In [3]:

```
import random
n = random.random()    #generates a float number between 0 and 1
print(n)
```

```
0.004977433016524535
```

In [8]:

```
import random
n = random.randint(0,20) # generates an integer between 0 and 20
print(n)
```

```
11
```

In [11]:

```
import random
randomlist = []
for i in range(0,5):
   n = random.randint(1,20)
   randomlist.append(n)
print(randomlist)
```

```
[10, 8, 6, 4, 3]
```

In [13]:

```
import random

randomlist = random.sample(range(10, 20), 5) #Generate 5 random numbers between 10 a
print(randomlist)
```

```
[16, 15, 10, 12, 18]
```

In [5]:

```
import random

mu = 100
sigma = 2
nrv = random.gauss(mu, sigma)

print(nrv)
```

```
99.5944193511432
```

In [14]:

```
# generate random Gaussian values
import random

# seed random number generator
random.seed(1)

mu = 100
sigma = 2

# generate some Gaussian values
for _ in range(10):
   value = random.gauss(mu, sigma)
   print(value)
```

```
102.57636950631093
102.89889121739954
100.13267161787653
98.47091269805674
97.81565356979172
100.06266903366344
97.95579365997826
```

```
97.12634110979494
100.39862395296751
100.26674920931721
```

In [17]:

```
# seed the pseudorandom number generator
import random
# seed random number generator
random.seed(1)
# generate some random numbers
print(random.random(), random.random(), random.random())
# reset the seed
random.seed(1) #reseeding the generator will result in the same sequence of numbers
# generate some random numbers
print(random.random(), random.random(), random.random())
```

```
0.13436424411240122 0.8474337369372327 0.763774618976614
0.13436424411240122 0.8474337369372327 0.763774618976614
```

In [22]:

```
# randomly shuffle a sequence
import random

# seed random number generator
random.seed(1)
# prepare a sequence
sequence = [i for i in range(20)]
print(sequence)
# randomly shuffle the sequence
random.shuffle(sequence)
print(sequence)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
[11, 5, 17, 19, 9, 0, 16, 1, 15, 6, 10, 13, 14, 12, 7, 3, 8, 2, 18, 4]
```

In [23]:

```
import numpy

x = numpy.random.rand()

print(x)
```

```
0.3894476503146842
```

In [25]:

```
import numpy

x = numpy.random.rand(3, 5)

print(x)
```

```
[[0.10989322 0.89979759 0.44543625 0.09611175 0.17782157]
 [0.82455019 0.76309334 0.32146687 0.3717578  0.20625402]
 [0.3506976  0.90059801 0.99410267 0.09357395 0.06388018]]
```

In [26]:

```
import numpy

#Generate a 2-D array with 3 rows, each row containing 5 random integers
#from 0 to 100
x = numpy.random.randint(100, size=(3, 5))

print(x)
```

```
[[41 12 75 65 21]
 [43 29 56 51  9]
 [84 44 52 59 63]]
```

In [28]:

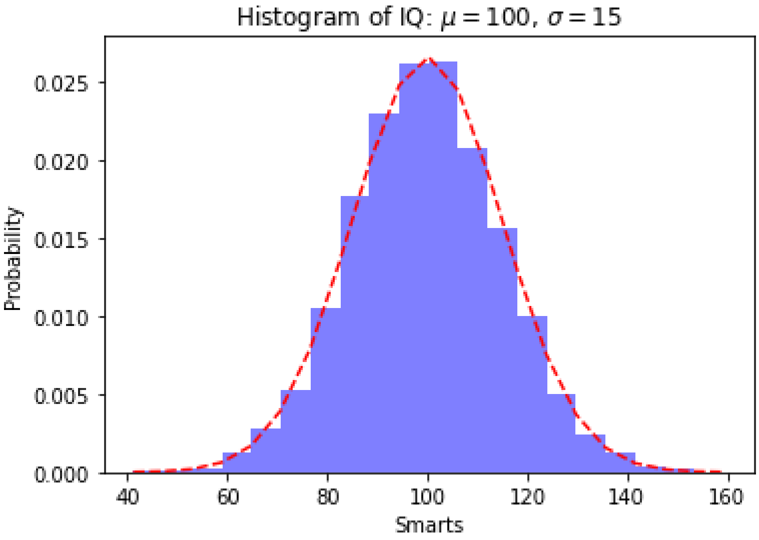
```python
import numpy as np
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
import scipy.stats as sp

# example data
mu = 100 # mean of distribution
sigma = 15 # standard deviation of distribution
x = mu + sigma * np.random.randn(10000)

num_bins = 20
# the histogram of the data
n, bins, patches = plt.hist(x, num_bins, facecolor='blue', density=True, alpha=0.5)

# add a 'best fit' line
y = sp.norm.pdf(bins, mu, sigma)
plt.plot(bins, y, 'r--')
plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title(r'Histogram of IQ: $\mu=100$, $\sigma=15$')

# Tweak spacing to prevent clipping of ylabel
plt.subplots_adjust(left=0.15)
plt.show()
```

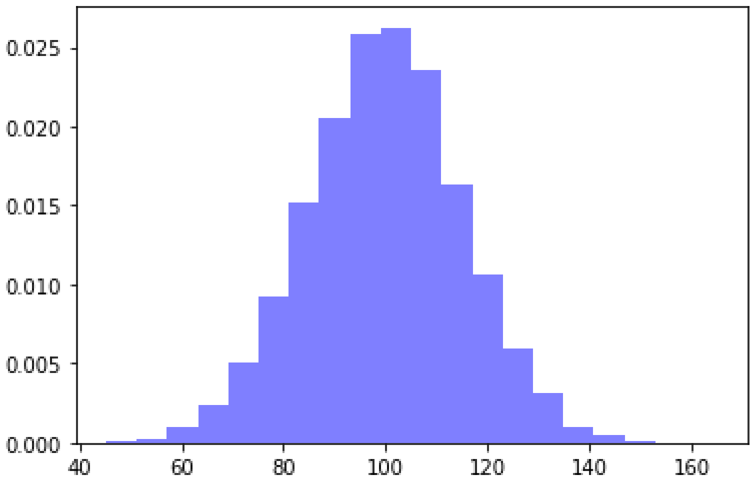


In [5]:

```python
import numpy as np
import matplotlib.pyplot as plt

mu, sigma = 100, 15 # mean and standard deviation
s = np.random.normal(mu, sigma, 10000)

num_bins = 20
n, bins, patches = plt.hist(s, num_bins, facecolor='blue', density=True, alpha=0.5)
```



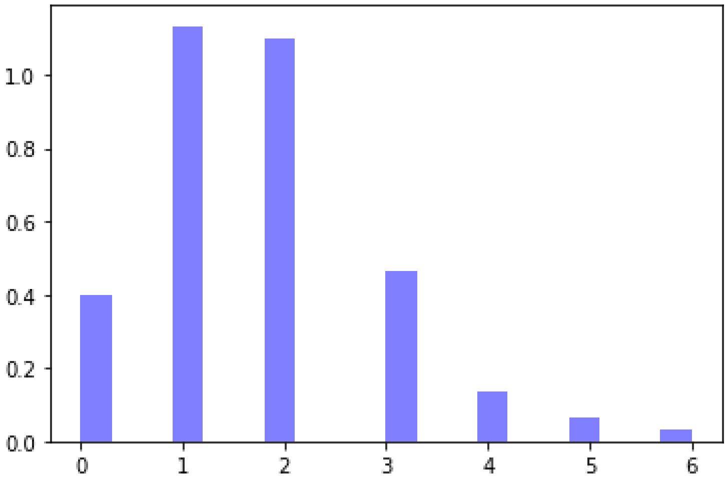
In [32]:

```
import numpy
import matplotlib.pyplot as plt

prv = numpy.random.poisson(lam=1.75, size=100)

num_bins = 20
# histogram of the data
n, bins, patches = plt.hist(prv, num_bins, facecolor='blue', density=True, alpha=0.5

plt.show()
```



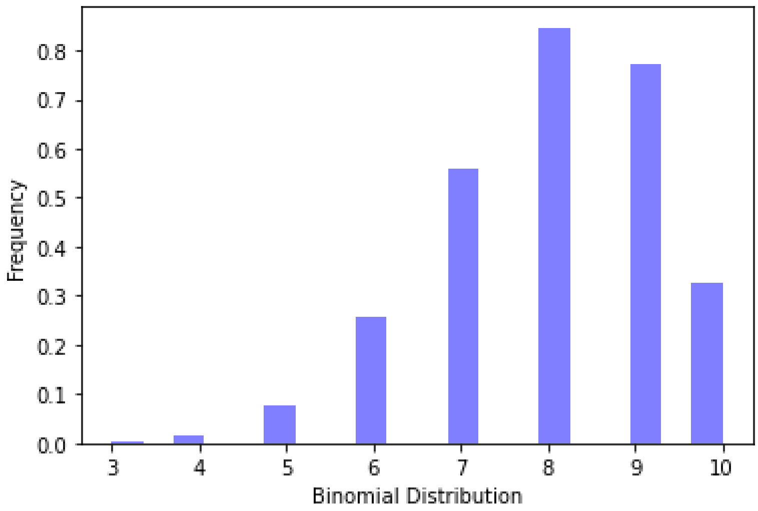
In [7]:

```
from scipy.stats import binom
import matplotlib.pyplot as plt

data_binom = binom.rvs(n=10, p=0.8, size=10000)

num_bins = 20
# histogram of the data
n, bins, patches = plt.hist(data_binom, num_bins, facecolor='blue', density=True, al

plt.xlabel('Binomial Distribution')
plt.ylabel('Frequency')
plt.show()
```



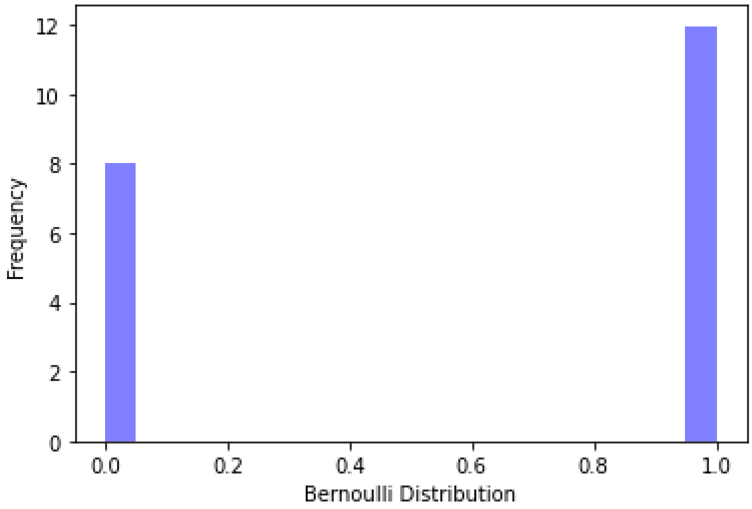
In [6]:

```
from scipy.stats import bernoulli
import matplotlib.pyplot as plt

data_bern = bernoulli.rvs(size=10000,p=0.6)

num_bins = 20
# histogram of the data
n, bins, patches = plt.hist(data_bern, num_bins, facecolor='blue', density=True, alp

plt.xlabel('Bernoulli Distribution')
plt.ylabel('Frequency')
plt.show()
```



In [ ]: