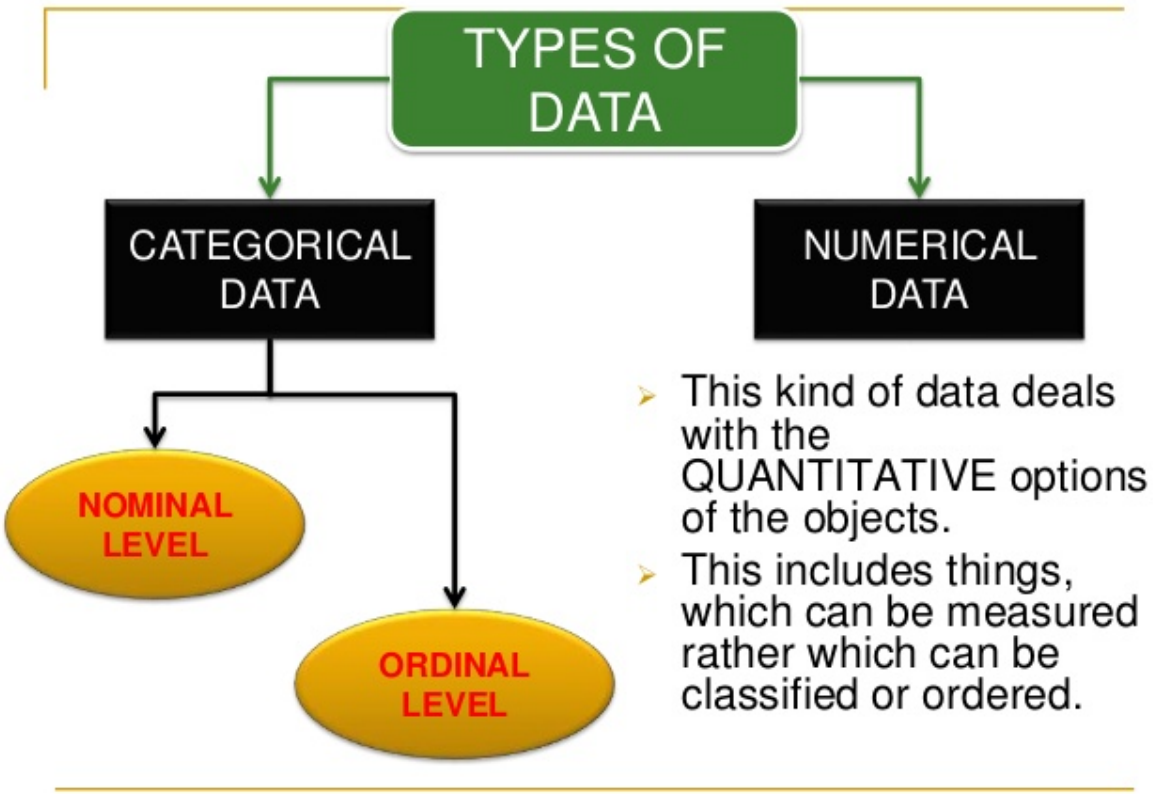
In [8]:
```
import numpy as np
import pandas as pd
import statistics as stat
```

In [2]:
```
from IPython.display import Image
Image (filename="E:/pyimages/typesofdata.jpg")
```

Out[2]:



Discrete Variable

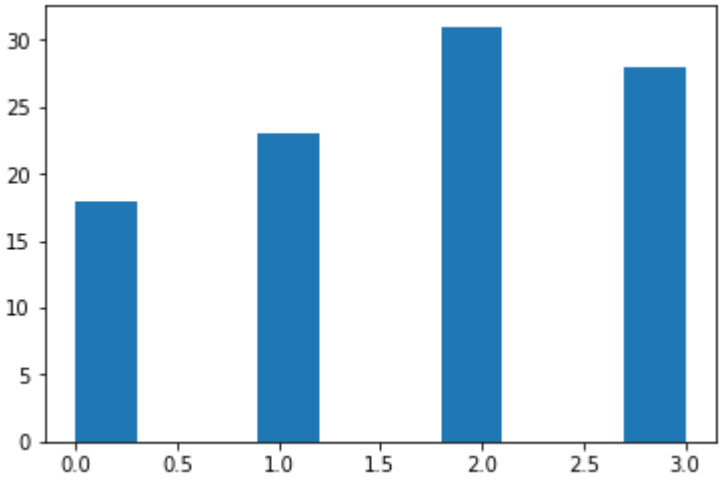
In [12]:
```
import matplotlib.pyplot as plt
```

