機率分配與抽樣分配

郭翊萱 711133115

2022.11

統計學的基礎是機率，而我們需要更了解每個分配的基本性質才能更好的理解統計學，並做出正確的分析。因此在此文件中，我們首先將討論在改變參數時，幾種常見的離散分配與連續分配的機率密度圖與累積機率圖，接著，我們會利用抽樣分配來探討幾種分配之間的關係，並以直方圖、qqplot 圖、箱型圖與 edcf 圖進行呈現。最後，我們將以一個小專題來進行此節的收尾。

1 Discrete Distributions

在此節中，我們將介紹幾種常見的離散型分配，包含二項分配 (Binomial Distribution)、超幾何分配 (Hypergeometric Distribution)、幾何分配 (Geometric Distribution) 與卜瓦松分配 (Poisson Distribution)。首先我們先介紹二項分配。

1.1 Binomial Distribution

二項分配是 n 個獨立的試驗中成功次數的離散機率分布，每次成功的機率都設為 p，而一次成功的分配則稱為伯努利分配。以下我們透過改變二項分配的參數來繪製其機率分布圖。

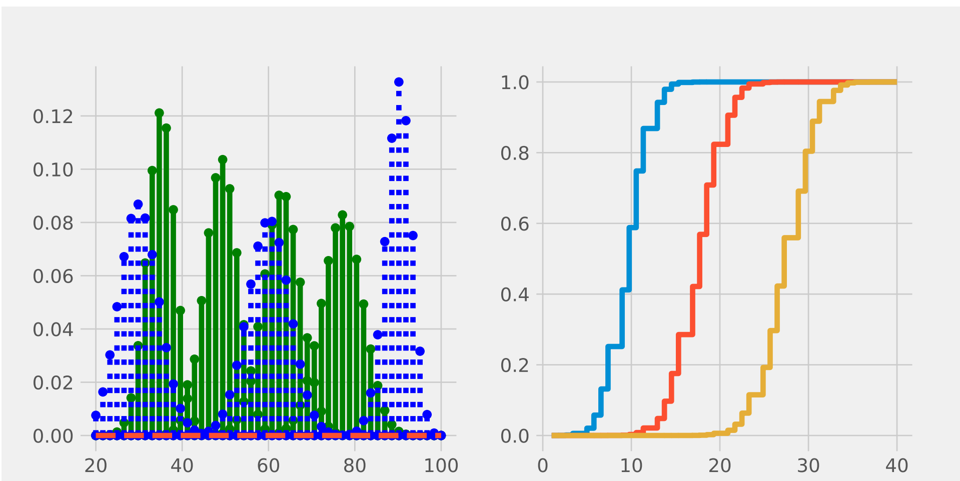

圖 1: cdf and pdf of Binomial Distribution

以上圖 1 即為二項分配的 pmf 圖與 cdf 圖。我們透過改變 n (綠色 stem 圖) 以及改變 p (藍色 stem 圖) 分別繪製二項分配的 pmf 圖。先觀察綠色的 pmf 圖，我們令 n = 50, 70, 90, 110、p 為 0.7 來繪製 pmf 圖，由圖 1 可知，隨著 n 值變大，整個圖型會往右偏移，且會逐漸趨近於鐘型分配，似乎與常態分配趨近。接著我們觀察藍色的 pmf 圖，我們令 n 為 100、p = 0.3, 0.6, 0.9 來繪製機率圖，由圖 1 同樣可以發現，隨著 p 增大，圖型也會逐漸向右移動，即期望值變大，且圖型的最高點也從 0.08 左右升至 0.12 左右。

```
n = np.arange(50, 120, 20)
p = 0.7
x = np.linspace(20, 100, 50)
for i in n:
    y = binom.pmf(x, i, p)
    axes[0].stem(x, y, linefmt='g-', markerfmt='o', basefmt =
       'C1--')
n = 100
p = np.arange(0.3, 0.9, 0.3)
for i in p:
    y = binom.pmf(x, n, i)
    axes[0].stem(x, y, linefmt='b:', markerfmt='o', basefmt =
       'C1--')
```

以上即為繪製 pmf 圖所需之程式碼，其中 linefmt 可以改變 stem 圖的線條型狀以及線條顏色，markerfmt 則可以改變端點的形狀。

透過圖 1 同樣可以觀察二項分配的 cdf 圖，我們繪製三組不同 (n, p) 情況下的 cdf 圖，包含 (20, 0.5), (30, 0.6) 以及 (40, 0.7)，隨著 n 與 p 的值增加，cdf 圖會整體向右移動。

Binomial Approximation

在此我們以二項分配 B(100, 0.1) 與常態分配 N(10, 3) 舉例，由圖 2 可知，二項分配在樣本數夠大且 $np > 5$ 時會趨近常態分配。

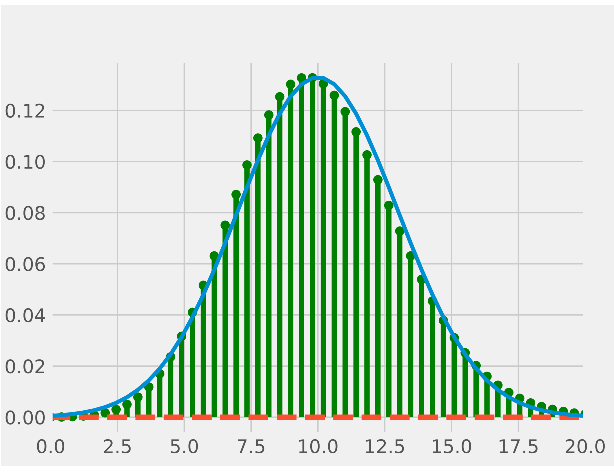

圖 2: Binomial Approximate to Normal Distribution

1.2 Hypergeometric Distribution

超幾何分配 (Hypergeometric Distribution) 是統計學上的一種離散型分配，代表從有限的 M 個物件中抽取 N 個物件，並成功從其中指定的 N 個物件中抽出 x 個物件的機率 (取出不放回)。

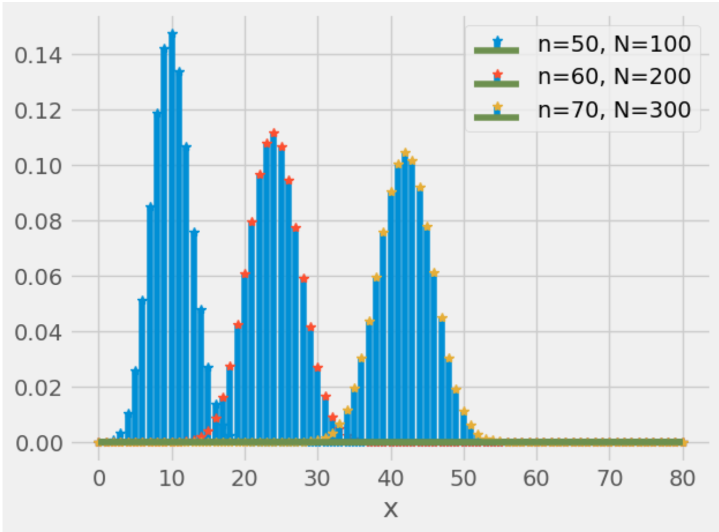

圖 3: Binomial Approximate to Normal Distribution

我們透過改變 n 以及 N 來繪製 pmf 圖進行觀察，透過圖 3 可知，隨著 n 與 N 的值增加，超幾何分配的圖形也會逐漸向右偏移，即期望值變大，而且圖型的最高點會逐漸變小，從 0.15 左右下降至 0.1 左右。

1.3 Geometric Distribution

幾何分配 (Geometric Distribution) 可以用兩者方法來進行解釋，第一種解釋是” 在伯努利試驗中，得到一次成功所需的試驗次數 (X)”，另一種意思則為” 在得到第一次成功之前所經歷的失敗次數 (Y=X-1)”。其 pmf 為:

$$P(X = k) = (1 - p)^{k-1}p \tag{1}$$

其中 $k = 1, 2, 3, ...$

我們透過變更參數 p 來觀察幾何分配的 pmf 圖，透過觀察圖 4 我們可知，我們設 p = 0.2, 0.4, 0.5, 0.7 來進行觀察，隨著 p 的值增大，pmf 圖的端點從 0.2 增至 0.7，其尾部也從遞減至 $x = 12$ 變成遞減至 $x = 6$ 左右。

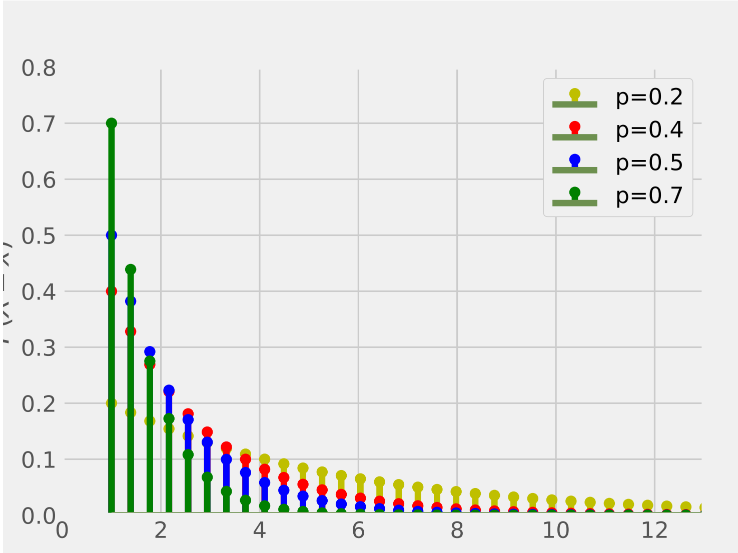

圖 4: Geometric Distribution

1.4 Poisson Distribution

卜瓦松分配 (Poisson Distribution) 是用於描述單位時間內隨機事件發生的次數的機率分配，其參數 λ 為隨機事件發生次數的期望值，其機率質量函數則為:

$$P(X=k)=\frac{e^{-\lambda}\lambda^{k}}{k!} \tag{2}$$

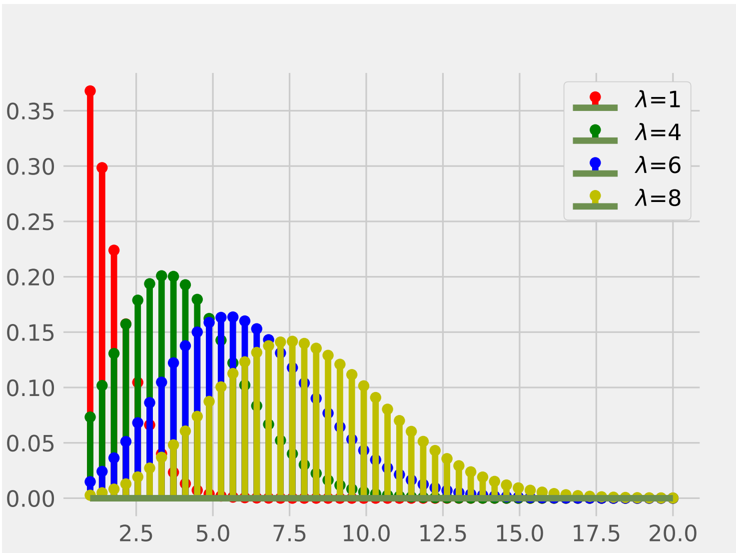

圖 5: Poisson Distribution

我們透過改變參數 λ 來觀察卜瓦松分配的 pmf 圖，由圖 5 可知，我們設定參數 $\lambda=1,4,6,8$，當 λ 較小時，卜瓦松分配為右偏分配，而當 λ 較大時，卜瓦松分配則為鐘型分配，可能趨近常態分配。

