

## 作品三、 機率分配的樣貌

姓名：謝元皓

學號：410978002

目標：

- 繪製曾學過的分配函數，含連續與離散型。
- 連續型分配包括常態、卡方、T、Beta、F 等五種。利用改變分配函數的參數，觀察其分配函數的「長相」；畫出所有可能的「形狀」並說明（或標示）與參數間的關係。
- 離散型則選擇 Binomial 分配。
- 連續型分配函數繪圖以 PDF 為主。離散型分配含 PMF（stem 圖）及 CDF（stairs 圖）。
- 其他：除上述的基本要求外，附加的內容。

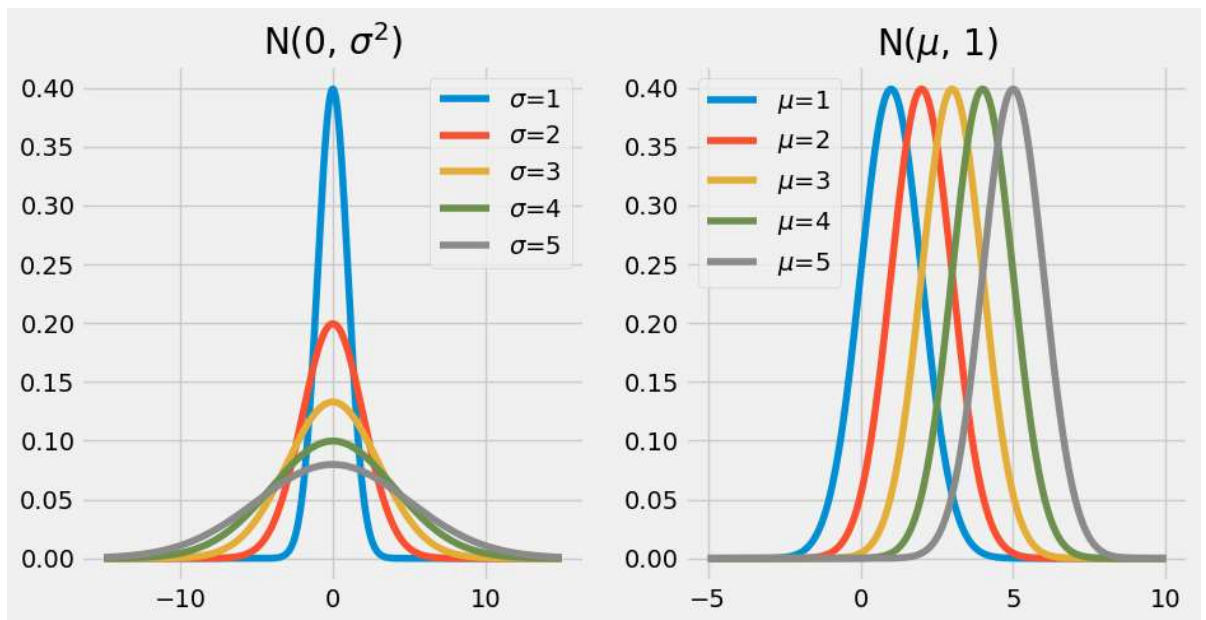
### 1. 常態分配的 PDF 圖形

工作敘述:

- 左圖為改變  $\sigma$  的值帶1~5進行計算
- 右圖則為改變  $\mu$  的數值從1~5進行計算。

```
In [9]: #picture 1
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm, chi2, t
fig, ax = plt.subplots(1,2,figsize = [10, 5])
xlim = [-15, 15]
mu = 0
s = np.arange(1, 6)
x = np.linspace(xlim[0], xlim[1], 1000)
y = norm.pdf(x.reshape(-1, 1), mu, s)
label = [" $\sigma = {}$ ".format(i) for i in s]
ax[0].plot(x, y, label=label)
ax[0].legend()
ax[0].set_title("N(0,  $\sigma^2$ )")

xlim = [-5, 10]
mu = np.arange(1, 6)
s = 1
x = np.linspace(xlim[0], xlim[1], 1000)
y = norm.pdf(x.reshape(-1, 1), mu, s)
label = [" $\mu = {}$ ".format(i) for i in mu]
ax[1].plot(x, y, label=label)
ax[1].legend()
ax[1].set_title("N( $\mu$ , 1)")
plt.style.use('fivethirtyeight')
```



