03/19/2019

Software Carpentry Workshop

Day 2 - Python with NATHY

Goals: 1. Use `pandas` (Python library) for statistics 2. Visualize data in publication quality plots 3. Automate tasks

Useful links:

- Software Carpentry Lesson: http://swcarpentry.github.io/python-novice-gapminder/ (http://swcarpentry.github.io/python-novice-gapminder/)
- Gapminder: https://www.gapminder.org (https://www.gapminder.org)
- Pandas Cheat Sheet: https://ugoproto.github.io/ugo_py_doc/Pandas_DataFrame_Notes.pdf (https://ugoproto.github.io/ugo_py_doc/Pandas_DataFrame_Notes.pdf)

1 - Workspace organization...

- Create a folder on your desktop
- Launch Jupyter and navigate to folder
- Download data: 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)

2 - Import data into DataFrames

Information about the data: - Gross domestic product (GDP) per capita or income per person - GDP/capita: per country per year (1952 to 2007)

In [2]:
```
cd data/
```

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

In [23]:
```
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
```

In [4]:
```
import pandas as pd
```

Create a variable that will contain the dataframe.
Choose a name that is informative (df = panda dataframe, oceania = gapminder_gdp_oceania.csv)

In [233]:
```
dfoceania=pd.read_csv('gapminder_gdp_oceania.csv',index_col='country')
```

In [9]:
```
dfoceania.head()
```

Out[9]:

	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 [10]:
```
dfoceania.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 [11]:
```
dfoceania.columns # this is a variable that has the columns names stored in it; don't use () bc it's not a method
```

Out[11]:
```
Index(['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')
```

3 - Obtain summary statistics

In [14]:
```
dfoceania.describe()
```

Out[14]:

	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPerc
count	2.000000	2.000000	2.000000	2.000000	2.00000
mean	10298.085650	11598.522455	12696.452430	14495.021790	16417.33
std	365.560078	917.644806	677.727301	43.986086	525.091
min	10039.595640	10949.649590	12217.226860	14463.918930	16046.03
25%	10168.840645	11274.086022	12456.839645	14479.470360	16231.68
50%	10298.085650	11598.522455	12696.452430	14495.021790	16417.33
75%	10427.330655	11922.958888	12936.065215	14510.573220	16602.98

max	10556.575660	12247.395320	13175.678000	14526.124650	16788.62

4 - Save to file

`to_csv`, create a NEW file name, otherwise will overwrite the files we downloaded!

In [20]:
```
dfoceania.describe().to_csv('oceania_summ_stats.csv')
```

In [24]:
```
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 EXERCISES!

1. Import `gapminder_gdp_americas.csv`, and save summary statistics to file

2. Transpose dataframe, and save to file

SOLUTIONS:

1

In [362]:
```
dfamericas=pd.read_csv('gapminder_gdp_americas.csv', index_col='countr
y')
dfamericas.head()
dfamericas.describe()
dfamericas.describe().to_csv('americas_summ_stats.csv')
```

In [37]:
```
ls
```

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

2

In [38]:
```
dfamericas.T
dfamericas.T.describe().to_csv('americasT_summ_stats.csv')
```

In [39]:
```
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    oceania_summ_stats.csv
gapminder_gdp_americas.csv
```

5 - Slicing and selecting values

Quick recap of list, index and slicing

In [40]:
```
fruits=['orange','strawberry','banana','mango']
```

In [41]:
```
fruits[2]
```

Out[41]:
```
'banana'
```

In [42]:
```
fruits[1:3]
```

Out[42]:
```
['strawberry', 'banana']
```

In [48]:
```
fruits[1:] # nothing means to the end of the list
```

Out[48]:
```
['strawberry', 'banana', 'mango']
```

In [43]:
```
basket=fruits[1:3]
```

In [44]:
```
basket
```

Out[44]:
```
['strawberry', 'banana']
```

Pandas Dataframe: - 2-dimensional representation of a table - Series is the data-structure Pandas use to represent a column.

Because it's a 2D, have to tell which rows and which columns want to select.
Let's use the `americas`

In [45]:
```
dfamericas.head()
```

Out[45]:

country	continent	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_
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

`.loc[]` to select values by the name

`.loc[a:b,i:j]`, where
a and b are the rows/countries
i and j are the columns/years

In [104]:
```
dfamericas.loc['Brazil'] # just the row that has 'Brazil'
```

Out[104]:
```
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 [46]:
```
dfamericas.loc["Brazil":,] # All rows from Brazil to the end, all years
```

Out[46]:

country	continent	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap
Brazil	Americas	2108.944355	2487.365989	3336.585802	3429.8643

Brazil	Americas	2108.944355	2487.365989	3336.585802	3429.8643
Canada	Americas	11367.161120	12489.950060	13462.485550	16076.588
Chile	Americas	3939.978789	4315.622723	4519.094331	5106.6543
Colombia	Americas	2144.115096	2323.805581	2492.351109	2678.7298
Costa Rica	Americas	2627.009471	2990.010802	3460.937025	4161.7278
Cuba	Americas	5586.538780	6092.174359	5180.755910	5690.2680
Dominican Republic	Americas	1397.717137	1544.402995	1662.137359	1653.7230
Ecuador	Americas	3522.110717	3780.546651	4086.114078	4579.0742
El Salvador	Americas	3048.302900	3421.523218	3776.803627	4358.5953
Guatemala	Americas	2428.237769	2617.155967	2750.364446	3242.5311
Haiti	Americas	1840.366939	1726.887882	1796.589032	1452.0576
Honduras	Americas	2194.926204	2220.487682	2291.156835	2538.2693
Jamaica	Americas	2898.530881	4756.525781	5246.107524	6124.7034
Mexico	Americas	3478.125529	4131.546641	4581.609385	5754.7338
Nicaragua	Americas	3112.363948	3457.415947	3634.364406	4643.3935
Panama	Americas	2480.380334	2961.800905	3536.540301	4421.0090
Paraguay	Americas	1952.308701	2046.154706	2148.027146	2299.3763
Peru	Americas	3758.523437	4245.256698	4957.037982	5788.0933
Puerto Rico	Americas	3081.959785	3907.156189	5108.344630	6929.2777
Trinidad and Tobago	Americas	3023.271928	4100.393400	4997.523971	5621.3684
United States	Americas	13990.482080	14847.127120	16173.145860	19530.365
Uruguay	Americas	5716.766744	6150.772969	5603.357717	5444.6196
Venezuela	Americas	7689.799761	9802.466526	8422.974165	9541.4741

In [50]:
```
dfamericas.loc[:,'gdpPercap_2002':] # All countries, years 2002 to end
```

Out[50]:

	gdpPercap_2002	gdpPercap_2007
country		
Argentina	8797.640716	12779.379640
Bolivia	3413.262690	3822.137084
Brazil	8131.212843	9065.800825
Canada	33328.965070	36319.235010
Chile	10778.783850	13171.638850
Colombia	5755.259962	7006.580419
Costa Rica	7723.447195	9645.061420
Cuba	6340.646683	8948.102923
Dominican Republic	4563.808154	6025.374752
Ecuador	5773.044512	6873.262326
El Salvador	5351.568666	5728.353514
Guatemala	4858.347495	5186.050003
Haiti	1270.364932	1201.637154
Honduras	3099.728660	3548.330846
Jamaica	6994.774861	7320.880262
Mexico	10742.440530	11977.574960
Nicaragua	2474.548819	2749.320965
Panama	7356.031934	9809.185636
Paraguay	3783.674243	4172.838464
Peru	5909.020073	7408.905561
Puerto Rico	18855.606180	19328.709010
Trinidad and Tobago	11460.600230	18008.509240
United States	39097.099550	42951.653090
Uruguay	7727.002004	10611.462990

Uruguay	7727.002004	10611.462990
Venezuela	8605.047831	11415.805690

In [52]:
```
dfamericas.loc['Ecuador':'Jamaica','gdpPercap_1997':'gdpPercap_2002']
```

Out[52]:

	gdpPercap_1997	gdpPercap_2002
country		
Ecuador	7429.455877	5773.044512
El Salvador	5154.825496	5351.568666
Guatemala	4684.313807	4858.347495
Haiti	1341.726931	1270.364932
Honduras	3160.454906	3099.728660
Jamaica	7121.924704	6994.774861

We selected part of the data that is of interest...

In [57]:
```
dfamericas.loc['Ecuador':'Jamaica','gdpPercap_1997':'gdpPercap_2002'].
min()
```

Out[57]:
```
gdpPercap_1997    1341.726931
gdpPercap_2002    1270.364932
dtype: float64
```

In [58]:
```
dfamericas.loc['Ecuador':'Jamaica','gdpPercap_1997':'gdpPercap_2002'].
max()
```

Out[58]:
```
gdpPercap_1997    7429.455877
gdpPercap_2002    6994.774861
dtype: float64
```

In [59]:
```
dfamericas.loc['Ecuador':'Jamaica','gdpPercap_1997':'gdpPercap_2002'].
describe()
```

Out[59]:

	gdpPercap_1997	gdpPercap_2002
count	6.000000	6.000000
mean	4815.450287	4557.971521
std	2327.702104	2052.767126
min	1341.726931	1270.364932
25%	3541.419631	3539.383369
50%	4919.569651	5104.958080
75%	6630.149902	5667.675550
max	7429.455877	6994.774861

In [60]:
```
dfamericas.loc['Ecuador':'Jamaica','gdpPercap_1997':'gdpPercap_2002'].
T.describe()
```

Out[60]:

country	Ecuador	El Salvador	Guatemala	Haiti	Honduras	Jamai
count	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000
mean	6601.250194	5253.197081	4771.330651	1306.045932	3130.091783	7058.3497
std	1171.259709	139.118430	123.060401	50.460553	42.939940	89.908516
min	5773.044512	5154.825496	4684.313807	1270.364932	3099.728660	6994.7748
25%	6187.147353	5204.011288	4727.822229	1288.205432	3114.910222	7026.5623
50%	6601.250194	5253.197081	4771.330651	1306.045932	3130.091783	7058.3497
75%	7015.353036	5302.382873	4814.839073	1323.886431	3145.273345	7090.1372
max	7429.455877	5351.568666	4858.347495	1341.726931	3160.454906	7121.9247

`.iloc[]` to select values by the index

`.iloc[a:b,i:j]`, where
a and b are the indexes of rows
i and j are the indexes of columns

In [73]:
```
dfamericas.iloc[9:16,-3:-1]
```

Out[73]:

	gdpPercap_1997	gdpPercap_2002

country		
Ecuador	7429.455877	5773.044512
El Salvador	5154.825496	5351.568666
Guatemala	4684.313807	4858.347495
Haiti	1341.726931	1270.364932
Honduras	3160.454906	3099.728660
Jamaica	7121.924704	6994.774861
Mexico	9767.297530	10742.440530

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.

Selecting data based on value

In [74]:
```
subset_americas=dfamericas.loc['Ecuador':'Jamaica','gdpPercap_1997':]
```

In [75]:
```
subset_americas.head()
```

Out[75]:

	gdpPercap_1997	gdpPercap_2002	gdpPercap_2007
country			
Ecuador	7429.455877	5773.044512	6873.262326
El Salvador	5154.825496	5351.568666	5728.353514
Guatemala	4684.313807	4858.347495	5186.050003
Haiti	1341.726931	1270.364932	1201.637154
Honduras	3160.454906	3099.728660	3548.330846

6 - Mask values

Mask values that are not important, maks outliers, or highlight some values...

2 ways to do it

In [78]:
```
subset_americas
```

Out[78]:

	gdpPercap_1997	gdpPercap_2002	gdpPercap_2007
country			
Ecuador	7429.455877	5773.044512	6873.262326
El Salvador	5154.825496	5351.568666	5728.353514
Guatemala	4684.313807	4858.347495	5186.050003
Haiti	1341.726931	1270.364932	1201.637154
Honduras	3160.454906	3099.728660	3548.330846
Jamaica	7121.924704	6994.774861	7320.880262

In [77]:
```
mask=subset_americas >=3500
subset_americas[mask] # 1way
```

Out[77]:

	gdpPercap_1997	gdpPercap_2002	gdpPercap_2007
country			
Ecuador	7429.455877	5773.044512	6873.262326
El Salvador	5154.825496	5351.568666	5728.353514
Guatemala	4684.313807	4858.347495	5186.050003
Haiti	NaN	NaN	NaN
Honduras	NaN	NaN	3548.330846
Jamaica	7121.924704	6994.774861	7320.880262

In [76]:
```
subset_americas[subset_americas >=3500] # 2way
```

Out[76]:

	gdpPercap_1997	gdpPercap_2002	gdpPercap_2007
country			
Ecuador	7429.455877	5773.044512	6873.262326
El Salvador	5154.825496	5351.568666	5728.353514
Guatemala	4684.313807	4858.347495	5186.050003
Haiti	NaN	NaN	NaN
Honduras	NaN	NaN	3548.330846
Jamaica	7121.924704	6994.774861	7320.880262

Jamaica	7421.024704	6994.774861	7320.880262

In [79]:
```
subset_americas[subset_americas >=3500].describe()
```

Out[79]:

	gdpPercap_1997	gdpPercap_2002	gdpPercap_2007
count	4.000000	4.000000	5.000000
mean	6097.629971	5744.433883	5731.375390
std	1379.526827	913.539507	1491.098916
min	4684.313807	4858.347495	3548.330846
25%	5037.197574	5228.263373	5186.050003
50%	6138.375100	5562.306589	5728.353514
75%	7198.807497	6078.477099	6873.262326
max	7429.455877	6994.774861	7320.880262

In [80]:
```
subset_americas[subset_americas >=3500].T.describe()
```

Out[80]:

country	Ecuador	El Salvador	Guatemala	Haiti	Honduras	Jamaica
count	3.000000	3.000000	3.000000	0.0	1.000000	3.000000
mean	6691.920905	5411.582559	4909.570435	NaN	3548.330846	7145.859942
std	842.963929	291.435837	254.759975	NaN	NaN	164.365005
min	5773.044512	5154.825496	4684.313807	NaN	3548.330846	6994.774861
25%	6323.153419	5253.197081	4771.330651	NaN	3548.330846	7058.349783
50%	6873.262326	5351.568666	4858.347495	NaN	3548.330846	7121.924704
75%	7151.359102	5539.961090	5022.198749	NaN	3548.330846	7221.402483
max	7429.455877	5728.353514	5186.050003	NaN	3548.330846	7320.880262

CHEEKY EXERCISES!

3. Slice `americas` to select `Paraguay` to `Uruguay` and `1962` to `1972`, mask values less than `8,000`, obtain summary statistics by country, and save to file.

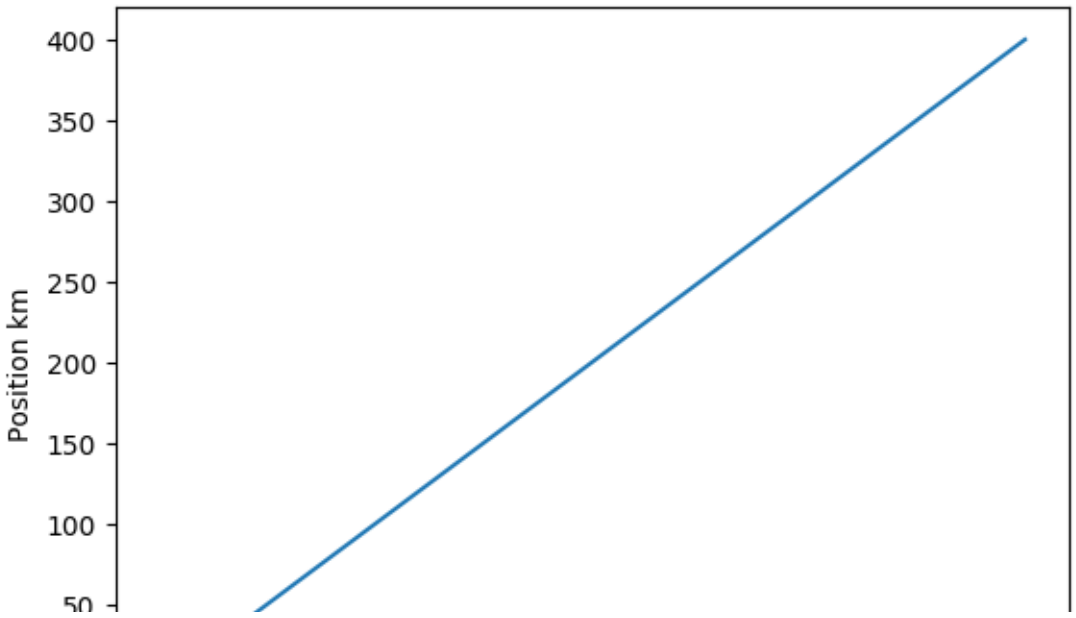
In [85]:
```
subset_americas2=dfamericas.loc['Paraguay':'Uruguay','gdpPercap_1962':
'gdpPercap_1972']
subset_americas2[subset_americas2 >=5000].T.describe().to_csv('america
ssubset2T_summ_stats.csv')
```

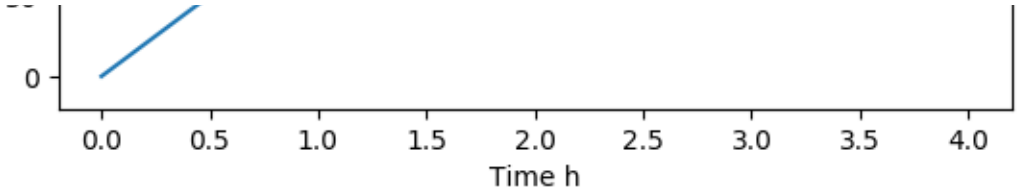
7 - Plots

In [292]:
```
%matplotlib inline
import matplotlib.pyplot as plt
```

In [300]:
```
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[300]: Text(0, 0.5, 'Position km')





Let's use `oceania` df

1. Fix the years (remove gdpPercap_)
2. Replace column headers
3. Plot Australia data
4. Explore customization

In [246]:
```
years=dfoceania.columns.str.strip('gdpPercap_')  # getting just the ye
ars and stripping gdpPercap_
```

In [247]:
```
years
```

Out[247]:
```
Index(['1952', '1957', '1962', '1967', '1972', '1977', '1982', '1987
', '1992',
       '1997', '2002', '2007'],
      dtype='object')
```

In [248]:
```
dfoceania.columns=years.astype(str) # replacing column headers
```

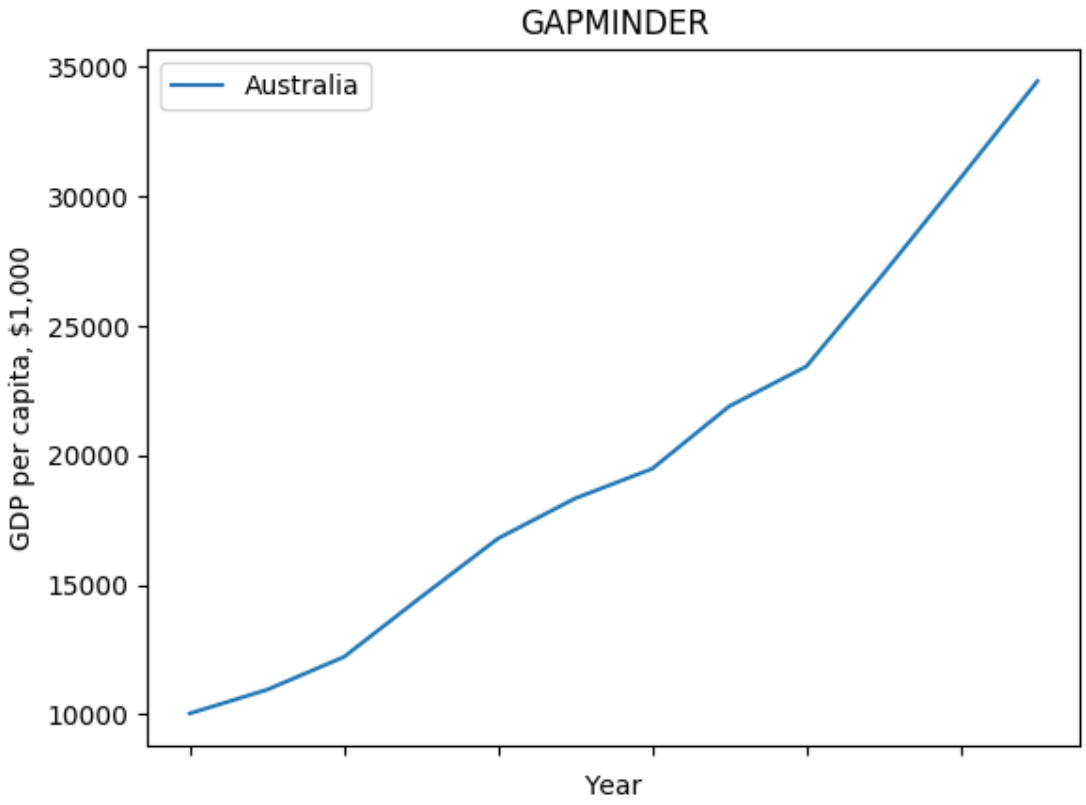
In [193]:
```
dfoceania.columns
```

Out[193]:
```
Index(['1952', '1957', '1962', '1967', '1972', '1977', '1982', '1987
', '1992',
       '1997', '2002', '2007'],
      dtype='object')
```

We'll plot data now. 2 ways to do it.

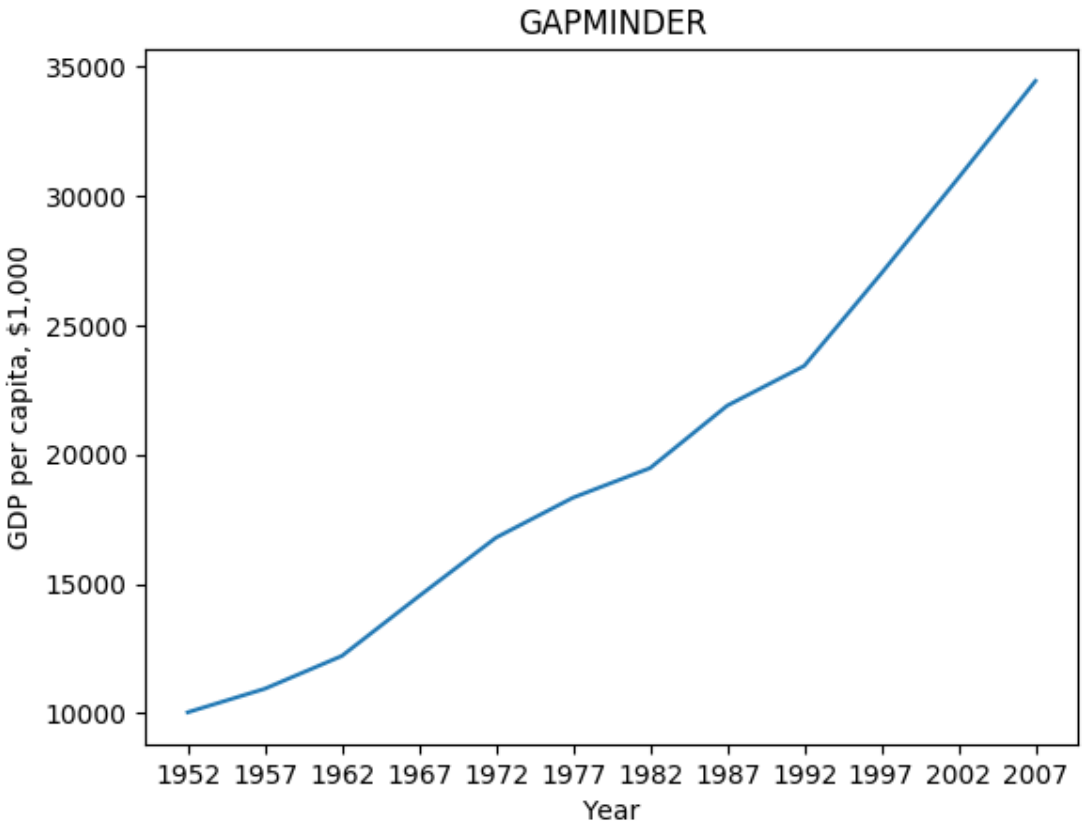
In [301]:
```
# 1 way
dfoceania.loc['Australia'].plot()
plt.xlabel('Year')
plt.ylabel('GDP per capita, $1,000')
plt.title('GAPMINDER')
plt.legend()
```

Out[301]: <matplotlib.legend.Legend at 0x1259069b0>



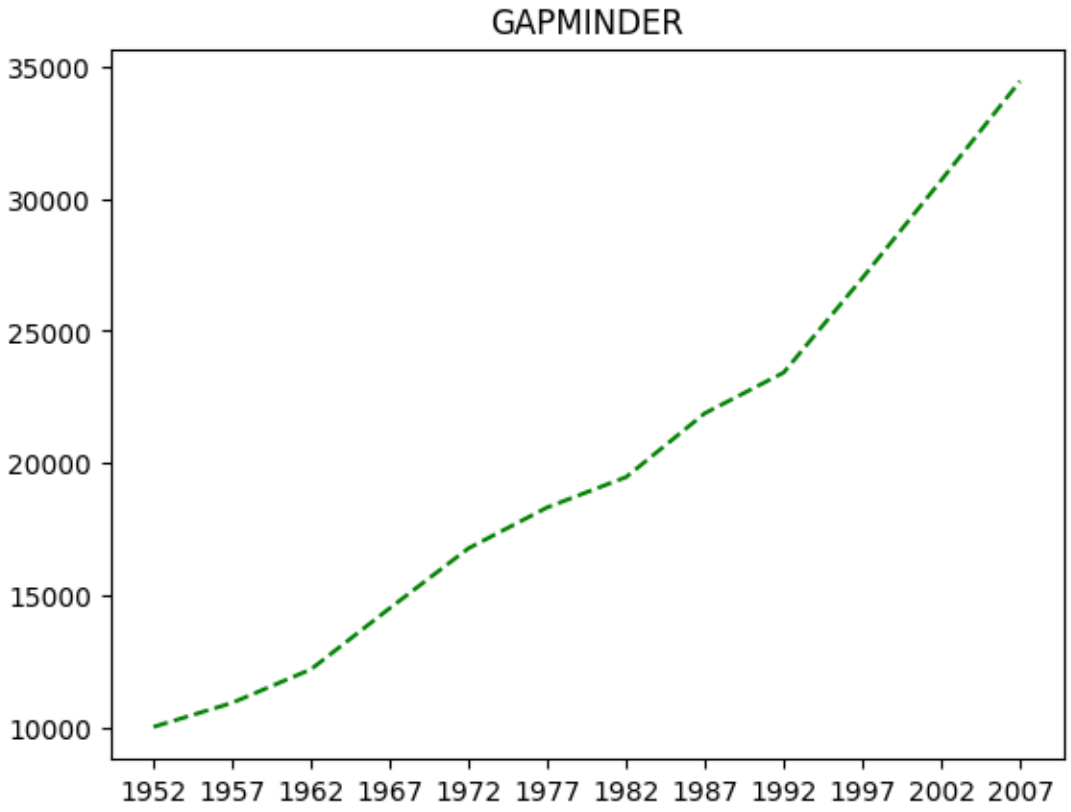
In [302]:
```
# 2 way
gdp_australia=dfoceania.loc['Australia']
plt.plot(years,gdp_australia)
plt.xlabel('Year')
plt.ylabel('GDP per capita, $1,000')
plt.title('GAPMINDER')
```

Out[302]: Text(0.5, 1.0, 'GAPMINDER')



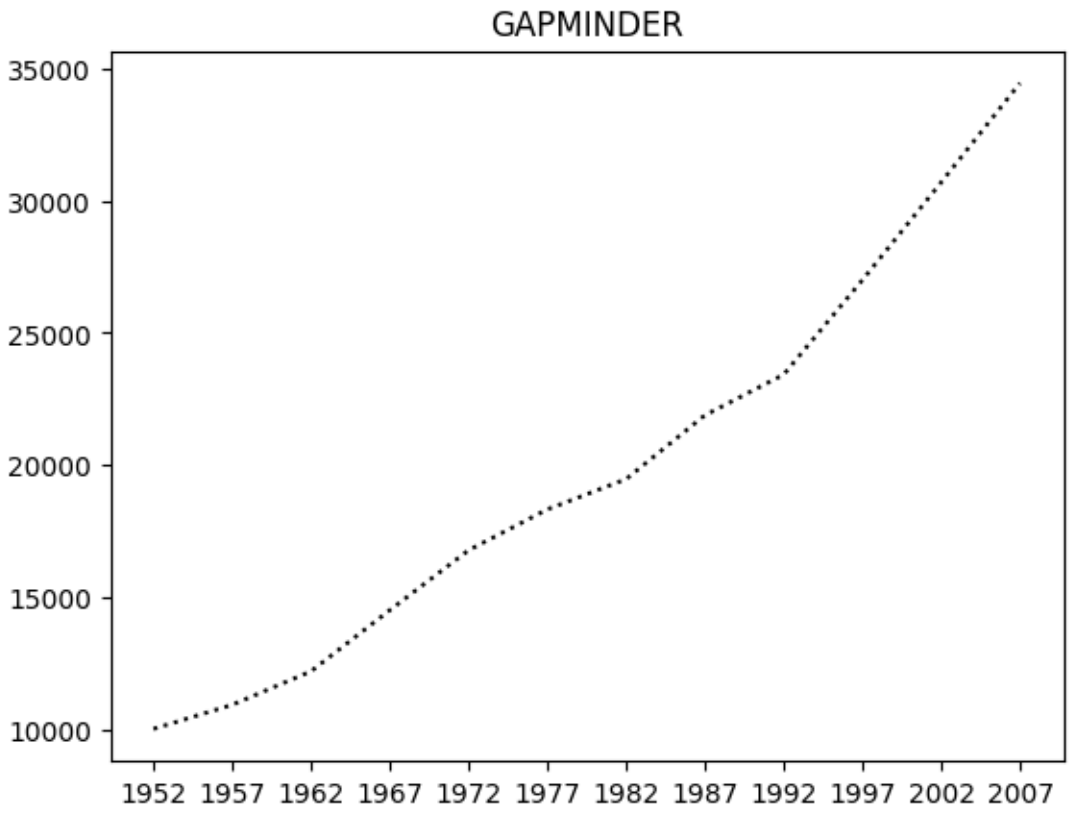
In [303]:
```
plt.plot(years,gdp_australia,'g--') # g == green; -- dashed line style
plt.title('GAPMINDER')
```

Out[303]: Text(0.5, 1.0, 'GAPMINDER')



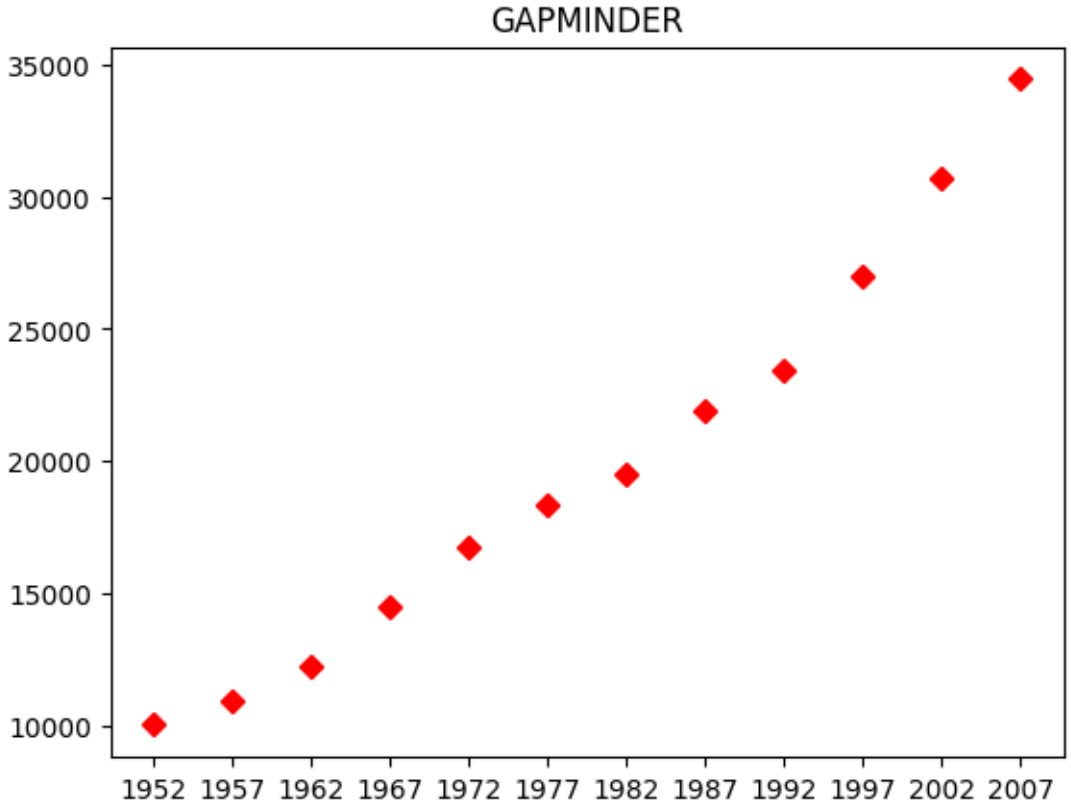
In [304]:
```
plt.plot(years,gdp_australia,'k:') # k == black; : dotted line
plt.title('GAPMINDER')
```

Out[304]: Text(0.5, 1.0, 'GAPMINDER')



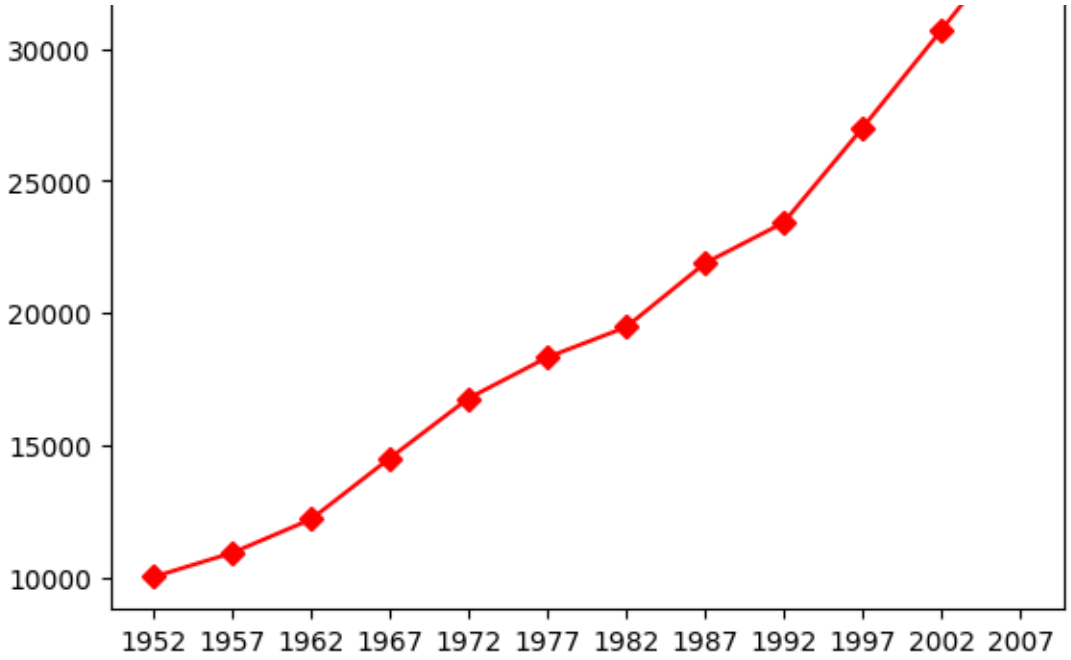
In [305]:
```
plt.plot(years, gdp_australia, 'rD')
plt.title('GAPMINDER')
```

Out[305]: Text(0.5, 1.0, 'GAPMINDER')



In [306]:
```
plt.plot(years, gdp_australia, 'rD-')
plt.title('GAPMINDER')
```

Out[306]: Text(0.5, 1.0, 'GAPMINDER')



In [200]:
```
plt.plot?
```

In [201]:
```
dfoceania
```

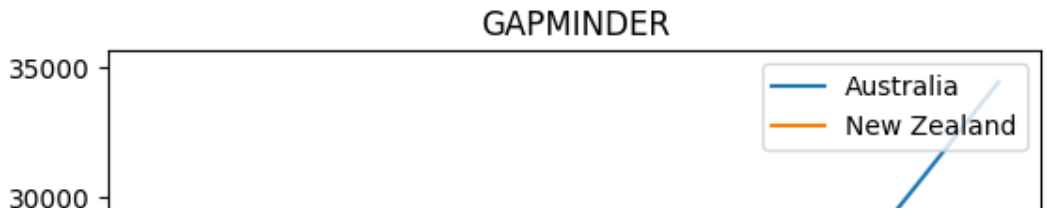
Out[201]:

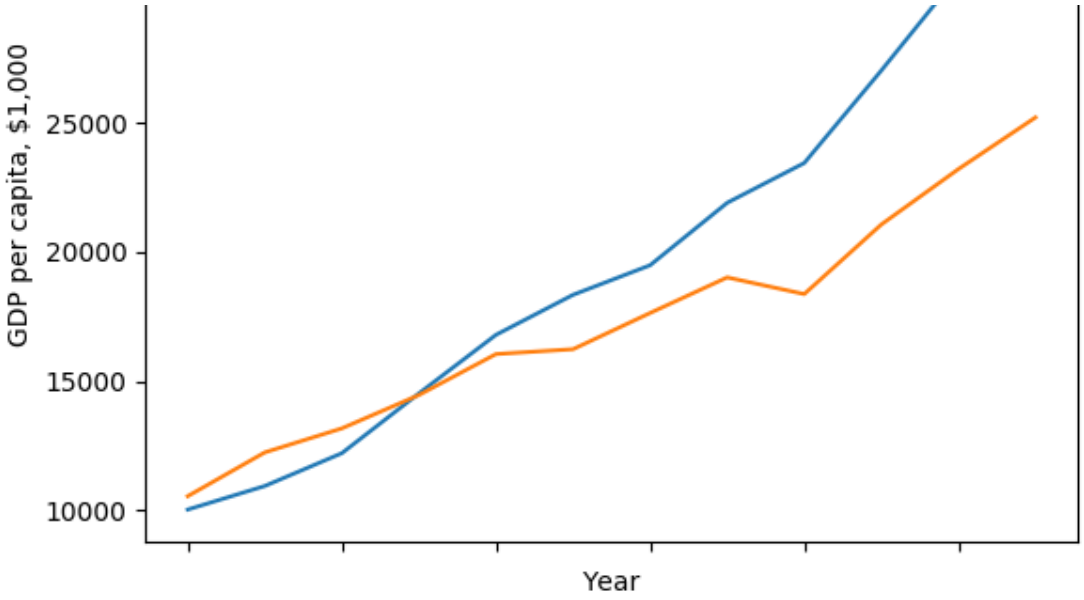
	1952	1957	1962	1967	1972	1
country						
Australia	10039.59564	10949.64959	12217.22686	14526.12465	16788.62948	18334.19
New Zealand	10556.57566	12247.39532	13175.67800	14463.91893	16046.03728	16233.71

\5. Plot more than one data

In [307]:
```
dfoceania.T.plot() # line plot is default
plt.xlabel('Year')
plt.ylabel('GDP per capita, $1,000')
plt.title('GAPMINDER')
#plt.legend() default is best, 1 is upper right, 2 upper left
plt.legend(loc=1)
```

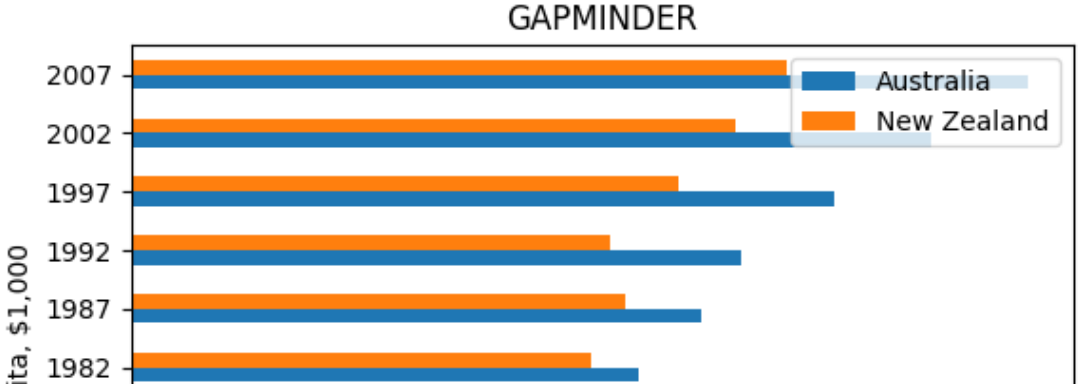
Out[307]: <matplotlib.legend.Legend at 0x125fd88d0>

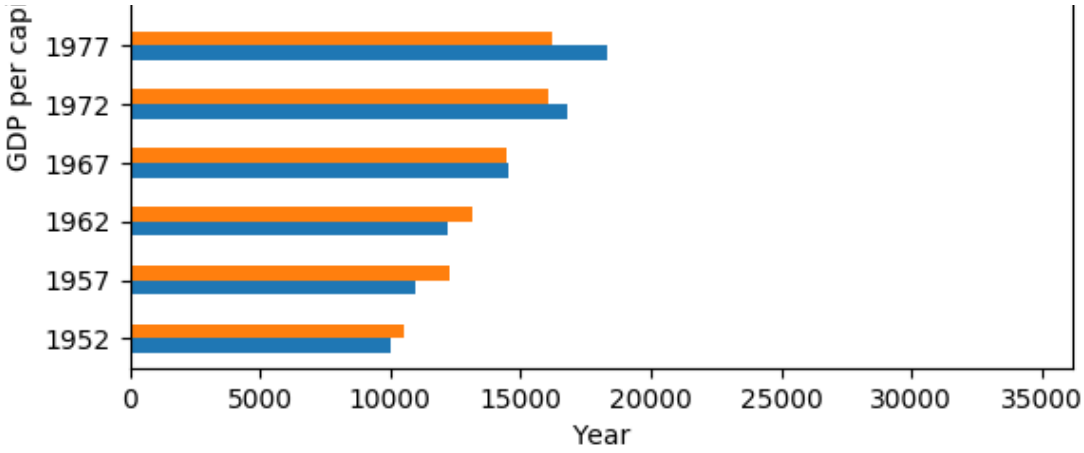




In [299]:
```
dfoceania.T.plot(kind='barh') # bar
plt.xlabel('Year')
plt.ylabel('GDP per capita, $1,000')
plt.title('GAPMINDER')
#plt.legend() default is best, 1 is upper right, 2 upper left
plt.legend(loc=1)
```

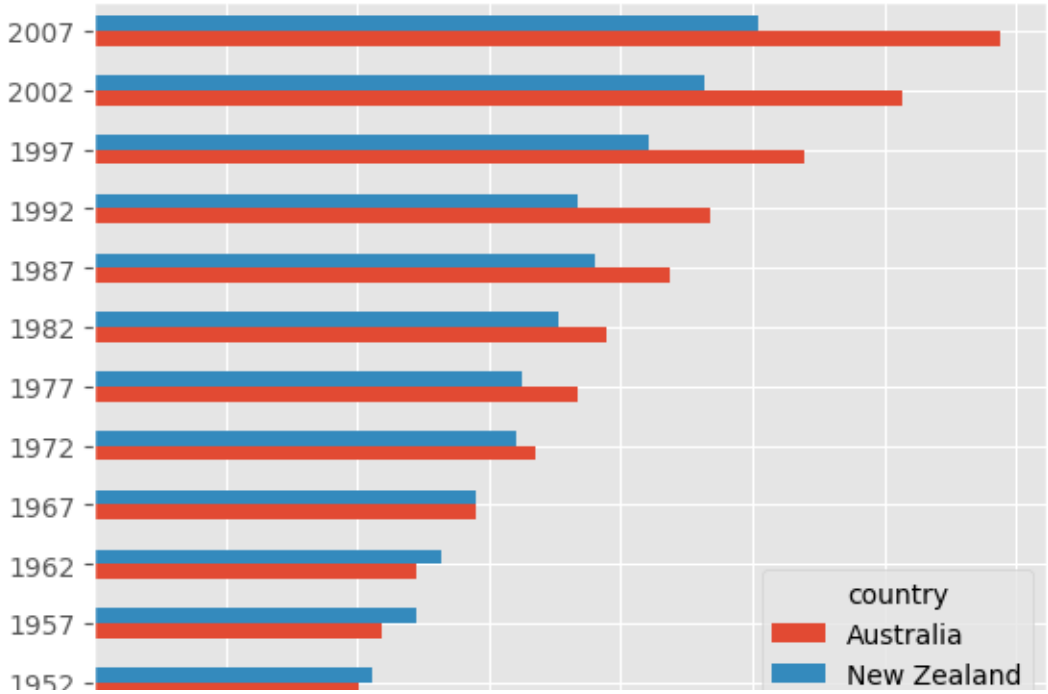
Out[299]: <matplotlib.legend.Legend at 0x1255bf9e8>





In [308]:
```
plt.style.use('ggplot')
dfoceania.T.plot(kind='barh')
```

Out[308]: <matplotlib.axes._subplots.AxesSubplot at 0x126284080>

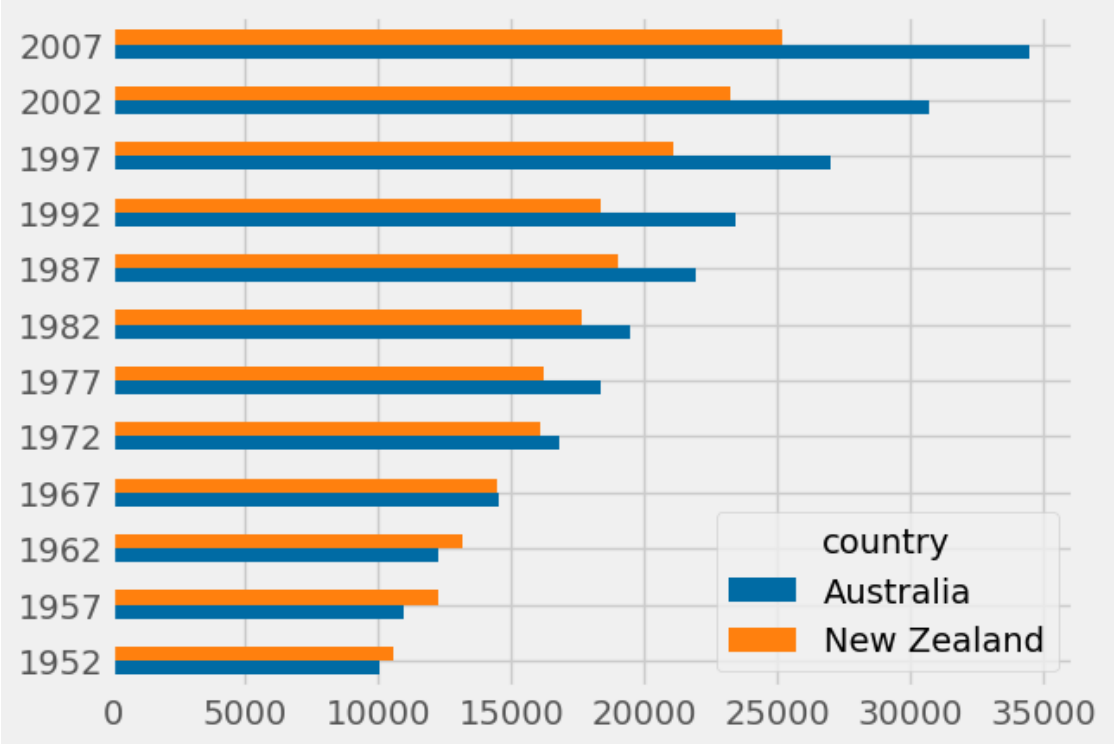


In [298]:
```
print(plt.style.available)
#plt.rcParams.update(plt.rcParamsDefault)
```

```
['seaborn-dark', 'seaborn-darkgrid', 'seaborn-ticks', 'fivethirtyeig
ht', 'seaborn-whitegrid', 'classic', '_classic_test', 'fast', 'seabo
rn-talk', 'seaborn-dark-palette', 'seaborn-bright', 'seaborn-pastel'
, 'grayscale', 'seaborn-notebook', 'ggplot', 'seaborn-colorblind', '
seaborn-muted', 'seaborn', 'Solarize_Light2', 'seaborn-paper', 'bmh'
, 'tableau-colorblind10', 'seaborn-white', 'dark_background', 'seabo
rn-poster', 'seaborn-deep']
```

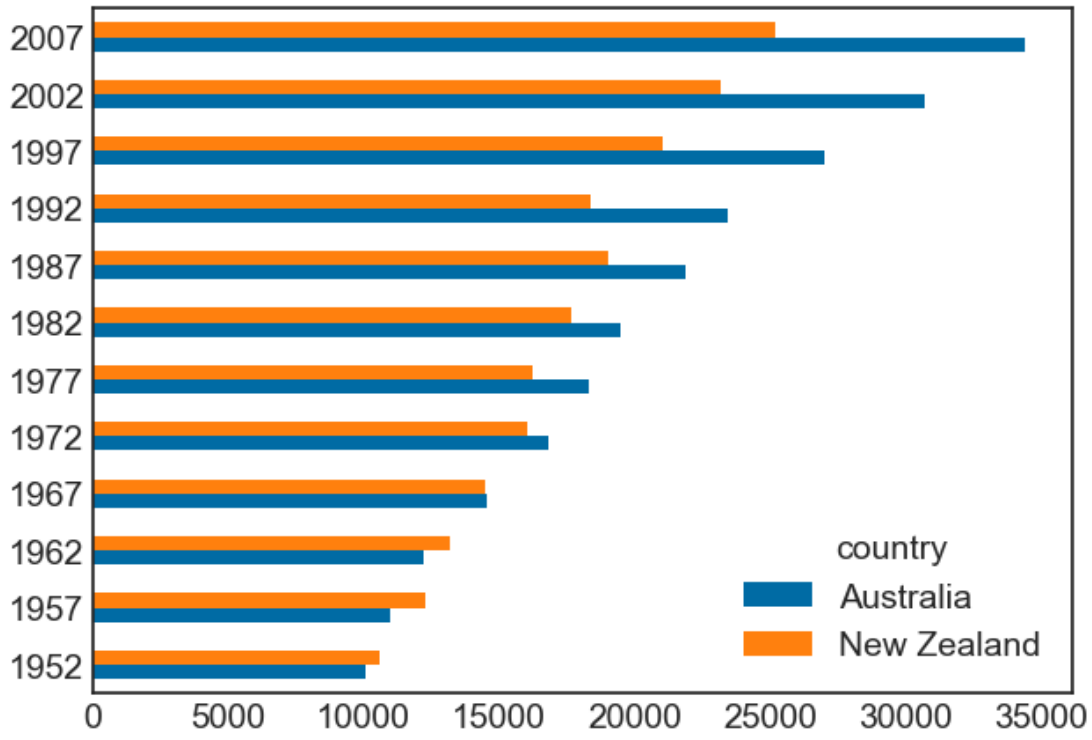
In [310]:
```
plt.style.use('tableau-colorblind10')
dfoceania.T.plot(kind='barh')
```

Out[310]: <matplotlib.axes._subplots.AxesSubplot at 0x12656c908>



In [311]:
```
plt.style.use('seaborn-white')
dfoceania.T.plot(kind='barh')
```

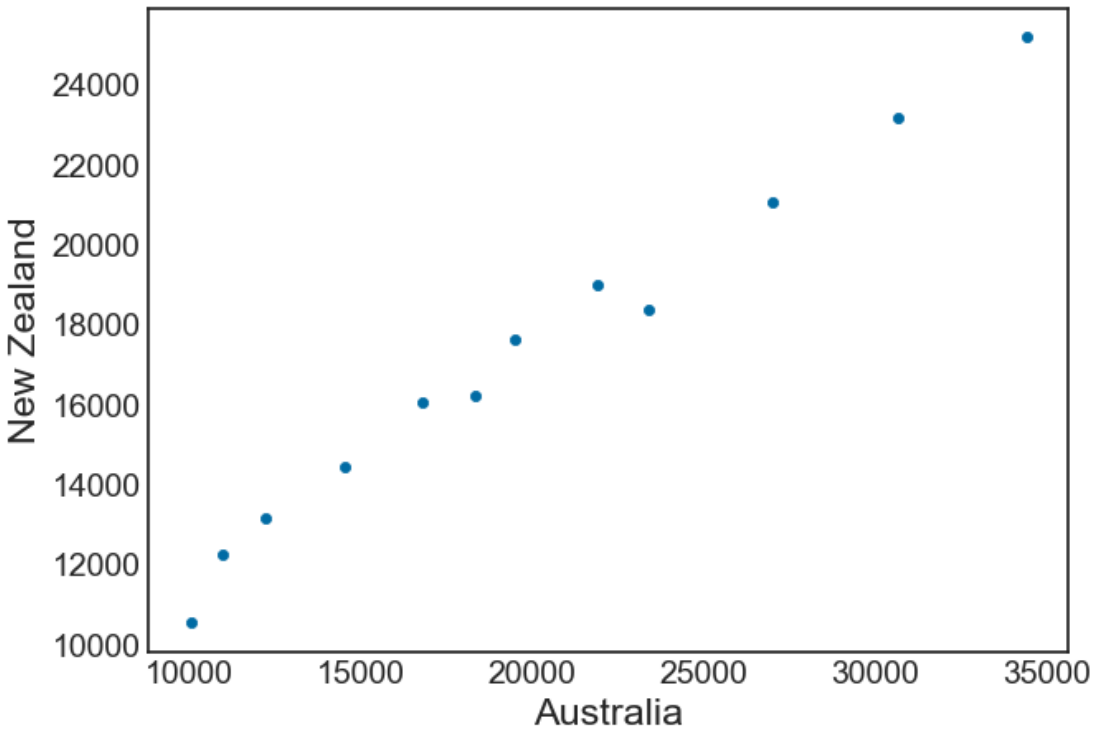
Out[311]: <matplotlib.axes._subplots.AxesSubplot at 0x1264301d0>



\6. Plot country vs country in scatter plot

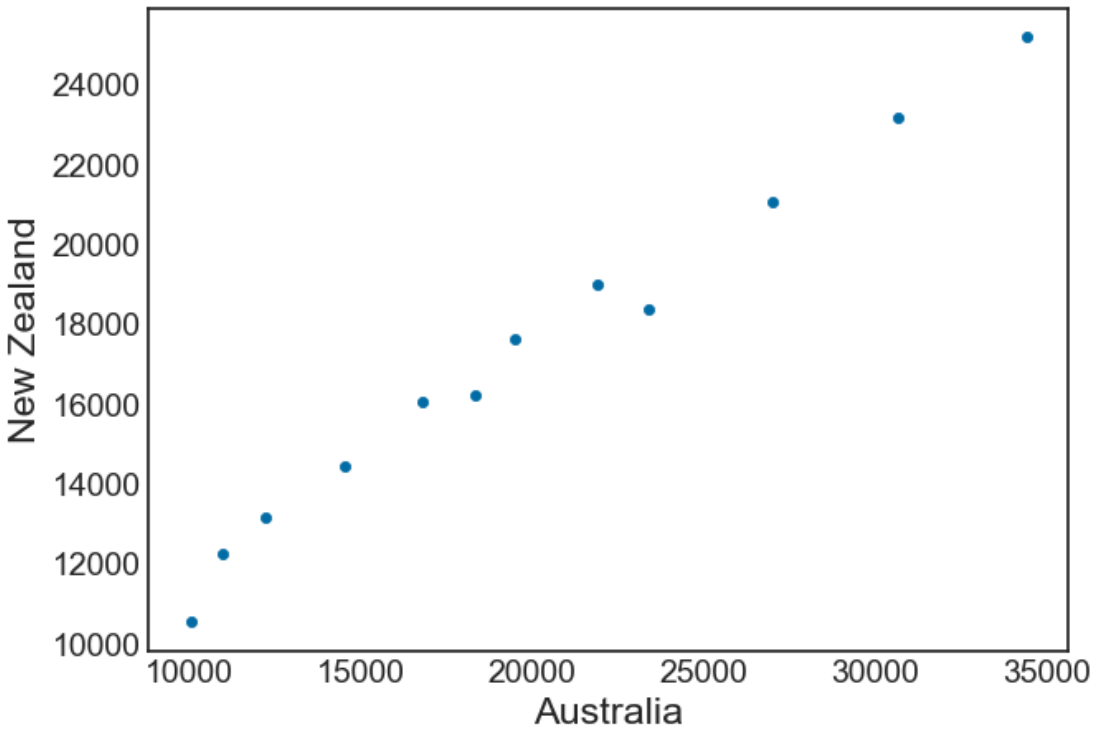
In [312]:
```
dfoceania.T.plot(x='Australia', y='New Zealand',kind='scatter')
```

Out[312]: <matplotlib.axes._subplots.AxesSubplot at 0x12693d8d0>



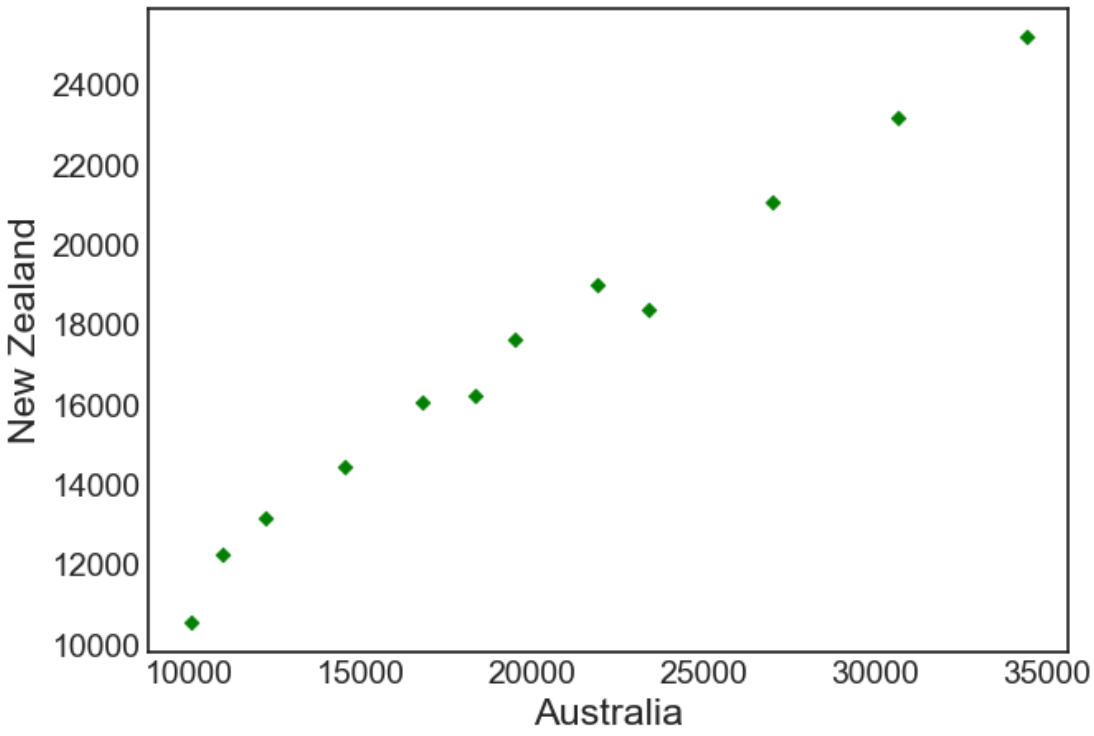
In [313]:
```
dfoceania.T.plot.scatter(x='Australia', y='New Zealand')
```

Out[313]: <matplotlib.axes._subplots.AxesSubplot at 0x126a9aa20>



In [314]:
```
dfoceania.T.plot(x='Australia',y='New Zealand', kind='scatter', c='gre
en', marker='D')
```

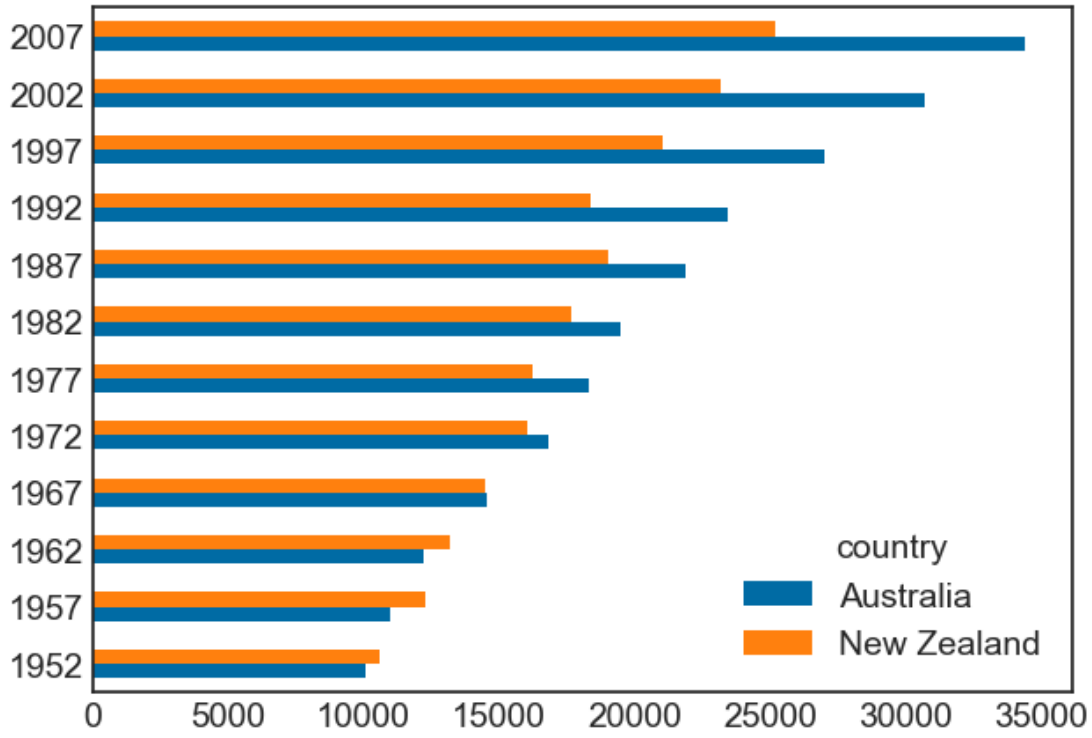
Out[314]: <matplotlib.axes._subplots.AxesSubplot at 0x126c170f0>



In [470]:
```
help(dfoceania)
```

8 - Saving plot to figure

In [322]:
```
plt.style.use('tableau-colorblind10')
dfoceania.T.plot(kind='barh')
plt.savefig('oceania_1.png',dpi=200)
```



In [318]:
```
ls
```

```
americasT_summ_stats.csv          gapminder_gdp_asia.csv
americas_summ_stats.csv           gapminder_gdp_europe.csv
americassubset2T_summ_stats.csv   gapminder_gdp_oceania.csv
gapminder_all.csv                 oceania_1.png
gapminder_gdp_africa.csv          oceania_summ_stats.csv
gapminder_gdp_americas.csv
```

In [319]:
```
%%bash
open .
```

CHEEKY EXERCISES

4. Use `americas`, plot a line graph with the `min`, `mean`, and `max` GDP per capita over time for all countries, use `bmh` style, and save plot to figure

...ball style, and save plot to figure

In [364]:
```
dfamericas.head()
```

Out[364]:

country	continent	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_
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 [365]:
```
# run once!!!
dfamericas = dfamericas.drop('continent', 1)

years2=dfamericas.columns.str.strip('gdpPercap_')  # getting just the years and stripping gdpPercap_
years2
#dfamericas.columns
dfamericas.columns=years.astype(str) # replacing column headers
```

In [366]:
```
dfamericas.head()
```

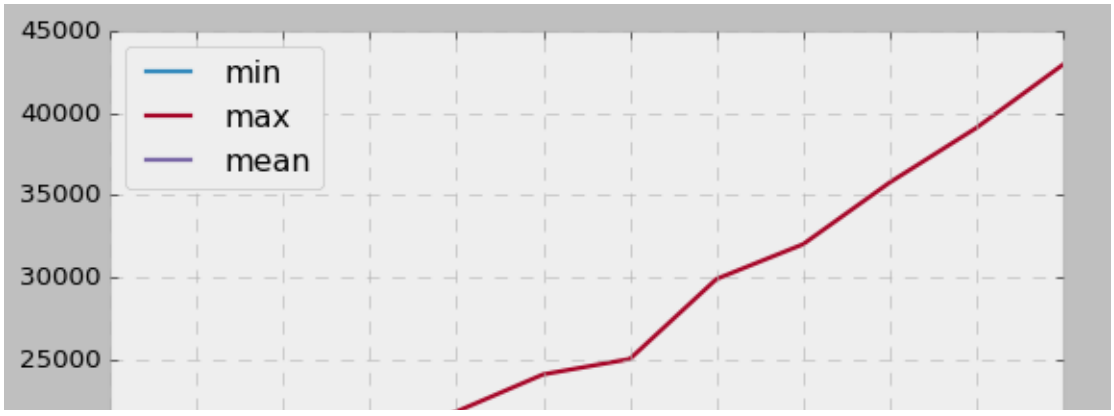
Out[366]:

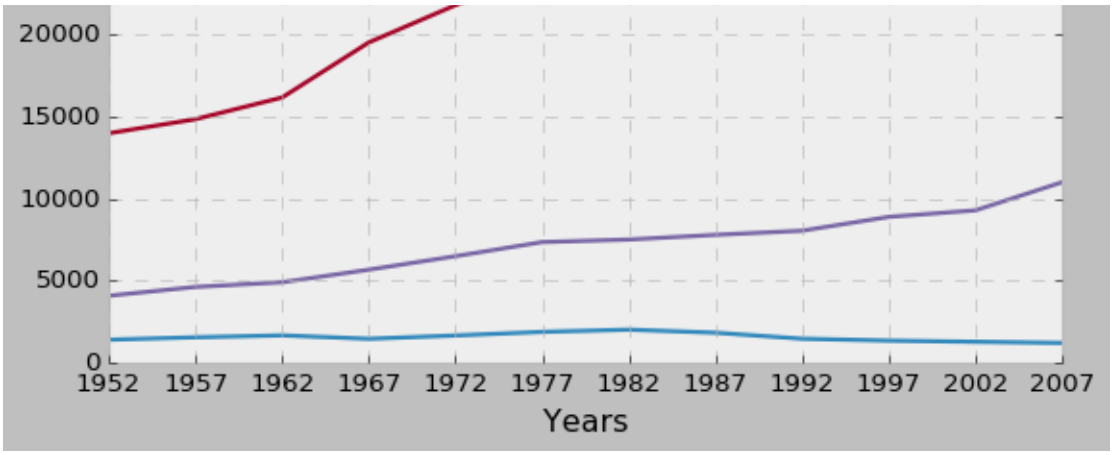
country	1952	1957	1962	1967	1972	
Argentina	5911.315053	6856.856212	7133.166023	8052.953021	9443.038526	10
Bolivia	2677.326347	2127.686326	2180.972546	2586.886053	2980.331339	35
Brazil	2108.944355	2487.365989	3336.585802	3429.864357	4985.711467	66
Canada	11367.161120	12489.950060	13462.485550	16076.588030	18970.570860	22
Chile	3939.978789	4315.622723	4519.094331	5106.654313	5494.024437	47

In [367]:
```
dfamericas.min()
```

Out[367]:
```
1952    1397.717137
1957    1544.402995
1962    1662.137359
1967    1452.057666
1972    1654.456946
1977    1874.298931
1982    2011.159549
1987    1823.015995
1992    1456.309517
1997    1341.726931
2002    1270.364932
2007    1201.637154
dtype: float64
```

In [370]:
```
plt.style.use('bmh')
plt.plot(dfamericas.min(), label='min')
plt.plot(dfamericas.max(),label='max')
plt.plot(dfamericas.mean(),label='mean')
plt.legend(loc='best')
plt.xlabel('Years')
plt.savefig('americas_basic_stats.png')
```





In [371]:
```
ls
```

```
americasT_summ_stats.csv          gapminder_gdp_americas.csv
americas_basic_stats.png          gapminder_gdp_asia.csv
americas_summ_stats.csv           gapminder_gdp_europe.csv
americassubset2T_summ_stats.csv   gapminder_gdp_oceania.csv
gapminder_all.csv                 oceania_1.png
gapminder_gdp_africa.csv          oceania_summ_stats.csv
```

In [369]:
```
# # 2 way without deleting the first column

# plt.style.use('bmh')
# plt.plot(dfamericas.iloc[:,1:].min(), label='min')
# plt.plot(dfamericas.iloc[:,1:].max(),label='max')
# plt.plot(dfamericas.iloc[:,1:].mean(),label='mean')
# plt.legend(loc='best')
# plt.xlabel('Years')
```

5. Import `gapminder_all`

5a - make a histogram of the `2002-2007` GDP of all countries

5b - make a scatter plot of the relationship between the `2007` GDP vs life expectancy

5c - make 3 box plots to explore the GDP trends for the years `1997, 2002 and 2007`

In [374]:
```
dfall = pd.read_csv('gapminder_all.csv', index_col='country')
```

In [412]:
```
dfall.head()
```

Out[412]:

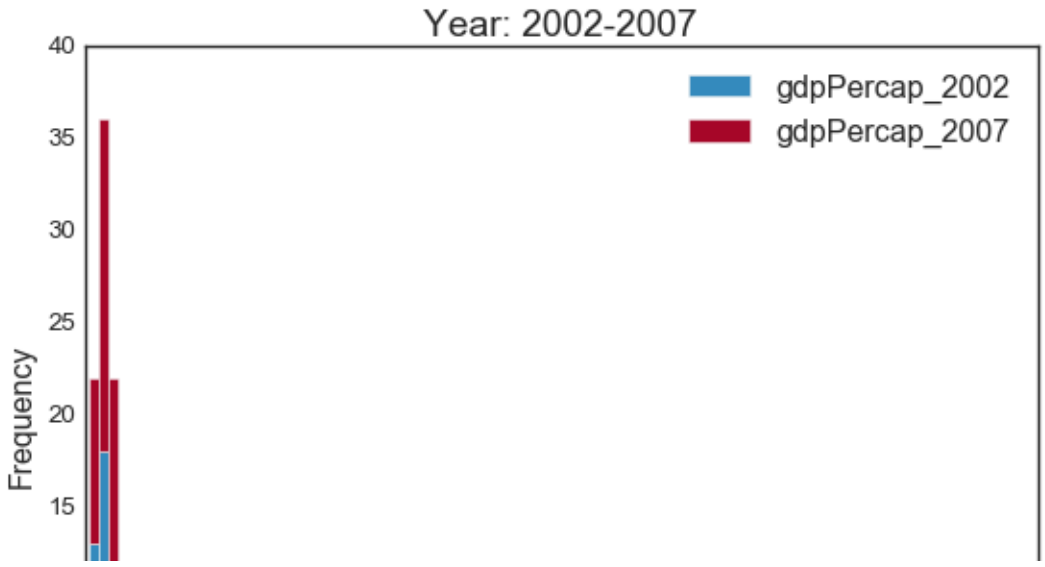
	continent	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap
country					
Algeria	Africa	2449.008185	3013.976023	2550.816880	3246.99177
Angola	Africa	3520.610273	3827.940465	4269.276742	5522.77637
Benin	Africa	1062.752200	959.601080	949.499064	1035.83141
Botswana	Africa	851.241141	918.232535	983.653976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

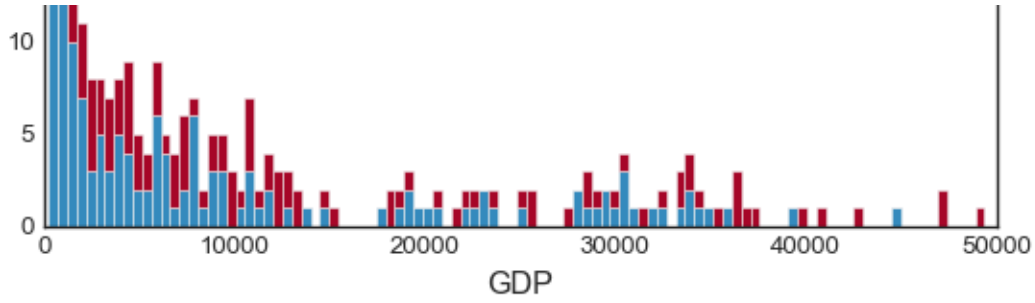
5 rows × 37 columns

A

In [439]:
```
plt.style.use('seaborn-white')
subset_all=dfall.loc[:,'gdpPercap_2002':'gdpPercap_2007']
subset_all.plot(kind='hist', bins=100,stacked=True)
plt.xlabel('GDP')
plt.ylabel('Frequency')
plt.title('Year: 2002-2007')
plt.legend()
```

Out[439]: <matplotlib.legend.Legend at 0x147874048>

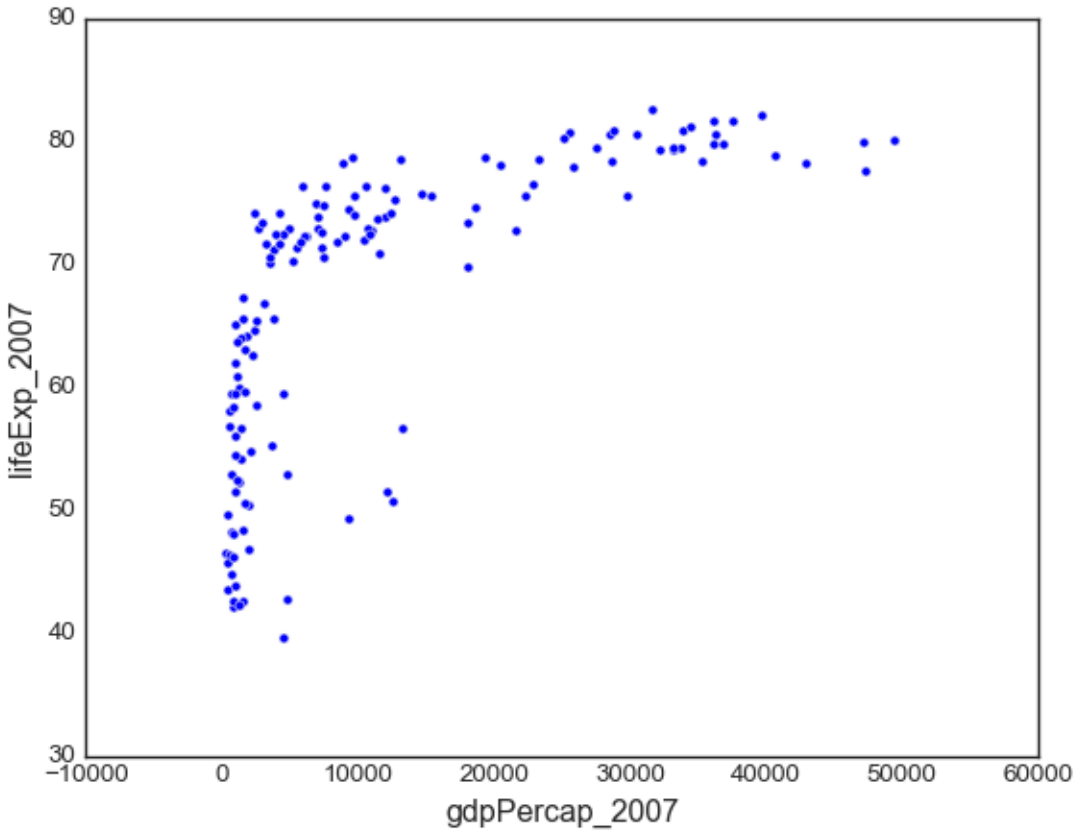




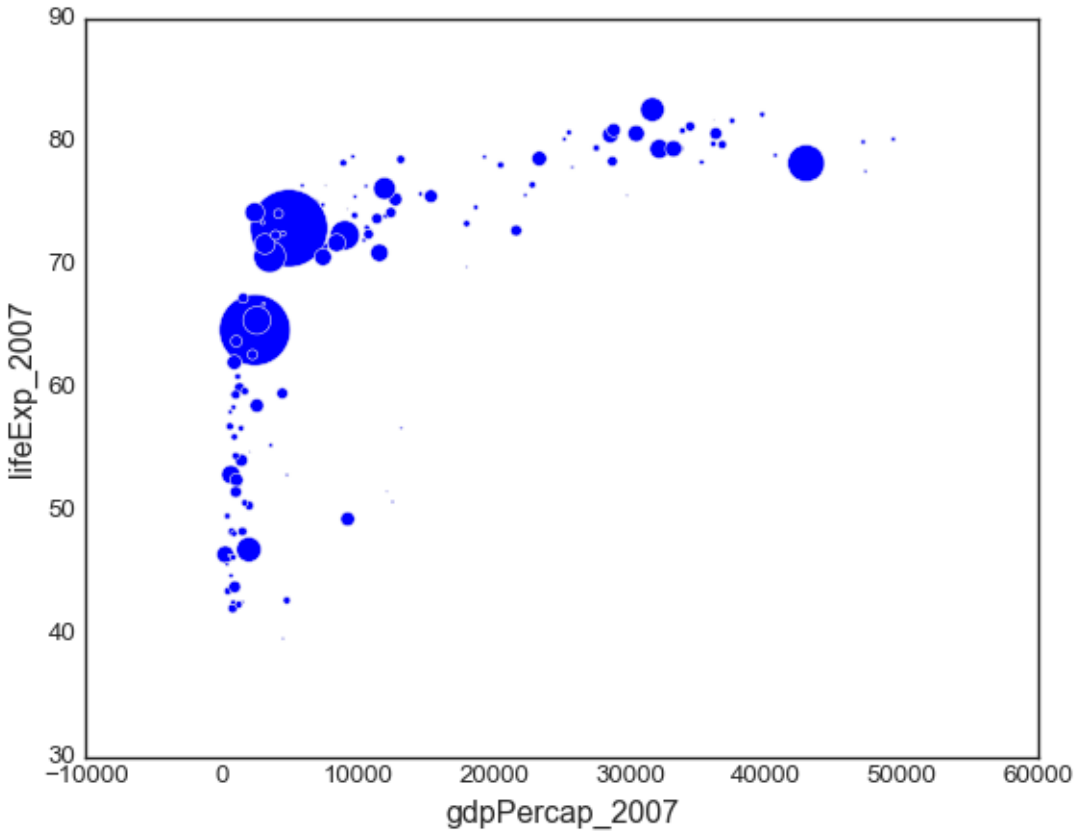
B

In [467]:
```
dfall.plot(kind='scatter', x='gdpPercap_2007', y='lifeExp_2007')
```

Out[467]: <matplotlib.axes._subplots.AxesSubplot at 0x149b79e80>



In [468]:
```
dfall.plot(kind='scatter', x='gdpPercap_2007', y='lifeExp_2007', s=dfa
ll['pop_2007']/1e6) # add scalar of the dots
plt.savefig('all_gdp_vs_lifeExp.png')
```



In [418]:
```
dfall.loc[:,'gdpPercap_1997':'gdpPercap_2007']
```

Out[418]:

	gdpPercap_1997	gdpPercap_2002	gdpPercap_2007
country			
Algeria	4797.295051	5288.040382	6223.367465
Angola	2277.140884	2773.287312	4797.231267
Benin	1232.975292	1372.877931	1441.284873

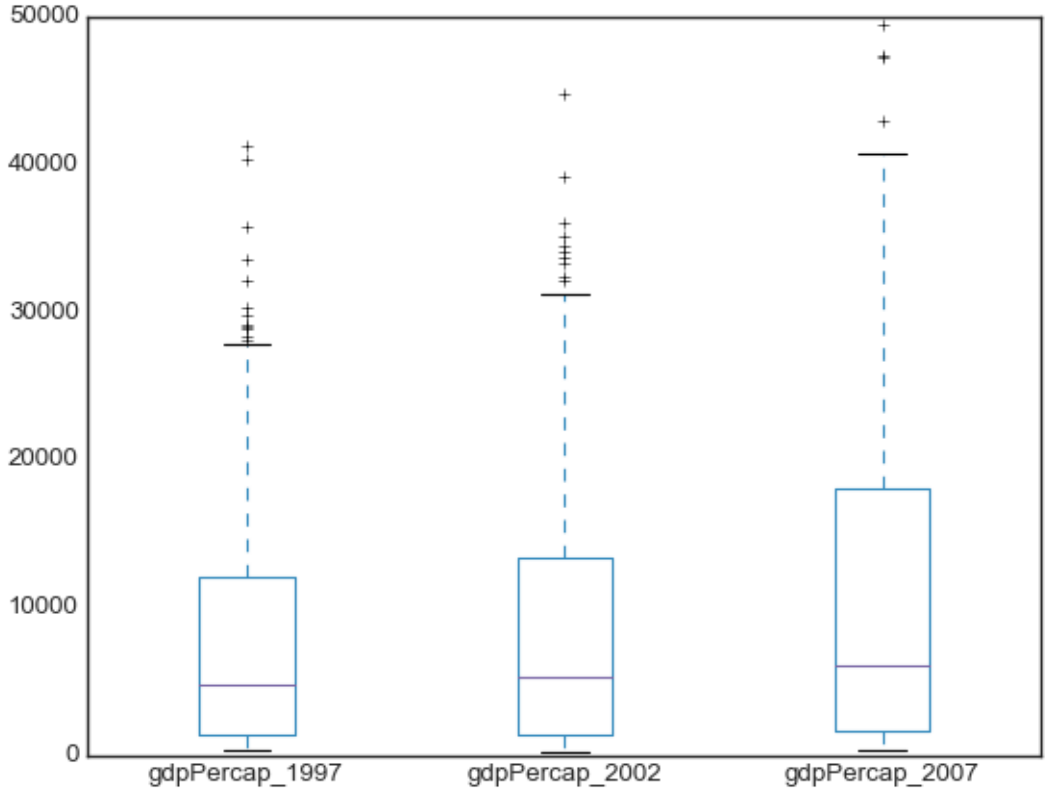
Botswana	8647.142313	11003.605080	12569.851770
Burkina Faso	946.294962	1037.645221	1217.032994
Burundi	463.115148	446.403513	430.070692
Cameroon	1694.337469	1934.011449	2042.095240
Central African Republic	740.506332	738.690607	706.016537
Chad	1004.961353	1156.181860	1704.063724
Comoros	1173.618235	1075.811558	986.147879
Congo Dem. Rep.	312.188423	241.165877	277.551859
Congo Rep.	3484.164376	3484.061970	3632.557798
Cote d'Ivoire	1786.265407	1648.800823	1544.750112
Djibouti	1895.016984	1908.260867	2082.481567
Egypt	4173.181797	4754.604414	5581.180998
Equatorial Guinea	2814.480755	7703.495900	12154.089750
Eritrea	913.470790	765.350001	641.369524
Ethiopia	515.889401	530.053532	690.805576
Gabon	14722.841880	12521.713920	13206.484520
Gambia	653.730170	660.585600	752.749726
Ghana	1005.245812	1111.984578	1327.608910
Guinea	869.449767	945.583584	942.654211
Guinea-Bissau	796.664468	575.704718	579.231743
Kenya	1360.485021	1287.514732	1463.249282
Lesotho	1186.147994	1275.184575	1569.331442
Liberia	609.173951	531.482368	414.507341
Libya	9467.446056	9534.677467	12057.499280
Madagascar	986.295896	894.637082	1044.770126
Malawi	692.275810	665.423119	759.349910
Mali	790.257985	951.409752	1042.581557
...	...	...	...
Belgium	27561.196630	30485.883750	33692.605080
Bosnia and Herzegovina	4766.355904	6018.975239	7446.298803
Bulgaria	5970.388760	7696.777725	10680.792820
Croatia	9875.604515	11628.388950	14619.222720
Czech Republic	16048.514240	17596.210220	22833.308510

Denmark	29804.345670	32166.500060	35278.418740
Finland	23723.950200	28204.590570	33207.084400
France	25889.784870	28926.032340	30470.016700
Germany	27788.884160	30035.801980	32170.374420
Greece	18747.698140	22514.254800	27538.411880
Hungary	11712.776800	14843.935560	18008.944440
Iceland	28061.099660	31163.201960	36180.789190
Ireland	24521.947130	34077.049390	40675.996350
Italy	24675.024460	27968.098170	28569.719700
Montenegro	6465.613349	6557.194282	9253.896111
Netherlands	30246.130630	33724.757780	36797.933320
Norway	41283.164330	44683.975250	49357.190170
Poland	10159.583680	12002.239080	15389.924680
Portugal	17641.031560	19970.907870	20509.647770
Romania	7346.547557	7885.360081	10808.475610
Serbia	7914.320304	7236.075251	9786.534714
Slovak Republic	12126.230650	13638.778370	18678.314350
Slovenia	17161.107350	20660.019360	25768.257590
Spain	20445.298960	24835.471660	28821.063700
Sweden	25266.594990	29341.630930	33859.748350
Switzerland	32135.323010	34480.957710	37506.419070
Turkey	6601.429915	6508.085718	8458.276384
United Kingdom	26074.531360	29478.999190	33203.261280
Australia	26997.936570	30687.754730	34435.367440
New Zealand	21050.413770	23189.801350	25185.009110

142 rows × 3 columns

In [469]:
```
dfall.loc[:,'gdpPercap_1997':'gdpPercap_2007'].plot(kind='box')
```

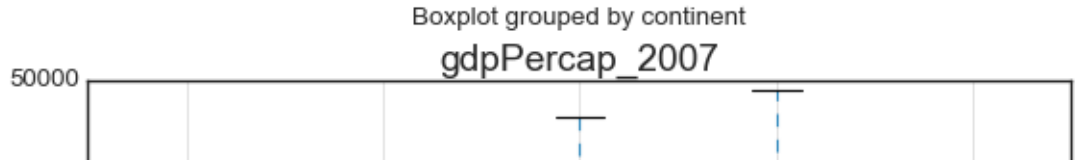
Out[469]: <matplotlib.axes._subplots.AxesSubplot at 0x14b0f2d68>

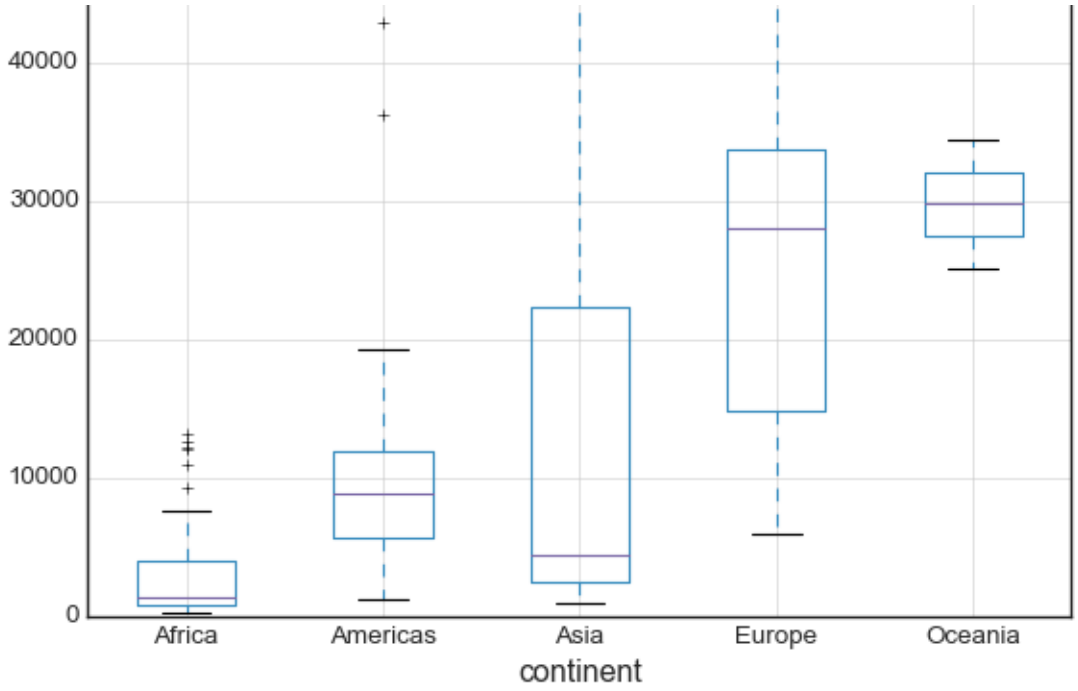


C

In [444]:
```
dfall.boxplot('gdpPercap_2007',by='continent')
```

Out[444]: <matplotlib.axes._subplots.AxesSubplot at 0x149e69128>





Quick recap: For loop

In [451]:
```
fruits
```

Out[451]:
```
['orange', 'strawberry', 'banana', 'mango']
```

In [453]:
```
for k in fruits:
    print(k,'has',len(k),'letters.')
```

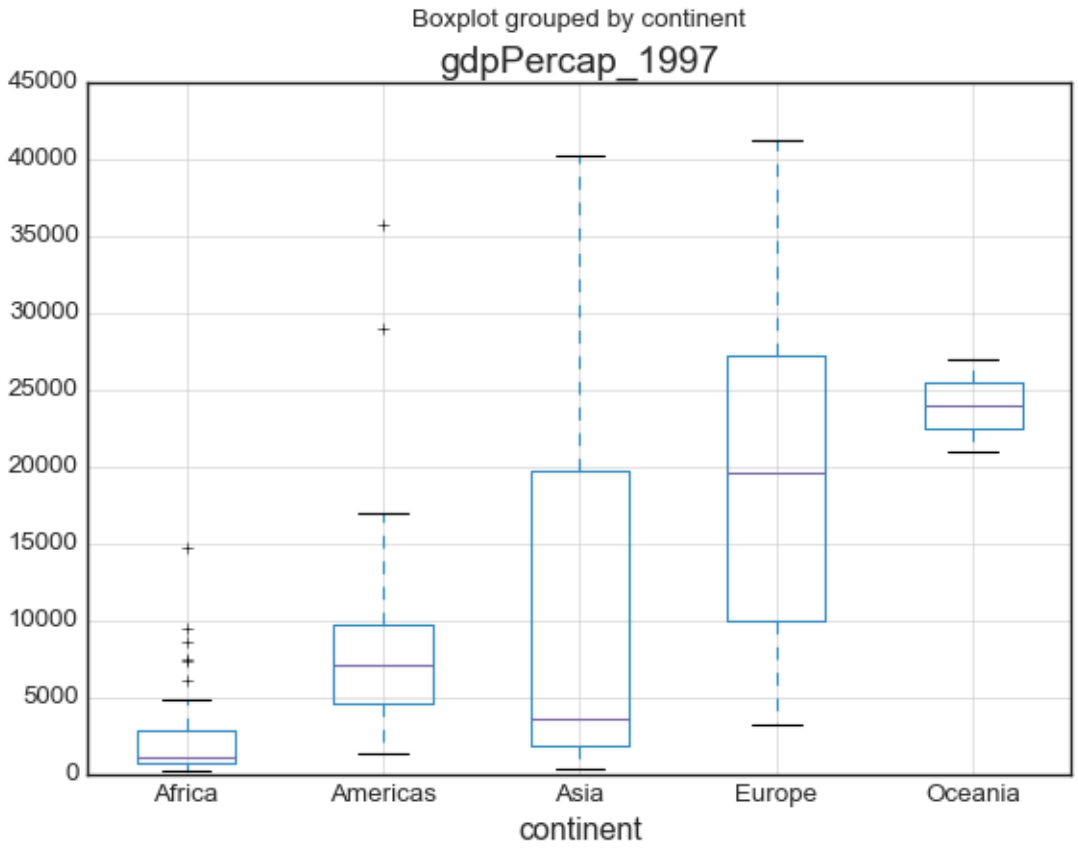
```
orange has 6 letters.
strawberry has 10 letters.
banana has 6 letters.
mango has 5 letters.
```

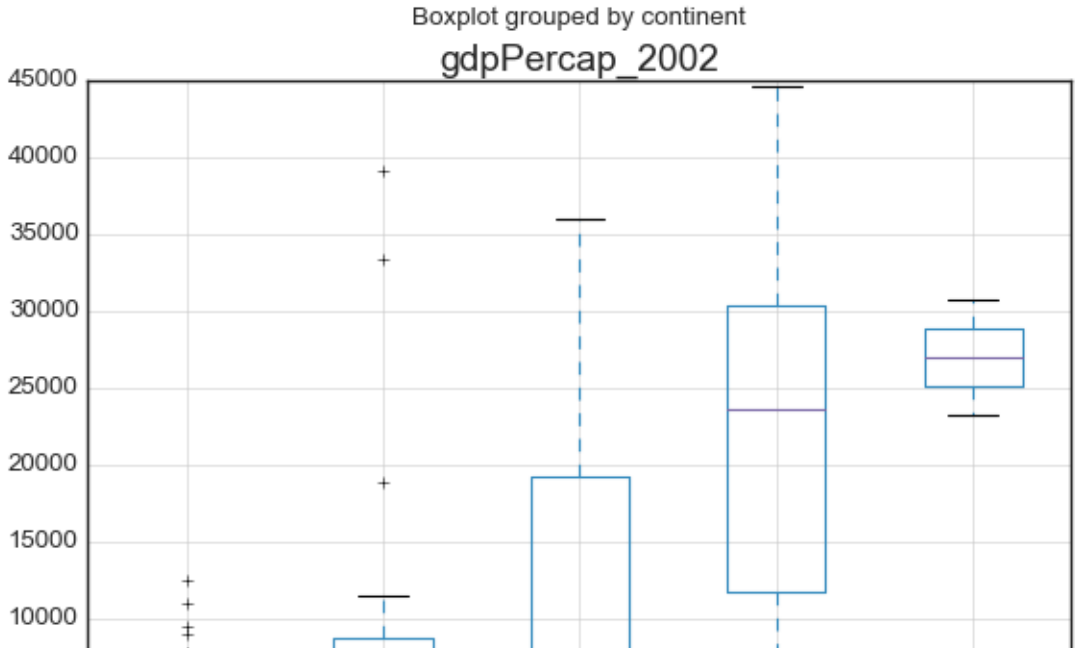
In [454]:
```
for i in range(1,10):
    print(i,'\t',i**2)
```

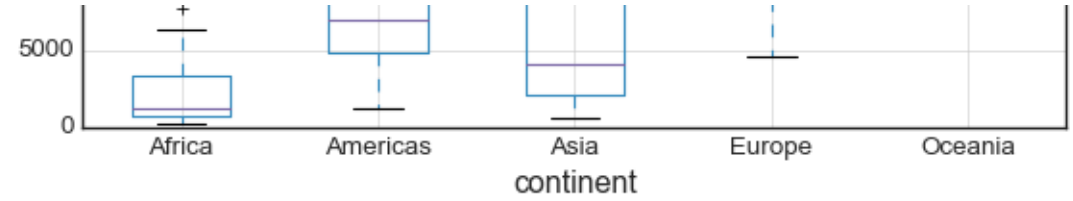
```
1        1
2        4
3        9
4        16
5        25
6        36
7        49
8        64
9        81
```

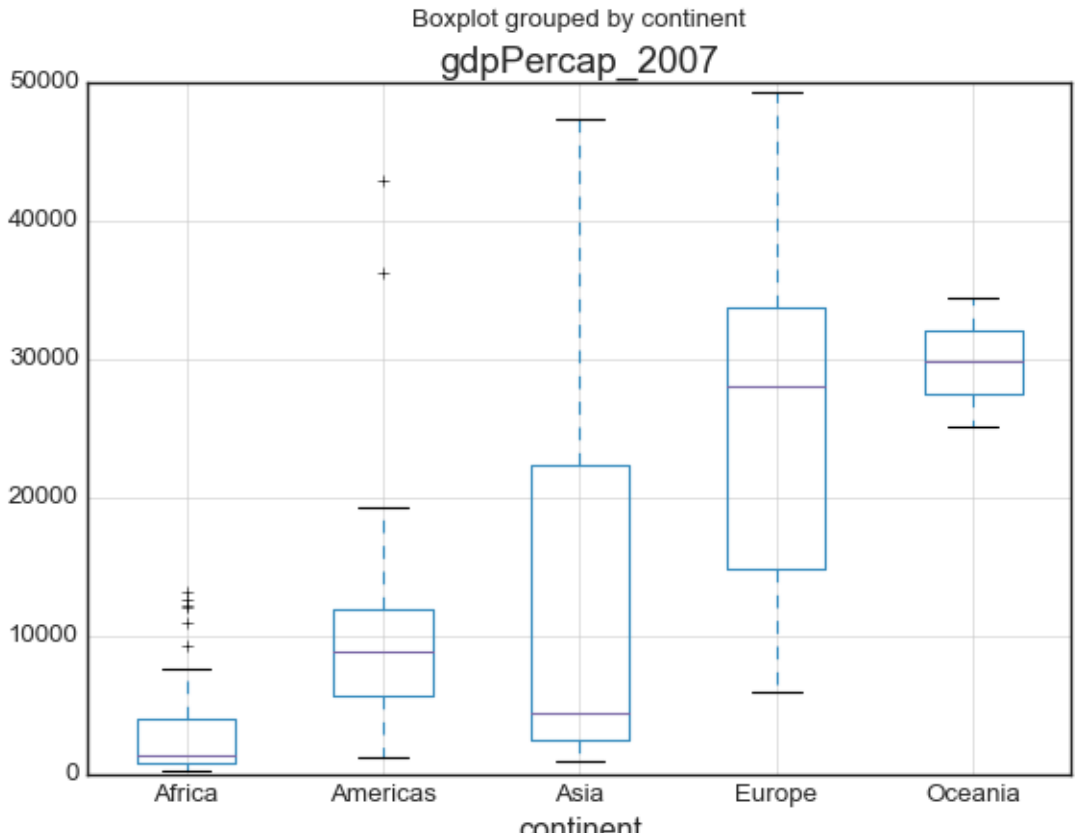
In [450]:
```
year4=['gdpPercap_1997','gdpPercap_2002','gdpPercap_2007']
```

```
for yr in year4:
    dfall.boxplot(yr, by='continent')
```






5000
0
Africa
Americas
Asia
Europe
Oceania
continent


Boxplot grouped by continent
gdpPercap_2007
50000
40000
30000
20000
10000
0
Africa
Americas
Asia
Europe
Oceania
continent

9 - Looping over data sets -- Automating tasks

In [455]:
```
import glob
```

Match files with a pattern == globbing

In [456]:
```
print(glob.glob('*.csv')) # we have extra files with summs... remove t
hose
```

```
['gapminder_gdp_americas.csv', 'gapminder_gdp_europe.csv', 'gapminde
r_all.csv', 'gapminder_gdp_oceania.csv', 'americas_summ_stats.csv',
'gapminder_gdp_africa.csv', 'americasT_summ_stats.csv', 'oceania_sum
m_stats.csv', 'americassubset2T_summ_stats.csv', 'gapminder_gdp_asia
.csv']
```

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

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

In [458]:
```
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 [459]:
```
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
Angola           2430.208311     2627.845685    ...      4826015.0
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  \
country
Algeria       12760499.0  14760787.0  17152804.0  20033753.0  232549
56.0
Angola         5247469.0   5894858.0   6162675.0   7016384.0   78742
30.0
Benin          2427334.0   2761407.0   3168267.0   3641603.0   42437
88.0
Botswana        553541.0    619351.0    781472.0    970347.0   11511
84.0
Burkina Faso   5127935.0   5433886.0   5889574.0   6634596.0   75865
51.0

               pop_1992    pop_1997  pop_2002  pop_2007
country
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  gdpPer
cap_1967  \
country
Australia       10039.59564     10949.64959     12217.22686     1452
6.12465
New Zealand     10556.57566     12247.39532     13175.67800     1446
```

```
New Zealand     10556.57566     12247.39532     13175.67800     1446
3.91893


            gdpPercap_1972  gdpPercap_1977  gdpPercap_1982  gdpPerc
ap_1987  \
country
Australia       16788.62948     18334.19751     19477.00928     2188
8.88903
New Zealand     16046.03728     16233.71770     17632.41040     1900
7.19129

            gdpPercap_1992  gdpPercap_1997  gdpPercap_2002  gdpPerc
ap_2007
country
Australia       23424.76683     26997.93657     30687.75473     3443
5.36744
New Zealand     18363.32494     21050.41377     23189.80135     2518
5.00911
```

gapminder_gdp_africa.csv

```
              gdpPercap_1952  gdpPercap_1957  gdpPercap_1962  gdpPe
rcap_1967  \
country
Algeria          2449.008185     3013.976023     2550.816880     324
6.991771
Angola           3520.610273     3827.940465     4269.276742     552
2.776375
Benin            1062.752200      959.601080      949.499064     103
5.831411
Botswana          851.241141      918.232535      983.653976     121
4.709294
Burkina Faso      543.255241      617.183465      722.512021      79
4.826560

             gdpPercap_1972  gdpPercap_1977  gdpPercap_1982  gdpPer
cap_1987  \
country
Algeria          4182.663766     4910.416756     5745.160213     568
1.358539
Angola           5473.288005     3008.647355     2756.953672     243
0.208311
Benin            1085.796879     1029.161251     1277.897616     122
5.856010
Botswana         2263.611114     3214.857818     4551.142150     620
5.883850
Burkina Faso      854.735976      743.387037      807.198586      91
2.063142

             gdpPercap_1992  gdpPercap_1997  gdpPercap_2002  gdpPer
cap_2007

country
Algeria          5023.216647     4797.295051     5288.040382     622
3.367465
```

```
Angola           2627.845685     2277.140884     2773.287312      479
7.231267
Benin            1191.207681     1232.975292     1372.877931      144
1.284873
Botswana         7954.111645     8647.142313    11003.605080     1256
9.851770
Burkina Faso      931.752773      946.294962     1037.645221      121
7.032994
```

gapminder_gdp_asia.csv

```
             gdpPercap_1952  gdpPercap_1957  gdpPercap_1962  gdpPer
cap_1967  \
country
Afghanistan      779.445314      820.853030      853.100710      836
.197138
Bahrain         9867.084765    11635.799450    12753.275140    14804
.672700
Bangladesh       684.244172      661.637458      686.341554      721
.186086
Cambodia         368.469286      434.038336      496.913648      523
.432314
China            400.448611      575.987001      487.674018      612
.705693

             gdpPercap_1972  gdpPercap_1977  gdpPercap_1982  gdpPerc
ap_1987  \
country
Afghanistan      739.981106      786.113360      978.011439      852
.395945
Bahrain        18268.658390    19340.101960    19211.147310    18524
.024060
Bangladesh       630.233627      659.877232      676.981866      751
.979403
Cambodia         421.624026      524.972183      624.475478      683
.895573
China            676.900092      741.237470      962.421380     1378
.904018

             gdpPercap_1992  gdpPercap_1997  gdpPercap_2002  gdpPerc
ap_2007
country
Afghanistan      649.341395      635.341351      726.734055      974
.580338
Bahrain        19035.579170    20292.016790    23403.559270    29796
.048340

Bangladesh       837.810164      972.770035     1136.390430     1391
.253792
Cambodia         682.303175      734.285170      896.226015     1713
.778686
China           1655.784158     2289.234136     3119.280896     4959
.114854
```

In [460]:
```
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(),subs
et.mean(),subset.max()))
```

```
Filename        Minimum Mean    Maxima
gapminder_gdp_america   1397.7171369999999      4079.0625522000005
13990.482080000002
gapminder_gdp_europe    973.5331947999999       5661.0574347599995
14734.23275
gapminder_all   298.8462121     3725.2760457992963      108382.3529
gapminder_gdp_oceania   10039.595640000001      10298.08565
10556.575659999999
gapminder_gdp_africa    298.8462121     1252.5724658211539
4725.295531000001
gapminder_gdp_asia      331.0   5195.484004030303       108382.3529
```

In [461]:
```
with open('loop_minTOmax.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'),subs
et.min(),subset.mean(),subset.max()))

output.close()
```

In [466]:
```
ls
```

```
all_gdp_vs_lifeExp.png          gapminder_gdp_americas.csv
americasT_summ_stats.csv        gapminder_gdp_asia.csv
americas_basic_stats.png        gapminder_gdp_europe.csv
americas_summ_stats.csv         gapminder_gdp_oceania.csv
americassubset2T_summ_stats.csv loop_minTOmax.txt
gapminder_all.csv               oceania_1.png
gapminder_gdp_africa.csv        oceania_summ_stats.csv
```

CHALLENGE TO TAKE HOME: AUTOMATE PLOTTING DATA

10 - Functions

In [463]:
```
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 [464]:
```
temp_converter('C',25)
```

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

In [465]:
```
temp_converter('F',101)
```

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