- `Binomial distribution`

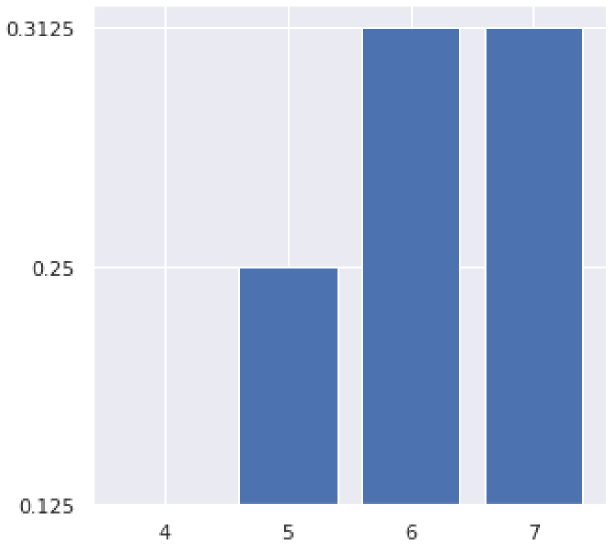
Example you should use to solve:

The Stanley Cup playoff in professional hockey is a seven-game series, where the first team to win four games is declared the champion. The series, then, can last anywhere from four to seven games (just like the World Series in baseball). Calculate the likelihoods that the series will last four, five, six, or seven games. Assume that (1) each game is an independent event and (2) the two teams are evenly matched.

In [ ]:

```
import matplotlib.pyplot as plt

from IPython.display import Math, Latex
import math

from IPython.core.display import Image

import seaborn as sns

sns.set(color_codes=True)

sns.set(rc={'figure.figsize':(5,5)})

from scipy.stats import binom

r_values=list(range(4,8))
dist=[]
for i in range(4,8):
    a=math.factorial(i-1)*(pow(p,4)*pow(1-p,i-4) + pow(p,i-4)*pow(1-p,4))
    a=a/(6*math.factorial(i-4))
    dist.append(str(a))
    print("Probability of finishing in " + str(i) + " games is: " + str(a))

plt.bar(r_values, dist)
plt.show()
```

```
Probability of finishing in 4 games is: 0.125
Probability of finishing in 5 games is: 0.25
Probability of finishing in 6 games is: 0.3125
Probability of finishing in 7 games is: 0.3125
```



Uniform distribution

In [ ]:

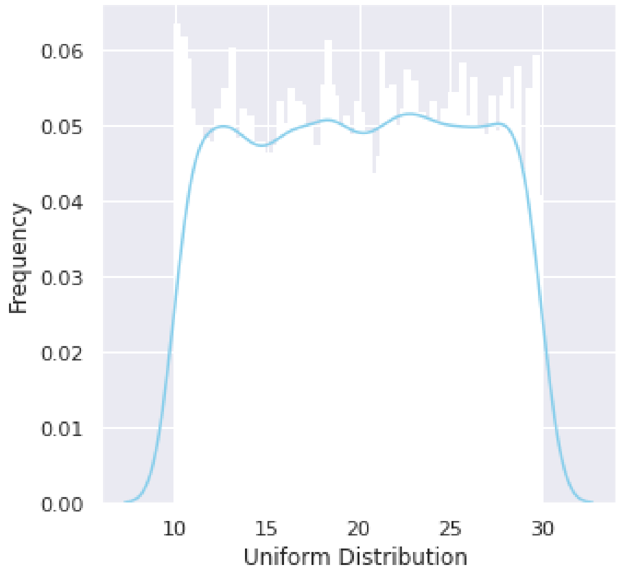
```
import matplotlib.pyplot as plt
```

```
from IPython.display import Math, Latex

from IPython.core.display import Image

import seaborn as sns

sns.set(color_codes=True)

sns.set(rc={'figure.figsize':(5,5)})

from scipy.stats import uniform

n = 10000
start = 10
width = 20
data_uniform = uniform.rvs(size=n, loc = start, scale=width)

ax = sns.distplot(data_uniform,
                  bins=100,
                  kde=True,
                  color='purple',
                  hist_kws={"linewidth": 2,'alpha':1})
ax.set(xlabel='Uniform Distribution ', ylabel='Frequency')
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/distributions.py:2557: FutureWarning: `dis
tplot` is a deprecated function and will be removed in a future version. Please adapt you
r code to use either `displot` (a figure-level function with similar flexibility) or `his
tplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[ ]:

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



- Normal distribution or Gaussian distribution

Example: table is given below.

In [ ]:

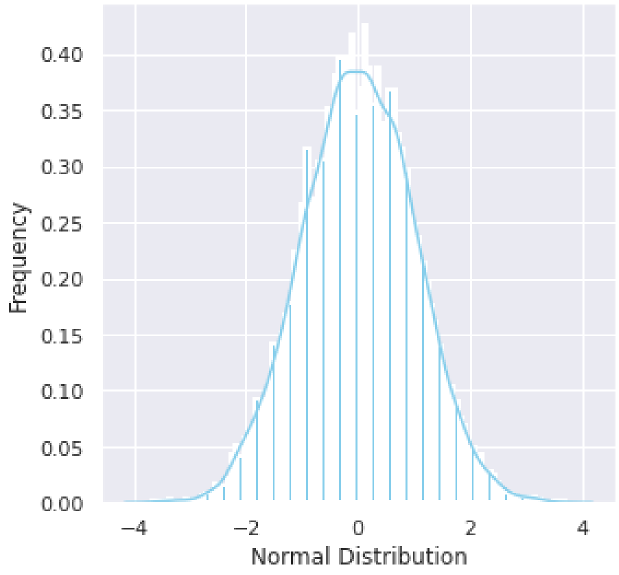
```
import matplotlib.pyplot as plt

from IPython.display import Math, Latex

from IPython.core.display import Image

import seaborn as sns

sns.set(color_codes=True)
sns.set(rc={'figure.figsize':(5,5)})

from scipy.stats import norm

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

ul=norm(loc=0,scale=1).cdf(2.07)
ll=norm(loc=0,scale=1).cdf(0)
print("0 <= Z <= 2.07:",(ul-ll))

ul=norm(loc=0,scale=1).cdf(-0.11)
ll=norm(loc=0,scale=1).cdf(-0.64)
print("-0.64 <= Z <= -0.11: ",(ul-ll))

ll=norm(loc=0,scale=1).cdf(-1.06)
print("Z >= -1.06: ",(1-ll))

print("Z <= -2.33: ",norm(loc=0,scale=1).cdf(-2.33))

ll=norm(loc=0,scale=1).cdf(4.61)
print("Z >= -4.61: ",(1-ll))
ax = sns.distplot(data_normal,
                  bins=100,
                  kde=True,
                  color='skyblue',
                  hist_kws={"linewidth": 2,'alpha':1})
ax.set(xlabel='Normal Distribution', ylabel='Frequency')
```

```
0 <= Z <= 2.07: 0.4807738277724827
-0.64 <= Z <= -0.11:  0.19511838776482165
Z >= -1.06:  0.8554277003360904
Z <= -2.33:  0.009903075559164245
Z >= -4.61:  2.013344854834287e-06
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/distributions.py:2557: FutureWarning: `dis
tplot` is a deprecated function and will be removed in a future version. Please adapt you
r code to use either `displot` (a figure-level function with similar flexibility) or `his
tplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[ ]:

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



Gamma distribution

In [ ]:

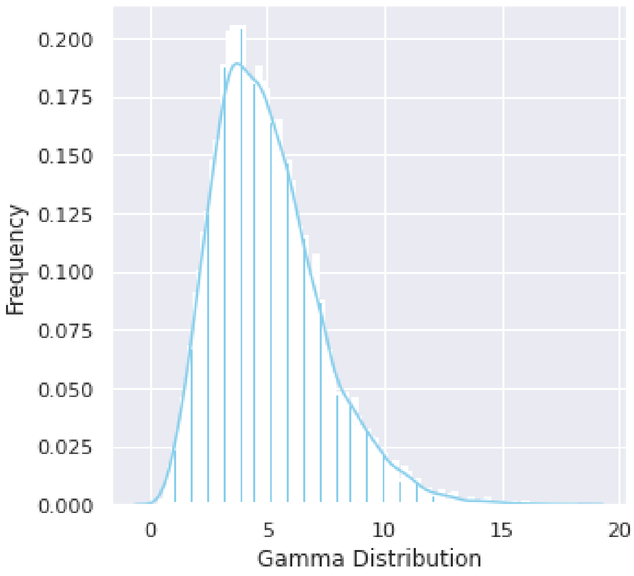
```
import matplotlib.pyplot as plt

from IPython.display import Math, Latex
```

```python
from IPython.core.display import Image

import seaborn as sns

sns.set(color_codes=True)

sns.set(rc={'figure.figsize':(5,5)})

from scipy.stats import gamma

data_gamma = gamma.rvs(a=5, size=10000)

ax = sns.distplot(data_gamma,
                  kde=True,
                  bins=100,
                  color='skyblue',
                  hist_kws={"linewidth": 2,'alpha':1})
ax.set(xlabel='Gamma Distribution', ylabel='Frequency')
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/distributions.py:2557: FutureWarning: `dis
tplot` is a deprecated function and will be removed in a future version. Please adapt you
r code to use either `displot` (a figure-level function with similar flexibility) or `his
tplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[ ]:

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



- Poisson distribution

Example:

A medical study recently documented that 905 mistakes were made among the 289,411 prescriptions written during one year at a large metropolitan teaching hospital. Suppose a patient is admitted with a condition serious enough to warrant 10 different prescriptions. Approximate the probability that at least one will contain an error.

In [73]:

```python
import math

from scipy.stats import poisson

errors=905
pres=289411
test_range=10
mean=test_range*(errors/pres)

print("Required Probability = ",(1 - math.exp(-mean)))
```

```
Required Probability =  0.030786545267269205
```

Exponential Distribution

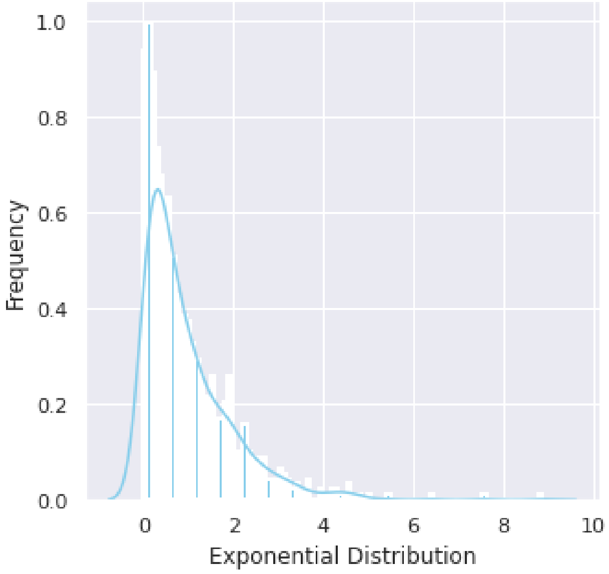
In [ ]:

```
import matplotlib.pyplot as plt

from IPython.display import Math, Latex

from IPython.core.display import Image

import seaborn as sns

sns.set(color_codes=True)

sns.set(rc={'figure.figsize':(5,5)})

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": 2,'alpha':1})
ax.set(xlabel='Exponential Distribution', ylabel='Frequency')
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/distributions.py:2557: FutureWarning: `dis
tplot` is a deprecated function and will be removed in a future version. Please adapt you
r code to use either `displot` (a figure-level function with similar flexibility) or `his
tplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[ ]:

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



Bernoulli Distribution

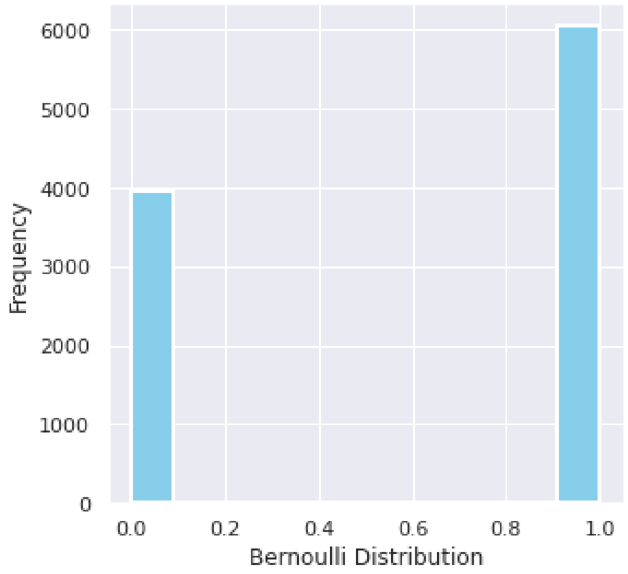
In [ ]:

```
import matplotlib.pyplot as plt

from IPython.display import Math, Latex

from IPython.core.display import Image

import seaborn as sns

sns.set(color_codes=True)
sns.set(rc={'figure.figsize':(5,5)})

from scipy.stats import bernoulli
data_bern = bernoulli.rvs(size=10000,p=0.6)

ax= sns.distplot(data_bern,
                 kde=False,
                 color="skyblue",
                 hist_kws={"linewidth": 2,'alpha':1})
ax.set(xlabel='Bernoulli Distribution', ylabel='Frequency')
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/distributions.py:2557: FutureWarning: `dis
tplot` is a deprecated function and will be removed in a future version. Please adapt you
r code to use either `displot` (a figure-level function with similar flexibility) or `his
tplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[ ]:

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


Frequency
Bernoulli Distribution