Binomial Approximate to Poisson Distribution

在數理統計課程中，我們可知當 n 趨近於無限大、p 趨近於 0 時二項分配會趨近於卜瓦松分配，因此在此處我們透過改變二項分配的參數 n, p 以及卜瓦松分配的參數 λ 來觀察此趨近特性。我們設置 $\lambda = np$。

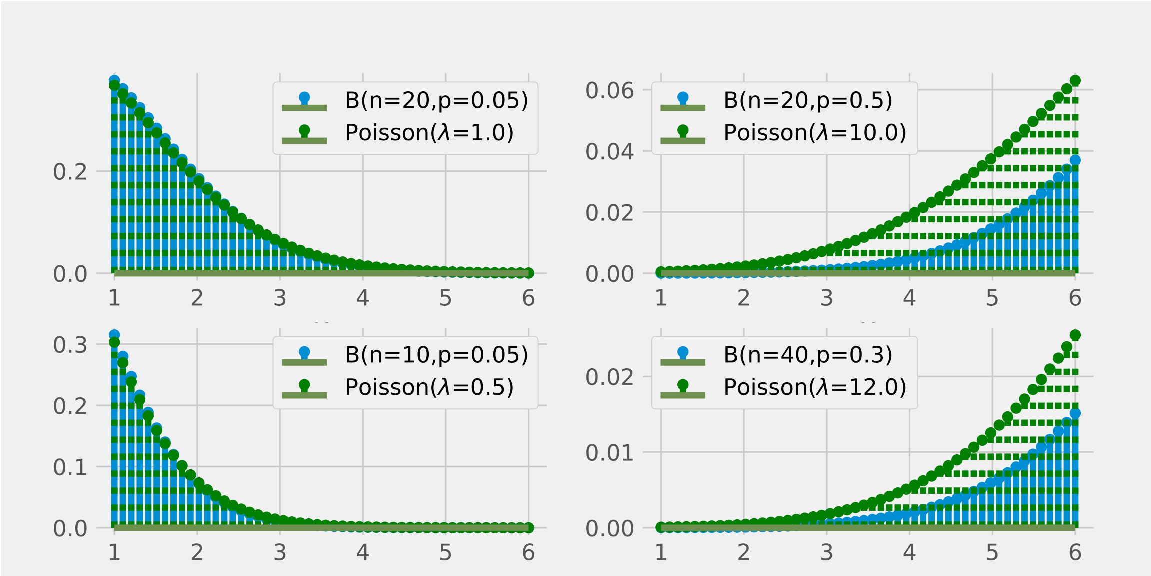

圖 6: Binomial approximate to Poisson Distribution

由圖 6 的左上圖與右上圖可知，當 $n = 20$、$p = 0.05$ 時，二項分配趨近於卜瓦松分配，但當 $n = 20$、$p = 0.5$ 時，二項分配不會趨近於卜瓦松分配。

接著，根據圖 6 的左下圖與右下圖可知，當 n 夠大且 p 夠小時二項分配會趨近於卜瓦松分配，而當 n 夠大但 p 不夠小時，二項分配不會趨近於卜瓦松分配。下面即為部分程式碼示例。

```
fig, axes = plt.subplots(2, 2, figsize=(10, 5))
x = np.linspace(1, 6, 50)

n, p = 10, 0.05
lamb = n*p
y = binom.pmf(x, n, p)
axes[1][0].stem(x, y, label="B(n={},p={})".format(n,p))
y1 = poisson.pmf(x, lamb)
axes[1][0].stem(x, y1, linefmt='g:', label="Poisson($\lambda
   $={})".format(lamb))
axes[1][1].set_xlabel("x")

plt.savefig(img_dir+"binom-poisson.eps", format="eps")
plt.show()
```

Poisson Distribution Addiction

在數理統計中，我們同樣學到當 $X \sim Poisson(\lambda_1)$、$X \sim Poisson(\lambda_2)$ 時，$X+Y \sim Poisson(\lambda_1+\lambda_2)$，即卜瓦松分配具有加成性。因此在此處，我們將利用亂數產生 X 與 Y 的卜瓦松分配隨機變數，並與 $X+Y \sim Poisson(\lambda_1+\lambda_2)$ 的 pmf 圖進行比較。在此處中，我們繪製直方圖 (Histogram)、箱型圖 (Boxplot)、常態機率圖與 Empirical CDF 圖來進行觀察，並觀察如果更改亂數產生的樣本大小 (n) 是否會使結果發生改變。

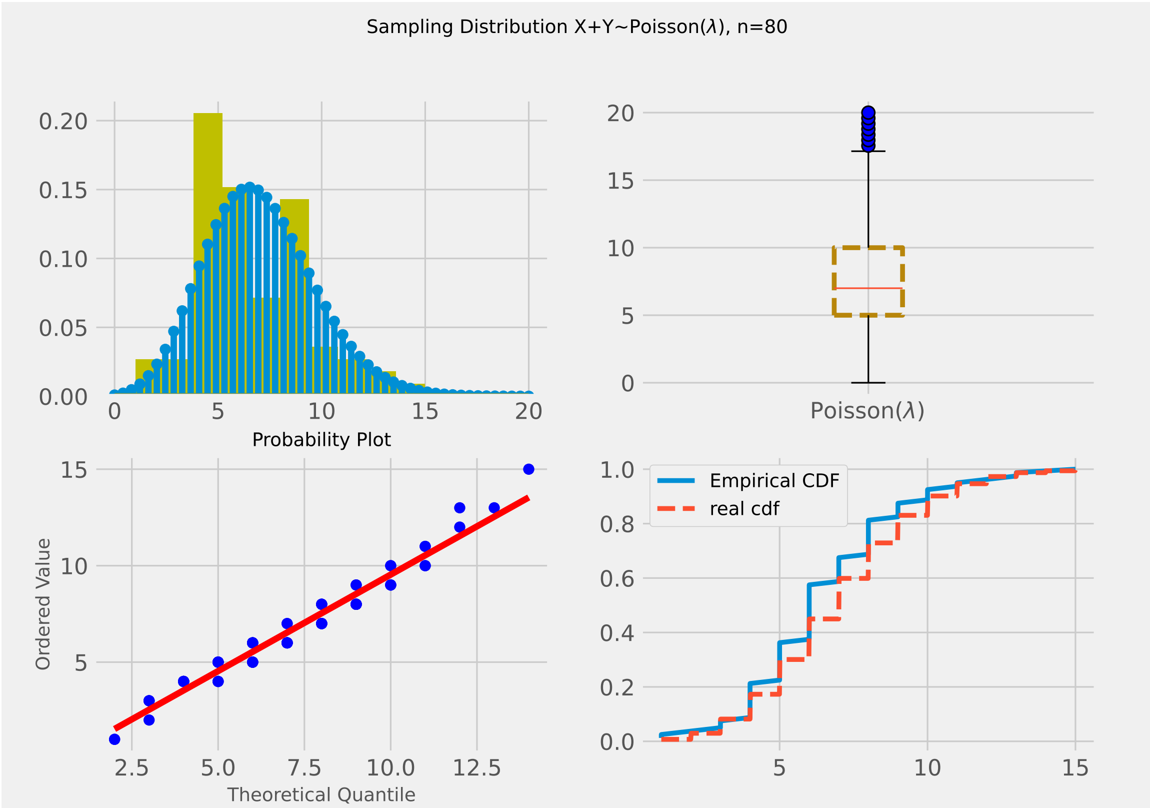

圖 7: $X+Y \sim Poisson(\lambda_1+\lambda_2), n = 80$

我們設定 $\lambda = 3$ 以及 $\lambda = 4$ 來繪製圖形，圖 7 即為利用亂數產生圖形的結果。由此圖可知，雖然 qqplot 圖基本上貼合 $Poisson(\lambda_1+\lambda_2)$ 的 pmf 圖，但從其直方圖與 ecdf 圖卻可以看出，似乎亂數產生的分配的 ecdf 與 $Poisson(\lambda_1+\lambda_2)$ 的 cdf 圖並未完全貼合。以下為部分繪圖程式碼。

```
np.random.seed(seed=1294) #設定種子

n = 1000
lamb1 = 3
x1 = poisson.rvs(lamb1, size = n)
lamb2 = 4
x2 = poisson.rvs(lamb2, size = n)

##兩個亂數抽樣分配相加
```

```
x4 = x1+x2
bins = 10
axes[0][0].hist(x4, density=True, bins = bins, alpha=0.5,
   color="y")

###Poisson分配(lamb1+lamb2)
x3 = np.linspace(0, 20, 50)
lamb3 = lamb1+lamb2
y = poisson.pmf(x3, lamb3)
axes[0][0].stem(x3, y)

##ECDF##
x_sort = np.sort(x4)
F = np.arange(1 ,n+1) / n
axes[1][1].plot(x_sort, F, lw =3, label = "Empirical CDF")

X = np.linspace(x_sort[0], x_sort[-1], 1000)
y = poisson.cdf(X, lamb3)
axes[1][1].plot(X, y, linestyle="--", lw = 3, label = "real
   cdf")
```

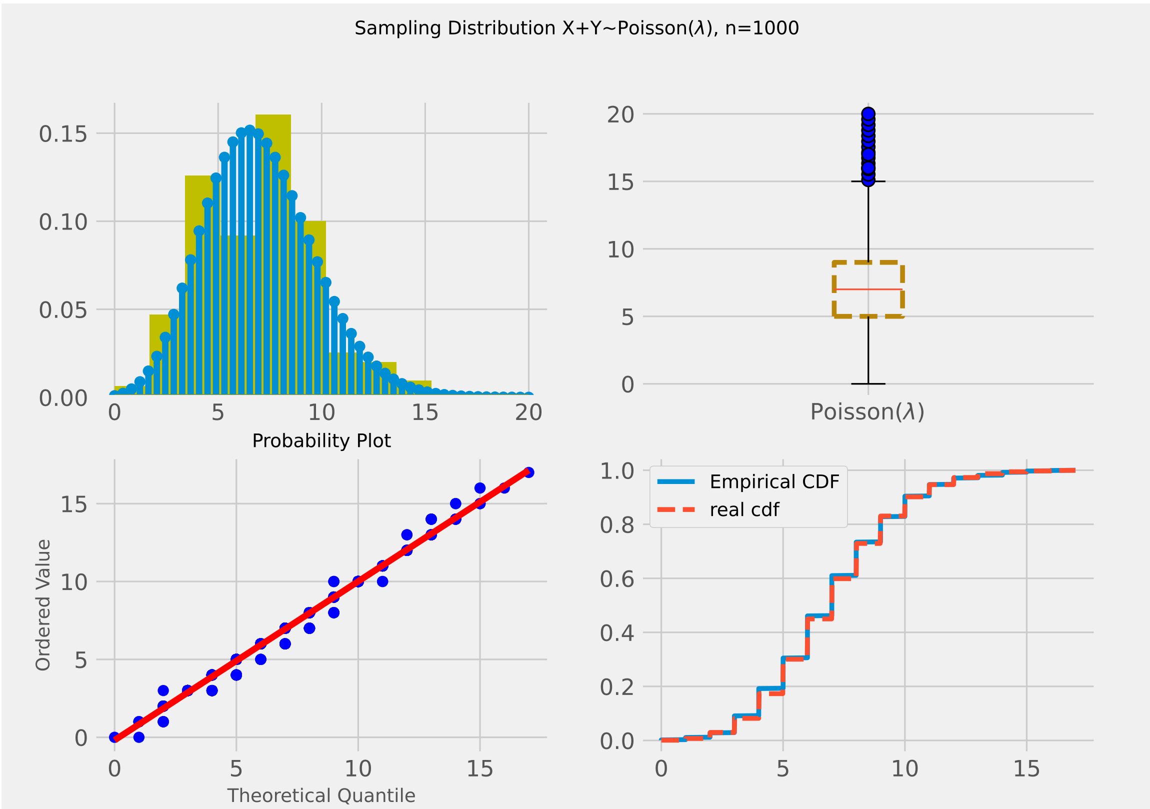

圖 8: $X + Y \sim Poisson(\lambda_1 + \lambda_2), n = 1000$

在圖 8 中我們可以看出，與 $n = 80$ 時不同的是，此時的 qqplot 圖完全貼合，且 ecdf 圖與真實的 $Poisson(\lambda_1 + \lambda_2)$ 也已完全貼近，由此可證。

Sampling Poisson Distribution

以下我們利用亂數抽取卜瓦松分配並繪製直方圖與 qqplot 圖，並觀察其與常態分配的差異以及樣本數改變可能導致的差異。在此我們以 Normal(μ, σ) 作為比較，觀察 Poisson(2), Poisson(5) 與 Poisson(20) 三個分配。我們首先觀察在樣本數 n=20, n=100, n=1000 時的直方圖。

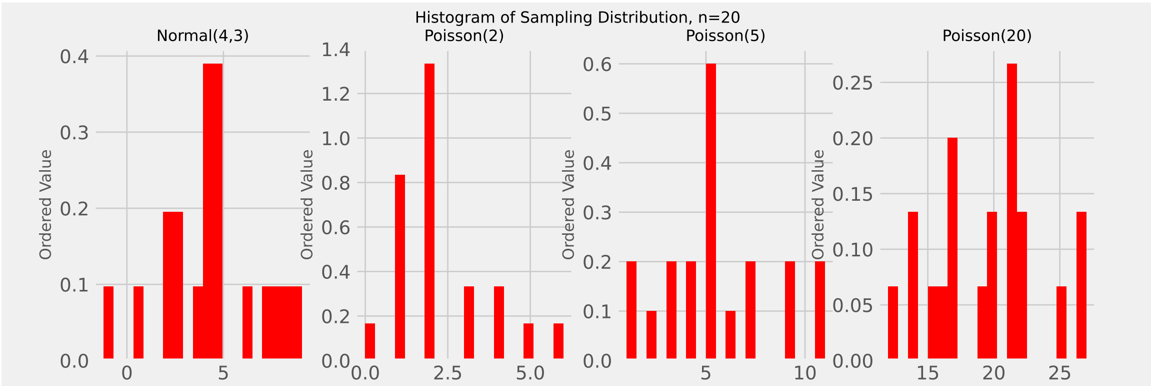

圖 9: Sampling Poisson Distribution, n=20

從圖 9 可知，在樣本數為 20 時，從直方圖中似乎無法觀察出甚麼特別的性質。接著，我們將樣本數增加至 $n = 100$，甚至是 $n = 1000$，如下圖 10 與 11。

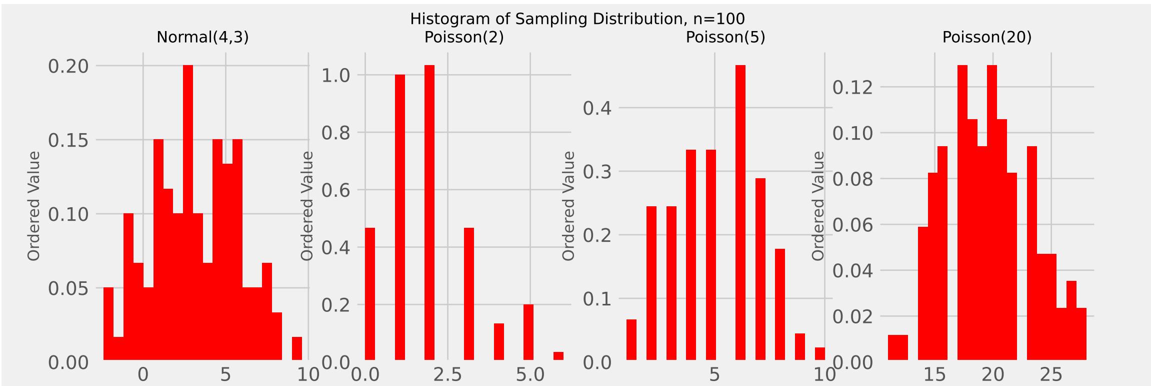

圖 10: Sampling Poisson Distribution, n=100

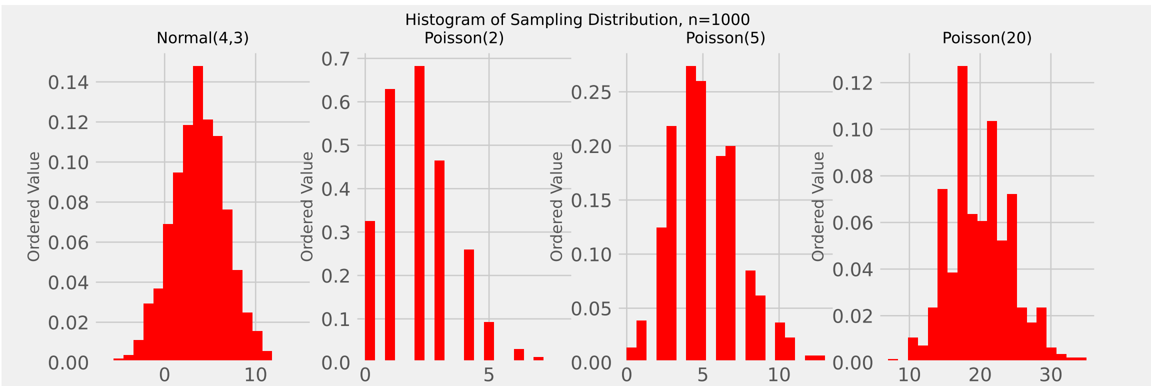

圖 11: Sampling Poisson Distribution, n=1000

根據上圖 10 與 11 我們可以發現，隨著樣本數變大以及卜瓦松分配的參數值變大，卜瓦松分配的直方圖會越來越趨近於鐘型分配的形狀。接著我們來觀察不同樣本數下卜瓦松分配的 qqplot 圖。

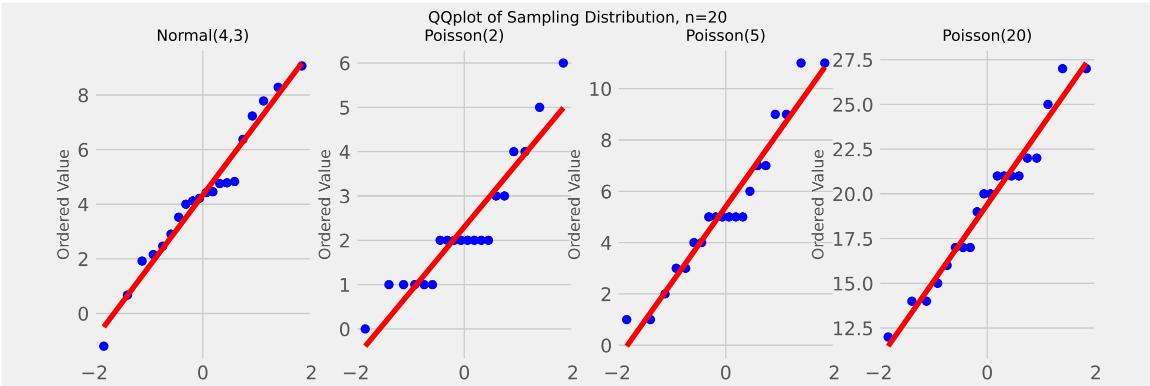

圖 12: Sampling Poisson Distribution, n=20

由圖 12 可觀察在樣本數為 20 的情況下，卜瓦松分配與常態分配的 qqplot 圖，在此圖中仍然可以觀察到卜瓦松分配的離散性質。接著，我們將樣本數增加至 n = 100 甚至是 n = 1000 再重新繪製 qqplot 圖，如圖 13 與 14。

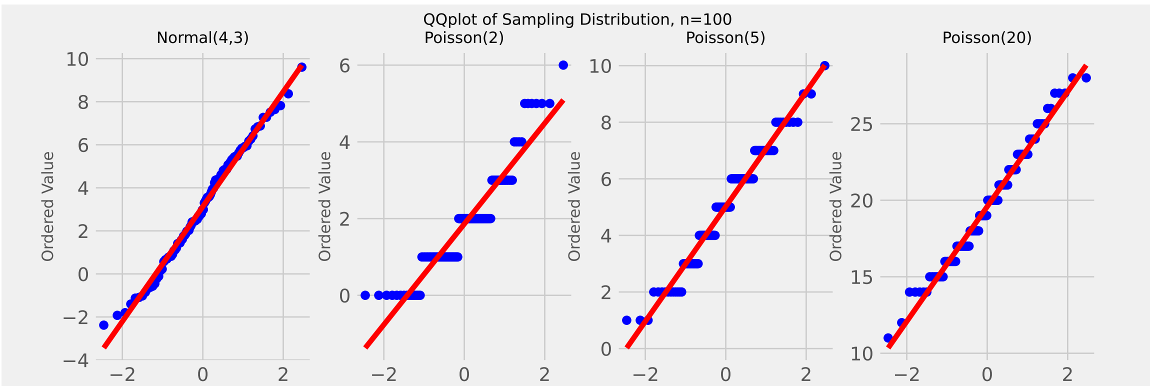

圖 13: Sampling Poisson Distribution, n=100

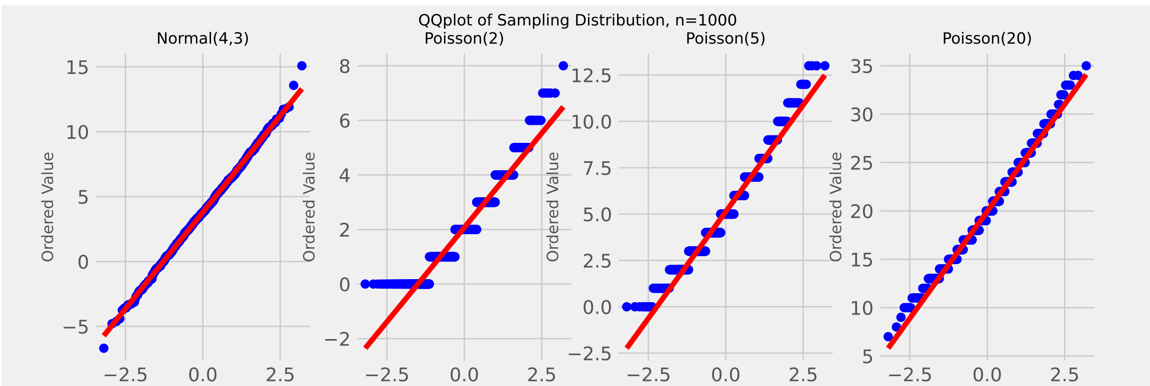

圖 14: Sampling Poisson Distribution, n=1000

由上圖 13 與 14 可觀察到，隨著樣本數增大，卜瓦松分配會與常態分配越來越貼近，此觀察到的結果與我們在數理統計課程中所了解到的性質一致。以下我們簡略呈現部分程式碼:

```
mu = 4
sigma = 3
n = 1000
rv_norm = norm.rvs(loc = mu, scale = sigma, size = n)
lamb1 = 2
lamb2 = 5
lamb3 = 20
rv_poi1 = poisson.rvs(lamb1, size = n)
rv_poi2 = poisson.rvs(lamb2, size = n)
rv_poi3 = poisson.rvs(lamb3, size = n)
ax1.hist(rv_norm, bins = bins, density=True, color="r", alpha
   = 0.5)
stats.probplot(rv_poi1, dist = "norm", plot = ax2)
```

2 Continuous Distributions

2.1 Chi-squared Distribution

k 個獨立的標準常態分配變數的平方和服從自由度為 k 卡方分配，卡方分配也是一種特殊的伽瑪分配。其機率密度函數為:

$$f_k(x) = \frac{1}{2^{\frac{k}{2}}\Gamma(\frac{k}{2})} 2^{\frac{k}{2}-1} e^{\frac{-x}{2}} \tag{3}$$

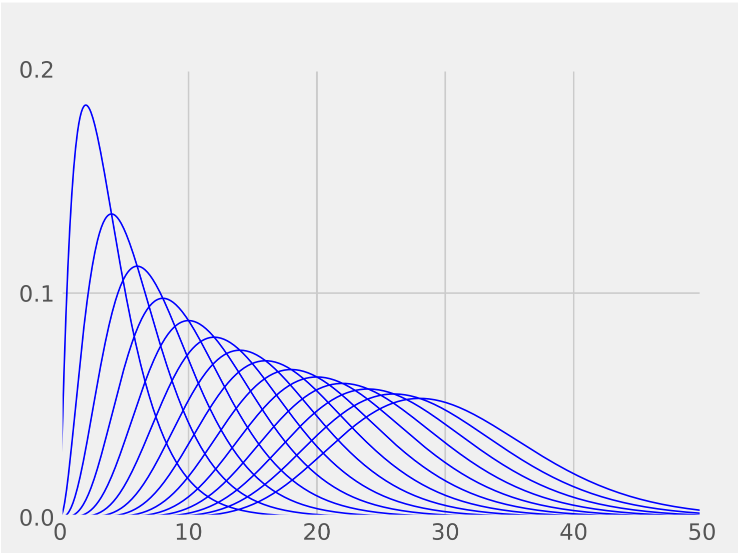

圖 15: Chi-squared Distribution

我們透過改變卡方分配的參數 df 來觀察卡方分配的性質。我們設置參數 $df = 4, 6, 8, ..., 32$，由圖 15 可知，隨著參數的值變大，卡方分配會漸漸從右偏分配變成趨近於常態分配。以下是部分的程式碼:

```
img_dir = "D:/vscodepython/Statistical Calculation/Homework3_
   Distribution/image_hw3/"
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', alpha=0.4)

plt.yticks([0, 0.1, 0.2])
plt.savefig(img_dir+"chi-squared.eps", format="eps")
plt.show()
```

Chi-squared Approximate to Normal and Gamma Distribution

我們透過繪製卡方分配 $\chi^2(1000)$ 與常態分配 Normal(μ,σ) 的 pdf 圖，以及繪製卡方分配 $\chi^2(10)$ 與伽瑪分配 $\Gamma(\frac{10}{2}, \beta)$ 的 pdf 圖來更好的理解卡方分配與常態分配的關係，並驗證若卡方分配的參數為 v，則此卡方分配與伽瑪分配 $\Gamma(\frac{v}{2}, \beta)$ 相等。

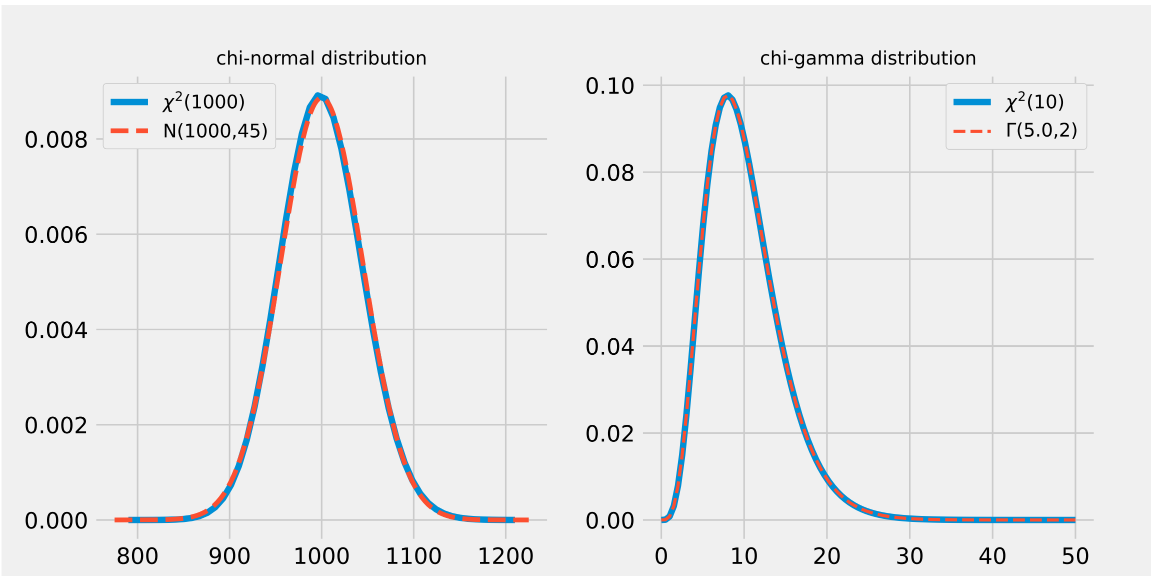

圖 16: Chi-squared Distribution Relationship

由上圖 16 可知，隨著卡方分配的參數增大，卡方分配會由右偏分配變成趨近於鐘型分配，且由左圖可知，卡方分配與常態分配的 pdf 圖相同，期望值為 1000。另外，由其右圖可知，卡方分配 $\chi^2(v)$ 與 $\Gamma(\frac{v}{2}, \beta)$ 確實相等。其部分相關程式碼如下:

```
####卡方分配與常態分配####
df = 1000
x = np.linspace(790, 1210)
y = chi2.pdf(x.reshape(-1,1), df=df)
mu = 1000
sigma = 45
X = np.linspace(mu-5*sigma, mu+5*sigma, 500)
Y = norm.pdf(X, loc=mu, scale=sigma)
axes[0].plot(X, Y, lw=3,linestyle="--", label="Normal
   Distribution")
axes[0].set_title("chi-normal distribution", fontsize = 'small
   ')
axes[0].legend(fontsize = 'small', loc="upper left")

####卡方分配與伽瑪分配####
v = 10
df = v/2
x = np.linspace(0, 50, 100)
y1 = chi2.pdf(x.reshape(-1,1), df = v)
y2 = gamma.pdf(x, a = df, scale = 2)
```

Sampling Chi-squared Distribution

接著我們在此利用亂數產生服從常態分配的隨機變數，並期望證明標準常態分配的平方 Z^2 與 $\chi^2(1)$ 相等。我們透過繪製直方圖、箱型圖、qqplot 圖與 ecdf 圖來觀察究竟兩分配是否相同。另外，我們亦透過改變生成亂數的樣本大小 n 來觀察是否樣本數大小會造成兩分配的差異。

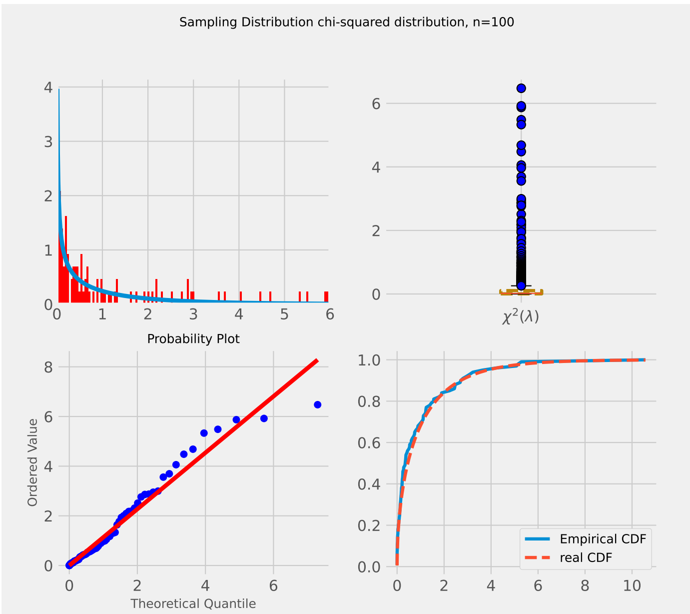

圖 17: Chi-squared Distribution Relationship, n=100

根據圖 17 我們可以明顯發現，當樣本數達到 100 筆時，亂數產生的常態隨機變數的機率圖與卡方分配 $\chi^2(1)$ 並未完全貼合，無法證明標準常態分配的平方 Z^2 與 $\chi^2(1)$ 相等。因此我們將樣本數增大至 10000 筆並重新觀察直方圖、箱型圖、qqplot 圖與 ecdf 圖，結果如圖 18。以下我們呈現部分繪圖之程式碼。

```
n = 100
x = norm.rvs(loc = 0, scale = 1, size = n)
x = x**2
bins = 150

x_z = np.linspace(0, 10, 1000)
z = chi2.pdf(x_z, df = 1)

####boxplot####
boxprops = dict(linestyle = '--', linewidth = 3, color = '
    darkgoldenrod')
flierprops = dict(marker='o', markerfacecolor = 'blue',
    markersize = 8, linestyle = 'none')
labels = ["$\chi^2(\lambda)$"]
axes[0][1].boxplot(np.r_[x, z], boxprops = boxprops,
    flierprops = flierprops, labels = labels)
####QQplot####
stats.probplot(x, dist = "chi2", sparams=(df), plot=axes
    [1][0])
```

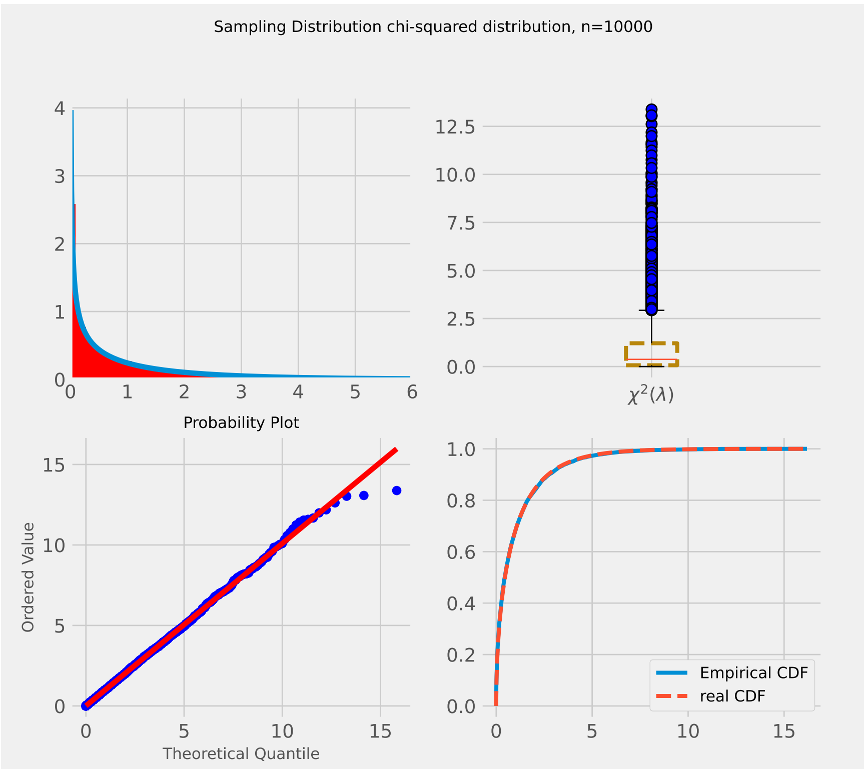

圖 18: Chi-squared Distribution Relationship, n=10000

由圖 18 可知，當樣本數由 $n = 100$ 增加至 $n = 10000$ 筆時，標準常態分配的平方 Z^2 與 $\chi^2(1)$ 相等。

2.2 Exponential Distribution

指數分配 (Exponential Distribution) 是一種連續機率分配，其用來表示獨立隨機事件發生所需的時間間隔。指數分配的機率密度函數為:

$$f(x; \lambda) = \lambda e^{-\lambda x}, x \geq 0 \tag{4}$$

其中 λ 為分配的母數，即代表每單位時間發生數件的次數，β 為比例母數，即代表該事件在每單位時間的發生率，可以利用 $\lambda = \frac{1}{\beta}$ 來代替 λ。指數分配的期望值是 $\frac{1}{\lambda}$，變異數則為 $\frac{1}{\lambda^2}$。接著我們透過改變參數 λ 來觀察指數分配的 pdf 圖與 cdf 圖。

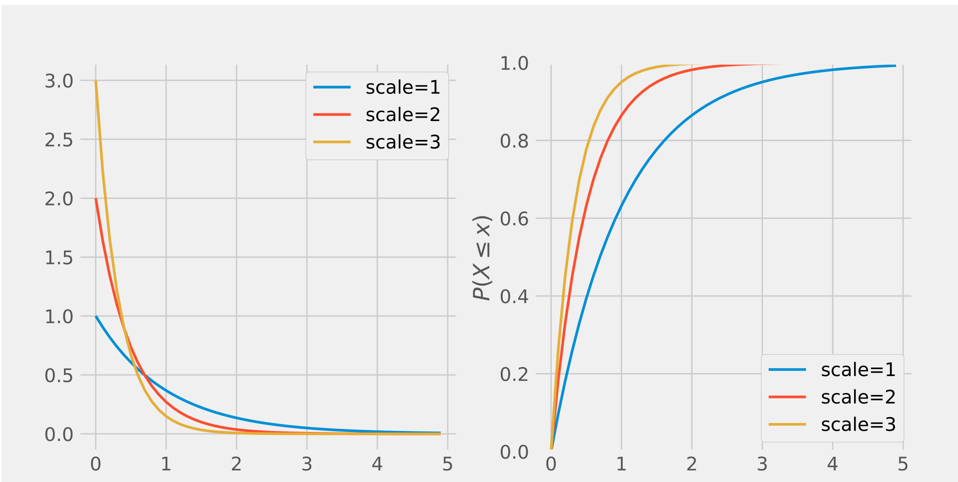

圖 19: Exponential Distribution

我們設定參數 $\lambda = 1, 2, 3$，根據圖 19 可知，隨著參數 λ 的值增大，單位時間發生事件次數是 0 的次數變大，且 pdf 圖遞減的速度加快。以下我們呈現部分的程式碼:

```
x1 = np.arange(0, 5, 0.1) #(15,)
scale = np.arange(1, 4) #(4,)
for i in scale:
    y1 = i * np.exp(-i*x1)
    axes[0].plot(x1, y1, lw=2, label="scale={}".format(i))
```

在此題繪圖時遇到一些問題。若我們使用 *expon.pdf* 來生成 pdf 圖，則繪製出的圖形會有點怪異，如下圖 20，但目前尚未找到原因。部分程式碼亦呈現於下。

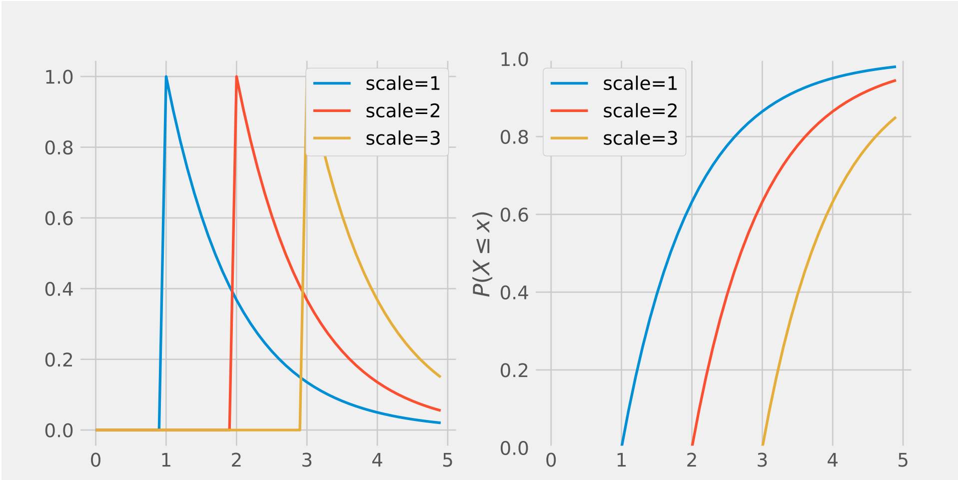

圖 20: Exponential Distribution1

```
x1 = np.arange(0, 5, 0.1) #(15,)
scale = np.arange(1, 4) #(4,)
for i in scale:
    y1 = expon.pdf(x1, i)
    axes[0].plot(x1, y1, lw=2, label="scale={}".format(i))
```

2.3 Double Exponential Distribution

雙指數分配 (Double Exponential Distribution)，亦稱為拉普拉斯分配 (Laplace Distribution)，此分配可以看作兩個平移指數分配背靠背拼接在一起，其機率密度函數為:

$$f(x|\mu, b) = \frac{1}{2b} exp(-\frac{|x-\mu|}{b}) \tag{5}$$

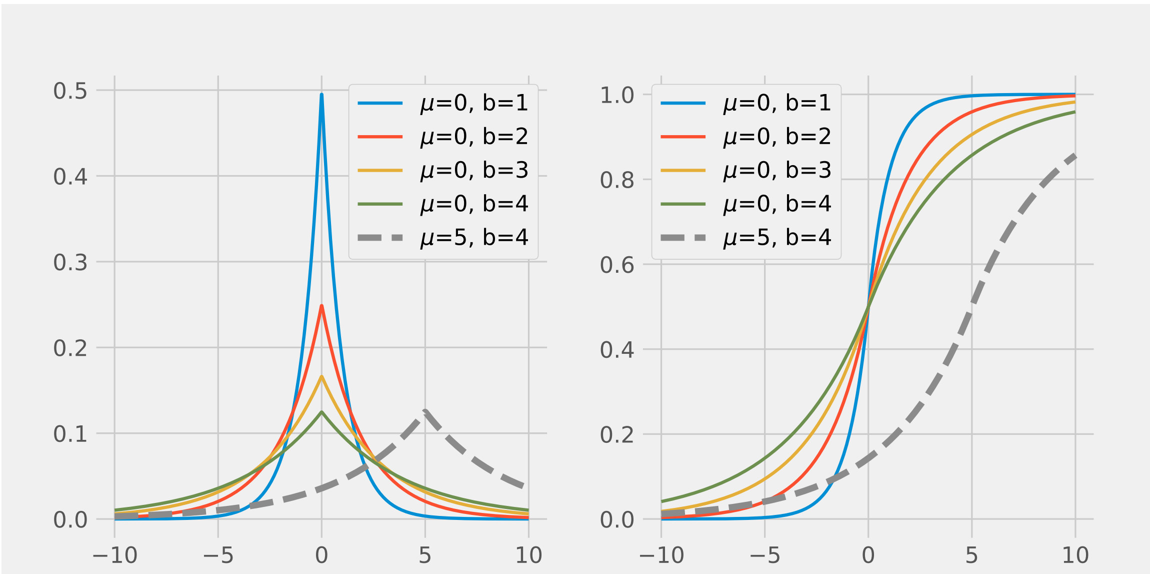

圖 21: Laplace Distribution

我們用函數 $laplace.pdf$ 進行繪圖。由圖 21 可知，當參數 b 從 1 增加到 4 時，整個 pdf 圖會逐漸變寬，變異數變大且端點的值從 0.5 左右下降至 0.1 左右。μ 的改變則會讓圖形產生左移或右移，若 μ 值變大會往右移，變小則往左移。

2.4 Gamma Distribution

假設 $X_1, X_2, ..X_n$ 為連續發生事件的等候時間，且這 n 次等候時間相互獨立，則 $Y = X_1 + X_2 + ... + X_n$ 服從伽瑪分配 (Gamma(α, β))，其中 $\alpha = n$，而 β 與 λ 互為倒數，λ 代表單位時間內事件的發生率。其機率密度函數為:

$$f(x) = \frac{x^{\alpha-1}\lambda^{-\alpha}e^{(-\lambda x)}}{\Gamma(\alpha)}, x > 0 \tag{6}$$

```
x = np.linspace(0, 20, 1000)
k = np.arange(1, 13, 1.5)# 0 1.5 3 4.5 6 7.5
theta = np.arange(1, 5, 0.5) #0 0.5 1 1.5 2 2.5
param = np.vstack((k,theta))#(2,6)#矩陣
param = param.T #(6,2)
for i in range(8):
     y = gamma.pdf(x, param[i][0], param[i][1])
     plt.plot(x, y, lw=2, c="blue", alpha=0.5)
```

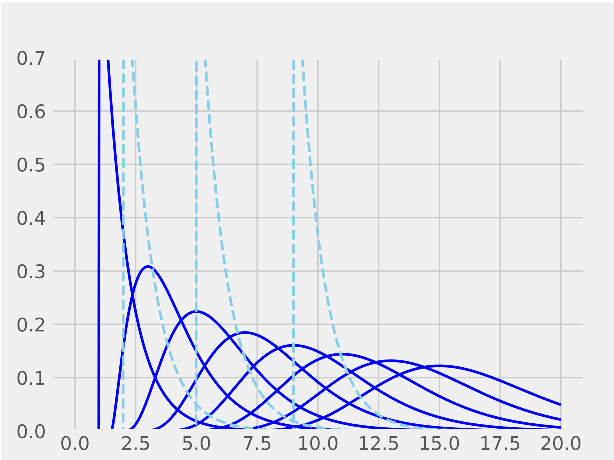

圖 22: Gamma Distribution

以上即為繪製伽瑪分配 pdf 圖與 cdf 圖的程式碼與結果。我們設置其參數 $\alpha = 1, 2.5, 4, 5.5, 7, 8.5, 10, 11.5$ 且 $\beta = 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5$ 共八組不同參數畫出深藍色的 pdf 圖，接著我們設置參數 $(\alpha, \beta) = (1, 2), (1, 5), (1, 9)$ 來觀察改變參數 β 造成的圖形變化。根據圖 22，隨著 α 值與 β 值變大，伽瑪分配會逐漸從右偏分配變成鐘型分配，此性質與卡方分配相近，而若我們僅僅增大參數 β 的值，分配則會逐漸向右偏移。

Gamma Distribution Addiction

在了解了伽瑪分配的基本性質後，我們接著利用亂數產生伽瑪分配來嘗試證明伽瑪分配的可加性，即當 $X \sim \Gamma(\alpha_1, \beta_1)$、$Y \sim \Gamma(\alpha_2, \beta_2)$ 時，$X + Y \sim \Gamma(\alpha_1 + \alpha_2, \beta_1 + \beta_2)$。我們將亂數產生 X, Y 兩個獨立分配並與 X+Y 分配共同繪製直方圖、qqplot 圖與 ecdf 圖，並更改樣本數來觀察樣本數可能造成的差異。

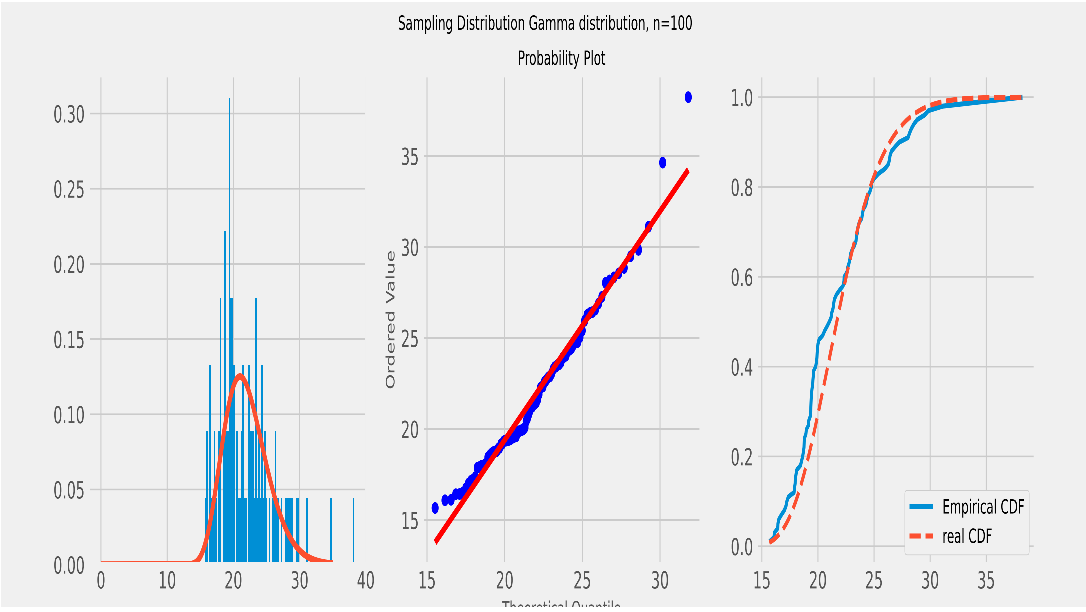

圖 23: Gamma Distribution Addiction, n=100

上圖 23 即為當樣本數為 100 時，抽樣分配 $X \sim \Gamma(1, 10), Y \sim \Gamma(10, 1)$ 與 $X + Y \sim \Gamma(1 + 10, 10 + 1)$ 的擬合結果。由圖可知，三種圖形得到的結果都是兩者並不是非常貼近，無法證明伽瑪分配的可加性，因此我們嘗試將樣本數提高到 n = 1000 再重新繪製三種圖形。以下提供部分程式碼:

```
alpha1 = 1
beta1 = 10
beta2 = 1
alpha2 = 10
n = 100
x1 = gamma.rvs(alpha1, beta1, size=n)
x2 = gamma.rvs(alpha2, beta2, size=n)
x1 = np.sort(x1)
x2 = np.sort(x2)
x3 = x1+x2
axes[0].hist(x3, bins=100, density=True)
x4 = np.linspace(0, 35, 1000)
y = gamma.pdf(x4, alpha1+alpha2, beta1+beta2)
axes[0].plot(x4, y)
####ecdf####
x_sort = np.sort(x3)
```

```
F = np.arange(1 ,n+1) / n
axes[2].plot(x_sort, F, lw =3, label="Empirical CDF")
X = np.linspace(x_sort[0], x_sort[-1], 1000)
y = gamma.cdf(X, alpha1+alpha2, beta1+beta2)
axes[2].plot(X, y, linestyle="--", lw = 3, label="real CDF")
```

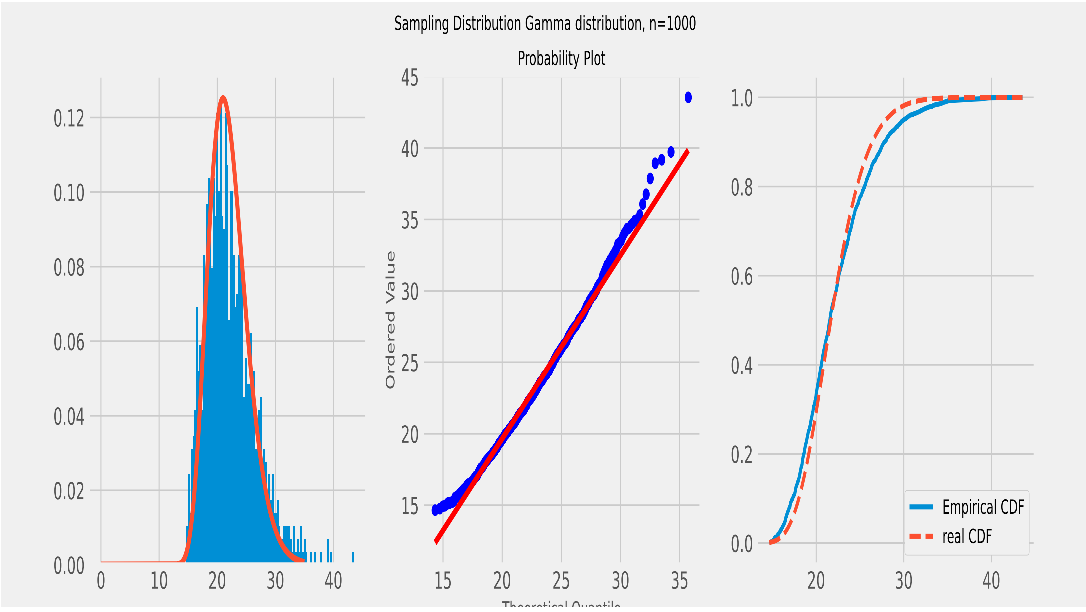

圖 24: Gamma Distribution Addiction, n=1000

由圖 24 可知，亂數產生的 $X \sim \Gamma(\alpha_1, \beta_1)$、$Y \sim \Gamma(\alpha_2, \beta_2)$ 與分配 $X + Y \sim \Gamma(\alpha_1 + \alpha_2, \beta_1 + \beta_2)$ 已經幾乎擬合，由此可證伽瑪分配的可加性。

Sampling Gamma Distribution

以下我們亂數抽取服從伽瑪分配的隨機變數並繪製直方圖與 qqplot 圖，並觀察其與常態分配的差異以及改變樣本數可能導致的差異。我們將觀察 Norma(4,3) 與 $\Gamma(1, 31)$、$\Gamma(31, 1)$ 與 $\Gamma(50, 50)$ 三個分配在樣本數 n=50, n=1000 時的圖形。

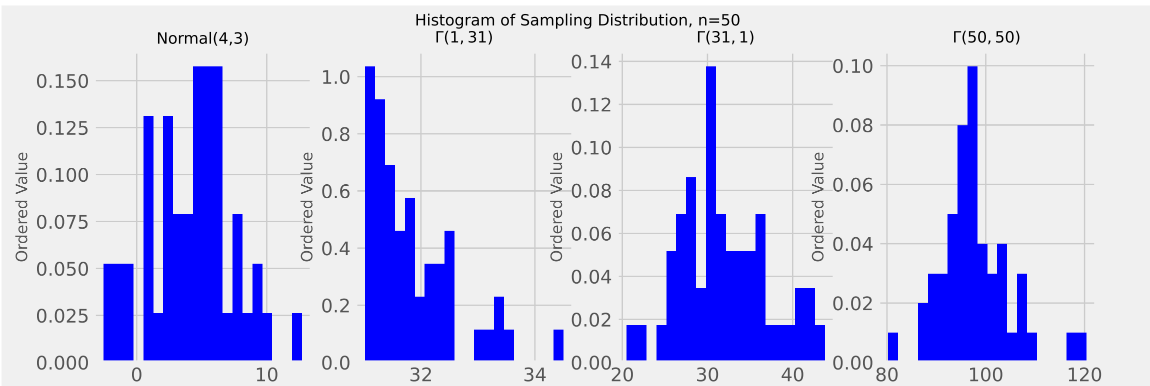

圖 25: Histogram of Sampling Gamma Distribution, n=50

圖 25 呈現了在樣本數 n=50 時的直方圖，從此直方圖可以看出伽瑪分配的右偏與左偏等特性。接著我們將抽取的樣本數增加至 n=1000，可得圖 26。

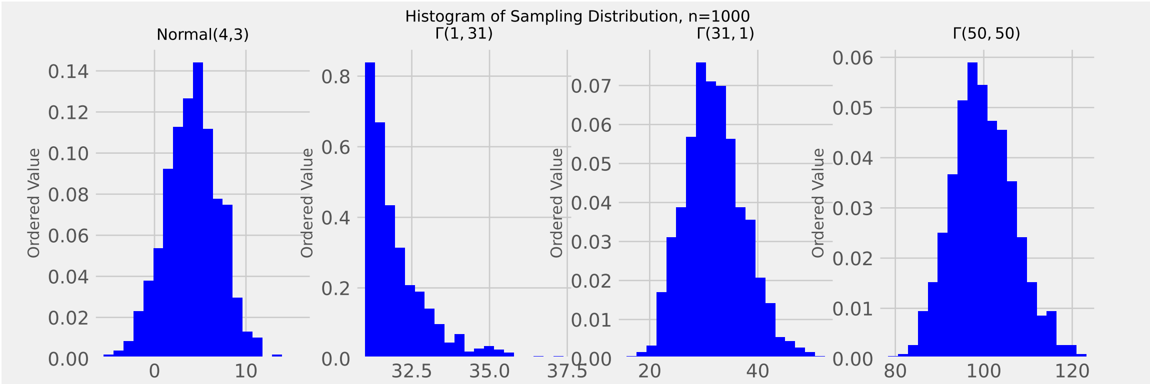

圖 26: Histogram of Sampling Gamma Distribution, n=1000

從圖 26 我們可以看出，當樣本數增加至 n=1000 時，$\Gamma(50, 50)$、$\Gamma(31, 1)$ 與常態分配的形狀趨近。接著我們來觀察 Norma(4,3) 與 $\Gamma(1, 31)$、$\Gamma(31, 1)$ 與 $\Gamma(50, 50)$ 三個分配的 qqplot 圖。

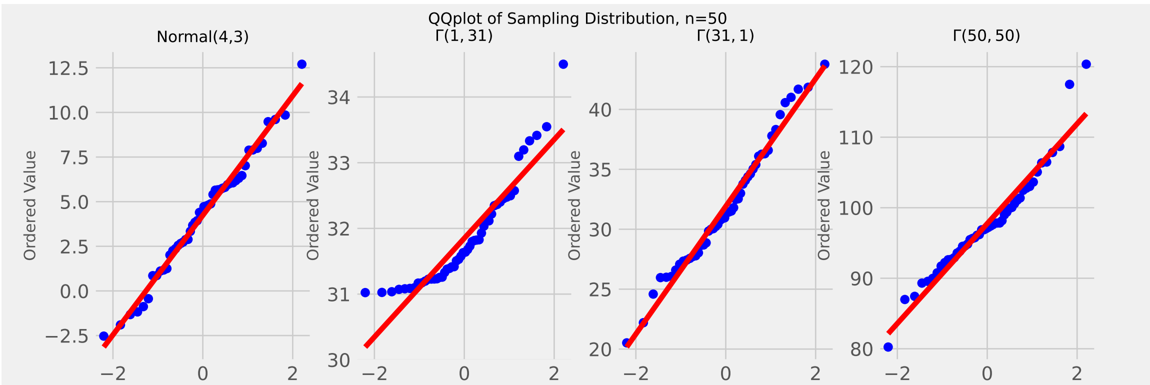

圖 27: QQplot of Sampling Gamma Distribution, n=50

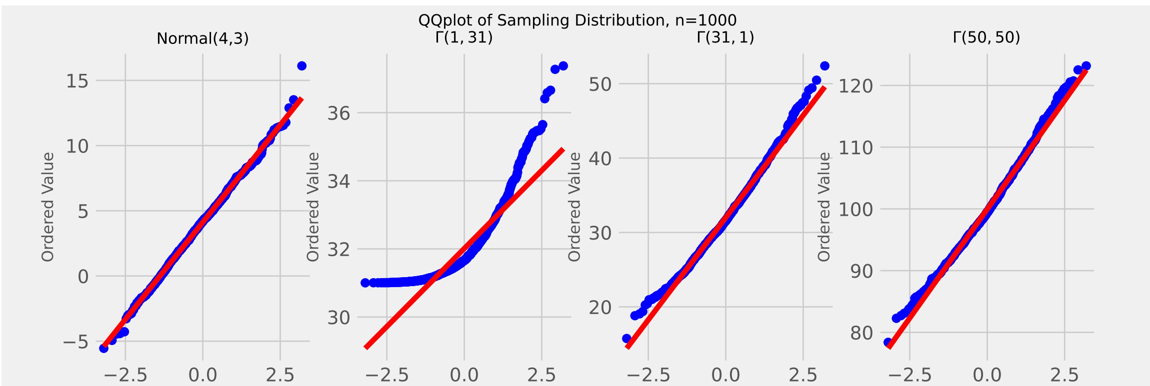

圖 28: QQplot of Sampling Gamma Distribution, n=1000

由圖 27 與 28 我們可以看出，無論樣本數有多大，$\Gamma(1,31)$ 都不會貼近常態分配，而 $\Gamma(31,1)$ 與 $\Gamma(50,50)$ 在樣本數僅有 50 時還不與常態分配貼近，但當樣本數達到 1000 時，兩個分配都會貼近常態分配。

2.5 Beta Distribution

貝塔分配 (Beta Distribution) 是一組定義在區間 (0,1) 上的連續機率分配，其機率密度函數為:

$$f(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}du} = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1} \tag{7}$$

另外，貝塔分配的期望值為 $\frac{\alpha}{\alpha+\beta}$、變異數為 $\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$。

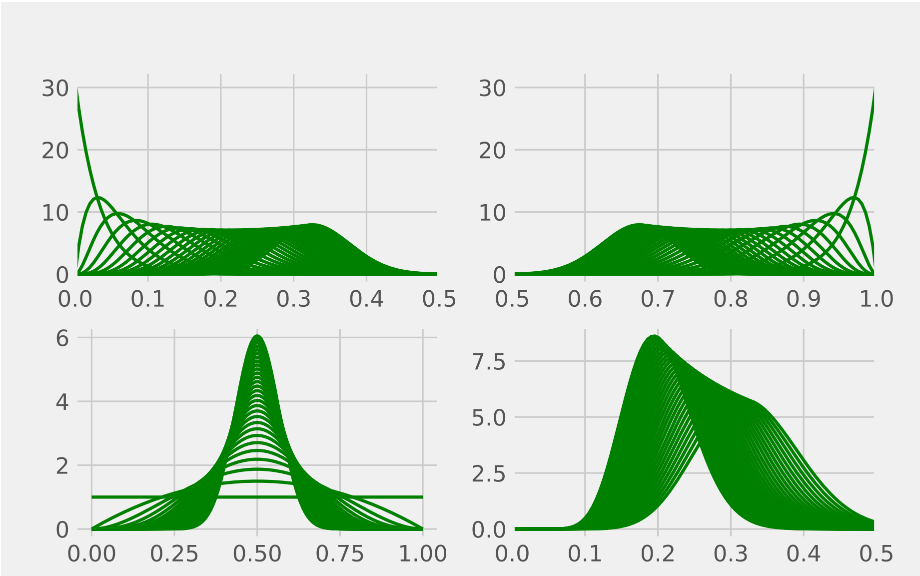

圖 29: Beta Distribution

接著我們設置不同的 α,β 來理解貝塔分配的機率密度函數。在下圖 29 中，左上圖我們設置參數 $a > b$，a 的值介在 1 到 30 之間，b 的值介在 31 到 60 之間。右上圖我們則設置參數為 $a < b$，即 a 的值介在 31 到 60 之間，b 的值介在 1 到 30 之間; 在左下圖中，我們設置 a 與 b 的值相同，且介在 1 到 30 之間; 在右下圖中，我們則繪製給定 $a = 15$，b 介在 31 到 60 之間的貝塔分配 pdf 圖。

由圖可知，在 $a > b$ 時，貝塔分配為右偏分配、在 $a < b$ 時，貝塔分配為左偏分配，而在 $a = b$ 時，貝塔分配則為鐘型分配。另外，在固定 a 的情況下，若 b 值增加時 $(a < b)$，貝塔分配會逐漸變成左偏分配。以下我們呈現部分程式碼:

```
a = np.arange(1, 30)
b = np.arange(31,60)
x = np.linspace(0, 1, 200) #向量
Y = beta.pdf(x.reshape(-1,1), a , b) #矩陣
axes[0][0].plot(x, Y, lw=2, c="g", alpha=0.5)
axes[0][0].set_xlim(0, 0.5)
```

Sampling Beta Distribution

接著，我們利用亂數產生抽樣分配 $Beta(15, 30)$, $Beta(30, 30)$ 以及 $Beta(30, 15)$，並繪製樣本數大小分別為 $n = 200, n = 1000$ 的直方圖。

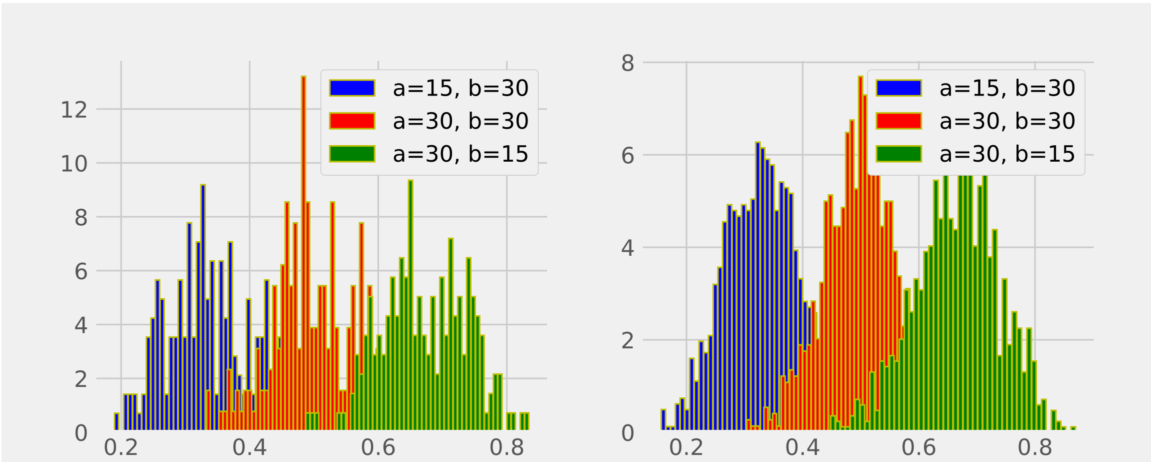

圖 30: Histogram of Sampling Beta Distribution

在 30 的左圖為樣本數 $n = 200$ 的直方圖，右圖則為樣本數 $n = 1000$ 的直方圖。由圖可知，樣本數夠大時，無論 $a < b, a = b$ 還是 $a > b$，貝塔分配的機率密度圖都會趨近於鐘型分配。接著，我們繪製在樣本數為 1000 時，$Beta(15, 30), Beta(30, 30)$ 以及 $Beta(30, 15)$ 的 qqplot 圖。

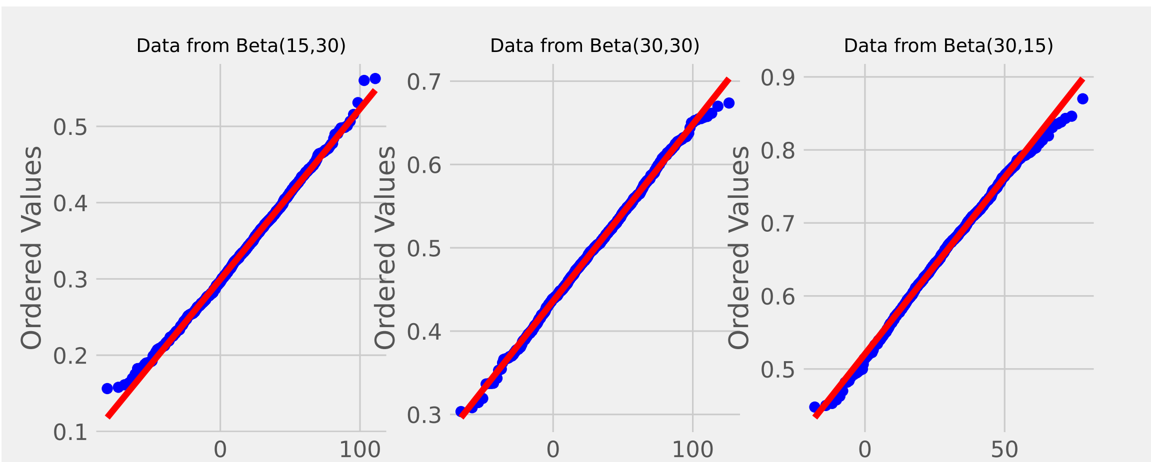

圖 31: QQplot of Sampling Beta Distribution

由圖 31 可知，亂數產生的服從貝塔分配的隨機變數會幾乎貼合紅色的線，即在樣本數夠大時，無論參數的值為何，貝塔分配都會趨近於常態分配。最後，我們亦繪製不同樣本數時的 ecdf 圖，如下圖 32。

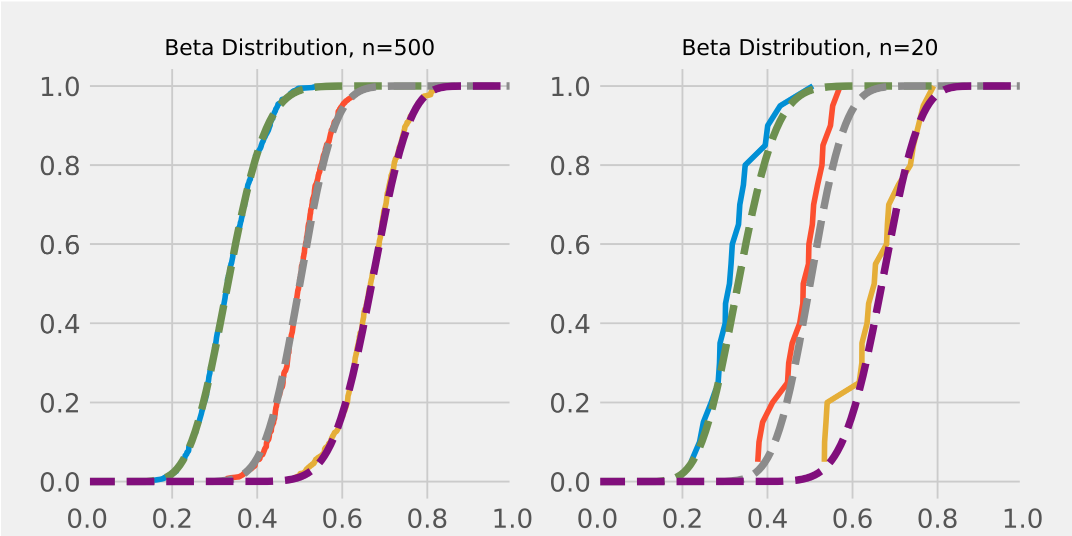

圖 32: ecdf of Sampling Beta Distribution

由圖 32 可知，當 $n = 20$ 時，亂數抽取的抽樣分配與貝塔分配的 cdf 不完全相同，而在樣本數 $n = 500$ 時，兩者則幾乎完全貼近。以下為其部分程式碼:

```
n = 500
x1 = beta.rvs(a1, b1, size=n)
x2 = beta.rvs(a2, b2, size=n)
x3 = beta.rvs(a3, b3, size=n)
x_sort1 = np.sort(x1)
F = np.arange(1 ,n+1) / n
axes[0].plot(x_sort1, F, lw =3)
x_sort2 = np.sort(x2)
x_sort3 = np.sort(x3)

x = np.linspace(0, 1 ,1000)
y1 = beta.cdf(x.reshape(-1, 1), a1, b1)
y2 = beta.cdf(x, a2, b2)
y3 = beta.cdf(x, a3, b3)
```

2.6 Cauchy Distribution

柯西分配 (Cauchy Distribution) 是一種連續機率分配，其機率密度函數為:

$$f(x; x_0, \gamma) = \frac{1}{\pi\gamma[1 + (\frac{x-x_0}{\gamma})^2]} = \frac{1}{\pi}[\frac{\gamma}{(x-x_0)^2+\gamma^2}] \tag{8}$$

在 $x_0 = 0$ 且 $\gamma = 1$ 的特例為標準柯西分配，其機率密度函數為:

$$f(x; 0, 1) = \frac{1}{\pi(1 + x^2)} \tag{9}$$

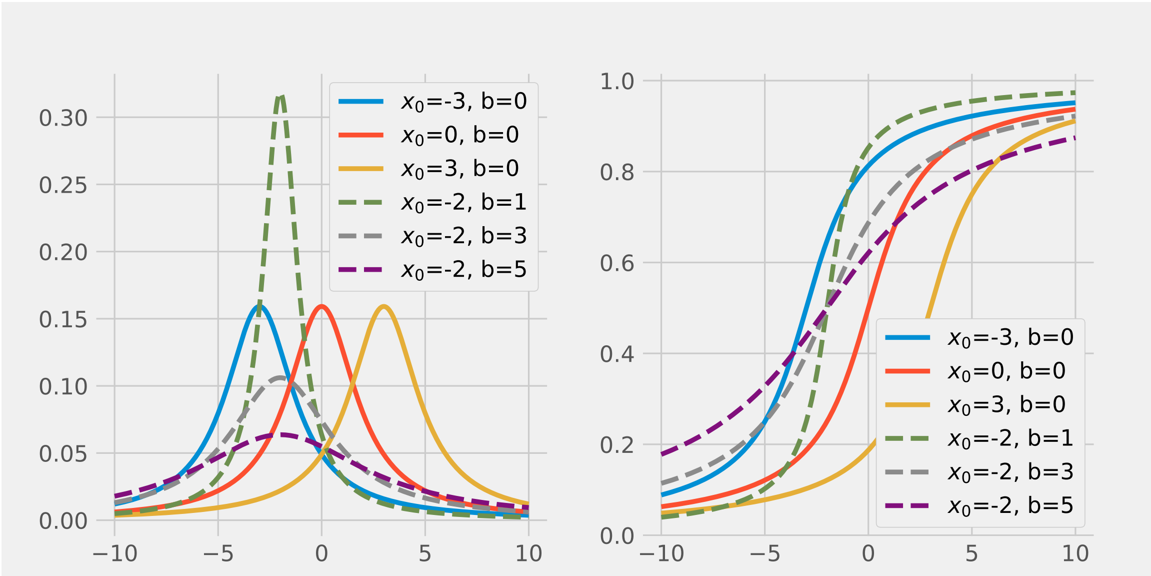

圖 33: Cauchy Distribution

在圖 33 中，我們透過改變參數 x_0 與 b 的值來觀察柯西分配的 pdf 與 cdf 圖。由其左圖可知，改變 x_0 會改變其分配的位置，改變 b 的值則會改變圖形的形狀，隨著 b 值變大，分配的端點會從 0.3 左右變成 0.05 左右，而其 cdf 圖同樣會隨著 x_0, b 的改變而變化。

2.7 Normal Distribution

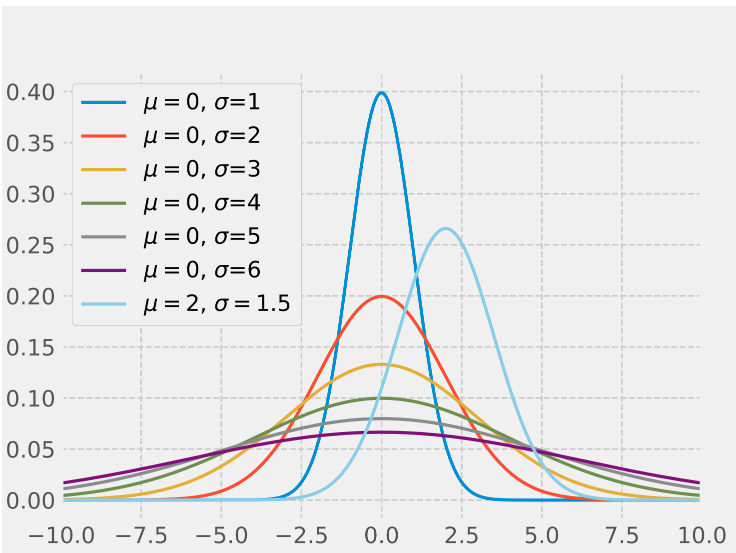

圖 34: Normal Distribution

上圖 34 即為常態分配改變參數所得之機率密度圖，由圖可知，μ 的改變會使分配平移，σ 的改變則會使分配的圖形改變。以下我們呈現其部分的程式碼:

```
mu = 0
sigma = np.arange(1,7)
xlim = [mu - 5 * sigma.max(), mu + 5 * sigma.max()]
x = np.linspace(xlim[0], xlim[1], 1000)
Y = norm.pdf(x.reshape(-1,1), loc=mu, scale=sigma)

plt.plot(x, Y, label = ["$\mu=0$, $\sigma$={}".format(i) for i
    in sigma], lw=2)
```

2.8 t-distribution

司徒頓 t 分配 (Student's t-distribution) 是用以根據小樣本來估計母體呈常態分配且標準差未知的期望值，若母體標準差已知或樣本數夠大時則用常態分配進行估計。t 分配的機率密度函數為:

$$f(t) = \frac{\Gamma\frac{v+1}{2}}{\sqrt{v\pi}\Gamma(\frac{v}{2})}(1+\frac{t^2}{v})^{\frac{-(v+1)}{2}} \tag{10}$$

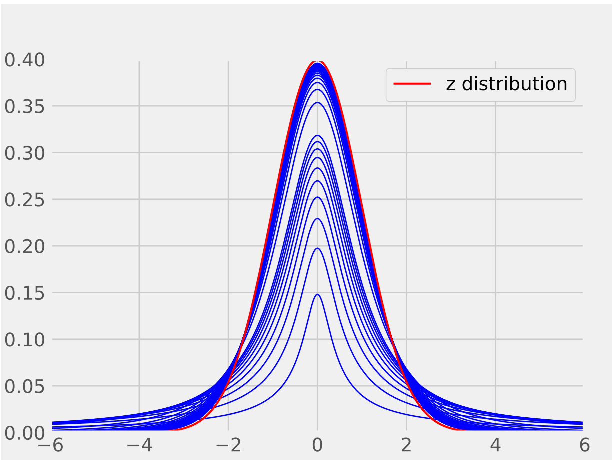

圖 35: t Distribution

由圖 35 我們可以發現 t 分配與標準常態分配都是鐘型分配，且在 t 分配的自由度夠大時，t 分配會與標準常態分配相同。以下呈現部分程式碼:

```
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)]
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', alpha=0.3)
```

2.9 F-distribution

定義隨機變數 X 有母數為 d_1 與 d_2 的 F 分配，寫作 $X \sim F(d_1, d_2)$，對於實數 $x \geq 0$，其機率密度函數為:

$$f(x; d_1, d_2) = \frac{\sqrt{\frac{(d_1 x)^{d_1} d_2^{d_2}}{(d_1 x + d_2)^{d_1 + d_2}}}}{x B(\frac{d_1}{2}, \frac{d_2}{2})} \tag{11}$$

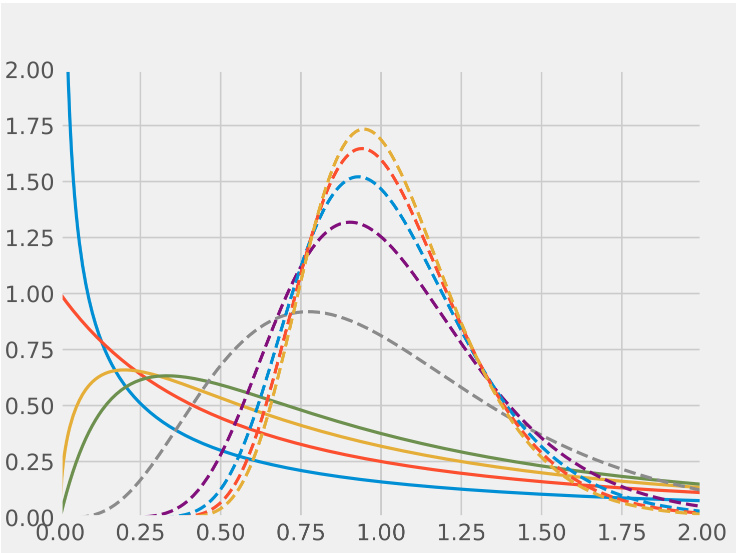

圖 36: F Distribution

我們設置參數 $d_1 = 1, 2, 3, 4$ 且 $d_2 = 1, 2, 3, 4$ 來繪製 F 分配的 pdf 圖與 cdf 圖，可得到圖 36 中的實線圖，另外，我們亦設置參數 $d_1 = 10, 30, 50, 70, 90$ 與 $d_2 = 60$ 來繪製 pdf 圖與 cdf 圖，可得到圖 36 中的虛線圖。

```
x = np.linspace(0, 5, 1000)
gamma1 = np.arange(1, 5)
gamma2 = np.arange(1, 5)
param = np.vstack((gamma1, gamma2))
param = param.T
for i in range(4):
    y = f.pdf(x, param[i][0], param[i][1])
    plt.plot(x, y, lw=2)
```

3 Project

給定四個數字 (2, 4, 9, 12)。從這四個數字中隨機抽取四個數字（取後放回）並計算其平均數。假設隨機變數 Y 代表這四個數字的平均數。請繪製隨機變數 Y 的 PMF。本題可以直接計算每個平均數的機率，但在此請使用隨機抽樣的方式，估計出這些機率值。其中抽樣的次數可以高至百萬以上。

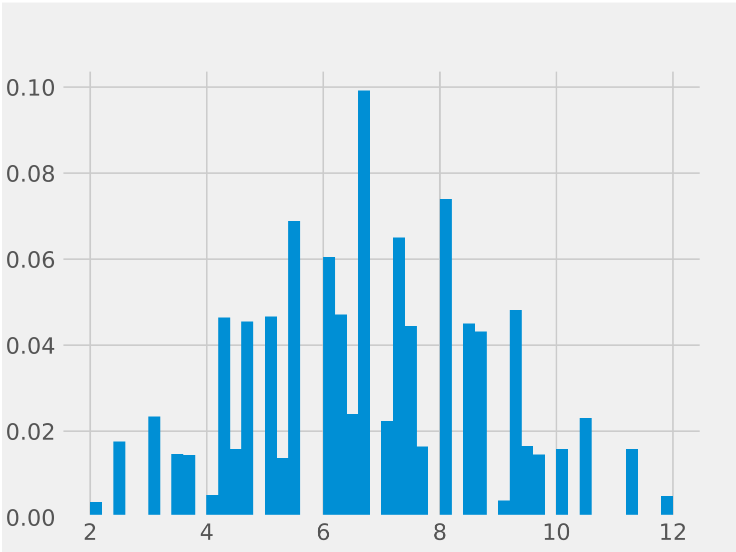

圖 37: Sampling

以上圖 37 即為此抽樣的結果。其部分程式碼亦呈現如下:

```
np.random.seed(seed=1294)
x = np.array([2, 4, 9, 12])
number = 10000
y = np.zeros(number)
for i in np.arange(len(y)):
    rc = np.random.choice(x, 4, replace=True)
    y[i] = rc.mean()
#plt.hist(y, bins= 50, density=True)#error

weight = np.ones_like(y)/len(y)
plt.hist(y, bins=50, weights=weight)
```

4 Conclusion

在本文中，我們成功利用各種圖形來更深入了解各個常用的機率分配的特性，望以後在對分配的特性產生疑問時，能直接利用此文件得到解答，也能更快速的利用 Python 來理解其他本文未提及之分配的特性。