CORRELATION AND COVARIENCE

In [1]:

```
from numpy.random import randn
from numpy.random import seed
from numpy import cov
seed(1)
data1=20*randn(1000)+100
data2=data1+(10*randn(1000)+50)
covarience=cov(data1,data2)
print(covarience)
```

```
[[385.33297729 389.7545618 ]
 [389.7545618  500.38006058]]
```

In [2]:

```
from numpy.random import randn
from numpy.random import seed
from scipy.stats import pearsonr
seed(1)
data1=20*randn(1000)+100
data2=data1+(10*randn(1000)+50)
corr,_=pearsonr(data1,data2)
print("pearsonr correelation:%.3f"%corr)
```

```
pearsonr correelation:0.888
```

In [4]:

```
import pandas as pd
import statsmodels as sm
from scipy import stats
from statsmodels.stats import weightstats as stests
```

In [6]:

```
data=[89,93,95,93,97,98,96,99,93,97,110,104,119,105,119,105,104,110,110,112,115,114]
z_test,p_val=stests.ztest(data,x2=None,value=160)
print(p_val)
if p_val<0.05:
    print("we can reject the null hypothesis")
else:
    print("we can accept the null hypothesis")
```

```
3.0151452189683223e-187
we can reject the null hypothesis
```

In [5]:

```
import scipy.stats
data1=[0.0842,0.0368,0.0847,0.0935,0.0376,0.0963,0.0684,0.0758,0.0854,0.0855]
data2=[0.0785,0.0845,0.0758,0.0853,0.0946,0.0785,0.0853,0.0685]
data3=[0.084,0.2522,0.0894,0.2724,0.083]
ftest,p_val=scipy.stats.f_oneway(data1,data2,data3)
print("p_value is:",p_val)
if p_val<0.05:
    print("we can reject null hypothesis")
else:
    print("we can accept null hypothesis")
```

```
p_value is: 0.010685939140579474
we can reject null hypothesis
```

CHI-SQUARE TEST

In [8]:

```
from scipy.stats import chi2_contingency
data=[[231,256,321],[245,312,213]]
test,p_val,dof,expected_val=chi2_contingency(data)
alpha=0.05
print("The p-value of our test is",str(p_val))
if p_val<0.05:
    print("we can reject null hypothesis")
else:
    print("we can accept null hypothesis")
```

```
The p-value of our test is 1.4585823594475804e-06
we can reject null hypothesis
```

In [6]:

```
from sklearn.metrics import mean_squared_error
y_act=[1,4,3,2,6]
y_pred=[0.6,1.29,1.99,2.69,3.4]
mean_squared_error(y_act,y_pred)
```

Out[6]:

```
3.15206
```

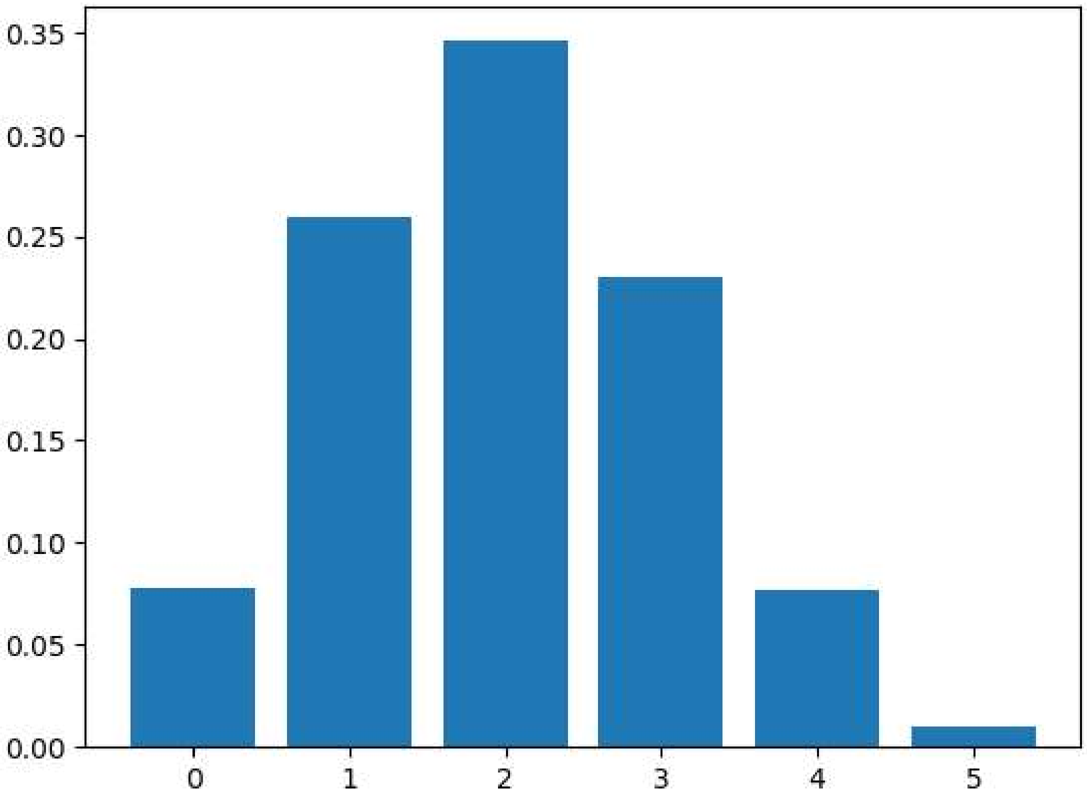
matpot

In [12]:

```
from scipy.stats import binom
import matplotlib.pyplot as plt
n=5
p=0.4
r_values=list(range(n+1))
dist=[binom.pmf(r,n,p) for r in r_values]
plt.bar(r_values,dist)
plt.show
```

Out[12]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

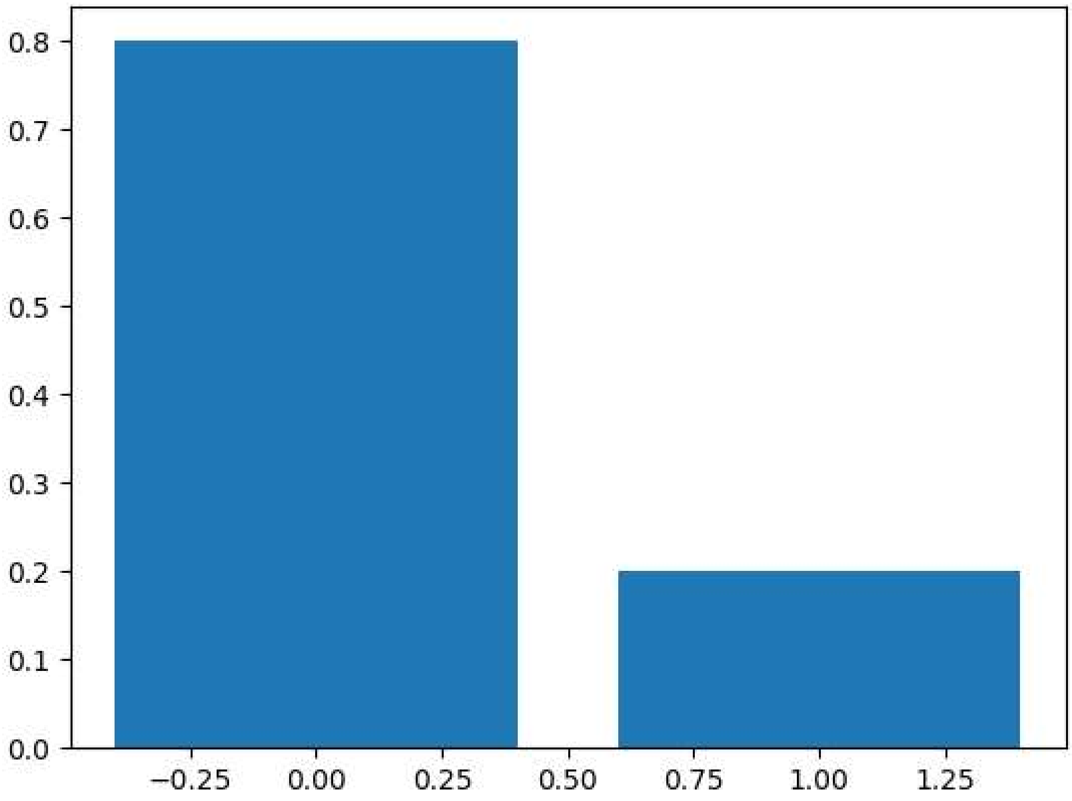


In [17]:

```
import matplotlib.pyplot as plt
from scipy.stats import bernoulli
bd=bernoulli(0.2)
x=[0,1]
plt.bar(x,bd.pmf(x))
plt.show
```

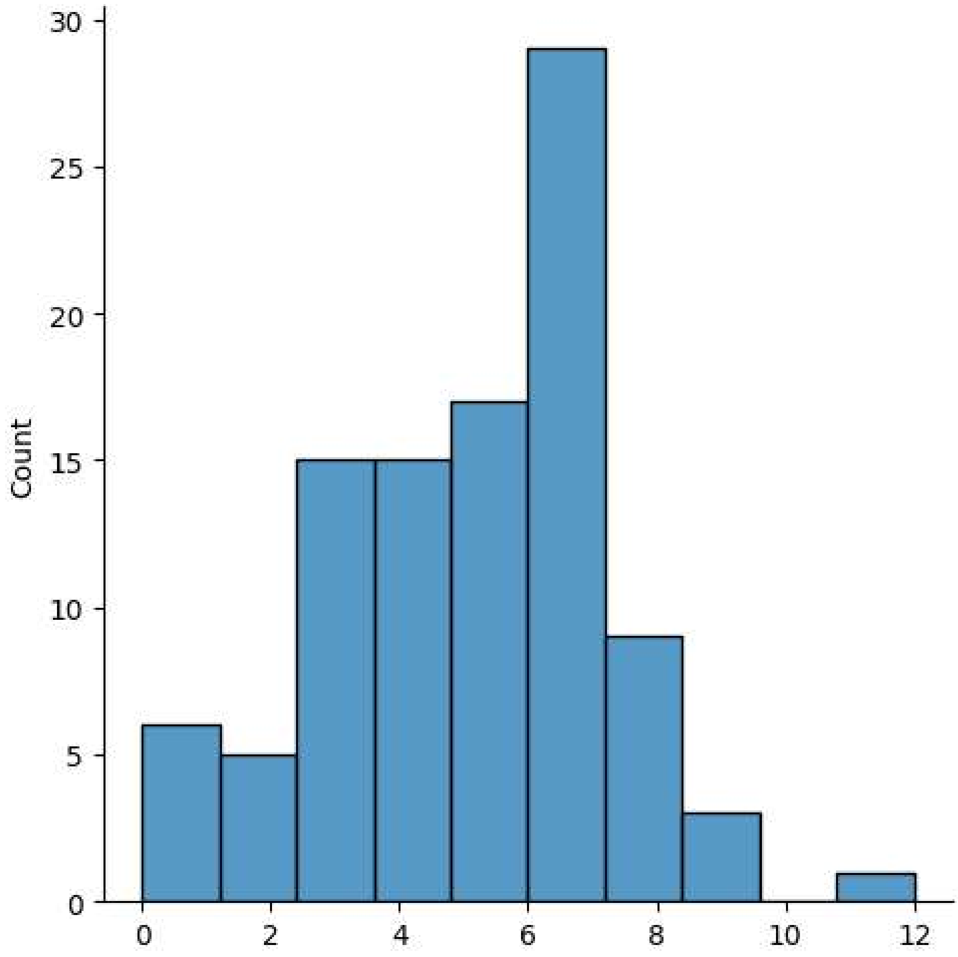
Out[17]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [20]:

```
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns
sns.displot(random.poisson(lam=5,size=100))
plt.show()
```

