
01/11/2019

Python - gapminder lesson

NGG

Goals

1. Use pandas, Python library for statistics
2. Visualize data in publication quality plots
3. Automate tasks

Write functions, time permitting.

Let's get organized...

- Create a folder on your desktop
- Launch jupyter and navigate to that folder

Quick recap

```
In [67]: print('This cell is a code')
```

```
This cell is a code
```

```
In [68]: # This cell is code, but this is a comment  
# Whatever is written in front of this # sign, is ignored by Python
```

This cell is text/mardown.

Who doesn't love a bullet point?!

- Great way to keep track of your code/analysis
- Reproducibility

- Easily re-run analysis
- Write definitions, descriptions, meanings, ... what the code does so you can remember what you did...

1. Hello!
2. Bye!

Level 1

Level 2

Level 3

This is italic

This is bold

Wow, this is italic AND bold

```
In [7]: a=3  
        b=2  
        c=a+b  
        print(c)
```

5

```
In [8]: a=10  
        print(a)
```

10

```
In [9]: print(c)
```

5

```
In [10]: fruit='orange'
```

```
In [11]: fruits=['orange','strawberry','banana','mango']
```

```
In [12]: fruits[1]
```

```
Out[12]: 'strawberry'
```

```
In [19]: fruits[:3]
```

```
Out[19]: ['orange', 'strawberry', 'banana']
```

```
In [13]: fruits[-1]
```

Out[13]: mango

```
In [17]: print('So I went to the supermarket last night and bought',a, fruits[-1]+'s','LOL')
```

So I went to the supermarket last night and bought 10 mangos LOL

```
In [27]: type(fruit)
```

Out[27]: str

```
In [28]: type(fruits)
```

Out[28]: list

```
In [29]: type(a)
```

Out[29]: int

```
In [32]: d='55'  
         type(d)
```

Out[32]: str

```
In [36]: e=int(d)
```

```
In [40]: type(e)
```

Out[40]: int

```
In [41]: type(d)
```

Out[41]: str

```
In [30]: len(fruits)
```

Out[30]: 4

```
In [46]: numbers=[1,5,6,19,33]
```

```
In [47]: print('min =', min(numbers), 'max =', max(numbers))
```

min = 1 max = 33

```
In [48]: fruits.append('coconut')
```

```
In [49]: fruits
```

Out[49]: ['orange', 'strawberry', 'banana', 'mango', 'coconut']

```
In [50]: del fruits[1]
```

```
In [51]: fruits
```

```
Out[51]: ['orange', 'banana', 'mango', 'coconut']
```

```
In [70]: for k in fruits:
          print(k.title(), 'has', len(k), 'letters.')
```

```
Orange has 6 letters.
Banana has 6 letters.
Mango has 5 letters.
Coconut has 7 letters.
```

```
In [55]: for i in range(1,55,4):
          print(i)
```

```
1
5
9
13
17
21
25
29
33
37
41
45
49
53
```

```
In [71]: for i in range(1,10):
          print(i, '\t', i**2)
```

```
1      1
2      4
3      9
4     16
5     25
6     36
7     49
8     64
9     81
```

1. Download gapminder dataset

<http://swcarpentry.github.io/python-novice-gapminder/files/python-novice-gapminder-data.zip>
(<http://swcarpentry.github.io/python-novice-gapminder/files/python-novice-gapminder-data.zip>)

```
In [21]: %%hash
```

```
curl -O http://swcarpentry.github.io/python-novice-gapminder/files/python-novice-gapminder-data.zip
```

```

% Total    % Received % Xferd  Average Speed   Time    Time     Ti
me  Current                        Dload  Upload   Total   Spent    Le
ft  Speed
100 38471  100 38471    0     0  157k      0 --:--:-- --:--:-- --:--
-:-- 157k

```

In [75]: `ls`

```

data/                                python-novice-gapminder-data.zip
ngg_notes.ipynb

```

In [5]: `%%bash`
`unzip python-novice-gapminder-data.zip`

```

Archive:  python-novice-gapminder-data.zip
  inflating: data/gapminder_all.csv
  inflating: data/gapminder_gdp_africa.csv
  inflating: data/gapminder_gdp_americas.csv
  inflating: data/gapminder_gdp_asia.csv
  inflating: data/gapminder_gdp_europe.csv
  inflating: data/gapminder_gdp_oceania.csv

```

gross domestic product; measure of a continent's output

In [76]: `ls`

```

data/                                python-novice-gapminder-data.zip
ngg_notes.ipynb

```

In [77]: `cd data/`

```
/Users/grachetng/Desktop/11-1-2018_Carp.Python/data
```

In [78]: `ls`

```

gapminder_all.csv      gapminder_gdp_asia.csv
gapminder_gdp_africa.csv  gapminder_gdp_europe.csv
gapminder_gdp_americas.csv  gapminder_gdp_oceania.csv

```

In [83]: `%%bash`
`head gapminder_gdp_africa.csv`

```
country,gdpPercap_1952,gdpPercap_1957,gdpPercap_1962,gdpPercap_1967,
gdpPercap_1972,gdpPercap_1977,gdpPercap_1982,gdpPercap_1987,gdpPerca
p_1992,gdpPercap_1997,gdpPercap_2002,gdpPercap_2007
Algeria,2449.008185,3013.976023,2550.81688,3246.991771,4182.663766,4
910.416756,5745.160213,5681.358539,5023.216647,4797.295051,5288.0403
82,6223.367465
Angola,3520.610273,3827.940465,4269.276742,5522.776375,5473.288005,3
008.647355,2756.953672,2430.208311,2627.845685,2277.140884,2773.2873
12,4797.231267
Benin,1062.7522,959.6010805,949.4990641,1035.831411,1085.796879,1029
.161251,1277.897616,1225.85601,1191.207681,1232.975292,1372.877931,1
441.284873
Botswana,851.2411407,918.2325349,983.6539764,1214.709294,2263.611114
,3214.857818,4551.14215,6205.88385,7954.111645,8647.142313,11003.605
08,12569.85177
Burkina Faso,543.2552413,617.1834648,722.5120206,794.8265597,854.735
9763,743.3870368,807.1985855,912.0631417,931.7527731,946.2949618,103
7.645221,1217.032994
Burundi,339.2964587,379.5646281,355.2032273,412.9775136,464.0995039,
556.1032651,559.603231,621.8188189,631.6998778,463.1151478,446.40351
26,430.0706916
Cameroon,1172.667655,1313.048099,1399.607441,1508.453148,1684.146528
,1783.432873,2367.983282,2602.664206,1793.163278,1694.337469,1934.01
1449,2042.09524
Central African Republic,1071.310713,1190.844328,1193.068753,1136.05
6615,1070.013275,1109.374338,956.7529907,844.8763504,747.9055252,740
.5063317,738.6906068,706.016537
Chad,1178.665927,1308.495577,1389.817618,1196.810565,1104.103987,113
3.98495,797.9081006,952.386129,1058.0643,1004.961353,1156.18186,1704
.063724
```

2. Import data into DataFrames

```
In [85]: import pandas as pd
```

```
In [348]: data=pd.read_csv('gapminder_gdp_oceania.csv', index_col='country')
```

```
In [349]: data.head()
```

```
Out[349]:
```

	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpP
country					
Australia	10039.59564	10949.64959	12217.22686	14526.12465	1678
New Zealand	10556.57566	12247.39532	13175.67800	14463.91893	1604

```
In [94]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2 entries, Australia to New Zealand
Data columns (total 12 columns):
gdpPercap_1952    2 non-null float64
gdpPercap_1957    2 non-null float64
gdpPercap_1962    2 non-null float64
gdpPercap_1967    2 non-null float64
gdpPercap_1972    2 non-null float64
gdpPercap_1977    2 non-null float64
gdpPercap_1982    2 non-null float64
gdpPercap_1987    2 non-null float64
gdpPercap_1992    2 non-null float64
gdpPercap_1997    2 non-null float64
gdpPercap_2002    2 non-null float64
gdpPercap_2007    2 non-null float64
dtypes: float64(12)
memory usage: 208.0+ bytes
```

```
In [96]: data.columns # this is a variable that has the columns names stored in it; don't use () bc it's not a method
```

```
Out[96]: Index(['gdpPercap_1952', 'gdpPercap_1957', 'gdpPercap_1962', 'gdpPer
cap_1967',
               'gdpPercap_1972', 'gdpPercap_1977', 'gdpPercap_1982', 'gdpPer
cap_1987',
               'gdpPercap_1992', 'gdpPercap_1997', 'gdpPercap_2002', 'gdpPer
cap_2007'],
              dtype=object, name='columns')
```

```
cap_1987',
    'gdpPercap_1992', 'gdpPercap_1997', 'gdpPercap_2002', 'gdpPer
cap_2007'],
    dtype='object')
```

```
In [98]: data.head()
```

```
Out[98]:
```

	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPercap_1987
country					
Australia	10039.59564	10949.64959	12217.22686	14526.12465	16788.6215
New Zealand	10556.57566	12247.39532	13175.67800	14463.91893	16046.0365

3. Obtain summary statistics

```
In [101]: data.describe()
```

```
Out[101]:
```

	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPercap_1987
count	2.000000	2.000000	2.000000	2.000000	2.000000
mean	10298.085650	11598.522455	12696.452430	14495.021790	16417.336555
std	365.560078	917.644806	677.727301	43.986086	525.091935
min	10039.595640	10949.649590	12217.226860	14463.918930	16046.036555
25%	10168.840645	11274.086022	12456.839645	14479.470360	16231.681555
50%	10298.085650	11598.522455	12696.452430	14495.021790	16417.336555
75%	10427.330655	11922.958888	12936.065215	14510.573220	16602.981555
max	10556.575660	12247.395320	13175.678000	14526.124650	16788.621555

4. Save to file

```
In [160]: data.describe().to_csv('oceania_summ_stats.csv')
```

```
In [161]: ls
```



```

gapminder_all.csv          gapminder_gdp_europe.csv
gapminder_gdp_africa.csv  gapminder_gdp_oceania.csv
gapminder_gdp_americas.csv oceania_summ_stats.csv
gapminder_gdp_asia.csv

```

Cheeky Exercise

1.

