

1 蒙地卡羅法

1.1 介紹

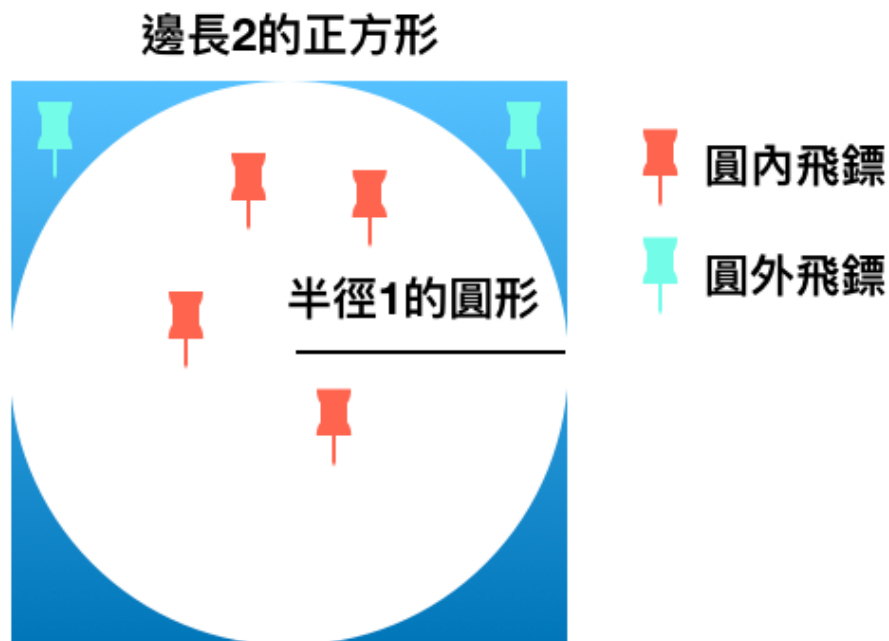
蒙地卡羅是摩納哥的一個城市，以賭博聞名。蒙地卡羅法使用暴力破解題目，簡單來說，就是以機率和統計去猜測一件我們不知道的事！雖然精確度會有一小小點的差異，但用這思維方式來解釋事情非常有意思，而且可以利用在許多機率統計的演算法中！

1.2 問題

幫大家複習一下圓面積的公式 $\pi \times r^2$ 我們今天假設我們已經知道圓的面積和半徑 r^2 成正比，但我們卻不知道前面的常數 π 究竟是多少我們並不知道，我們該如何求出 π 呢？

1.3 構思

你也許會告訴我，我們可以用積分來求得面積，但今天我們用另外一個想法，也就是蒙地卡羅法，我假設我有一個方形的區域，裡面有一個圓形的區域，我們是一個隨機的飛鏢射手，我們的飛鏢只會落在方形的區域中，如下圖所示



圖：區域示意圖

那我們的面積會是



$$\text{圓面積} = \pi \times r^2 = \text{正方形邊長}^2 \times \frac{\text{落入圓內的飛鏢數}}{\text{總共投擲飛鏢數}}$$

圖: 圓面積

由於 $r = 1$ ，所以我們就可以輕鬆得到 π 究竟是多少了！

1.4 寫作程式

廢話不多說，先來寫個程式，這裡比較要注意的是判定圓內還是圓外，我們要利用圓的方程式 $x^2 + y^2 = r^2$ 來實現判定，由於我們的半徑 $r = 1$ ，所以 $x^2 + y^2 < 1$ 即是圓內

1.4.1 先來一次

築夢踏實，先試試看一次

```
[程式]: import random
# 我們使用 random 的 uniform 來幫我們產生 x(-1~1) 和 y(-1~1)
x = random.uniform(-1, 1)
y = random.uniform(-1, 1)
# 判定在內還外，這裡我將圓上 (=1) 當作外，這些點非常稀少，無傷大雅
print("座標點: (" + x + ", " + y + ")")
if x ** 2 + y ** 2 < 1:
    print("圓內")
else:
    print("圓外")
```

座標點: (0.9212116196837699 , -0.13742701294155113)
圓內

1.4.2 來試試看多次

把迴圈加進去，由於我們是固定次數的迴圈，所以我們使用 `for in range(次數)` 來達成

```
[程式]: import random
# 紀錄圓內的飛鏢數
incircle = 0
# 紀錄圓外的飛鏢數
outcircle = 0
# 丟一千萬支飛鏢
for i in range(10000000):
```



```

# 我們使用 random 的 uniform 來幫我們產生 x(-1~1) 和 y(-1~1)
x = random.uniform(-1, 1)
y = random.uniform(-1, 1)
# 判定在內還外，這裡我將圓上 (=1) 當作外，這些點非常稀少，無傷大雅
if x ** 2 + y ** 2 < 1:
    incircle = incircle + 1
else:
    outcircle = outcircle + 1
print("圓內數:", incircle)
print("圓外數:", outcircle)
total = incircle + outcircle
probability = incircle / total
print("圓內機率:", probability * 100, "%")
# 4 是正方形面積
print("圓周率:", 4 * probability)

```

圓內數: 7853147
 圓外數: 2146853
 圓內機率: 78.53147 %
 圓周率: 3.1412588

1.4.3 結論

你可以看到我們算出的圓周率事實上已經開始穩定到小數第三位，如果你做更多次，就會得到一個更穩定的結果！

1.5 視覺化

接下來就比較進階一點了，我們要來把結果畫成一個動圖，Just For Fun，覺得太難的讀者可以跳過！

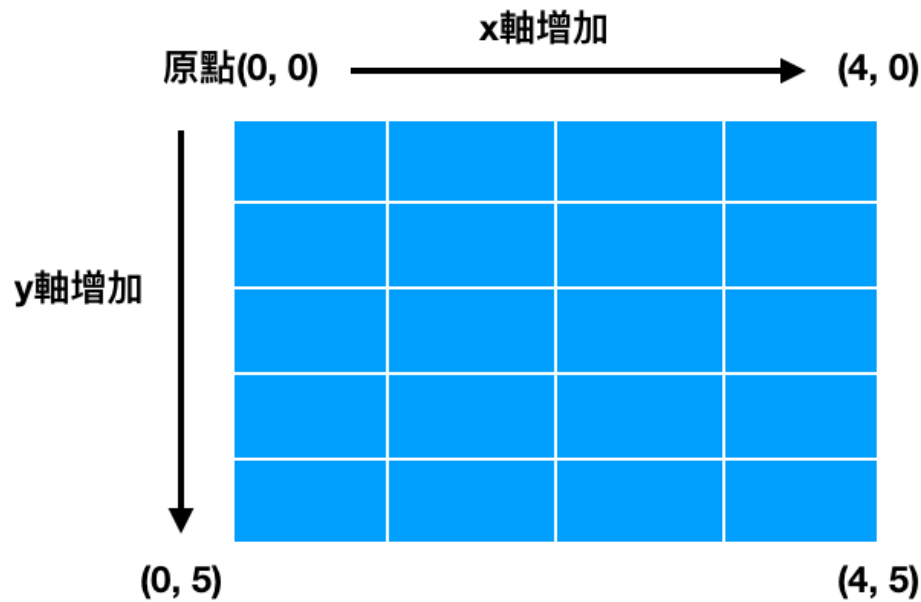
1.5.1 環境設置

1. PIL 函式庫: 讀者請透過 PyCharm 或者命令列安裝 Pillow 函式庫，PIL 函式庫是 Python 一個影像處理的常用函式庫，跟 opencv 並稱 Python 兩大影像處理函式庫
2. 中文字型: 由於 PIL 內建使用英文字型，會導致中文字型顯示有誤，建議讀者先在 https://drive.google.com/open?id=1jzbbm9s_H608zvV7I2h5-oq9GQYL0qPn 下載 Google 提供的 Noto 系列中文字型，並放在跟專案同一層的資料夾下

1.5.2 座標軸

在程式中的座標軸系統跟我們平常習慣的不太一樣，左上為原點，往下 y 軸增加



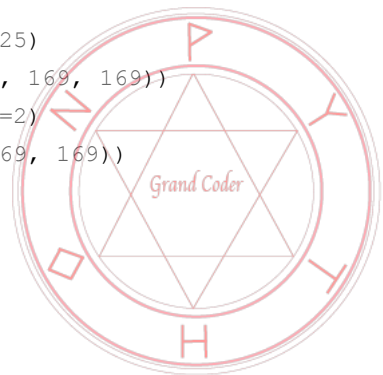


圖：程式的座標軸系統

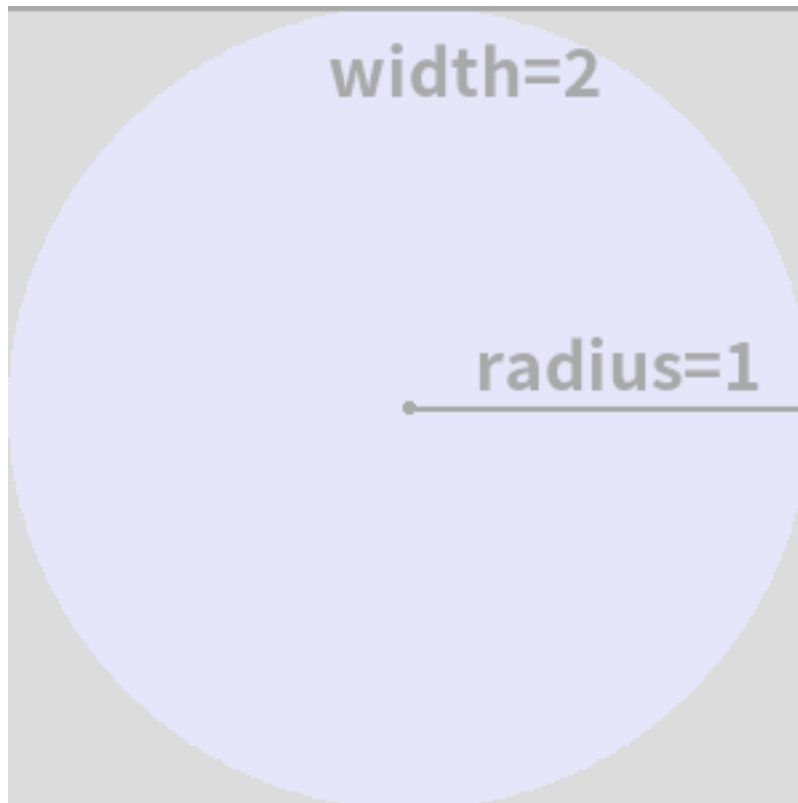
1.5.3 步驟 1. 準備環境

我們先把要畫的圓和方框準備好

```
[程式]: from PIL import Image, ImageDraw, ImageFont
width = 300
height = 300
# new 出一個 300x300 的空白圖片, 背景設置成灰色 (R=220, G=220, B=220)
image = Image.new('RGBA', (width, height), color=(220,220,220))
# 準備畫筆
draw = ImageDraw.Draw(image)
# 畫圓, 這裡的 tuple 四個位置分別是 (左上 x, 左上 y, 右下 x, 右下 y)
draw.ellipse((0, 0, width, height), fill=(230,230,250))
# 畫線, 畫半徑的線
draw.line((width/2, height/2, width, height/2), fill=(169, 169, 169), width=2)
# 畫圓心
draw.ellipse((width/2 - 2, height/2 - 2, width/2 + 2, height/2 + 2),
             fill=(169, 169, 169))
# 準備字型, 25 字體大小
fnt = ImageFont.truetype("./NotoSansCJKtc-Bold.otf", 25)
draw.text((175, 115), "radius=1", font=fnt, fill=(169, 169, 169))
draw.line((0, 0, 300, 0), fill=(169, 169, 169), width=2)
draw.text((120, 5), "width=2", font=fnt, fill=(169, 169, 169))
# 如果在普通.py 請使用 image.show()
image
```