In [3]:
```
np.unique(np.random.randint(0,4,100))
```

Out[3]:
```
array([0, 1, 2, 3])
```

In [13]:
```
plt.hist(np.random.randint(0,4,100))
```

Out[13]:
```
(array([18.,  0.,  0., 23.,  0.,  0., 31.,  0.,  0., 28.]),
 array([0. , 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3. ]),
 <a list of 10 Patch objects>)
```



Continuous Variable

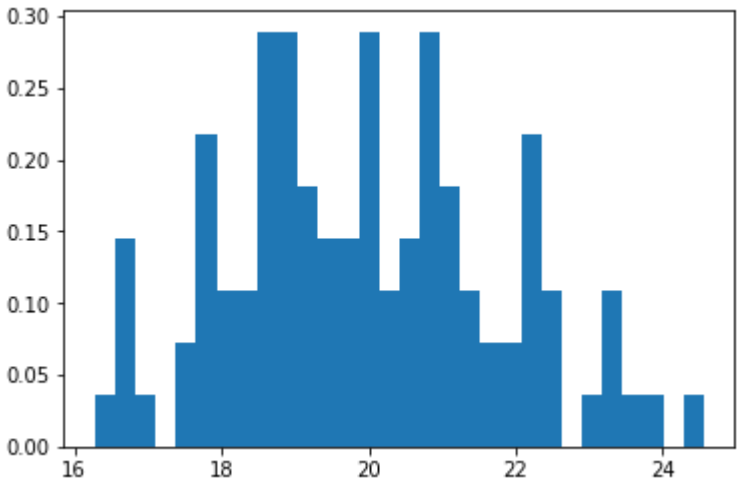
In [14]:
```
import scipy.stats as stats
```

In [15]:
```
nu=20
sigma =2
h=sorted(np.random.normal(nu,sigma,100)) # Draws random samples of guassian di
stribustion

abs(20-np.mean(h))<2
```

Out[15]:
```
True
```

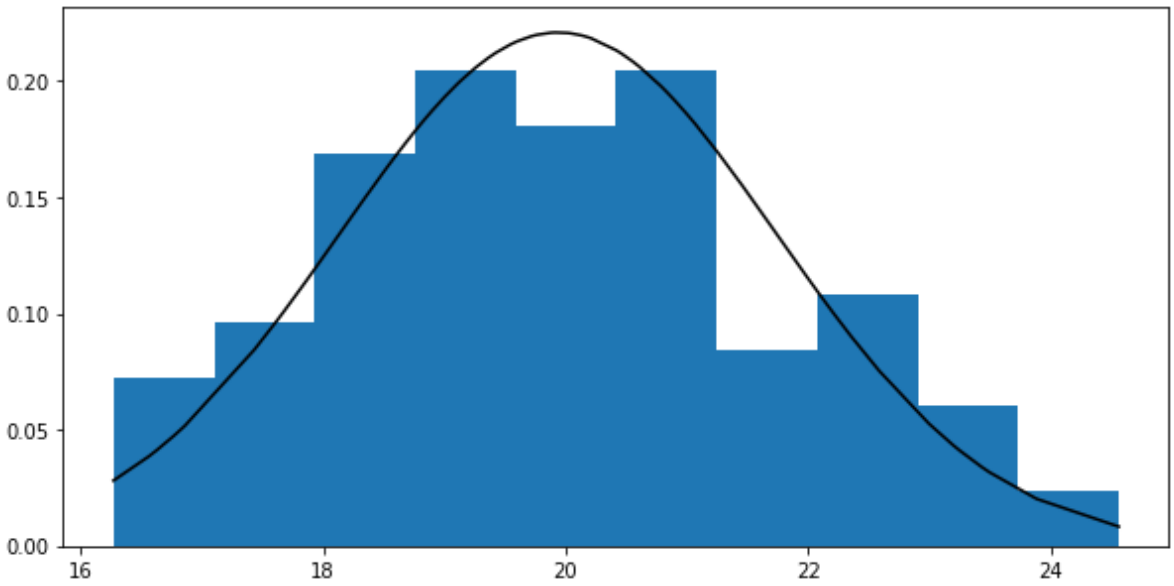
In [17]:
```
plt.hist(h,30,normed=True)
```

Out[17]:
```
(array([0.03616729, 0.14466915, 0.03616729, 0.        , 0.07233458,
        0.21700373, 0.10850186, 0.10850186, 0.2893383 , 0.2893383 ,
        0.18083644, 0.14466915, 0.14466915, 0.2893383 , 0.10850186,
        0.14466915, 0.2893383 , 0.18083644, 0.10850186, 0.07233458,
        0.07233458, 0.21700373, 0.10850186, 0.        , 0.03616729,
        0.10850186, 0.03616729, 0.03616729, 0.        , 0.03616729]),
 array([16.27062976, 16.5471227 , 16.82361565, 17.10010859, 17.37660154,
        17.65309449, 17.92958743, 18.20608038, 18.48257333, 18.75906627,
        19.03555922, 19.31205217, 19.58854511, 19.86503806, 20.141531  ,
        20.41802395, 20.6945169 , 20.97100984, 21.24750279, 21.52399574,
        21.80048868, 22.07698163, 22.35347457, 22.62996752, 22.90646047,
        23.18295341, 23.45944636, 23.73593931, 24.01243225, 24.2889252 ,
        24.56541814]),
 <a list of 30 Patch objects>)
```



In [23]:
```
import scipy.stats as stats
plt.figure(figsize=(10,5))
fit=stats.norm.pdf(h,np.mean(h),np.std(h))
plt.plot(h,fit,'-0')
plt.hist(h,normed=True)
```

Out[23]:
```
(array([0.07233458, 0.0964461 , 0.16878068, 0.20494796, 0.18083644,
        0.20494796, 0.08439034, 0.10850186, 0.06027881, 0.02411153]),
 array([16.27062976, 17.10010859, 17.92958743, 18.75906627, 19.58854511,
        20.41802395, 21.24750279, 22.07698163, 22.90646047, 23.73593931,
        24.56541814]),
 <a list of 10 Patch objects>)
```



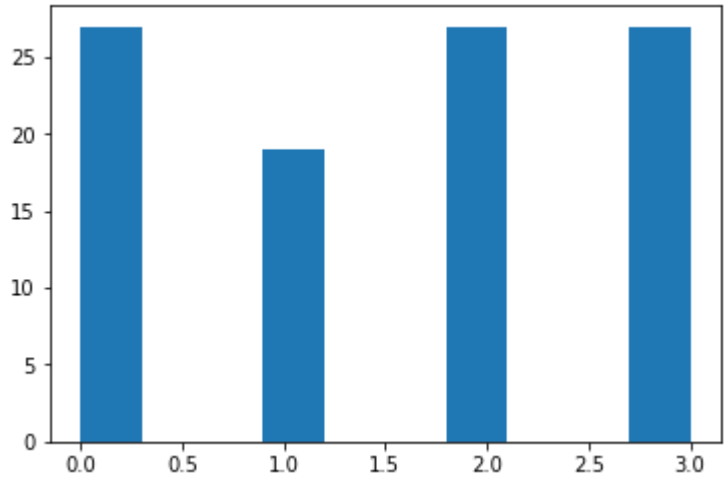
Discrete Variables

In [25]:
```
np.unique(np.random.randint(0,4,100))
```

Out[25]:
```
array([0, 1, 2, 3])
```

In [26]:
```
plt.hist(np.random.randint(0,4,100))
```

Out[26]:
```
(array([27.,  0.,  0., 19.,  0.,  0., 27.,  0.,  0., 27.]),
 array([0. , 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3. ]),
 <a list of 10 Patch objects>)
```



continuous variables

In [36]:
```
import pandas as pd
import numpy as np
```

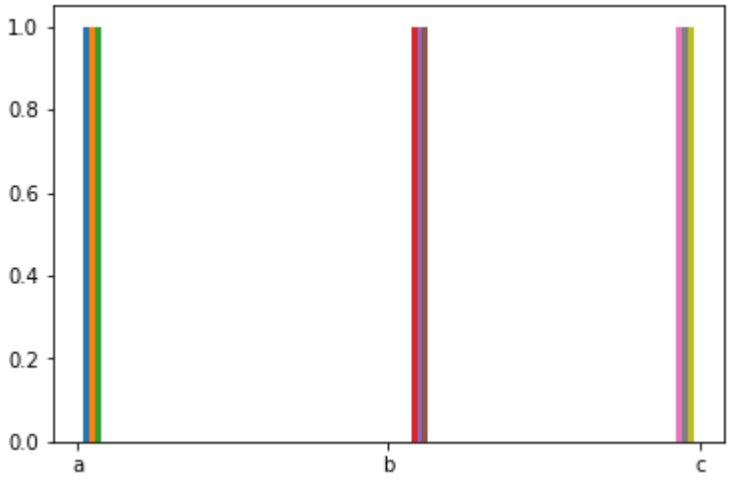
In [39]:
```
data = pd.DataFrame({'group':['a','a','a','b','b','b','c','c','c'],'ounces':[4
,3,12,6,7,5,8,3,5]})

data
```

Out[39]:

	group	ounces
0	a	4
1	a	3
2	a	12
3	b	6
4	b	7
5	b	5
6	c	8
7	c	3
8	c	5

In [43]:
```
plt.hist(data.group)
```

Out[43]:
```
([array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]),
  array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]),
  array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]),
  array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.]),
  array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.]),
  array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.]),
  array([0., 0., 0., 0., 0., 0., 0., 0., 0., 1.]),
  array([0., 0., 0., 0., 0., 0., 0., 0., 0., 1.]),
  array([0., 0., 0., 0., 0., 0., 0., 0., 0., 1.])],
 array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. ]),
 <a list of 9 Lists of Patches objects>)
```



Nominal Data:

In [47]:
```
data['rating'] = np.random.randint(0,5,9) # adding Row rating
```

In [46]:
```
data
```

Out[46]:

	group	ounces	rating
0	a	4	0
1	a	3	4
2	a	12	4
3	b	6	2
4	b	7	0
5	b	5	4
6	c	8	1
7	c	3	4
8	c	5	3

To read data using Excel

In [67]:
```
df=pd.read_excel("E:/pyimages/dataset2.xlsx")
```

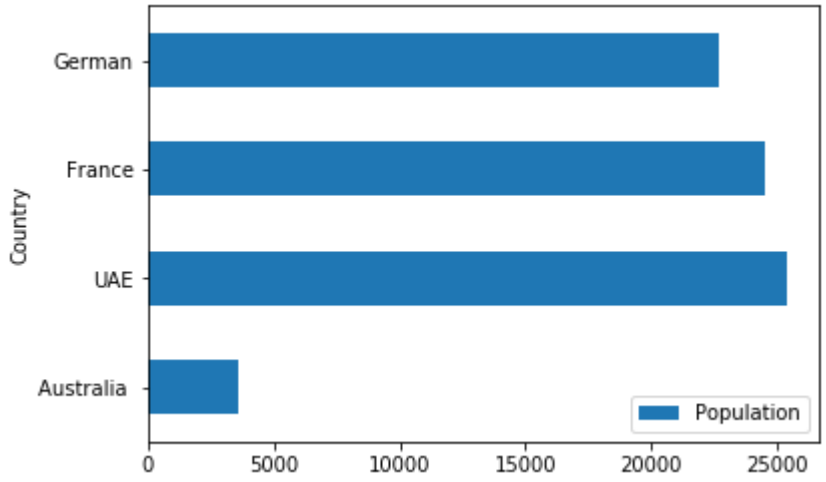
In [68]:
```
df.head()
```

Out[68]:

	S.No	Country	Population
0	1	India	32766
1	2	Japan	26360
2	3	USA	31322
3	4	Australia	3630
4	5	UAE	25365

In [71]:
```
df.tail(4).plot.barh('Country','Population')
```

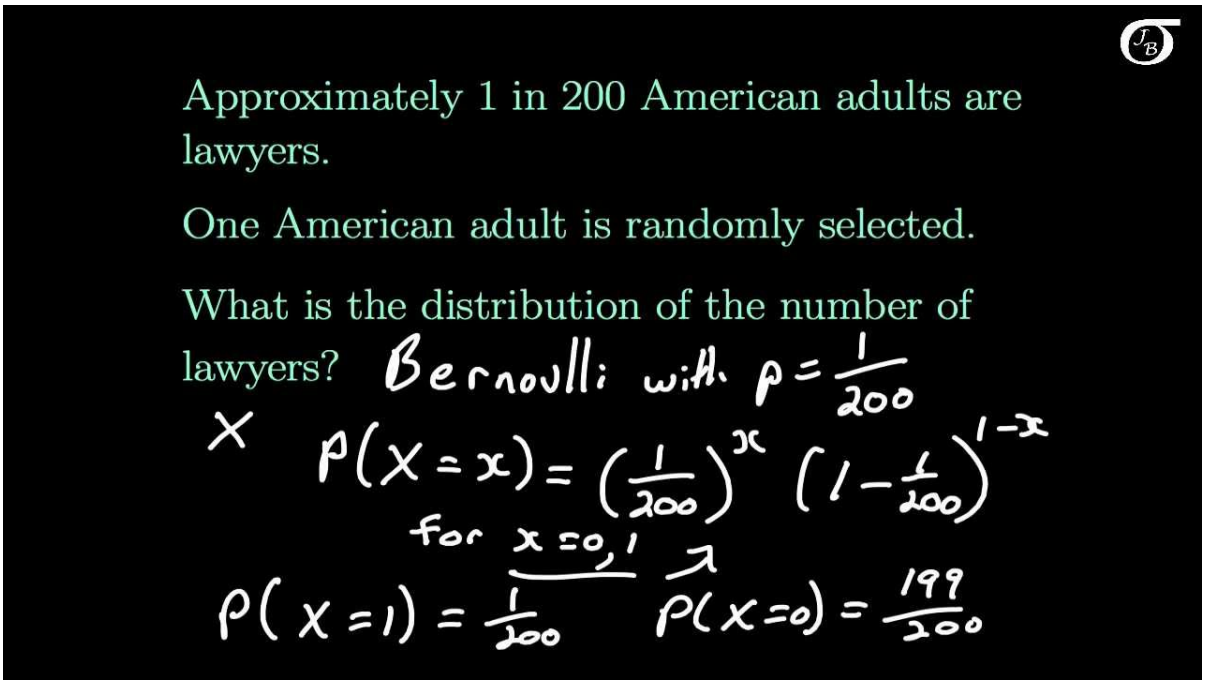
Out[71]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x20891e70cc0>
```



Bernoullis Distribution

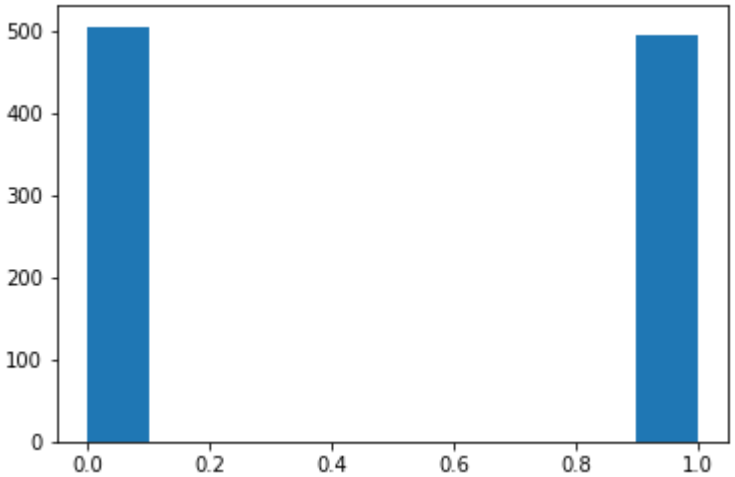
In [72]:
```
from IPython.display import Image
Image (filename="E:/pyimages/ber.png")
```

Out[72]:



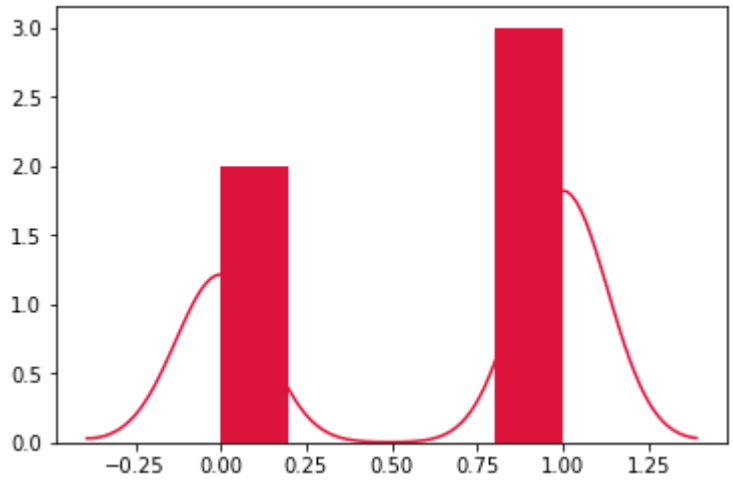
In [75]:
```
## Flipping a Coin

n,p = 1,0.5  # number of trails, probability of each trails

s=np.random.binomial(n,p,1000) # tested 100 times

plt.hist(s)
```

Out[75]:
```
(array([505.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0., 495.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



In [80]:
```
from scipy.stats import bernoulli
import seaborn as sb

bernoulli_data = bernoulli.rvs(size=1000,p=0.6) # probability is 0.6
ax= sb.distplot(bernoulli_data,
                kde=True,
                color='crimson',
                hist_kws={"linewidth":25,'alpha':1})
ax= set(x_label='bernoulli',y_label='Frequency')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-80-655391b0b219> in <module>
      7                color='crimson',
      8                hist_kws={"linewidth":25,'alpha':1})
----> 9 ax= set(x_label='bernoulli',y_label='Frequency')

TypeError: set() takes no keyword arguments
```

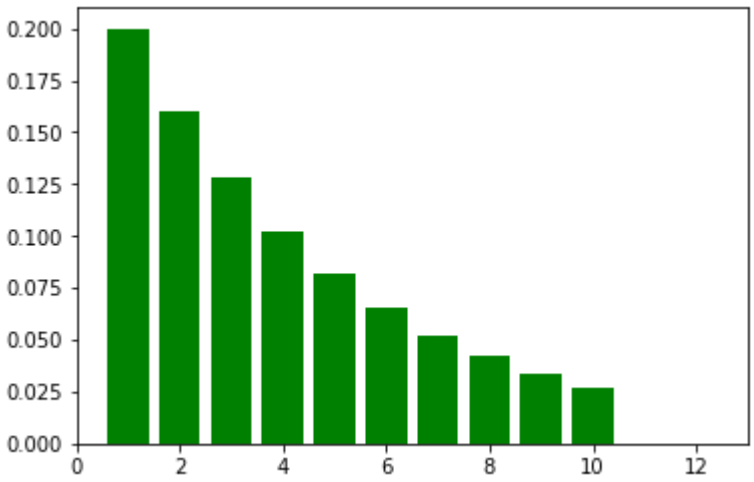


Geometric Distribution

In [87]:
```
import numpy as np
from scipy.stats import geom
```

In [88]:
```
prob=0.2
n=12
p=np.zeros(n)
q=np.zeros(n)
for k in range(1,n-1):
    p[k-1]=(1-prob)**(k-1)*prob
    q[k-1]=geom.pmf(k,prob)
plt.bar(range(1,13),p,color='green')
plt.xlin(1,15)
plt.xlabel('n')
plt.ylabel('p(n)')
plt.title("first sucess at n")
plt.show()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-88-6255dc810440> in <module>
      7     q[k-1]=geom.pmf(k,prob)
      8 plt.bar(range(1,13),p,color='green')
----> 9 plt.xlin(1,15)
     10 plt.xlabel('n')
     11 plt.ylabel('p(n)')

AttributeError: module 'matplotlib.pyplot' has no attribute 'xlin'
```



In [ ]: