467790         421.353465
25%                66.402475      33331.980040         514.338131
50%                67.762344      38912.581780         565.816810
75%                70.139712      41952.495500         795.357847
max                72.556635      47907.100480        1225.200715

       GDP_PerCapita_World
count            39.000000
mean          11473.853487
std            1826.705486
min            9030.555571
25%            9865.750042
50%           11346.111108
75%           13253.793166
max           14246.914126
```

```
In [54]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39 entries, 0 to 38
Data columns (total 13 columns):
Year                     39 non-null int64
Democracy_Score_US       32 non-null float64
Democracy_Score_ETH      32 non-null float64
Democracy_Score_World    32 non-null float64
Food_consumption_US      28 non-null float64
Food_consumption_ETH     28 non-null float64
Food_consumption_World   28 non-null float64
Life_Expectancy_US       37 non-null float64
Life_Expectancy_ETH      37 non-null float64
Life_Expectancy_World    37 non-null float64
GDP_PerCapita_US         39 non-null float64
GDP_PerCapita_ETH        39 non-null float64
GDP_PerCapita_World      39 non-null float64
dtypes: float64(12), int64(1)
memory usage: 4.0 KB
```

```
In [55]: df.shape
```

```
Out[55]: (39, 13)
```

```
In [56]: df.nunique()
```

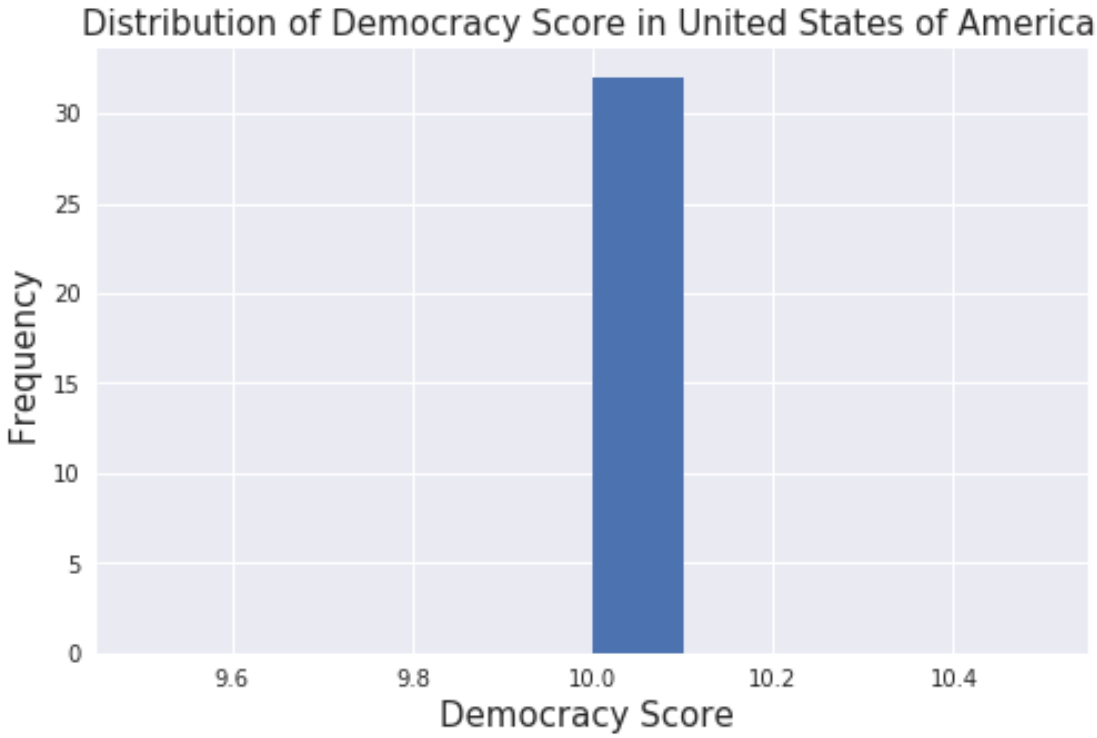
```
Out[56]: Year                    39
         Democracy_Score_US       1
         Democracy_Score_ETH      5
         Democracy_Score_World    31
         Food_consumption_US      28
         Food_consumption_ETH     28
         Food_consumption_World   28
         Life_Expectancy_US       30
         Life_Expectancy_ETH      36
         Life_Expectancy_World    37
         GDP_PerCapita_US         39
         GDP_PerCapita_ETH        39
         GDP_PerCapita_World      39
         dtype: int64
```

Exploratory Data Analysis

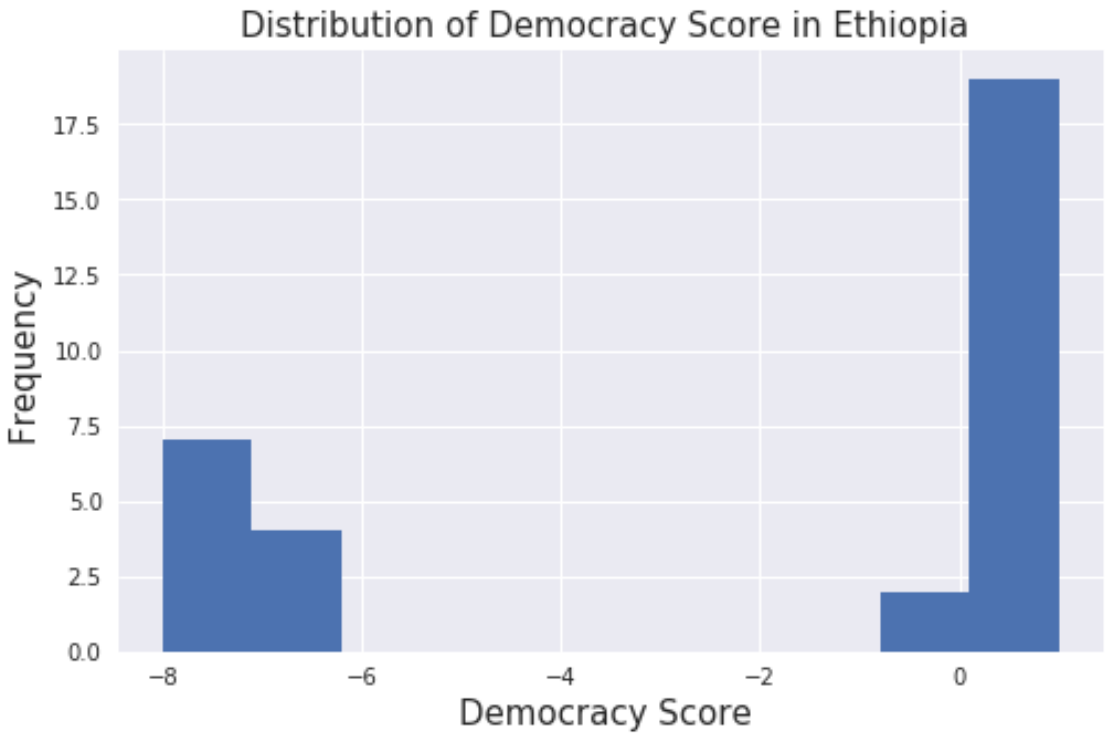
1.1.3 Research Question 1 :

How was the changes in Democracy Score of the two Countries over the years? How about as compared to the World ?

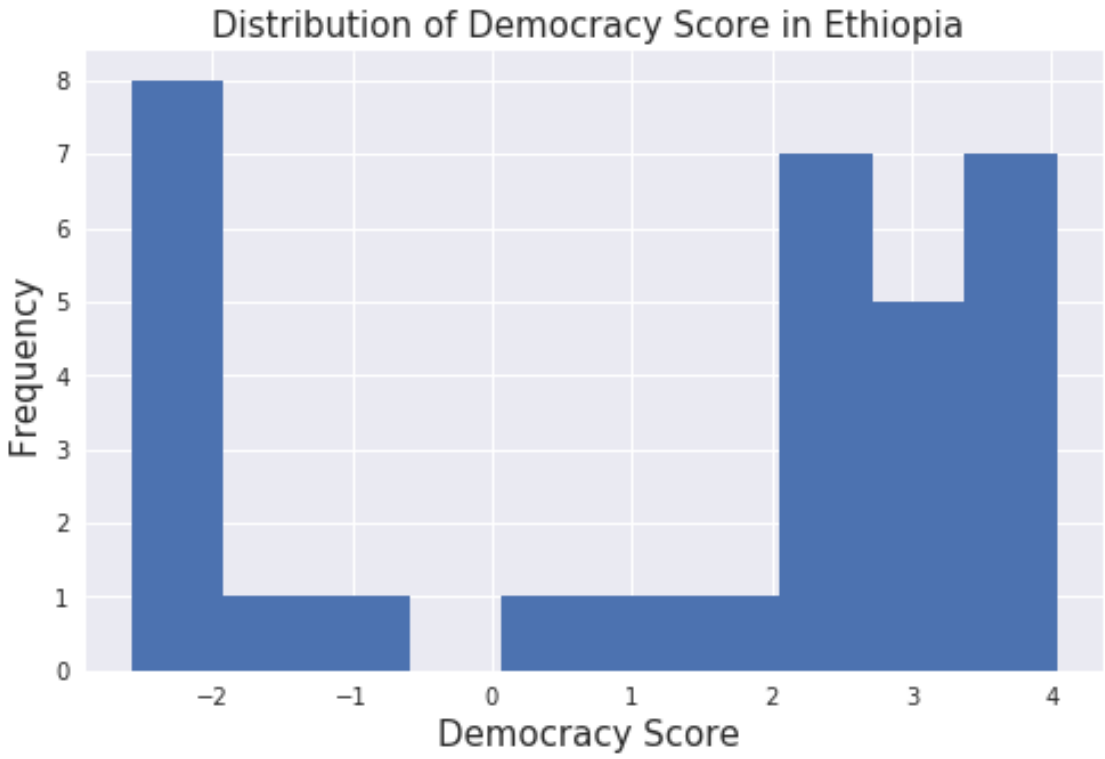
```
In [86]: # Ploting  the distribution democracy score of America
         import pylab as pl
         from pandas import *
         df_D=pd.read_csv ('Project_Cleaned_df.csv')
         #df_D=df_D.drop(df_F.columns[0], axis=1)
         df_D.hist( column= 'Democracy_Score_US',figsize=(8,5))
         pl.title("Distribution of Democracy Score in United States of America ",fontsize=15)
         pl.xlabel("Democracy Score ",fontsize=15)
         pl.ylabel("Frequency ",fontsize=15);
```



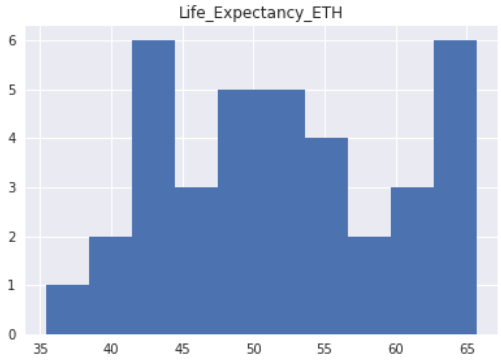
In [87]:
```
# Ploting  the distribution of   democracy score of Ethiopia
df_D=pd.read_csv ('Project_Cleaned_df.csv')
df_D.hist( column= 'Democracy_Score_ETH',figsize=(8,5))
pl.title("Distribution of Democracy Score in Ethiopia ",fontsize=15)
pl.xlabel("Democracy Score ",fontsize=15)
pl.ylabel("Frequency ",fontsize=15);
```

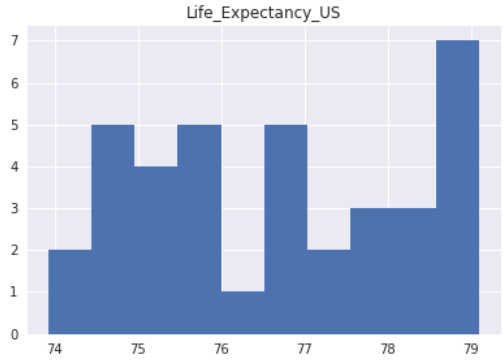


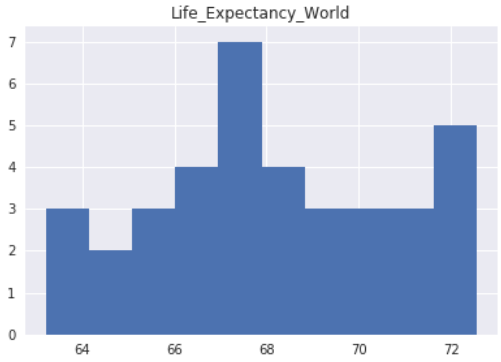

```
In [109]: # Ploting Global  democracy score distribution.
          df_D=pd.read_csv ('Project_Cleaned_df.csv')
          df_D.hist( column= 'Democracy_Score_World',figsize=(8,5))
          pl.title("Distribution of Democracy Score in Ethiopia ",fontsize=15)
          pl.xlabel("Democracy Score ",fontsize=15)
          pl.ylabel("Frequency ",fontsize=15);
```



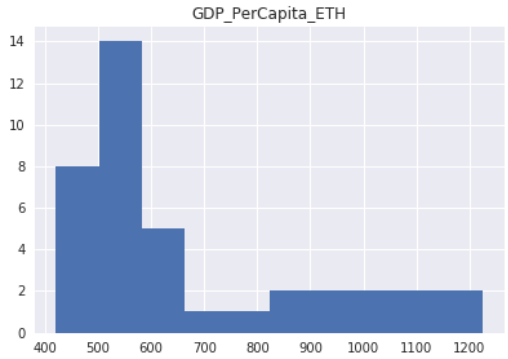

```
In [78]: df_L=pd.read_csv ('Life_Exp_csv')
         df_L=df_L.drop(df_L.columns[0], axis=1)
         df_L.hist(figsize=(15,10), label='frequency')

         ;

Out[78]: ''
```

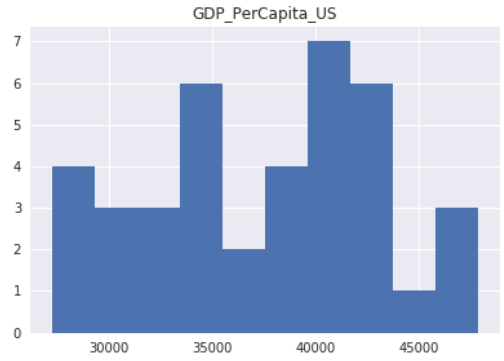


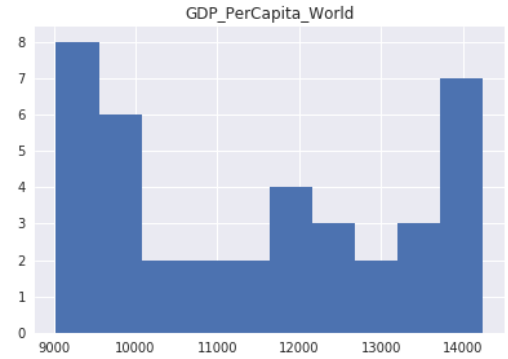





```
In [75]: df_G=pd.read_csv ('GDP_Per_csv')
         df_G=df_G.drop(df_G.columns[0], axis=1)
         df_G.hist(figsize=(15,10))
         plt.ylabel( 'frequency')

         ;

Out[75]: ''
```

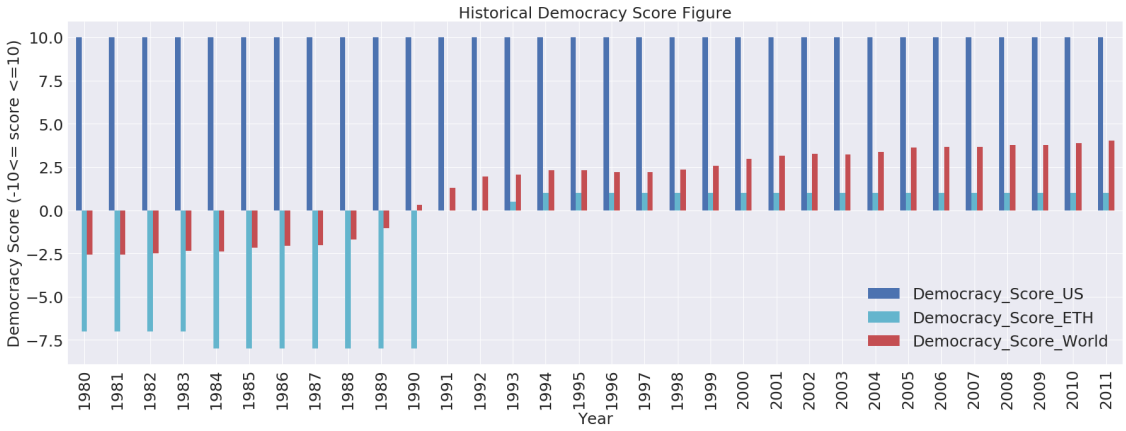




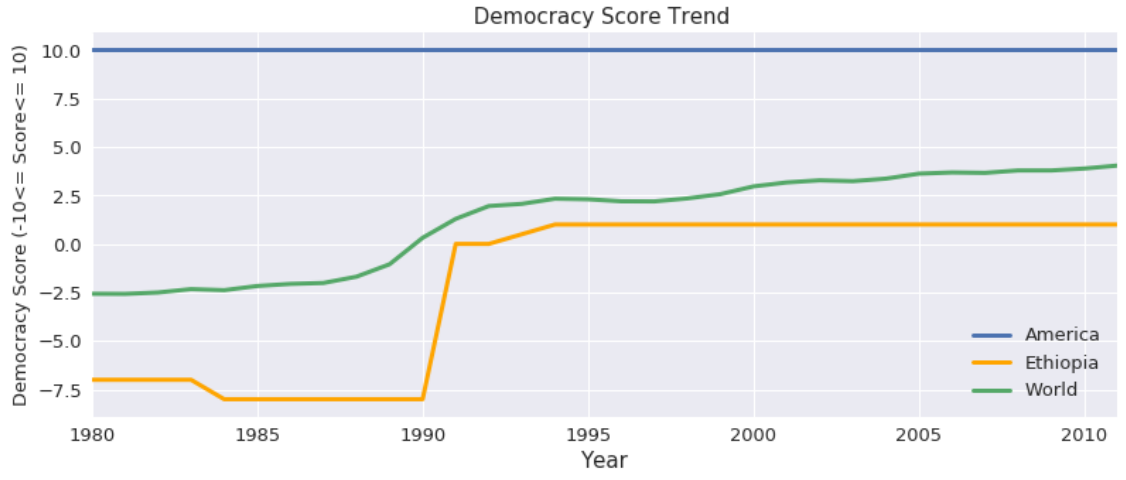


In [155]:
```
# Ploting the historical democracy score of both the countries and world .

df_d= pd.read_csv('democracy_csv')
df_D=df_d.drop(df_d.columns[0], axis=1)

df_D.plot(kind ='bar' ,x=df_d['Year'],figsize=(30,10),fontsize =25, color=('B','C','R'
pl.title("Historical Democracy Score Figure",fontsize=25)
pl.xlabel("Year",fontsize=25)
pl.ylabel("Democracy Score (-10<= score <=10)",fontsize=25)
pl.legend (fontsize=25)

;
```

Out[155]:
```
''
```



In [114]:
```
#Ploting the democracy score as time series
%matplotlib inline
sns.set()
df_demo=pd.read_csv ('democracy_csv')
#df_demo.head(1)
df_demo.columns=[ 'Year','America','Ethiopia','World']
df_demo.set_index('Year', inplace=True )
df_demo.plot(figsize=(13,5), linewidth=3, fontsize=13,color=('B','Orange','G'))
plt.xlabel('Year', fontsize=15)
plt.title(" Democracy Score Trend  ",fontsize=15)
plt.ylabel('Democracy Score (-10<= Score<= 10)' ,fontsize=13);
plt.legend(fontsize=13);
```



In [ ]:
```
x= df['Food_consumption_World']
y=df['Democracy_Score_World']
sns.regplot(x, y, data=df);
```

1.1.4 Discussion

Based on the data and the analysis performed :

1. United states of America had consistant democracy score of 10 on the researched period(which is the highest end of the democracy score scale).
2. Ethiopia's democracy score , has generally improved at the begining of 90's as compared to the 80's.The score has been observed ,however , to be almost stagnant from mid of 90's to 2011 with the democracy score of 1.During this period , when compared to the Global score ,Ethiopia is way lagging .Moreover, on comparion to the global mean scores the countryis not showing any visible progress .These findings are based of the data obtained from Gapminder and source referd as *Polity IV Project: Political Regime Characteristics and Transitions, 1800-2009*. The democracy score is described as *It is a summary measure of a country's democratic and free nature. -10 is the lowest value, 10 the highest* and anarchy or interregnum has been coded as 0 . The mean democracy score for Ethiopia is -2.047 , the global is 1.271 and that of US is 10.
3. The Global democraty score overall seems improving .It was in the less than zero range before 1990 and had increased to more than 2.5 in last years of the research period .

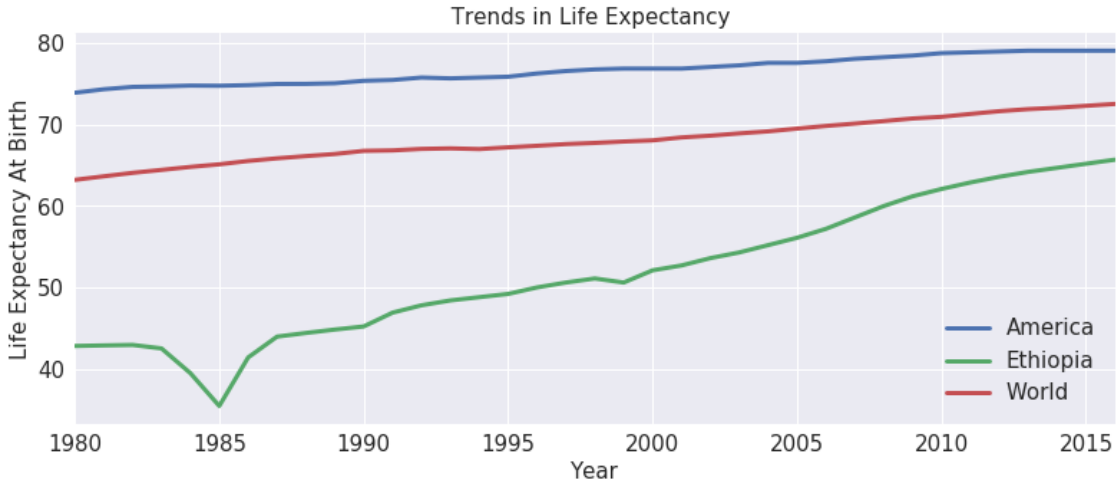
1.1.5 Research Question 2 :

Are there any noticeable trends in the independent variables , as Democracy Score has been changing over the years ?

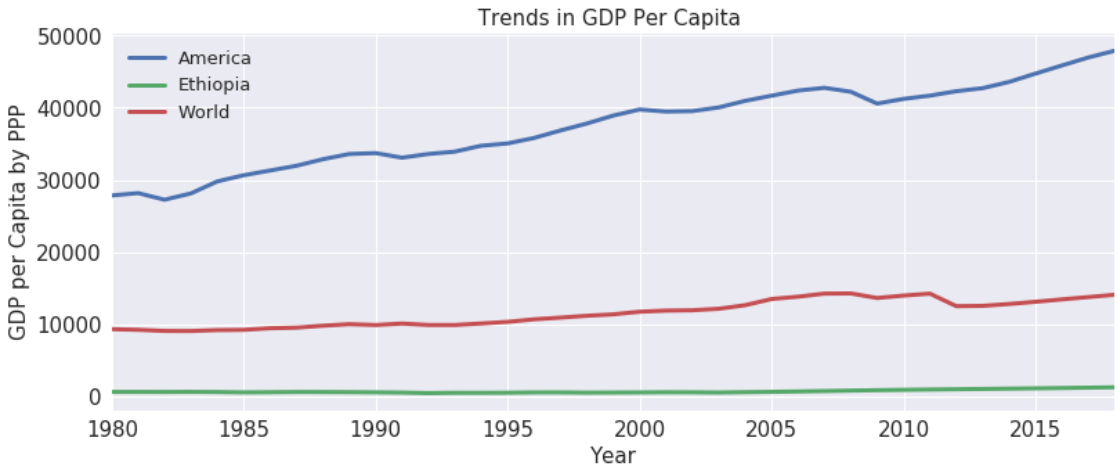
1.1.6 Research Question 3 :

What kind of correlations and influences observed between dependent and independent Variables ?

```
In [69]: # Life Expectancy
         %matplotlib inline
         sns.set()
         df_life=pd.read_csv ('Life_Exp_csv')
         df_life.columns=[ 'Year','America','Ethiopia','World']
         df_life.set_index('Year', inplace=True )
         df_life.plot(figsize=(13,5), linewidth=3, fontsize=15)
         plt.xlabel('Year', fontsize=15)
         plt.ylabel('Life Expectancy At Birth' ,fontsize=15);
         plt.legend(fontsize=15)
         plt.title(" Trends in Life Expectancy  ",fontsize=15);
```

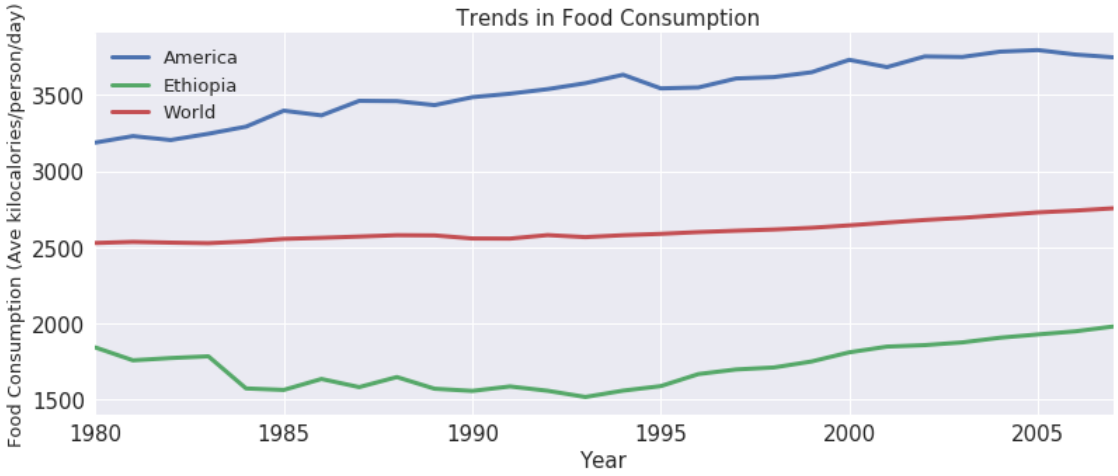


In [29]:
```
GDP per Capita
%matplotlib inline
sns.set()
df_gdp=pd.read_csv ('GDP_Per_csv')
df_gdp.columns=[ 'Year','America','Ethiopia','World']
df_gdp.set_index('Year', inplace=True )
df_gdp.plot(figsize=(13,5), linewidth=3, fontsize=15)
plt.xlabel('Year', fontsize=15)
plt.title(" Trends in GDP Per Capita  ",fontsize=15)
plt.ylabel('GDP per Capita by PPP' ,fontsize=15)
plt.legend(fontsize=13);
```



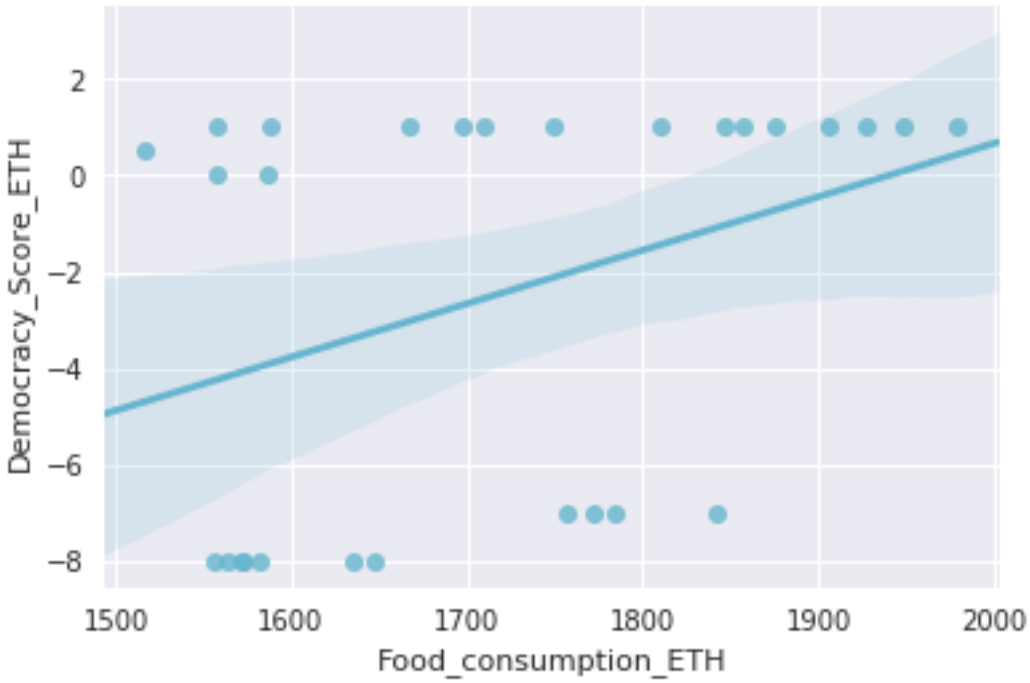
In [30]:
```
%matplotlib inline
sns.set()
```

```
df_food=pd.read_csv ('Food_Cons_csv')
df_food.columns=[ 'Year','America','Ethiopia','World']
df_food.set_index('Year', inplace=True )
df_food.plot(figsize=(13,5), linewidth=3, fontsize=15)
plt.xlabel('Year', fontsize=15)
plt.ylabel('Food Consumption (Ave kilocalories/person/day)' ,fontsize=13)
plt.title("  Trends in Food Consumption ",fontsize=15)
plt.legend(fontsize=13);
```

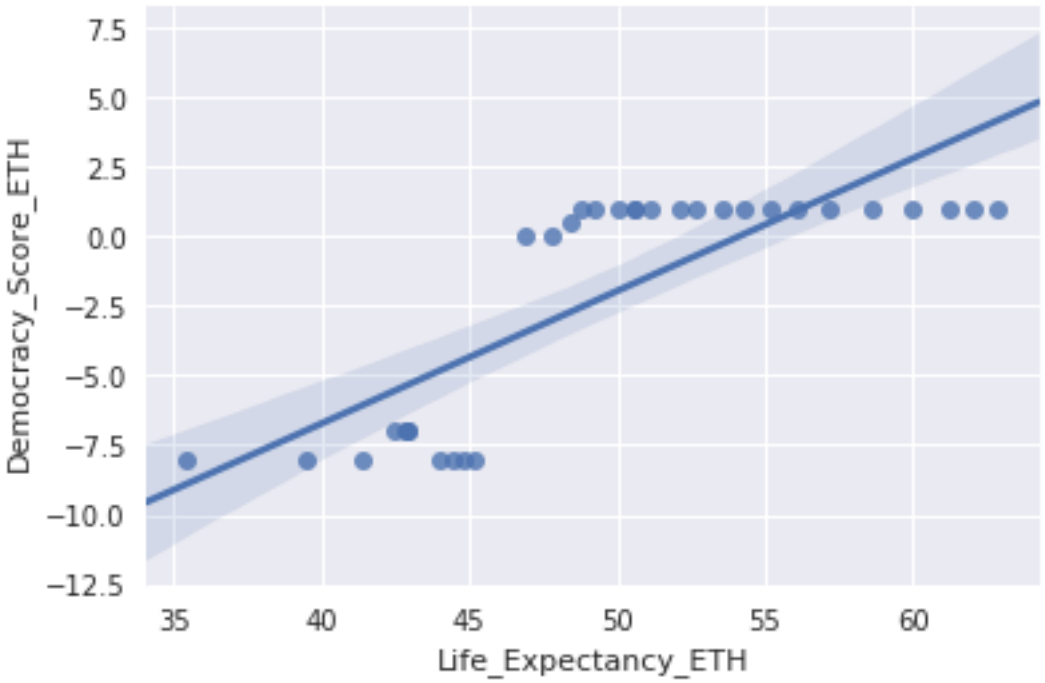


In [138]:
```
import seaborn as sns;
sns.set(color_codes=True)
y=df['Democracy_Score_ETH']
x= df['Food_consumption_ETH']
sns.regplot(x, y, data=df,color='C')
;
```

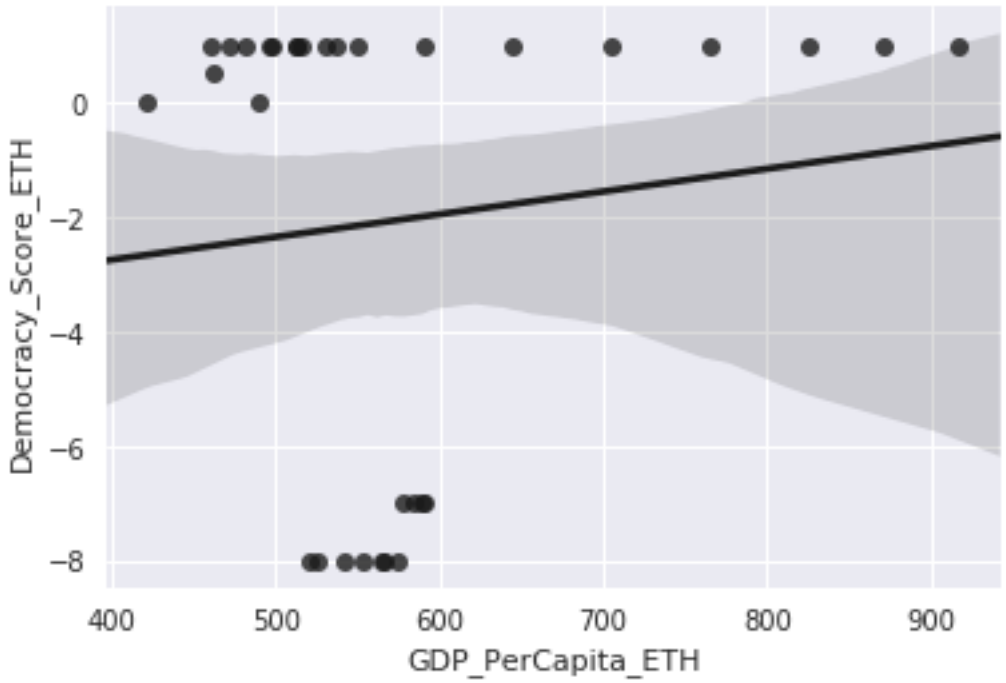
Out[138]: ''



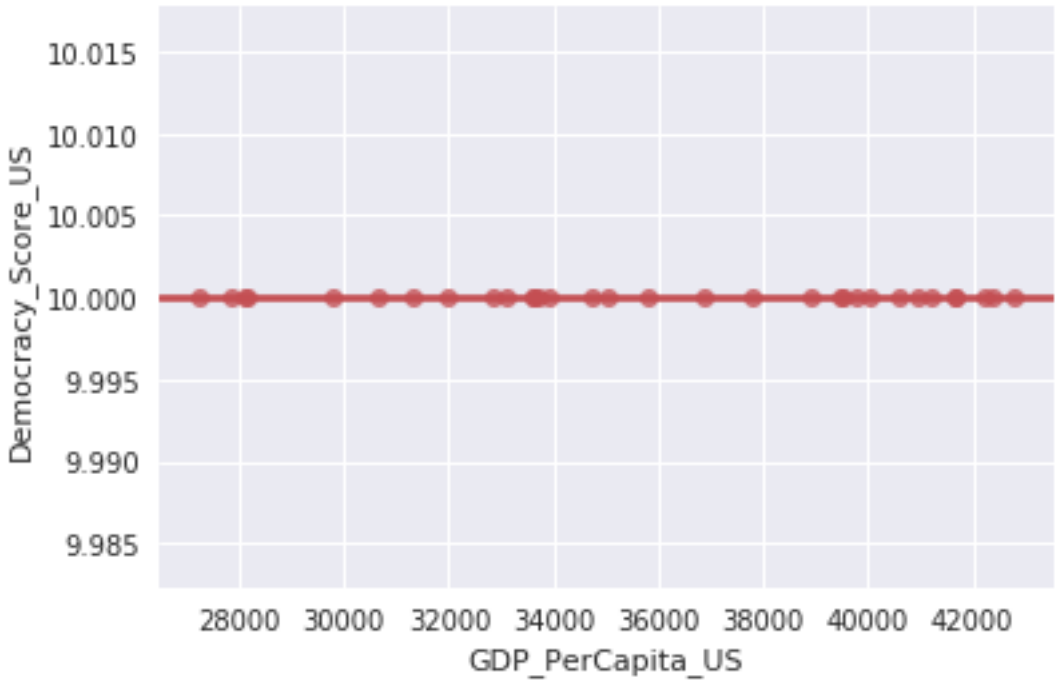
In [134]:
```
sns.set(color_codes=True)

x= df['Life_Expectancy_ETH']
y=df['Democracy_Score_ETH']

sns.regplot(x, y, data=df,color='B');
```



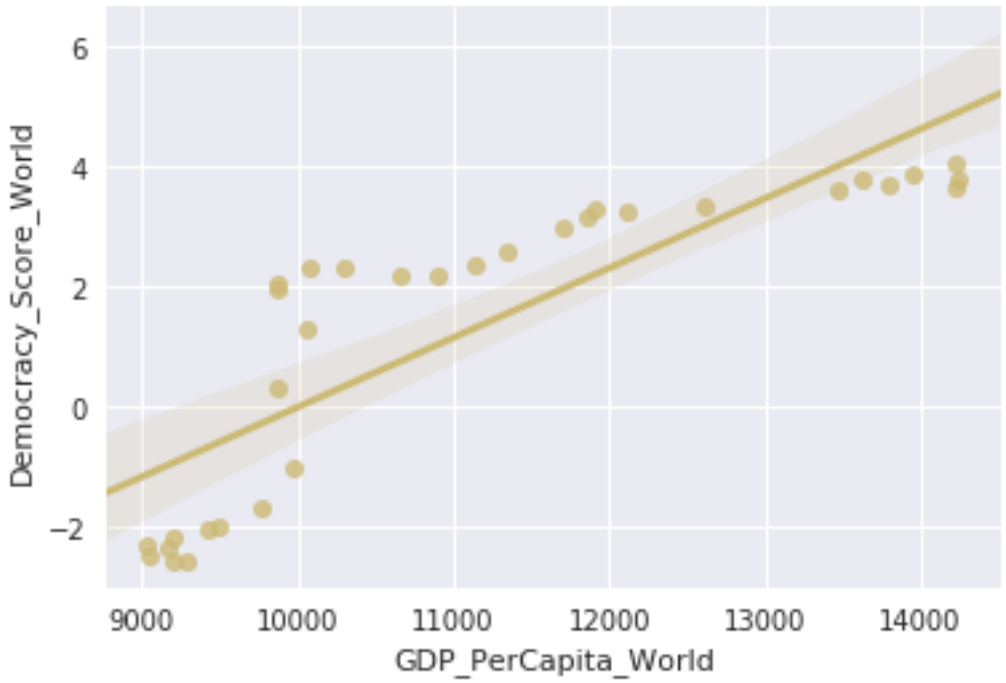
In [147]:
```
sns.set(color_codes=True)

x= df['GDP_PerCapita_ETH']
y=df['Democracy_Score_ETH']

sns.regplot(x, y, data=df,color='K');
```



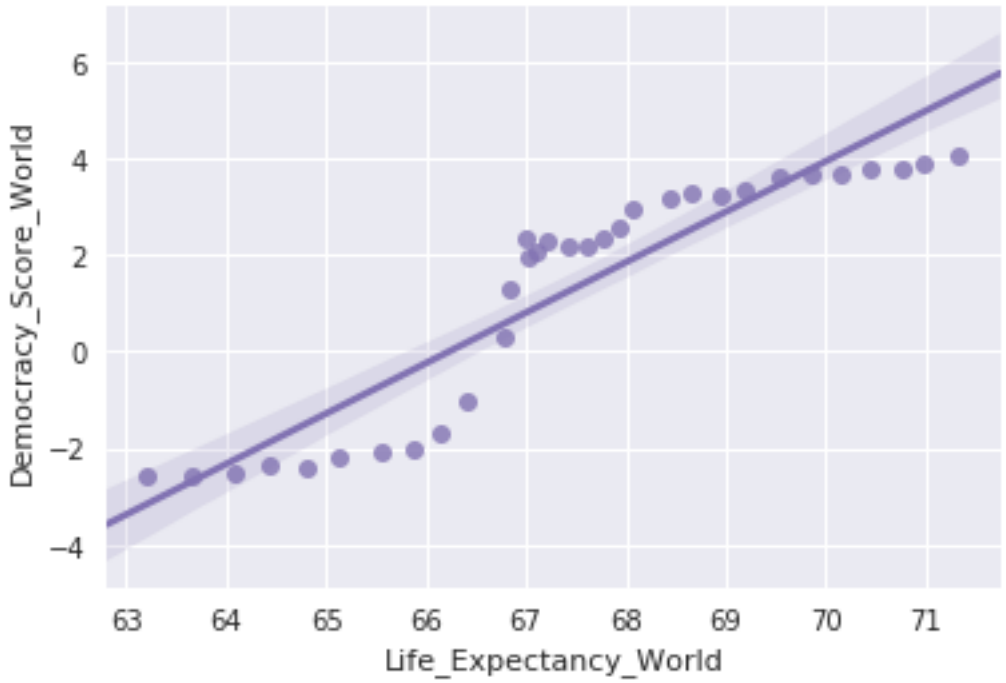
In [136]:
```
sns.set(color_codes=True)

x= df['GDP_PerCapita_US']
y=df['Democracy_Score_US']

sns.regplot(x, y, data=df,color='R');
```




```
In [137]: df['Democracy_Score_US'].describe ()
          # As the standard devation of the Democracy score of US is 0.
          #All the scatter plot graphs expected to have points on a single horizontal line

Out[137]: count    32.0
          mean     10.0
          std       0.0
          min      10.0
          25%      10.0
          50%      10.0
          75%      10.0
          max      10.0
          Name: Democracy_Score_US, dtype: float64

In [160]: sns.set(color_codes=True)

          x= df['GDP_PerCapita_World']
          y=df['Democracy_Score_World']

          sns.regplot(x, y, data=df,color='Y');
```

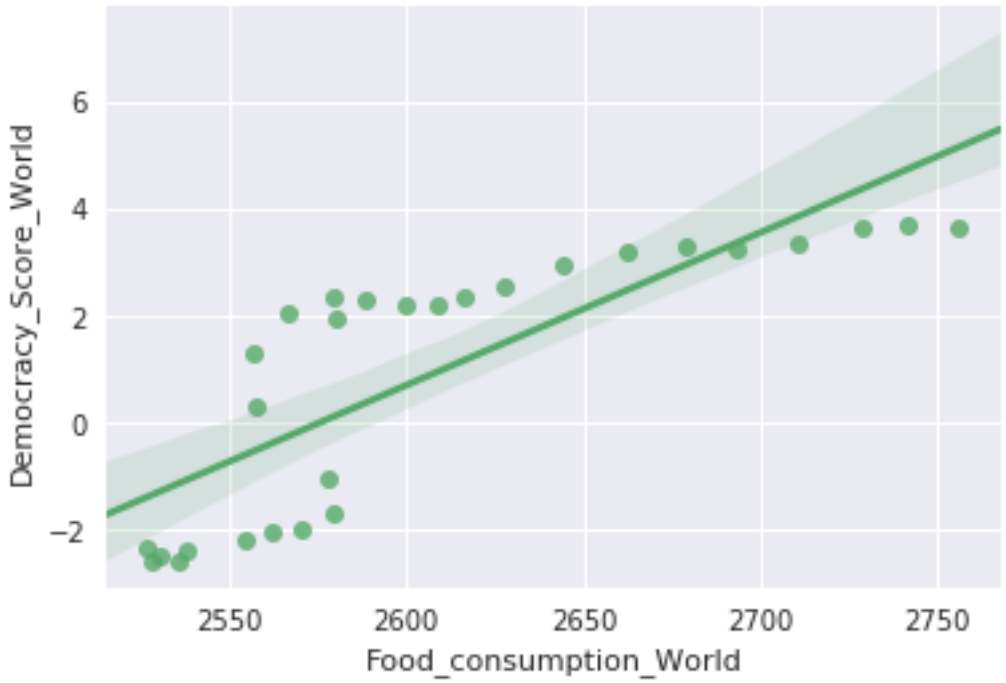



```
In [152]: sns.set(color_codes=True)

          x= df['Life_Expectancy_World']
          y=df['Democracy_Score_World']
          sns.regplot(x, y, data=df,color='M')

          ;

Out[152]: ''
```



In [161]:
```
sns.set(color_codes=True)

x= df['Food_consumption_World']
y=df['Democracy_Score_World']
sns.regplot(x, y, data=df,color='G')
;
```

Out[161]: ''



1.1.7 Discussion

(Research question 1&2)

All variables namely Food Consumption, GDP Per Capita and Life Expectancy have been observed to have obvious variations as the Democracy Score changes. The exploration suggests that Democracy Score could have irrefutable influence on the independent variables .The democracy Score seems positively correlated to the variables under study. The Ethiopian case indicates that the democracy score has higher influence on Food Consumption (Kilocalories available per person per day) and Life Expectancy at birth than on GDP per Capita . While with the world wide data, the exploration strengths the suggestion that democracy has positive correlation to all the three variables.

Conclusion

This preliminary study conducted with the assumption that the result could be utilized as starting point for further research or study. The results are not conclusive and are not supported with indepth stastical analysis .However, the observations strongly suggest that the democracy level of any nation (depicted in democracy score) has positive correlation to the Life Expectancy at Birth , GDP Per Capita and the Food Consumption in kilocalories per person per day .

```
In [162]: from subprocess import call
          call(['python', '-m', 'nbconvert', 'Investigate_a_Dataset.ipynb'])
```

```
Out[162]: 0
```