輕鬆學習 Python | 從基礎到應用，成為初級 Python 資料分析師

matplotlib.pyplot：探索性分析與資料視覺化的 Python 模組

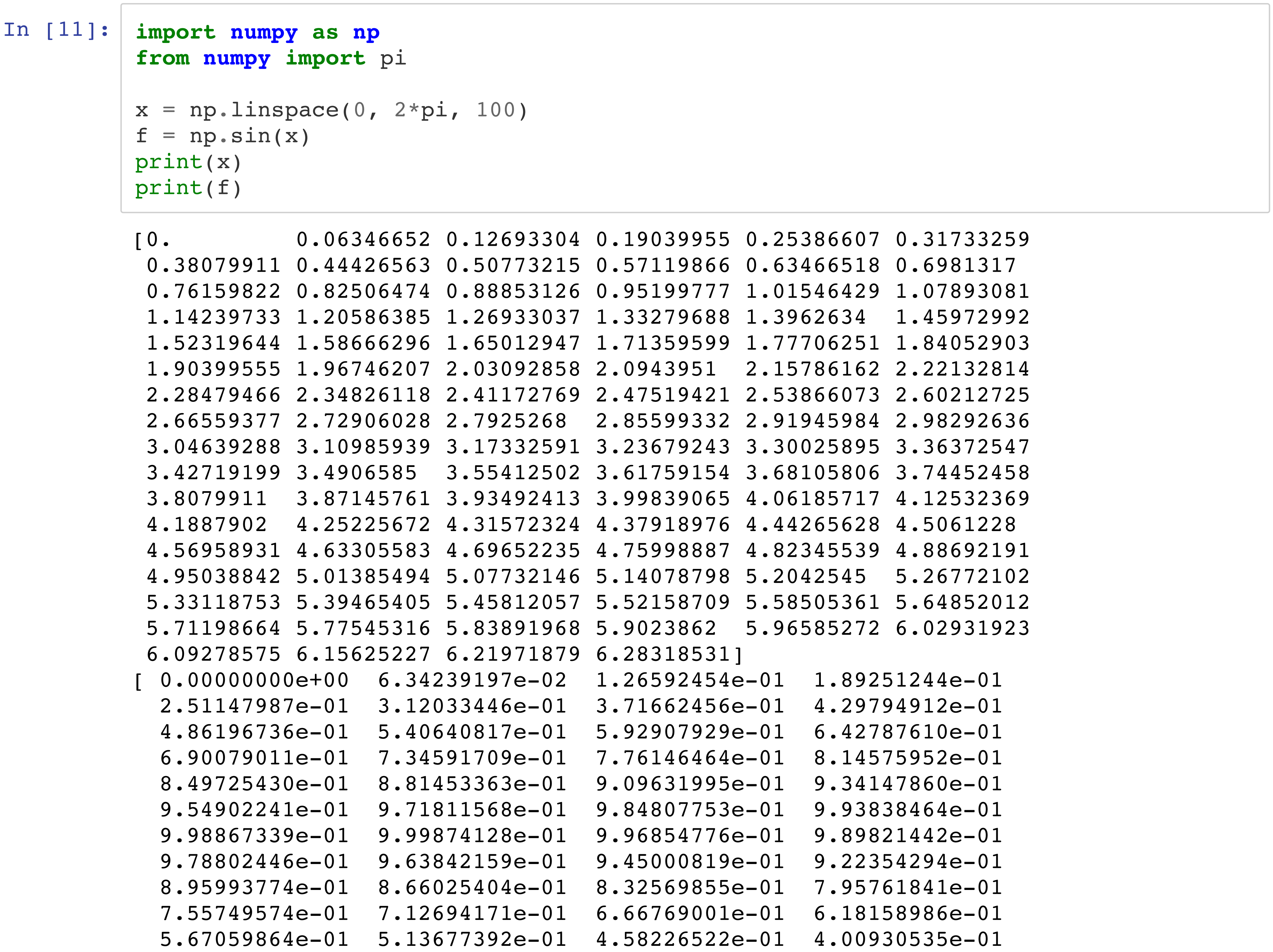
郭耀仁

課程綱要

- matplotlib.pyplot 解決了什麼問題
- 認識範例資料集： gapminder
- matplotlib.pyplot 基礎
- matplotlib.pyplot 技巧