A. Import `gapminder_gdp_americas.csv` as `americas`,

B. display summary statistics, and

C. save to `americas_summ_stats.csv`

```

In [146]: americas=pd.read_csv('gapminder_gdp_americas.csv', index_col='country'
        )
        americas.head()

```

Out[146]:

	continent	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967
country					
Argentina	Americas	5911.315053	6856.856212	7133.166023	8052.95302
Bolivia	Americas	2677.326347	2127.686326	2180.972546	2586.88605
Brazil	Americas	2108.944355	2487.365989	3336.585802	3429.86435
Canada	Americas	11367.161120	12489.950060	13462.485550	16076.5880
Chile	Americas	3939.978789	4315.622723	4519.094331	5106.65431

```
In [165]: americas.describe()
```

```
Out[165]:
```

	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPercap_1972
count	25.000000	25.000000	25.000000	25.000000	25.000000
mean	4079.062552	4616.043733	4901.541870	5668.253496	6491.334552
std	3001.727522	3312.381083	3421.740569	4160.885560	4754.404552
min	1397.717137	1544.402995	1662.137359	1452.057666	1654.451137
25%	2428.237769	2487.365989	2750.364446	3242.531147	4031.404552
50%	3048.302900	3780.546651	4086.114078	4643.393534	5305.444552
75%	3939.978789	4756.525781	5180.755910	5788.093330	6809.404552
max	13990.482080	14847.127120	16173.145860	19530.365570	21806.044552

```
In [164]: americas.describe().to_csv('americas_summ_stats.csv')
```

```
In [135]: americas.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 25 entries, Argentina to Venezuela
Data columns (total 13 columns):
continent                25 non-null object
gdpPercap_1952            25 non-null float64
gdpPercap_1957            25 non-null float64
gdpPercap_1962            25 non-null float64
gdpPercap_1967            25 non-null float64
gdpPercap_1972            25 non-null float64
gdpPercap_1977            25 non-null float64
gdpPercap_1982            25 non-null float64
gdpPercap_1987            25 non-null float64
gdpPercap_1992            25 non-null float64
gdpPercap_1997            25 non-null float64
gdpPercap_2002            25 non-null float64
gdpPercap_2007            25 non-null float64
dtypes: float64(12), object(1)
memory usage: 3.4+ KB
```

```
In [136]: americas.columns
```

```
Out[136]: Index(['continent', 'gdpPercap_1952', 'gdpPercap_1957', 'gdpPercap_1962',
                  'gdpPercap_1967', 'gdpPercap_1972', 'gdpPercap_1977', 'gdpPercap_1982',
                  'gdpPercap_1987', 'gdpPercap_1992', 'gdpPercap_1997', 'gdpPercap_2002',
                  'gdpPercap_2007'],
              dtype='object', length=13)
```

```

        'gdpPercap_1987', 'gdpPercap_1992', 'gdpPercap_1997', 'gdpPercap_2002',
        'gdpPercap_2007'],
        dtype='object')

```

2.

A. Transpose americas, and

B. save to americasT_summ_stats.csv

In [154]: `americas.T`

Out[154]:

country	Argentina	Bolivia	Brazil	Canada	Chile	Colombia	Cos Ri
continent	Americas	Americas	Americas	Americas	Americas	Americas	Americ
gdpPercap_1952	5911.32	2677.33	2108.94	11367.2	3939.98	2144.12	2627.0
gdpPercap_1957	6856.86	2127.69	2487.37	12490	4315.62	2323.81	2990.0

gdpPercap_1962	7133.17	2180.97	3336.59	13462.5	4519.09	2492.35	3460.9
gdpPercap_1967	8052.95	2586.89	3429.86	16076.6	5106.65	2678.73	4161.7
gdpPercap_1972	9443.04	2980.33	4985.71	18970.6	5494.02	3264.66	5118.1
gdpPercap_1977	10079	3548.1	6660.12	22090.9	4756.76	3815.81	5926.8
gdpPercap_1982	8997.9	3156.51	7030.84	22898.8	5095.67	4397.58	5262.7
gdpPercap_1987	9139.67	2753.69	7807.1	26626.5	5547.06	4903.22	5629.9
gdpPercap_1992	9308.42	2961.7	6950.28	26342.9	7596.13	5444.65	6160.4
gdpPercap_1997	10967.3	3326.14	7957.98	28954.9	10118.1	6117.36	6677.0
gdpPercap_2002	8797.64	3413.26	8131.21	33329	10778.8	5755.26	7723.4
gdpPercap_2007	12779.4	3822.14	9065.8	36319.2	13171.6	7006.58	9645.0

13 rows × 25 columns

```
In [166]: americas.T.describe().to_csv('americasT_summ_stats.csv')
```

5. Slicing, and selecting values

DataFrame is the way Pandas represents a table.

Series is the data-structure Pandas use to represent a column.

Selecting values, .loc[]

```
In [242]: americas.head()
```

Out[242]:

	continent	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap
country					
Argentina	Americas	5911.315053	6856.856212	7133.166023	8052.95302
Bolivia	Americas	2677.326347	2127.686326	2180.972546	2586.88605
Brazil	Americas	2108.944355	2487.365989	3336.585802	3429.86435
Canada	Americas	11367.161120	12489.950060	13462.485550	16076.5880
Chile	Americas	3939.978789	4315.622723	4519.094331	5106.65431

```
In [244]: americas.loc["Brazil",:]
```

```
Out[244]: continent      Americas
gdpPercap_1952      2108.94
gdpPercap_1957      2487.37
gdpPercap_1962      3336.59
gdpPercap_1967      3429.86
gdpPercap_1972      4985.71
gdpPercap_1977      6660.12
gdpPercap_1982      7030.84
gdpPercap_1987      7807.1
gdpPercap_1992      6950.28
gdpPercap_1997      7957.98
gdpPercap_2002      8131.21
gdpPercap_2007      9065.8
Name: Brazil, dtype: object
```

```
In [205]: americas.loc[:, "gdpPercap_1952"]
```

```
Out[205]: country
Argentina      5911.315053
Bolivia        2677.326347
Brazil         2108.944355
Canada         11367.161120
Chile          3939.978789
Colombia       2144.115096
Costa Rica     2627.009471
Cuba           5586.538780
Dominican Republic 1397.717137
Ecuador        3522.110717
El Salvador    3048.302900
Guatemala      2428.237769
Haiti          1840.366939
Honduras       2194.926204
Jamaica        2898.530881
Mexico         3478.125529
Nicaragua      3112.363948
Panama         2480.380334
~
```

```

Paraguay          1952.308701
Peru              3758.523437
Puerto Rico      3081.959785
Trinidad and Tobago 3023.271928
United States     13990.482080
Uruguay           5716.766744
Venezuela         7689.799761
Name: gdpPercap_1952, dtype: float64

```

```
In [222]: americas.loc['Ecuador':'Jamaica',:]
```

```
Out[222]:
```

	continent	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967
country					
Ecuador	Americas	3522.110717	3780.546651	4086.114078	4579.074215
El Salvador	Americas	3048.302900	3421.523218	3776.803627	4358.595393
Guatemala	Americas	2428.237769	2617.155967	2750.364446	3242.531147
Haiti	Americas	1840.366939	1726.887882	1796.589032	1452.057666
Honduras	Americas	2194.926204	2220.487682	2291.156835	2538.269358
Jamaica	Americas	2898.530881	4756.525781	5246.107524	6124.703451

```
In [224]: americas.loc['Ecuador':'Jamaica','gdpPercap_1957':'gdpPercap_1972']
```

```
Out[224]:
```

	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPercap_1972
country				
Ecuador	3780.546651	4086.114078	4579.074215	5280.994710
El Salvador	3421.523218	3776.803627	4358.595393	4520.246008
Guatemala	2617.155967	2750.364446	3242.531147	4031.408271
Haiti	1726.887882	1796.589032	1452.057666	1654.456946
Honduras	2220.487682	2291.156835	2538.269358	2529.842345
Jamaica	4756.525781	5246.107524	6124.703451	7433.889293

```
In [234]: americas.loc['Ecuador':'Jamaica','gdpPercap_1957':'gdpPercap_1972'].min()
```

```

Out[234]: gdpPercap_1957    1726.887882
gdpPercap_1962    1796.589032
gdpPercap_1967    1452.057666
gdpPercap_1972    1654.456946
dtype: float64

```

```
In [235]: americas.loc['Ecuador':'Jamaica','gdpPercap_1957':'gdpPercap_1972'].max()
```

```
Out[235]: gdpPercap_1957    4756.525781
gdpPercap_1962    5246.107524
gdpPercap_1967    6124.703451
gdpPercap_1972    7433.889293
dtype: float64
```

```
In [237]: americas.loc['Ecuador':'Jamaica','gdpPercap_1957':'gdpPercap_1972'].T
```

```
Out[237]:
```

country	Ecuador	El Salvador	Guatemala	Haiti	Honduras	
gdpPercap_1957	3780.546651	3421.523218	2617.155967	1726.887882	2220.487682	4
gdpPercap_1962	4086.114078	3776.803627	2750.364446	1796.589032	2291.156835	5
gdpPercap_1967	4579.074215	4358.595393	3242.531147	1452.057666	2538.269358	6
gdpPercap_1972	5280.994710	4520.246008	4031.408271	1654.456946	2529.842345	7

```
In [239]: americas.loc['Ecuador':'Jamaica','gdpPercap_1957':'gdpPercap_1972'].T.min()
```

```
Out[239]: country
Ecuador    3780.546651
El Salvador 3421.523218
Guatemala  2617.155967
Haiti       1452.057666
Honduras    2220.487682
Jamaica     4756.525781
dtype: float64
```

```
In [240]: americas.loc['Ecuador':'Jamaica','gdpPercap_1957':'gdpPercap_1972'].T.describe()
```

```
Out[240]:
```

country	Ecuador	El Salvador	Guatemala	Haiti	Honduras	Jamaica
count	4.000000	4.000000	4.000000	4.000000	4.000000	4.000000
mean	4431.682413	4019.292061	3160.364958	1657.497881	2394.939055	5890.3065
std	654.841196	510.623749	639.959416	148.746148	163.244708	1174.4560
min	3780.546651	3421.523218	2617.155967	1452.057666	2220.487682	4756.5257
25%	4009.722221	3687.983525	2717.062326	1603.857126	2273.489547	5123.7120
50%	4332.594146	4067.699510	2996.447796	1690.672414	2410.499590	5685.4054
75%	4754.554339	4399.008047	3439.750428	1744.313169	2531.949098	6451.9999
max	5280.994710	4520.246008	4031.408271	1796.589032	2538.269358	7433.8892

```
In [241]: americas.loc['Ecuador':'Jamaica','gdpPercap_1957':'gdpPercap_1972'].describe()
```

```
Out[241]:
```

	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPercap_1972
count	6.000000	6.000000	6.000000	6.000000
mean	3087.187863	3324.522590	3715.871872	4241.806262
std	1113.708296	1281.339276	1655.073126	2050.185340
min	1726.887882	1796.589032	1452.057666	1654.456946
25%	2319.654753	2405.958738	2714.334805	2905.233826
50%	3019.339592	3263.584037	3800.563270	4275.827140
75%	3690.790793	4008.786465	4523.954509	5090.807534
max	4756.525781	5246.107524	6124.703451	7433.889293

Selecting values, .iloc[]

```
In [257]: americas.iloc[9:16,2:6]
```

```
Out[257]:
```

	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPercap_1972
country				
Ecuador	3780.546651	4086.114078	4579.074215	5280.994710
El Salvador	3421.523218	3776.803627	4358.595393	4520.246008
Guatemala	2617.155967	2750.364446	3242.531147	4031.408271
Haiti	1726.887882	1796.589032	1452.057666	1654.456946
Honduras	2220.487682	2291.156835	2538.269358	2529.842345

Jamaica	4756.525781	5246.107524	6124.703451	7433.889293
Mexico	4131.546641	4581.609385	5754.733883	6809.406690

```
In [262]: americas.iloc[:,1:].head() # omits the 'continent' column
```

```
Out[262]:
```

	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPercap_1972
country					
Argentina	5911.315053	6856.856212	7133.166023	8052.953021	9446.105061
Bolivia	2677.326347	2127.686326	2180.972546	2586.886053	2980.951861
Brazil	2108.944355	2487.365989	3336.585802	3429.864357	4987.677044
Canada	11367.161120	12489.950060	13462.485550	16076.588030	18979.371261
Chile	3939.978789	4315.622723	4519.094331	5106.654313	5494.511815

```
In [319]: americas.iloc[0:3,1:3] # 1:3 -- 1 omits the 'continent' column
```

```
Out[319]:
```

	gdpPercap_1952	gdpPercap_1957
country		
Argentina	5911.315053	6856.856212
Bolivia	2677.326347	2127.686326
Brazil	2108.944355	2487.365989

Observation:

1:3, omits the final index (i.e. index 3) in the range provided, while a named slice, 'gdpPercap_1952':'gdpPercap_1962', includes the final element.

```
In [315]: americas.loc['Argentina':'Brazil','gdpPercap_1952':'gdpPercap_1962']
```

```
Out[315]:
```

	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962
country			
Argentina	5911.315053	6856.856212	7133.166023

Bolivia	2677.326347	2127.686326	2180.972546
Brazil	2108.944355	2487.365989	3336.585802

Selecting data based on value

```
In [285]: subset_americas=americas.loc['Ecuador':'Jamaica','gdpPercap_1957':'gdpPercap_1972']
```

```
In [286]: print('\t--> Subset of Americas:\n\n', subset_americas,'\n\n')
print('\t--> Where are values larger than 3,500 \n\n',subset_americas
>=3500)
```

--> Subset of Americas:

	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPer
cap_1972				
country				
Ecuador	3780.546651	4086.114078	4579.074215	5280
.994710				
El Salvador	3421.523218	3776.803627	4358.595393	4520
.246008				
Guatemala	2617.155967	2750.364446	3242.531147	4031
.408271				
Haiti	1726.887882	1796.589032	1452.057666	1654
.456946				
Honduras	2220.487682	2291.156835	2538.269358	2529
.842345				
Jamaica	4756.525781	5246.107524	6124.703451	7433
.889293				

--> Where are values larger than 3,500

	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPer
cap_1972				
country				
Ecuador	True	True	True	
True				
El Salvador	False	True	True	
True				
Guatemala	False	False	False	
True				
Haiti	False	False	False	
False				
Honduras	False	False	False	
False				
Jamaica	True	True	True	
True				

Mask values

```
In [290]: subset_americas[subset_americas >=3500]
```

```
Out[290]:
```

	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPercap_1972
country				
Ecuador	3780.546651	4086.114078	4579.074215	5280.994710
El Salvador	NaN	3776.803627	4358.595393	4520.246008
Guatemala	NaN	NaN	NaN	4031.408271
Haiti	NaN	NaN	NaN	NaN
Honduras	NaN	NaN	NaN	NaN
Jamaica	4756.525781	5246.107524	6124.703451	7433.889293

```
In [296]: mask=subset_americas >=3500
subset_americas[mask] # another way to do it
```

```
Out[296]:
```

	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPercap_1972
country				
Ecuador	3780.546651	4086.114078	4579.074215	5280.994710
El Salvador	NaN	3776.803627	4358.595393	4520.246008
Guatemala	NaN	NaN	NaN	4031.408271
Haiti	NaN	NaN	NaN	NaN
Honduras	NaN	NaN	NaN	NaN
Jamaica	4756.525781	5246.107524	6124.703451	7433.889293

```
In [293]: subset_americas[mask].describe()
```

```
Out[293]:
```

	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPercap_1972
--	-----------------------	-----------------------	-----------------------	-----------------------

count	2.000000	3.000000	3.000000	4.000000
mean	4268.536216	4369.675076	5020.791020	5316.634571
std	690.121461	774.608685	962.351142	1502.229359
min	3780.546651	3776.803627	4358.595393	4031.408271
25%	4024.541433	3931.458852	4468.834804	4398.036574
50%	4268.536216	4086.114078	4579.074215	4900.620359
75%	4512.530999	4666.110801	5351.888833	5819.218356
max	4756.525781	5246.107524	6124.703451	7433.889293

```
In [295]: americas.loc['Ecuador':'Jamaica','gdpPercap_1957':'gdpPercap_1972'].describe() # compare to the original
```

```
Out[295]:
```

	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPercap_1972
count	6.000000	6.000000	6.000000	6.000000
mean	3087.187863	3324.522590	3715.871872	4241.806262
std	1113.708296	1281.339276	1655.073126	2050.185340
min	1726.887882	1796.589032	1452.057666	1654.456946
25%	2319.654753	2405.958738	2714.334805	2905.233826
50%	3019.339592	3263.584037	3800.563270	4275.827140
75%	3690.790793	4008.786465	4523.954509	5090.807534
max	4756.525781	5246.107524	6124.703451	7433.889293

Cheeky Exercise

3.

A. Import `gapminder_gdp_europe.csv` as `europe`,

B. slice to keep Italy to Poland, and

C. also slice 1962 to 1972

D. Mask values less than 8000

```
In [297]: europe=pd.read_csv('gapminder_gdp_europe.csv', index_col='country')
```

```
In [300]: subset_europe=europe.loc['Italy':'Poland','gdpPercap_1962':'gdpPercap_1972']
subset_europe
```

Out[300]:

	gdpPercap_1962	gdpPercap_1967	gdpPercap_1972
country			
Italy	8243.582340	10022.401310	12269.273780
Montenegro	4649.593785	5907.850937	7778.414017
Netherlands	12790.849560	15363.251360	18794.745670
Norway	13450.401510	16361.876470	18965.055510
Poland	5338.752143	6557.152776	8006.506993

```
In [307]: masker=subset_europe>8000
subset_europe[masker]
```

Out[307]:

	gdpPercap_1962	gdpPercap_1967	gdpPercap_1972
country			
Italy	8243.58234	10022.40131	12269.273780
Montenegro	NaN	NaN	NaN
Netherlands	12790.84956	15363.25136	18794.745670
Norway	13450.40151	16361.87647	18965.055510
Poland	NaN	NaN	8006.506993

```
In [311]: europe.mean()
```

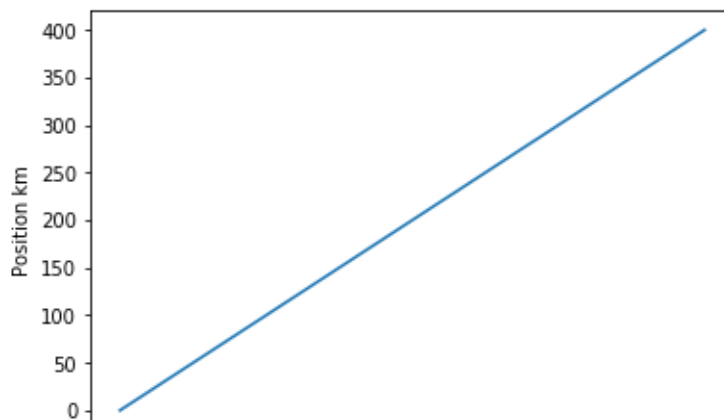
```
Out[311]: gdpPercap_1952    5661.057435
          gdpPercap_1957    6963.012816
          gdpPercap_1962    8365.486814
          gdpPercap_1967   10143.823757
          gdpPercap_1972   12479.575246
          gdpPercap_1977   14283.979110
          gdpPercap_1982   15617.896551
          gdpPercap_1987   17214.310727
          gdpPercap_1992   17061.568084
          gdpPercap_1997   19076.781802
          gdpPercap_2002   21711.732422
          gdpPercap_2007   25054.481636
          dtype: float64
```

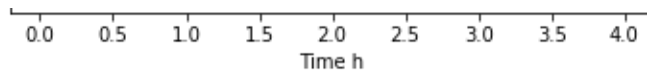
6. Plotting!

```
In [320]: %matplotlib inline
          import matplotlib.pyplot as plt
```

```
In [322]: time=[0,1,2,3,4]
          position=[0,100,200,300,400]
          plt.plot(time,position)
          plt.xlabel('Time h')
          plt.ylabel('Position km')
```

```
Out[322]: Text(0, 0.5, 'Position km')
```





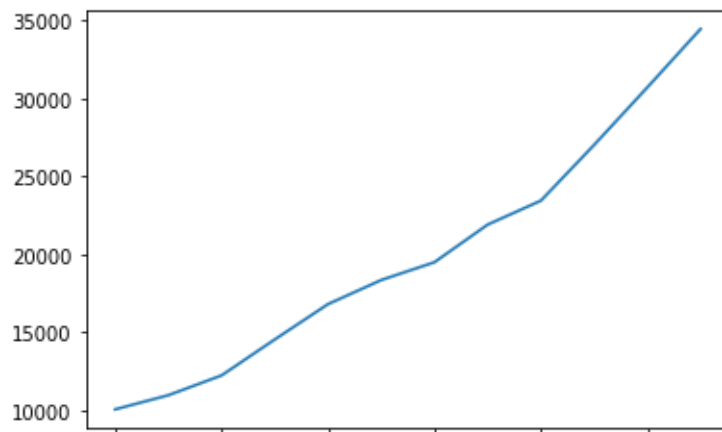
In [350]: data

Out[350]:

	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpP
country					
Australia	10039.59564	10949.64959	12217.22686	14526.12465	16780
New Zealand	10556.57566	12247.39532	13175.67800	14463.91893	16040

In [351]: data.loc['Australia'].plot()

Out[351]: <matplotlib.axes._subplots.AxesSubplot at 0x1185d90b8>



In [355]: years = data.columns.str.strip('gdpPercap_')
data.columns=years.astype(int)

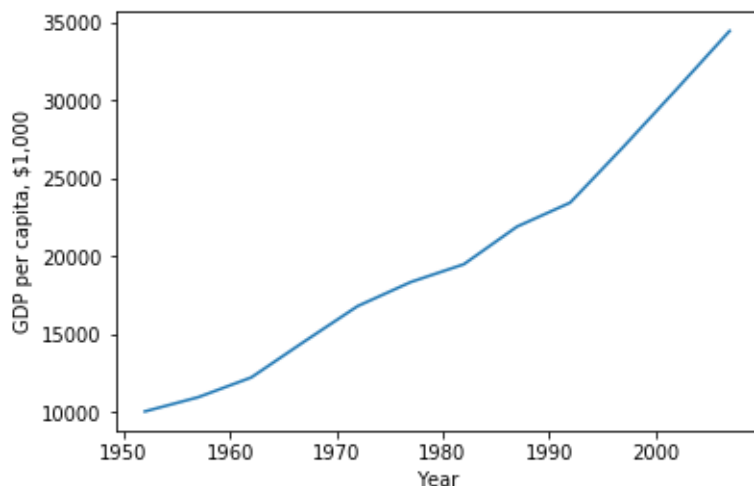
In [357]: data.columns

Out[357]: Int64Index([1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, 2002],

```
2007],  
dtype='int64')
```

```
In [362]: data.loc['Australia'].plot()  
plt.xlabel('Year')  
plt.ylabel('GDP per capita, $1,000')
```

```
Out[362]: Text(0, 0.5, 'GDP per capita, $1,000')
```



```
In [370]: years
```

```
Out[370]: Index(['1952', '1957', '1962', '1967', '1972', '1977', '1982', '1987',  
'1992',  
'1997', '2002', '2007'],  
dtype='object')
```

```
In [372]: gdp_au=data.loc['Australia']  
gdp_au
```

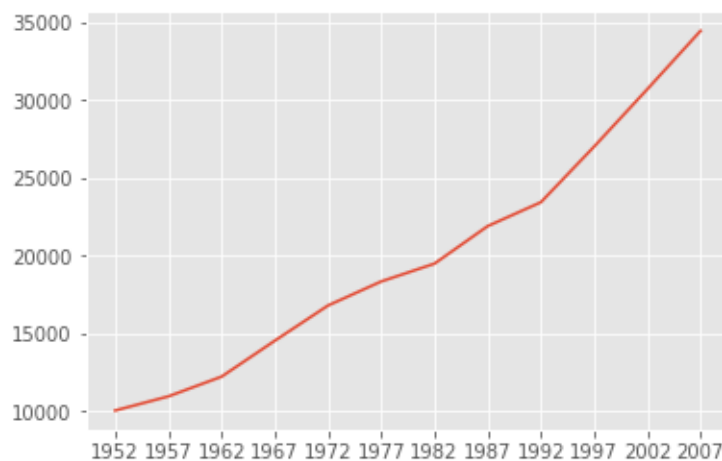


```
Out[372]: 1952    10039.59564
          1957    10949.64959
          1962    12217.22686
          1967    14526.12465
          1972    16788.62948
          1977    18334.19751
          1982    19477.00928
          1987    21888.88903
          1992    23424.76683
          1997    26997.93657
          2002    30687.75473
          2007    34435.36744
          Name: Australia, dtype: float64
```

Another way to plot data

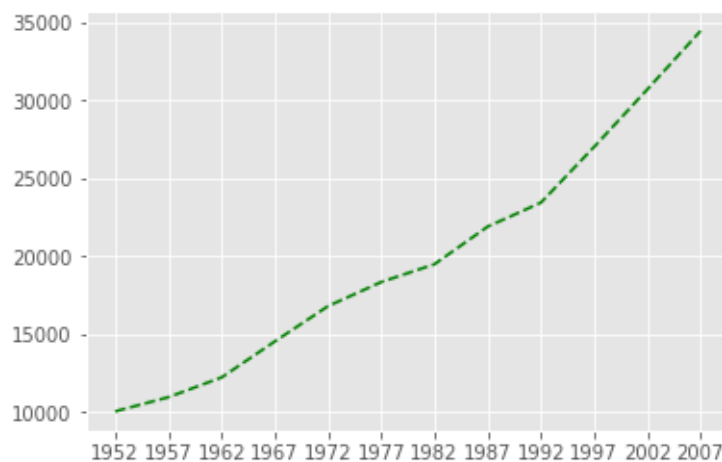
```
In [376]: plt.plot(years,gdp_au) ## add one at each time each time!
```

```
Out[376]: [<matplotlib.lines.Line2D at 0x11d223518>]
```



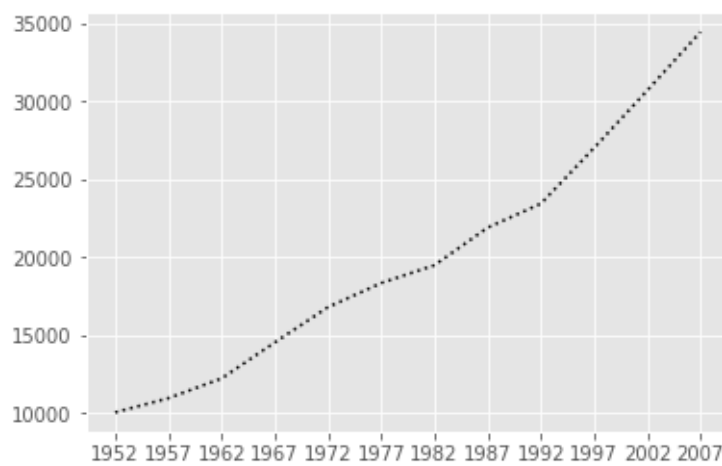
```
In [373]: plt.plot(years, gdp_au, 'g--') # g == green; -- dashed line style
```

```
Out[373]: [<matplotlib.lines.Line2D at 0x11cf3c9e8>]
```



