به نام خدا

درس : آمار و احتمال مهندسی

نام : محسن کمال آبادی فراهانی

شماره دانشجویی : ۹۹۱۰۲۰۸۳

گزارش تمرین کامپیوتری دوم

.۱

قسمت اول :

کد:

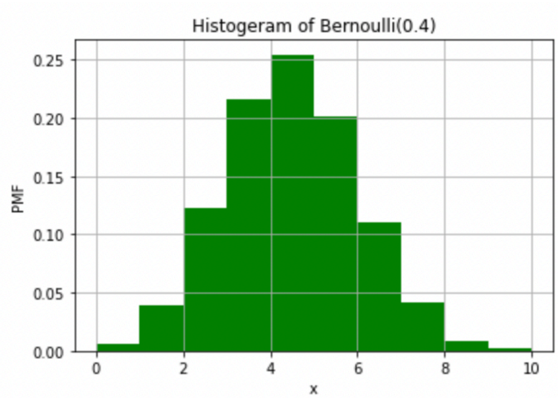
```
In [50]: import matplotlib.pyplot as plt
         from scipy.stats import bernoulli
         y1=[]
         for i in range (10000):
             s=0
             for j in range (10) :
                 x=bernoulli.rvs(0.4)
                 s=s+x
             y1.append(s)

         plt.hist(y1,color="g",density="true")
         plt.title("Histogeram of Bernoulli(0.4)")
         plt.xlabel("x")
         plt.ylabel("PMF")
         plt.grid(True)
         plt.show()
```

ما برای این قسمت از تابع Bernoulli.rvs(p) استفاده می کنیم که با توجه به عددی که به ورودی اش وارد می شود در خروجی به ما ۱ یا ۰ می دهد و طبعا هر چه p بزرگتر (به ۱ نزدیک تر) باشد بیشتر خروجی ۱ می دهد و بالعکس. از آنجا که ۱۰۰۰ بار باید ۱۰ تا متغیر تصادفی برنولی محاسبه کنیم دو تا حلقه فور (تو در تو) می زنیم و هر ۱۰ تا متغیر تصادفی که در حلقه دوم ایجاد می شود را با هم جمع می کنیم و هر بار آن مجموع (s) را به لیست y اضافه می کنیم و در آخر هیستوگرام y را رسم می کنیم که به ما می گوید احتمال رخداد p چگونه است (به ازای p های مختلف تعداد رخداد آن ها را چاپ می کند)

نتیجه :

همانگونه که مشاهده می شود این توزیع شبیه توزیع (Binomial (n=۱۰۰۰ , p= ۰.۴ می باشد



قسمت دوم:

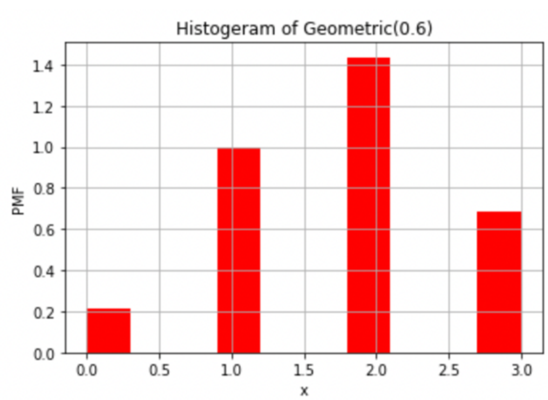
کد:

```
In [2]: import matplotlib.pyplot as plt
        from scipy.stats import bernoulli
        import numpy as np
        z=[]
        y2=[]
        for i in range (1000) :
            z = np.random.geometric(p=0.6, size=3)
            s2=0
            for j in range (3) :
                if z[j]==1 :
                    s2=s2+ 1
            y2.append(s2)

        plt.hist(y2,color="r",density="true")
        plt.title("Histogeram of Geometric(0.6)")
        plt.xlabel("x")
        plt.ylabel("PMF")
        plt.grid(True)
        plt.show()
```

فرق این قسمت با قسمت اول این است که به جای bernoulli از geometric استفاده می کنیم این تابع از ما دو ورودی احتمال و تعداد می خواهد از آنجا که سوال خواستار تولید سه تا داده می باشد تعداد را سه در نظر می گیریم . به ازای تولید هر یک ، یکی به مجموع اضافه می کنیم و سپس مجموع را تقسیم بر ۳ کرده و به ۲y اضافه کنیم و سپس دوباره داده تولید می کنیم و این فرایند را ۱۰۰۰۰ تکرار می کنیم و در آخرهیستوگرام y۲ را رسم میکنیم.

نتیجه :



.۲

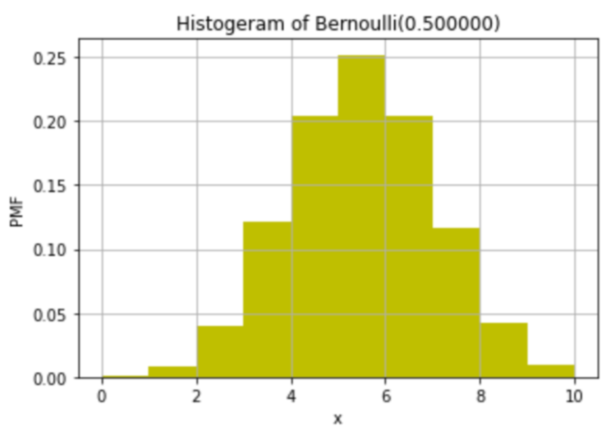
کد:

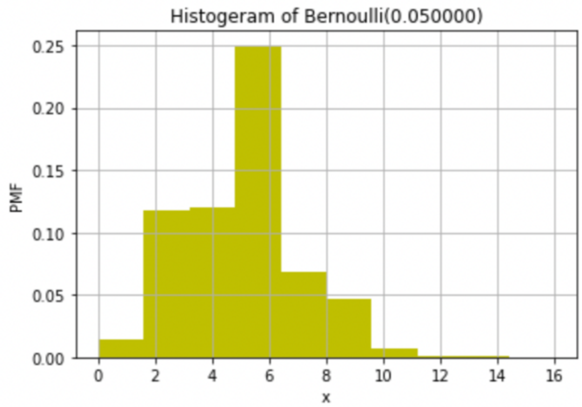
```
In [*]: import matplotlib.pyplot as plt
        from scipy.stats import bernoulli
        y=[]
        list_n=[10,100,1000,10000,100000]
        for i in range(len(list_n)) :
            n=list_n[i]
            p=5/n
            for i in range (10000):
                s=0
                for j in range (n) :
                    x=bernoulli.rvs(p)
                    s=s+x
                y.append(s)
            plt.hist(y,color="y",density="true")
            plt.title("Histogeram of Bernoulli(%f)" %p )
            plt.xlabel("x")
            plt.ylabel("PMF")
            plt.grid(True)
            plt.show()
```

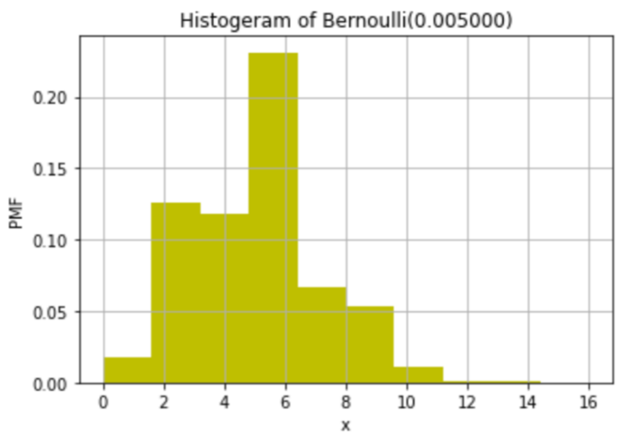
در این سوال قصد داریم تابع هیستوگرام برنولی را برای ۵ ورودی متفاوت را رسم کنیم و از هر n به اندازه خودش داده تولید کنیم و در آخر تقسیم بر خودش کنیم و این فرایند را ۱۰۰۰۰ بار تکرار کنیم و هر بار مجموع تقسیم بر تعداد را به لیست y اضافه کنیم و در پایان نمودار هیستوگرام y را رسم می کنیم

نتایج:

با توجه به نمودار ها مقادیر PMF به ۵ همگرا می شوند و با افزایش n امید ریاضی به ۵ و واریانس به صفر همگرا می شود.







۳.
قسمت اول:
کد:

```
In [8]: import matplotlib.pyplot as plt
        from scipy.stats import bernoulli
        import numpy as np
        w=[]
        for i in range (10000):
            x_1=np.random.binomial(5,0.4)
            x_2=np.random.poisson(1.6)
            x_3=np.random.geometric(0.1)
            w.append((x_1+x_2+x_3))
        plt.hist(w,color="r",density="True")
        plt.xlabel('x_1 + x_2 + x_3')
        plt.ylabel("Number")
        plt.grid(True)
        plt.show()
```

در این سوال در هر فرایند سه داده تصادفی به ترتیب باینومیال ، پواسون و جئومتریک تولید می کنیم ، هر سه را با هم جمع می کنیم و به لیست w اضافه می کنیم و این فرایند را ۱۰۰۰۰ تکرار میکنیم.

امید ریاضی:

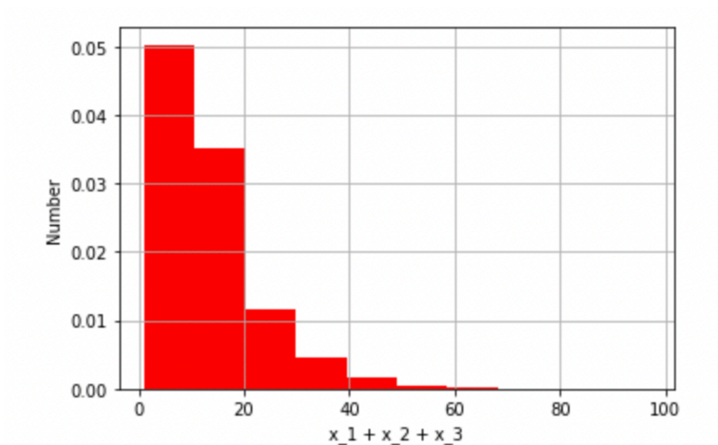
y = 13.6
Binomial = n.p = $5 \times 0.4 = 2$
Poisson = λ = 1.6
Geometric = 1/p = 10

واریانس :

y = 902.8
Binomail = n.p(1-p) = 5 $5 \times 0.4 \times 0.6 = 1.2$
Poisson = λ = 1.6
Geometric = (1-p)/p^2 =0.9/0.01 = 900

امید ریا ضی y برابر است با جمع هر سه امید ریاضی و واریانس y هم همینطور جمع هر سه واریانس است .

نتیجه :



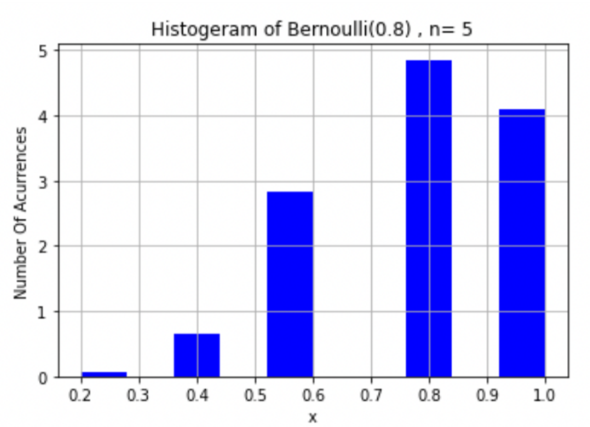
قسمت دوم:

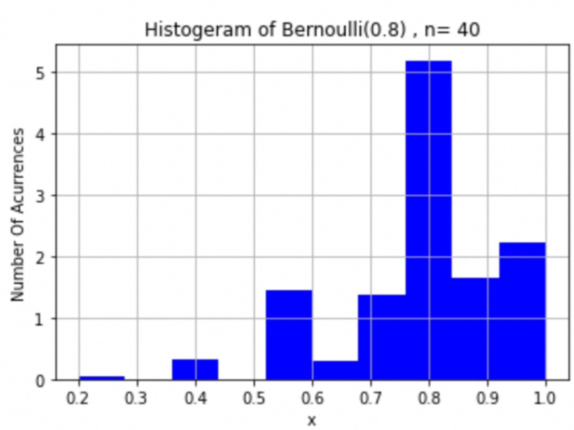
```python
In [5]: import matplotlib.pyplot as plt
        from scipy.stats import bernoulli
        import numpy as np
        list_n=[5,40,300]
        s=0
        y=[]
        for i in range (3):
            n=list_n[i]
            for j in range (1000) :
                s=0
                for j in range (n) :
                    x=bernoulli.rvs(0.8)
                    s=s+x
                y.append(s/n)
            plt.hist(y,color="b",density="True")
            plt.title("Histogeram of Bernoulli(0.8) , n= %d"%n)
            plt.xlabel("x")
            plt.ylabel("Number Of Acurrences")
            plt.grid(True)
            plt.show()
```

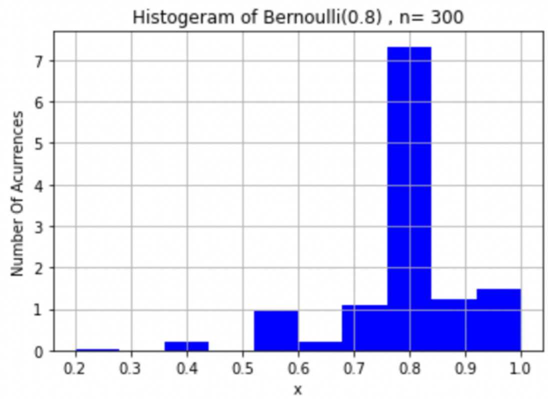
در این سوال همانند سوال ۱ می خواهیم داده تصادفی برنولی تولید کنیم با این تفاوت که در سوال ۱ در هر مرحله ۱۰ تا تولید می کردیم اینجا می خواهیم ۵ ، ۴۰ و ۳۰۰ تا تولید کنیم در اینجا یک لیست حاوی ۵ و ۴۰ و ۳۰۰ داریم که هر بار یکی از آن ها را وارد حلقه کرده و در انتها هیستوگرامش را همانطور که در سوال ۱ انجام دادیم رسم می کنیم.

همانگونه که در نتایج پایین مشاهده می شود با افزایش n امید ریاضی به ۰.۸ نزدیک تر و واریانش کوچک تر می شود.

نتایج :





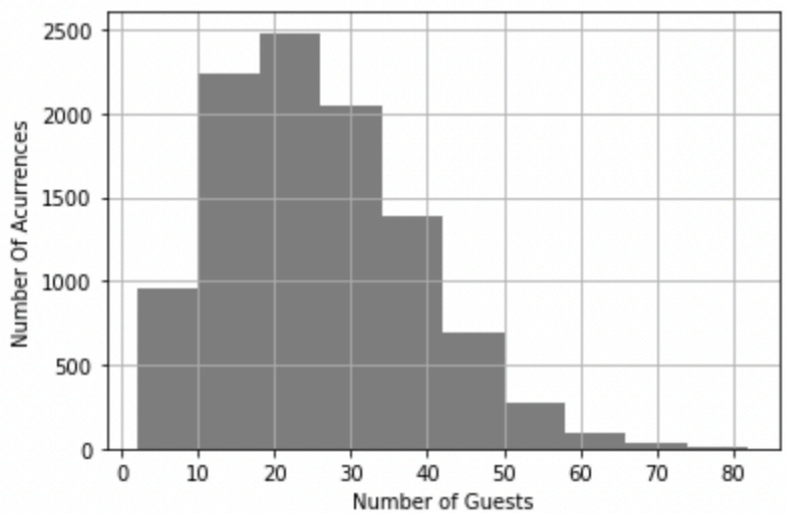


.۴
قسمت اول :
کد :

```
In [1]: import matplotlib.pyplot as plt
from scipy.stats import bernoulli
import numpy as np
p=[]
temperory=[]
counter=0
for i in range (10000) :
    b=0
    counter=0
    while b==0 :
        a=np.random.randint(1,366)
        temperory.append(a)
        for j in range (counter-1) :
            if a == temperory[j] :
                p.append(counter)
                b=1
        counter=counter+1
plt.hist(p,color="gray")
plt.xlabel("Number of Guests")
plt.ylabel("Number Of Acurrences")
plt.grid(True)
plt.show()
```

برای پیاده سازی این قسمت در ابتدا یک منغیر b در نظر گرفتیم و مقدار اولیه آن را ۰ در نظر گرفتیم سپس یک داده تصادفی بین ۱ تا ۳۶۶ (که شامل ۳۶۵ هم می شود) تولید کردیم و یکی یکی به لیست temperory اضافه کردیم با اعضای قبلیش مقایشه کردیم و در صورت وجود تشابه b رو یک می کنیم و از وایل بیرون میاییم و هر بار این فرایند رخ می دهد به متغیر شمارنده یکی اضافه می کنیم و هر وقت از وایل بیرون می آییم شمارنده را به لیست b اضافه می کنیم و در آخر هیستوگرام p را رسم می کنیم.

نتیجه :



2500
2000
1500
1000
500
0
Number Of Acurrences
0
10
20
30
40
50
60
70
80
Number of Guests