دانشکده مهندسی برق

آمار و احتمال مهندسی

فاز اول پروژه

محمد مهدی رزمجو - ۴۰۰۱۰۱۲۷۲

بهار ۱۴۰۲

۱ تناقض روز تولد

مساله تناقض روز تولد نقش مهمی در مسائل رمزنگاری در مخابرات دارد. در این قسمت می خواهیم این مساله را به دقت بررسی کنیم. فرض کنید در یک مهمانی n نفر حضور دارند. رویداد A را تعریف می کنیم با رخداد اینکه حداقل ۲ نفر از این n نفر در یک روز از سال به دنیا آمده باشند. حال مساله اینجا است که مقدار n چقدر باشد که احتمال رخ دادن این پیشامد حداقل ۵۰ درصد باشد. طبیعی است اگر $n > 365$ باشد آنگاه احتمال رخ دادن رویداد A طبق اصل لانه کبوتری برابر با ۱ است. بنابراین در ادامه مساله را برای n های کوچکتر مساوی از ۳۶۵ بررسی می کنیم.

پرسش تئوری ۱

فرض کنید این n نفر را به ترتیب شماره گذاری کرده ایم. چقدر احتمال دارد که نفر اول و نفر دوم در یک روز به دنیا آمده باشند.

پاسخ ۱

صرف نظر از اینکه نفر اول در کدام روز از سال متولد شده باشد ، احتمال اینکه نفر دوم، دقیقا در همان روز متولد شده باشد برابر است با:

$$\frac{1}{365}$$

پرسش تئوری ۲

احتمال رخداد A را به کمک به دست آوردن احتمال رخداد A' به دست آورید.

پاسخ ۲

اگر p احتمال این باشد که حداقل دو نفر در یک روز به دنیا آمده باشند، میتوان گفت که p' احتمال این است که هیچ دو نفری در یک روز به دنیا نیامده اند. با این تفاسیر داریم:

$$p' = \frac{365}{365} \times \frac{364}{365} \times \frac{363}{365} \times ... \times (1 - \frac{n-1}{365}) = \frac{n! \times \binom{365}{n}}{365^n}$$

پرسش شبیه سازی ۱

ابتدا $Seed$ را برابر با شماره دانشجویی خود قرار داده و سپس به تعداد ۱۰۰۰ دفعه، هر بار $Seed$ را برابر با با شماره دانشجویی خود به علاوه شماره آن دفعه کنید. حال در هر تکرار بعد از تنظیم کردن $Seed$ به طریقی که گفته شد، $n = 30$ عدد تصادفی بین ۱ تا ۳۶۵ تولید کنید و تعداد دفعاتی که رویداد A رخ می دهد را گزارش کنید. با استفاده از عدد به دست آمده احتمال رخداد A را به ازای $n = 30$ به دست آورید. آیا اگر تعداد دفعات تکرار را بیشتر کنیم جواب دقیق تر می شود؟

قطعه کد این بخش به صورت زیر می باشد:

```
n = 30
A = 0
for i in range(1, 1001):
  random.seed(400101272+i)
  list = []
  for i in range(1,n+1):
    chosen_number = random.randint(1,365)
    list.append(chosen_number)
  if len(list) != len(set(list)):
    A = A+1

print("Number of times A occurred = ",A)
```

خروجی قطعه کد بالا برابر ۷۱۲ می باشد. در نتیجه احتمال روی دادن رخداد A را می توان ۰.۷۱۲ در نظر گرفت.
از طرفی با افزایش تعداد دفعات تکرار به جواب دقیق تری دست خواهیم یافت. به طور مثال به ازای ۱۵۰۰ دفعه تکرار ، خروجی برابر ۱۰۵۳ (معادل احتمال ۷۰۲.۰) و به ازای ۲۰۰۰ خروجی برابر ۱۴۱۶ (معادل احتمال ۷۰۸.۰) خواهد بود.

