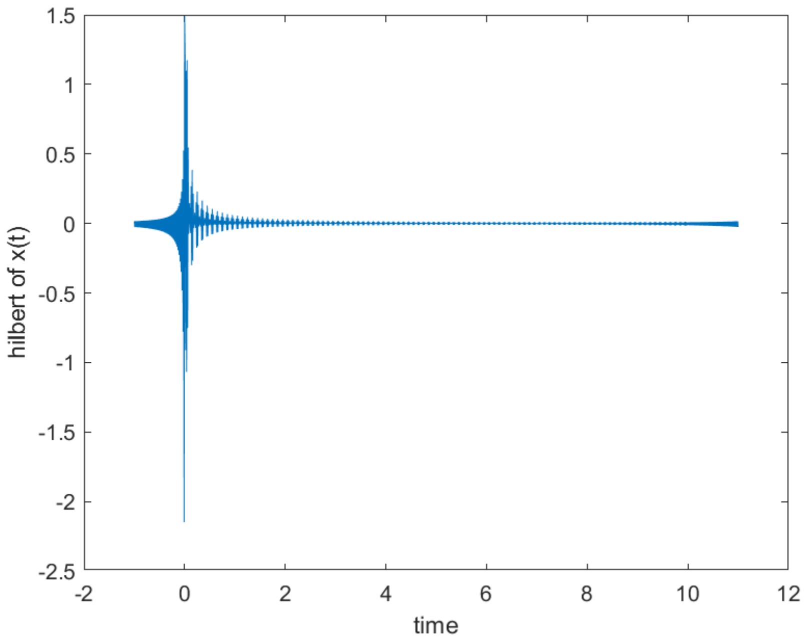
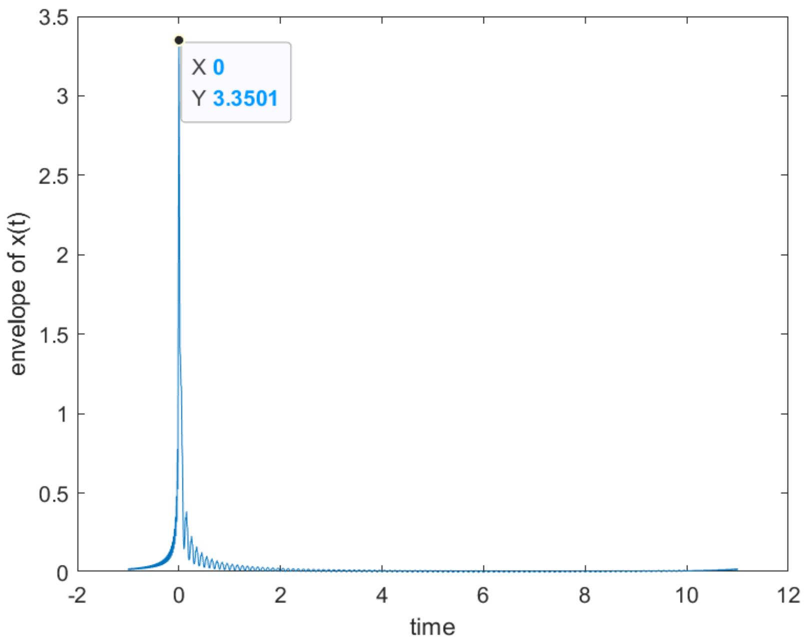
1. **(a)**

對原始訊號 x(t)做 matlab 內建的 hilbert(x)，就可以得到 x(t)的 analytic signal，接著取 analytic signal 的虛部，就是原始訊號 x(t)的 hilbert transform，結果輸出圖如下：



(b)

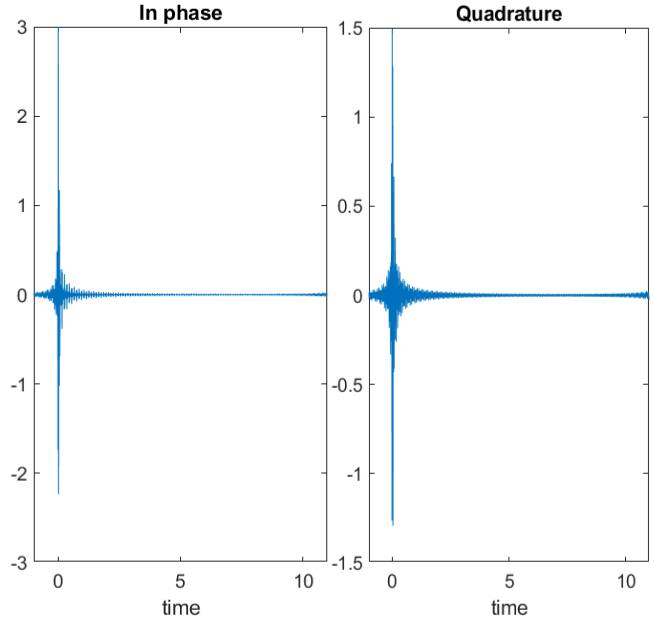
繼上一題得到 analytic signal 後，取其絕對值便可得到原始訊號 x(t)的 envelope，結果輸出圖如下：



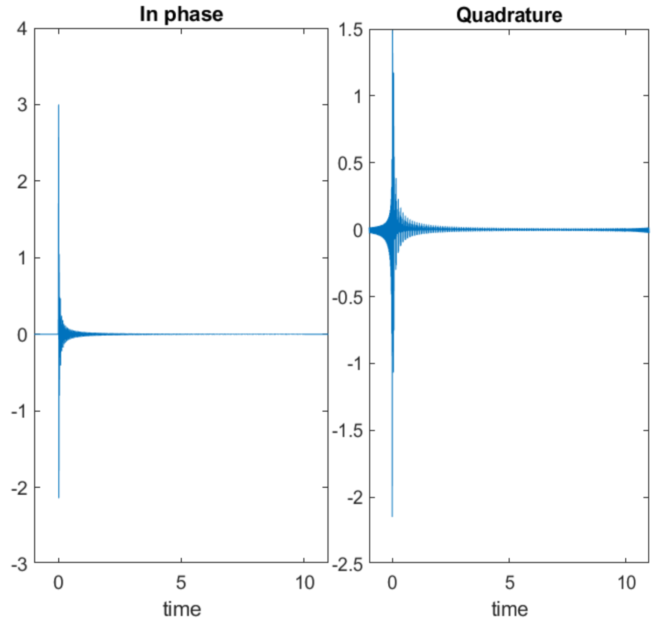
觀察原始訊號 x(t)的 peak，以及 hilbert transfer 的 peak，取其平方相加再開根號，發現計算結果的 envelope peak 值與輸出結果一致。

(c)

接著我們將 analytic signal 與 $e^{-j2\pi f_0 t}$ 相乘得到 lowpass equivalent signal，然後分別取其實部與虛部，就可以得到 lowpass equivalent signal 的 in-phase component 和 quadrature component 了，輸出圖如下：



與原始訊號 x(t)的 in-phase component 和 quadrature component 比較：



程式碼如下：

```
clear; clc; close all;

fs = 100;
ts = 1/fs;
t = -1:ts:11;

x = (2*sinc(20*t).*cos(2*pi*50*t) + sinc(20*t).^2.*cos(2*pi*130*t)).*(t>=0).*(t<=10);
z = hilbert(x);

figure
plot(t,real(z)); xlabel('time'); ylabel('hilbert of x(t)');

% (a)
figure
plot(t,imag(z)); xlabel('time'); ylabel('hilbert of x(t)');

% (b)
figure
plot(t,abs(z)); xlabel('time'); ylabel('envelope of x(t)');

% (c)
f0 = 85;
t2 = 0:ts:(length(x)-1)*ts;
x_low = z.*exp(-1i*2*pi*f0*t2);
figure;
subplot(1,2,1); plot(t,real(x_low)); title('In phase'); xlabel('time');          % In phase component
subplot(1,2,2); plot(t,imag(x_low)); title('Quadrature'); xlabel('time');      % Quadrature component
figure;
subplot(1,2,1); plot(t,real(z)); title('In phase'); xlabel('time');          % In phase component
subplot(1,2,2); plot(t,imag(z)); title('Quadrature'); xlabel('time');      % Quadrature component
```

2. (a)

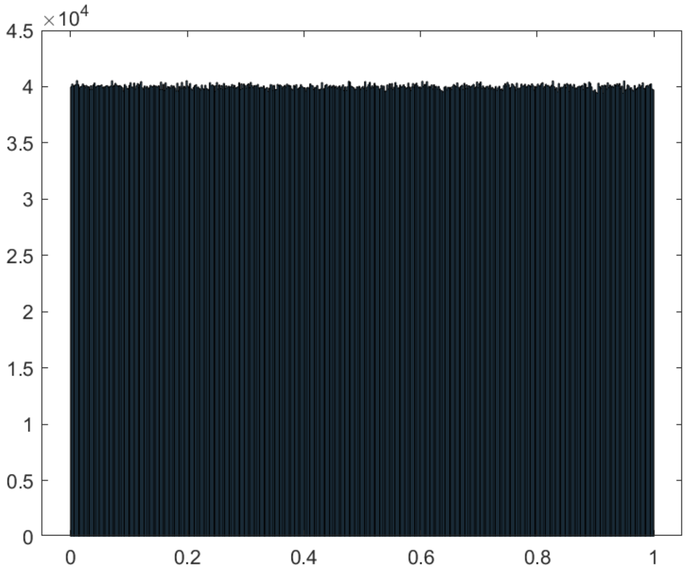
將數量定爲 $2*10^7$ 個，並且將 histogram bar 的數量定為 500 根，使用 rand 函數產生 $2*10^7$ by 1 的 matrix 並用 histogram 函數將 random distribution 畫出來。

程式碼如下：

```
n = 2e7;
u = rand(n,1);

figure;
histogram(u,500);
```

histogram 如下：



(b)

Pareto distribution 的 CDF 為$F(x) = 1 - ({}^{x_m}/_{x})^{\alpha}$, 可以反推得:

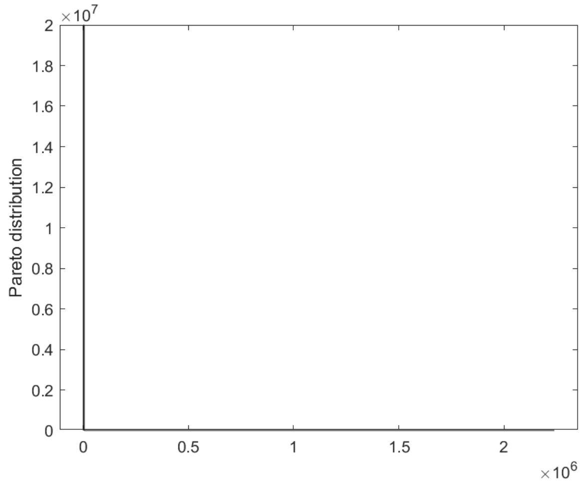
$$x = {}^{x_m}\Big/_{(1-U)^{\frac{1}{\alpha}}}$$

其中 U 是 uniform random variable，將 $x_m = 2.5$, α=1.25 代入，並用 histogram 函數將 Pareto distribution 畫出來。

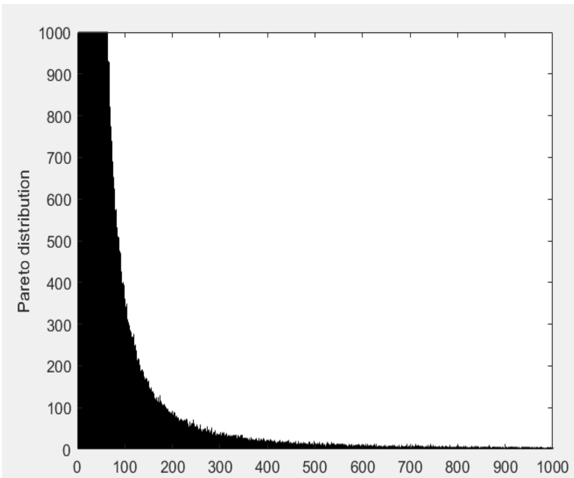
程式碼如下：

```
x_m = 2.5;
alpha = 1.25;
pareto = x_m./(1-u).^(1/alpha);
figure;
histogram(pareto,500);
```

