

第一次作品：探索機率分配與隨機樣本的樣貌

學號：411078064

姓名：謝意盛

作品目標：

1. 繪製出包含常態分佈、卡方分佈、T 分佈、Beta 分佈以及 F 分佈五種連續型分佈的 PDF 函數。通過調整各分佈的參數來觀察其 PDF 函數的「形狀」變化，展示所有可能的「形狀」並說明其與參數之間的關聯。展示方式有兩種類型，分別為將不同參數下的 PDF 函數繪製在同一張圖上，或是使用子圖 (Subplot) 的方式來呈現各個分佈在不同參數下的表現。
2. 從上述五種分佈中各選取一組參數，生成 N 個樣本，繪製出四種圖，分別是直方圖 (Histogram)、盒鬚圖 (Boxplot)、機率圖 (Probability Plot) 與經驗累積分佈函數圖 (Empirical CDF)，並將這四張圖以 2×2 子圖的方式進行展示。此外，每一小題都將選擇兩個不同的樣本數，以展示出不同大小的 N 對於圖形的影響。

第一大題：畫出常態、卡方、T、Beta、F 等五種分配所有可能的『形狀』，進行觀察並討論說明分佈模樣的形成原因

1. Normal Distribution

- 分佈：

$$X \sim N(\mu, \sigma^2), \quad -\infty < \mu < \infty, \quad \sigma > 0$$

- PDF :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty$$

說明：以下將繪畫出 *Normal Distribution* 在不同的參數設定下，即通過改變 μ 和 σ 這兩個參數，來呈現出此分佈不同的樣貌。

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

plt.style.use('fivethirtyeight')
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize = (13, 7))
plt.suptitle('Normal Distributions', fontsize = 22, fontweight='bold')

# 设置全局的 y 轴标签
fig.text(-0.02, 0.5, 'Density', va = 'center', rotation = 'vertical', fontsize = 22)
fig.text(0.5, -0.02, '$\mu$', fontsize = 22, color = 'black')

# Plot 1
xlim = [-4, 4]
mu = 0
```

```

s = 1
x = np.linspace(xlim[0], xlim[1], 1000)           # xlim[0] 為 xlim 的最小值 · xlim[1] 為 xlim 的最大值
y = norm.pdf(x, mu, s)
ax1.plot(x, y, linewidth = 2.5, color = 'blueviolet', alpha = 0.8)
ax1.set_xticks(np.arange(-4, 4.1, 2))
ax1.set_yticks(np.arange(0, 0.41, 0.1))
ax1.tick_params(axis = 'both', colors = 'black') # 设置 x, y 轴刻度线的颜色
# ax1.legend(['$\mu$ = 0, $\sigma$ = 1'], bbox_to_anchor = (1, 1), fontsize = 11)
ax1.set_title('$\mu$ = {}, $\sigma$ = {}'.format(mu, s), fontsize = 20)

# Plot 2
xlim = [-15, 15]
mu = 0
s = np.arange(1, 6, 1)                           # 出來的是一個列向量 · 為 (5, )
x = np.linspace(xlim[0], xlim[1], 1000)           # xlim[0] 為 xlim 的最小值 · xlim[1] 為 xlim 的最大值

# 定义颜色列表
colors = plt.colormaps['YlOrRd'](np.linspace(0.4, 1, len(s)))

for pr, color in zip(s, colors):
    # x.reshape(-1, 1) 會變成一個行向量 · -1 表示自动计算这个维度的大小 · 所以这里是 1000
    y = norm.pdf(x, mu, pr)
    ax2.plot(x, y, linewidth = 2.5, color = color, label = "$\sigma$ = {}".format(pr))
    ax2.set_yticks(np.arange(0, 0.41, 0.1))
    # ax2.legend(bbox_to_anchor = (1, 1), fontsize = 11) # 调整图例的位置
    ax2.tick_params(axis = 'both', colors = 'black') # 设置 x, y 轴刻度线的颜色
    ax2.set_title('$\mu$ = {}, ${} \leq \sigma \leq {}$'.format(mu, s[0], s[-1]), fontstyle = 'italic')

# Plot 3
xlim = [-5, 5]
mu = np.arange(-2, 3, 1)
s = 1
x = np.linspace(xlim[0], xlim[1], 1000)

# 定义颜色列表
colors = plt.colormaps['magma'](np.linspace(0.4, 0.8, len(mu)))

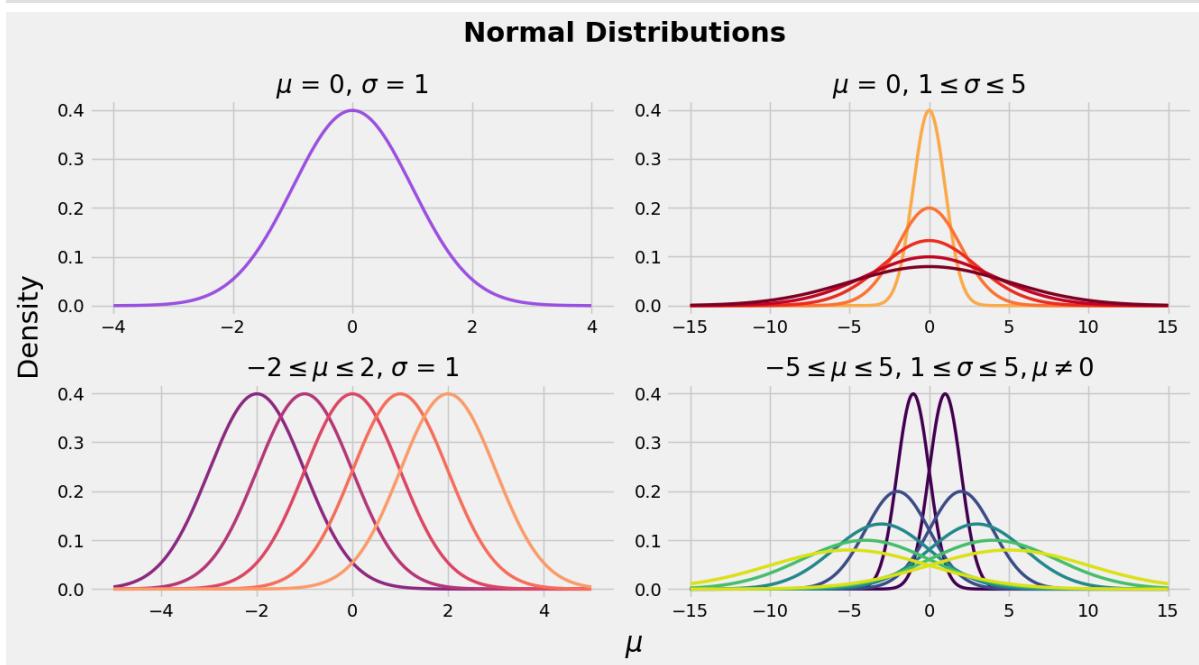
for pr, color in zip(mu, colors):
    y = norm.pdf(x, pr, s)
    ax3.plot(x, y, linewidth = 2.5, color = color, label = "$\mu$ = {}".format(pr))
    # ax3.set_xticks(np.arange(-4, 4, 2))
    ax3.set_yticks(np.arange(0, 0.41, 0.1))
    # ax3.legend(bbox_to_anchor = (1, 1), fontsize = 11) # 调整图例的位置
    ax3.tick_params(axis = 'both', colors = 'black') # 设置 x, y 轴刻度线的颜色
    ax3.set_title('${} \leq \mu \leq {}$, $\sigma$ = {}'.format(mu[0], mu[-1], s), fontstyle = 'italic')

# Plot 4
xlim = [-15, 15]
mu1 = np.arange(1, 6, 1); mu2 = np.arange(-5, 0, 1)[::-1]
s = np.arange(1, 6, 1)
x = np.linspace(xlim[0], xlim[1], 1000)
colors = plt.colormaps['viridis'](np.linspace(0, 0.95, len(s)))

for pr11, pr2, pr21, color in zip(mu1, s, mu2, colors):
    y1 = norm.pdf(x, pr11, pr2)
    y2 = norm.pdf(x, pr21, pr2)
    ax4.plot(x, y1, linewidth = 2.5, color = color, label = "$\mu$ = {}, $\sigma$ = {}".format(pr11, pr2))
    ax4.plot(x, y2, linewidth = 2.5, color = color, label = "$\mu$ = {}, $\sigma$ = {}".format(pr21, pr2))
    ax4.set_yticks(np.arange(0, 0.41, 0.1))
    # ax4.legend(loc = 'upper right', fontsize = 11, ncol = 2) # 调整图例的位置
    ax4.tick_params(axis = 'both', colors = 'black') # 设置 x, y 轴刻度线的颜色
    ax4.set_title(r'$\mu$ = {}, $\sigma$ = {}, $\mu_1 \leq \mu \leq \mu_2$, $\sigma_1 \leq \sigma \leq \sigma_2$'.format(pr11, pr2, pr21, color), fontstyle = 'italic', fontsize = 20)

```

```
plt.tight_layout()
plt.show()
```



注意事項與討論：

觀察：

- Plot 1 所呈現的是標準常態分佈，也稱 Z 分佈，其由 $\mu = 0 \cdot \sigma = 1$ 所構成，整體呈現對稱。
- Plot 2 呈現出在給定 $\mu = 0$ 的情況下，分佈隨著 σ 值由小到大的變動，會變得越來越分散。
- Plot 3 呈現出在給定 $\sigma = 0$ 的情況下，分佈隨著 μ 值由小到大的變動，會逐漸向右移動。
- Plot 4 所呈現的是 Plot 2 與 Plot 3 的合體版，同樣讓 σ 值由小到大開始變動，觀察當 μ 值 “由小到大” 和 “由大到小” 所會呈現出的模樣，可見分佈會形成一個類似蝴蝶的模樣，兩邊對稱。

結論：通過改變常態分佈裡的兩個參數 μ 與 σ ，可以觀察出分佈的不同模樣。

- μ 小往左移， μ 大向右移。
- σ 小越分散， σ 大越集中。
- 無論參數如何變動，分佈只會呈現出對稱的模樣。

2. χ^2 Distribution

- 分佈：

$$X \sim \chi^2(\nu), \quad \nu = 1, 2, \dots$$

- PDF：

$$f(x) = \frac{1}{\Gamma(\frac{\nu}{2}) 2^{\frac{\nu}{2}}} x^{\frac{\nu}{2}-1} e^{-\frac{x}{2}}, \quad 0 < x < \infty$$

說明：以下將繪畫出 χ^2 Distribution 在不同的參數設定下，即通過改變自由度 *d.f.*，來呈現出此分佈不同的樣貌。

In [2]:

```
import numpy as np
from scipy.stats import chi2
import matplotlib.pyplot as plt

plt.style.use('fivethirtyeight')
plt.figure(figsize = (8, 6))
fig = plt.gcf() # get current figure (获取当前图形)
ax = plt.gca() # get current axis (获取当前轴对象)

xlim = [0, 50]
x = np.linspace(xlim[0], xlim[1], 1000)
dfs = np.arange(4, 32, 1) # 自由度

# 定义颜色列表
colors = plt.colormaps['Blues'](np.linspace(0.5, 1, len(dfs)))

# fix xlim before animation
plt.axis([xlim[0], xlim[1], 0, 0.21])

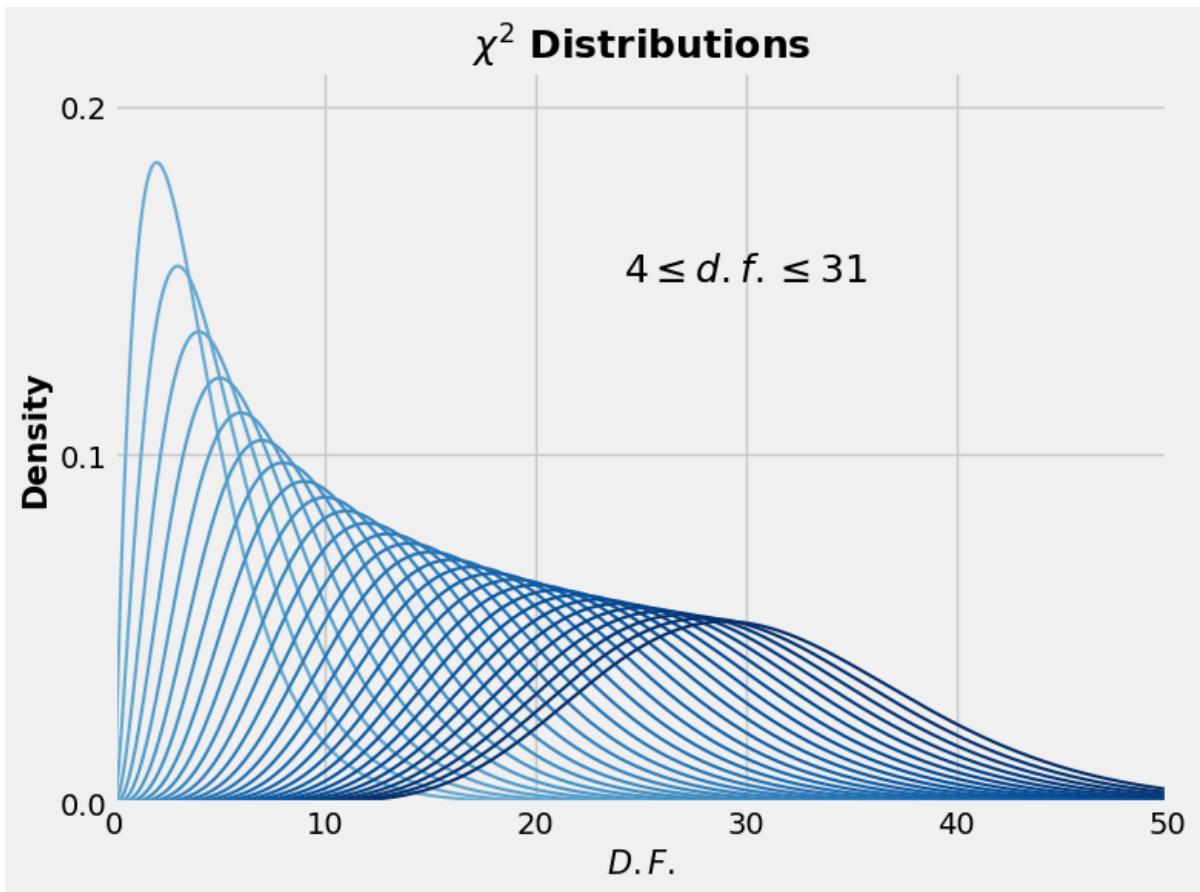
for df, color in zip(dfs, colors):
    y = chi2.pdf(x, df)
    line, = plt.plot(x, y, lw = 1.5, color = color, label = r'$\nu$ = {}'.format(df))
    plt.xlabel('$D.F.$', fontsize = 15, color = 'black', fontweight = 'bold')
    plt.ylabel('Density', fontsize = 15, color = 'black', fontweight = 'bold')

    # 添加新的图例
    ax.legend([line], [line.get_label()], loc = 'upper right') # 调整图例的位置

    plt.tick_params(axis = 'both', colors = 'black') # 设置 x, y 轴刻度线的颜色
    plt.title('$\chi^2$ Distributions', fontsize = 18, fontweight='bold')
    plt.yticks([0, 0.1, 0.2])
    plt.tight_layout()
    # plt.pause(0.5)

    # 清除之前的图例
    if ax.get_legend() is not None:
        ax.get_legend().remove()

plt.text(30, 0.15, r'$\nu \leq d.f. \leq 30$', fontsize = 18,
        ha = 'center')
plt.show()
```



注意事項與討論：

觀察：

- 上圖繪畫出自由度由 4 至 31 間隔 1 所變動的卡方分佈圖。
- 隨著自由度逐漸增加，分佈會逐漸往右移，且會越來越分散，整體形成右偏的情況。
- 以上程式碼若以 .py 形態並搭配 plt.pause(0.5) 執行，可以達成動畫的效果，可更直觀感受分佈在不同參數設定下的變化。

結論：

- 卡方分佈的自由度只能接受正整數。
- 通過改變卡方分佈的參數，即自由度，可以觀察出分佈趨勢逐漸右偏，且變得分散。

3. T Distribution

- 分佈：

$$X \sim T(\nu), \quad \nu > 0$$

- PDF：

$$f(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})} (1 + \frac{x^2}{\nu})^{-\frac{\nu+1}{2}}, \quad -\infty < x < \infty$$

說明：以下將繪畫出 *T Distribution* 在不同的參數設定下，即通過改變自由度 *d. f.*，來呈現出此分佈不同的樣貌。

In [3]:

```
import numpy as np
from scipy.stats import t, norm
import matplotlib.pyplot as plt

plt.style.use('fivethirtyeight')
fig, ax = plt.subplots(figsize = (8, 6))

xlim = [-6, 6]
x1 = np.linspace(xlim[0], xlim[1], 1000)
df1 = np.arange(0.1, 1.1, 0.1)
df2 = np.arange(3, 31, 3) # t 分佈自由度 >= 30 時 · t 分佈近似於標準正態分佈
df3 = np.arange(33, 61, 3) # 觀察自由度大於 30 後的樣子
dfs = np.concatenate((df1, df2, df3))
# dfs = np.append(df1, df2)

# 定义颜色列表
colors = plt.colormaps['YlGnBu'](np.linspace(0.3, 1, len(dfs)))

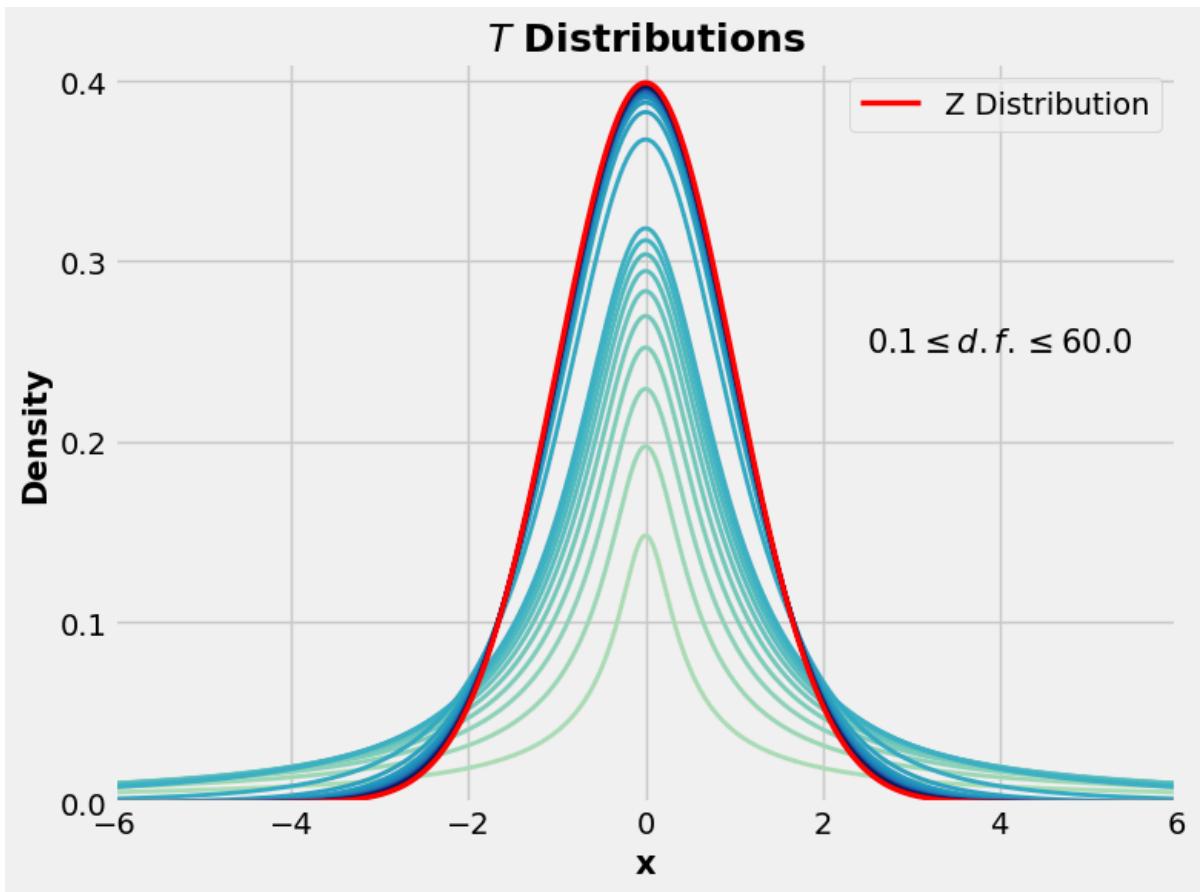
# fix xlim before animation
ax.axis([xlim[0], xlim[1], 0, 0.4])

for df, color in zip(dfs, colors):
    y1 = t.pdf(x1, df)
    line, = ax.plot(x1, y1, linewidth = 2, color = color, label = r'$\nu$ = {}'.format(df))
    ax.legend([line], [line.get_label()], loc = 'upper right')
    ax.tick_params(axis = 'both', colors = 'black')
    ax.set_title('T Distributions', fontsize = 18, fontweight = 'bold')
    ax.set_xlabel('x', fontsize = 15, color = 'black', fontweight = 'bold')
    ax.set_ylabel('Density', fontsize = 15, color = 'black', fontweight = 'bold')
    ax.set_ylim([0, 0.41])
    ax.set_yticks(np.arange(0, 0.41, 0.1))
    plt.tight_layout()
    # plt.pause(0.5)

    # 清除之前的图例
    if ax.get_legend() is not None:
        ax.get_legend().remove()

x2 = np.linspace(xlim[0], xlim[1], 1000)
y2 = norm.pdf(x2, loc = 0, scale = 1)
line2, = ax.plot(x2, y2, color = 'red', linewidth = 2.5, label = 'Z Distribution')
ax.legend([line2], [line2.get_label()], loc = 'upper right')

plt.text(4, 0.25, r'${} \leq d.f. \leq {}$'.format(dfs[0], dfs[-1]), fontsize = 15,
         ha = 'center')
plt.show()
```



注意事項與討論：

觀察：

- 上圖分別繪畫出自由度由 0.1 至 1 間隔 0.1 、3 至 30 間隔 3 、33 至 60 間隔 3 所變動的 T 分佈圖，以及服從標準常態分佈 $N(0, 1)$ 的 Z 分佈圖。
- 隨著自由度逐漸增加，可見 T 分佈呈現出集中的趨勢，且當自由度 ≥ 30 時，T 分佈將會無限近似於 Z 分佈，且之後無論自由度多大，它都不會超出 Z 分佈的涵蓋範圍。
- 以上程式碼若以 .py 形態並搭配 plt.pause(0.5) 執行，可以達成動畫的效果，可更直觀感受分佈在不同參數設定下的變化。

結論：

- T 分佈的自由度與卡方分佈不同，可以接受小數。
- 通過改變 T 分佈的參數，即自由度，可以觀察出分佈趨勢逐漸變得集中，且最終近似於標準常態分佈。

4. β Distribution

- 分佈：

$$X \sim Beta(\alpha, \beta), \quad \alpha, \beta > 0$$

- PDF :

$$f(x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}, \quad 0 < x < 1$$

說明：以下將繪畫出 β Distribution 在不同的參數設定下，即通過改變 α 和 β 這兩個參數，來呈現出此分佈不同的樣貌。

第一大圖：

```
In [4]: import numpy as np
from scipy.stats import beta
import matplotlib.pyplot as plt

plt.style.use('fivethirtyeight')
fig, ax = plt.subplots(figsize = (8, 6))

xlim = [0, 1]
x = np.linspace(xlim[0], xlim[1], 1000)
a = 9
b = np.arange(1, 31, 1)

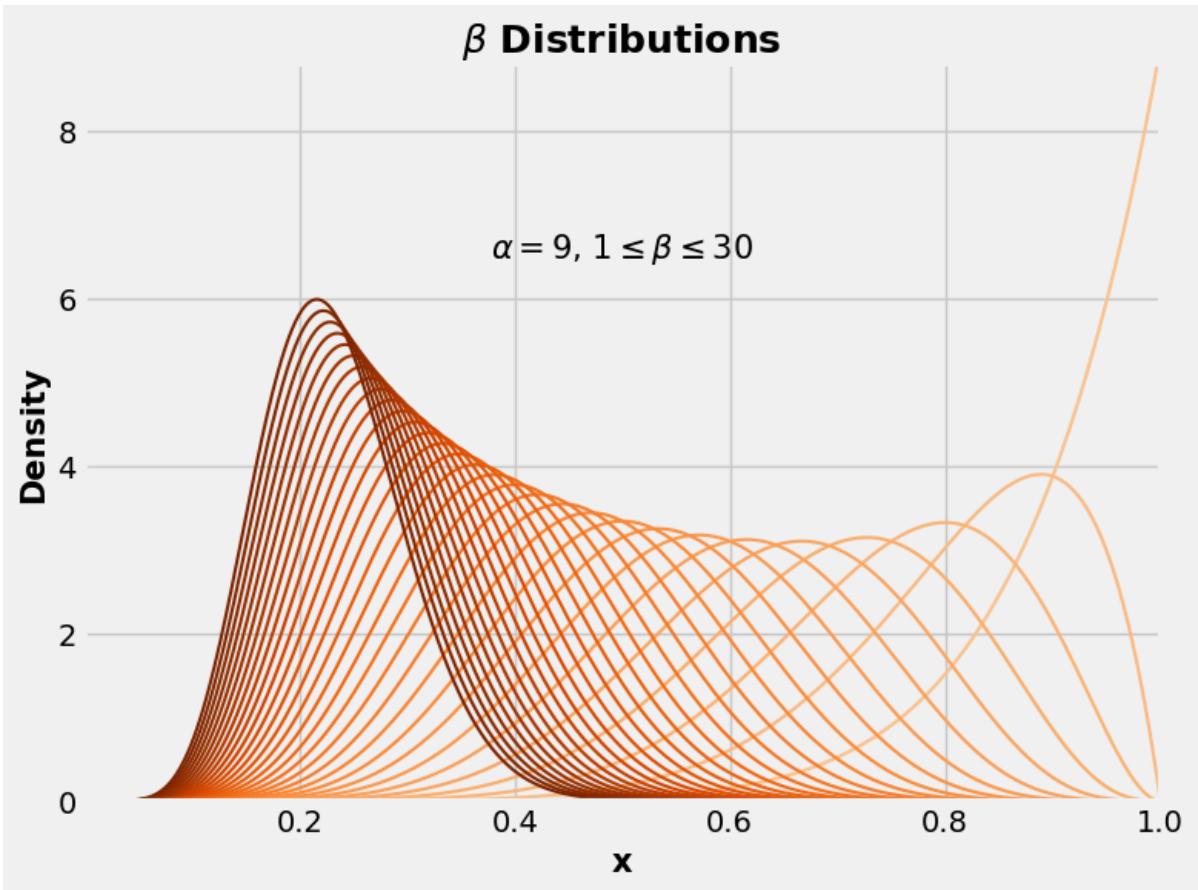
# 定义颜色列表
colors = plt.colormaps['Oranges'](np.linspace(0.3, 1, len(b)))

# fix xlim before animation
ax.axis([xlim[0], xlim[1], 0, 8.8])

for b1, color in zip(b, colors):
    y = beta.pdf(x, a = a, b = b1)
    line, = ax.plot(x, y, linewidth = 1.5, color = color, label = r'$\alpha$ = {}',
                    ax.legend([line], [line.get_label()], loc = 'upper right')
    ax.tick_params(axis = 'both', colors = 'black')
    ax.set_title(r'$\beta$ Distributions', fontsize = 18, fontweight = 'bold')
    ax.set_xlabel('x', fontsize = 15, color = 'black', fontweight = 'bold')
    ax.set_ylabel('Density', fontsize = 15, color = 'black', fontweight = 'bold')
    ax.set_xlim([0, 8.8])
    ax.set_xticks(np.arange(0.2, 1.2, 0.2))
    ax.set_yticks(np.arange(0, 9, 2))
    plt.tight_layout()
    # plt.pause(0.5)

    # 清除之前的图例
    if ax.get_legend() is not None:
        ax.get_legend().remove()

plt.text(0.5, 6.5, r'$\alpha = {}$, ${} \leq \beta \leq {}$'.format(a, b[0], b[-1]),
         ha = 'center')
plt.tight_layout()
plt.show()
```



第二大圖：

```
In [5]: import numpy as np
from scipy.stats import beta
import matplotlib.pyplot as plt

plt.style.use('fivethirtyeight')
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize = (13, 8))
xlim = [0, 1]
x = np.linspace(xlim[0], xlim[1], 1000)

# plot 1 a 固定 · b 變動 · b 越大 · 分佈越集中 · 且左偏
a = 12
b = np.arange(1, 21, 1)
colors = plt.colormaps['rainbow'](np.linspace(0.1, 1, len(b))) # 定义颜色列表
for pr, color in zip(b, colors):
    y = beta.pdf(x.reshape(-1, 1), a = a, b = pr)
    ax1.plot(x, y, linewidth = 1.5, color = color)
    ax1.set_xlim([0, 12.3])
    # ax1.text(0.3, 8.5, r'$\alpha = 12, \ 1 \leq \beta \leq 20$', fontsize = 15, color = 'black')
    ax1.set_title(r'$\alpha = {} , \ {} \leq \beta \leq {}$'.format(a, b[0], b[-1]), color = 'black')

# plot 2 a 變動 · b 固定 · a 越大 · 分佈越集中 · 且右偏
a = np.arange(1, 21, 1)
b = 12
colors = plt.colormaps['jet'](np.linspace(0.1, 1, len(a))) # 定义颜色列表
for pr, color in zip(a, colors):
    y = beta.pdf(x.reshape(-1, 1), a = pr, b = b)
    ax2.plot(x, y, linewidth = 1.5, color = color)
    ax2.set_xlim([0, 12.3])
    # ax2.text(0.3, 8.5, r'$1 \leq \alpha \leq 20, \ \beta = 12$', fontsize = 15, color = 'black')
    ax2.set_title(r'$\{} \leq \alpha \leq \{}, \ \beta = \{}$'.format(a[0], a[-1], b), color = 'black')

# plot 3 a 和 b 一樣 · 分佈對稱
a = np.linspace(1, 10, 20)
```

```

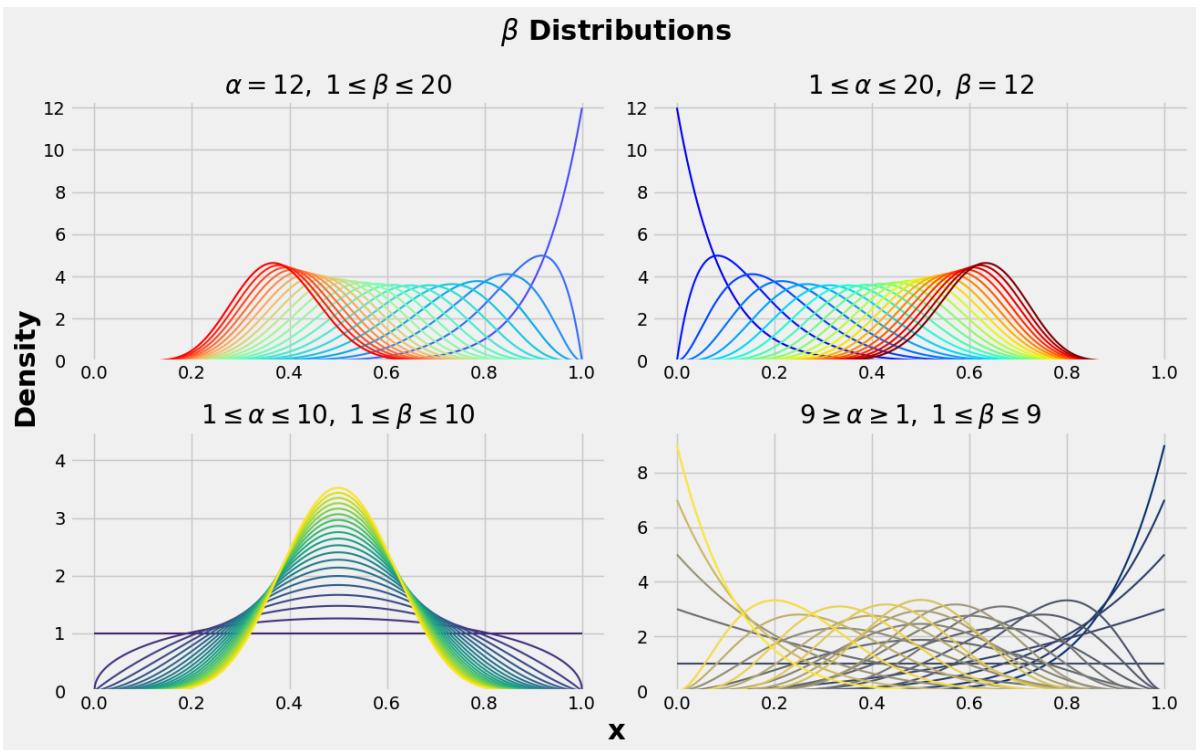
b = np.linspace(1, 10, 20)
colors = plt.colormaps['viridis'](np.linspace(0.1, 1, len(a))) # 定义颜色列表
for pr1, pr2, color in zip(a, b, colors):
    y = beta.pdf(x.reshape(-1, 1), a = pr1, b = pr2)
    ax3.plot(x, y, linewidth = 1.5, color = color)
    ax3.set_ylim([0, 4.5])
    ax3.set_yticks(np.arange(0, 4.5, 1))
    # ax3.text(0.3, 4, r'$1 \leq \alpha \leq 10, 1 \leq \beta \leq 10$', fontsize = 12)
    ax3.set_title(r'$1 \leq \alpha \leq 10, 1 \leq \beta \leq 10$', fontsize = 20)

# plot 4 a 和 b 順序相反 · 分佈包含左偏、右偏、對稱
a = np.tile(np.linspace(9, 1, 5), 5) # 重複 [9, 7, 5, 3, 1] 5 次
b = np.arange(1, 10, 2).repeat(5) # [1, 3, 5, 7, 9] 裡面每個數字重複 5 次
colors = plt.colormaps['cividis'](np.linspace(0.1, 1, len(a))) # 定义颜色列表
for pr1, pr2, color in zip(a, b, colors):
    y = beta.pdf(x.reshape(-1, 1), a = pr1, b = pr2)
    ax4.plot(x, y, linewidth = 1.5, color = color)
    ax4.set_ylim([0, 9.5])
    # ax4.text(0.3, 8, r'$1 \leq \alpha \leq 9, 1 \leq \beta \leq 9$', fontsize = 12)
    ax4.set_title(r'$9 \geq \alpha \geq 1, 1 \leq \beta \leq 9$', fontsize = 20)

for ax in [ax1, ax2, ax3, ax4]:
    ax.tick_params(axis = 'both', colors = 'black')

fig.text(0.5, -0.01, 'x', ha = 'center', fontsize = 22, color = 'black', fontweight = 'bold')
fig.text(-0.01, 0.5, 'Density', va = 'center', rotation = 'vertical', fontsize = 22, color = 'black', fontweight = 'bold')
plt.suptitle(r'$\beta$ Distributions', fontsize = 22, fontweight = 'bold')
plt.tight_layout()
plt.show()

```



注意事項與討論：

觀察：

- 第一大圖：
 - 圖中繪畫出在給定 $\alpha = 9$ 的情況下 · 分佈隨著 β 值由小到大的變動。

- 當 β 值變大時，可見分佈會逐漸往左移，當 β 與 α 值接近時，可見分佈會逐漸變得不偏，而當 β 比 α 值大時，可見分佈會逐漸變得右偏，且隨著 β 值的增加，分佈會逐漸集中。
- 以上程式碼若以 .py 形態並搭配 plt.pause(0.5) 執行，可以達成動畫的效果，可更直觀感受分佈在不同參數設定下的變化。
- 第二大圖：
- Plot 1 呈現出的模樣與第一大圖整體類似，只是參數值的設定不同，分佈情況也於之相同。
- Plot 2 呈現出在給定 $\beta = 12$ 的情況下，分佈會隨著 α 值由小到大的變動逐漸從左到右移動，當 α 與 β 值接近時，可見與 Plot 1 相同，分佈會逐漸變得不偏，而當 α 比 β 值大時，可見分佈會逐漸變得左偏，且隨著 α 值的增加，分佈也會逐漸集中。
- 從前兩張圖可發現 β 分佈在兩個參數值相同時，會出現不偏的情況，因此 Plot 3 呈現出在給定 α 值與 β 值都相同的情況下，分佈隨著兩個參數值由小到大一起變動，分佈會變得不偏，且隨著參數值的增加，逐漸變得集中。另外，可以發現當兩個參數值皆為 1 時，分佈會呈現成一條直線，這是因為當兩個參數同時為 1 時， $f(x) = 1$ ，故它會呈現出均勻分佈 Uniform(0, 1) 的模樣，故均勻分佈也為 β 分佈的其中一個特例。
- Plot 4 為前三張圖的合體版，參數設定同時包含 $\alpha > \beta$ 、 $\alpha < \beta$ 以及 $\alpha = \beta$ 三種情況，故可呈現出包含左偏、右偏以及對稱的 β 分佈。

結論：通過改變 β 分佈裡的兩個參數 α 與 β ，可以觀察出分佈的不同模樣。

- $\alpha > \beta$ 時，分佈趨勢將呈現左偏
- $\alpha = \beta$ 時，分佈趨勢將呈現不偏
- $\alpha < \beta$ 時，分佈趨勢將呈現右偏
- 參數值越高，分佈趨勢將會越集中

5. F Distribution

- 分佈：

$$X \sim F(n_1, n_2), \quad n_1, n_2 > 0$$

- PDF：

$$f(x) = \frac{\Gamma(\frac{n_1+n_2}{2})}{\Gamma(\frac{n_1}{2}) + \Gamma(\frac{n_2}{2})} \left(\frac{n_1}{n_2}\right)^{\frac{n_1}{2}} x^{\frac{n_1}{2}-1} (1 + \frac{n_1}{n_2}x)^{-\frac{n_1+n_2}{2}}, \quad x > 0$$

說明：以下將繪畫出 *F Distribution* 在不同的參數設定下，即通過改變 n_1 和 n_2 這兩個參數，來呈現出此分佈不同的樣貌。

第一大圖：

```
In [6]: import numpy as np
       from scipy.stats import f
```

```

import matplotlib.pyplot as plt

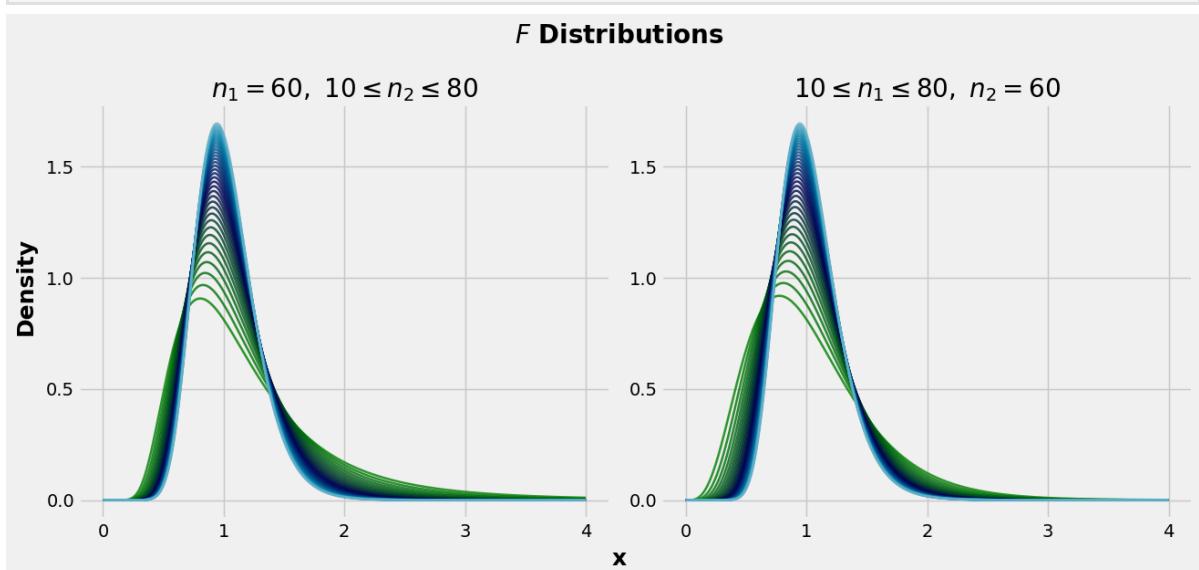
plt.style.use('fivethirtyeight')
fig, (ax1, ax2) = plt.subplots(1, 2, figsize = (13, 6))
xlim = [0, 4]
x = np.linspace(xlim[0], xlim[1], 1000)

# Plot 1 n1、n2 其中一個值固定，另一個值越大，越集中，且右偏
n1 = 60
n2 = np.arange(10, 81, 2)
colors = plt.colormaps['ocean'](np.linspace(0, 0.8, len(n2))) # 定义颜色列表
for pr, color in zip(n2, colors):
    y = f.pdf(x.reshape(-1, 1), dfn = n1, dfd = pr)
    ax1.plot(x, y, lw = 1.8, color = color, alpha = 0.8)
    ax1.tick_params(axis = 'both', colors = 'black')
    ax1.set_xticks(np.arange(0, 4.1, 1))
    ax1.set_yticks(np.arange(0, 1.76, 0.5))
    # ax1.text(1.25, 1.7, '$n_1 = 60, 10 \leq n_2 \leq 80$', fontsize = 15, color = 'black')
    ax1.set_title(r'$n_1 = 60, 10 \leq n_2 \leq 80$', fontsize = 20, color = 'black')

# Plot 2 n1、n2 互換差別不大
n1 = np.arange(10, 81, 2)
n2 = 60
colors = plt.colormaps['ocean'](np.linspace(0, 0.8, len(n1))) # 定义颜色列表
for pr, color in zip(n1, colors):
    y = f.pdf(x.reshape(-1, 1), dfn = pr, dfd = n2)
    ax2.plot(x, y, lw = 1.8, color = color, alpha = 0.8)
    ax2.tick_params(axis = 'both', colors = 'black')
    ax2.set_xticks(np.arange(0, 4.1, 1))
    ax2.set_yticks(np.arange(0, 1.76, 0.5))
    # ax2.text(1.25, 1.7, '$10 \leq n_1 \leq 80, n_2 = 60$', fontsize = 15, color = 'black')
    ax2.set_title(r'$10 \leq n_1 \leq 80, n_2 = 60$', fontsize = 20, color = 'black')

fig.text(0.5, -0.01, 'x', ha = 'center', fontsize = 18, color = 'black', fontweight = 'bold')
fig.text(-0.01, 0.5, 'Density', va = 'center', rotation = 'vertical', fontsize = 18, color = 'black', fontweight = 'bold')
plt.suptitle(r'$F$ Distributions', fontsize = 20, fontweight = 'bold')
plt.tight_layout()
plt.show()

```



第二大圖：

```
In [7]: import numpy as np
from scipy.stats import f
import matplotlib.pyplot as plt
```

```

plt.style.use('fivethirtyeight')
fig, ax = plt.subplots(figsize = (8, 6))
xlim = [0, 2]
x = np.linspace(xlim[0], xlim[1], 1000)

# n1、n2 互換差別不大，其中一個值固定，另一個值變得超大，越集中，且越來越接近不偏
n1 = 750
n2 = np.arange(150, 1501, 50)
colors = plt.colormaps['twilight'](np.linspace(0.3, 0.9, len(n2))) # 定义颜色列表

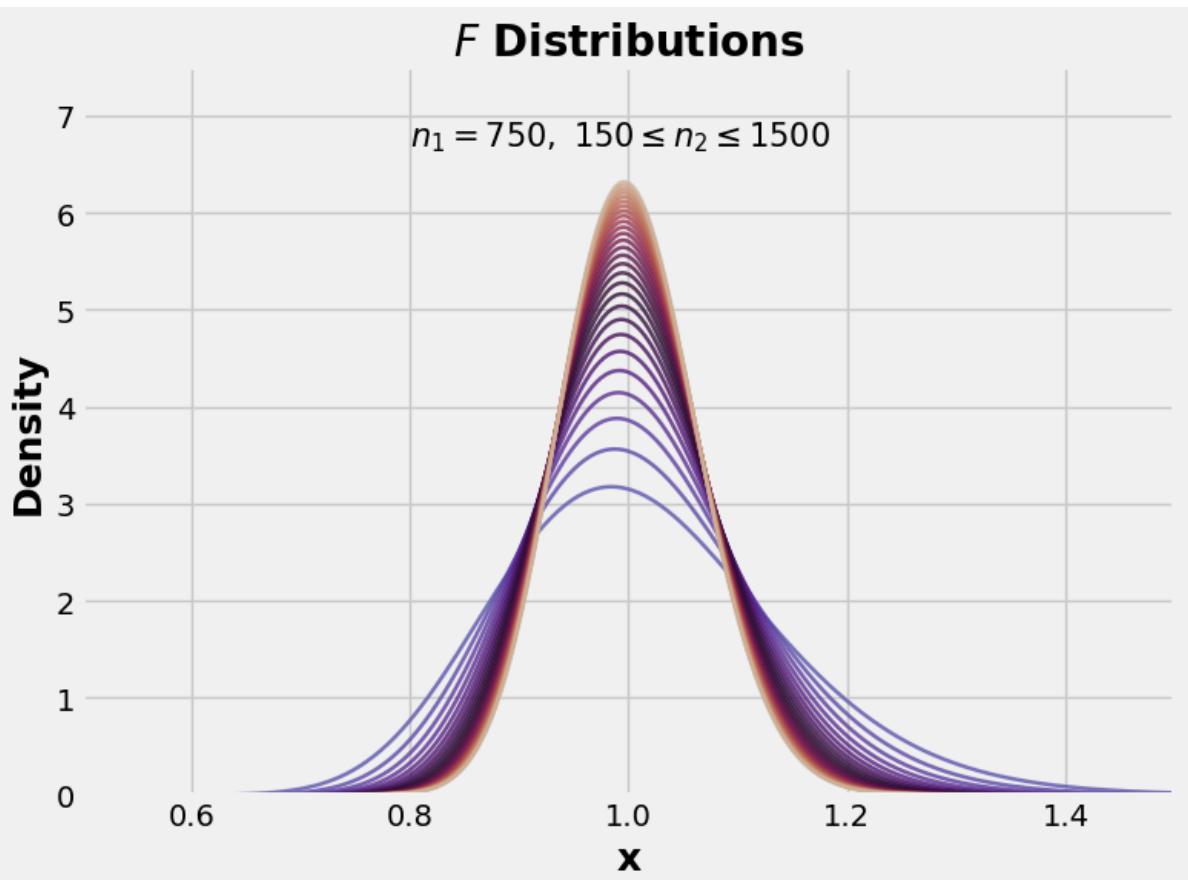
# fix xlim before animation
# ax.axis([xlim[0], xlim[1], 0, 0.4])
plt.xlabel('x', fontsize = 18, color = 'black', fontweight = 'bold')
plt.ylabel('Density', fontsize = 18, color = 'black', fontweight = 'bold')
plt.title(r'$F$ Distributions', fontsize = 20, fontweight = 'bold')

for pr, color in zip(n2, colors):
    y = f.pdf(x, dfn = n1, dfd = pr)
    line, = ax.plot(x, y, lw = 1.8, color = color, alpha = 0.8, label = r'$n_1$ = {}  
$n_2$ = {}'.format(n1, pr))
    ax.legend([line], [line.get_label()], loc = 'upper right')
    ax.tick_params(axis = 'both', colors = 'black')
    ax.set_xlim([0.5, 1.5])
    ax.set_ylim([0, 7.5])
    ax.set_yticks(np.arange(0, 7.5, 1))
    plt.tight_layout()
    # plt.pause(0.5)

    # 清除之前的图例
    if ax.get_legend() is not None:
        ax.get_legend().remove()

ax.text(0.8, 6.7, '$n_1$ = {}, $n_2$ = {}'.format(n1, n2[0]), n2[-1]), for
plt.show()

```



注意事項與討論：

觀察：

- 第一大圖：
 - Plot 1 呈現出在給定 $n_1 = 60$ 的情況下，分佈隨著 n_2 值由小到大的變動，可見分佈的趨勢從右偏逐漸變得不偏。
 - Plot 2 呈現出在給定 $n_2 = 60$ 的情況下，分佈隨著 n_1 值由小到大的變動，可見除了
一開始分佈趨勢會比 Plot 1 來得更加右偏以外，整體來看與 Plot 1 的分佈情況差別
不到。
- 第二大圖：
 - 從第一大圖裡觀察到 F 分佈隨著參數值變大，分佈趨勢似乎會逐漸變得不偏，故第
二大圖呈現出在給定 $n_1 = 750$ 的情況下，分佈隨著 n_2 值由 150 至 1500 的變動，
可見分佈趨勢在參數值很大時會逐漸接近不偏。
 - 以上程式碼若以 .py 形態並搭配 plt.pause(0.5) 執行，可以達成動畫的效果，可更直
觀感受分佈在不同參數設定下的變化。

結論：通過改變 F 分佈裡的兩個參數 n_1 與 n_2 ，可以觀察出分佈的不同模樣。

- n_1 與 n_2 的值互換對於分佈趨勢影響不大。
- 參數值越高，分佈趨勢將會越接近不偏。

第二大題：利用以上所用的分佈畫出 Histogram (直方圖) 、 Boxplot (盒鬚圖) 、 Probability plot (機率圖) 以及 Empirical CDF (經驗 CDF 圖)

1. Normal Distribution

- 分佈：

$$X \sim N(\mu, \sigma^2), \quad -\infty < \mu < \infty, \quad \sigma > 0$$

- PDF :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty$$

說明：以下將繪畫出 Normal Distribution 在 $\mu = 0, \sigma = 1$ 下的 Histogram (直方
圖) 、 Boxplot (盒鬚圖) 、 Probability plot (機率圖) 以及 Empirical CDF (經驗 CDF
圖) 。

第一大圖：

```
In [8]: import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.stats import norm
from scipy.stats import cumfreq # for ECDF
```

```

plt.style.use('fivethirtyeight')
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize = (13, 8))

# 參數設定
n, mu, s = 100, 0, 1
x = norm.rvs(loc = mu, scale = s, size = n)

# 直方圖 Histogram
ax1.hist(x, bins = 17, density = True, color = 'mediumslateblue', edgecolor = 'lightgray')
# PDF
pdf_x = np.linspace(-4, 4, 1000)
y = norm.pdf(pdf_x, loc = mu, scale = s)
ax1.plot(pdf_x, y, color = 'orangered', linestyle = '--', lw = 3, alpha = 0.8
         , label = 'Normal PDF')
ax1.set_title('The Histogram with {} samples'.format(n), fontsize = 18)
ax1.legend(loc = 'upper left')

# 盒鬚圖 Boxplot
boxprops = dict(linestyle = '--', linewidth = 3, color = 'royalblue')
whiskerprops = dict(color = 'cornflowerblue', linewidth = 2.5, linestyle = '--')
capprops = dict(color = 'cornflowerblue', linewidth = 2.5)
flierprops = dict(marker = 'o', markerfacecolor = 'gold', markersize = 7
                  , linestyle = 'none', markeredgecolor = 'darkkhaki') # define outlier
medianprops = dict(color = 'darkkhaki', linewidth = 3, linestyle = '-') # 设置中位数
labels = ['$N({})', '{}$]'.format(mu, s)]
ax2.boxplot(x, notch = True, vert = True, boxprops = boxprops, flierprops = flierprops
            , whiskerprops = whiskerprops, capprops = capprops, medianprops = medianprops
            , labels = labels)
ax2.set_title('The Box Plot with {} samples'.format(n), fontsize = 18)

# Normal Probability plot
stats.probplot(x, dist = 'norm', plot = ax3)
# [0] 為 QQ plot 的點 · [1] 為 QQ plot 的線
ax3.get_lines()[0].set_marker('o')
ax3.get_lines()[0].set_markerfacecolor('mediumspringgreen') # 设置点的颜色
ax3.get_lines()[0].set_markeredgecolor('none') # 设置点的边缘颜色
ax3.get_lines()[0].set_markersize(7) # 设置点的大小
ax3.get_lines()[1].set_color('orangered') # 设置拟合线的颜色
ax3.get_lines()[1].set_linewidth(3.5) # 设置拟合线的粗细
ax3.get_lines()[1].set_linestyle('--')
ax3.get_lines()[1].set_alpha(0.8)
ax3.set_title('N({}) , {} Probability Plot with {} samples'.format(mu, s, n), fontsize = 18)
ax3.set_xlabel('')
ax3.set_ylabel('')

# 經驗累積密度函數 Empirical CDF (ECDF)
res = cumfreq(x, n) # 生成 100 個區間的 ECDF

# 原理 : cumfreq(sample, n) 生成 n(放入的參數) 個區間 · 每個區間間隔(bin)相同 res.binsize
#       並計算每個區間的樣本累積數量 res.cumcount
#       每個 res.cumcount 代表了小於等於該區間的樣本數量 · 所以 res.cumcount.size 代表了總樣本數量
# print(res.lowerlimit, res.binsize, res.cumcount.size)
# res.cumcount 為累積次數 · res.binsize 為每個 bin 的寬度;
# res.lowerlimit 為最小值 · res.lowerlimit + (res.binsize * res.cumcount.size) 為最大值

# 利用 min 和 max 間隔 res.cumcount.size 生成 x 軸
ecdf_x = res.lowerlimit + np.linspace(0, res.binsize * res.cumcount.size, res.cumcount.size)

```

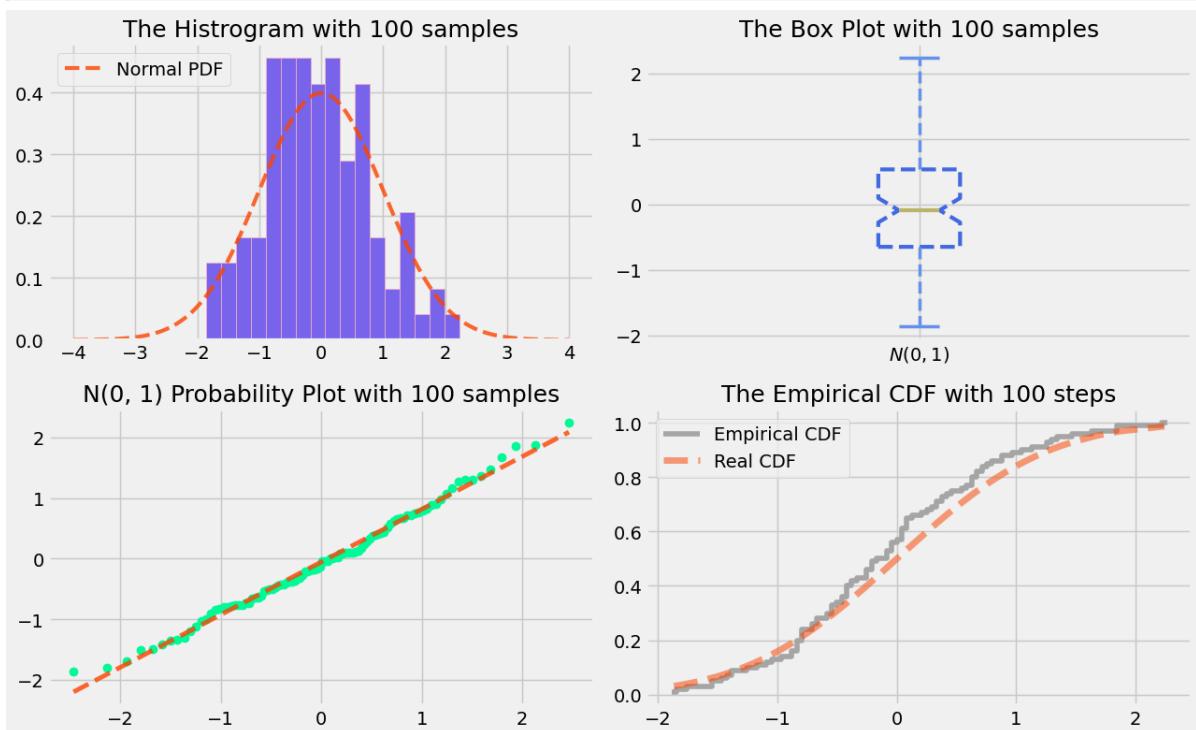
```

cumcountprob = res.cumcount / n # 累積次數除以總樣本數量 = 累積機率
# ax4.bar(x, cumcountprob, width = res.binsize)
ax4.plot(ecdf_x, cumcountprob, drawstyle = 'steps-pre', label = 'Empirical CDF'
         , color = 'darkgray')
ax4.set_title('The Empirical CDF with {} steps'.format(n), fontsize = 18)

# # draw the theoretical CDF of Z
x = np.sort(x)
cdf_y = norm.cdf(x, loc = mu, scale = s)
ax4.plot(x, cdf_y, color = 'orangered', linestyle = '--', linewidth = 5
         , alpha = 0.5, label = 'Real CDF')
ax4.legend(loc = 'upper left')

plt.tight_layout()
plt.show()

```



第二大圖：

```

In [9]: import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.stats import norm
from scipy.stats import cumfreq # for ECDF

plt.style.use('fivethirtyeight')
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize = (13, 8))

# 參數設定
n, mu, s = 100000, 0, 1
x = norm.rvs(loc = mu, scale = s, size = n)

# 直方圖 Histogram
ax1.hist(x, bins = 30, density = True, color = 'mediumslateblue', edgecolor = 'lightblue'
# PDF
pdf_x = np.linspace(-4, 4, 1000)
y = norm.pdf(pdf_x, loc = mu, scale = s)
ax1.plot(pdf_x, y, color = 'orangered', linestyle = '--', lw = 3, alpha = 0.8
         , label = 'Normal PDF')

```

```

ax1.set_title('The Histogram with {} samples'.format(n), fontsize = 18)
ax1.legend(loc = 'upper left')

# 盒鬚圖 Boxplot
boxprops = dict(linestyle = '--', linewidth = 3, color = 'royalblue')
whiskerprops = dict(color = 'cornflowerblue', linewidth = 2.5, linestyle = '--')
capprops = dict(color = 'cornflowerblue', linewidth = 2.5)
flierprops = dict(marker = 'o', markerfacecolor = 'gold', markersize = 7
                  , linestyle = 'none', markeredgecolor = 'darkkhaki') # define flierprops
medianprops = dict(color = 'darkkhaki', linewidth = 3, linestyle = '-') # 设置中位数
labels = ['$N({}), {}'.format(mu, s)] 
ax2.boxplot(x, notch = True, vert = True, boxprops = boxprops, flierprops = flierprops
            , whiskerprops = whiskerprops, capprops = capprops, medianprops = medianprops
            , labels = labels)
ax2.set_title('The Box Plot with {} samples'.format(n), fontsize = 18)

# Normal Probability plot
stats.probplot(x, dist = 'norm', plot = ax3)
# [0] 為 QQ plot 的點 · [1] 為 QQ plot 的線
ax3.get_lines()[0].set_marker('o')
ax3.get_lines()[0].set_markerfacecolor('mediumspringgreen') # 设置点的颜色
ax3.get_lines()[0].set_markeredgecolor('none') # 设置点的边缘颜色
ax3.get_lines()[0].set_markersize(7) # 设置点的大小
ax3.get_lines()[1].set_color('orangered') # 设置拟合线的颜色
ax3.get_lines()[1].set_linewidth(3.5) # 设置拟合线的粗细
ax3.get_lines()[1].set_linestyle('--')
ax3.get_lines()[1].set_alpha(0.8)
ax3.set_title('N({}), {} Probability Plot with {} samples'.format(mu, s, n), fontsize = 18)
ax3.set_xlabel('')
ax3.set_ylabel('')

# 經驗累積密度函數 Empirical CDF (ECDF)
res = cumfreq(x, n) # 生成 100 個區間的 ECDF

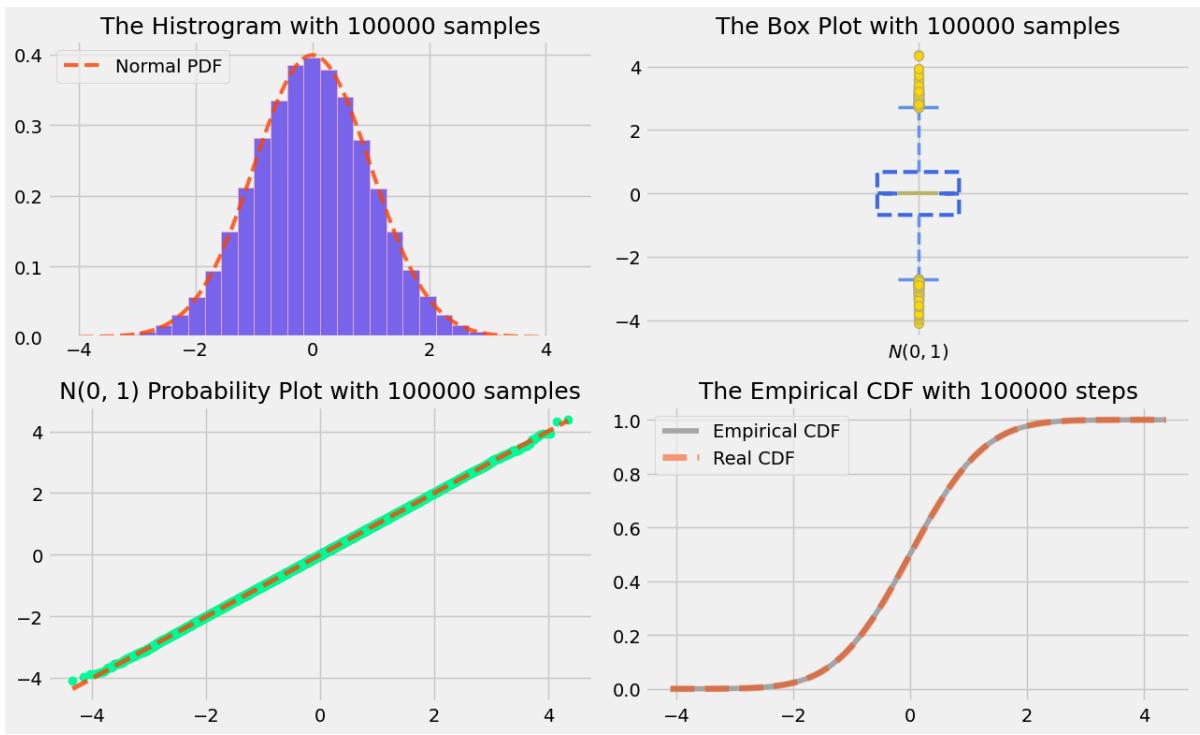
# 利用 min 和 max 間隔 res.cumcount.size 生成 x 軸
ecdf_x = res.lowerlimit + np.linspace(0, res.binsize * res.cumcount.size, res.cumcount.size)

cumcountprob = res.cumcount / n # 累積次數除以總樣本數量 = 累積機率
ax4.plot(ecdf_x, cumcountprob, drawstyle = 'steps-pre', label = 'Empirical CDF'
          , color = 'darkgray')
ax4.set_title('The Empirical CDF with {} steps'.format(n), fontsize = 18)

## draw the theoretical CDF of Z
x = np.sort(x)
cdf_y = norm.cdf(x, loc = mu, scale = s)
ax4.plot(x, cdf_y, color = 'orangered', linestyle = '--', linewidth = 5
          , alpha = 0.5, label = 'Real CDF')
ax4.legend(loc = 'upper left')

plt.tight_layout()
plt.show()

```



注意事項與討論：

觀察：

- 第一大圖：當樣本數為 100 時，無論是 Histogram、Boxplot、Probability plot 還是 Empirical CDF plot 的分佈情況都還無法很直觀地看出其母體分佈 $N(0, 1)$ 的模樣，但是已經可以大致判斷出它是個對稱的分佈。這表示樣本數還不夠，或是所收集的樣本不具有很強的代表性，無法代表母體分佈，因此還需要更多的樣本。
- 第二大圖：當樣本數增加到 100000 時，可以觀察到無論是 Histogram、Boxplot、Probability plot 還是 Empirical CDF plot 的分佈情況皆呈現出對稱的趨勢，其中 Empirical CDF plot 則幾乎與真實的 CDF 分佈重疊在一起，表示樣本已具有足夠的代表性，已近似於母體分佈 $N(0, 1)$ 的模樣。

結論：樣本數越大，抽樣分佈將越接近母體分佈，故當收集到的樣本數過少時，除非樣本具有強烈的代表性，否則將只能繼續收集樣本，直到該樣本數足夠代表其母體分佈。

2. χ^2 Distribution

- 分佈：

$$X \sim \chi^2(\nu), \quad \nu = 1, 2, \dots$$

- *PDF* :

$$f(x) = \frac{1}{\Gamma(\frac{\nu}{2}) 2^{\frac{\nu}{2}}} x^{\frac{\nu}{2}-1} e^{-\frac{x}{2}}, \quad 0 < x < \infty$$

說明：以下將繪畫出 χ^2 Distribution 在 $d.f. = 10$ 下的 Histogram (直方圖)、Boxplot (盒鬚圖)、Probability plot (機率圖) 以及 Empirical CDF (經驗 CDF 圖)。

第一大圖：

```
In [10]: import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.stats import norm, chi2
from scipy.stats import cumfreq

plt.style.use('fivethirtyeight')
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize = (13, 8))

# 參數設定
n, df = 100, 10
x = chi2.rvs(df = df, size = n)

# 直方圖 Histogram
ax1.hist(x, bins = 17, density = True, color = 'mediumslateblue', edgecolor = 'lightblue')
# PDF
pdf_x = np.linspace(0, 30, 1000)
y = chi2.pdf(pdf_x, df = df)
ax1.plot(pdf_x, y, color = 'orangered', linestyle = '--', lw = 3, alpha = 0.8
         , label = '$\chi^2$ PDF')
ax1.set_title('The Histogram with {} samples'.format(n), fontsize = 18)
ax1.legend()

# 盒鬚圖 Boxplot
boxprops = dict(linestyle = '--', linewidth = 3, color = 'royalblue')
whiskerprops = dict(color = 'cornflowerblue', linewidth = 2.5, linestyle = '--')
capprops = dict(color = 'cornflowerblue', linewidth = 2.5)
flierprops = dict(marker = 'o', markerfacecolor = 'gold', markersize = 7
                  , linestyle = 'none', markeredgecolor = 'darkkhaki') # define colors
medianprops = dict(color = 'darkkhaki', linewidth = 3, linestyle = '-') # 设置中位数
labels = ['$\chi^2({})'.format(df)] 
ax2.boxplot(x, notch = True, vert = True, boxprops = boxprops, flierprops = flierprops
            , whiskerprops = whiskerprops, capprops = capprops, medianprops = medianprops
            , labels = labels)
ax2.set_title('The Box Plot with {} samples'.format(n), fontsize = 18)

# Normal Probability plot
stats.probplot(x, dist = 'chi2', sparams = (df,), plot = ax3)
# [0] 為 QQ plot 的點 · [1] 為 QQ plot 的線
ax3.get_lines()[0].set_marker('o')
ax3.get_lines()[0].set_markerfacecolor('mediumspringgreen') # 设置点的颜色
ax3.get_lines()[0].set_markeredgecolor('none') # 设置点的边缘颜色
ax3.get_lines()[0].set_markersize(7) # 设置点的大小
ax3.get_lines()[1].set_color('orangered') # 设置拟合线的颜色
ax3.get_lines()[1].set_linewidth(3.5) # 设置拟合线的粗细
ax3.get_lines()[1].set_linestyle('--')
ax3.get_lines()[1].set_alpha(0.8)
ax3.set_title('$\chi^2({})$ Probability Plot with {} samples'.format(df, n), fontsize = 18)
ax3.set_xlabel('')
ax3.set_ylabel('')

# 經驗累積密度函數 Empirical CDF (ECDF)
res = cumfreq(x, n) # 生成 100 個區間的 ECDF

# 利用 min 和 max 間隔 res.cumcount.size 生成 x 軸
ecdf_x = res.lowerlimit + np.linspace(0, res.binsize * res.cumcount.size, res.cumcount.size)
```

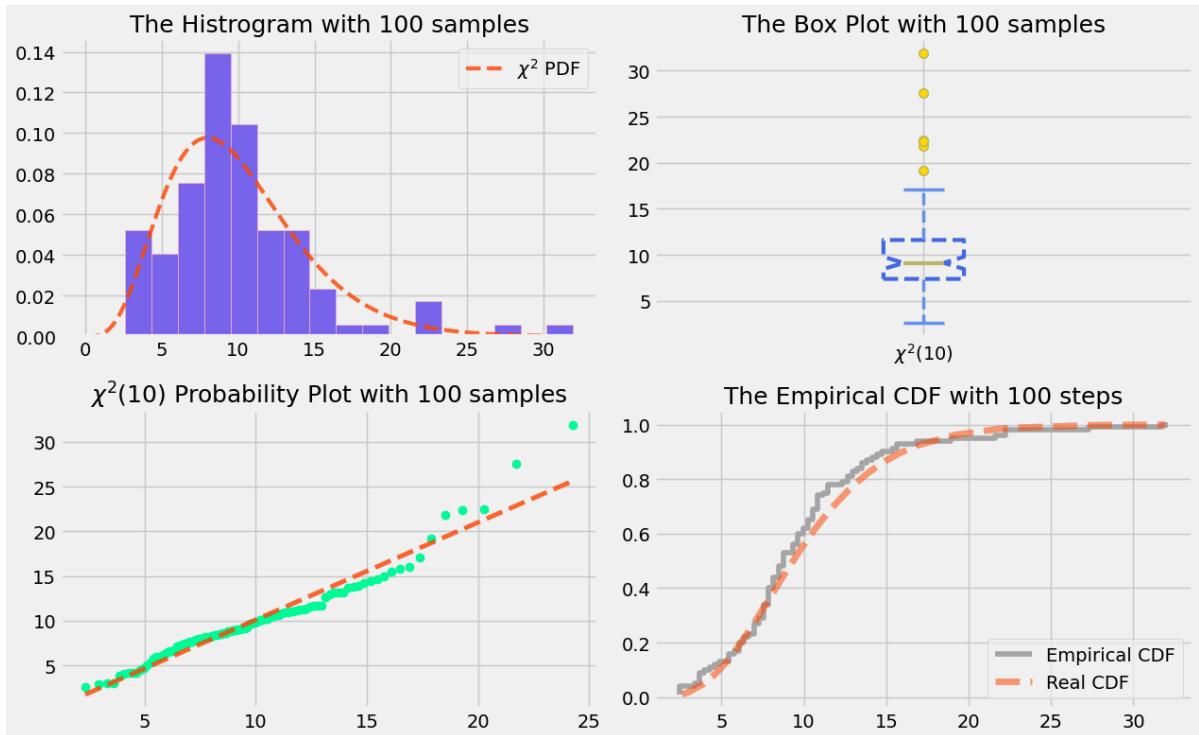
```

cumcountprob = res.cumcount / n # 累積次數除以總樣本數量 = 累積機率
ax4.plot(ecdf_x, cumcountprob, drawstyle = 'steps-pre', label = 'Empirical CDF'
          , color = 'darkgray')
ax4.set_title('The Empirical CDF with {} steps'.format(n), fontsize = 18)

# # draw the theoretical CDF of Z
x = np.sort(x)
cdf_y = chi2.cdf(x, df = df)
ax4.plot(x, cdf_y, color = 'orangered', linestyle = '--', linewidth = 5
          , alpha = 0.5, label = 'Real CDF')
ax4.legend(loc = 'lower right')

plt.tight_layout()
plt.show()

```



第二大圖：

```

In [11]: import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.stats import norm, chi2
from scipy.stats import cumfreq

plt.style.use('fivethirtyeight')
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize = (13, 8))

# 參數設定
n, df = 100000, 10
x = chi2.rvs(df = df, size = n)

# 直方圖 Histogram
ax1.hist(x, bins = 30, density = True, color = 'mediumslateblue', edgecolor = 'lightgray')
# PDF
pdf_x = np.linspace(0, 30, 1000)
y = chi2.pdf(pdf_x, df = df)
ax1.plot(pdf_x, y, color = 'orangered', linestyle = '--', lw = 3, alpha = 0.8
          , label = '$\chi^2$ PDF')
ax1.set_title('The Histogram with {} samples'.format(n), fontsize = 18)

```

```

ax1.legend()

# 盒鬚圖 Boxplot
boxprops = dict(linestyle = '--', linewidth = 3, color = 'royalblue')
whiskerprops = dict(color = 'cornflowerblue', linewidth = 2.5, linestyle = '--')
capprops = dict(color = 'cornflowerblue', linewidth = 2.5)
flierprops = dict(marker = 'o', markerfacecolor = 'gold', markersize = 7
                  , linestyle = 'none', markeredgecolor = 'darkkhaki') # define flierprops
medianprops = dict(color = 'darkkhaki', linewidth = 3, linestyle = '-') # 设置中位数
labels = ['$\chi^2({})'.format(df)] # 样本量的标签
ax2.boxplot(x, notch = True, vert = True, boxprops = boxprops, flierprops = flierprops
            , whiskerprops = whiskerprops, capprops = capprops, medianprops = medianprops
            , labels = labels)
ax2.set_title('The Box Plot with {} samples'.format(n), fontsize = 18)

# Normal Probability plot
stats.probplot(x, dist = 'chi2', sparams = (df,), plot = ax3)
# [0] 为 QQ plot 的点 · [1] 为 QQ plot 的线
ax3.get_lines()[0].set_marker('o')
ax3.get_lines()[0].set_markerfacecolor('mediumspringgreen') # 设置点的颜色
ax3.get_lines()[0].set_markeredgecolor('none') # 设置点的边缘颜色
ax3.get_lines()[0].set_markersize(7) # 设置点的大小
ax3.get_lines()[1].set_color('orangered') # 设置拟合线的颜色
ax3.get_lines()[1].set_linewidth(3.5) # 设置拟合线的粗细
ax3.get_lines()[1].set_linestyle('--')
ax3.get_lines()[1].set_alpha(0.8)
ax3.set_title('$\chi^2({})$ Probability Plot with {} samples'.format(df, n), fontsize = 18)
ax3.set_xlabel('')
ax3.set_ylabel('')

# 絏驗累積密度函數 Empirical CDF (ECDF)
res = cumfreq(x, n) # 生成 100 個區間的 ECDF

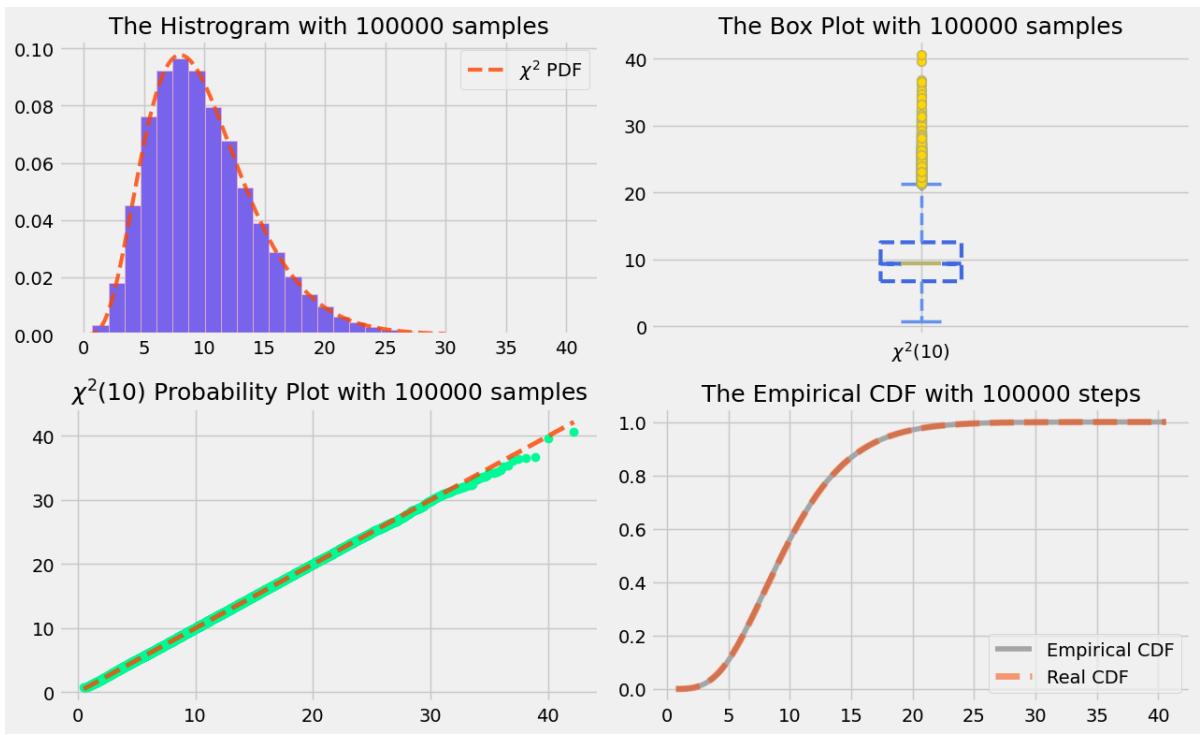
# 利用 min 和 max 間隔 res.cumcount.size 生成 x 軸
ecdf_x = res.lowerlimit + np.linspace(0, res.binsize * res.cumcount.size, res.cumcount.size)

cumcountprob = res.cumcount / n # 累積次數除以總樣本數量 = 累積機率
ax4.plot(ecdf_x, cumcountprob, drawstyle = 'steps-pre', label = 'Empirical CDF'
          , color = 'darkgray')
ax4.set_title('The Empirical CDF with {} steps'.format(n), fontsize = 18)

## draw the theoretical CDF of Z
x = np.sort(x)
cdf_y = chi2.cdf(x, df = df)
ax4.plot(x, cdf_y, color = 'orangered', linestyle = '--', linewidth = 5
          , alpha = 0.5, label = 'Real CDF')
ax4.legend(loc = 'lower right')

plt.tight_layout()
plt.show()

```



注意事項與討論：

觀察：

- 第一大圖：當樣本數為 100 時，無論是 Histogram、Boxplot、Probability plot 還是 Empirical CDF plot 的分佈情況都還無法很直觀地看出其母體分佈 $\chi^2(10)$ 的模樣，但是已經可以大致判斷出它是個右偏的分佈。這表示樣本數還不夠，或是所收集的樣本不具有很強的代表性，無法代表母體分佈，因此還需要更多的樣本。
- 第二大圖：當樣本數增加到 100000 時，可以觀察到無論是 Histogram、Boxplot、Probability plot 還是 Empirical CDF plot 的分佈情況皆呈現出右偏的趨勢，其中 Empirical CDF plot 則幾乎與真實的 CDF 分佈重疊在一起，表示樣本已具有足夠的代表性，已近似於母體分佈 $\chi^2(10)$ 的模樣。

結論：樣本數越大，抽樣分佈將越接近母體分佈，故當收集到的樣本數過少時，除非樣本具有強烈的代表性，否則將只能繼續收集樣本，直到該樣本數足夠代表其母體分佈。

3. T Distribution

- 分佈：

$$X \sim T(\nu), \quad \nu > 0$$

- PDF :

$$f(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}, \quad -\infty < x < \infty$$

說明：以下將繪畫出 T Distribution 在 $d.f. = 10$ 下的 Histogram (直方圖) 、Boxplot (盒鬚圖) 、Probability plot (機率圖) 以及 Empirical CDF (經驗 CDF 圖) 。

第一大圖：

```
In [12]: import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.stats import norm, chi2, t
from scipy.stats import cumfreq

plt.style.use('fivethirtyeight')
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize = (13, 8))

# 參數設定
n, df = 100, 10
x = t.rvs(df = df, size = n)

# 直方圖 Histogram
ax1.hist(x, bins = 17, density = True, color = 'mediumslateblue', edgecolor = 'lightblue',
         # PDF
         pdf_x = np.linspace(-4, 4, 1000)
         y = t.pdf(pdf_x, df = df)
         ax1.plot(pdf_x, y, color = 'orangered', linestyle = '--', lw = 3, alpha = 0.8
                  , label = '$T$ PDF')
ax1.set_title('The Histogram with {} samples'.format(n), fontsize = 18)
ax1.legend()

# 盒鬚圖 Boxplot
boxprops = dict(linestyle = '--', linewidth = 3, color = 'royalblue')
whiskerprops = dict(color = 'cornflowerblue', linewidth = 2.5, linestyle = '--')
capprops = dict(color = 'cornflowerblue', linewidth = 2.5)
flierprops = dict(marker = 'o', markerfacecolor = 'gold', markersize = 7
                  , linestyle = 'none', markeredgecolor = 'darkkhaki') # define outlier
medianprops = dict(color = 'darkkhaki', linewidth = 3, linestyle = '-') # 设置中位数
labels = ['$T({})$'.format(df)]
ax2.boxplot(x, notch = True, vert = True, boxprops = boxprops, flierprops = flierprops
            , whiskerprops = whiskerprops, capprops = capprops, medianprops = medianprops
            , labels = labels)
ax2.set_title('The Box Plot with {} samples'.format(n), fontsize = 18)

# Normal Probability plot
stats.probplot(x, dist = 't', sparams = (df,), plot = ax3)
# [0] 為 QQ plot 的點 · [1] 為 QQ plot 的線
ax3.get_lines()[0].set_marker('o')
ax3.get_lines()[0].set_markerfacecolor('mediumspringgreen') # 设置点的颜色
ax3.get_lines()[0].set_markeredgecolor('none') # 设置点的边缘颜色
ax3.get_lines()[0].set_markersize(7) # 设置点的大小
ax3.get_lines()[1].set_color('orangered') # 设置拟合线的颜色
ax3.get_lines()[1].set_linewidth(3.5) # 设置拟合线的粗细
ax3.get_lines()[1].set_linestyle('--')
ax3.get_lines()[1].set_alpha(0.8)
ax3.set_title('T({}) Probability Plot with {} samples'.format(df, n), fontsize = 18)
ax3.set_xlabel('')
ax3.set_ylabel('')

# 經驗累積密度函數 Empirical CDF (ECDF )
res = cumfreq(x, n) # 生成 100 個區間的 ECDF

# 利用 min 和 max 間隔 res.cumcount.size 生成 x 軸
ecdf_x = res.lowerlimit + np.linspace(0, res.binsize * res.cumcount.size, res.cumcount.size)
```

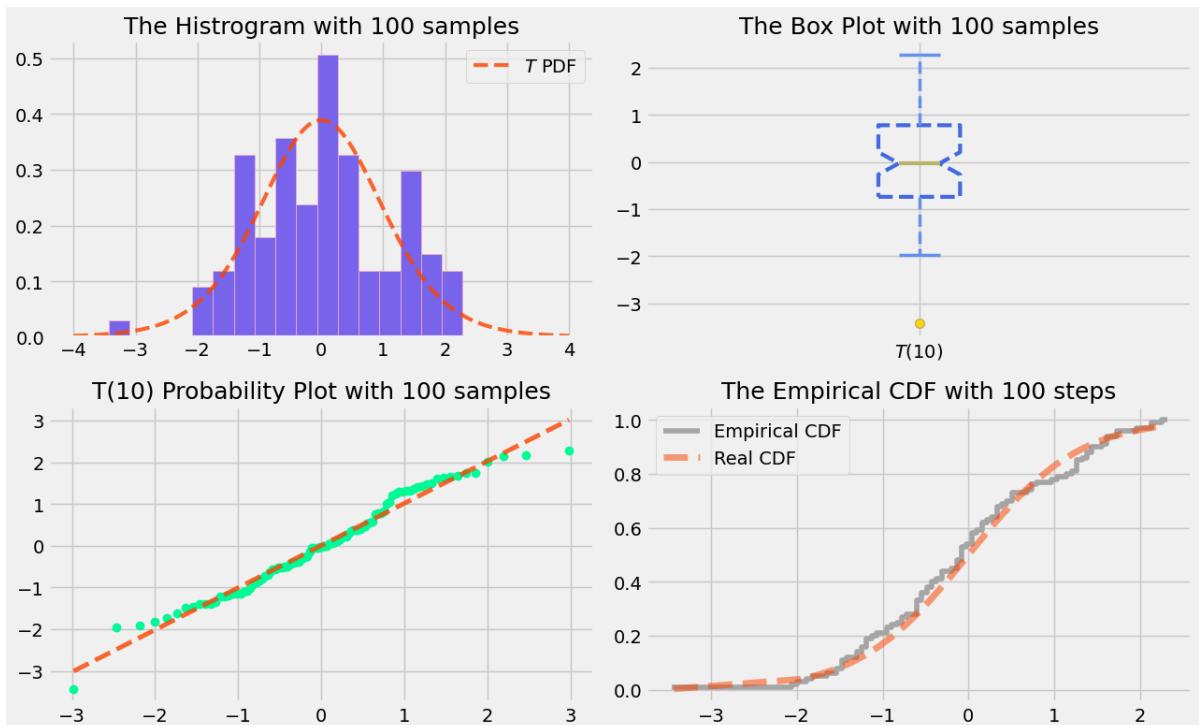
```

cumcountprob = res.cumcount / n # 累積次數除以總樣本數量 = 累積機率
ax4.plot(ecdf_x, cumcountprob, drawstyle = 'steps-pre', label = 'Empirical CDF'
          , color = 'darkgray')
ax4.set_title('The Empirical CDF with {} steps'.format(n), fontsize = 18)

# # draw the theoretical CDF of Z
x = np.sort(x)
cdf_y = t.cdf(x, df = df)
ax4.plot(x, cdf_y, color = 'orangered', linestyle = '--', linewidth = 5
          , alpha = 0.5, label = 'Real CDF')
ax4.legend()

plt.tight_layout()
plt.show()

```



第二大圖：

```

In [13]: import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.stats import norm, chi2, t
from scipy.stats import cumfreq

plt.style.use('fivethirtyeight')
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize = (13, 8))

# 參數設定
n, df = 1000000, 30
x = t.rvs(df = df, size = n)

# 直方圖 Histogram
ax1.hist(x, bins = 50, density = True, color = 'mediumslateblue', edgecolor = 'lightblue')
# PDF
pdf_x = np.linspace(-4, 4, 1000)
y = t.pdf(pdf_x, df = df)
ax1.plot(pdf_x, y, color = 'orangered', linestyle = '--', lw = 3, alpha = 0.8
          , label = '$T$ PDF')
ax1.set_title('The Histogram with {} samples'.format(n), fontsize = 18)

```

```

ax1.set_xlim([-4, 4])
ax1.legend()

# 盒鬚圖 Boxplot
boxprops = dict(linestyle = '--', linewidth = 3, color = 'royalblue')
whiskerprops = dict(color = 'cornflowerblue', linewidth = 2.5, linestyle = '--')
capprops = dict(color = 'cornflowerblue', linewidth = 2.5)
flierprops = dict(marker = 'o', markerfacecolor = 'gold', markersize = 7
                  , linestyle = 'none', markeredgecolor = 'darkkhaki') # define flierprops
medianprops = dict(color = 'darkkhaki', linewidth = 3, linestyle = '-') # 设置中位数
labels = ['$T({})$'.format(df)]
ax2.boxplot(x, notch = True, vert = True, boxprops = boxprops, flierprops = flierprops
            , whiskerprops = whiskerprops, capprops = capprops, medianprops = medianprops
            , labels = labels)
ax2.set_title('The Box Plot with {} samples'.format(n), fontsize = 18)

# Normal Probability plot
stats.probplot(x, dist = 't', sparams = (df,), plot = ax3)
# [0] 為 QQ plot 的點 · [1] 為 QQ plot 的線
ax3.get_lines()[0].set_marker('o')
ax3.get_lines()[0].set_markerfacecolor('mediumspringgreen') # 设置点的颜色
ax3.get_lines()[0].set_markeredgecolor('none') # 设置点的边缘颜色
ax3.get_lines()[0].set_markersize(7) # 设置点的大小
ax3.get_lines()[1].set_color('orangered') # 设置拟合线的颜色
ax3.get_lines()[1].set_linewidth(3.5) # 设置拟合线的粗细
ax3.get_lines()[1].set_linestyle('--')
ax3.get_lines()[1].set_alpha(0.8)
ax3.set_title('T({}) Probability Plot with {} samples'.format(df, n), fontsize = 18)
ax3.set_xlabel('')
ax3.set_ylabel('')

# 經驗累積密度函數 Empirical CDF (ECDF)
res = cumfreq(x, n) # 生成 100 個區間的 ECDF

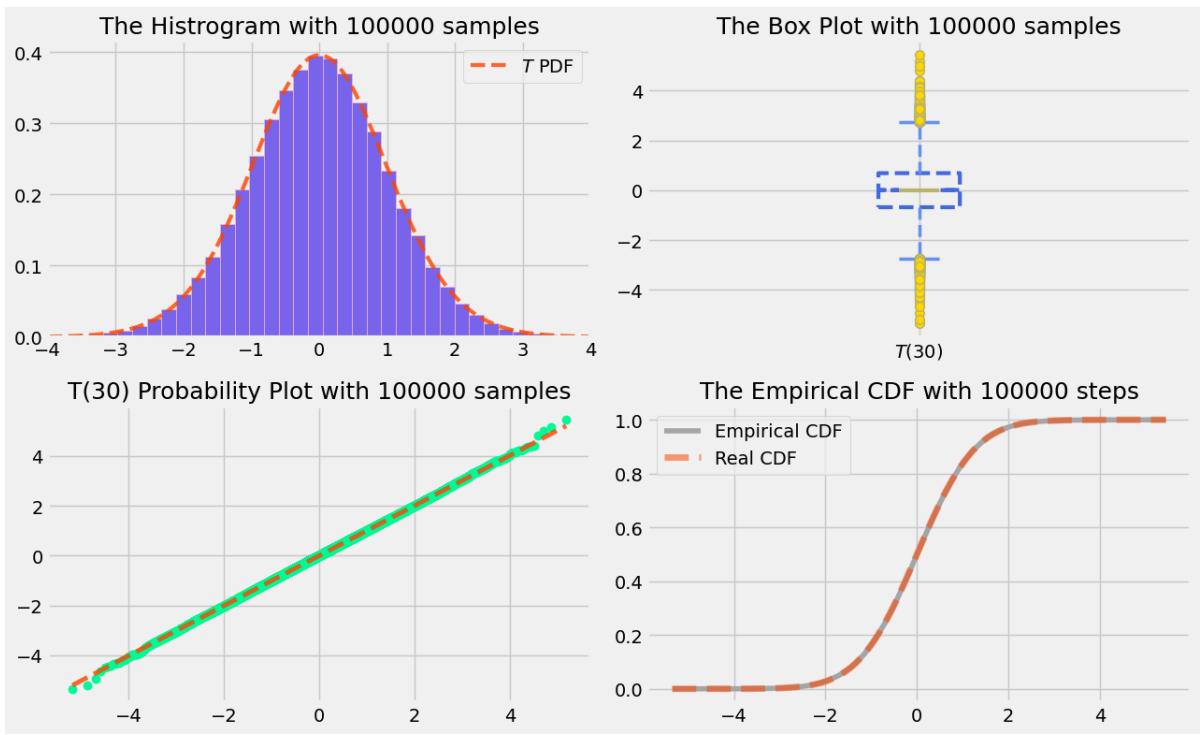
# 利用 min 和 max 間隔 res.cumcount.size 生成 x 軸
ecdf_x = res.lowerlimit + np.linspace(0, res.binsize * res.cumcount.size, res.cumcount.size)

cumcountprob = res.cumcount / n # 累積次數除以總樣本數量 = 累積機率
ax4.plot(ecdf_x, cumcountprob, drawstyle = 'steps-pre', label = 'Empirical CDF'
          , color = 'darkgray')
ax4.set_title('The Empirical CDF with {} steps'.format(n), fontsize = 18)

## draw the theoretical CDF of Z
x = np.sort(x)
cdf_y = t.cdf(x, df = df)
ax4.plot(x, cdf_y, color = 'orangered', linestyle = '--', linewidth = 5
          , alpha = 0.5, label = 'Real CDF')
ax4.legend()

plt.tight_layout()
plt.show()

```



注意事項與討論：

觀察：

- 第一大圖：當樣本數為 100 時，無論是 Histogram、Boxplot、Probability plot 還是 Empirical CDF plot 的分佈情況都還無法很直觀地看出其母體分佈 $T(30)$ 的模樣，但是已經可以大致判斷出它是個對稱的分佈。這表示樣本數還不夠，或是所收集的樣本不具有很強的代表性，無法代表母體分佈，因此還需要更多的樣本。
- 第二大圖：當樣本數增加到 100000 時，可以觀察到無論是 Histogram、Boxplot、Probability plot 還是 Empirical CDF plot 的分佈情況皆呈現出對稱的趨勢，其中 Empirical CDF plot 則幾乎與真實的 CDF 分佈重疊在一起，表示樣本已具有足夠的代表性，已近似於母體分佈 $T(30)$ 的模樣。

結論：樣本數越大，抽樣分佈將越接近母體分佈，故當收集到的樣本數過少時，除非樣本具有強烈的代表性，否則將只能繼續收集樣本，直到該樣本數足夠代表其母體分佈。

4. β Distribution

- 分佈：

$$X \sim Beta(\alpha, \beta), \quad \alpha, \beta > 0$$

- PDF ：

$$f(x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}, \quad 0 < x < 1$$

說明：以下將繪畫出 β Distribution 在 $\alpha = 10, \beta = 5$ 下的 Histogram (直方圖)、Boxplot (盒鬚圖)、Probability plot (機率圖) 以及 Empirical CDF (經驗 CDF 圖)。

第一大圖：

In [14]:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.stats import norm, chi2, t, beta
from scipy.stats import cumfreq

plt.style.use('fivethirtyeight')
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize = (13, 8))

# 參數設定
n, a, b = 100, 10, 5
x = beta.rvs(a = a, b = b, size = n)

# 直方圖 Histogram
ax1.hist(x, bins = 17, density = True, color = 'mediumslateblue', edgecolor = 'lightblue')
# PDF
pdf_x = np.linspace(0, 1, 1000)
y = beta.pdf(pdf_x, a = a, b = b)
ax1.plot(pdf_x, y, color = 'orangered', linestyle = '--', lw = 3, alpha = 0.8
          , label = r'$\beta$ PDF')
ax1.set_title('The Histogram with {} samples'.format(n), fontsize = 18)
ax1.legend(loc = 'upper left')

# 盒鬚圖 Boxplot
boxprops = dict(linestyle = '--', linewidth = 3, color = 'royalblue')
whiskerprops = dict(color = 'cornflowerblue', linewidth = 2.5, linestyle = '--')
capprops = dict(color = 'cornflowerblue', linewidth = 2.5)
flierprops = dict(marker = 'o', markerfacecolor = 'gold', markersize = 7
                  , linestyle = 'none', markeredgecolor = 'darkkhaki') # define colors
medianprops = dict(color = 'darkkhaki', linewidth = 3, linestyle = '-') # 设置中位数颜色
labels = ['$\Beta({}, {})'.format(a, b)]
ax2.boxplot(x, notch = True, vert = True, boxprops = boxprops, flierprops = flierprops
            , whiskerprops = whiskerprops, capprops = capprops, medianprops = medianprops
            , labels = labels)
ax2.set_title('The Box Plot with {} samples'.format(n), fontsize = 18)

# Normal Probability plot
stats.probplot(x, dist = 'beta', sparams = (a, b), plot = ax3)
# [0] 為 QQ plot 的點 · [1] 為 QQ plot 的線
ax3.get_lines()[0].set_marker('o')
ax3.get_lines()[0].set_markerfacecolor('mediumspringgreen') # 设置点的颜色
ax3.get_lines()[0].set_markeredgecolor('none') # 设置点的边缘颜色
ax3.get_lines()[0].set_markersize(7) # 设置点的大小
ax3.get_lines()[1].set_color('orangered') # 设置拟合线的颜色
ax3.get_lines()[1].set_linewidth(3.5) # 设置拟合线的粗细
ax3.get_lines()[1].set_linestyle('--')
ax3.get_lines()[1].set_alpha(0.8)
ax3.set_title('Beta({}, {}) Probability Plot with {} samples'.format(a, b, n), fontweight = 'bold')
ax3.set_xlabel('')
ax3.set_ylabel('')

# 經驗累積密度函數 Empirical CDF (ECDF)
res = cumfreq(x, n) # 生成 100 個區間的 ECDF

# 利用 min 和 max 間隔 res.cumcount.size 生成 x 軸
ecdf_x = res.lowerlimit + np.linspace(0, res.binsize * res.cumcount.size, res.cumcount.size)

cumcountprob = res.cumcount / n # 累積次數除以總樣本數量 = 累積機率
ax4.plot(ecdf_x, cumcountprob, drawstyle = 'steps-pre', label = 'Empirical CDF')
```

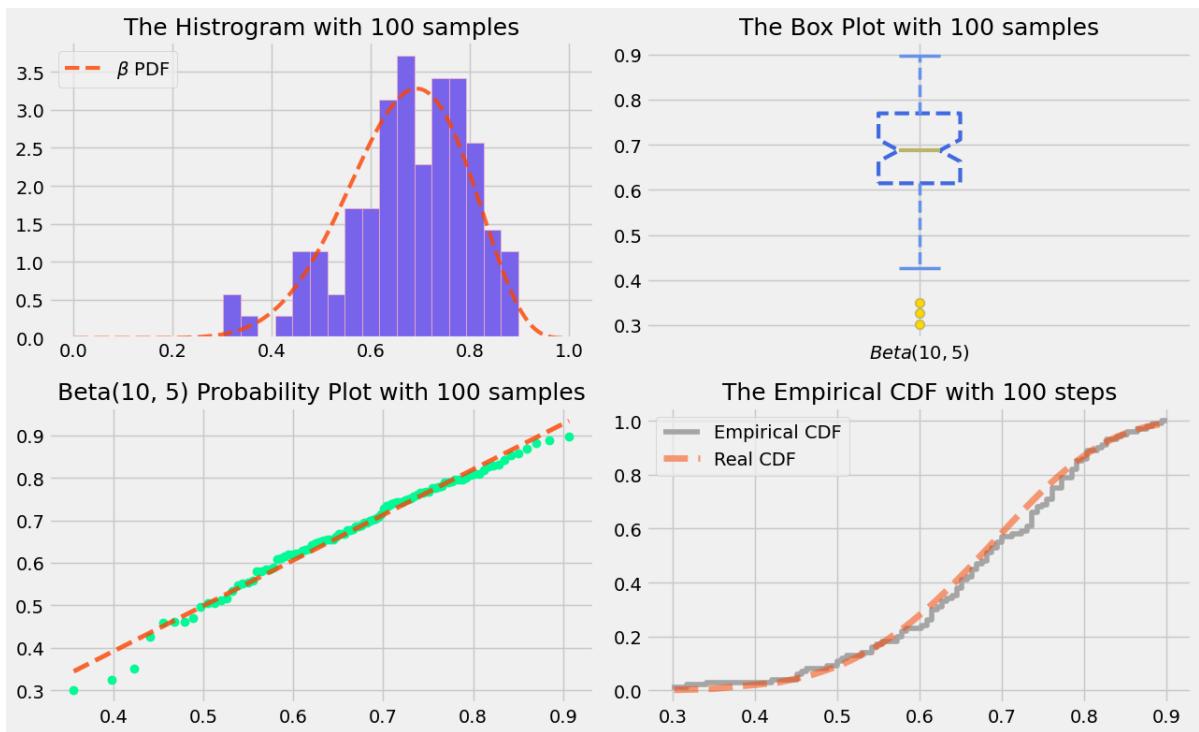
```

        , color = 'darkgray')
ax4.set_title('The Empirical CDF with {} steps'.format(n), fontsize = 18)

# # draw the theoretical CDF of Z
x = np.sort(x)
cdf_y = beta.cdf(x, a = a, b = b)
ax4.plot(x, cdf_y, color = 'orangered', linestyle = '--', linewidth = 5
          , alpha = 0.5, label = 'Real CDF')
ax4.legend()

plt.tight_layout()
plt.show()

```



第二大圖：

```

In [15]: import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.stats import norm, chi2, t, beta
from scipy.stats import cumfreq

plt.style.use('fivethirtyeight')
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize = (13, 8))

# 參數設定
n, a, b = 100000, 10, 5
x = beta.rvs(a = a, b = b, size = n)

# 直方圖 Histogram
ax1.hist(x, bins = 17, density = True, color = 'mediumslateblue', edgecolor = 'lightblue'
# PDF
pdf_x = np.linspace(0, 1, 1000)
y = beta.pdf(pdf_x, a = a, b = b)
ax1.plot(pdf_x, y, color = 'orangered', linestyle = '--', lw = 3, alpha = 0.8
          , label = r'$\beta$ PDF')
ax1.set_title('The Histogram with {} samples'.format(n), fontsize = 18)
ax1.legend(loc = 'upper left')

```

```

# 盒鬚圖 Boxplot
boxprops = dict(linestyle = '--', linewidth = 3, color = 'royalblue')
whiskerprops = dict(color = 'cornflowerblue', linewidth = 2.5, linestyle = '--')
capprops = dict(color = 'cornflowerblue', linewidth = 2.5)
flierprops = dict(marker = 'o', markerfacecolor = 'gold', markersize = 7
                  , linestyle = 'none', markeredgecolor = 'darkkhaki') # define outlier
medianprops = dict(color = 'darkkhaki', linewidth = 3, linestyle = '-') # 设置中位数
labels = ['$Beta({}, {})'.format(a, b)]
ax2.boxplot(x, notch = True, vert = True, boxprops = boxprops, flierprops = flierprops
            , whiskerprops = whiskerprops, capprops = capprops, medianprops = medianprops
            , labels = labels)
ax2.set_title('The Box Plot with {} samples'.format(n), fontsize = 18)

# Normal Probability plot
stats.probplot(x, dist = 'beta', sparams = (a, b), plot = ax3)
# [0] 為 QQ plot 的點 · [1] 為 QQ plot 的線
ax3.get_lines()[0].set_marker('o')
ax3.get_lines()[0].set_markerfacecolor('mediumspringgreen') # 设置点的颜色
ax3.get_lines()[0].set_markeredgecolor('none') # 设置点的边缘颜色
ax3.get_lines()[0].set_markersize(7) # 设置点的大小
ax3.get_lines()[1].set_color('orangered') # 设置拟合线的颜色
ax3.get_lines()[1].set_linewidth(3.5) # 设置拟合线的粗细
ax3.get_lines()[1].set_linestyle('--')
ax3.get_lines()[1].set_alpha(0.8)
ax3.set_title('Beta({}, {}) Probability Plot with {} samples'.format(a, b, n), fontweight = 'bold')
ax3.set_xlabel('')
ax3.set_ylabel('')

# 經驗累積密度函數 Empirical CDF (ECDF)
res = cumfreq(x, n) # 生成 100 個區間的 ECDF

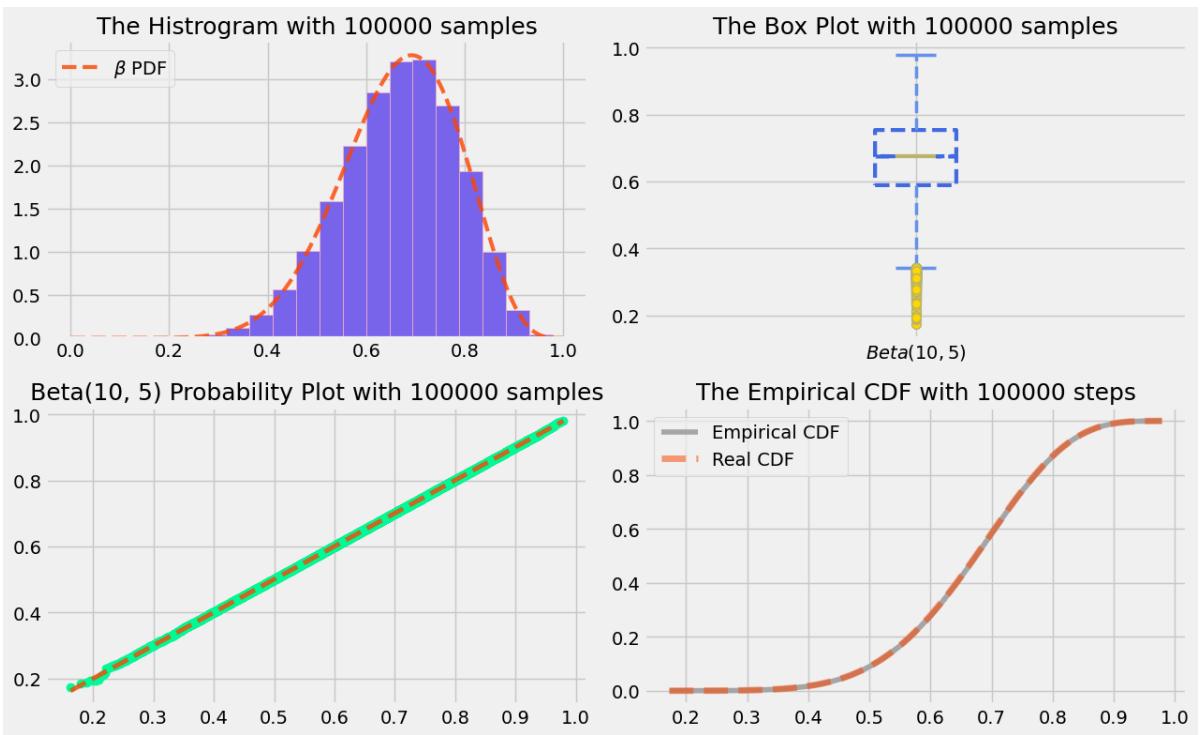
# 利用 min 和 max 間隔 res.cumcount.size 生成 x 軸
ecdf_x = res.lowerlimit + np.linspace(0, res.binsize * res.cumcount.size, res.cumcount.size)

cumcountprob = res.cumcount / n # 累積次數除以總樣本數量 = 累積機率
ax4.plot(ecdf_x, cumcountprob, drawstyle = 'steps-pre', label = 'Empirical CDF'
          , color = 'darkgray')
ax4.set_title('The Empirical CDF with {} steps'.format(n), fontsize = 18)

## draw the theoretical CDF of Z
x = np.sort(x)
cdf_y = beta.cdf(x, a = a, b = b)
ax4.plot(x, cdf_y, color = 'orangered', linestyle = '--', linewidth = 5
          , alpha = 0.5, label = 'Real CDF')
ax4.legend()

plt.tight_layout()
plt.show()

```



注意事項與討論：

觀察：

- 第一大圖：當樣本數為 100 時，無論是 Histogram、Boxplot、Probability plot 還是 Empirical CDF plot 的分佈情況都還無法很直觀地看出其母體分佈 Beta(10, 5) 的模樣，但是已經可以大致判斷出它是個左偏的分佈。這表示樣本數還不夠多，或是所收集的樣本不具有很強的代表性，無法代表母體分佈，因此還需要更多的樣本。
- 第二大圖：當樣本數增加到 100000 時，可以觀察到無論是 Histogram、Boxplot、Probability plot 還是 Empirical CDF plot 的分佈情況皆呈現出左偏的趨勢，其中 Empirical CDF plot 則幾乎與真實的 CDF 分佈重疊在一起，表示樣本已具有足夠的代表性，已近似於母體分佈 Beta(10, 5) 的模樣。

結論：樣本數越大，抽樣分佈將越接近母體分佈，故當收集到的樣本數過少時，除非樣本具有強烈的代表性，否則將只能繼續收集樣本，直到該樣本數足夠代表其母體分佈。

5. F Distribution

- 分佈：

$$X \sim F(n_1, n_2), \quad n_1, n_2 > 0$$

- PDF：

$$f(x) = \frac{\Gamma(\frac{n_1+n_2}{2})}{\Gamma(\frac{n_1}{2}) + \Gamma(\frac{n_2}{2})} \left(\frac{n_1}{n_2}\right)^{\frac{n_1}{2}} x^{\frac{n_1}{2}-1} (1 + \frac{n_1}{n_2}x)^{-\frac{n_1+n_2}{2}}, \quad x > 0$$

說明：以下將繪畫出 *F Distribution* 在 $n_1 = 60, n_2 = 30$ 下的 Histogram (直方圖) 、Boxplot (盒鬚圖) 、Probability plot (機率圖) 以及 Empirical CDF (經驗 CDF 圖) 。

第一大圖：

In [16]:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.stats import norm, chi2, t, beta, f
from scipy.stats import cumfreq

plt.style.use('fivethirtyeight')
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize = (13, 8))

# 參數設定
n, n1, n2 = 100, 60, 30
x = f.rvs(dfn = n1, dfd = n2, size = n)

# 直方圖 Histogram
ax1.hist(x, bins = 13, density = True, color = 'mediumslateblue', edgecolor = 'lightblue')
# PDF
pdf_x = np.linspace(0, 4, 1000)
y = f.pdf(pdf_x, dfn = n1, dfd = n2)
ax1.plot(pdf_x, y, color = 'orangered', linestyle = '--', lw = 3, alpha = 0.8
         , label = r'$\beta$ PDF')
ax1.set_title('The Histogram with {} samples'.format(n), fontsize = 18)
ax1.legend()

# 盒鬚圖 Boxplot
boxprops = dict(linestyle = '--', linewidth = 3, color = 'royalblue')
whiskerprops = dict(color = 'cornflowerblue', linewidth = 2.5, linestyle = '--')
capprops = dict(color = 'cornflowerblue', linewidth = 2.5)
flierprops = dict(marker = 'o', markerfacecolor = 'gold', markersize = 7
                  , linestyle = 'none', markeredgecolor = 'darkkhaki') # define colors
medianprops = dict(color = 'darkkhaki', linewidth = 3, linestyle = '-') # 设置中位数颜色
labels = ['$F({}, {})'.format(n1, n2)]
ax2.boxplot(x, notch = True, vert = True, boxprops = boxprops, flierprops = flierprops
            , whiskerprops = whiskerprops, capprops = capprops, medianprops = medianprops
            , labels = labels)
ax2.set_title('The Box Plot with {} samples'.format(n), fontsize = 18)

# Normal Probability plot
stats.probplot(x, dist = 'f', sparams = (n1, n2), plot = ax3)
# [0] 為 QQ plot 的點 · [1] 為 QQ plot 的線
ax3.get_lines()[0].set_marker('o')
ax3.get_lines()[0].set_markerfacecolor('mediumspringgreen') # 设置点的颜色
ax3.get_lines()[0].set_markeredgecolor('none') # 设置点的边缘颜色
ax3.get_lines()[0].set_markersize(7) # 设置点的大小
ax3.get_lines()[1].set_color('orangered') # 设置拟合线的颜色
ax3.get_lines()[1].set_linewidth(3.5) # 设置拟合线的粗细
ax3.get_lines()[1].set_linestyle('--')
ax3.get_lines()[1].set_alpha(0.8)
ax3.set_title('F({}, {}) Probability Plot with {} samples'.format(n1, n2, n), fontsize = 18)
ax3.set_xlabel('')
ax3.set_ylabel('')

# 經驗累積密度函數 Empirical CDF (ECDF)
res = cumfreq(x, n) # 生成 100 個區間的 ECDF

# 利用 min 和 max 間隔 res.cumcount.size 生成 x 軸
ecdf_x = res.lowerlimit + np.linspace(0, res.binsize * res.cumcount.size, res.cumcount.size)
```

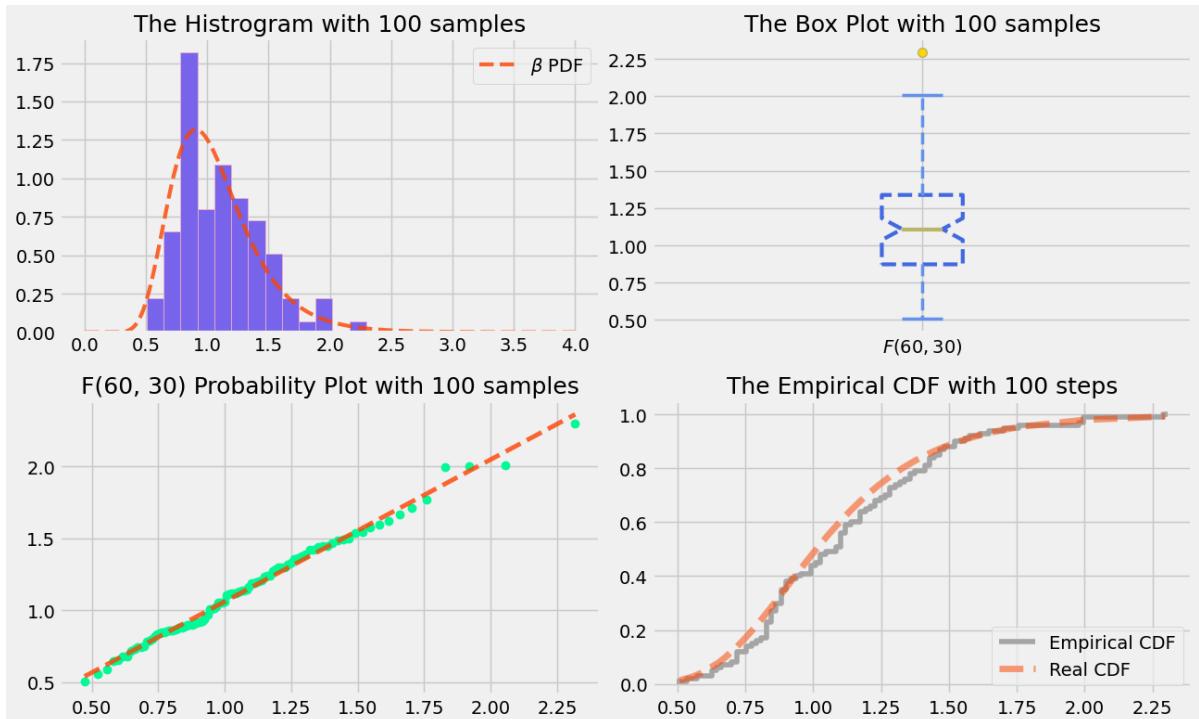
```

cumcountprob = res.cumcount / n # 累積次數除以總樣本數量 = 累積機率
ax4.plot(ecdf_x, cumcountprob, drawstyle = 'steps-pre', label = 'Empirical CDF'
          , color = 'darkgray')
ax4.set_title('The Empirical CDF with {} steps'.format(n), fontsize = 18)

# # draw the theoretical CDF of Z
x = np.sort(x)
cdf_y = f.cdf(x, dfn = n1, dfd = n2)
ax4.plot(x, cdf_y, color = 'orangered', linestyle = '--', linewidth = 5
          , alpha = 0.5, label = 'Real CDF')
ax4.legend(loc = 'lower right')

plt.tight_layout()
plt.show()

```



第二大圖：

```

In [17]: import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.stats import norm, chi2, t, beta, f
from scipy.stats import cumfreq

plt.style.use('fivethirtyeight')
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize = (13, 8))

# 參數設定
n, n1, n2 = 100000, 60, 30
x = f.rvs(dfn = n1, dfd = n2, size = n)

# 直方圖 Histogram
ax1.hist(x, bins = 35, density = True, color = 'mediumslateblue', edgecolor = 'lightblue')
# PDF
pdf_x = np.linspace(0, 4, 1000)
y = f.pdf(pdf_x, dfn = n1, dfd = n2)
ax1.plot(pdf_x, y, color = 'orangered', linestyle = '--', lw = 3, alpha = 0.8
          , label = r'$\beta$ PDF')
ax1.set_title('The Histogram with {} samples'.format(n), fontsize = 18)
ax1.legend()

```

```

# 盒鬚圖 BoxPlot
boxprops = dict(linestyle = '--', linewidth = 3, color = 'royalblue')
whiskerprops = dict(color = 'cornflowerblue', linewidth = 2.5, linestyle = '--')
capprops = dict(color = 'cornflowerblue', linewidth = 2.5)
flierprops = dict(marker = 'o', markerfacecolor = 'gold', markersize = 7
                  , linestyle = 'none', markeredgecolor = 'darkkhaki') # define outlier
medianprops = dict(color = 'darkkhaki', linewidth = 3, linestyle = '-') # 设置中位数
labels = ['$F(\{ \}, \{ \})$'.format(n1, n2)]
ax2.boxplot(x, notch = True, vert = True, boxprops = boxprops, flierprops = flierprops
            , whiskerprops = whiskerprops, capprops = capprops, medianprops = medianprops
            , labels = labels)
ax2.set_title('The Box Plot with {} samples'.format(n), fontsize = 18)

# Normal Probability plot
stats.probplot(x, dist = 'f', sparams = (n1, n2), plot = ax3)
# [0] 為 QQ plot 的點 · [1] 為 QQ plot 的線
ax3.get_lines()[0].set_marker('o')
ax3.get_lines()[0].set_markerfacecolor('mediumspringgreen') # 设置点的颜色
ax3.get_lines()[0].set_markeredgecolor('none') # 设置点的边缘颜色
ax3.get_lines()[0].set_markersize(7) # 设置点的大小
ax3.get_lines()[1].set_color('orangered') # 设置拟合线的颜色
ax3.get_lines()[1].set_linewidth(3.5) # 设置拟合线的粗细
ax3.get_lines()[1].set_linestyle('--')
ax3.get_lines()[1].set_alpha(0.8)
ax3.set_title('F(\{ \}, \{ \}) Probability Plot with {} samples'.format(n1, n2, n), fontsize = 18)
ax3.set_xlabel('')
ax3.set_ylabel('')

# 經驗累積密度函數 Empirical CDF (ECDF)
res = cumfreq(x, n) # 生成 100 個區間的 ECDF

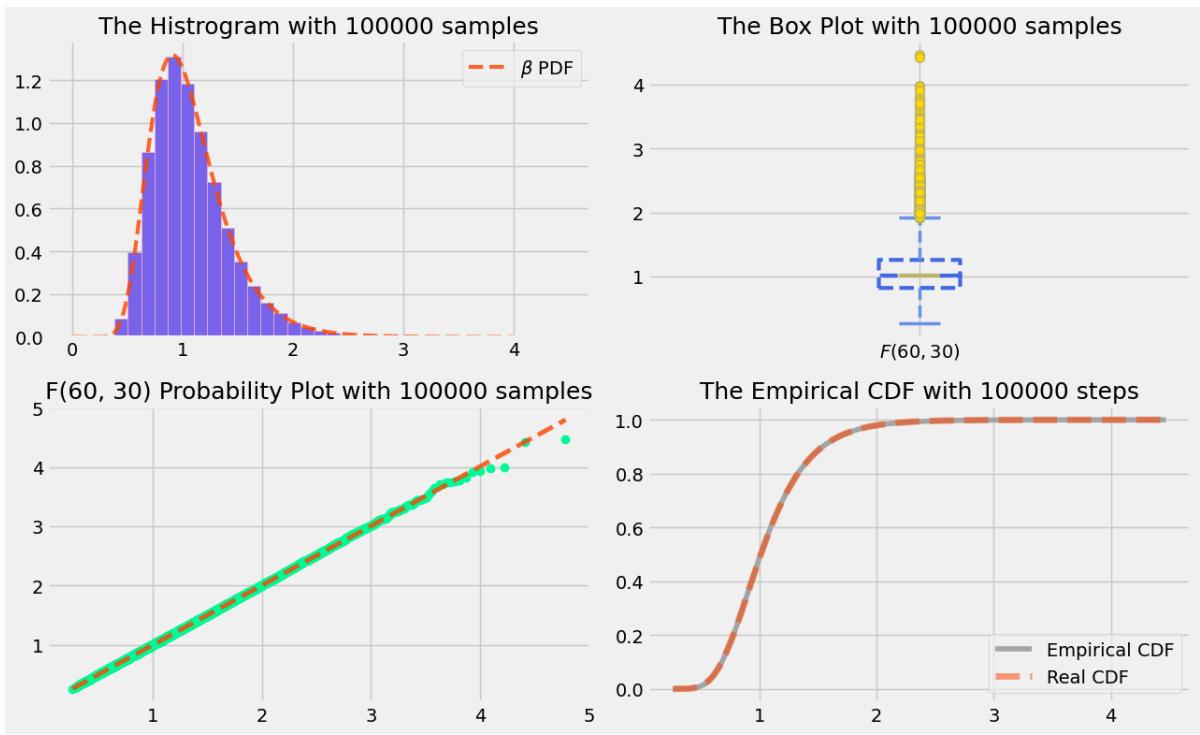
# 利用 min 和 max 間隔 res.cumcount.size 生成 x 軸
ecdf_x = res.lowerlimit + np.linspace(0, res.binsize * res.cumcount.size, res.cumcount.size)

cumcountprob = res.cumcount / n # 累積次數除以總樣本數量 = 累積機率
ax4.plot(ecdf_x, cumcountprob, drawstyle = 'steps-pre', label = 'Empirical CDF'
          , color = 'darkgray')
ax4.set_title('The Empirical CDF with {} steps'.format(n), fontsize = 18)

# # draw the theoretical CDF of Z
x = np.sort(x)
cdf_y = f.cdf(x, dfn = n1, dfd = n2)
ax4.plot(x, cdf_y, color = 'orangered', linestyle = '--', linewidth = 5
          , alpha = 0.5, label = 'Real CDF')
ax4.legend()

plt.tight_layout()
plt.show()

```



注意事項與討論：

觀察：

- 第一大圖：當樣本數為 100 時，無論是 Histogram、Boxplot、Probability plot 還是 Empirical CDF plot 的分佈情況都還無法很直觀地看出其母體分佈 $F(60, 30)$ 的模樣，但是已經可以大致判斷出它是個右偏的分佈。這表示樣本數還不夠多，或是所收集的樣本不具有很強的代表性，無法代表母體分佈，因此還需要更多的樣本。
- 第二大圖：當樣本數增加到 100000 時，可以觀察到無論是 Histogram、Boxplot、Probability plot 還是 Empirical CDF plot 的分佈情況皆呈現出右偏的趨勢，其中 Empirical CDF plot 則幾乎與真實的 CDF 分佈重疊在一起，表示樣本已具有足夠的代表性，已近似於母體分佈 $F(60, 30)$ 的模樣。

結論：樣本數越大，抽樣分佈將越接近母體分佈，故當收集到的樣本數過少時，除非樣本具有強烈的代表性，否則將只能繼續收集樣本，直到該樣本數足夠代表其母體分佈。