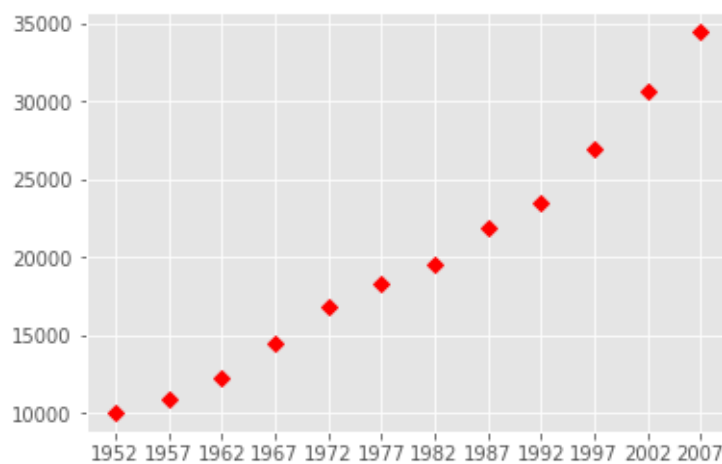
```
In [379]: plt.plot(years, gdp_au, 'k:')
```

```
Out[379]: [<matplotlib.lines.Line2D at 0x11d397da0>]
```



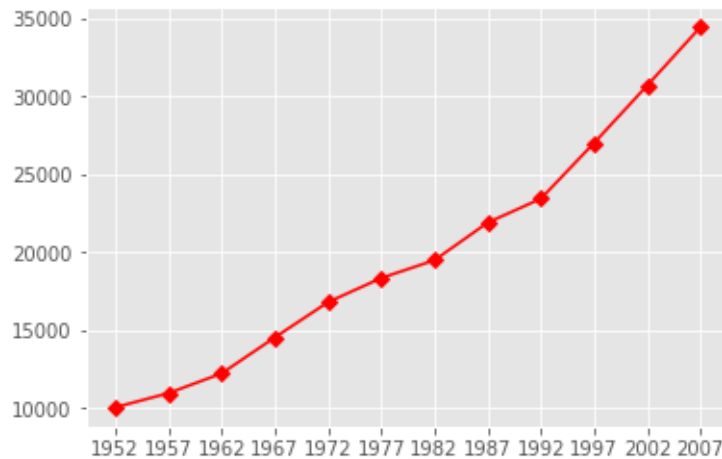
```
In [384]: plt.plot(years, gdp_au, 'rD')
```

```
Out[384]: [<matplotlib.lines.Line2D at 0x11c3fb3c8>]
```



```
In [385]: plt.plot(years, gdp_aus, 'rD-')
```

```
Out[385]: [<matplotlib.lines.Line2D at 0x11c60aac8>]
```



```
In [407]: plt.plot?
```

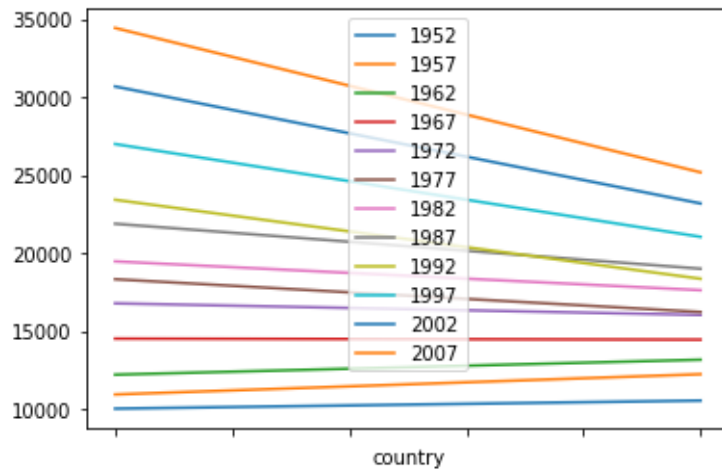
```
In [363]: data
```

```
Out[363]:
```

	1952	1957	1962	1967	1972	1
country						
Australia	10039.59564	10949.64959	12217.22686	14526.12465	16788.62948	18334.19
New Zealand	10556.57566	12247.39532	13175.67800	14463.91893	16046.03728	16233.71

```
In [366]: data.plot()
```

```
Out[366]: <matplotlib.axes._subplots.AxesSubplot at 0x11c82f748>
```



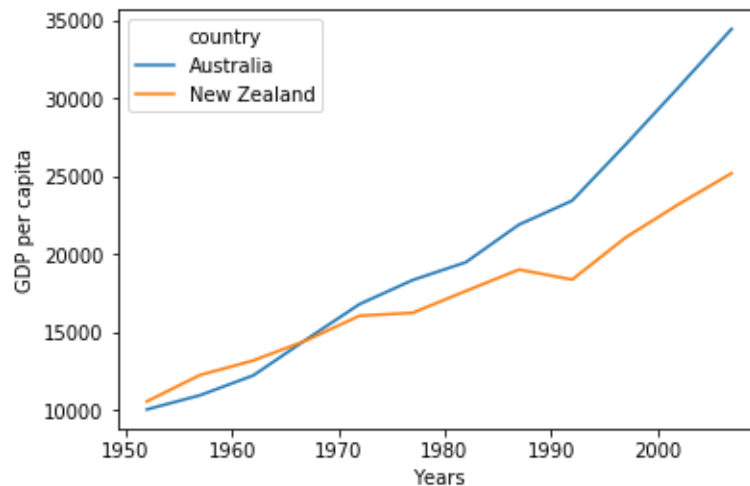
```
In [364]: data.T
```

```
Out[364]:
```

country	Australia	New Zealand
1952	10039.59564	10556.57566
1957	10949.64959	12247.39532
1962	12217.22686	13175.67800
1967	14526.12465	14463.91893
1972	16788.62948	16046.03728
1977	18334.19751	16233.71770
1982	19477.00928	17632.41040
1987	21888.88903	19007.19129
1992	23424.76683	18363.32494
1997	26997.93657	21050.41377
2002	30687.75473	23189.80135
2007	34435.36744	25185.00911

```
In [367]: data_transposed=data.T
plt.ylabel('GDP per capita')
plt.xlabel('Years')
```

```
Out[367]: Text(0.5, 0, 'Years')
```



```
In [386]: gdp_aus
```

```
Out[386]: 1952    10039.59564
          1957    10949.64959
          1962    12217.22686
          1967    14526.12465
          1972    16788.62948
          1977    18334.19751
          1982    19477.00928
          1987    21888.88903
          1992    23424.76683
          1997    26997.93657
          2002    30687.75473
          2007    34435.36744
          Name: Australia, dtype: float64
```

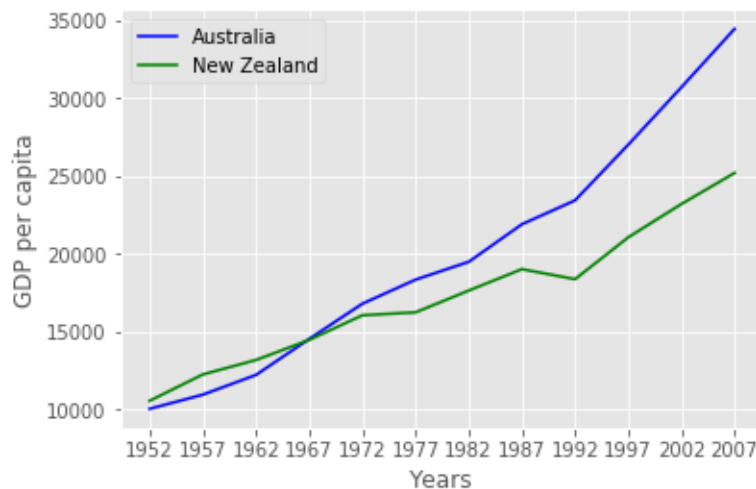
```
In [388]: gdp_nz=data.loc['New Zealand']
          gdp_nz
```

```
Out[388]: 1952    10556.57566
          1957    12247.39532
          1962    13175.67800
          1967    14463.91893
          1972    16046.03728
          1977    16233.71770
```

```
1982    17632.41040
1987    19007.19129
1992    18363.32494
1997    21050.41377
2002    23189.80135
2007    25185.00911
Name: New Zealand, dtype: float64
```

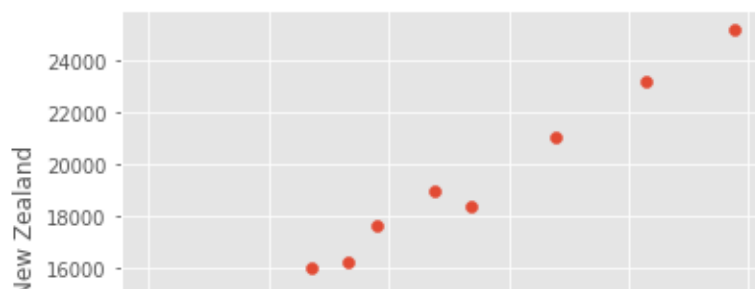
```
In [393]: plt.plot(years, gdp_au, 'b-', label='Australia')
plt.plot(years, gdp_nz, 'g-', label='New Zealand')
plt.legend(loc='upper left')
plt.xlabel('Years')
plt.ylabel('GDP per capita')
```

