گزارش:

سوال 1 :

بخش اول:

-در قسمت ابتدایی ما کتابخانه هایی در طی پروژه لازم داریم را import میکنیم

```
import the library
```

```
import numpy as np
import matplotlib.pyplot as plt
```
[183]

-در این قسمت تابع نمونه برداری از توزیع دوجمله ای را بر مبنای توزیع برنولی را پیاده سازی میکنیم که این تابع m و n را به عنوان ورودی میگیرد و همچنین p به عنوان احتمال موفقیت در هر کدام از ازمایش های برنولی که در خرجی تابع لیستی از تعداد برد ها در هر کدام از ازمایش های دوجمله ای را به ما میدهد

ساخت تابع نمونه برداری

```
def randbin(n,m,p):
    x=np.random.choice(np.array([1,0]),(m,n),p=np.array([p, 1-p]))
    return np.sum(x,1)
```
[2] ✓ 0.0s Python

بخش دوم:

-در این قسمت از ما خواسته شده به ازای مقادیر ($m = ۵۰۰۰$) و $(n = ۵۰۰)$ میانگین و واریانس را به دو روش تئوری و عملی به دست بیاریم و با هم مقایسه کنیم

میانگین(exp):

```
#practical exp
z=np.array([])
for p in range (0,100):
    a=randbin(500,5000,p/100)
    z = np.append(z, np.average(a))
print(np.average(z))

# theory exp
sum=0
for p in range (0,100):
    sum+=500*(p/100)
exp=sum/100
print(exp)
```

[5] ✓ 7.0s

```
247.48841199999998
247.5
```

واریانس(var):

```
# practical var
k=np.array([])
for p in range (0,100):
    b=randbin(500,5000,p/100)
    for i in b:
        gashtavar=np.array([])
        gashtavar=np.append(gashtavar,(b-np.average(b))*(b-np.average(b)))
    k=np.append(k,np.average(gashtavar))
print(np.average(k))

# theory var
sumvar=0
for p in range (0,100):
    sumvar+=500*(p/100)*(1-(p/100))
var=sumvar/100
print(var)
```

[6] ✓ 45.0s

```
83.50089266
83.325
```

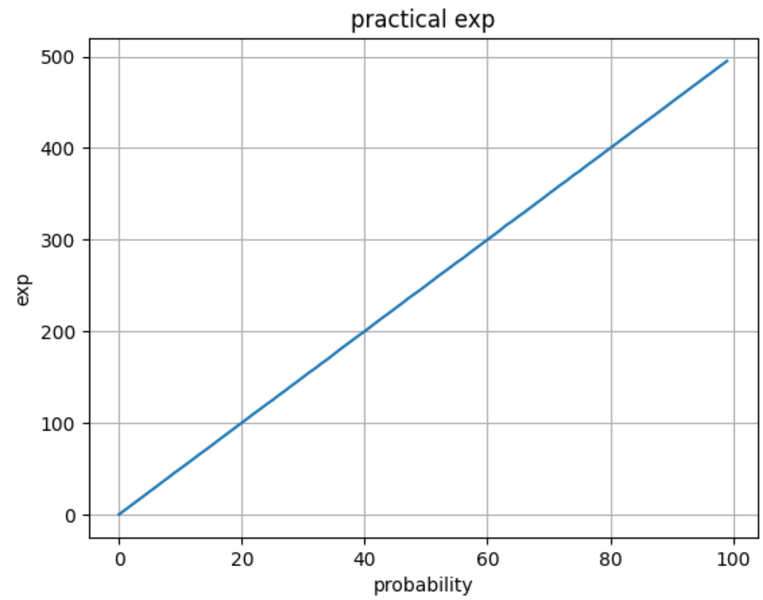
بخش سوم:

–در این قسمت با استفاده از matplotlib نمودار های مربوط به مقادیر تئوری و عملی میانگین و واریانس را رسم میکنیم

میانگین(exp):

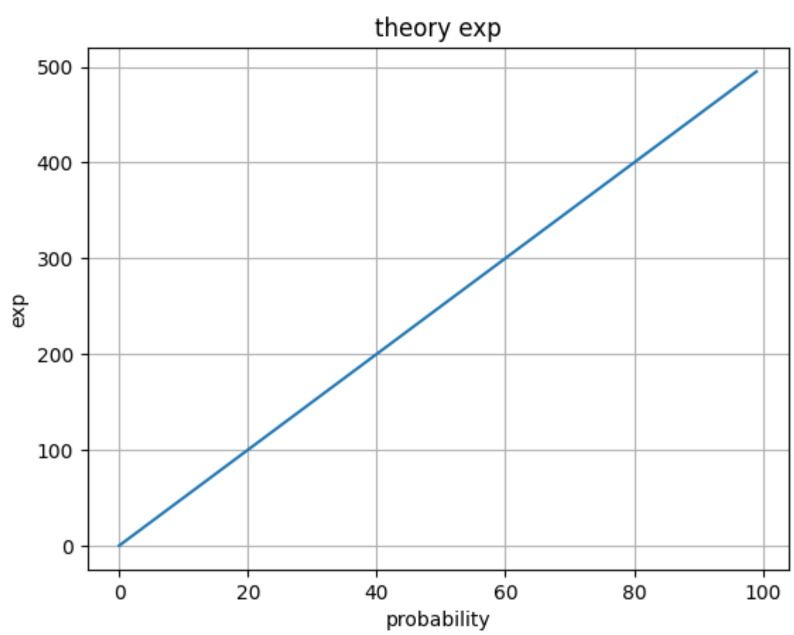
```python
x=np.array([])
for p in range (0,100):
    x = np.append(x,p)
y = z

plt.title("practical exp")
plt.xlabel("probability")
plt.ylabel("exp")

plt.plot(x, y)
plt.grid()
plt.show()
```

✓ 0.1s



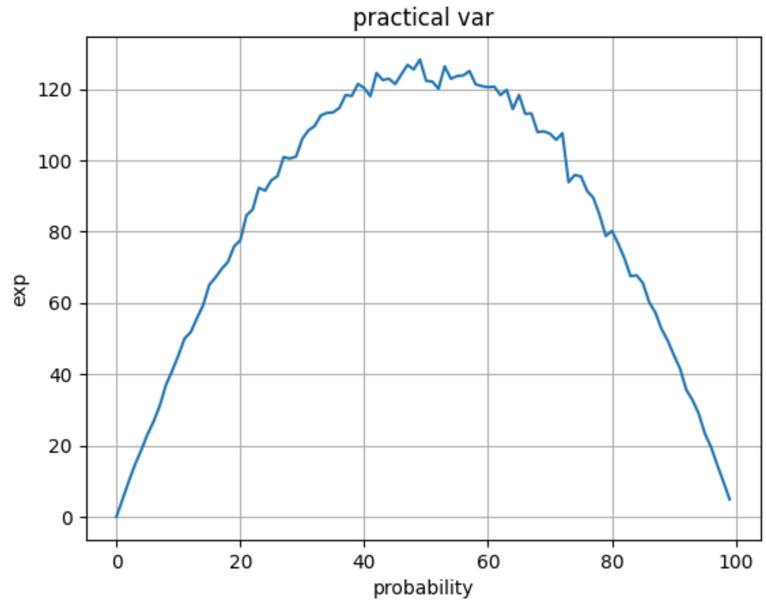

```python
x=np.array([])
y=np.array([])
for p in range (0,100):
    x = np.append(x,p)
    y = np.append(y,500*(p/100))

plt.title("theory exp")
plt.xlabel("probability")
plt.ylabel("exp")

plt.plot(x, y)
plt.grid()
plt.show()
```

✓ 0.2s

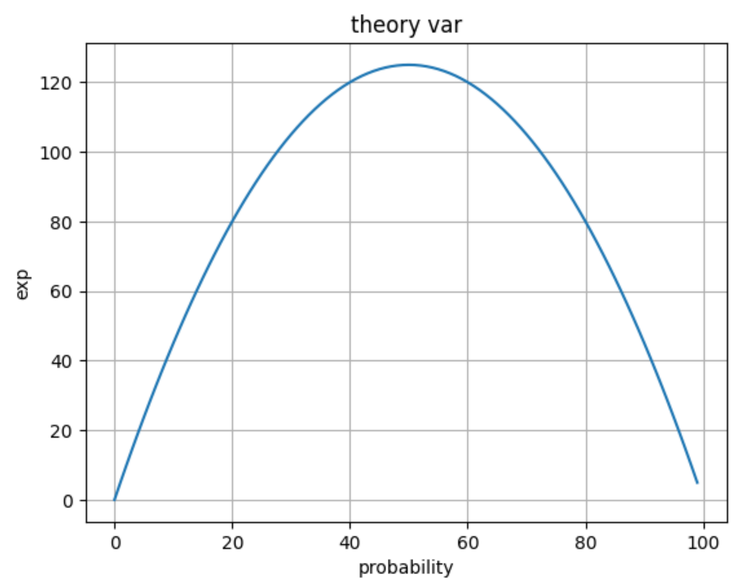


واریانس (var):

```
x=np.array([])
for p in range (0,100):
    x = np.append(x,p)
y = k

plt.title("practical var")
plt.xlabel("probability")
plt.ylabel("exp")

plt.plot(x, y)

plt.grid()

plt.show()
```




```
x=np.array([])
y=np.array([])
for p in range (0,100):
    x = np.append(x,p)
    y = np.append(y,500*(p/100)*(1-(p/100)))

plt.title("theory var")
plt.xlabel("probability")
plt.ylabel("exp")

plt.plot(x, y)

plt.grid()

plt.show()
```



بخش چهارم:

-با توجه به نمودار ها و مقادیر بدست امده نتیجه میگیریم از انجا که تعداد ازمایش های تصادفی ما بسیار زیاد است مقدار عملی و تئوری واریانس و میانگین ما هم با تقریب خوبی یکی میشوند اینبه این معناست که اگر یک اطمایش تصادفی را بیشمار بار انجام دهیم احتمال هرکدام از فضای نمونه به مقدار ریاضی ان نزدیک میشود

سوال 2:

بخش اول:

-در قسمت ابتدایی ما کتابخانه هایی در طی پروژه لازم داریم را import میکنیم

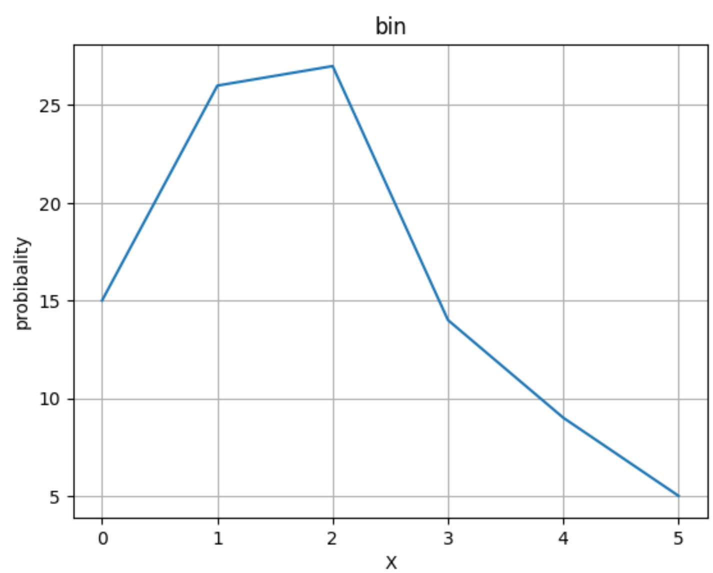
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import poisson
```

[303]

–در قسمت بعدی نمودار توزیع دوجمله ای را رسم میکنیم

```
X=np.random.binomial(250,0.008,100)

x=np.array([0,1,2,3,4,5])
y=np.array([])

for i in range(0,6):
    y = np.append(y,np.count_nonzero(X == i))

plt.title("bin")
plt.xlabel("X")
plt.ylabel("probibality")

plt.plot(x, y)
plt.grid()
plt.show()
```

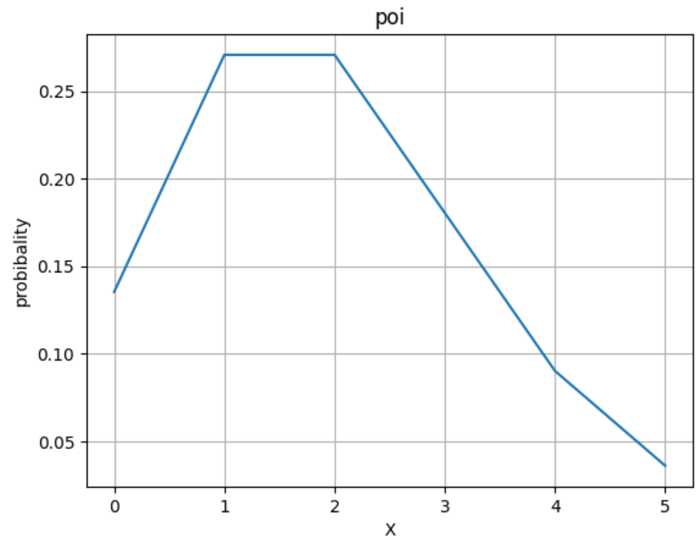
4] ✓ 0.1s



-حال نمودار توزیع پواسون را رسم میکنیم

```python
x=np.arange(6)
y=poisson.pmf(x , mu=2,loc=0)
plt.title("poi")
plt.xlabel("X")
plt.ylabel("probibality")

plt.plot(x, y)
plt.grid()
plt.show()
```

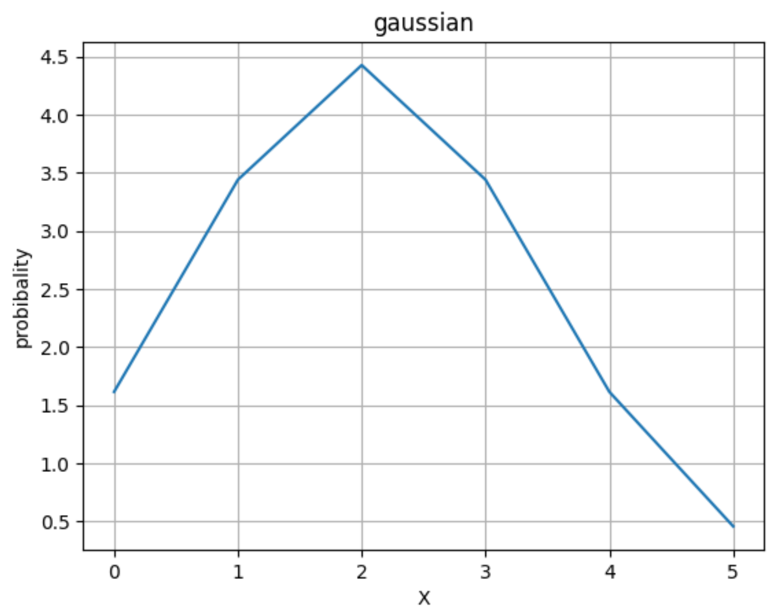
✓ 0.1s



-حال نمودار توزیع نرمال(گاوسی)را رسم میکنیم

```python
def normal_dist(x, mean, sd):
    prob_density = (np.pi*sd) * np.exp(-0.5*((x-mean)/sd)**2)
    return prob_density

x=np.arange(6)
y=np.array([])
for i in range(0,6):
    y=np.append(y,normal_dist(i,2,1.4085453489))

plt.title("gaussian")
plt.xlabel("X")
plt.ylabel("probibality")

plt.plot(x, y)
plt.grid()
plt.show()
```

✓ 0.2s



بخش دوم:

-همانطور که در نمودار ها مشاهده میکنید تقریب پواسون تقریب بهتری نسبت به نرمال است و به توزیع دوجمله ای بیشتر شبیه است

سوال ۳:

بخش اول:

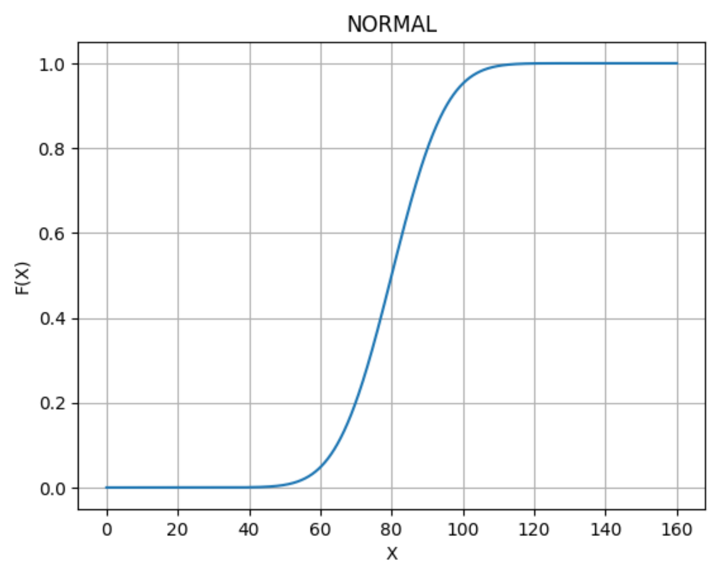
-در قسمت ابتدایی ما کتابخانه هایی در طی پروژه لازم داریم را import میکنیم

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
```

[1] ✓ 1.1s

-در این قسمت برای اینکه دید بهتری به مسئله داشته باشیم نمودار cdf توضیع نورمال را رسم میکنیم

```
x=np.arange(0,160,0.0001)
y=norm.cdf(x,loc=80,scale=12)

plt.title("NORMAL")
plt.xlabel("X")
plt.ylabel("F(X)")

plt.plot(x, y)
plt.grid()
plt.show()
```



-حال بارای قسمت اول سوال کافیست اولین نمره ای که cdf ان بزرگ تر از 0.9 شده را بدست بیاریم

```
for i in range(int(len(x)/2),len(x)):
    if(y[i]>0.9):
        print(x[i])
        break
```

```
95.37870000000001
```

بخش دوم:

-حال برای قسمت دوم سوال کافیست اولین نمره بالای 0.5 و اخرین نمره پایین 0.75 را بدست بیاریم

```
for i in range(0,len(x)):
    if(0.5<y[i]):
        print(x[i])
        break
for i in range(0,len(x)):
    if(0.75<y[i]):
        print(x[i])
        break
```

```
80.0001
88.0939
```

بخش سوم:

-برای حل قسمت ۳ از انجا که احتمال اینکه نمره بین ۸۰ تا ۹۰ باشد را میخواهد کافیست cdf(90)-cdf(80) بکنیم

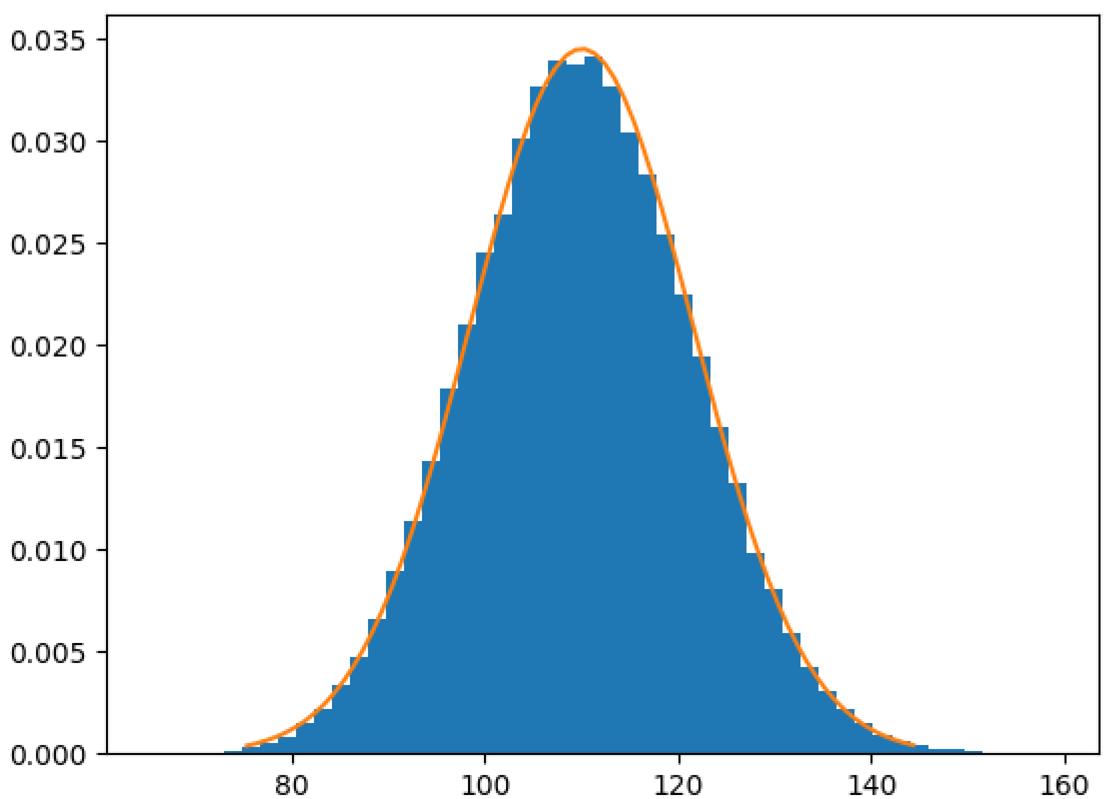
```
for i in range(0,len(x)):
    if(x[i]==80):
        m=i
        break
for i in range(0,len(x)):
    if(x[i]==90):
        n=i
        break
print(y[n]-y[m])
```

```
0.2976716190363568
```

بخش چهارم(امتیازی)

```
n=200
p=0.5
sample_size=100000

physic=np.random.uniform(low=0,high=20,size=sample_size)
ap=np.random.exponential(scale=1/(n*p),size=sample_size)
dm=np.random.poisson(lam=n*p,size=sample_size)
total=physic+ap+dm
plt.hist(total,bins=50,density=True)

x=np.arange(np.mean(total)-3*np.std(total),np.mean(total)+3*np.std(total))
plt.plot(x, norm.pdf(x, np.mean(total), np.std(total)))
plt.show()
```

✓ 0.2s



-همونطور که مشاهده میکنید با فرضیات داده شده در اول کد و اجرا کردن ان خواهیم دید که ترکیب توزیعات به به توزیع نورمال شبیه است که با افزایش مقدار n و کوچکتر کردن بازه هیستوگرام خواهیم دید این تقریب بیشتر هم میشود

سوال ۴:

بخش ۱:

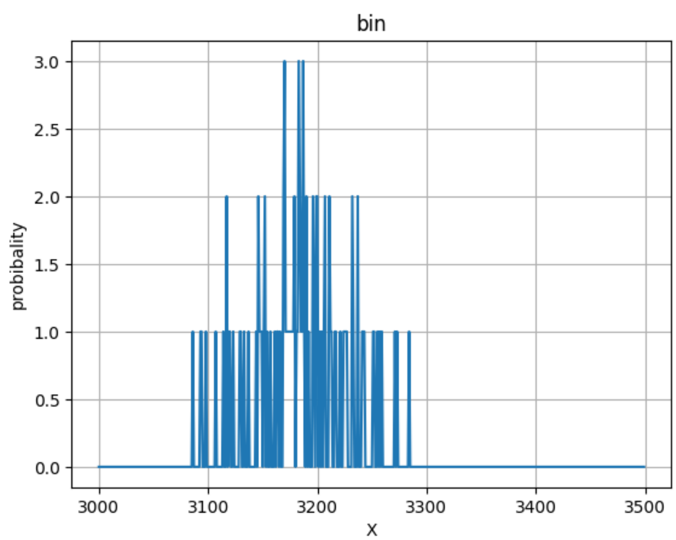
-در قسمت ابتدایی ما کتابخانه هایی که لازم داریم را import میکنیم

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import poisson
```

روش اول حل:بدست اوردن جدای هر کدام از نمودار ها

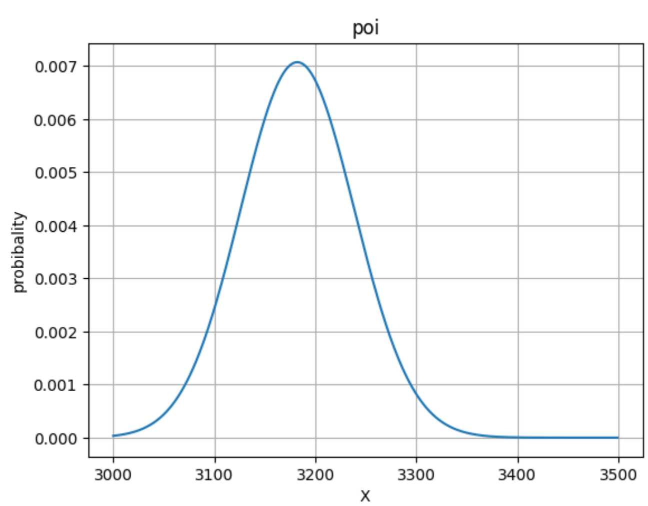
-برای توزیع دوجمله ای داریم

```
X=np.random.binomial(7072,0.45,100)

x=np.arange(3000,3500)
y=np.array([])

for i in range(3000,3500):
    y = np.append(y,np.count_nonzero(X == i))

plt.title("bin")
plt.xlabel("X")
plt.ylabel("probibality")
plt.plot(x, y)
plt.grid()
plt.show()
```



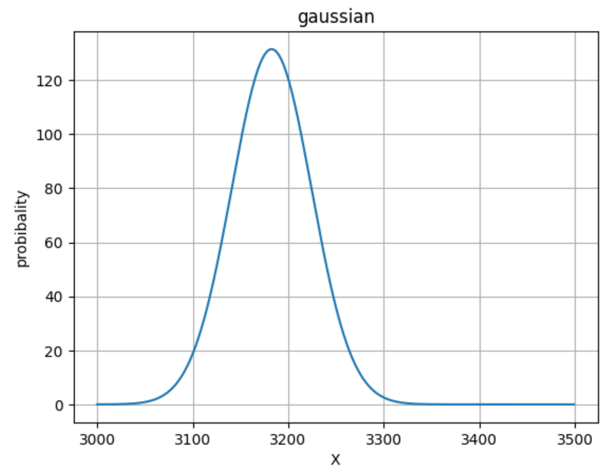
-برای توزیع پواسون داریم

```
x=np.arange(3000,3500)
y=poisson.pmf(x , mu=3182.59,loc=0)
plt.title("poi")
plt.xlabel("X")
plt.ylabel("probibality")
plt.plot(x, y)
plt.grid()
plt.show()
```



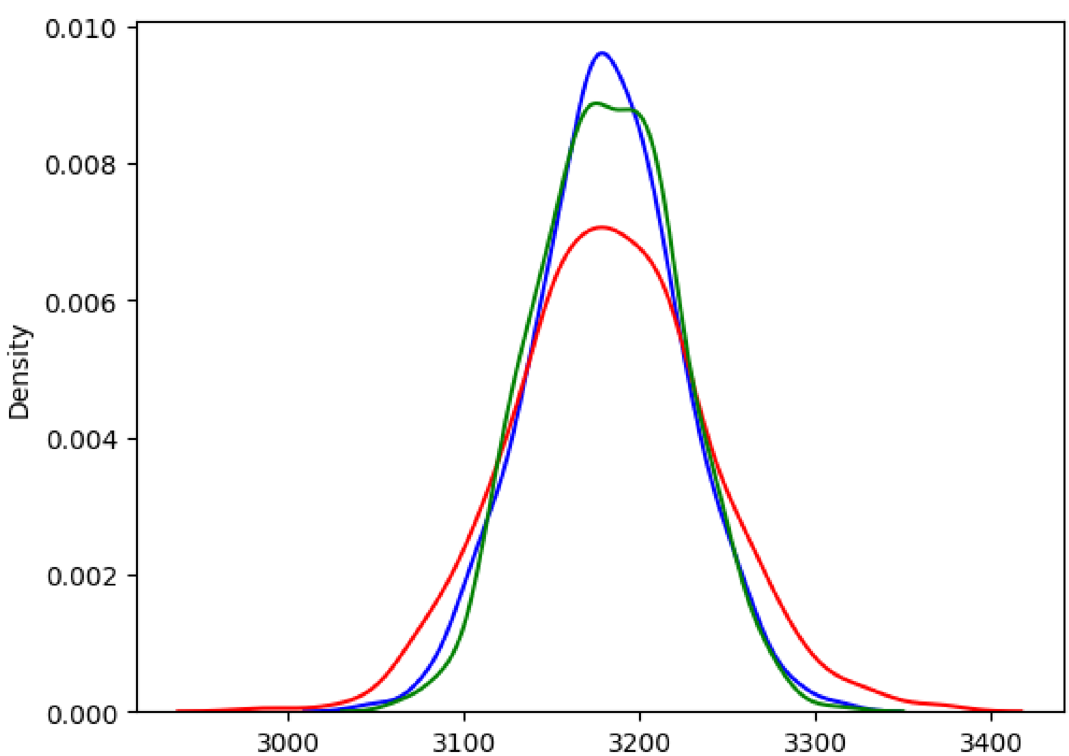
-برای توزیع نرمال داریم

```python
def normal_dist(x, mean, sd):
    prob_density = (np.pi*sd) * np.exp(-0.5*((x-mean)/sd)**2)
    return prob_density

x=np.arange(3000,3500)
y=np.array([])
for i in range(3000,3500):
    y=np.append(y,normal_dist(i,3182.59,41.84))

plt.title("gaussian")
plt.xlabel("X")
plt.ylabel("probibality")
plt.plot(x, y)
plt.grid()
plt.show()
```



روش دوم حل:بدست اوردن هر ۳ نمودار با seaborn

```python
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns

sns.distplot(random.binomial(n=7072, p=0.45, size=1000), hist=False,color='Blue', label='binomial')
sns.distplot(random.poisson(lam=3182.59, size=1000), hist=False,color='red', label='poisson')
sns.distplot(random.normal(loc=3182.59, scale=41.84 , size=1000), hist=False,color='green', label='normal')

plt.show()
```



بخش۲:

-همانطور که در نمودار های شکل بالا میبینیم توزیع احتمال نرمال تقریب بهتری نسبت به توزیع پواسون انجام داده و به توزیع دوجمله ای نزدیک تر است بنابر این میتوان گفت تقریب نرمال سازگار تر است

با تشکر از توجه شما ☺