討論：描述上述程式與執行結果，有甚麼值得說明或強調的。

- 改變  $\sigma$  的數值從  $\sigma=1 \sim \sigma=5$  我們可以觀察到，當  $\sigma$  值越大時，常態分佈曲線越平坦。
- 改變  $\mu$  的數值從  $\mu=1 \sim \mu=5$  我們可以觀察到，當  $\mu$  改變時，整條常態曲線會跟著左右平移。

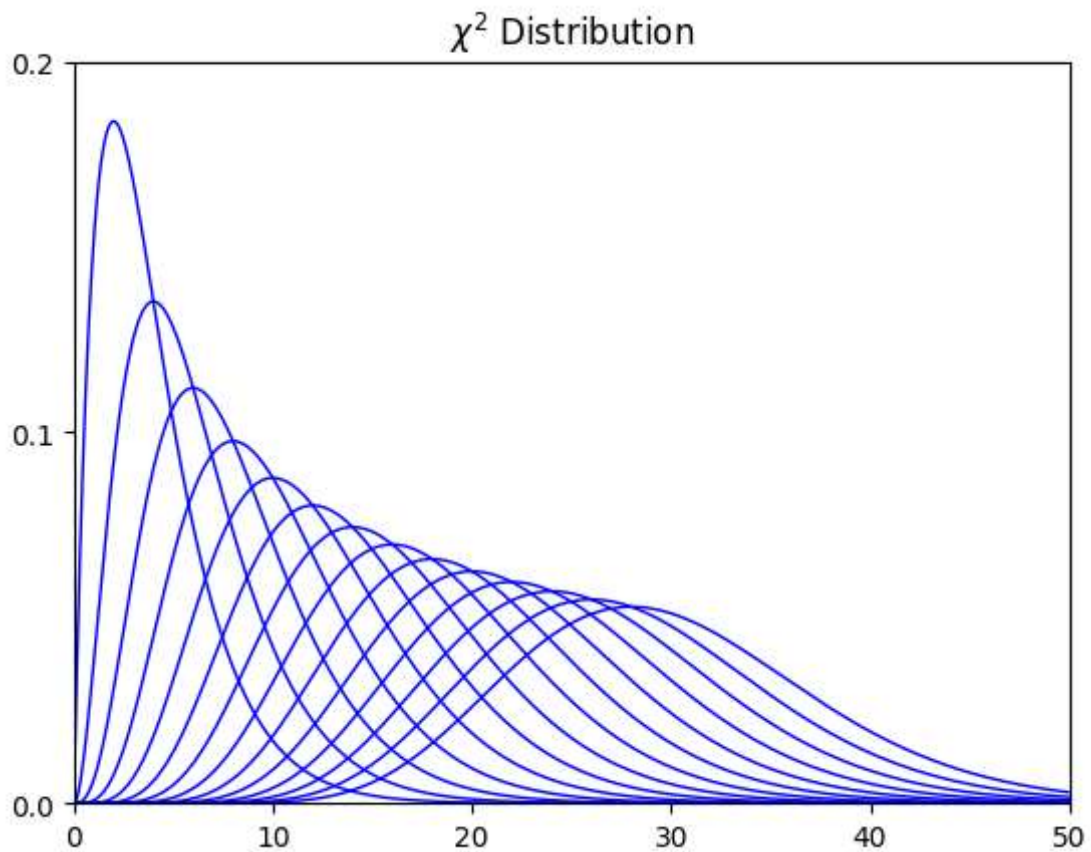
## 2. 卡方分配的 PDF 函數

工作敘述:

