In [1]:

```
#importing Python packages
import pandas as pd
import numpy as np
import matplotlib as mp
get_ipython().run_line_magic('matplotlib', 'inline')
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('darkgrid')
```

In [2]:

```
# Importing time series data from the local drive through creating dataframes
# (the original source: https://www.gapminder.org/data/)
# The economic time series data is on GDP, Gini index and Billionaires per 1M po
pulation for countries across the world
gdp_df = pd.read_csv(r'/Users/ainurartay/Desktop/DAND/project2/data (gapminder)/
gdppercapita_us_inflation_adjusted.csv')
gini_df = pd.read_csv(r'/Users/ainurartay/Desktop/DAND/project2/data (gapminde
r)/gini.csv')
bln_df = pd.read_csv(r'/Users/ainurartay/Desktop/DAND/project2/data (gapminder)/
dollar_billionaires_per_million_people.csv')
```

In [3]:

```
# We start exploring the data by viewing the contents (structure, time range) of
the dataframes using .head() function
gdp_df.head()
```

Out[3]:

	country	1960	1961	1962	1963	1964	1965	1966	1967	1968	...	2
0	Afghanistan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	4
1	Albania	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	37
2	Algeria	2470.0	2080.0	1630.0	2130.0	2200.0	2280.0	2110.0	2240.0	2410.0	...	43
3	Andorra	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	439
4	Angola	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	35

5 rows × 59 columns

In [4]:

```
gini_df.head()
```

Out[4]:

	country	1800	1801	1802	1803	1804	1805	1806	1807	1808	...	2031	2032	2033
0	Afghanistan	30.5	30.5	30.5	30.5	30.5	30.5	30.5	30.5	30.5	...	36.8	36.8	36.8
1	Albania	38.9	38.9	38.9	38.9	38.9	38.9	38.9	38.9	38.9	...	29.0	29.0	29.0
2	Algeria	56.2	56.2	56.2	56.2	56.2	56.2	56.2	56.2	56.2	...	27.6	27.6	27.6
3	Andorra	40.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0	...	40.0	40.0	40.0
4	Angola	57.2	57.2	57.2	57.2	57.2	57.2	57.2	57.2	57.2	...	42.6	42.6	42.6

5 rows × 242 columns

In [5]:

```
bln_df.head()
```

Out[5]:

	country	2004	2005	2006	2007
0	Argentina	0.0255	0.0253	0.0250	0.0248
1	Australia	0.2510	0.2990	0.3450	0.5870
2	Austria	0.3670	0.4890	0.3660	0.3660
3	Belgium	0.0966	0.0965	0.1930	0.1930
4	Brazil	0.0326	0.0430	0.0851	0.1050

In [6]:

```
#Exploring the data for completeness (nulls) and data types
gdp_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 190 entries, 0 to 189
Data columns (total 59 columns):
country    190 non-null object
1960       88 non-null float64
1961       89 non-null float64
1962       89 non-null float64
1963       89 non-null float64
1964       89 non-null float64
1965       93 non-null float64
1966       96 non-null float64
1967       97 non-null float64
1968       99 non-null float64
1969       99 non-null float64
1970       108 non-null float64
1971       108 non-null float64
1972       108 non-null float64
1973       108 non-null float64
1974       110 non-null float64
1975       114 non-null float64
1976       115 non-null float64
1977       120 non-null float64
1978       120 non-null float64
1979       121 non-null float64
1980       132 non-null float64
1981       136 non-null float64
1982       138 non-null float64
1983       138 non-null float64
1984       140 non-null float64
1985       141 non-null float64
1986       144 non-null float64
1987       147 non-null float64
1988       149 non-null float64
1989       150 non-null float64
1990       161 non-null float64
1991       162 non-null float64
1992       164 non-null float64
1993       165 non-null float64
1994       167 non-null float64
1995       176 non-null float64
1996       178 non-null float64
1997       180 non-null float64
1998       180 non-null float64
1999       181 non-null float64
2000       184 non-null float64
2001       185 non-null float64
2002       186 non-null float64
2003       186 non-null float64
2004       186 non-null float64
2005       186 non-null float64
2006       186 non-null float64
2007       187 non-null float64
2008       187 non-null float64
2009       187 non-null float64
2010       190 non-null int64
2011       187 non-null float64
2012       186 non-null float64
2013       186 non-null float64
2014       186 non-null float64
2015       185 non-null float64
2016       184 non-null float64
```

```
2017       183 non-null float64
dtypes: float64(57), int64(1), object(1)
memory usage: 87.7+ KB
```

