CODE REPORT

Theory:

We've fading channel coefficient is,

$$h = \sum_{i=0}^{l-1} a_i \cos 2\pi F_c \tau_i - j \sum_{i=0}^{l-1} a_i \sin 2\pi F_c \tau_i$$

where, F_c is career frequency

τ_i is delay in i^{th} path.

a_i is attenuation factor

It can be written as,

$$h = x + jy$$

where, $x = \sum_{i=0}^{l-1} a_i \cos 2\pi F_c \tau_i$, $y = -\sum_{i=0}^{l-1} a_i \sin 2\pi F_c \tau_i$

Where x and y are sum of a large set of random variables because of random attenuation (a_i) and random delay (τ_i). When we add a large set of random component variables, it results a Gaussian distribution by central limit theorem.

Now x and y assumed to be Gaussian Random variable with probability distribution function are $F_X(x) = \frac{1}{\sqrt{\pi}} e^{-x^2}$ and $F_{Y(y)} = \frac{1}{\sqrt{\pi}} e^{-y^2}$ respectively.

Writing h as magnitude and phase form we get,

$$h = ae^{j\emptyset}$$

where, $a = \sqrt{x^2 + y^2}$, $\phi = \tan^{-1}\frac{y}{x}$

where, the marginal distribution of a we get after integration is $F_A(a) = 2ae^{-a^2}$($i.e$ Rayleigh distribution.)

In case of $\emptyset$, we get the marginal distribution $1/2\pi$($i.e$ uniform distribution).

Code Workflow:

Using the function

```
numpy.random.randn(N)
```

We can generate an array of normally distributed random numbers of size N with mean zero and unity standard variation.

The snippet below

```
h = (1/numpy.sqrt(2))*(numpy. random.randn(N) + numpy. random.randn(N)*1j)
```

creates an array of complex numbers with random components of size N=100000 with mean zero and variance 1/2.

With the help of abs function we calculated magnitude(modulus) of all elements and stored in array a.

And created the histogram class of above array a with uniform class size of 0.05 using the following snippet

```
pdfa, bin_edges = np.histogram(a,np.arange(0,4.05,0.05))
```

Probability of each histogram class can be calculated as,

$$P(i^{th}\ class) = \frac{frequency\ of\ i^{th}\ class}{N}$$

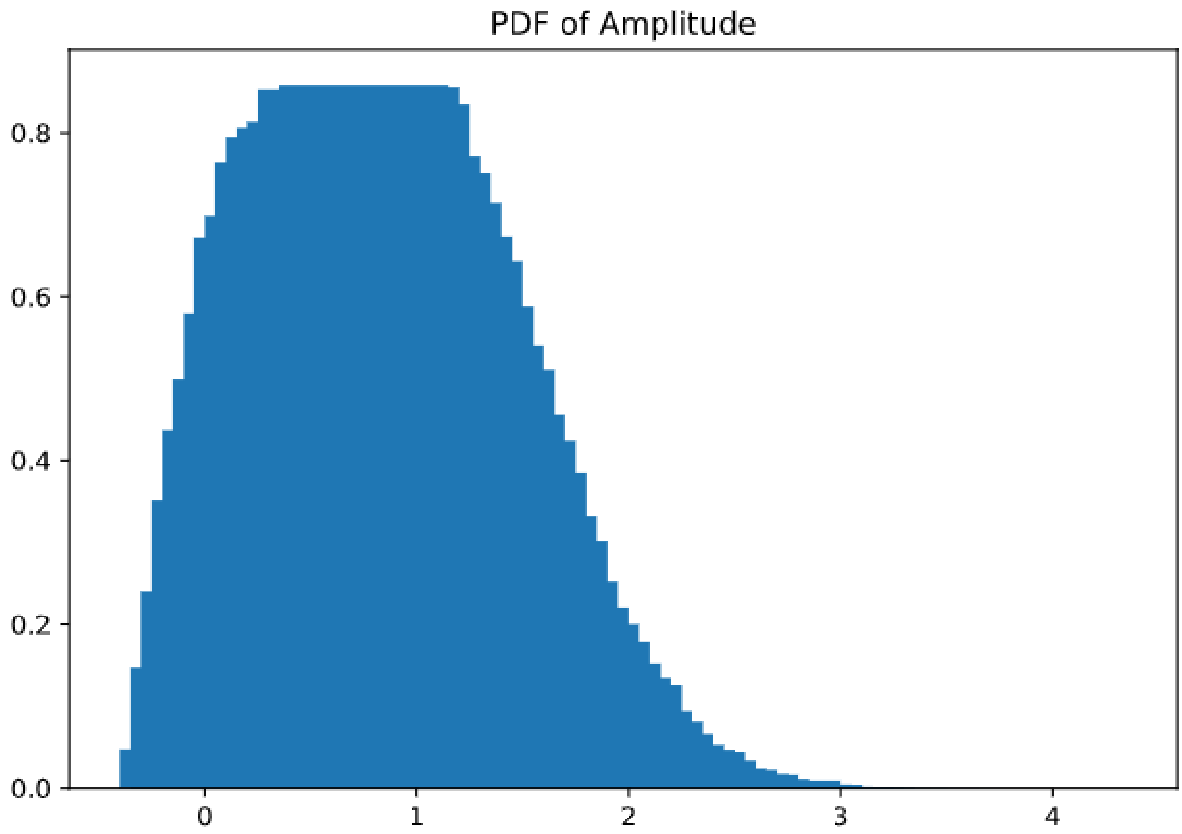
And, approximate density of each histogram class can be seen as,

$$PD(i^{th} class) = \frac{P(i^{th} class)}{size\ of\ i^{th} class}$$

Now plot of amplitude(magnitude) probability density is figured using following snippet,

```
#get plot of modulus graph
fig1 = plt.figure()
ax = fig1.add_axes([0,0,1,1])
ax.bar(np.arange(0,4,0.05), [ele/N/0.05 for ele in pdfa])
ax.set_title('PDF of Amplitude')
```

which generates graph like,



Indeed, the graph comes out to a Rayleigh distribution at a = $(Approx.)\ 0.71 \pm\ 0.05$ have maximum probability density value 0.85.

Similarly, in case of $\phi(Phase\ Graph)$ we get the graph below, In the following graph it can be seen that it follows nearly uniform distribution with $p = .16(approx.)$