matplotlib.pyplot 解決了什麼問題

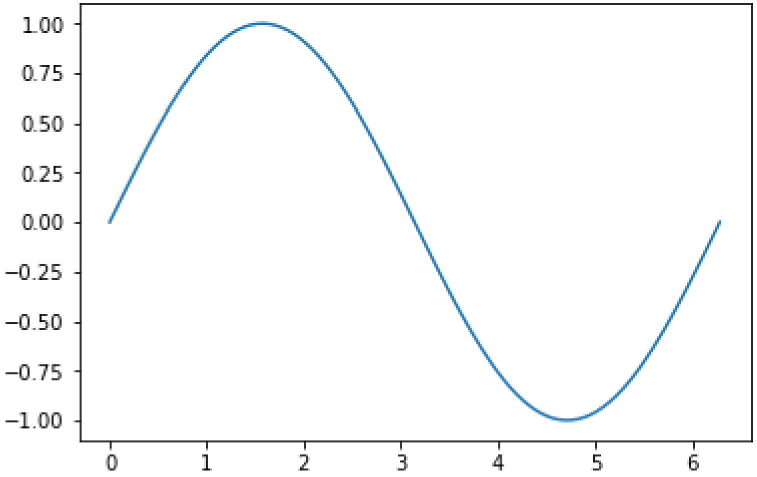
將抽象的資料轉換為具體的輸出。

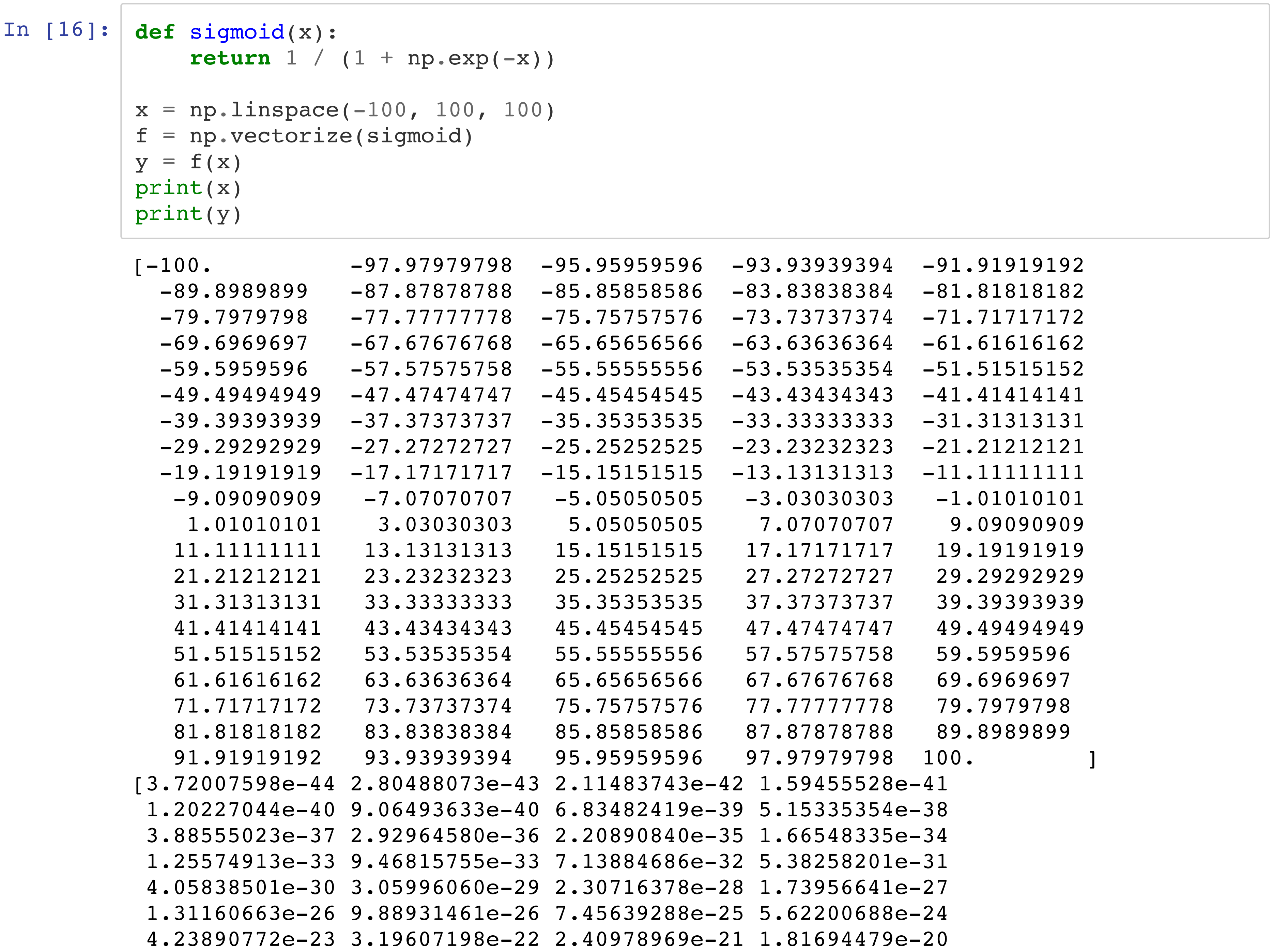


In [11]:
```python
import numpy as np
from numpy import pi

x = np.linspace(0, 2*pi, 100)
f = np.sin(x)
print(x)
print(f)
```

```
[0.         0.06346652 0.12693304 0.19039955 0.25386607 0.31733259
 0.38079911 0.44426563 0.50773215 0.57119866 0.63466518 0.6981317
 0.76159822 0.82506474 0.88853126 0.95199777 1.01546429 1.07893081
 1.14239733 1.20586385 1.26933037 1.33279688 1.3962634  1.45972992
 1.52319644 1.58666296 1.65012947 1.71359599 1.77706251 1.84052903
 1.90399555 1.96746207 2.03092858 2.0943951  2.15786162 2.22132814
 2.28479466 2.34826118 2.41172769 2.47519421 2.53866073 2.60212725
 2.66559377 2.72906028 2.7925268  2.85599332 2.91945984 2.98292636
 3.04639288 3.10985939 3.17332591 3.23679243 3.30025895 3.36372547
 3.42719199 3.4906585  3.55412502 3.61759154 3.68105806 3.74452458
 3.8079911  3.87145761 3.93492413 3.99839065 4.06185717 4.12532369
 4.1887902  4.25225672 4.31572324 4.37918976 4.44265628 4.5061228
 4.56958931 4.63305583 4.69652235 4.75998887 4.82345539 4.88692191
 4.95038842 5.01385494 5.07732146 5.14078798 5.2042545  5.26772102
 5.33118753 5.39465405 5.45812057 5.52158709 5.58505361 5.64852012
 5.71198664 5.77545316 5.83891968 5.9023862  5.96585272 6.02931923
 6.09278575 6.15625227 6.21971879 6.28318531]
[ 0.00000000e+00  6.34239197e-02  1.26592454e-01  1.89251244e-01
  2.51147987e-01  3.12033446e-01  3.71662456e-01  4.29794912e-01
  4.86196736e-01  5.40640817e-01  5.92907929e-01  6.42787610e-01
  6.90079011e-01  7.34591709e-01  7.76146464e-01  8.14575952e-01
  8.49725430e-01  8.81453363e-01  9.09631995e-01  9.34147860e-01
  9.54902241e-01  9.71811568e-01  9.84807753e-01  9.93838464e-01
  9.98867339e-01  9.99874128e-01  9.96854776e-01  9.89821442e-01
  9.78802446e-01  9.63842159e-01  9.45000819e-01  9.22354294e-01
  8.95993774e-01  8.66025404e-01  8.32569855e-01  7.95761841e-01
  7.55749574e-01  7.12694171e-01  6.66769001e-01  6.18158986e-01
  5.67059864e-01  5.13677392e-01  4.58226522e-01  4.00930535e-01
  3.42020143e-01  2.81732557e-01  2.20310533e-01  1.58001396e-01
  9.50560433e-02  3.17279335e-02 -3.17279335e-02 -9.50560433e-02
 -1.58001396e-01 -2.20310533e-01 -2.81732557e-01 -3.42020143e-01
 -4.00930535e-01 -4.58226522e-01 -5.13677392e-01 -5.67059864e-01
 -6.18158986e-01 -6.66769001e-01 -7.12694171e-01 -7.55749574e-01
 -7.95761841e-01 -8.32569855e-01 -8.66025404e-01 -8.95993774e-01
 -9.22354294e-01 -9.45000819e-01 -9.63842159e-01 -9.78802446e-01
 -9.89821442e-01 -9.96854776e-01 -9.99874128e-01 -9.98867339e-01
 -9.93838464e-01 -9.84807753e-01 -9.71811568e-01 -9.54902241e-01
 -9.34147860e-01 -9.09631995e-01 -8.81453363e-01 -8.49725430e-01
 -8.14575952e-01 -7.76146464e-01 -7.34591709e-01 -6.90079011e-01
 -6.42787610e-01 -5.92907929e-01 -5.40640817e-01 -4.86196736e-01
 -4.29794912e-01 -3.71662456e-01 -3.12033446e-01 -2.51147987e-01
 -1.89251244e-01 -1.26592454e-01 -6.34239197e-02 -2.44929360e-16]
```

```
In [13]: %matplotlib inline
         import matplotlib.pyplot as plt

         plt.plot(x, f)
         plt.show()
```





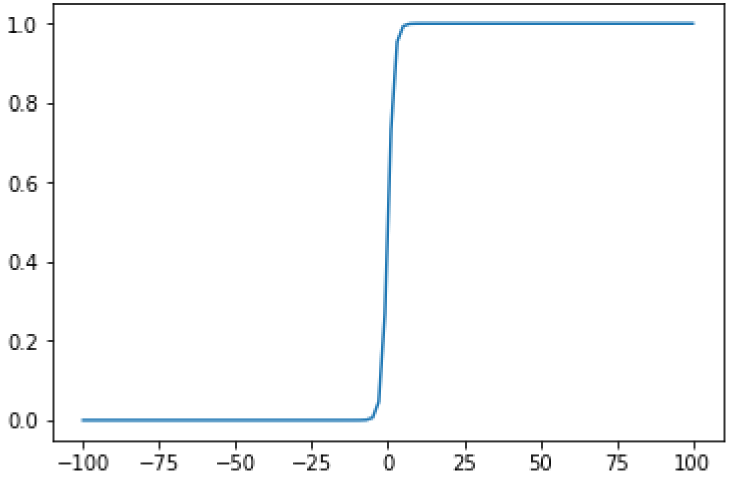
In [16]:

```python
def sigmoid(x):
    return 1 / (1 + np.exp(-x))

x = np.linspace(-100, 100, 100)
f = np.vectorize(sigmoid)
y = f(x)
print(x)
print(y)
```

```
[-100.          -97.97979798  -95.95959596  -93.93939394  -91.91919192
  -89.8989899   -87.87878788  -85.85858586  -83.83838384  -81.81818182
  -79.7979798   -77.77777778  -75.75757576  -73.73737374  -71.71717172
  -69.6969697   -67.67676768  -65.65656566  -63.63636364  -61.61616162
  -59.5959596   -57.57575758  -55.55555556  -53.53535354  -51.51515152
  -49.49494949  -47.47474747  -45.45454545  -43.43434343  -41.41414141
  -39.39393939  -37.37373737  -35.35353535  -33.33333333  -31.31313131
  -29.29292929  -27.27272727  -25.25252525  -23.23232323  -21.21212121
  -19.19191919  -17.17171717  -15.15151515  -13.13131313  -11.11111111
   -9.09090909   -7.07070707   -5.05050505   -3.03030303   -1.01010101
    1.01010101    3.03030303    5.05050505    7.07070707    9.09090909
   11.11111111   13.13131313   15.15151515   17.17171717   19.19191919
   21.21212121   23.23232323   25.25252525   27.27272727   29.29292929
   31.31313131   33.33333333   35.35353535   37.37373737   39.39393939
   41.41414141   43.43434343   45.45454545   47.47474747   49.49494949
   51.51515152   53.53535354   55.55555556   57.57575758   59.5959596
   61.61616162   63.63636364   65.65656566   67.67676768   69.6969697
   71.71717172   73.73737374   75.75757576   77.77777778   79.7979798
   81.81818182   83.83838384   85.85858586   87.87878788   89.8989899
   91.91919192   93.93939394   95.95959596   97.97979798  100.        ]
[3.72007598e-44 2.80488073e-43 2.11483743e-42 1.59455528e-41
 1.20227044e-40 9.06493633e-40 6.83482419e-39 5.15335354e-38
 3.88555023e-37 2.92964580e-36 2.20890840e-35 1.66548335e-34
 1.25574913e-33 9.46815755e-33 7.13884686e-32 5.38258201e-31
 4.05838501e-30 3.05996060e-29 2.30716378e-28 1.73956641e-27
 1.31160663e-26 9.88931461e-26 7.45639288e-25 5.62200688e-24
 4.23890772e-23 3.19607198e-22 2.40978969e-21 1.81694479e-20
 1.36994875e-19 1.03292053e-18 7.78806380e-18 5.87208171e-17
 4.42746035e-16 3.33823780e-15 2.51698055e-14 1.89776507e-13
 1.43088601e-12 1.07886630e-11 8.13448792e-11 6.13328023e-10
 4.62440004e-09 3.48672722e-08 2.62893872e-07 1.98217642e-06
 1.49451152e-05 1.12672884e-04 8.48910885e-04 6.36532037e-03
 4.60755062e-02 2.66960083e-01 7.33039917e-01 9.53924494e-01
 9.93634680e-01 9.99151089e-01 9.99887327e-01 9.99985055e-01
 9.99998018e-01 9.99999737e-01 9.99999965e-01 9.99999995e-01
 9.99999999e-01 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00]
```

In [17]:

```
%matplotlib inline
import matplotlib.pyplot as plt

plt.plot(x, y)
plt.show()
```



認識範例資料集： gapminder

視覺化的典範

https://youtu.be/jbkSRLYSojo (https://youtu.be/jbkSRLYSojo)

關於 Gapminder, Hans Rosling 與 Factfulness

- Gapminder (https://www.gapminder.org/)
- Hans Rosling (https://en.wikipedia.org/wiki/Hans_Rosling)
- Factfulness (https://www.youtube.com/watch?v=5uooRe07mYM)

獲得 gapminder 範例資料集

In [18]:

```
import pandas as pd

csv_url = "https://storage.googleapis.com/learn_pd_like_tidyverse/gapminder.csv"
gapminder = pd.read_csv(csv_url)
gapminder.head()
```

Out[18]:

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
3	Afghanistan	Asia	1967	34.020	11537966	836.197138
4	Afghanistan	Asia	1972	36.088	13079460	739.981106

matplotlib.pyplot 基礎

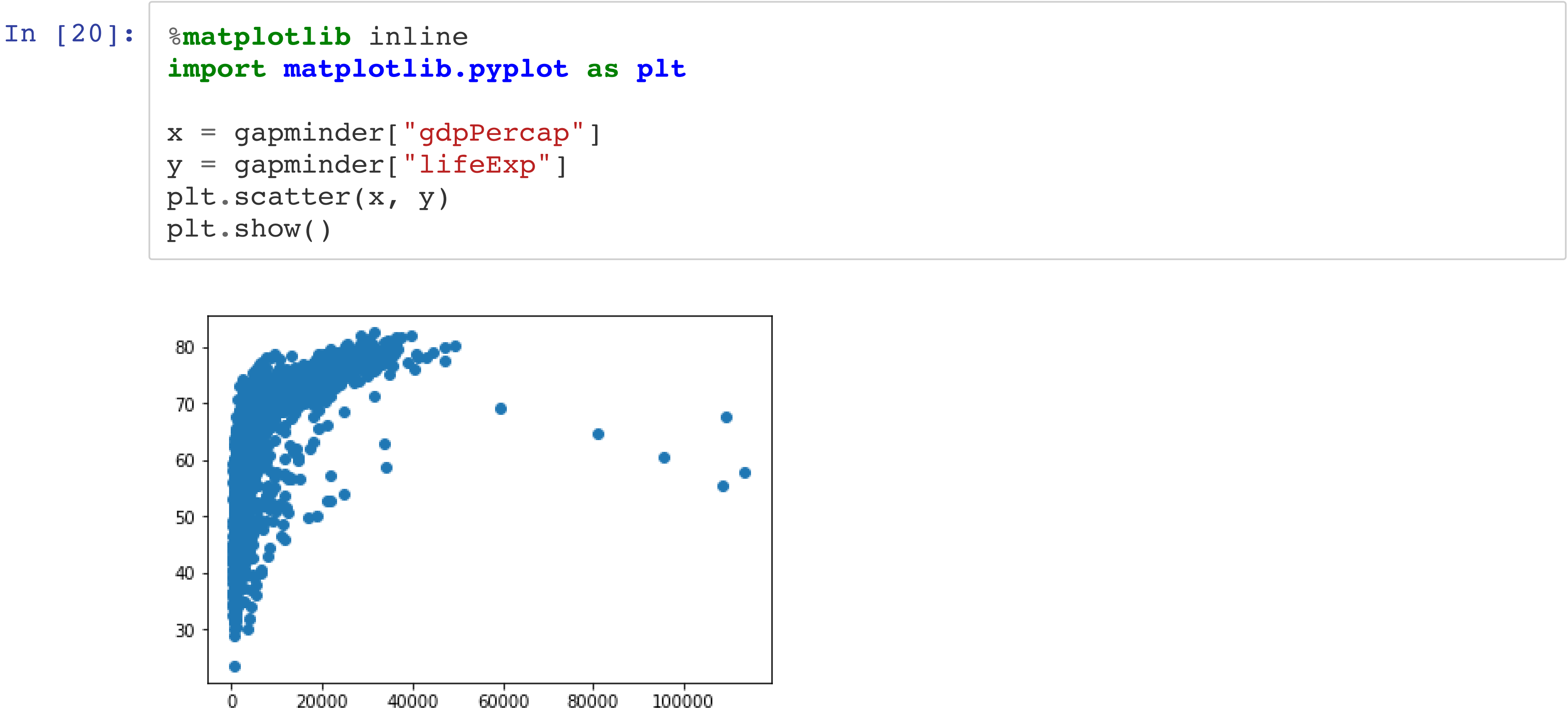
什麼是 matplotlib?

> *Matplotlib is a Python 2D plotting library which produces publication quality figures.*

什麼是 pyplot?

> *matplotlib.pyplot is a collection of command style functions that make matplotlib work like MATLAB.*

觀察資料相關性的散佈圖（Scatter Plot）



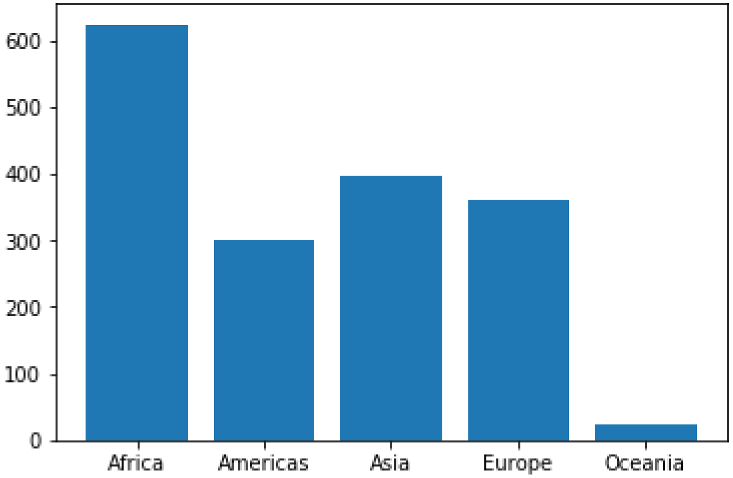
In [20]:

```
%matplotlib inline
import matplotlib.pyplot as plt

x = gapminder["gdpPercap"]
y = gapminder["lifeExp"]
plt.scatter(x, y)
plt.show()
```

觀察類別型資料的長條圖（Bar Plot）

In [29]:

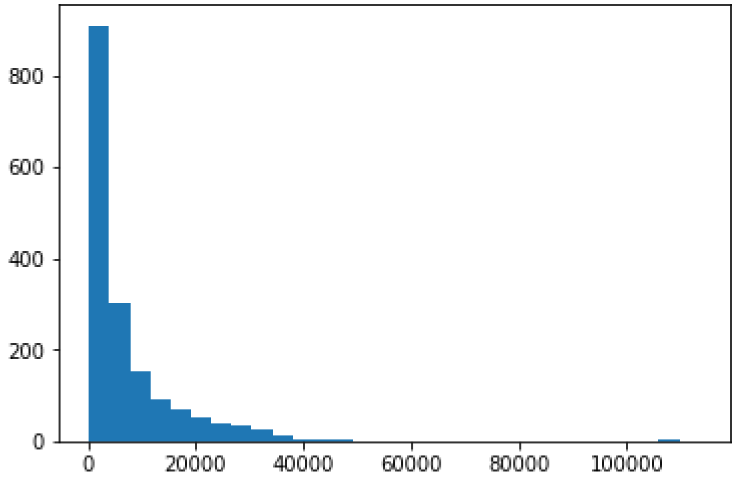
```
%matplotlib inline
import matplotlib.pyplot as plt

y = gapminder.groupby("continent").count()
labels = y.index
x = range(y.shape[0])
y = y["country"].values
plt.bar(x, y)
plt.xticks(x, labels)
plt.show()
```



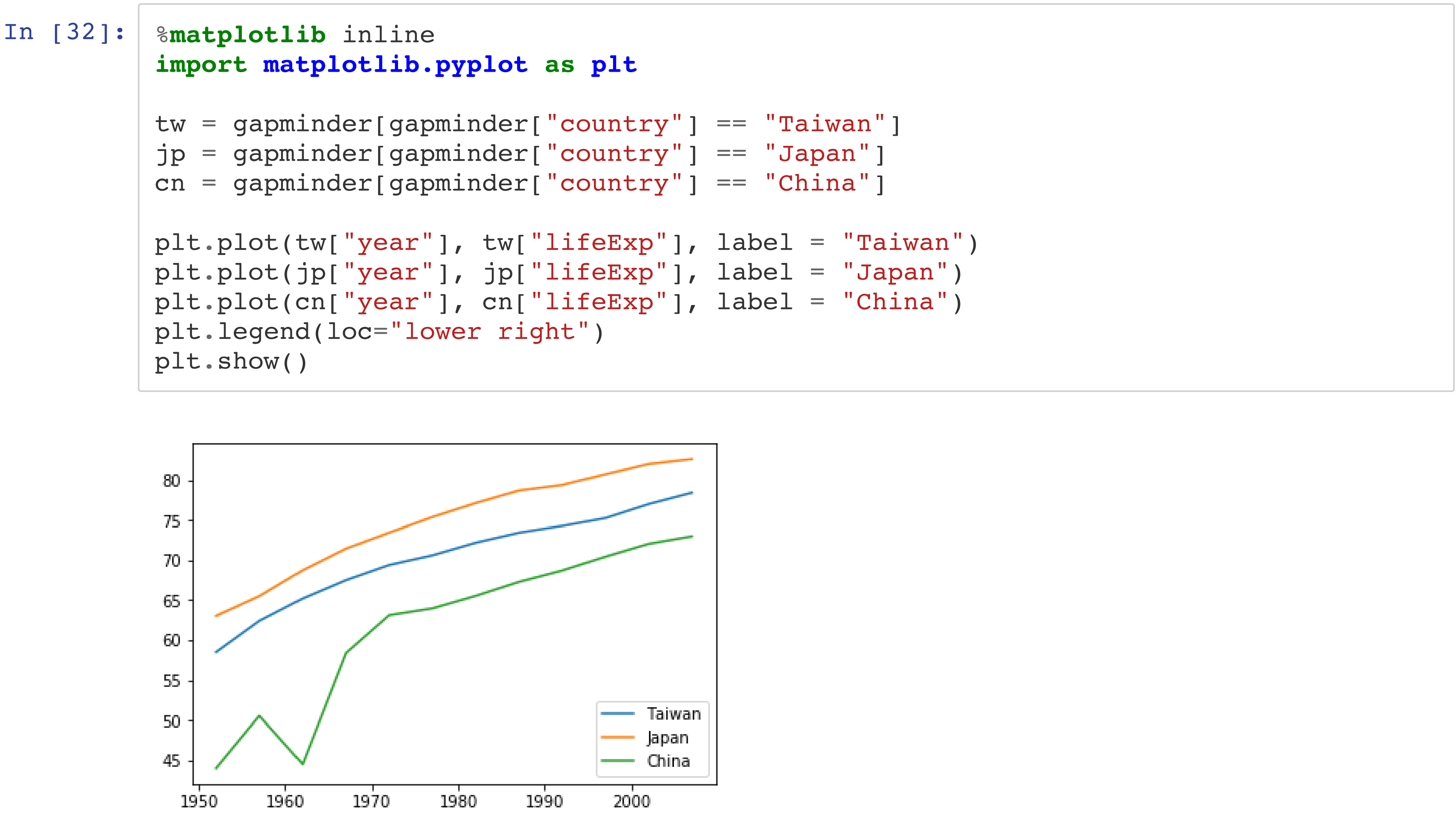
觀察資料散佈的直方圖（Histogram）

In [31]:

```
%matplotlib inline
import matplotlib.pyplot as plt

plt.hist(gapminder["gdpPercap"], bins=30)
plt.show()
```



觀察數值變化趨勢的線圖（Line Plot）



In [32]:

```
%matplotlib inline
import matplotlib.pyplot as plt

tw = gapminder[gapminder["country"] == "Taiwan"]
jp = gapminder[gapminder["country"] == "Japan"]
cn = gapminder[gapminder["country"] == "China"]

plt.plot(tw["year"], tw["lifeExp"], label = "Taiwan")
plt.plot(jp["year"], jp["lifeExp"], label = "Japan")
plt.plot(cn["year"], cn["lifeExp"], label = "China")
plt.legend(loc="lower right")
plt.show()
```

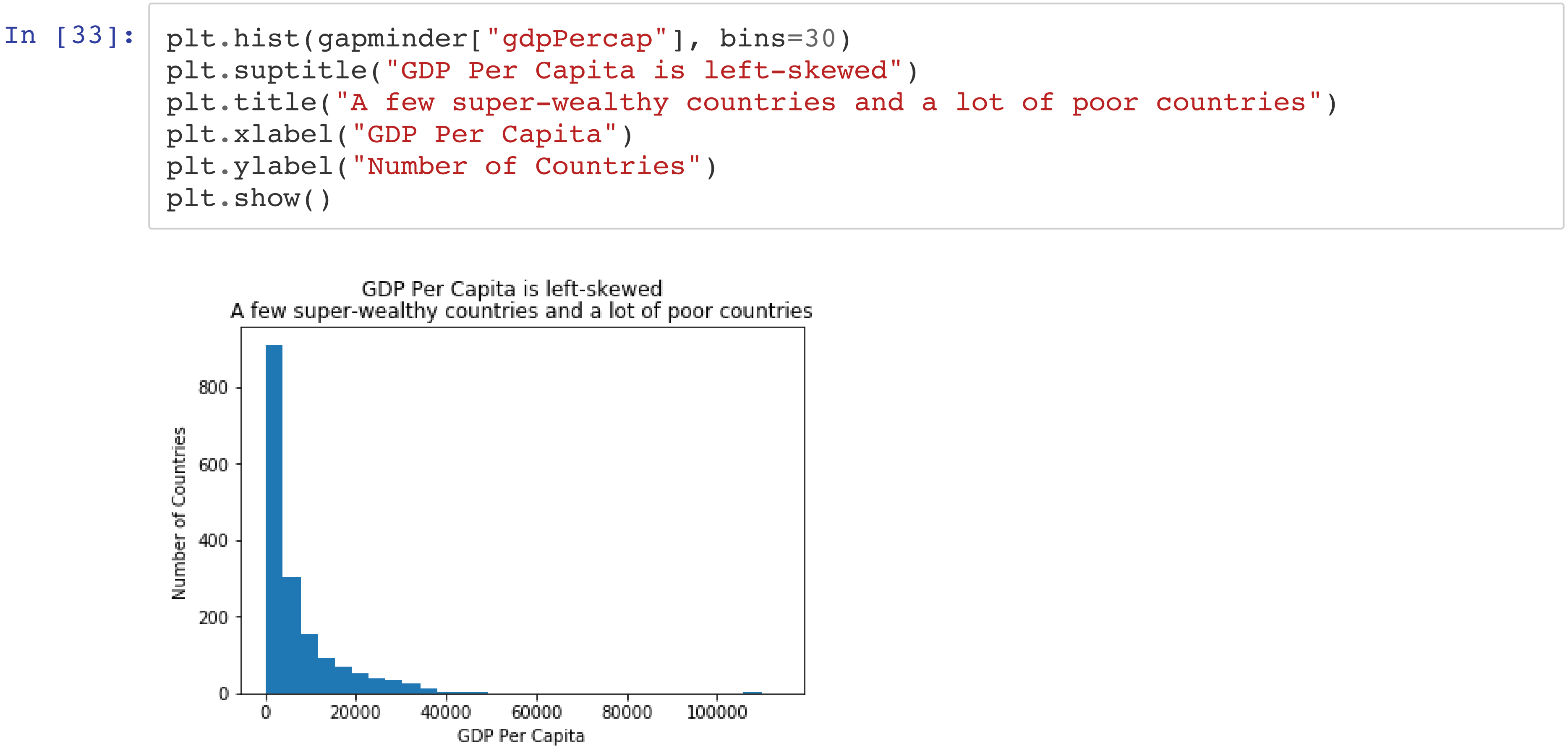
matplotlib.pyplot 技巧

其他 matplotlib.pyplot 觀念

- 加入圖標題與軸標籤
- 加入註釋
- 調整座標軸
- 加入與調整圖例
- 在一個畫布上繪製多個子圖形

加入圖標題與軸標籤

- `plt.title()`
- `plt.suptitle()`
- `plt.xlabel()`
- `plt.ylabel()`



In [33]:

```
plt.hist(gapminder["gdpPercap"], bins=30)
plt.suptitle("GDP Per Capita is left-skewed")
plt.title("A few super-wealthy countries and a lot of poor countries")
plt.xlabel("GDP Per Capita")
plt.ylabel("Number of Countries")
plt.show()
```

GDP Per Capita is left-skewed
A few super-wealthy countries and a lot of poor countries
Number of Countries
0
200
400
600
800
0
20000
40000
60000
80000
100000
GDP Per Capita

調整座標軸

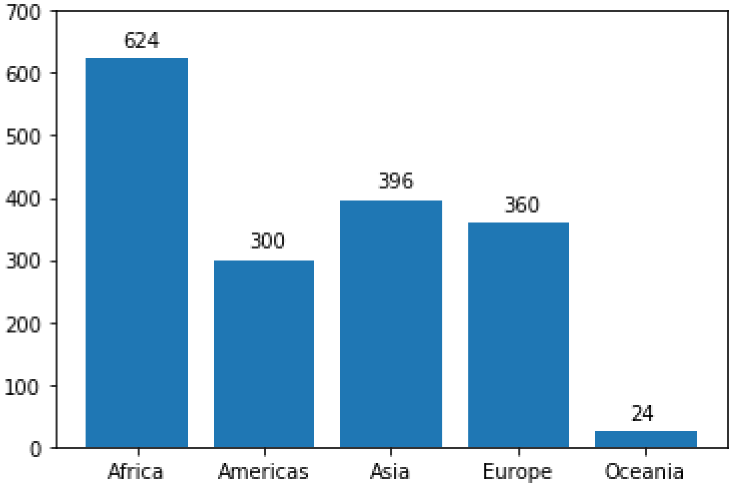
- `plt.xlim()`
- `plt.ylim()`
- `plt.xticks()`
- `plt.yticks()`

加入註釋

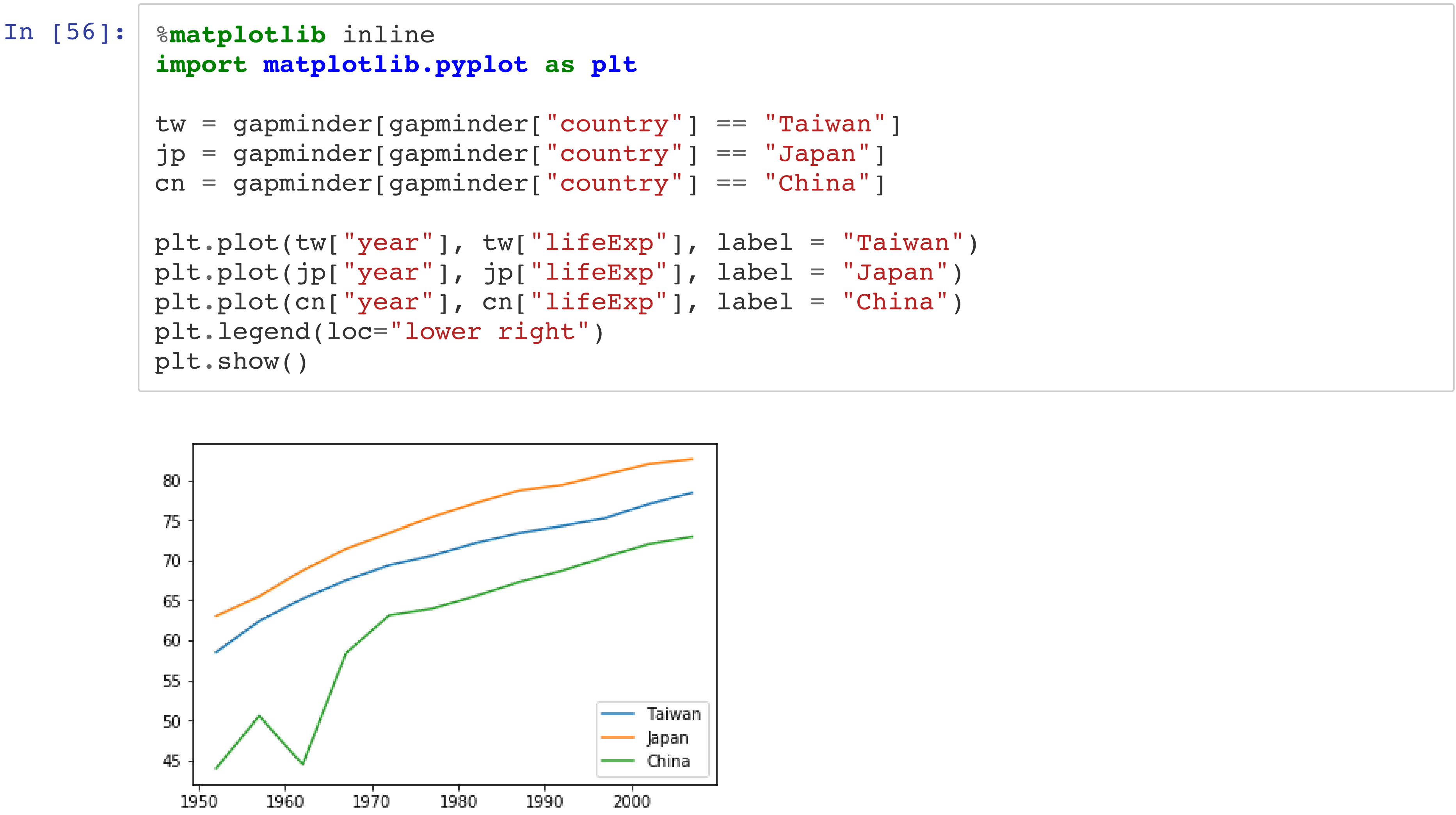
- `plt.text()`

In [55]:

```
y = gapminder.groupby("continent").count()
labels = y.index
x = range(y.shape[0])
y = y["country"].values
plt.bar(x, y)
plt.xticks(x, labels)
for i, v in enumerate(y):
    plt.text(i - 0.12, v + 20, "{:.0f}".format(v))
plt.ylim([0, 700])
plt.show()
```



加入與調整圖例



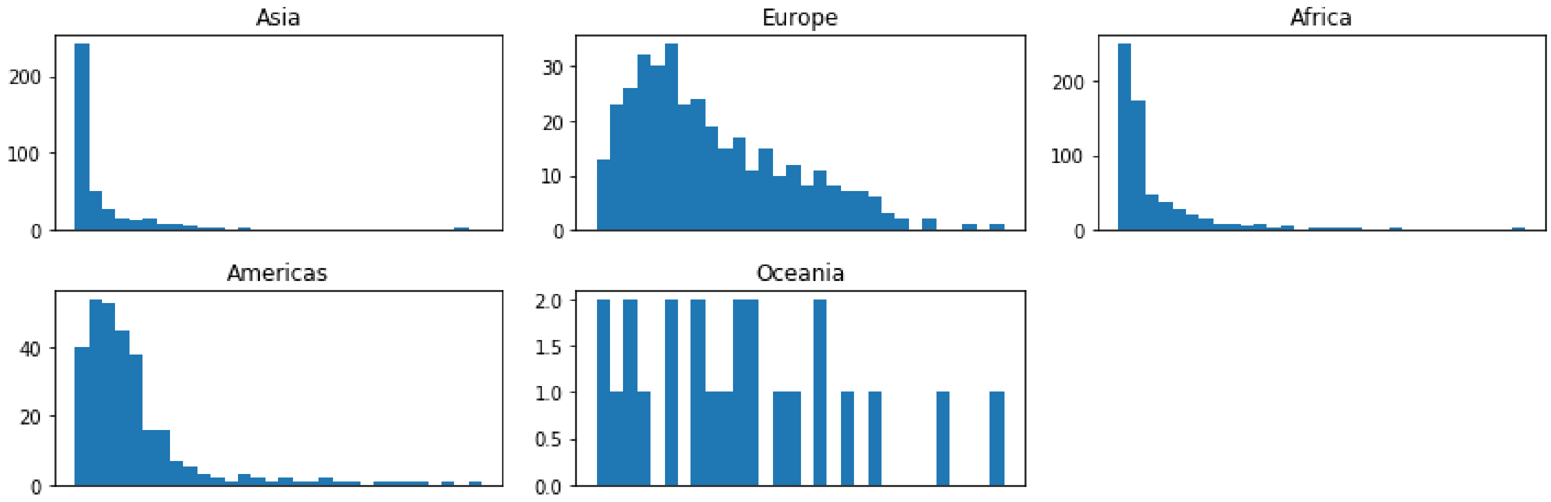
In [56]:

```
%matplotlib inline
import matplotlib.pyplot as plt

tw = gapminder[gapminder["country"] == "Taiwan"]
jp = gapminder[gapminder["country"] == "Japan"]
cn = gapminder[gapminder["country"] == "China"]

plt.plot(tw["year"], tw["lifeExp"], label = "Taiwan")
plt.plot(jp["year"], jp["lifeExp"], label = "Japan")
plt.plot(cn["year"], cn["lifeExp"], label = "China")
plt.legend(loc="lower right")
plt.show()
```

在一個畫布上繪製多個子圖形

- `plt.subplots()`

```
In [59]: fig, axes = plt.subplots(2, 3, figsize=(12, 4))
         continents = gapminder["continent"].unique()
         # 繪製子圖
         for ax, cont in zip(axes.ravel(), continents):
             ax.hist(gapminder[gapminder["continent"] == cont]["gdpPercap"], bins=30)
             ax.set_xticks([])
             ax.set_title(cont)
         axes[1, 2].set_visible(False)
         plt.tight_layout()
         plt.show()
```



延伸閱讀

- Matplotlib: Python plotting - Matplotlib 3.0.3 documentation (https://matplotlib.org/)