Common Probability Distributions

1. Bernoulli Distribution

Dicrete probabiltiy distribution of Random variables which takes the value 1 with probabilty p and the value 0 with probabilty q.

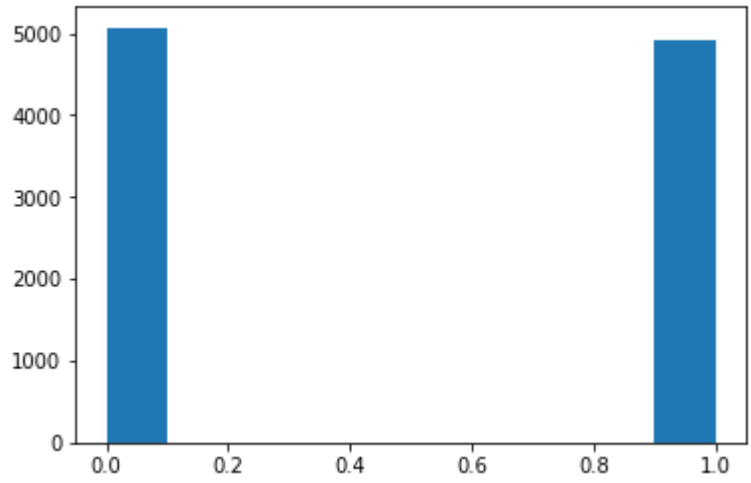
q=1-p

Ex: Call from customer care to check whether you need a credit card or not.

Coin Toss results in head or tail, Game win or Loss etc;

In [19]:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
n=1 #number of trials
p=0.5 #probability of each trial
s= np.random.binomial(n,p,10000) #Create 1000 samples
plt.hist(s)
plt.show()
```



generate a bernoulli distributed discrete random variable using scipy.stats module's bernoulli.rvs() method

which takes p (probability of success) as a shape parameter.

To shift distribution use the loc parameter.

size decides the number of times to repeat the trials.

In [20]:

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

Use Seaborn's distplot to plot the histogram of the distribution you just created.

Seaborn's distplot takes in multiple arguments to customize the plot.

You first create a plot object ax.

Here, you can specify the number of bins in the histogram, specify the color of the histogram and specify density plot option with kde and linewidth option with hist_kws.

You can also set labels for x and y axis using the xlabel and ylabel arguments.

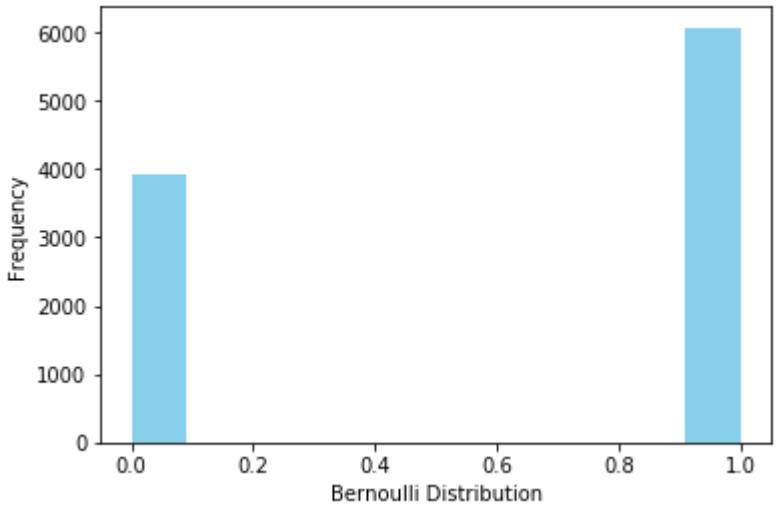
In [21]:

```
ax= sns.distplot(data_bern,
                 kde=False,
                 color="skyblue",
                 hist_kws={"linewidth": 15,'alpha':1})
ax.set(xlabel='Bernoulli Distribution', ylabel='Frequency')
```

```
C:\Users\devi\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: U
serWarning: The 'normed' kwarg is deprecated, and has been replaced by the
'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

Out[21]:

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



2.Binomial Distribution

Generate a binomial distributed discrete random variable using scipy.stats module's binom.rvs()

method which takes n (number of trials) and p (probability of success) as shape parameters.

To shift distribution use the loc parameter.

size decides the number of times to repeat the trials.

In [22]:

```
from scipy.stats import binom
data_binom = binom.rvs(n=10,p=0.8,size=10000)
```

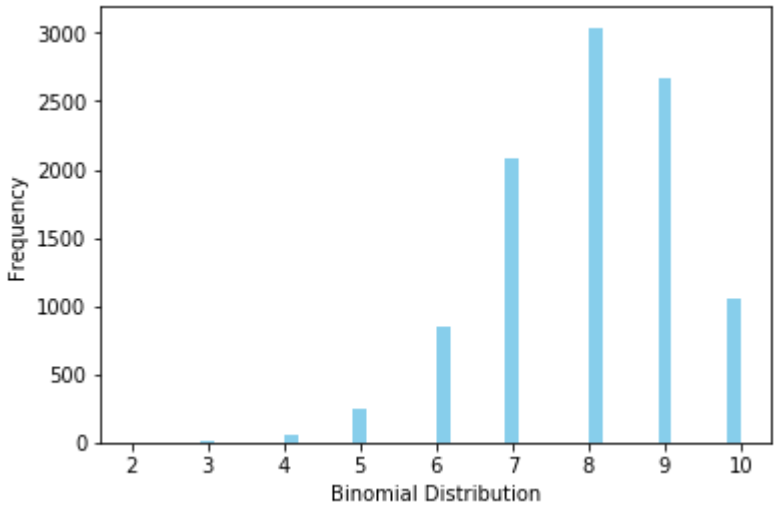
In [23]:

```
ax = sns.distplot(data_binom,
                  kde=False,
                  color='skyblue',
                  hist_kws={"linewidth": 15,'alpha':1})
ax.set(xlabel='Binomial Distribution', ylabel='Frequency')
```

```
C:\Users\devi\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: U
serWarning: The 'normed' kwarg is deprecated, and has been replaced by the
'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

Out[23]:

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



3.Poisson Distribution

Number of times an event occured with in a specific period.

Ex: In a restaurant, avg number of diners for 7 days is 500. You can predict the probability of a certain day having more number of customers.

In [24]:

```
from scipy.stats import poisson
data_poisson = poisson.rvs(mu=5, size=1000)
```

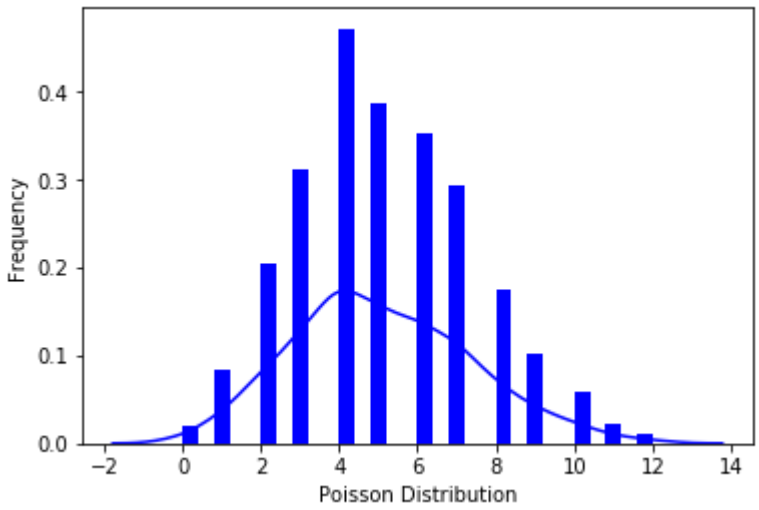
In [25]:

```
ax = sns.distplot(data_poisson,
                  bins=30,
                  kde=True,
                  color='blue',
                  hist_kws={"linewidth": 15,'alpha':1})
ax.set(xlabel='Poisson Distribution', ylabel='Frequency')
```

```
C:\Users\devi\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: U
serWarning: The 'normed' kwarg is deprecated, and has been replaced by the
'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

Out[25]:

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



4. Normal Distribution

Generate a normally distributed random variable using scipy.stats module's norm.rvs() method.

The loc argument corresponds to the mean of the distribution.

Scale corresponds to standard deviation and size to the number of random variates.

If you want to maintain reproducibility, include a random_state argument assigned to a number

In [26]:

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

In [28]:

```
import seaborn as sns
#sns.set(color_codes=True)
#sns.set(rc={'figure.figsize':(5,5)})
ax = sns.distplot(data_normal,
                  bins=100,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
ax.set(xlabel='Normal Distribution', ylabel='Frequency')
```

```
C:\Users\devi\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: U
serWarning: The 'normed' kwarg is deprecated, and has been replaced by the
'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

Out[28]:

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