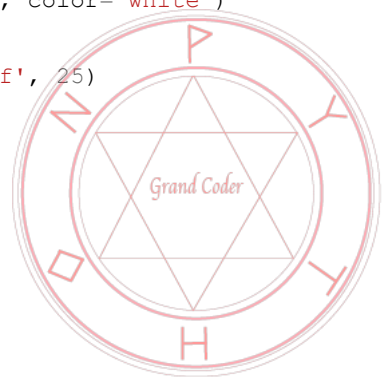
Out[16]:



1.5.4 步驟 2. 開始丟飛鏢

開始把我們隨機的點畫入環境

```
[程式]: import random
# 準備的 list 是來接等等我們要做 GIF 的所有 image
imglist = []
incircle = 0
outcircle = 0
for times in range(100):
    # 準備 900*300 來接剛剛的圓
    fullimage = Image.new('RGBA', (3 * width, height), color="white")
    fulldraw = ImageDraw.Draw(fullimage)
    fnt = ImageFont.truetype('./NotoSansCJKtc-Bold.otf', 25)
    for point in range(100):
        x = random.uniform(-1, 1)
        y = random.uniform(-1, 1)
        # 把 x: -1, 1 轉化到座標系統的 0, 300
```



```

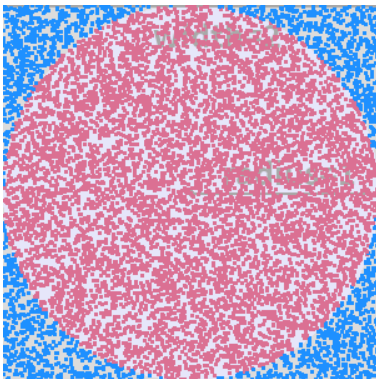
# 把 y: 1, -1 轉化到座標系統的 0, 300 (反過來)
coordx = int(x * width/2 + width/2)
coordy = int(-1 * y * height/2 + height/2)
if x ** 2 + y ** 2 < 1:
    incircle = incircle + 1
    # 在圓內的我以紅圓點表示
    draw.ellipse((coordx - 1, coordy - 1, coordx + 1, coordy + 1),
                  fill=(219, 112, 147))
else:
    outcircle = outcircle + 1
    # 在圓外的我以紅圓點表示
    draw.ellipse((coordx - 1, coordy - 1, coordx + 1, coordy + 1),
                  fill=(30, 144, 255))

# 把 image 貼在整張大圖的原點
fullimage.paste(image.copy())
fnt = ImageFont.truetype('./NotoSansCJKtc-Bold.otf', 25)
msg = "已經射了 " + str(incircle + outcircle) + " 支飛鏢\n"
msg = msg + "圓內: " + str(incircle) + "支\n"
msg = msg + "圓外: " + str(outcircle) + "支\n"
msg = msg + "圓周率 (圓的面積): " + str(4 * incircle / (incircle + outcircle)) + "\n"
# 把準備好的 message 畫入圖中
fulldraw.multiline_text((width + 20, height/4),
                          msg,
                          font=fnt,
                          fill=(105, 105, 105))

# 先放入 imglist, 等等把所有 image 做成 gif
# 這裡要注意, 一定要使用 copy 把現在的狀態 copy 一份, 不然所有 append 進去的都會是後續狀態
imglist.append(fullimage.copy())
# 秀出最後一張, .py 一樣要加上.show()
imglist[-1]

```

Out[17]:



已經射了 10000 支飛鏢
 圓內: 7865支
 圓外: 2135支
 圓周率(圓的面積): 3.146



1.5.5 做成 GIF

其實 GIF 就只是很多張圖片的連結而已，我們把我們剛剛儲存在 `imglist` 裡面的所有東西儲存即可

```
[程式]: # 先拿出 imglist[0] 當第一張
        # 再將後續的圖接續上去
        # duration: 每一張圖放映間隔
        # loop: 1 代表重播一次, 0 代表永久循環
imglist[0].save('circle.gif',
               save_all=True,
               append_images=imglist[1:],
               duration=200,
               loop=1)
```

