In [1]:

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

In [2]:

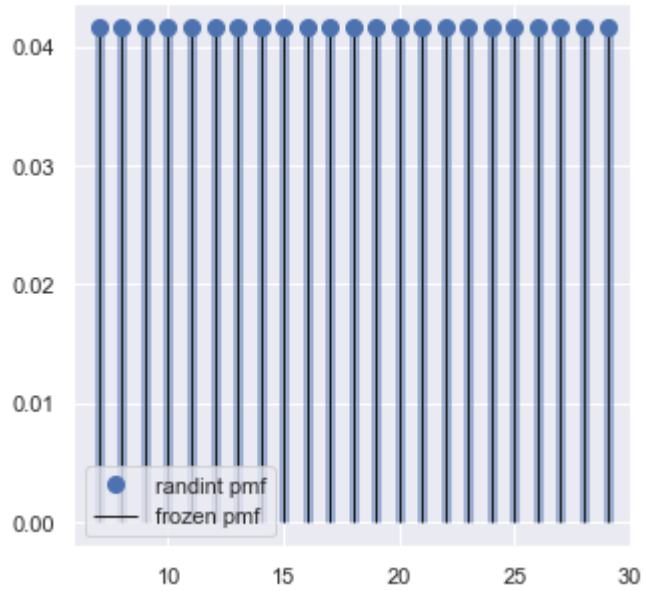
```
import seaborn as sns
sns.set(color_codes=True)
sns.set(rc={"figure.figsize":(5,5)})
```

Uniform Distribution

In [3]:

```
from scipy.stats import randint
fig, ax= plt.subplots(1,1)
# calculate a few first moments:
low, high=7,31
mean, var,skew,kurt =randint.stats(low,high, moments='mvsk')
# Display the probability mass function (''pmf''):
x=np.arange(randint.ppf(0.01,low,high),
            randint.ppf(0.99,low,high))
ax.plot(x, randint.pmf(x,low,high),'bo', ms=8,label='randint pmf')
ax.vlines(x,0, randint.pmf(x,low,high), colors='b', lw=5, alpha=0.5)

# Alternately, the distribution object can be called ( as a function)
# to fix the shape and location. This returns a "frozen" RV object holding
# the given parametres fixed
# Freeze the distribution and display the frozen "pmf"
rv = randint(low, high)
ax.vlines(x, 0, rv.pmf(x), colors = "k", linestyles = "-", lw = 1, label = "frozen pmf")
ax.legend(loc = "best") #frameon = False)
plt.show()
prob= randint.cdf(x,low,high)
np.allclose(x,randint.ppf(prob, low, high))
#True
# Generate random numbers
r=randint.rvs(low,high, size=1000)
```



In [4]:

```
prob= randint.cdf(x,low,high)
np.allclose(x,randint.ppf(prob, low, high))
```

Out[4]:

True

In [5]:

```
from numpy import random as r
import seaborn as sns
import matplotlib.pyplot as plt
uniformMatrix=r.uniform(0.2,0.4,size=(10))
print('n/n',uniformMatrix)
```

```
n/n [0.26415027 0.21595338 0.31380566 0.28682409 0.32085036 0.33442704
 0.30218281 0.32585694 0.32030874 0.23842769]
```

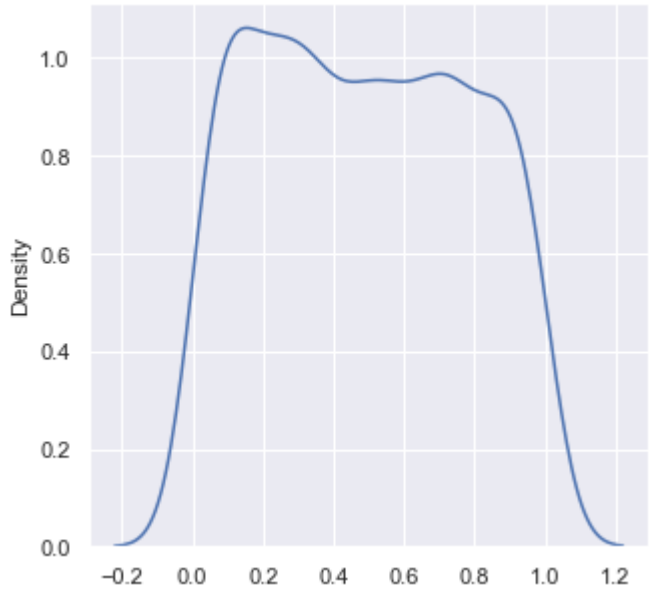
In [6]:

```
sns.distplot(r.uniform(size=(1000)),hist=False)
```

```
C:\Users\MSCIT\anaconda3.1\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-lev
el function with similar flexibility) or `kdeplot` (an axes-level function f
or kernel density plots).
  warnings.warn(msg, FutureWarning)
```

Out[6]:

```
<AxesSubplot:ylabel='Density'>
```



Bernoulli Distribution

p(x) = { 1-p , x=0

p , x=1 }

In [16]:

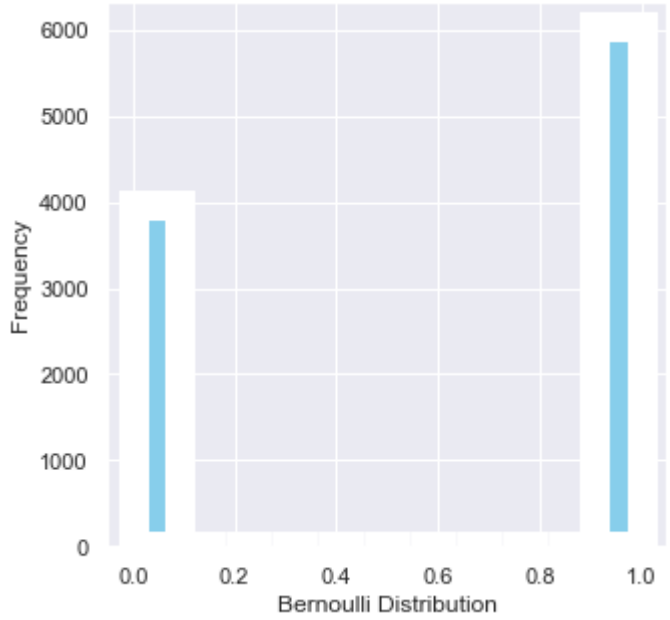
```
from scipy.stats import bernoulli
data_bern = bernoulli.rvs(size=10000,p=0.6)
```

In [17]:

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

Out[17]:

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



Binomial Distribution

In [18]:

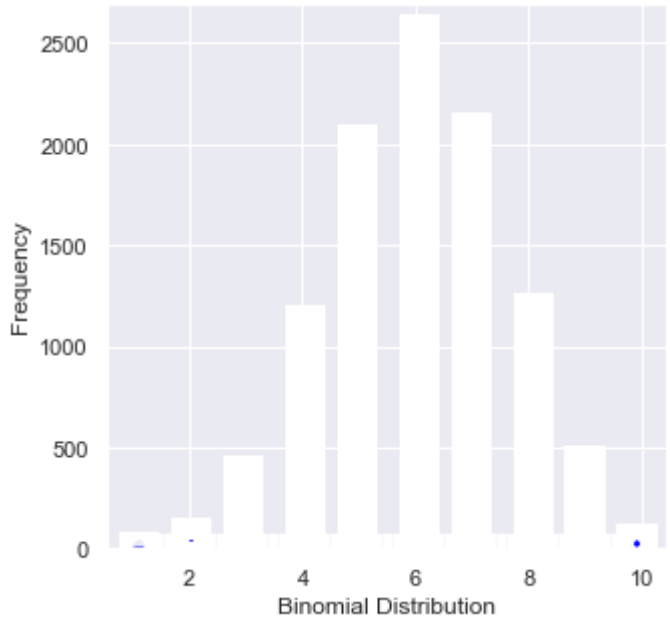
```
from scipy.stats import binom
data_binom = binom.rvs(size=10000,p=0.6,n=10)
```

In [26]:

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

Out[26]:

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



Poisson Distribution

In [27]:

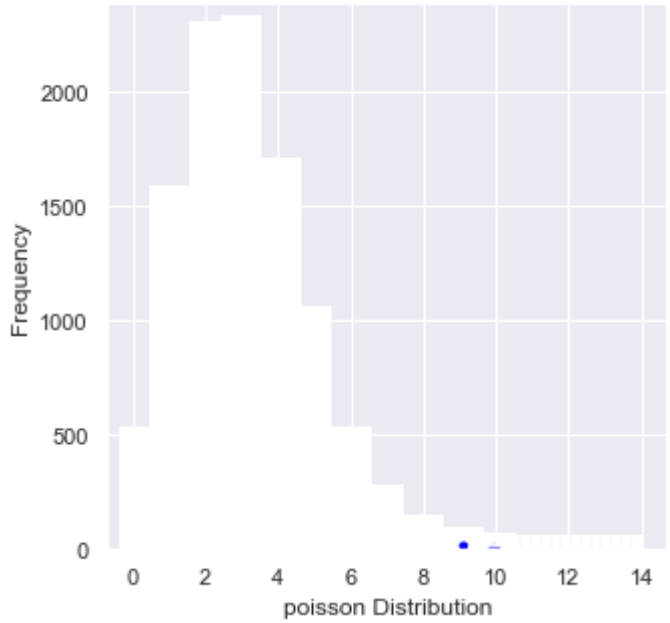
```
from scipy.stats import poisson
data_poisson = poisson.rvs(size=10000,mu=3)
```

In [28]:

```
ax=sns.distplot(data_poisson,
                kde=False,
                color="blue",
                hist_kws={"linewidth":15,"alpha":1})
ax.set(xlabel="poisson Distribution",ylabel="Frequency")
```

Out[28]:

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



In [29]:

```
from scipy.stats import poisson
# A wearhouse typically recives 8 delivery 4 and 5 on friday
# What is the prob. that the only 4 delivery will arrive between 4 and 5 pm on friday?
poisson.pmf(4,8)
```

Out[29]:

```
0.057252288495362
```

In [30]:

```
# What is the probability of having less than 3 delivers on friday between 4 and 5pm?
poisson.cdf(3,8)
```

Out[30]:

```
0.04238011199168396
```

In [31]:

```
# What is the prob. of having no deliveries on friday between 4 ans 5pm?
poisson.pmf(0,8)
```

Out[31]:

```
0.00033546262790251185
```

In [ ]:

```

```