Continuous distribution

In [2]:

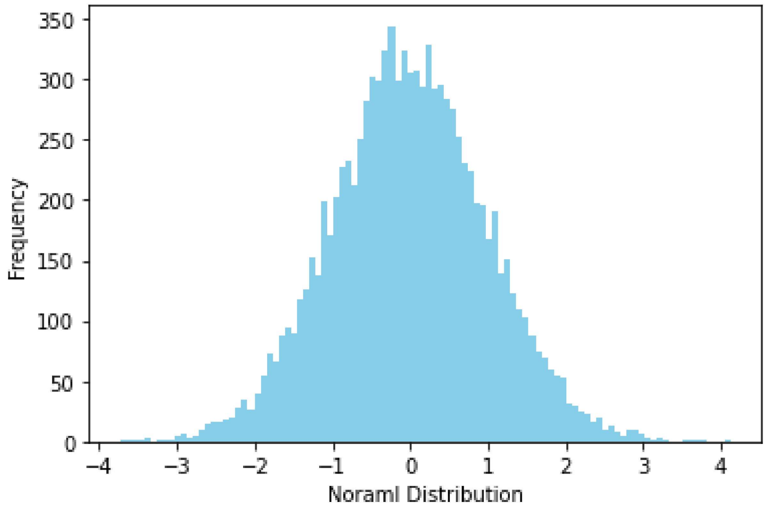
```
from scipy.stats import norm
import pandas as  pd
import matplotlib.pyplot as plt
import seaborn as sns
#gengerate random number N(0,1)
data_normal = norm.rvs(size=10000, loc=0, scale = 1)
```

In [4]:

```
ax = sns.distplot(data_normal,bins=100,kde=False,color="skyblue",hist_kws={"linewidth":
ax.set(xlabel="Noraml Distribution", ylabel="Frequency")
```

Out[4]:

```
[Text(0.5, 0, 'Noraml Distribution'), Text(0, 0.5, 'Frequency')]
```



In [5]:

```
df = pd.read_csv("weight-height.csv")
df.head()
```

Out[5]:

	Gender	Height
0	Male	73.847017
1	Male	68.781904
2	Male	74.110105
3	Male	71.730978
4	Male	69.881796

In [6]:

```
df.Height.describe()
```

Out[6]:

```
count    10000.000000
mean        66.367560
std          3.847528
min         54.263133
25%         63.505620
50%         66.318070
75%         69.174262
max         78.998742
Name: Height, dtype: float64
```

In [14]:

```
mean = df.Height.mean()
mean
```

Out[14]:

```
66.3675597548656
```

In [15]:

```
std_dev = df.Height.std()
std_dev
```

Out[15]:

```
3.847528120795573
```

In [16]:

```
mean-3*std_dev
```

Out[16]:

```
54.824975392478876
```

In [17]:

```
mean+3*std_dev
```

Out[17]:

```
77.91014411725232
```

In [19]:

```
outlier = df[(df.Height<54.82) | (df.Height>77.91)]
outlier
```

Out[19]:

	Gender	Height
994	Male	78.095867
1317	Male	78.462053
2014	Male	78.998742
3285	Male	78.528210
3757	Male	78.621374
6624	Female	54.616858
9285	Female	54.263133

In [25]:

```
no_outlier = df[(df.Height<77.91) & (df.Height>54.82)]
no_outlier.shape
```

Out[25]:

```
(9993, 2)
```

Outlier detection using Z-score

In [33]:

```
df["zscore"]= (df.Height-df.Height.mean())/df.Height.std()
df.head(5)
```

Out[33]:

	Gender	Height	zscore
0	Male	73.847017	1.943964
1	Male	68.781904	0.627505
2	Male	74.110105	2.012343
3	Male	71.730978	1.393991
4	Male	69.881796	0.913375

In [27]:

```
df.Height.mean()
```

Out[27]:

```
66.3675597548656
```

In [28]:

```
df.Height.std()
```

Out[28]:

```
3.847528120795573
```

In [29]:

```
(77.84-66.37)/3.84
```

Out[29]:

```
2.9869791666666665
```

In [30]:

```
df[df["zscore"]>3]
```

Out[30]:

	Gender	Height	zscore
994	Male	78.095867	3.048271
1317	Male	78.462053	3.143445
2014	Male	78.998742	3.282934
3285	Male	78.528210	3.160640
3757	Male	78.621374	3.184854

In [31]:

```
df[df["zscore"]<-3]
```

Out[31]:

	Gender	Height	zscore
6624	Female	54.616858	-3.054091
9285	Female	54.263133	-3.146027

Continous Distribution

In [2]:

```
import matplotlib.pyplot as plt
from IPython.display import Math, Latex
from IPython.core.display import Image
import numpy as np
import seaborn as sns

```

In [3]:

```python
sns.set(color_codes = True)
#setting for seaborn plot sizes
sns.set(rc={'figure.figsize':(5,5)})
```

Uniform Distribution

In [4]:

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

In [5]:

```python
n = 10000
start = 10000
width = 20
data_uniform = uniform.rvs(size=n,loc= start, scale = width)
```

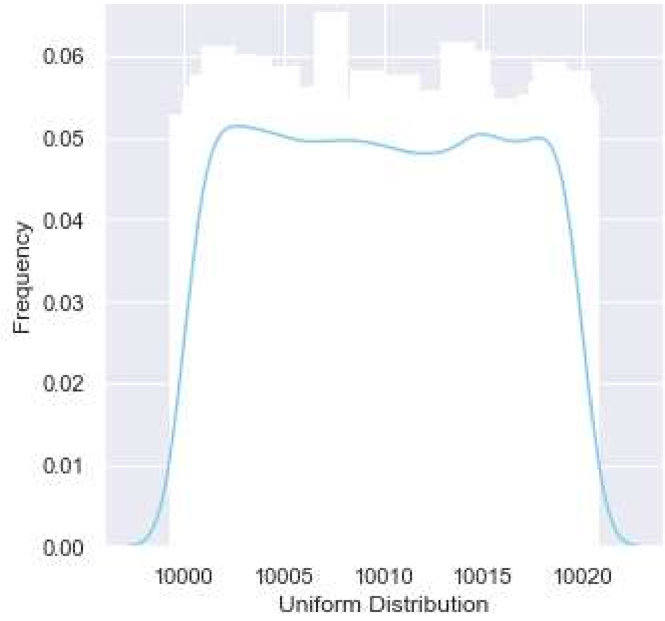
In [6]:

```python
ax = sns.distplot(data_uniform, bins = 100, kde = True, color = "skyblue", hist_kws={"
ax.set(xlabel = "Uniform Distribution", ylabel = 'Frequency')
```

```
C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[6]:

```
[Text(0.5, 0, 'Uniform Distribution'), Text(0, 0.5, 'Frequency')]
```



In [7]:

```python
# Normal Distributiion

from scipy.stats import norm

data_normal = norm.rvs(size=10000,loc=0,scale=1)
```

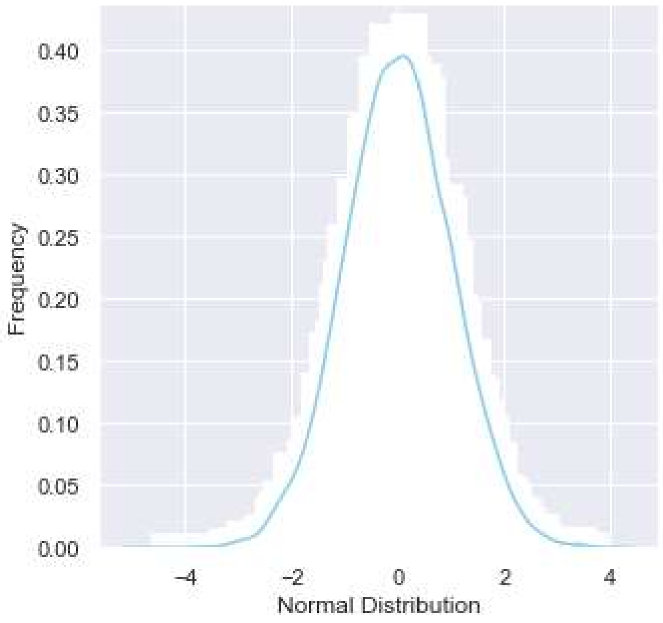
In [9]:

```python
ax = sns.distplot(data_normal,bins = 100, kde=True,color = "skyblue", hist_kws={"linewi
ax.set(xlabel ="Normal Distribution", ylabel = 'Frequency')
```

```
C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[9]:

```
[Text(0.5, 0, 'Normal Distribution'), Text(0, 0.5, 'Frequency')]
```



Exponential Distribution

In [10]:

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

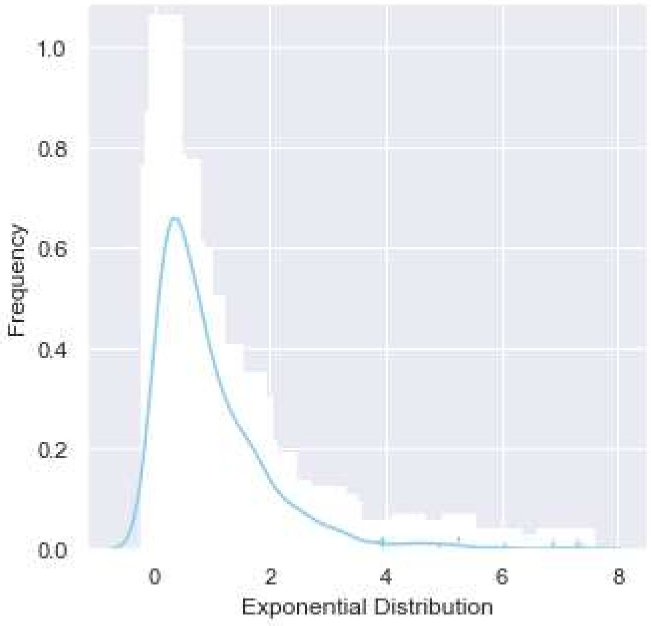
In [11]:

```
ax = sns.distplot(data_expon,bins = 100, kde=True,color = "skyblue", hist_kws={"linewic
ax.set(xlabel ="Exponential Distribution", ylabel = 'Frequency')
```

```
C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[11]:

```
[Text(0.5, 0, 'Exponential Distribution'), Text(0, 0.5, 'Frequency')]
```



Chi-square Distribution

In [12]:

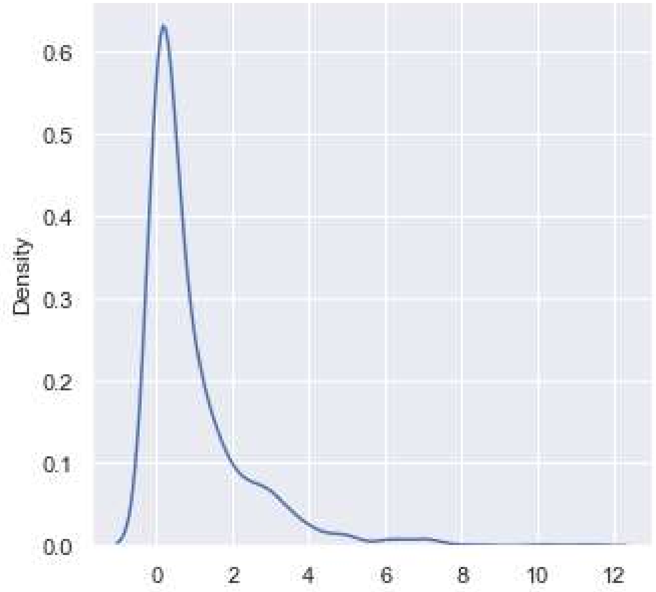
```
from numpy import random

x = random.chisquare(df=2, size=(2,3))

print(x)
```

```
[[0.64419875 2.00803478 2.96603393]
 [0.35941459 3.31861264 1.19927384]]
```

In [13]:

```
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()
```

```
C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `kdeplot` (an axes-level function for
kernel density plots).
  warnings.warn(msg, FutureWarning)
```



In [14]:

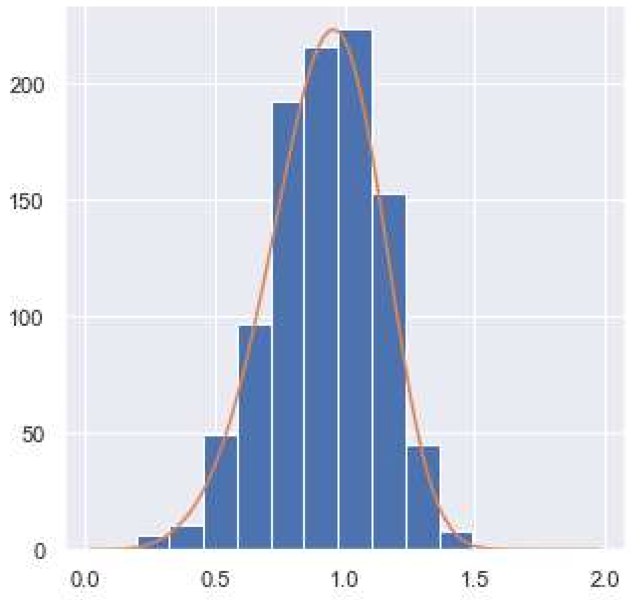
```
a = 5 #shape

s = np.random.weibull(a,1000)
```

In [17]:

```
# Display the histogram of the sample, along with the pdf:

import matplotlib.pyplot as plt

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

def weib(x,n,a):
    return (a / n) * (x / n)**(a-1)*np.exp(-(x / n)**a)
```

In [18]:

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

x = np.arange(1,100.)/50.
scsale = count.max()/weib(x, 1.,5.).max()
plt.plot(x, weib(x, 1., 5.)*scsale)

plt.show()
```



In [ ]:

```

```