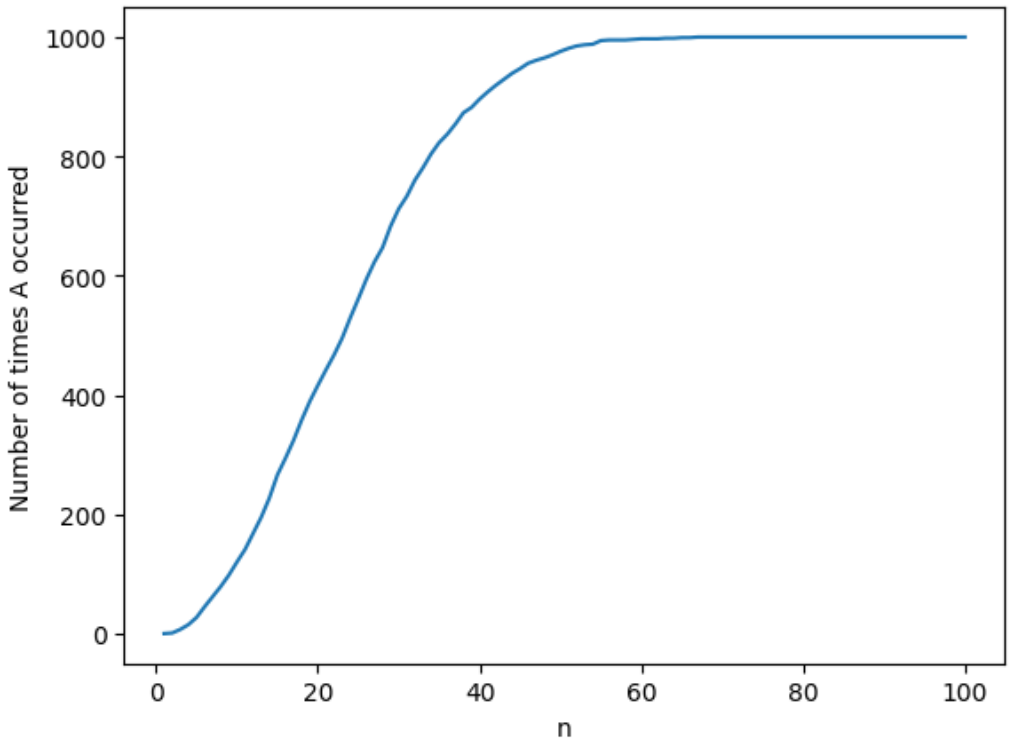
پرسش شبیه سازی ۲

آزمایش قسمت قبل را به ازای n های ۱ تا ۱۰۰ تکرار کنید و نمودار تعداد دفعات رخ دادن رویداد A را به ازای n رسم کنید.

قطعه کد این بخش به صورت زیر است:

```
A = 0
As = []
for j in range(1,101):
    for i in range(1, 1001):
        random.seed(400101272+i)
        list = []
        for i in range(1,j+1):
            chosen_number = random.randint(1,365)
            list.append(chosen_number)
        if len(list) != len(set(list)):
            A = A+1
    As.append(A)
    A=0

n = np.arange(1,101,1)
plt.plot(n,As)
plt.xlabel('n')
plt.ylabel('Number of times A occurred')
plt.show()
```

خروجی این قسمت به صورت زیر است:



پرسش شبیه سازی ۳

تابعی بنویسید که با ورودی گرفتن مقدار n احتمال رخداد A را خروجی دهد(با توجه به پرسش تئوری دوم). سپس با استفاده از این تابع احتمال رخداد A را به ازای ورودی های $n = 10, 30, 60$ به دست آورید و در گزارش بیاورید.

تابع نوشته شده بر اساس بخش تئوری دوم به صورت زیر است:

```
def probabilityOfA(n):
  A = 1-(math.factorial(n)*math.comb(365,n))/(365**n)
  print(round(A,3))

```

خروجی به ازای n های داده شده به صورت زیر است:

n	$p(A)$
۱۰	۰.۱۱۷
۳۰	۰.۷۰۶
۶۰	۰.۹۹۴

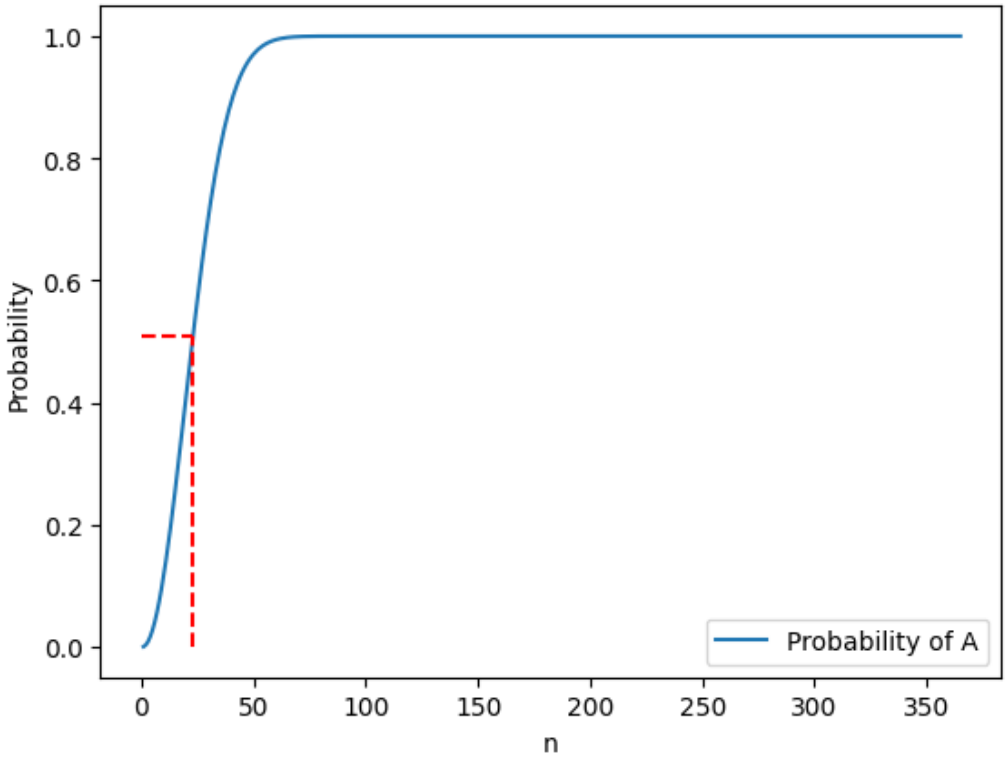
پرسش شبیه سازی ٤

نمودار احتمال رخداد A را بر حسب n رسم کنید. و اولین مقدار n که احتمال این رخداد بزرگتر مساوی ۰.۵ می شود را در گزارش بیاورید. نمودار به دست آمده را با نمودار پرسش شبیه سازی ۲ مقایسه کنید این دو نمودار باید مشابه باشند(چرا؟).

قطعه کد این بخش با توجه به تابع نوشته شده در بخش قبل به صورت زیر می باشد:

```
n = np.arange(1,366,1)
As = []
check = False
for i in range(1,366):
    As.append(probabilityOfA(i))
    if probabilityOfA(i)>=0.5 and check==False:
        the_point_n=i
        the_point_A=probabilityOfA(i)
        check=True

plt.plot(n,As,label='Probability of A')
plt.plot([the_point_n, the_point_n],[0, the_point_A],'r--')
plt.plot([0, the_point_n],[the_point_A, the_point_A],'r--')
plt.legend()
plt.ylabel('Probability')
plt.xlabel('n')
plt.show()
```

خروجی کد بالا به صورت زیر می باشد:



نقطه مشخص شده در نمودار بالا معادل $n = 23$ است.
همانطور که مشخص است، این نمودار شبیه به نمودار نمایش داده شده در بخش شبیه سازی دوم است. (دلیل شباهت نمودار ها این است به ازای افزایش n احتمال و یا معادلا تعداد دفعات رخداد A به ازای مقدار مشخصی تکرار آزمایش، بیشتر می شود)

در ادامه می خواهیم مساله را به طور عمومی تر حل کنیم. فرض کنید تعداد روز های سال به جای ۳۶۵ برابر با d روز باشد. با این تفسیر اگر بخواهیم جواب مساله یعنی حداقل مقدار n که احتمال رخداد A بیشتر از نیم شود را به دست آوریم ممکن است سر و کار با اعداد خیلی بزرگی داشته باشیم (فرض کنید مثالا d برابر با 10^6 باشد) آنگاه شاید با روش های معمول به دست آوردن احتمال رخداد A کمی سخت و زمانبر باشد. به همین دلیل باید به دنبال تقریب خوبی از احتمال این رخداد باشیم که محاسبه آن ساده تر باشد.

پرسش تئوری ۳

با فرض اینکه تعداد روز های سال برابر با d و تعداد افراد در کلاس برابر با n باشند. یک کران بالا با تقریب مناسب برای احتمال این رخداد به دست آورید.

پاسخ ۳

اگر تعداد روز های سال را برابر با d در نظر بگیریم خواهیم داشت:

$$p' = 1 \times (1 - \frac{1}{d}) \times (1 - \frac{2}{d}) \times ... \times (1 - \frac{n-1}{d})$$

از طرفی با توجه به نامساوی $e^{-x} > 1 - x$ می توان گفت که:

$$p' > 1 \times e^{-\frac{1}{d}} \times e^{-\frac{2}{d}} \times ... \times e^{-\frac{n-1}{d}}$$

در نتیجه داریم:

$$1 - e^{-\frac{n(n-1)}{2d}} > p$$

پرسش تئوری ٤

با استفاده از تقریبی که به دست آوردید با فرض $d = 365$ مقدار n مطلوب که در پرسش شبیه سازی ۴ به دست آوردید را راستی آزمایی کنید آیا به ازای این مقدار احتمال رخداد A هنوز بیشتر از نیم هست؟ اختلاف رخداد احتمال A را در تقریب و عبارت اصلی به ازای این مقدار از n بررسی کنید.

پاسخ ۴

کران بالای به دست آمده در قسمت قبل به ازای $d = 365$ و $n = 23$ برابر است با:

$$1 - e^{-\frac{23(23-1)}{730}} \backsimeq 0.500$$

مقدار حاصل هنوز بزرگ تر از ۵.۰ می باشد.مقدار احتمال با استفاده از عبارت اصلی برابر است با:

$$1 - \frac{23! \times \binom{365}{23}}{365^{23}} \backsimeq 0.507$$

در نتیجه اختلاف احتمال رخداد A با استفاده از تقریب و با استفاده از فرمول اصلی برابر $0.507 - 0.500 = 0.007$ می باشد.

برای اینکه مطمئن شویم تقریبی که در قسمت قبل به دست آوردیم درست است یک شبیه سازی کوچک انجام می دهیم.

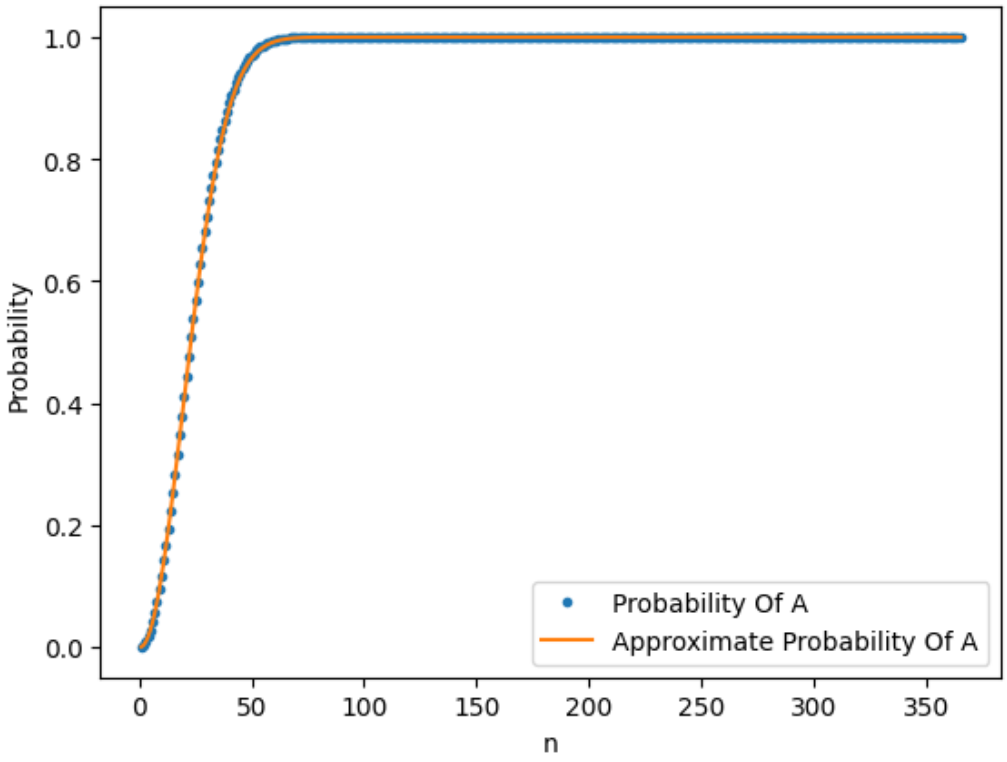
پرسش شبیه سازی ۵

نمودار احتمال رخداد A بر حسب n را به ازای $d = 365$ با استفاده از هر دو رابطه (رابطه اصلی و رابطه کران بالا) در یک نمودار رسم کنید.(رابطه کران بالا به صورت پیوسته و رابطه اصلی به صورت نقطه ای رسم شده باشد.)

قطعه کد این بخش به صورت زیر می باشد:

```
def probabilityOfA(n):
    A = 1-(math.factorial(n)*math.comb(365,n))/(365**n)
    return A

def approximateProbabilityOfA(n):
    A = math.exp(-(math.factorial(n)*math.comb(365,n))/(365**n))
    return A

n = np.arange(1,366,1)
As = []
AAs = []
check = False
for i in range(1,366):
    As.append(probabilityOfA(i))
    AAs.append(approximateProbabilityOfA(i))

plt.plot(n,As,'.',label='Probability Of A')
plt.plot(n,AAs,label='Approximate Probability Of A')
plt.legend()
plt.ylabel('Probability')
plt.xlabel('n')
plt.show()
```

که خروجی آن به صورت زیر می باشد:



فرض کنید برای سورت کردن n شی، به هر یک از آن ها یک $hash$ نسبت داده ایم اگر این $hash$ تنها متشکل از ارقام ۰ تا ۹ باشد و ۱۸ رقم داشته باشد.

پرسش شبیه سازی ٦

حداقل چند شی باید داشته باشیم تا احتمال یکی شدن کد $hash$ حداقل دو تا از شی ها بیشتر یا مساوی ۵.۰ شود؟

با توجه به کران بالایی که به دست آوردیم داریم:

```
def approximateProbabilityOfA(n,d):
    A = 1-math.exp(-((n*(n-1))/(2*d)))
    return A

check = True
d = 10**18
max = d
min = 1
while(check):
    mid = (max+min)//2
    if approximateProbabilityOfA(mid,d)>0.5:
        if(approximateProbabilityOfA(mid-1,d)<0.5):
            x = mid
            check = False
        else:
            max = mid-1
    else:
        min = mid+1

print(x)
```

خروجی قطعه کد بالا برابر با ۱۱۷۷۴۱۰۰۲۴ می باشد.

طبق آخرین مقالات که به دست آمده است، برای تمام $d < 10^{18}$ ، حداقل مقدار n که به ازای آن احتمال یکسان شدن دو عدد تصادفی با توزیع یونیفرم در فضای نمونه حداقل ۵۰ درصد شود از رابطه زیر به دست می آید:

$$n(d) = \left\lceil \sqrt{2dln2} + \frac{3-2ln2}{6} + \frac{9-4(ln2)^2}{72\sqrt{2dln2}} - \frac{2(ln2)^2}{135d} \right\rceil$$

پرسش تئوری ۵

ا استفاده از رابطه داده شده مقداری که در آزمایش قبلی به دست آوردید را راستی آزمایی کنید.

پاسخ ۵

به ازای $d = 10^{18}$ داریم:

$$n(10^{18}) = \left\lceil \sqrt{210^{18} ln2} + \frac{3-2ln2}{6} + \frac{9-4(ln2)^2}{72\sqrt{210^{18} ln2}} - \frac{2(ln2)^2}{13510^{18}} \right\rceil = 1177410023$$

که با عدد به دست آمده در شبیه سازی تنها یک واحد اختلاف دارد.

۲ تلفنچی نمایی

فرض کنید شما در مرکز مخابرات نشسته اید و زمان هایی که یک تماس تلفنی برقرار می شود را نگاه می کنید. با بررسی دقیق تر شما متوجه می شوید که فاصله زمانی بین دو تماس تلفنی متوالی از توزیع نمایی با پارامتر λ پیروی می کند و این فاصله ها بین تماس های مختلف مستقل است. حال برای شما سوال می شود که توزیع حاکم بر تعداد تماس ها در واحد زمان (۱ ثانیه) چیست؟! به همین علت به بازه های زمان به طول ۱ ثانیه نگاه می کنید. یعنی بازه های [۰، ۱)، [۱، ۲)و ... نگاه کرده و با توجه به فاصله زمانی بین تماس های متوالی تعداد تماس ها را در هر بازه به دست می آورید.