- 設定x的界線為0-50
- 繪製  $\chi^2$  分配的 PDF 圖，其中自由度  $\nu = 4 : 2 : 32$ , 如下圖。

```
In [2]: import numpy as np
from scipy.stats import chi2
import matplotlib.pyplot as plt

xlim = [0, 50]
x = np.linspace(xlim[0], xlim[1], 1000)
df = np.arange(4, 32, 2)
plt.figure()
plt.axis([xlim[0], xlim[1], 0, 0.2])
for i in df:
    y=chi2.pdf(x, i)
    plt.plot(x,y, lw=1, color='blue')
plt.title('$\chi^2$ Distribution')
plt.yticks([0, 0.1, 0.2])
plt.show()
```



討論：描述上述程式與執行結果，有甚麼值得說明或強調的。

- 改變自由度的數值從4, 6, 8, ... ~ 32，我們可以觀察到，當自由度值越大時，卡方分佈曲線圖越平坦。

### 3. T分配的 PDF 函數

工作敘述

- 繪製  $t$  分配的 PDF 圖，其中自由度  $\nu$  從 0.1 : 0.1 : 1 延續到 3 : 3 : 30，如下圖藍色部分。
- 畫上  $z$  分佈曲線，即為常態分佈  $\mu = 0, \sigma = 1$ ，為下圖紅線。

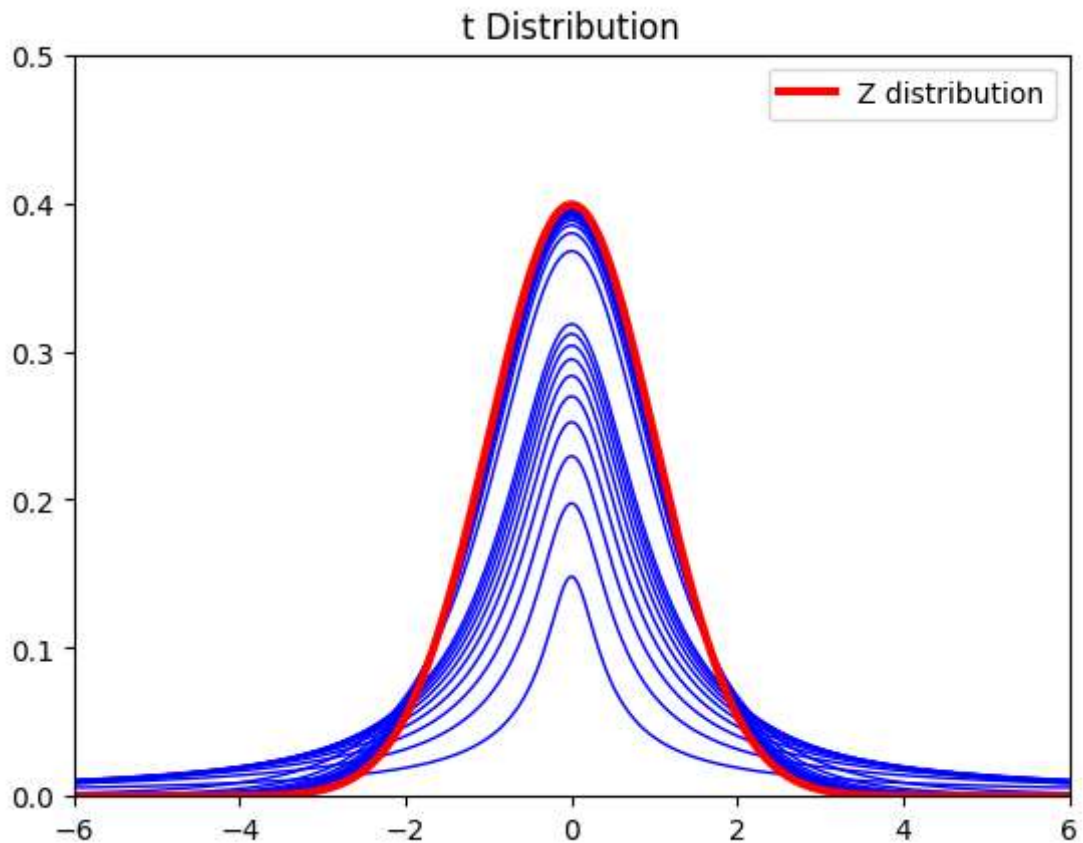
```
In [22]: import numpy as np
from scipy.stats import t
import matplotlib.pyplot as plt

xlim = [-6, 6]
x = np.linspace(xlim[0], xlim[1], 1000)

df = np.r_[np.arange(0.1, 1, 0.1), np.arange(1, 30, 2)]
plt.figure()
plt.axis([xlim[0], xlim[1], 0, 0.2])
for i in df:
    y = t.pdf(x, i)
    plt.plot(x, y, lw=1, color='blue')

Y = norm.pdf(x, 0, 1)
plt.plot(x, Y, color = "red", lw=3, label="Z distribution")
plt.title('t Distribution')
plt.yticks([0, 0.1, 0.2, 0.3, 0.4, 0.5])
```

```
plt.legend()
plt.show()
```



討論：描述上述程式與執行結果，有甚麼值得說明或強調的。

- 當t分配的自由度小於30的時候，其pdf圖形和常態分佈曲線有較顯著的差異。
- 當自由度接近30，t值和z值相當接近，就沒有再特別採用t檢定的必要。

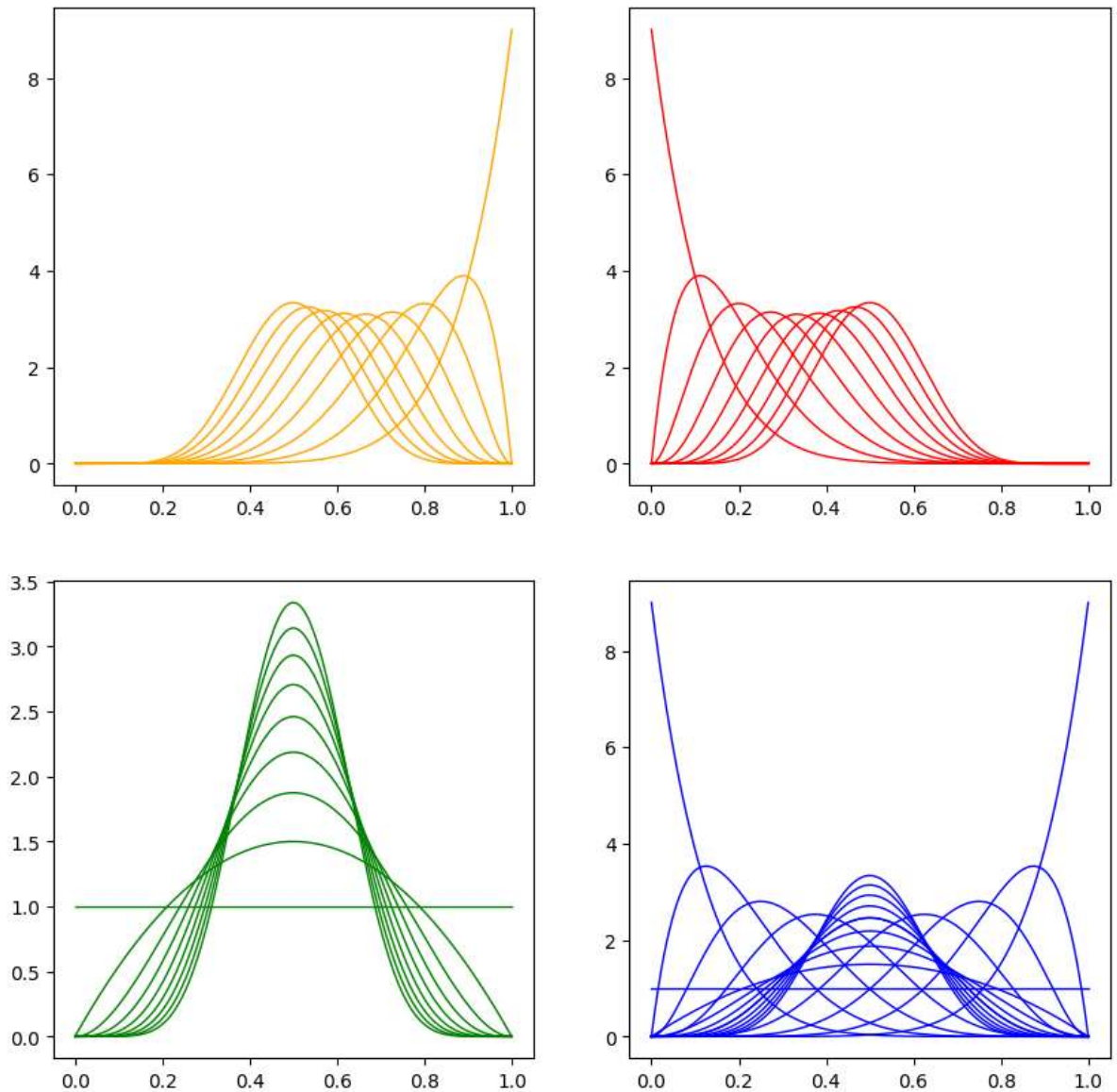
### $\beta$ 分配的PDF函數圖

工作敘述：描述做了甚麼或有甚麼值得一提的。

- 以下四張圖。
- 圖1(黃圖)固定a。
- 圖2(紅圖)固定b。
- 圖3(綠圖) a, b 皆由小到大變動。
- 圖4(藍圖) a由小到大變動,b由大到小變動，並加上圖3。

```
In [7]: from scipy.stats import beta
n = 9
x = np.linspace(0, 1, 1000)
a = np.linspace(1, 9, n)
b1 = a
b2 = np.linspace(9, 1, n)
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(nrows=2, ncols=2, figsize=(10, 10))
for i in range(n):
    y1 = beta.pdf(x, a[8], b1[i])
    y2 = beta.pdf(x, a[i], b1[8])
    y3 = beta.pdf(x, a[i], b1[i])
    y41 = beta.pdf(x, a[i], b1[i])
    y42 = beta.pdf(x, a[i], b2[i])
```

```
ax1.plot(x, y1, lw=1, color='orange')
ax2.plot(x, y2, lw=1, color='red')
ax3.plot(x, y3, lw=1, color='green')
ax4.plot(x, y41, lw=1, color='blue')
ax4.plot(x, y42, lw=1, color='blue')
```



## F 分配的PDF函數圖

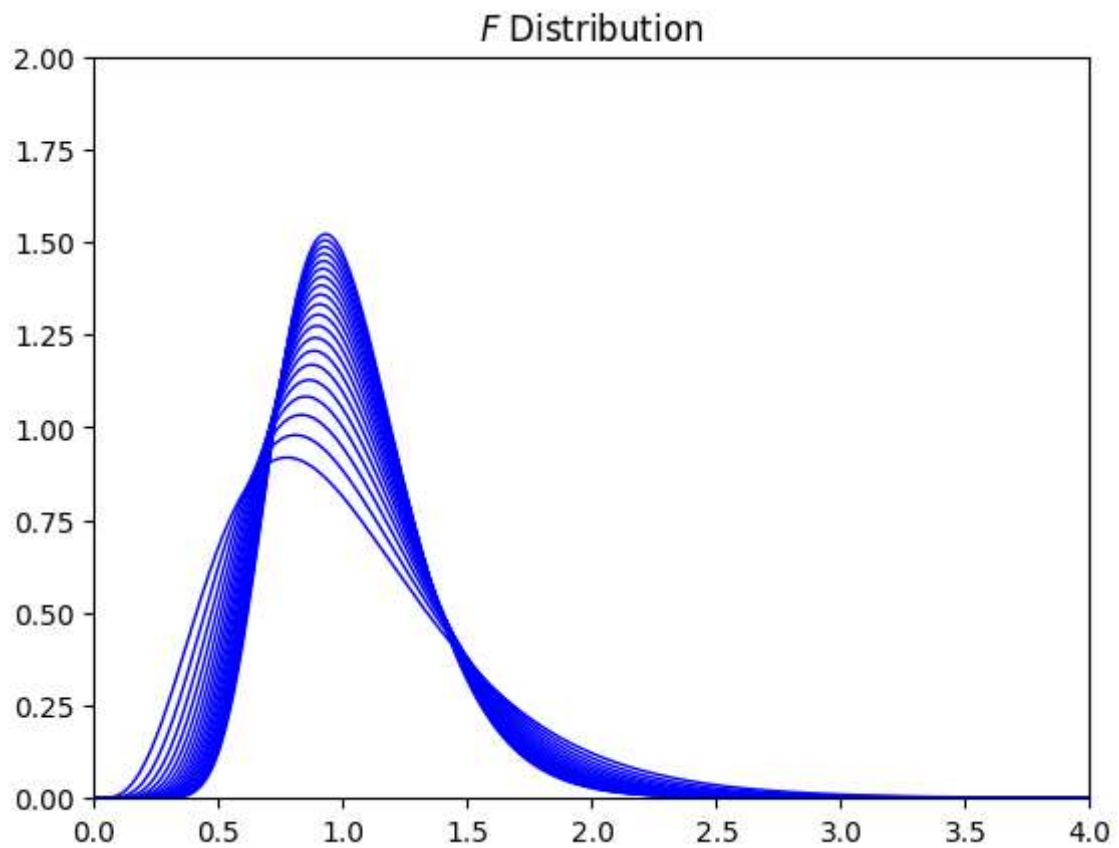
工作敘述：描述做了甚麼或有甚麼值得一提的。

- 繪製  $F(n1, 60)$  分配的 PDF 圖。(改變分子自由度)
- 繪製  $F(60, n2)$  分配的 PDF 圖。(改變分母自由度)
- $n1, n2$  數值皆為10-50，中間切20等分。

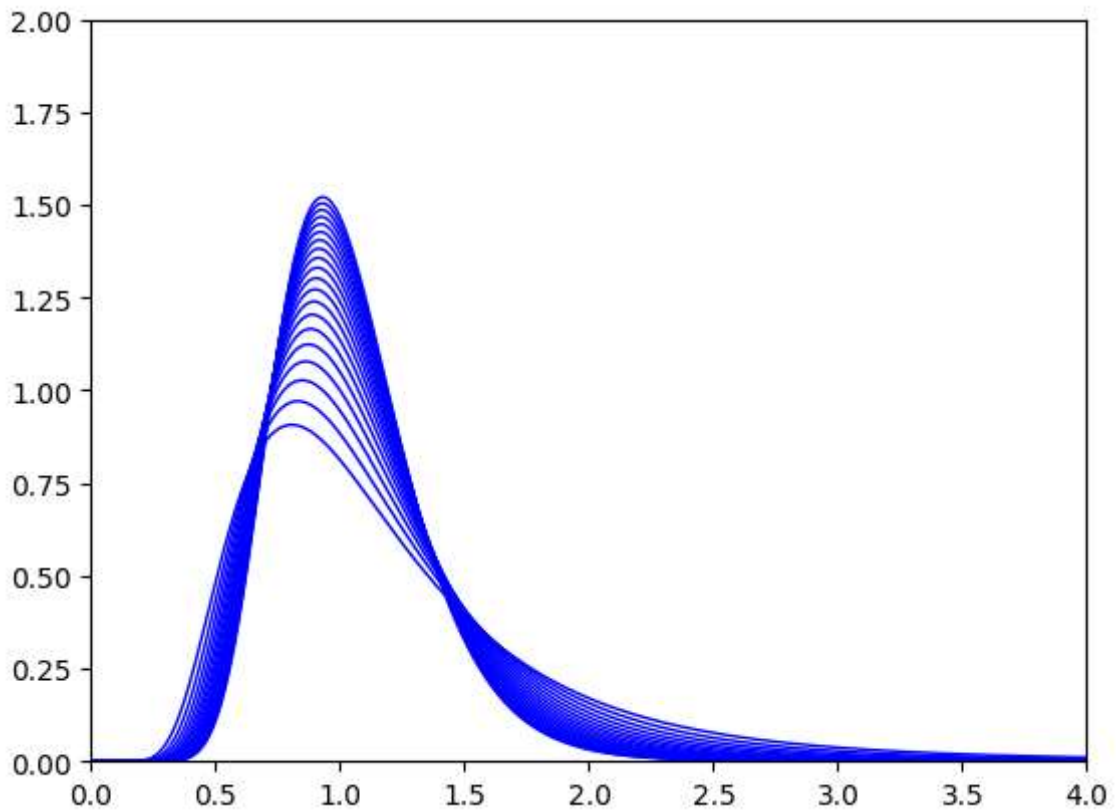
```
In [32]: import numpy as np
from scipy.stats import f
import matplotlib.pyplot as plt
fig, ax = plt.subplots(1,2,figsize = [10, 5])
xlim = [0, 4]
x = np.linspace(xlim[0], xlim[1], 1000)
n1 = np.linspace(10,50,20)
plt.figure()
plt.axis([xlim[0], xlim[1], 0, 2])
for i in n1:
    y=f.pdf(x, i, 100)
```

```
plt.plot(x,y, lw=1, color='blue')
plt.title('$F$ Distribution')

xlim = [0, 4]
x = np.linspace(xlim[0], xlim[1], 1000)
n2 = np.linspace(10,50,20)
plt.figure()
plt.axis([xlim[0], xlim[1], 0, 2])
for i in n2:
    y=f.pdf(x, 60, i)
    plt.plot(x,y, lw=1, color='blue')
plt.show()
```