In [7]:

```
#checking the completeness of the data
gdp_df.isnull().sum()
```

Out[7]:

```
country      0
1960       102
1961       101
1962       101
1963       101
1964       101
1965        97
1966        94
1967        93
1968        91
1969        91
1970        82
1971        82
1972        82
1973        82
1974        80
1975        76
1976        75
1977        70
1978        70
1979        69
1980        58
1981        54
1982        52
1983        52
1984        50
1985        49
1986        46
1987        43
1988        41
1989        40
1990        29
1991        28
1992        26
1993        25
1994        23
1995        14
1996        12
1997        10
1998        10
1999         9
2000         6
2001         5
2002         4
2003         4
2004         4
2005         4
2006         4
2007         3
2008         3
2009         3
2010         0
2011         3
2012         4
2013         4
2014         4
2015         5
2016         6
```

```
2017           7
dtype: int64
```

In [8]:

```
# Creating a new (sub) data frame with selected 4 countries (China, South Korea,
United States, Venezuela).
# "_mod" in the name of the data frame stands for "modified"
gdp_mod = gdp_df.loc[[35, 155, 181, 185], :]
```

In [9]:

```
# Checking if the new data farme captures the right countires
gdp_mod
```

Out[9]:

	country	1960	1961	1962	1963	1964	1965	1966	1967	19
35	China	192.0	141.0	132.0	142.0	164.0	187.0	202.0	185.0	173
155	South Korea	944.0	980.0	989.0	1050.0	1120.0	1170.0	1280.0	1360.0	151
181	United States	17000.0	17100.0	17900.0	18400.0	19200.0	20200.0	21300.0	21600.0	2240
185	Venezuela	12400.0	12400.0	12900.0	12900.0	13800.0	13900.0	13600.0	13500.0	1410

4 rows × 59 columns

In [10]:

```
# Doing the same for the 2 other data frames and limiting the time range for 196
0-2017 as in the gdp_mod dataframe
gini_mod = gini_df.loc[[35, 158, 186, 190], '1960' :'2017']
```

In [11]:

```
# Checking the new dataframe
gini_mod
```

Out[11]:

	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	...	2008	2009	2010	20
35	25.3	25.1	24.9	24.7	24.5	24.3	24.1	23.9	23.6	23.4	...	42.3	42.5	42.5	41
158	31.5	31.7	31.9	32.1	32.4	32.6	32.8	33.0	33.3	33.7	...	32.0	32.1	32.0	31
186	34.7	34.6	34.4	34.3	34.1	33.9	33.7	33.5	33.4	33.1	...	40.8	40.7	40.7	40
190	63.1	63.4	63.7	64.2	64.8	65.3	65.9	66.5	67.1	67.5	...	46.9	46.9	46.9	46

4 rows × 58 columns

In [12]:

```
# For convenience, I am transposing the dataframe. This transposed frame will be
used to plot a graph.
# "_t" in the name stands for "transposed"
gdp_mod_t=gdp_mod.transpose()
```

In [13]:

```
# Checking the transposed dataframe
gdp_mod_t
```

Out[13]:

