NAME: ASIF ERFAN KHAN

ROLL NUMBER: 546

COURSE: MSc CS

SUBJECT: FUNDAMENTALS OF DATA SCIENCE

TOPIC: PRACTICAL 6 CONTINOUOUS DISTRIBUTIONS

CONTINOUS DISTRIBUTION

```
# for inline plots in jupyter
%matplotlib inline
# import matplotlib
import matplotlib.pyplot as plt
# for latex equations
from IPython.display import Math, Latex
# for displaying images
from IPython.core.display import Image
import numpy as np
```

```
# import seaborn
import seaborn as sns
# settings for seaborn plotting style
sns.set(color_codes=True)
# settings for seaborn plot sizes
sns.set(rc={'figure.figsize':(5,5)})
```

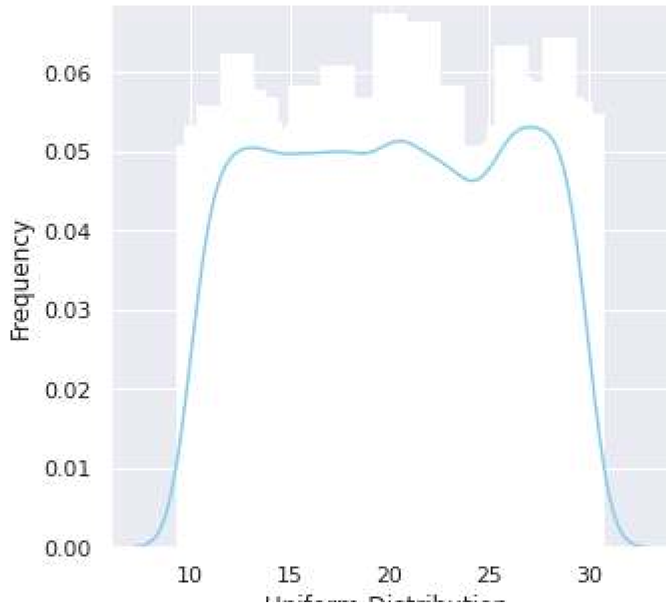
UNIFORM DISTRIBUTION

You can visualize uniform distribution in python with the help of a random number generator acting over an interval of numbers (a,b). You need to import the uniform function from scipy.stats module.

```
# import uniform distribution
from scipy.stats import uniform
```

```
# random numbers from uniform distribution
n = 10000
start = 10
width = 20
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `di
  warnings.warn(msg, FutureWarning)
[Text(0, 0.5, 'Frequency'), Text(0.5, 0, 'Uniform Distribution ')]
```



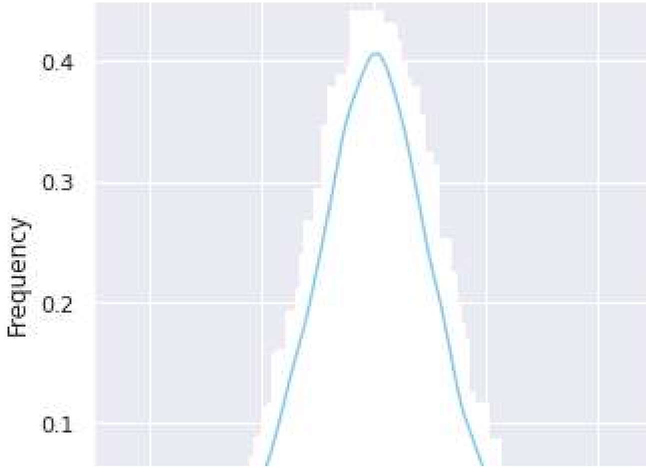
NORMAL DISTRIBUTION

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

```
from scipy.stats import norm
# generate random numbers from N(0,1)
data_normal = norm.rvs(size=10000,loc=0,scale=1)
```

```
ax = sns.distplot(data_normal,
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarnin
  warnings.warn(msg, FutureWarning)
[Text(0, 0.5, 'Frequency'), Text(0.5, 0, 'Normal Distribution')]
```



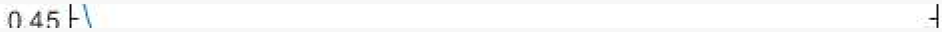
Exponential Distribution

The exponential distribution describes the time between events in a Poisson point process, i.e., a process in which events occur continuously and independently at a constant average rate. It has a parameter λ

called rate parameter, and its equation is described as : $f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0, \\ 0 & x < 0. \end{cases}$

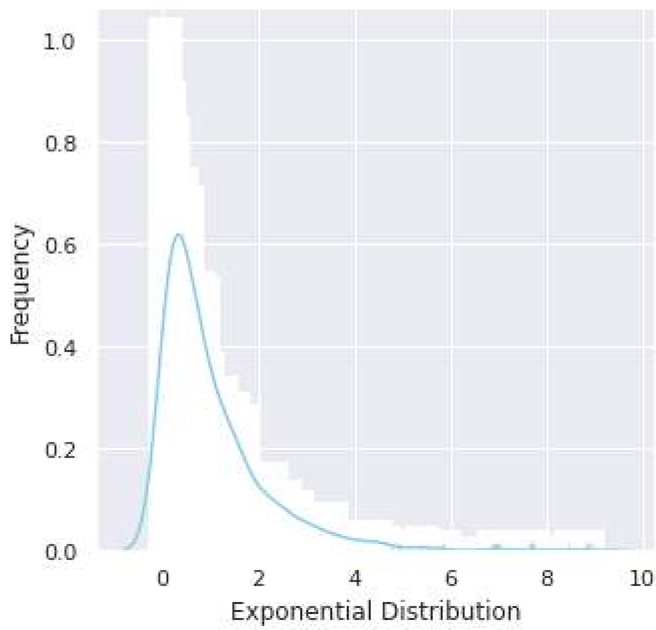
A decreasing exponential distribution looks like :

```
from scipy.stats import expon
data_expon = expon.rvs(scale=1,loc=0,size=1000)
```




```
ax = sns.distplot(data_expon,
                  kde=True,
                  bins=100,
                  color='skyblue',
                  hist_kws={"linewidth": 15,'alpha':1})
ax.set(xlabel='Exponential Distribution', ylabel='Frequency')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarnin
  warnings.warn(msg, FutureWarning)
[Text(0, 0.5, 'Frequency'), Text(0.5, 0, 'Exponential Distribution')]
```



Chi Square Distribution

Chi Square distribution is used as a basis to verify the hypothesis.

```
print(x)
```

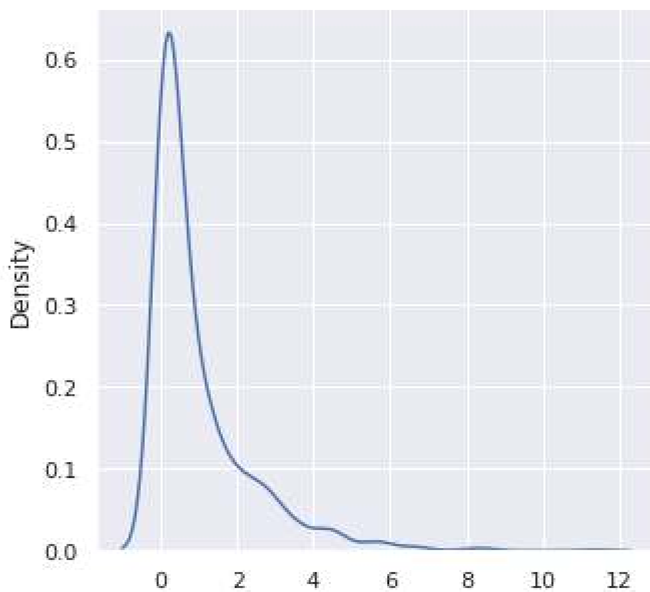
```
[[0.04103389 1.57798989 1.85507302]
 [5.82944896 1.46579974 0.8402198 ]]
```

```
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns

sns.distplot(random.chisquare(df=1, size=1000), hist=False)

plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarnin
  warnings.warn(msg, FutureWarning)
```



Weibull Distribution

```
    return (a / n) * (x / n)**(a - 1) * np.exp(-(x / n)**a)
```

```
count, bins, ignored = plt.hist(np.random.weibull(5.,1000))

x = np.arange(1,100.)/50.

scale = count.max()/weib(x, 1., 5.).max()

plt.plot(x, weib(x, 1., 5.)*scale)

plt.show()
```