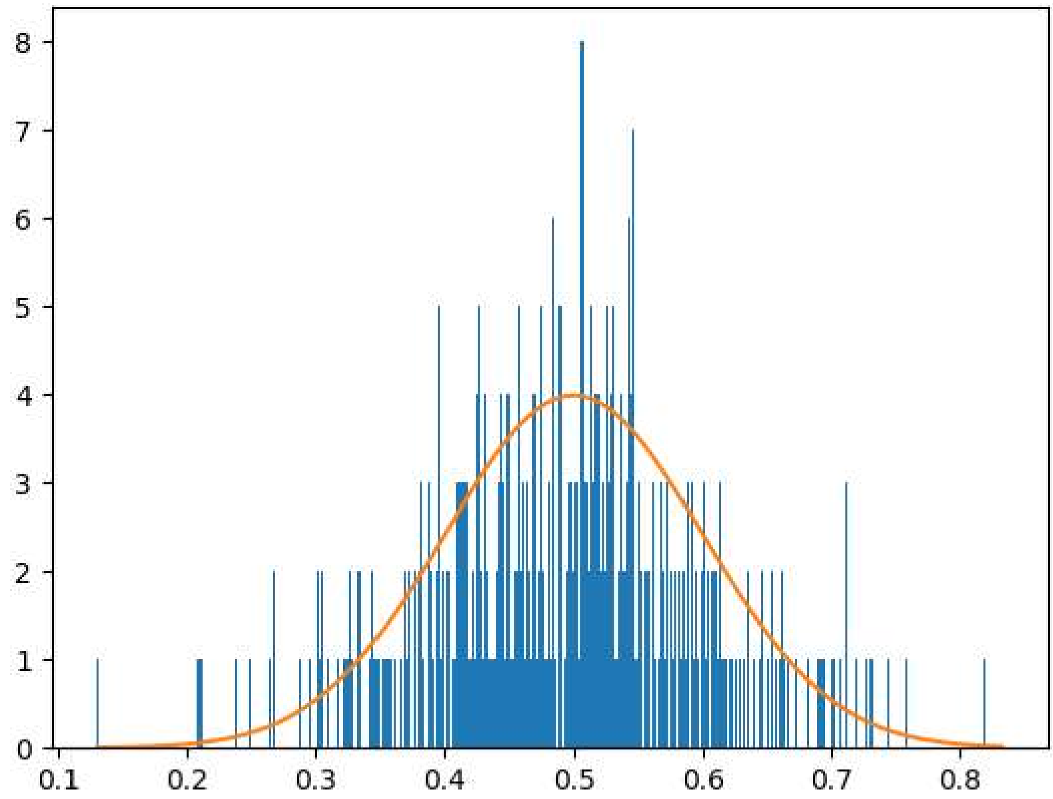


In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
mu,sigma=0.5,0.1
s=np.random.normal(mu,sigma,1000)
count,bins,ignored=plt.hist(s,1000)
plt.plot(bins,1/(sigma*np.sqrt(2*np.pi))*np.exp(-(bins-mu)**2/(2*sigma**2)))
plt.show
```

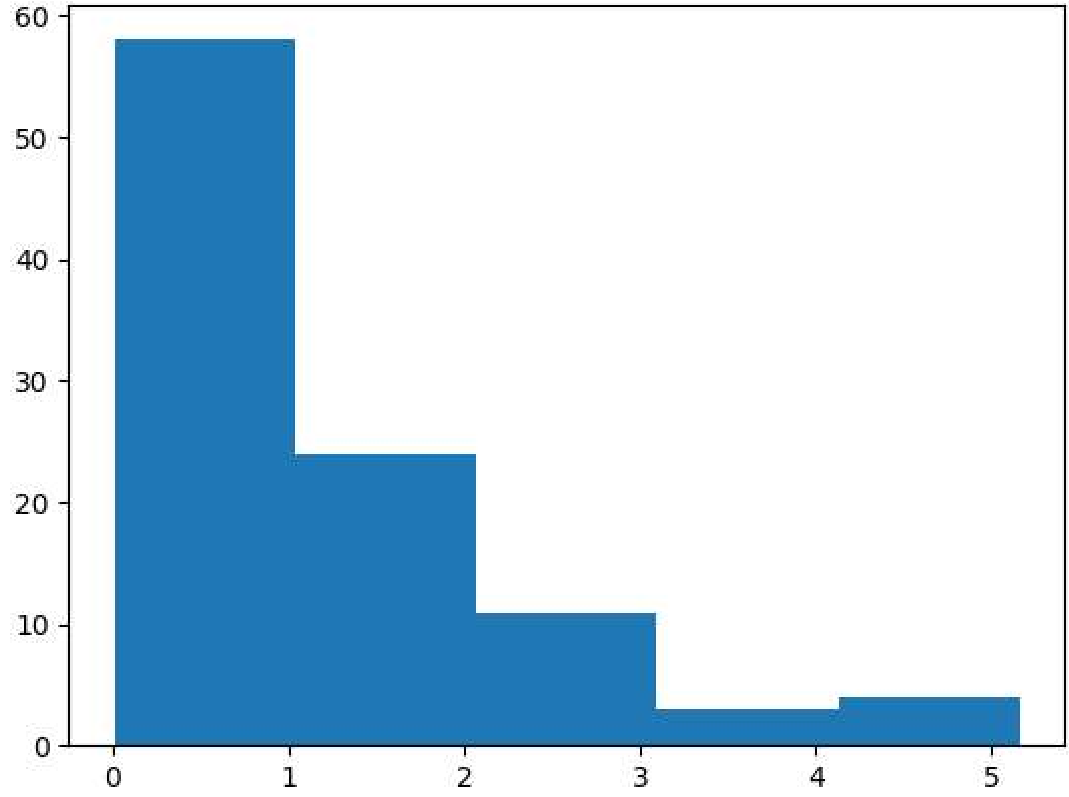
Out[1]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [3]:

```
import numpy as np
import matplotlib.pyplot as plt
exp=np.random.exponential(1,100)
count,bins,ignored=plt.hist(exp,5)
plt.show()
```



In [ ]:

In [ ]: