

Rayleigh Distribution

Rayleigh distribution is used in signal processing.

It has two parameters.

Scale - (Standard deviation) decides how flat the distribution will be default 1.0)

Size - The shape of the returned array.

E.g: Draw out a sample for Rayleigh distribution with Scale of 2 with Size 2x3;

from numpy import random

x = random.rayleigh(scale=2, size=(2, 3))

print(x)

Visualization of Rayleigh Distribution

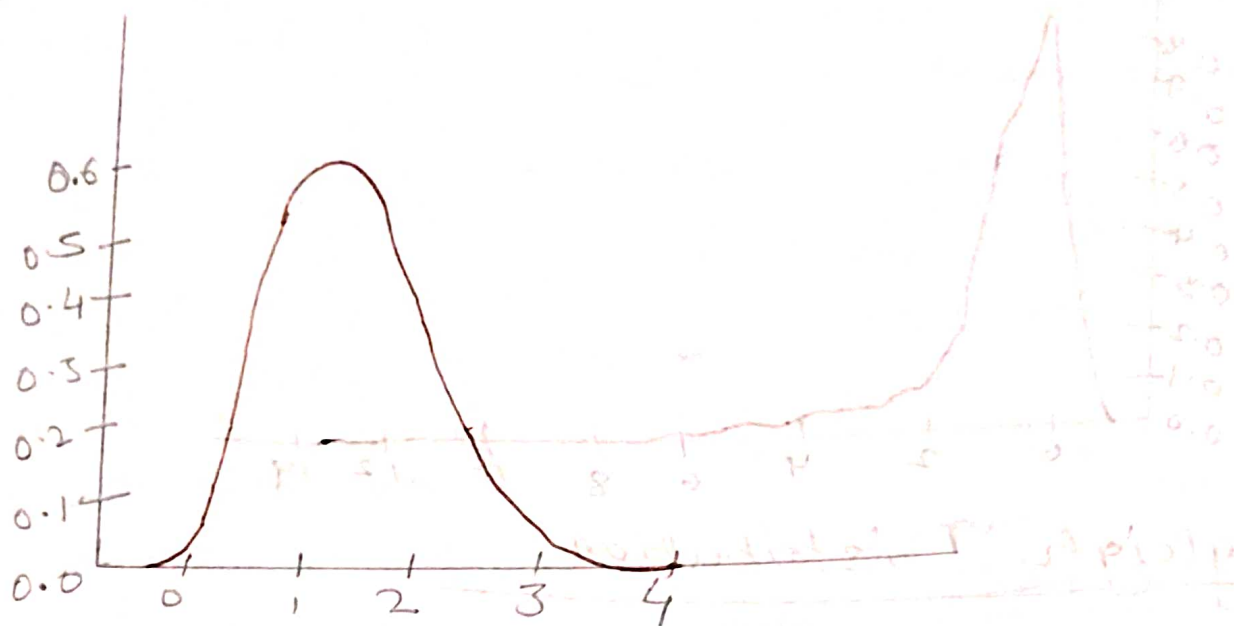
from numpy import random

import matplotlib.pyplot as plt

import seaborn as sns

sns.distplot(random.rayleigh(size=1000), hist=False)

plt.show()



Pareto Distribution

A distribution following Pareto's law i.e. 80-20 distribution (20% factors cause 80% outcome).

It has two parameters.

a - Shape of the returned array parameter.

Size - The shape of the returned array.

eg: Draw out a sample for Pareto distribution with shape of 2 with shape ~~2~~ size 2x3

from numpy import random

$x = \text{random.pareto}(a=2, \text{size}=(2, 3))$

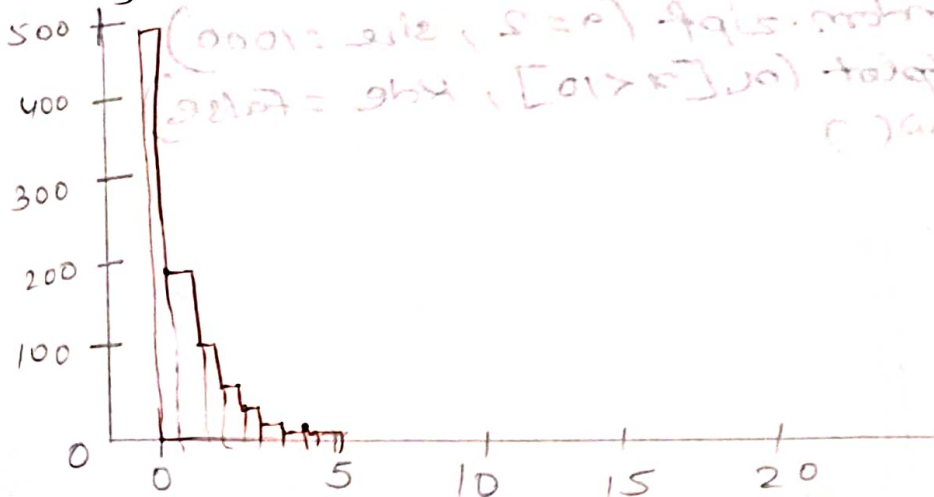
$\text{print}(x)$

Output

```
[[ 0.77719  0.480  1.3300
  4.71001  0.1650  0.1328]]
```


Visualization of Pareto Distribution

```
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns
sns.distplot(random.pareto(a=2, size=1000),
              kde=False)
plt.show()
```



Zipf Distribution

Zipf distributions are used to sample data based on zipf's law.

Zipf's Law: In a collection the n th common term is $1/n$ times of the most common term. e.g 5th common word in english has occurs nearly $1/5$ th times as of the most used word.

It has two parameters:

a - distribution parameter.

size - The shape of the returned array.

e.g Draw out a sample for zipf distribution with distribution parameter 2 with size 2x3

```
from numpy import random
n = random.zipf(a=2, size=(2,3))
```

print(n)

Output
[[1 3]]

[1 8 2]

Visualization of Zipf Distribution

Sample 1000 points but plotting only ones with value < 10 for more meaningful chart.

```
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns
n = random.zipf(a=2, size=1000)
sns.distplot(n[n < 10], kde=False)
plt.show()
```