	35	155	181	185
country	China	South Korea	United States	Venezuela
1960	192	944	17000	12400
1961	141	980	17100	12400
1962	132	989	17900	12900
1963	142	1050	18400	12900
1964	164	1120	19200	13800
1965	187	1170	20200	13900
1966	202	1280	21300	13600
1967	185	1360	21600	13500
1968	173	1510	22400	14100
1969	197	1690	22900	13700
1970	228	1820	23300	14300
1971	238	1970	23800	14100
1972	241	2070	24800	13900
1973	254	2330	25900	14500
1974	254	2510	25500	14400
1975	272	2660	25200	14400
1976	263	2960	26300	15100
1977	279	3280	27300	15600
1978	308	3570	28500	15500
1979	327	3820	29100	15200
1980	348	3700	28700	14100
1981	361	3900	29200	13700
1982	388	4160	28400	13100
1983	424	4640	29400	12200
1984	481	5070	31300	12100
1985	539	5410	32300	11800
1986	578	5950	33100	12300
1987	635	6630	34000	12400
1988	696	7350	35100	12800
1989	714	7780	36000	11400
1990	731	8460	36300	11800
1991	788	9250	35800	12700
1992	889	9720	36600	13100
1993	1000	10300	37100	12900
1994	1120	11100	38100	12300

	35	155	181	185
1995	1230	12100	38700	12500
1996	1340	12800	39700	12300
1997	1440	13500	41000	12800
1998	1540	12700	42300	12600
1999	1650	14000	43800	11600
2000	1770	15100	45100	11800
2001	1910	15700	45000	12000
2002	2070	16700	45400	10700
2003	2260	17100	46300	9710
2004	2470	17900	47600	11300
2005	2740	18600	48800	12200
2006	3070	19400	49600	13200
2007	3490	20400	50000	14100
2008	3810	20800	49400	14700
2009	4140	20800	47600	14000
2010	4560	22100	48400	13500
2011	4970	22700	48800	13900
2012	5340	23100	49500	14500
2013	5720	23700	50000	14500
2014	6110	24300	50900	13700
2015	6500	24900	51900	NaN
2016	6890	25500	52300	NaN
2017	7330	26200	53100	NaN

In [14]:

```
# Moving the first row with names of the countries as column names (next 3 lines
of code)
new_header = gdp_mod_t.iloc[0]
```

In [15]:

```
gpd_mod_t=gdp_mod_t[1:]
```

In [16]:

```
gdp_mod_t.columns=new_header
```

In [17]:

```
# Deleting the first row with the country names and the last 3 rows with null va
lues for Venezuela
gdp_mod_t.drop(['country', '2015', '2016', '2017'], inplace=True)
```

In [18]:

```
# Checking the result
gdp_mod_t
```

Out[18]:

country	China	South Korea	United States	Venezuela
1960	192	944	17000	12400
1961	141	980	17100	12400
1962	132	989	17900	12900
1963	142	1050	18400	12900
1964	164	1120	19200	13800
1965	187	1170	20200	13900
1966	202	1280	21300	13600
1967	185	1360	21600	13500
1968	173	1510	22400	14100
1969	197	1690	22900	13700
1970	228	1820	23300	14300
1971	238	1970	23800	14100
1972	241	2070	24800	13900
1973	254	2330	25900	14500
1974	254	2510	25500	14400
1975	272	2660	25200	14400
1976	263	2960	26300	15100
1977	279	3280	27300	15600
1978	308	3570	28500	15500
1979	327	3820	29100	15200
1980	348	3700	28700	14100
1981	361	3900	29200	13700
1982	388	4160	28400	13100
1983	424	4640	29400	12200
1984	481	5070	31300	12100
1985	539	5410	32300	11800
1986	578	5950	33100	12300
1987	635	6630	34000	12400
1988	696	7350	35100	12800
1989	714	7780	36000	11400
1990	731	8460	36300	11800
1991	788	9250	35800	12700
1992	889	9720	36600	13100
1993	1000	10300	37100	12900
1994	1120	11100	38100	12300
1995	1230	12100	38700	12500

country	China	South Korea	United States	Venezuela
1996	1340	12800	39700	12300
1997	1440	13500	41000	12800
1998	1540	12700	42300	12600
1999	1650	14000	43800	11600
2000	1770	15100	45100	11800
2001	1910	15700	45000	12000
2002	2070	16700	45400	10700
2003	2260	17100	46300	9710
2004	2470	17900	47600	11300
2005	2740	18600	48800	12200
2006	3070	19400	49600	13200
2007	3490	20400	50000	14100
2008	3810	20800	49400	14700
2009	4140	20800	47600	14000
2010	4560	22100	48400	13500
2011	4970	22700	48800	13900
2012	5340	23100	49500	14500
2013	5720	23700	50000	14500
2014	6110	24300	50900	13700

In [19]:

```
# Similar manipulation for Gini index and Billionaires per 1M population
gini_mod_t=gini_mod.transpose()
```

In [20]:

```
# Renameing the column names from indeces to country names
gini_mod_t.rename(columns={35:'China', 158:'South Korea', 186:'United States', 1
90:'Venezuela'}, inplace=True)
```

In [21]:

```
gini_mod_t
```

Out[21]:

	China	South Korea	United States	Venezuela
1960	25.3	31.5	34.7	63.1
1961	25.1	31.7	34.6	63.4
1962	24.9	31.9	34.4	63.7
1963	24.7	32.1	34.3	64.2
1964	24.5	32.4	34.1	64.8
1965	24.3	32.6	33.9	65.3
1966	24.1	32.8	33.7	65.9
1967	23.9	33.0	33.5	66.5
1968	23.6	33.3	33.4	67.1
1969	23.4	33.7	33.1	67.5
1970	23.3	34.1	33.0	67.3
1971	23.3	34.5	32.9	66.7
1972	23.4	35.0	32.7	65.7
1973	23.5	35.6	32.7	64.3
1974	23.6	36.2	32.8	62.9
1975	23.8	36.8	32.9	61.6
1976	23.9	37.4	33.3	60.2
1977	24.0	37.9	33.7	58.8
1978	24.2	38.5	34.1	57.4
1979	24.3	38.9	34.6	56.5
1980	24.5	39.0	35.0	55.8
1981	24.8	38.9	35.4	55.3
1982	25.0	38.6	35.8	55.0
1983	25.4	38.1	36.3	54.9
1984	26.0	37.6	36.7	54.5
1985	26.7	37.0	37.0	54.2
1986	27.6	36.5	37.4	53.1
1987	28.7	36.0	37.6	51.2
1988	29.8	35.4	37.8	49.3
1989	30.9	35.0	38.0	47.2
1990	32.0	34.5	38.2	45.0
1991	33.0	34.1	38.6	44.0
1992	33.9	33.8	39.0	44.1
1993	34.5	33.5	39.5	44.8
1994	35.1	33.2	39.9	45.8
1995	35.7	32.9	40.3	47.2

	China	South Korea	United States	Venezuela
1996	36.2	32.6	40.5	48.3
1997	36.9	32.3	40.6	48.7
1998	37.6	32.0	40.6	48.8
1999	38.5	31.8	40.5	48.8
2000	39.3	31.7	40.5	49.0
2001	40.0	31.8	40.5	49.1
2002	40.4	31.9	40.5	49.4
2003	40.7	31.9	40.5	50.3
2004	40.9	32.0	40.6	50.0
2005	41.1	32.0	40.7	49.3
2006	41.5	32.0	40.8	48.6
2007	41.9	32.0	40.8	48.0
2008	42.3	32.0	40.8	46.9
2009	42.5	32.1	40.7	46.9
2010	42.5	32.0	40.7	46.9
2011	41.9	31.8	40.7	46.9
2012	41.3	31.7	40.8	46.9
2013	40.6	31.6	41.0	46.9
2014	40.0	31.6	41.2	46.9
2015	39.4	31.6	41.3	46.9
2016	39.2	31.6	41.4	46.9
2017	39.1	31.6	41.5	46.9

In [22]:

```
# For comparability, dropping the last 3 years from the data frame
gini_mod_t.drop(['2015','2016','2017'])
```

Out[22]:

	China	South Korea	United States	Venezuela
1960	25.3	31.5	34.7	63.1
1961	25.1	31.7	34.6	63.4
1962	24.9	31.9	34.4	63.7
1963	24.7	32.1	34.3	64.2
1964	24.5	32.4	34.1	64.8
1965	24.3	32.6	33.9	65.3
1966	24.1	32.8	33.7	65.9
1967	23.9	33.0	33.5	66.5
1968	23.6	33.3	33.4	67.1
1969	23.4	33.7	33.1	67.5
1970	23.3	34.1	33.0	67.3
1971	23.3	34.5	32.9	66.7
1972	23.4	35.0	32.7	65.7
1973	23.5	35.6	32.7	64.3
1974	23.6	36.2	32.8	62.9
1975	23.8	36.8	32.9	61.6
1976	23.9	37.4	33.3	60.2
1977	24.0	37.9	33.7	58.8
1978	24.2	38.5	34.1	57.4
1979	24.3	38.9	34.6	56.5
1980	24.5	39.0	35.0	55.8
1981	24.8	38.9	35.4	55.3
1982	25.0	38.6	35.8	55.0
1983	25.4	38.1	36.3	54.9
1984	26.0	37.6	36.7	54.5
1985	26.7	37.0	37.0	54.2
1986	27.6	36.5	37.4	53.1
1987	28.7	36.0	37.6	51.2
1988	29.8	35.4	37.8	49.3
1989	30.9	35.0	38.0	47.2
1990	32.0	34.5	38.2	45.0
1991	33.0	34.1	38.6	44.0
1992	33.9	33.8	39.0	44.1
1993	34.5	33.5	39.5	44.8
1994	35.1	33.2	39.9	45.8
1995	35.7	32.9	40.3	47.2

	China	South Korea	United States	Venezuela
1996	36.2	32.6	40.5	48.3
1997	36.9	32.3	40.6	48.7
1998	37.6	32.0	40.6	48.8
1999	38.5	31.8	40.5	48.8
2000	39.3	31.7	40.5	49.0
2001	40.0	31.8	40.5	49.1
2002	40.4	31.9	40.5	49.4
2003	40.7	31.9	40.5	50.3
2004	40.9	32.0	40.6	50.0
2005	41.1	32.0	40.7	49.3
2006	41.5	32.0	40.8	48.6
2007	41.9	32.0	40.8	48.0
2008	42.3	32.0	40.8	46.9
2009	42.5	32.1	40.7	46.9
2010	42.5	32.0	40.7	46.9
2011	41.9	31.8	40.7	46.9
2012	41.3	31.7	40.8	46.9
2013	40.6	31.6	41.0	46.9
2014	40.0	31.6	41.2	46.9

In [23]:

```
# Checking info in the new data frame
gdp_mod_t.info()
```

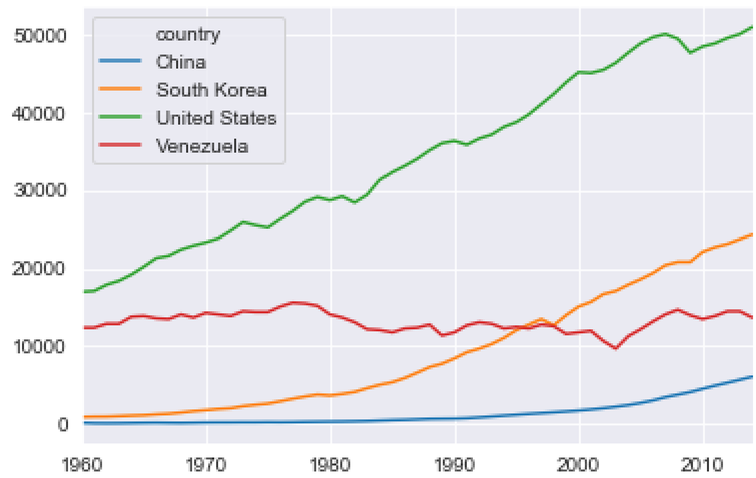
```
<class 'pandas.core.frame.DataFrame'>
Index: 55 entries, 1960 to 2014
Data columns (total 4 columns):
China            55 non-null object
South Korea      55 non-null object
United States    55 non-null object
Venezuela        55 non-null object
dtypes: object(4)
memory usage: 2.1+ KB
```

In [24]:

```
# Plotting line graphs for 4 countries to see the GDP trends.
# Upward movement for all except for Venezuela
gdp_mod_t.plot()
```

Out[24]:

<matplotlib.axes._subplots.AxesSubplot at 0x1a1bc71470>

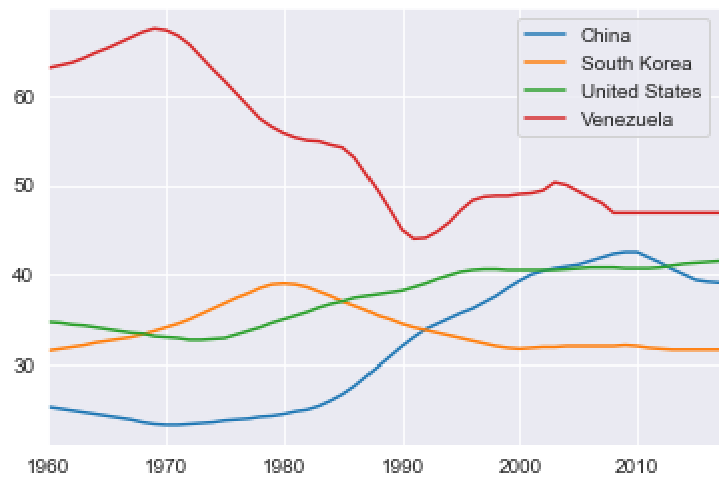


In [25]:

```
# Visualizing Gini index trends
gini_mod_t.plot()
```

Out[25]:

<matplotlib.axes._subplots.AxesSubplot at 0x10e3e6710>



In [26]:

```
# Manipulating the data on billionaires per 1M population -- selecting the 4 cou
ntires into a new dataframe
# and transposing the dataframe
bln_mod_t=(bln_df.loc[[7, 43, 52, 53], :]).transpose()
```

In [27]:

```
# Checking the dataframe
bln_mod_t
```

Out[27]:

	7	43	52	53
country	China	South Korea	United States	Venezuela
2004	0.00077	0.0413	0.945	0.0799
2005	0.00153	0.0617	1.15	0.0788
2006	0.0061	0.0819	1.24	0.0777
2007	0.0152	0.204	1.38	0.0767

In [28]:

```
# Renaming the column names from indeces to appropriate country names
bln_mod_t.columns=['China', 'South Korea', 'United States', 'Venezuela']
```

In [29]:

```
# Dropping the first row (non-numerical)
bln_mod_t=bln_mod_t.drop('country')
```

In [30]:

```
# Checking the new dataframe
bln_mod_t.info()
```

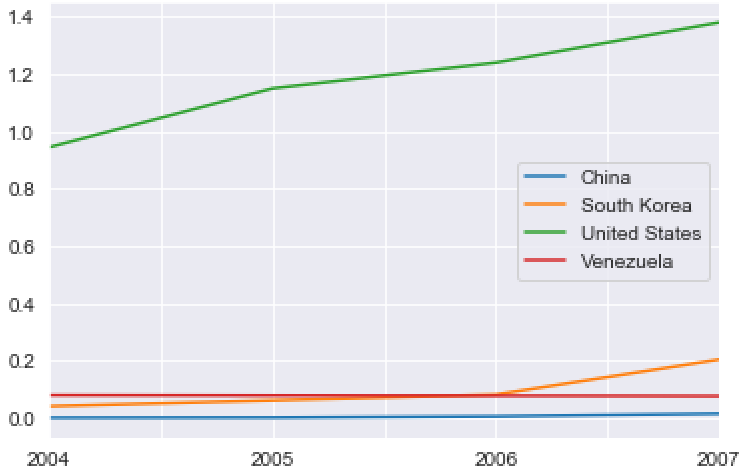
```
<class 'pandas.core.frame.DataFrame'>
Index: 4 entries, 2004 to 2007
Data columns (total 4 columns):
China            4 non-null object
South Korea      4 non-null object
United States    4 non-null object
Venezuela        4 non-null object
dtypes: object(4)
memory usage: 160.0+ bytes
```

In [31]:

```
# Visualizing the data on billionaires per 1M population
bln_mod_t.plot()
```

Out[31]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x10efd67f0>
```



In [ ]:

```

```