

Python 小畫家教學

🕒 Created	@December 29, 2023 3:11 PM
🏷️ Tags	

[numpy](#)

[產生假資料](#)

[matplotlib](#)

[sympy](#)

[代數宣告](#)

[求交點](#)

[求微分](#)

[求代入值](#)

[成為小畫家](#)

[第一步 - 畫一條線](#)

[第二步 - 畫二條線](#)

[第三步 - 畫一個點](#)

[恭喜成為小畫家](#)

Tech stack

- matplotlib
 - 極懶人繪圖套件
- sympy
 - 代數運算
 - T 檢定
- numpy
 - 陣列運算

使用 terminal 安裝套件

```
pip install matplotlib numpy sympy
```

隨便開一個 py 檔案將以下程式依序複製貼上

numpy

產生假資料

```
import matplotlib.pyplot as plt
import numpy as np

# 產生最小為 0 最大為 100 長度為 10 的線性陣列
x = np.linspace(start=0, stop=100, num=10)
print(f"x={x}")

# [  0.          11.11111111  22.22222222  33.33333333  44.44444444
#  55.55555556  66.66666667  77.77777778  88.88888889 100.]

# 對 x 進行運算
y = 2 * x
print(f"y={y}")

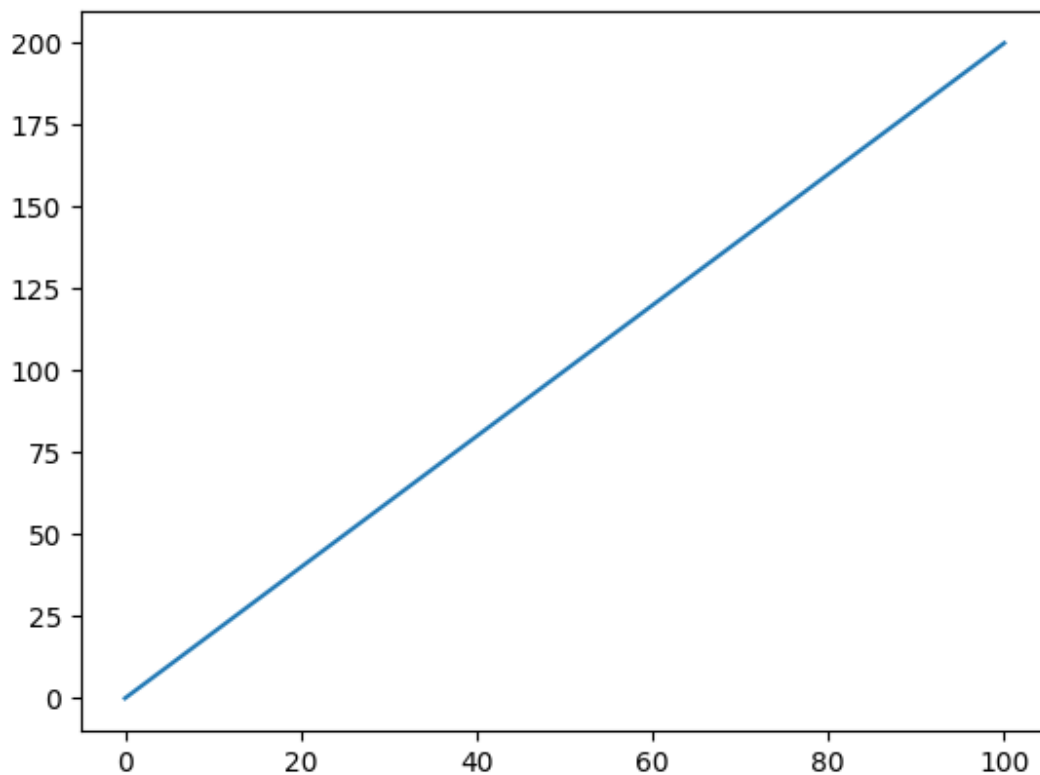
# [  0.          22.22222222  44.44444444  66.66666667  88.88888889
# 111.11111111 133.33333333 155.55555556 177.77777778 200.]
```

matplotlib

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(start=0, stop=100, num=10)
y = 2 * x
```

```
# 加上這兩行就能成為小畫家惹  
plt.plot(x, y)  
plt.show()
```



sympy

代數宣告

```
import sympy as sp  
  
# 建構時代入的字串是代數名稱，相同名稱的會被視為同一個代數  
x = sp.Symbol("x")  
y = sp.Symbol("x")
```

```
print(f"y * x = {y * x}")  
# x**2
```

求交點

```
import sympy as sp  
  
x = sp.Symbol("x")  
  
y1 = x  
y2 = 10 - x  
  
# 兩式相減對 x 解聯立  
intersection = sp.solve(y1 - y2, x)  
print(intersection)  
# [5]
```

求微分

```
import sympy as sp  
  
x = sp.Symbol("x")  
  
y = x**3 + x**2 + 1  
  
# y 對 x 做一次微分  
dy_dx = sp.diff(y, x, 1)  
print(dy_dx)  
# 3*x**2 + 2*x
```

求代入值

```
import sympy as sp

x = sp.Symbol("x")

y = x**3 + x**2 + 1

# 帶入 x = 1
print(y.subs(x, 1))
# 3
```

成為小畫家

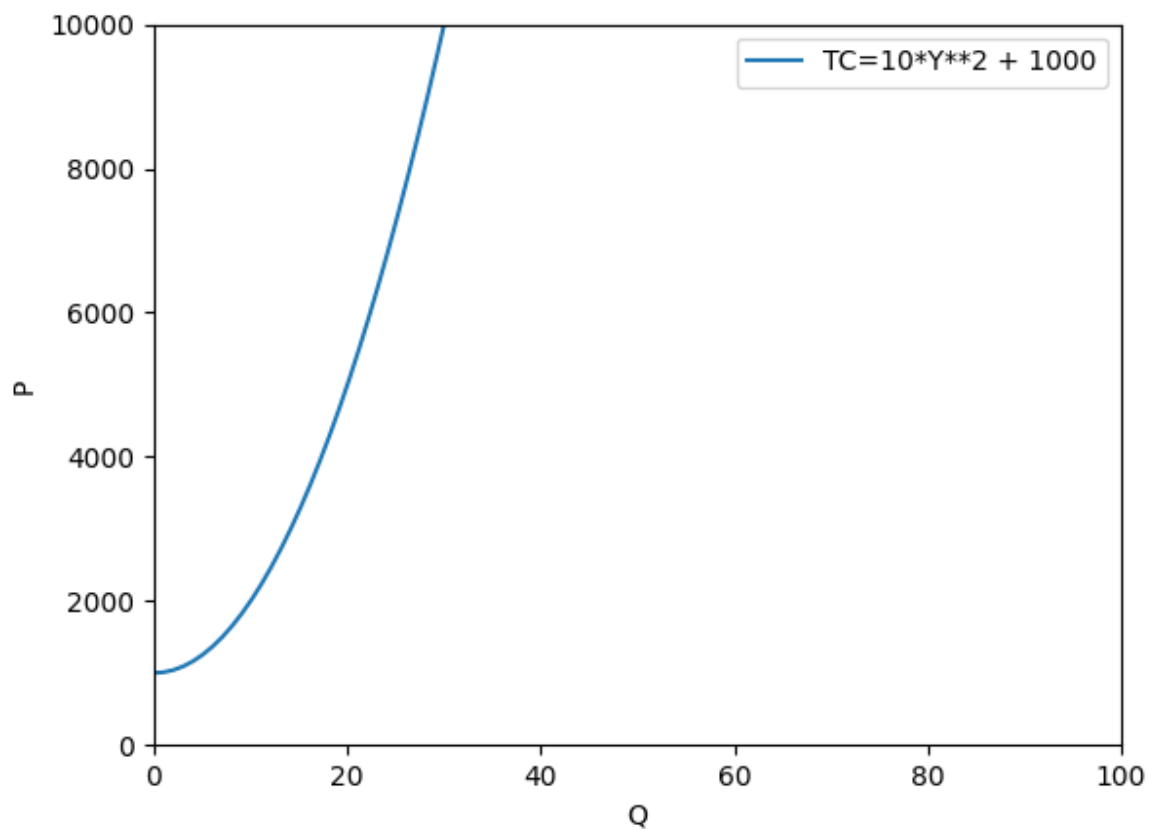
第一步 - 畫一條線

```
import matplotlib.pyplot as plt
import numpy as np
import sympy as sp

# Example.
#  $TC = 10Y^2 + 1,000$ 
y = sp.Symbol("Y")
TC = 10 * y**2 + 1000

# X 軸產生假資料
y_values = np.linspace(0, 100, 100)
# Sympy 跟 Numpy 互相不認識，需要將 Sympy 表達式轉成 Numpy 看得懂的格式
TC_to_numpy = sp.lambdify(y, TC, 'numpy')
# 產生 Y 軸假資料
TC_values = TC_to_numpy(y_values)
# 畫線
plt.plot(y_values, TC_values, label=f"TC={str(TC)}")
# 座標軸名稱
plt.xlabel("Q")
```

