作品三、 機率分配的樣貌

姓名：黃冠翔
學號：410978040

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。

---

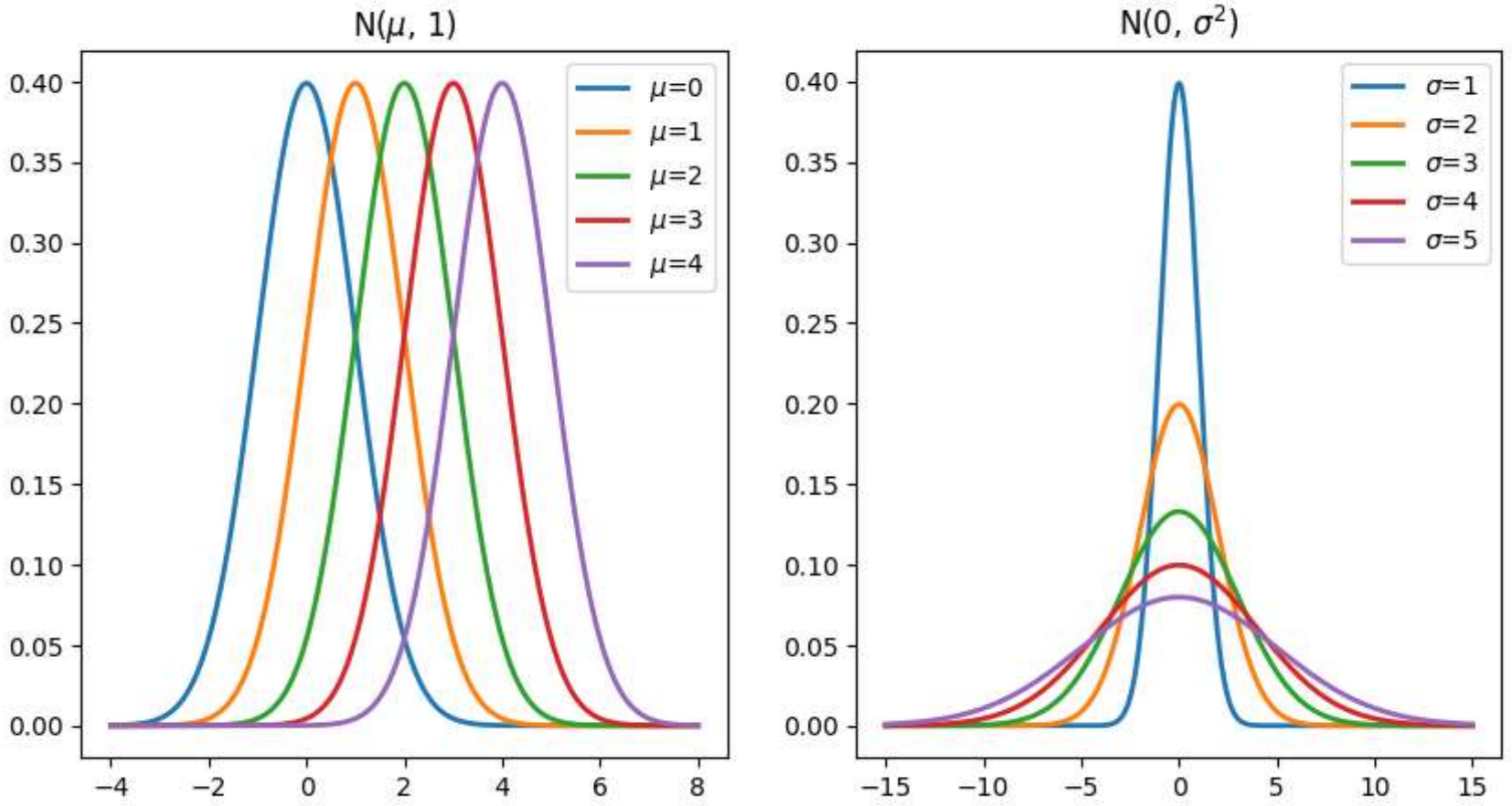
1. 常態分配的 PDF 函數

工作敘述：

- 繪製常態分配圖形進行比較。
- 左圖將變異數固定為1，並將平均值由0到4代入進行繪圖。
- 右圖將平均值固定為0，並將變異數由1到5代入進行繪圖。

In [ ]:
```
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
fig, ax = plt.subplots(1, 2, figsize=[10, 5])

#變異數固定為1
xlim = [-4, 8]
mu = np.arange(0, 5)
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[0].plot(x, y, label=label, lw=2)
```

```
ax[0].set_title("N($\mu$, 1)")
ax[0].legend()

#平均值固定為0
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[1].plot(x, y, label=label, lw=2)
ax[1].set_title("N(0, $\sigma^2$)")
ax[1].legend()
plt.show()
```



討論：

- 左圖中可以發現，當變異數固定為1時，平均值的變動會使常態分佈圖形左右平移。平均值變大時向右平移，平均值變小則向左平移。
- 右圖中可以發現，當平均值固定為0時，變異數的變動會影響常態分佈圖形的密集程度。變異數較大時，分布較分散;變異數較小時，分布較密集。

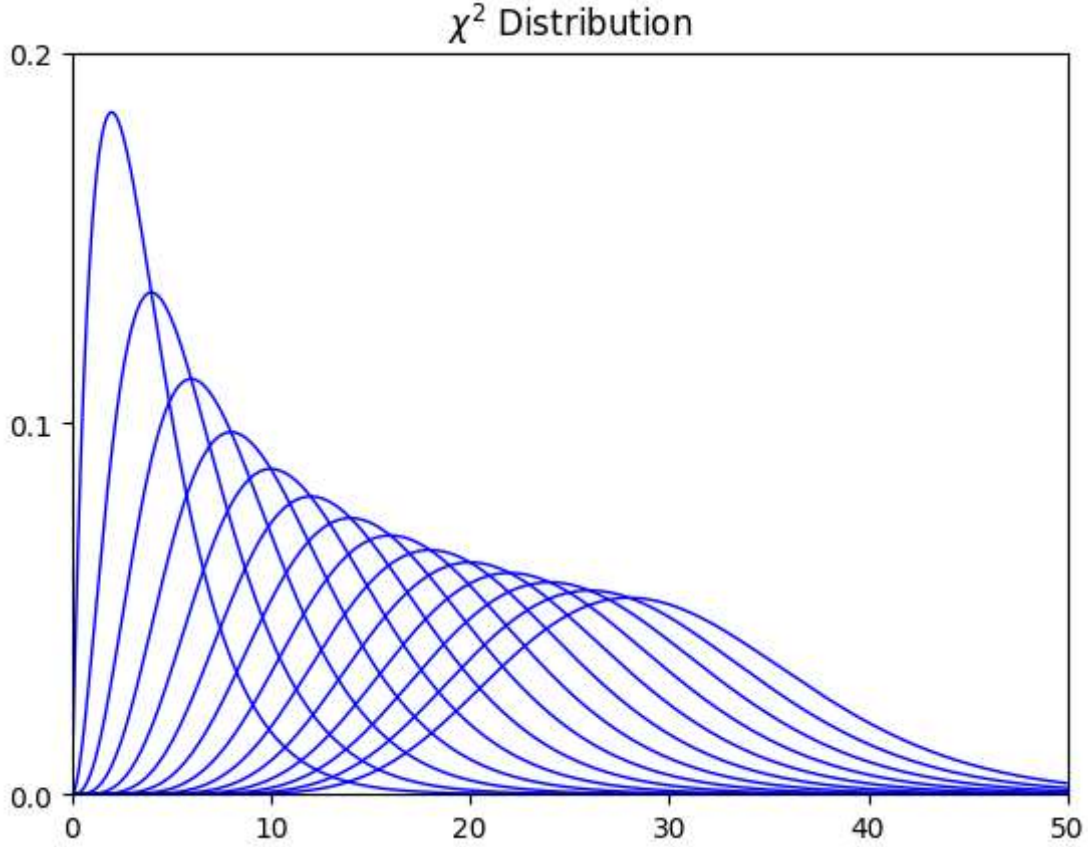
---

2. 卡方分配的 PDF 函數

工作敘述：

- 繪製卡方分配圖形。
- 設定x範圍0~50。
- 設定自由度範圍4、6、8、···、28、30、32。

In [ ]:
```
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
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(r'$\chi^2$ Distribution')
plt.yticks([0, 0.1, 0.2])
plt.show()
```



討論：

- 當自由度值愈大時，卡方分佈曲線圖愈像右移動，且愈平坦。

---

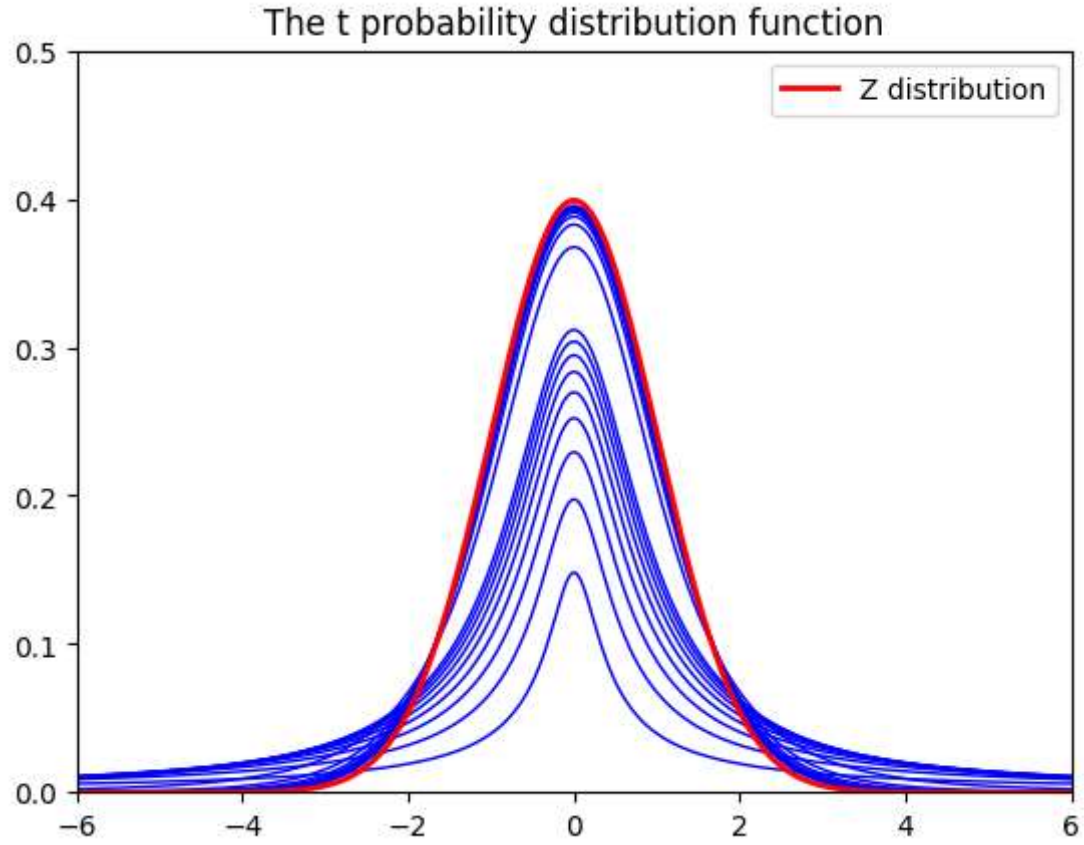
3. T分配的 PDF 函數

工作敘述：

- 繪製t分配圖形(藍線)。
- (1)設定x範圍-6~6。
- (2)設定自由度範圍0.1、0.2、···、0.9、1 並延續到 3、6、···、27、30。
- 繪製Z分配圖形，意即標準常態分配圖形，平均值為0，變異數為1(紅線)。

In [ ]:

```
import numpy as np
from scipy.stats import t
from scipy.stats import norm
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(3, 30, 3)]

plt.figure()
plt.axis([xlim[0], xlim[1], 0, 0.4])
for i in df:
    y=t.pdf(x, i)
    plt.plot(x, y, lw=1, color='blue')

mu=0
s=1
y = norm.pdf(x.reshape(-1, 1), mu, s)
plt.plot(x, y, lw=2, color='r', label="Z distribution")
plt.legend()
plt.title("The t probability distribution function")
plt.yticks([0, 0.1, 0.2, 0.3, 0.4, 0.5])

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時，t分配和常態分配有較明顯的差異。
- 當自由度接近30時，t分配和常態分配相當接近。

---

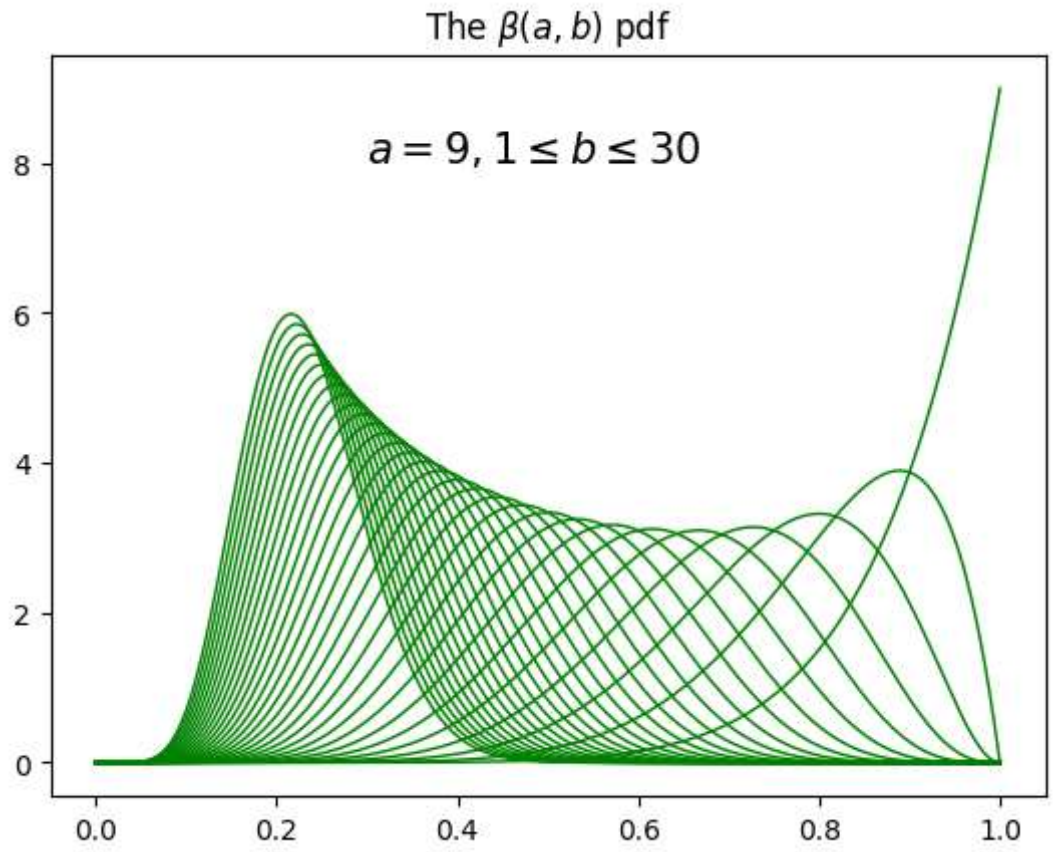
4. Beta分配的 PDF 函數

工作敘述：

- 繪製Beta分配圖形(綠線)。
- (1)設定x範圍0~1。
- (2)設定a=9。

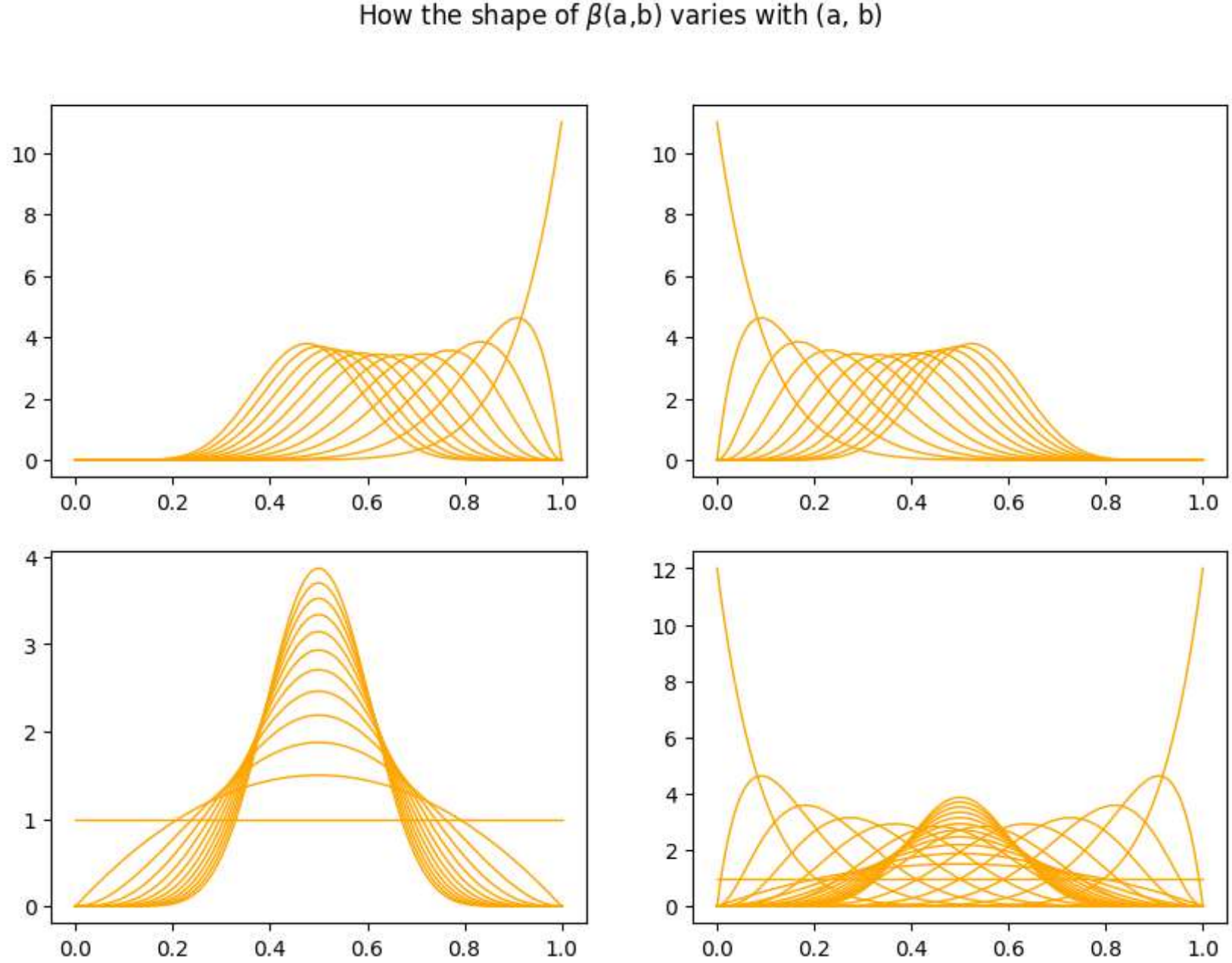
- (3)設定b範圍1~30。
- 繪製Beta分配圖形進行比較(橘線)。
- (1)設定x範圍0~1。
- (2)左上圖固定a值進行繪圖。
- (3)右上圖固定b值進行繪圖。
- (4)左下圖a值和b值皆逐漸增加進行繪圖。
- (5)右下圖a值逐漸增加，b值逐漸減少，並將左下圖疊加進行繪圖。

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

x = np.linspace(0, 1, 1000)
a = 9
b = np.linspace(1, 30, 30)
for i in range(30):
    y = beta.pdf(x, a, b[i])
    plt.plot(x, y, lw=1, color='green')

plt.title("The $\\beta(a,b)$ pdf")
plt.text(0.3, 8, '$a=9,1\\leq b \\leq 30$', fontsize=15)

plt.show()
```



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

n =11
x = np.linspace(0, 1, 1000)
a = np.linspace(1, 11, n)
b1 = a
b2 = np.linspace(11, 1, n)
fig, ((ax1, ax2), (ax3, ax4))  = plt.subplots(2, 2, figsize=(10, 7))
for i in range(n):
    y1 = beta.pdf(x, a[10], b1[i])
    y2 = beta.pdf(x, a[i], b1[10])
    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='orange')
    ax3.plot(x, y3, lw=1, color='orange')
    ax4.plot(x, y41, lw=1, color='orange')
    ax4.plot(x, y42, lw=1, color='orange')

plt.suptitle("How the shape of $\\beta$(a,b) varies with (a, b)")
plt.show()
```



討論：

- 左上圖固定a值時，圖形右偏;右上圖固定b值時，圖形左偏。

---

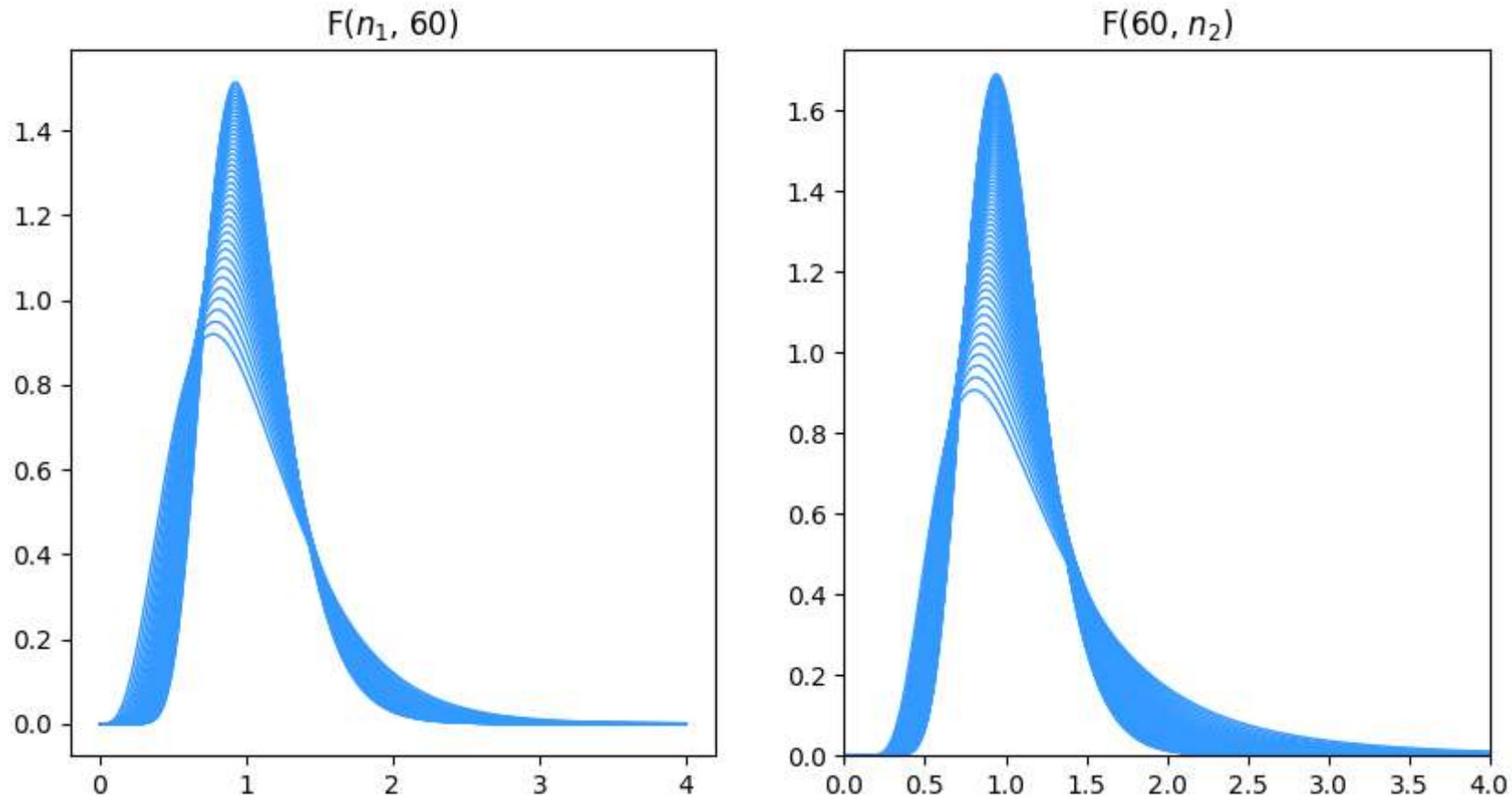
5. F分配的 PDF 函數

工作敘述：

- 繪製F分配圖形進行比較。
- 設定x範圍0~4。
- 設定n1和n2數值為10~50。
- 左圖將分母自由度固定為60，繪製 $F(n1, 60)$ 分配的圖形。(改變分子自由度)
- 右圖將分子自由度固定為60，繪製 $F(60, n2)$ 分配的圖形。(改變分母自由度)

In [ ]:
```
import numpy as np
from scipy.stats import f
import matplotlib.pyplot as plt
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=[10, 5])

#dfd = 60
xlim = [0, 4]
x = np.linspace(xlim[0], xlim[1], 1000)

dfn = np.arange(10, 50, 1)
dfd = 60

plt.axis([xlim[0], xlim[1], 0, 1.75])

for i in dfn:
    y=f.pdf(x, i, dfd)
    ax1.plot(x, y, lw=1, color='#3399FF')
ax1.set_title("F($n_1$, 60)")


#dfn = 60
xlim = [0, 4]
x = np.linspace(xlim[0], xlim[1], 1000)

dfn = 60
dfd = np.arange(10, 80, 1)
```

```
plt.axis([xlim[0], xlim[1], 0, 1.75])

for j in dfd:
    y=f.pdf(x, dfn, j)
    ax2.plot(x, y, lw=1, color='#3399FF')
ax2.set_title("F(60, $n_2$)")

plt.show()
```



討論：

- 兩張圖比較可以發現，改變分子自由度的圖形左尾較厚右尾較薄;改變分母自由度的圖形左尾較薄右尾較厚。

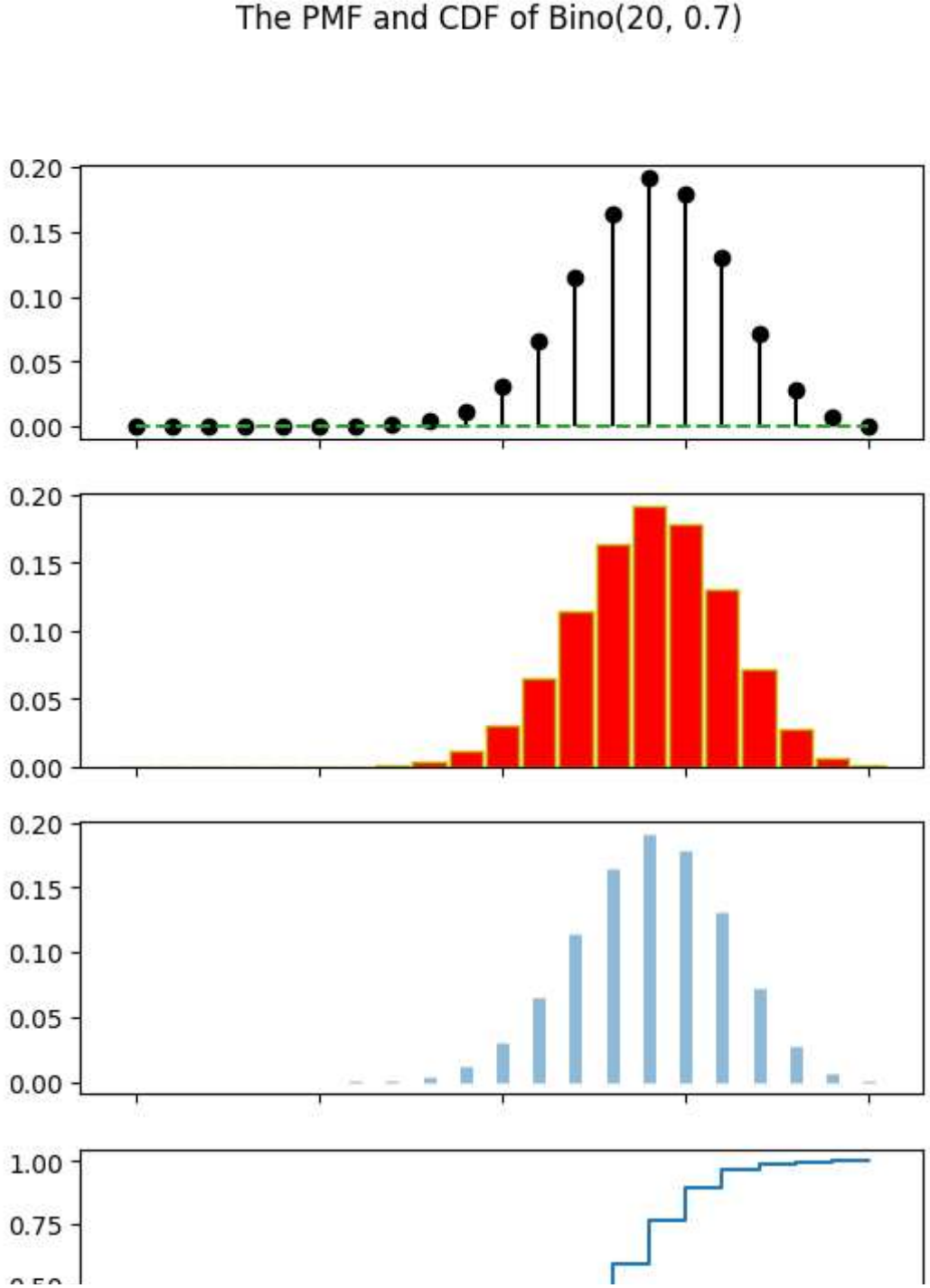
---

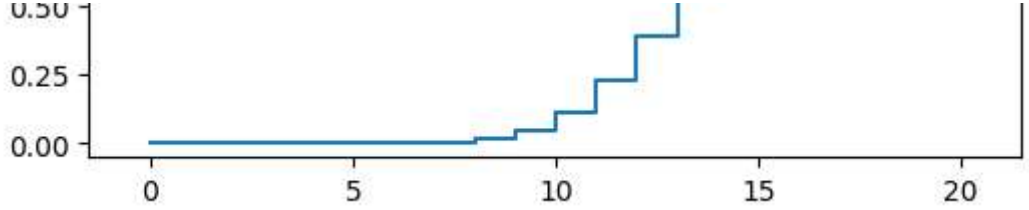
6. Binomial分配的 PMF 圖和 CDF 圖

工作敘述：

- 設定n=20和p=0.7。
- 第一張圖採用莖葉圖(stem)畫法，繪製Binomial分配的 PMF 圖。
- 第二張圖採用長條圖(bar)畫法，繪製Binomial分配的 PMF 圖。
- 第三張圖採用垂直線(vlines)畫法，繪製Binomial分配的 PMF 圖。
- 第四張圖繪製Binomial分配的 CDF 圖。

In [ ]:
```
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()
```


The PMF and CDF of Bino(20, 0.7)
0.20
0.15
0.10
0.05
0.00
0.20
0.15
0.10
0.05
0.00
0.20
0.15
0.10
0.05
0.00
1.00
0.75



討論：

- 在n=14時，p有最大值，圖形的p值不論是從n=14向左或向右均呈現遞減的情況。
- 第四章的 CDF 圖呈現階梯狀，在n到20時，總和為1。