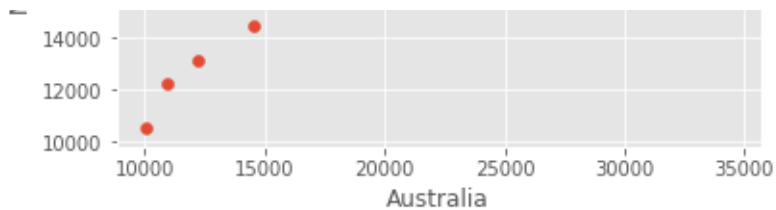
```
Out[393]: Text(0, 0.5, 'GDP per capita')
```



```
In [398]: plt.scatter(gdp_au, gdp_nz)
plt.xlabel('Australia')
plt.ylabel('New Zealand')
```

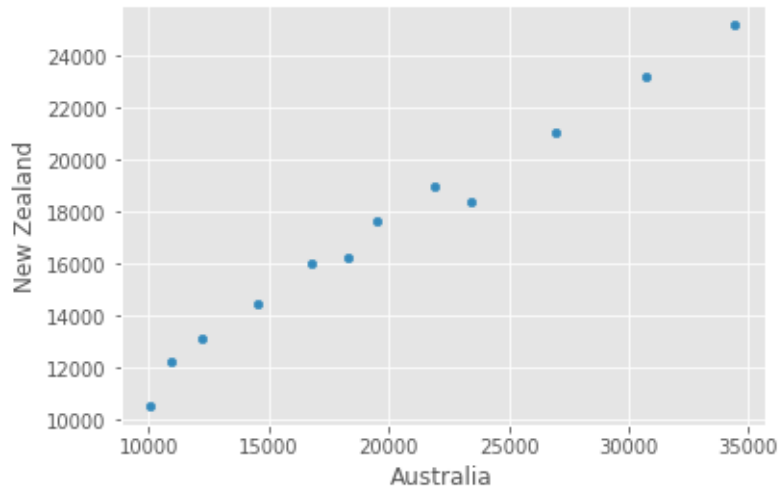
```
Out[398]: Text(0, 0.5, 'New Zealand')
```





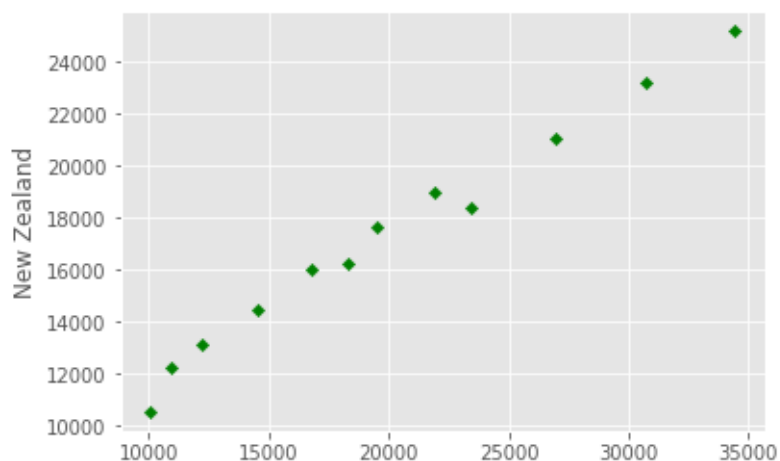
```
In [410]: # another way to scatter plot
data.T.plot.scatter(x='Australia',y='New Zealand')
```

```
Out[410]: <matplotlib.axes._subplots.AxesSubplot at 0x11e815278>
```



```
In [411]: # another way to scatter plot
data.T.plot.scatter(x='Australia',y='New Zealand', c='green', marker='D')
```

```
Out[411]: <matplotlib.axes._subplots.AxesSubplot at 0x11e7bca58>
```

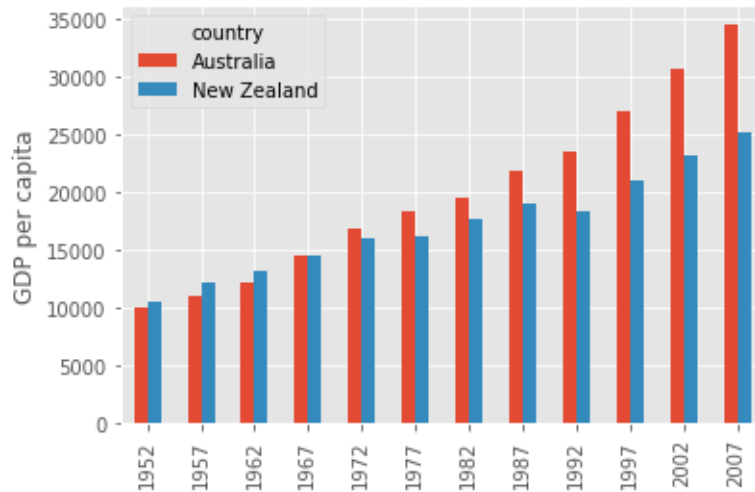


Australia

```
In [400]: plt.scatter?
```

```
In [368]: plt.style.use('ggplot')
data.T.plot(kind='bar')
plt.ylabel('GDP per capita')
```

```
Out[368]: Text(0, 0.5, 'GDP per capita')
```



Cheeky Exercise

4.

A. Use europe

B. Plot a line graph with the min, mean, and max GDP per capita over time for all countries

```
In [413]: europe.head()
```

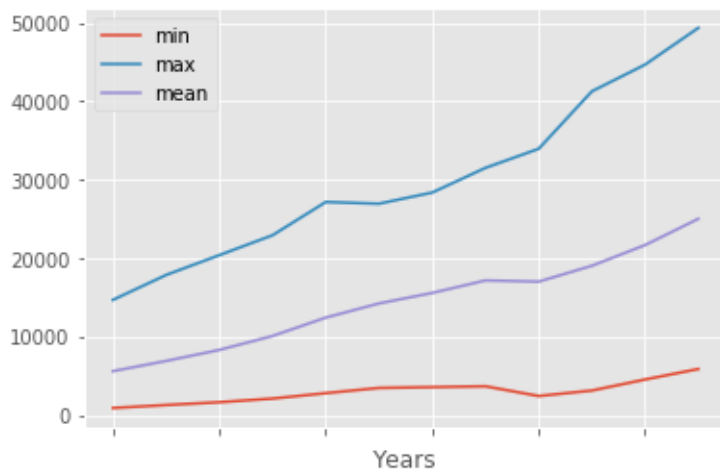
```
Out[413]:
```

	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	g
country					
Albania	1601.056136	1942.284244	2312.888958	2760.196931	3
Austria	6137.076492	8842.598030	10750.721110	12834.602400	1

Belgium	8343.105127	9714.960623	10991.206760	13149.041190	1
Bosnia and Herzegovina	973.533195	1353.989176	1709.683679	2172.352423	2
Bulgaria	2444.286648	3008.670727	4254.337839	5577.002800	6

```
In [425]: europe.min().plot(label='min')
europe.max().plot(label='max')
europe.mean().plot(label='mean')
plt.legend(loc='best')
plt.xlabel('Years')
```

```
Out[425]: Text(0.5, 0, 'Years')
```



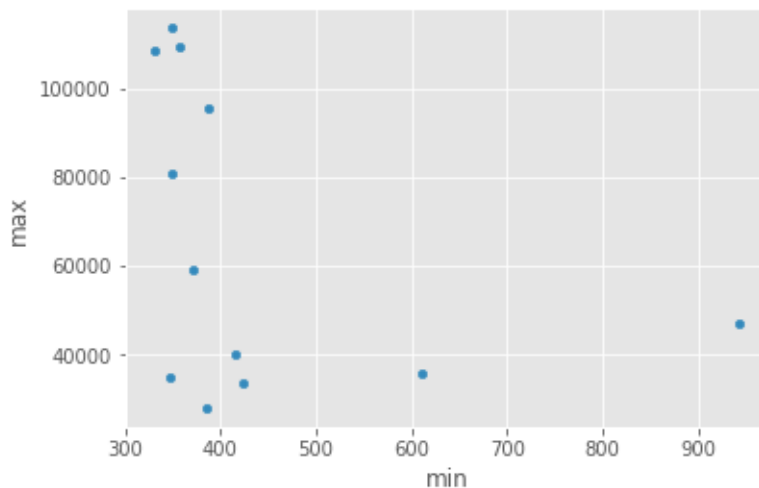
More plotting...

Shows the relationship between min and max GDP in Asian countries for each year. No particular correlations can be seen between the minimum and maximum gdp values year on year. It seems the fortunes of asian countries do not rise and fall together

variability of Asian countries as measured by minimum and maximum GDP.

```
In [426]: asia = pd.read_csv('gapminder_gdp_asia.csv', index_col='country')
asia.describe().T.plot(kind='scatter', x='min', y='max')
```

```
Out[426]: <matplotlib.axes._subplots.AxesSubplot at 0x11f01c208>
```

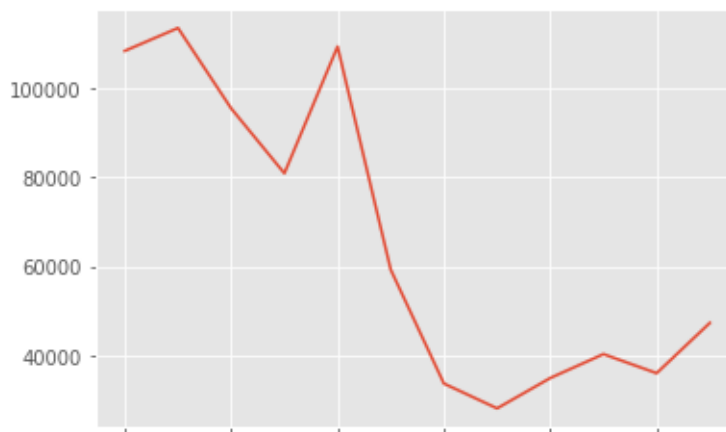


You might note that the variability in the maximum is much higher than that of the minimum. Take a look at the maximum and the max indexes:

Seems the variability in this value is due to a sharp drop after 1972. Some geopolitics at play perhaps? Given the dominance of oil producing countries, maybe the Brent crude index would make an interesting comparison? Whilst Myanmar consistently has the lowest gdp, the highest gdb nation has varied more notably.

```
In [428]: asia.max().plot()
print(asia.idxmax())
print(asia.idxmin())
```

```
gdpPercap_1952      Kuwait
gdpPercap_1957      Kuwait
gdpPercap_1962      Kuwait
gdpPercap_1967      Kuwait
gdpPercap_1972      Kuwait
gdpPercap_1977      Kuwait
gdpPercap_1982      Saudi Arabia
gdpPercap_1987      Kuwait
gdpPercap_1992      Kuwait
gdpPercap_1997      Kuwait
gdpPercap_2002      Singapore
gdpPercap_2007      Kuwait
dtype: object
gdpPercap_1952      Myanmar
gdpPercap_1957      Myanmar
gdpPercap_1962      Myanmar
gdpPercap_1967      Myanmar
gdpPercap_1972      Myanmar
gdpPercap_1977      Myanmar
gdpPercap_1982      Myanmar
gdpPercap_1987      Myanmar
gdpPercap_1992      Myanmar
gdpPercap_1997      Myanmar
gdpPercap_2002      Myanmar
gdpPercap_2007      Myanmar
dtype: object
```



All...

```
In [430]: data_all = pandas.read_csv('gapminder_all.csv', index_col='country')
data_all.head()
```

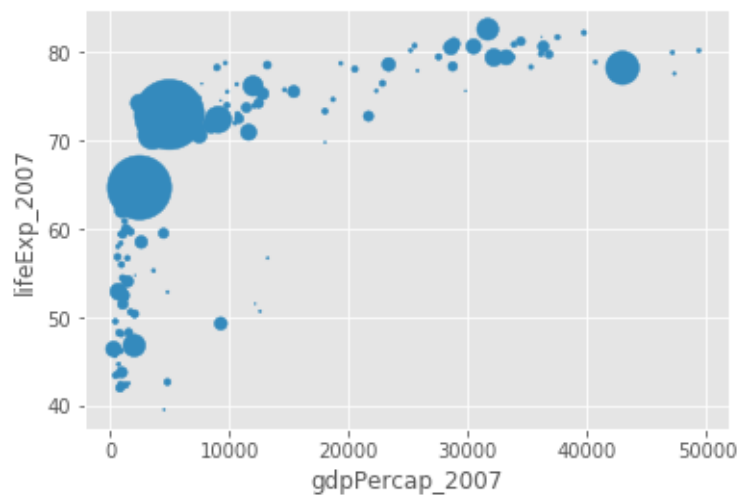
```
Out[430]:
```

	continent	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap
country					
Algeria	Africa	2449.008185	3013.976023	2550.816880	3246.99177
Angola	Africa	3520.610273	3827.940465	4269.276742	5522.77637
Benin	Africa	1062.752200	959.601080	949.499064	1035.83141
Botswana	Africa	851.241141	918.232535	983.653976	1214.70929
Burkina Faso	Africa	543.255241	617.183465	722.512021	794.826560

5 rows × 37 columns

```
In [431]: data_all.plot(kind='scatter', x='gdpPercap_2007', y='lifeExp_2007', s=
data_all['pop_2007']/1e6)
```

```
Out[431]: <matplotlib.axes._subplots.AxesSubplot at 0x11f54c278>
```

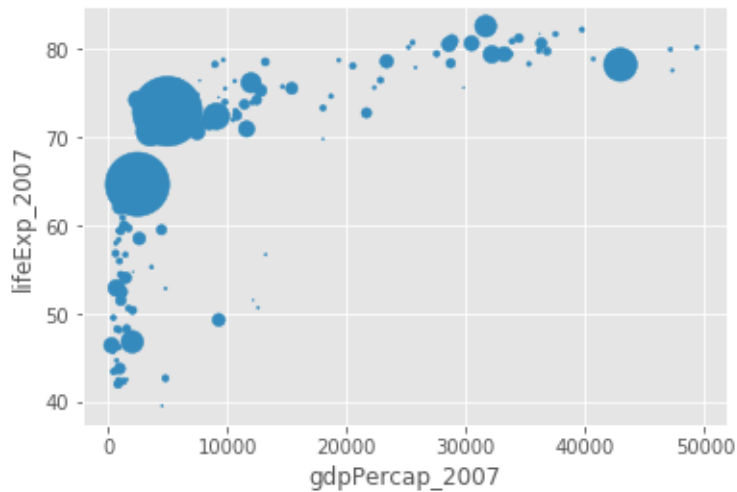


```
In [ ]: help(data_all.plot)
```

7. Saving figures

- save a reference to the current figure in a local variable (with plt.gcf)
- call the savefig class method from that variable.

```
In [446]: data_all.plot(kind='scatter', x='gdpPercap_2007', y='lifeExp_2007', s=
data_all['pop_2007']/1e6)
plt.savefig('my_figure.png')
```



```
In [447]: ls
```

```
americasT_summ_stats.csv    gapminder_gdp_asia.csv
americas_summ_stats.csv    gapminder_gdp_europe.csv
gapminder_all.csv          gapminder_gdp_oceania.csv
gapminder_gdp_africa.csv   my_figure.png
gapminder_gdp_americas.csv oceania_summ_stats.csv
```

```
In [448]: %%bash
open .
```

8. Looping over data sets -- Automating tasks

```
In [450]: import glob
```

```
In [451]: # match files with a pattern == globbing
```

```
In [452]: print(glob.glob('*.csv')) # we have extra files with sums... remove t
hose
```

```
['gapminder_gdp_americas.csv', 'gapminder_gdp_europe.csv', 'gapminder_all.csv', 'gapminder_gdp_oceania.csv', 'americas_summ_stats.csv', 'gapminder_gdp_africa.csv', 'americasT_summ_stats.csv', 'oceania_summ_stats.csv', 'gapminder_gdp_asia.csv']
```

```
In [456]: print(glob.glob('gap*.csv'))
```

```
['gapminder_gdp_americas.csv', 'gapminder_gdp_europe.csv', 'gapminder_all.csv', 'gapminder_gdp_oceania.csv', 'gapminder_gdp_africa.csv', 'gapminder_gdp_asia.csv']
```

```
In [457]: for file in glob.glob('gap*.csv'):
          print(file)
```

```
gapminder_gdp_americas.csv
gapminder_gdp_europe.csv
gapminder_all.csv
gapminder_gdp_oceania.csv
gapminder_gdp_africa.csv
gapminder_gdp_asia.csv
```

```
In [460]: for file in glob.glob('gap*.csv'):
          dataframe=pd.read_csv(file,index_col='country')
          print(file, '\n\n',dataframe.head(),'\n\n')
```

```
gapminder_gdp_americas.csv
```

	continent	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962
\				
country				
Argentina	Americas	5911.315053	6856.856212	7133.166023
Bolivia	Americas	2677.326347	2127.686326	2180.972546
Brazil	Americas	2108.944355	2487.365989	3336.585802
Canada	Americas	11367.161120	12489.950060	13462.485550
Chile	Americas	3939.978789	4315.622723	4519.094331

	gdpPercap_1967	gdpPercap_1972	gdpPercap_1977	gdpPercap_1982
\				
country				
Argentina	8052.953021	9443.038526	10079.026740	8997.8
97412				
Bolivia	2586.886053	2980.331339	3548.097832	3156.5
10452				
Brazil	3429.864357	4985.711467	6660.118654	7030.8
35878				
Canada	16076.588030	18970.570860	22090.883060	22898.7
92140				
Chile	5106.654313	5494.024437	4756.763836	5095.6
65738				

	gdpPercap_1987	gdpPercap_1992	gdpPercap_1997	gdpPercap_2002
\				
country				
Argentina	9139.671389	9308.418710	10967.281950	8797.6
40716				

Bolivia	2753.691490	2961.699694	3326.143191	3413.2
62690				
Brazil	7807.095818	6950.283021	7957.980824	8131.2
12843				
Canada	26626.515030	26342.884260	28954.925890	33328.9
65070				
Chile	5547.063754	7596.125964	10118.053180	10778.7
83850				

	gdpPercap_2007
country	
Argentina	12779.379640
Bolivia	3822.137084
Brazil	9065.800825
Canada	36319.235010
Chile	13171.638850

gapminder_gdp_europe.csv

	gdpPercap_1952	gdpPercap_1957	gdpPercap_1
962 \			
country			
Albania	1601.056136	1942.284244	2312.8889
58			
Austria	6137.076492	8842.598030	10750.7211
10			
Belgium	8343.105127	9714.960623	10991.2067
60			
Bosnia and Herzegovina	973.533195	1353.989176	1709.6836
79			
Bulgaria	2444.286648	3008.670727	4254.3378
39			

	gdpPercap_1967	gdpPercap_1972	gdpPercap_19
77 \			
country			
Albania	2760.196931	3313.422188	3533.0039
10			
Austria	12834.602400	16661.625600	19749.4223
00			
Belgium	13149.041190	16672.143560	19117.9744
80			
Bosnia and Herzegovina	2172.352423	2860.169750	3528.4813
05			
Bulgaria	5577.002800	6597.494398	7612.2404
38			

	gdpPercap_1982	gdpPercap_1987	gdpPercap_19
92 \			
country			
Albania	3630.880722	3738.932735	2497.4379
01			
Austria	21597.083620	23687.826070	27042.0186
80			
Belgium	20979.845890	22525.563080	25575.5706

```

90 -
Bosnia and Herzegovina      4126.613157      4314.114757      2546.7814
45
Bulgaria                     8224.191647      8239.854824      6302.6234
38

```

```

                                gdpPercap_1997  gdpPercap_2002  gdpPercap_20
07
country
Albania                      3193.054604      4604.211737      5937.0295
26
Austria                      29095.920660      32417.607690      36126.4927
00
Belgium                     27561.196630      30485.883750      33692.6050
80
Bosnia and Herzegovina      4766.355904      6018.975239      7446.2988
03
Bulgaria                     5970.388760      7696.777725      10680.7928
20

```

gapminder_all.csv

```

                                continent  gdpPercap_1952  gdpPercap_1957  gdpPercap_1
962 \
country
Algeria                      Africa      2449.008185      3013.976023      2550.8168
80
Angola                      Africa      3520.610273      3827.940465      4269.2767
42
Benin                      Africa      1062.752200      959.601080      949.4990
64
Botswana                   Africa      851.241141      918.232535      983.6539
76
Burkina Faso               Africa      543.255241      617.183465      722.5120
21

```

```

                                gdpPercap_1967  gdpPercap_1972  gdpPercap_1977  gdpPer
cap_1982 \
country
Algeria                      3246.991771      4182.663766      4910.416756      574
5.160213
Angola                      5522.776375      5473.288005      3008.647355      275
6.953672
Benin                      1035.831411      1085.796879      1029.161251      127
7.897616
Botswana                   1214.709294      2263.611114      3214.857818      455
1.142150
Burkina Faso               794.826560      854.735976      743.387037      80
7.198586

```

```

                                gdpPercap_1987  gdpPercap_1992  ...      pop_1962
\
country
Algeria                      5681.358539      5023.216647      ...      11000948.0
3.4320000000000000
Angola                      3430.000000      3607.045000      ...      4000000.0
6.953672
Benin                      1035.831411      1085.796879      ...      1029161251
7.897616
Botswana                   1214.709294      2263.611114      ...      3214857818
1.142150
Burkina Faso               794.826560      854.735976      ...      743387037
7.198586

```


Angola	2430.208311	2021.843083	...	4820013.0
Benin	1225.856010	1191.207681	...	2151895.0
Botswana	6205.883850	7954.111645	...	512764.0
Burkina Faso	912.063142	931.752773	...	4919632.0

	pop_1967	pop_1972	pop_1977	pop_1982	pop_1987
Algeria	12760499.0	14760787.0	17152804.0	20033753.0	23254956.0
Angola	5247469.0	5894858.0	6162675.0	7016384.0	7874230.0
Benin	2427334.0	2761407.0	3168267.0	3641603.0	4243788.0
Botswana	553541.0	619351.0	781472.0	970347.0	1151184.0
Burkina Faso	5127935.0	5433886.0	5889574.0	6634596.0	7586551.0

	pop_1992	pop_1997	pop_2002	pop_2007
Algeria	26298373.0	29072015.0	31287142	33333216
Angola	8735988.0	9875024.0	10866106	12420476
Benin	4981671.0	6066080.0	7026113	8078314
Botswana	1342614.0	1536536.0	1630347	1639131
Burkina Faso	8878303.0	10352843.0	12251209	14326203

[5 rows x 37 columns]

gapminder_gdp_oceania.csv

	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967
Australia	10039.59564	10949.64959	12217.22686	14526.12465
New Zealand	10556.57566	12247.39532	13175.67800	14463.91893

	gdpPercap_1972	gdpPercap_1977	gdpPercap_1982	gdpPercap_1987
Australia	16788.62948	18334.19751	19477.00928	21888.88903
New Zealand	16046.03728	16233.71770	17632.41040	19007.19129

	gdpPercap_1992	gdpPercap_1997	gdpPercap_2002	gdpPercap_2007
Australia	23424.76683	26997.93657	30687.75473	34435.36744
New Zealand	18363.32494	21050.41377	23189.80135	25185.00911

gapminder_gdp_africa.csv

gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	country
2449.008185	3013.976023	2550.816880	324.6991771	Algeria
3520.610273	3827.940465	4269.276742	552.2776375	Angola
1062.752200	959.601080	949.499064	103.5831411	Benin
851.241141	918.232535	983.653976	121.4709294	Botswana
543.255241	617.183465	722.512021	79.4826560	Burkina Faso

gdpPercap_1972	gdpPercap_1977	gdpPercap_1982	gdpPercap_1987	country
4182.663766	4910.416756	5745.160213	568.1358539	Algeria
5473.288005	3008.647355	2756.953672	243.0208311	Angola
1085.796879	1029.161251	1277.897616	122.5856010	Benin
2263.611114	3214.857818	4551.142150	620.5883850	Botswana
854.735976	743.387037	807.198586	91.2063142	Burkina Faso

gdpPercap_1992	gdpPercap_1997	gdpPercap_2002	gdpPercap_2007	country
5023.216647	4797.295051	5288.040382	622.3367465	Algeria
2627.845685	2277.140884	2773.287312	479.7231267	Angola
1191.207681	1232.975292	1372.877931	144.1284873	Benin
7954.111645	8647.142313	11003.605080	1256.9851770	Botswana
931.752773	946.294962	1037.645221	121.7032994	Burkina Faso

gapminder_gdp_asia.csv

gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	country
779.445314	820.853030	853.100710	836.197138	Afghanistan
9867.084765	11635.799450	12753.275140	14804.672700	Bahrain
684.244172	661.637458	686.341554	721.684172	Bangladesh

Bangladesh	368.469286	434.038336	496.913648	523
Cambodia	400.448611	575.987001	487.674018	612
China				

	gdpPercap_1972	gdpPercap_1977	gdpPercap_1982	gdpPercap_1987
Afghanistan	739.981106	786.113360	978.011439	852
Bahrain	18268.658390	19340.101960	19211.147310	18524
Bangladesh	630.233627	659.877232	676.981866	751
Cambodia	421.624026	524.972183	624.475478	683
China	676.900092	741.237470	962.421380	1378

	gdpPercap_1992	gdpPercap_1997	gdpPercap_2002	gdpPercap_2007
Afghanistan	649.341395	635.341351	726.734055	974
Bahrain	19035.579170	20292.016790	23403.559270	29796
Bangladesh	837.810164	972.770035	1136.390430	1391
Cambodia	682.303175	734.285170	896.226015	1713
China	1655.784158	2289.234136	3119.280896	4959

```
In [486]: print('{}\t{}\t{}\t{}'.format('Filename','Minimum','Mean','Maxima'))

for file in glob.glob('gap*.csv'):
    dataframe=pd.read_csv(file,index_col='country')
    subset=dataframe.loc[:, "gdpPercap_1952"]
    print('{}\t{}\t{}\t{}'.format(file.strip('.csv'),subset.min(),subset.mean(),subset.max()))
```

Filename	Minimum	Mean	Maxima
gapminder_gdp_america	1397.7171369999999	4079.0625522000005	
gapminder_gdp_europe	973.5331947999999	5661.0574347599995	
gapminder_all	298.8462121	3725.2760457992963	108382.3529
gapminder_gdp_oceania	10039.595640000001	10298.08565	
gapminder_gdp_africa	298.8462121	1252.5724658211539	
gapminder_gdp_asia	331.0	5195.484004030303	108382.3529

```
In [534]: with open('loop_minT0max.txt','w') as output:
            output.write('{}\t{}\t{}\t{}\n'.format('Filename','Minimum','Mean',
            , 'Maxima'))
            for file in glob.glob('gap*.csv'):
                dataframe=pd.read_csv(file,index_col='country')
                subset=dataframe.loc[:, "gdpPercap_1952"]
                output.write('{}\t{}\t{}\t{}\n'.format(file.strip('.csv'),subset.
                min(),subset.mean(),subset.max()))

            output.close()
```

```
In [535]: ls
```

```
americasT_summ_stats.csv    gapminder_gdp_europe.csv
americas_summ_stats.csv    gapminder_gdp_oceania.csv
gapminder_all.csv          loop_minT0max.txt
gapminder_gdp_africa.csv   my_figure.png
gapminder_gdp_americas.csv oceania_summ_stats.csv
gapminder_gdp_asia.csv
```

```
In [536]: %%bash
open .
```

9. Functions

```
In [581]: def temp_converter(a, b):
            if a == 'C':
                fahr=9/5*b+32
                print('Converting from Celsius to Fahrenheit')
                print('{}C is {}F'.format(b,int(fahr)))
            elif a == 'F':
                celsius=(b-32)*5/9
                print('Converting from Fahrenheit to Celsius')
                print('{}F is {}C'.format(b,int(celsius)))
```

```
In [582]: temp_converter('C',25)
```

```
Converting from Celsius to Fahrenheit
25C is 77F
```

```
In [583]: temp_converter('F',101)
```

```
Converting from Fahrenheit to Celsius
101F is 38C
```