```
plt.ylabel("P")
# 座標軸範圍
plt.ylim(bottom=0, top=10000)
plt.xlim(left=0, right=100)
# 顯示圖標
plt.legend()
plt.show()
```



第二步 - 畫二條線

```
import matplotlib.pyplot as plt
import numpy as np
import sympy as sp
```

```

# Example.
#  $TC=10Y^2+ 1,000$ 
y = sp.Symbol("Y")
TC = 10 * y**2 + 1000
AC = TC / y
MC = sp.diff(TC, y, 1)
# 求 MC 與 AC 的交點
intersect_x = sp.solve(sp.Eq(AC, MC), y)
print(intersect_x)

y_values = np.linspace(-100, 100, 100)

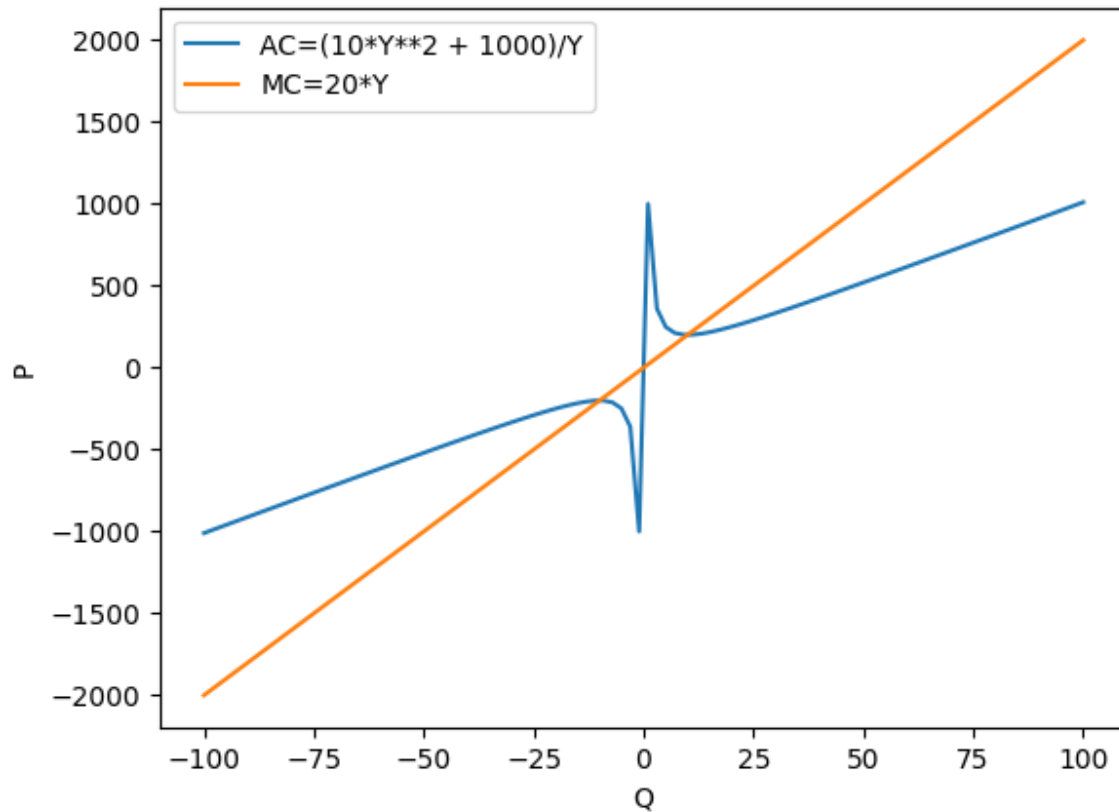
# 產生 AC 資料
AC_values = sp.lambdify(y, AC, 'numpy')(y_values)
# 產生 MC 資料
MC_values = sp.lambdify(y, MC, 'numpy')(y_values)

plt.plot(y_values, AC_values, label=f"AC={str(AC)}")
plt.plot(y_values, MC_values, label=f"MC={str(MC)}")

plt.xlabel("Q")
plt.ylabel("P")
plt.ylim()
plt.xlim()
plt.legend()

# 折線圖會強制連接兩點
plt.show()

```



第三步 - 畫一個點

```
import matplotlib.pyplot as plt
import numpy as np
import sympy as sp

# Example.
# TC=10Y^2+ 1,000
y = sp.Symbol("Y")
TC = 10 * y**2 + 1000
AC = TC / y
MC = sp.diff(TC, y, 1)
# 求 MC 與 AC 的交點(正整數)
intersect_q = sp.solve(sp.Eq(AC, MC), y)[1]
# 10
```



```

# 代入求值
intersect_p = MC.subs(y, intersect_q)
# 200

y_values = np.linspace(1, 100, 100)

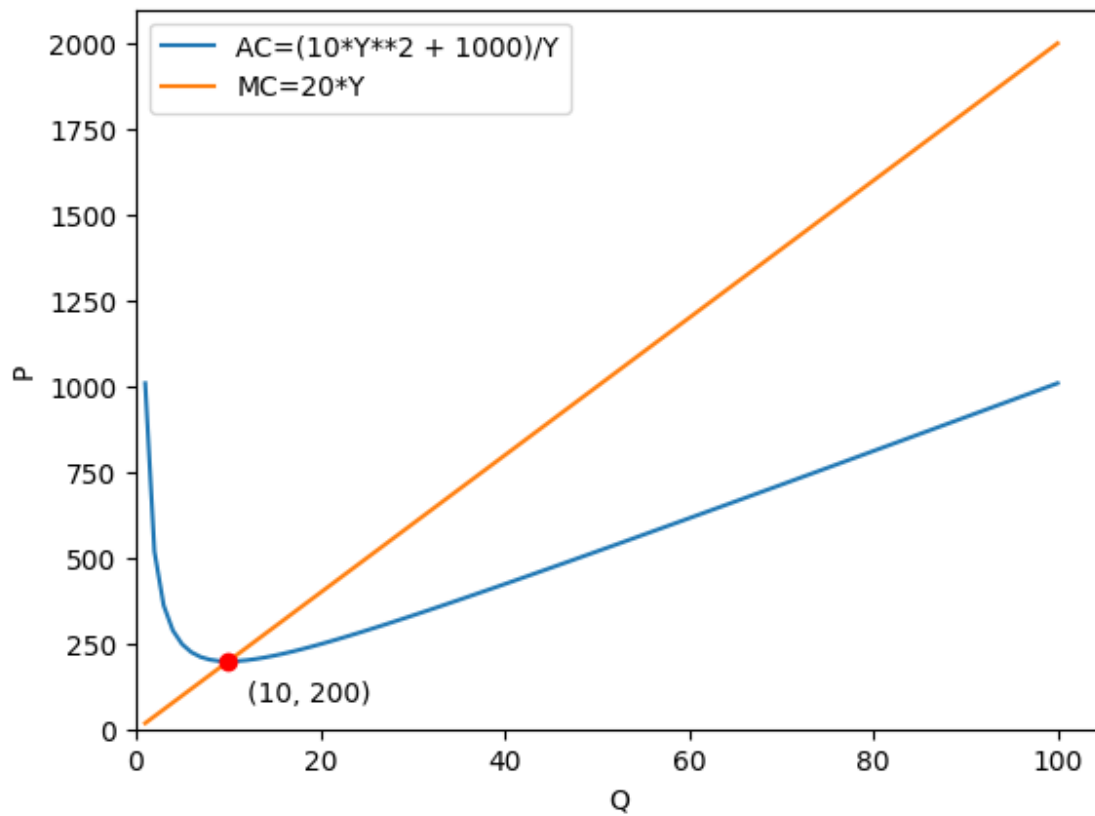
AC_values = sp.lambdify(y, AC, 'numpy')(y_values)
MC_values = sp.lambdify(y, MC, 'numpy')(y_values)

plt.plot(y_values, AC_values, label=f"AC={str(AC)}")
plt.plot(y_values, MC_values, label=f"MC={str(MC)}")

# 畫一個紅色圓點
plt.plot(intersect_q, intersect_p, 'ro')
# 在這個點的右下方加入說明
plt.annotate(f"({intersect_q}, {intersect_p})", (intersect_q, intersect_p),
             xytext=(30, -15), ha='center')

plt.xlabel("Q")
plt.ylabel("P")
plt.ylim(bottom=0)
plt.xlim(left=0)
plt.legend()
plt.show()

```



恭喜成為小畫家