討論：描述上述程式與執行結果，有甚麼值得說明或強調的。

- 改變分子自由度和改變分母自由度的兩張圖，從原本的左尾較厚右尾較薄，變成右尾較厚左尾較薄，兩張圖略有對稱的樣貌。

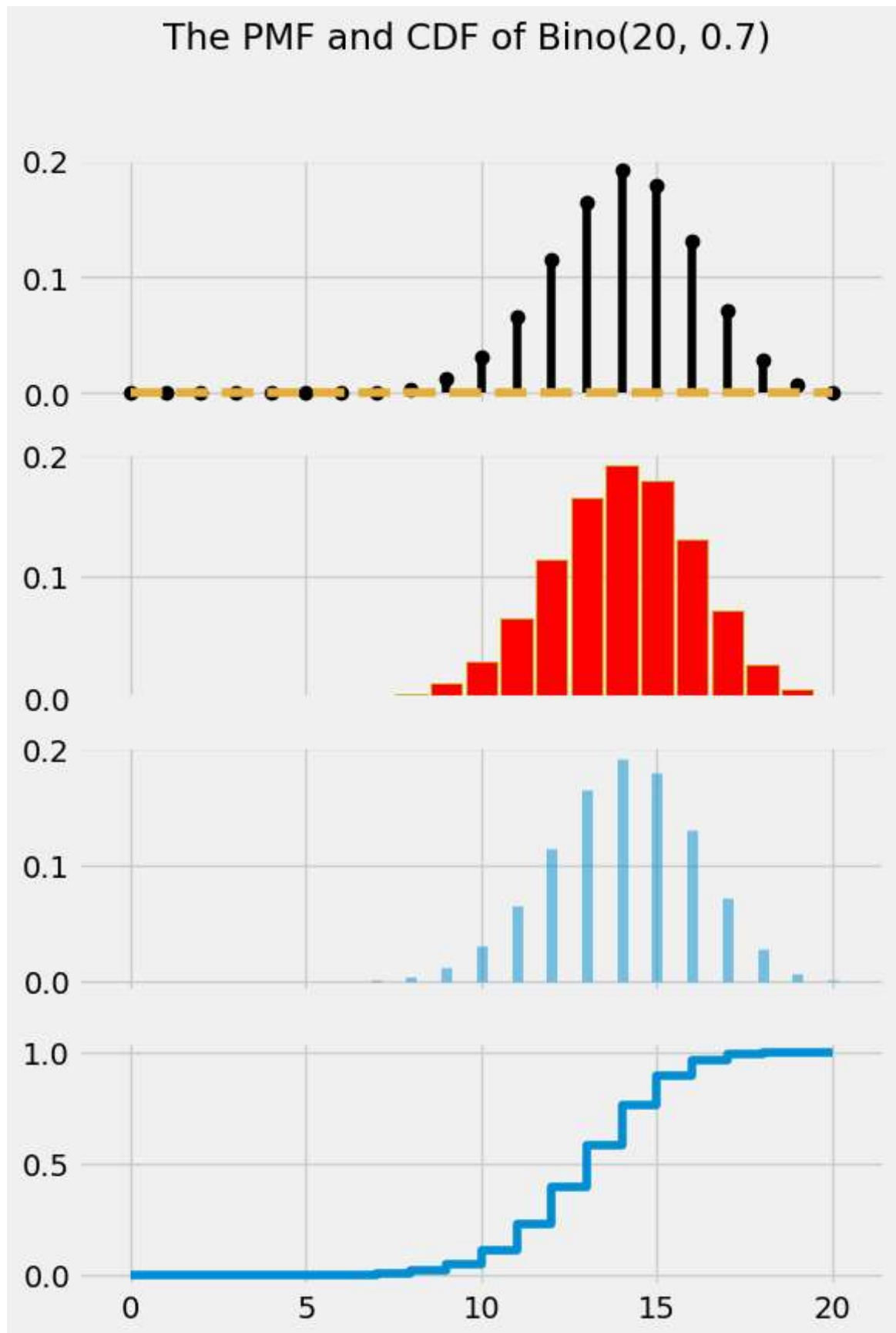
## Binomial 離散型分配

工作敘述：描述做了甚麼或有甚麼值得一提的。

- 示範三種不同的表現方式來繪製 PMF 圖。另外加上一張 CDF 圖。用了四種繪製離散分配

```
In [38]: import numpy as np
from scipy.stats import binom
import matplotlib.pyplot as plt

n, p = 20, 0.7
x = np.arange(n + 1)
y = binom.pmf(x, n, p)
fig, ax = plt.subplots(4, 1, sharex = 'col', figsize = [6, 9])
ax[0].stem(x, y, linefmt='k-', markerfmt='ko', basefmt = 'C2--')
ax[1].bar(x, y, width = 0.9, color = 'r', edgecolor = 'y' )
ax[2].vlines(x, 0, y, lw = 5, alpha = 0.5)
Y = binom.cdf(x, n, p)
ax[3].plot(x, Y, drawstyle = 'steps-pre')
plt.suptitle('The PMF and CDF of Bino({}, {})'.format(n, p))
plt.show()
```



討論：描述上述程式與執行結果，有甚麼值得說明或強調的。

- 離散型分配不可用上面五張連續型分配的方式做圖，會違反其定義。
- 觀察cdf圖可以發現圖形呈現階梯狀。