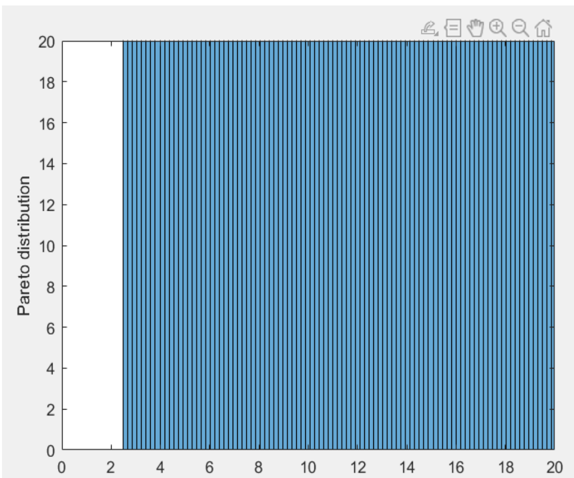
histogram 如下：



觀察發現幾乎所有值都落在 0 的那條 bar 上，推測是因爲數量太大了，並且 x_m 也在同一支 bar 裏面。
如果將每一個 outcome 都輸出成一條 bar，將範圍縮小，則較可以看見輪廓：



將範圍進一步縮小，可以發現小於 x_m 就沒有分佈了，與預期相同。

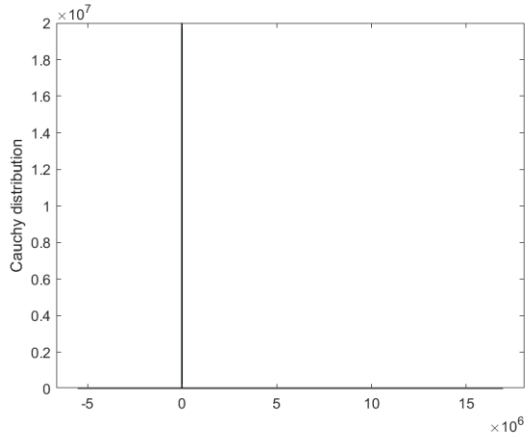


(c)

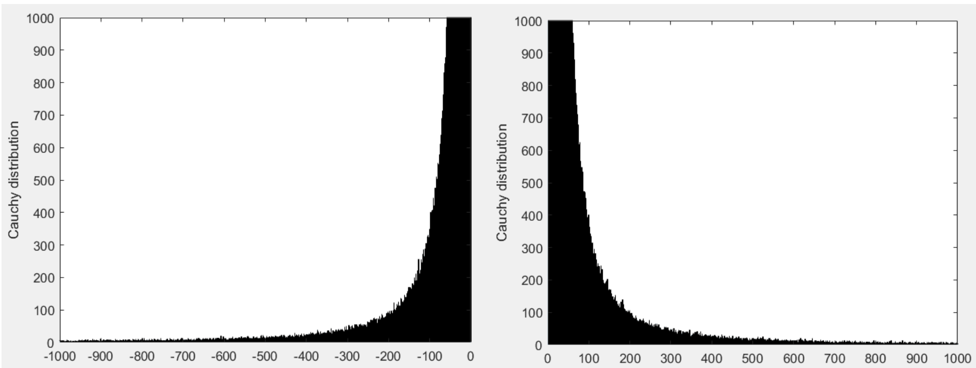
先用 randn 函數得到兩個 normal random variables x_1，x_2，接著算 x_1/x_2，程式碼如下：

```
x_1 = randn(n,1);
x_2 = randn(n,1);
cauchy = x_1./x_2;
figure;
histogram(cauchy,500); ylabel('Cauchy distribution')
```

histogram 如下：



可以發現還是不易辨識，因此和上一題一樣將其放大：



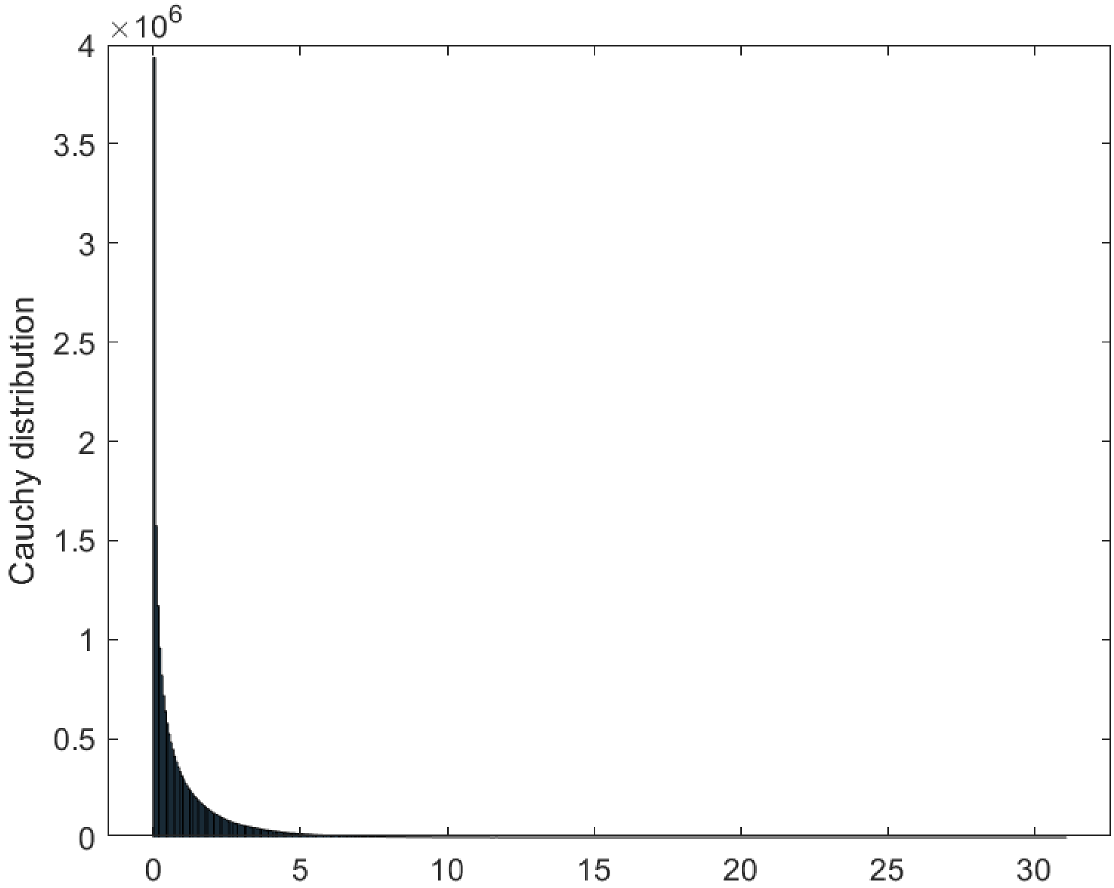
這樣就比較符合預期。

(d)

同樣利用 randn 函數得到 10 by 2 的 matrix，接著自己內積自己得 $Y=X_1^2+X_2^2+\ldots\ldots+X_{10}^2$，以下是程式碼：

```
k = randn(n,10);
chi = zeros(n,1);
for i=1:length(k)
   chi(i)=k(i)*k(i)';
end
figure;
histogram(chi,500); ylabel('Cauchy distribution')
```

接著輸出 histogram：



輸出的結果基本與預期相同。

Histogram 的橫軸是這個 distribution 的值，而縱軸是對應出現的數量，Histogram 就是想要表達一組數值分佈的狀況；而 bar 的數量既不能太多也不能太少，前者會造成運算及輸出時間拉的較長，後者則會讓“解析度”太差，看不到更細緻的分佈。

既然是數值分佈的狀況，那將其除以數量的 summation，也就是將其 normalization，就得到這組數值的機率質量函數 PMF，當 bar 足夠多，也能將其想像成 PDF，將 PMF 與數值相乘並相加，就可以得到這組數據的 expectation，因此 histogram 能大略看出 random variable 的統計分布。