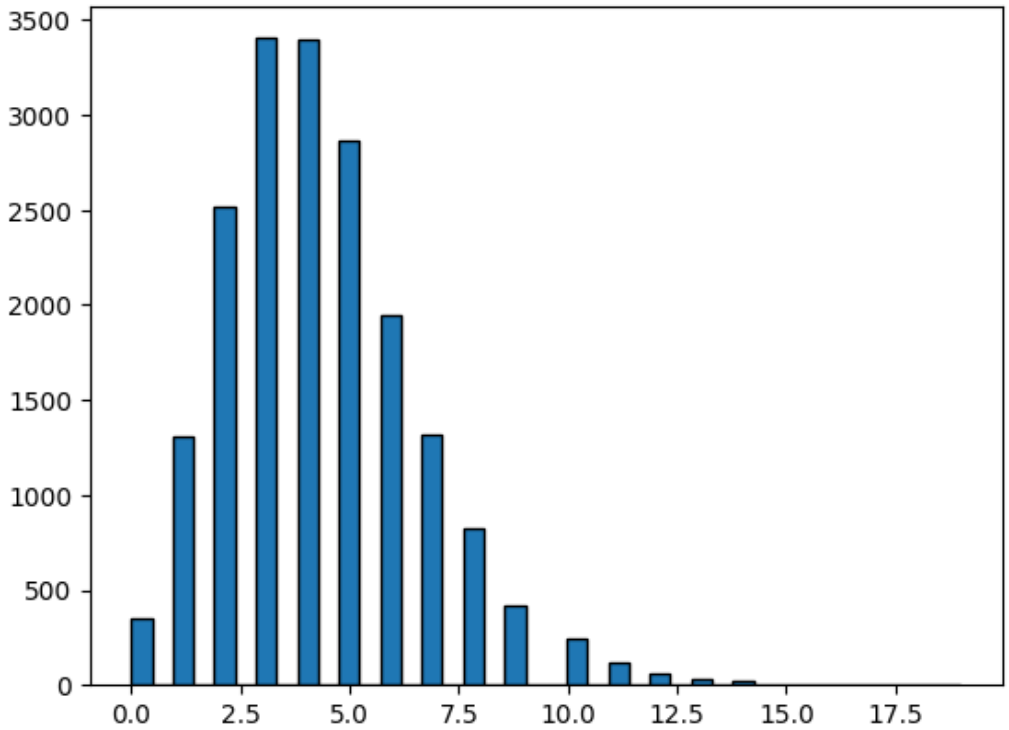
پرسش شبیه سازی ۷

ا توجه به توضیحات داده شده برای به دست آوردن توزیع حاکم بر تعداد تماس ها در یک بازه ۱ ثانیه ، ۱۰۰۰۰۰ نمونه از توزیع نمایی بگیرید و سپس با حاصل جمع آن ها زمان هر تماس را به دست آورید. در نهایت تعداد تماس ها در بازه های گفته شده(یعنی بازه های [۰، ۱)، [۱، ۲)و ...)را تا زمانی که تماسی وجود دارد شمرده و سپس تعداد بازه ها را بر حسب تعداد تماس ها بشمرید. یعنی به ازای هر تعداد بازه هایی که تعداد تماس ها در آن k بار بوده رو به دست می آورید. در نهایت مقادیر به دست آمده را به روی نمودار ببرید.

قطه کد این بخش به صورت زیر می باشد:

```
random_data = expon.rvs(scale=0.3, size=100000)
rounded_data = [ round(elem, 4) for elem in random_data ]

N=[]
x=0
n=0
for i in range(len(rounded_data)):
    x+=rounded_data[i]
    if x<1:
        n+=1
    else:
        N.append(n)
        n=0
        x=1-x

plt.hist(N,density=False, bins=40,edgecolor='black')
plt.show()
```

خروجی به صورت زیر خواهد بود:

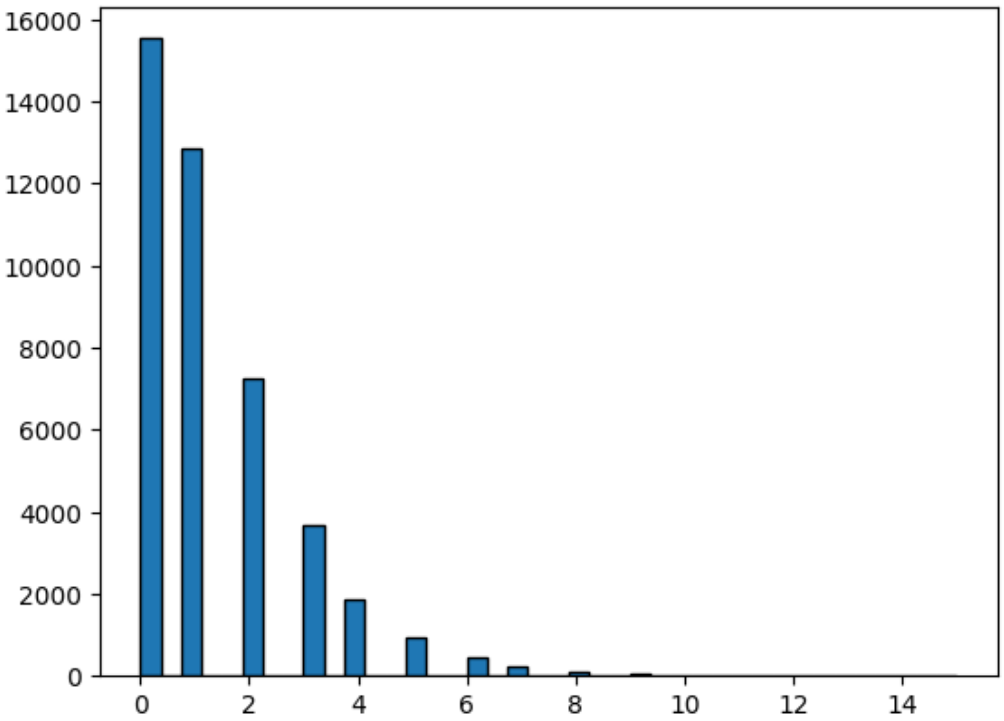


پرسش شبیه سازی ۸

شبیه سازی را برای $\lambda = 3$ تکرار کنید و نتایج را در گزارش بیاورید.

```
random_data = expon.rvs(scale=3, size=100000)
rounded_data = [ round(elem, 4) for elem in random_data ]

N=[]
x=0
n=0
for i in range(len(rounded_data)):
    x+=rounded_data[i]
    if x<1:
        n+=1
    else:
        N.append(n)
        n=0
        x=1-x

plt.hist(N,density=False, bins=40,edgecolor='black')
plt.show()
```

به ازای $\lambda = 3$ خروجی به صورت زیر خواهد بود:



پرسش تئوری ٦

توجه به نتایج در قسمت شبیه سازی حدس می زنید تعداد تماس ها چه توزیعی دارد؟

پاسخ ۶

تعداد تماس ها بایستی دارای توزیع نمایی باشد.

پرسش تئوری ۷

رابطه موجود بین توزیع نمایی و توزیعی که در سوال قبلی نام بردید را اثبات کنید.

پاسخ ۷

به طور کلی، توزیع پواسون با تعداد رخدادها در یک بازه زمانی ثابت سر و کار دارد، و توزیع نمایی با زمان بین وقوع رویدادهای متوالی .
برای نشان دادن این موضوع، اگر فرض کنیم تعداد رخدادها در بازه زمانی t برابر N_t باشد(با فرض رخ دادن آخرین رخداد در زمان t) و همچنین مدت زمانی که طول می کشد تا رخداد بعدی اتفاق بیافتد برابر X باشد، می توان گفت که احتمال اینکه مدت زمانی که طول می کشد تا رخداد بعدی روی دهد بیشتر از x باشد معادل این است که تعداد رخداد های بازه زمانی t با بازه زمانی $t+x$ یکی باشد:

$$P(X > x) \equiv P(N_t = N_{t+x})$$

$$\Rightarrow 1-P(X>x)\equiv 1-P(N_t=N_{t+x})\equiv 1-P(N_t-N_{t+x}=0)$$

با توجه به اینکه $1-P(N_t-N_{t+x}=0)=1-P(N_x=0)$ اگر تعداد رخداد ها از توزیع پواسون پیروی کند، داریم:

$$P(N_x=0)=\frac{(\lambda x)^0}{0!}e^{-\lambda x}=e^{-\lambda x}$$

در نتیجه داریم:

$$1-P(X>x)=P(X\leq x)=1-e^{-\lambda x}$$

به هر متغیر تصادفی که دارای CDFای مانند بالا باشد گفته می شود که به صورت نمایی توزیع شده است.

فرض کنید به جای بازه هایی به طول ۱ بازه هایی به طول T داریم. با این فرض می خواهیم مجددا شبیه سازی را تکرار کنیم.

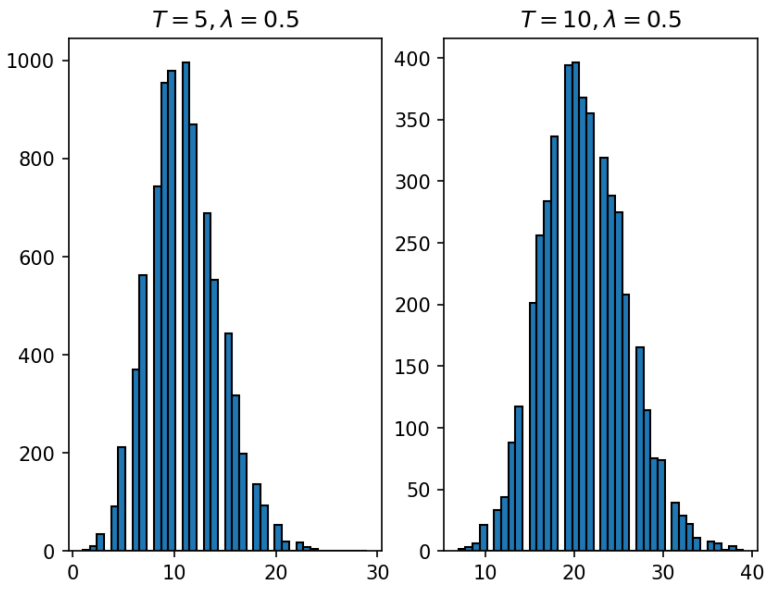
پرسش شبیه سازی ۹

مجددا شبیه سازی را با فرض $T=5$ و $T=10$ و مقدار دلخواه λ انجام دهید و توضیح دهید که پارامتر های توزیع تعداد تماس ها در یک بازه T ثانیه چگونه به دست آید.

قطعه کد این بخش به صورت زیر می باشد:

```
def numberOfCalls(T,S):
    random_data = expon.rvs(scale=S, size=100000)
    rounded_data = [ round(elem, 4) for elem in random_data ]

    N=[]
    x=0
    n=0
    for i in range(len(rounded_data)):
        x+=rounded_data[i]
        if x<T:
            n+=1
        else:
            N.append(n)
            n=0
            x=T-x
    return N

fig, (ax1, ax2) = plt.subplots(1, 2)
ax1.hist(numberOfCalls(5,0.5),density=False, bins=40,edgecolor='black')
ax1.set_title("$T=5 , \lambda = 0.5$")
ax2.hist(numberOfCalls(10,0.5),density=False, bins=40,edgecolor='black')
ax2.set_title("$T=10 , \lambda = 0.5$")
fig.set_dpi(150)
plt.show()
```

خروجی این بخش به صورت زیر است:



اگر داشته باشیم $X \sim Exponential(\lambda)$ به این معناست که به طور متوسط برای روی دادن رخداد بعدی بایستی که $\frac{1}{\lambda}$ واحد زمانی متنظر ماند. درنتیجه می توان گفت که در یک واحد زمانی، به طور متوسط $\frac{1}{\frac{1}{\lambda}} = \lambda$ رخداد روی می دهد. در نیتجه تعداد روی داد ها در یک واحد زمانی به صورت $y \sim Poisson(\lambda)$ توزیع شده است. در نتیجه می توان گفته که در بازه زمانی T، فاصله بین رخداد ها به طور متوسط برابر $\frac{1}{T\lambda}$ می باشد و از طرفی به طور میانگین، $T\lambda$ رخداد در این بازه زمانی روی می دهد.

۳ کدام توزیع را می پسندی!

دو توزیع پیوسته زیر را در نظر بگیرید:

$1 - Gaussian\ distribution \quad \mathcal{N}(\mu = 0, \sigma^2 = 1)$

$2 - Exponential\ distribution$

پرسش شبیه سازی ۱۰

برای هر کدام از توزیع های بالا تابع PDF و CDF را رسم کنید. (برای توزیع گاوسی بازه ی محور x را $[-۱۰، ۱۰]$ و برای توزیع نمایی باز ه ی $[۰، ۲۰]$بگیرید.)

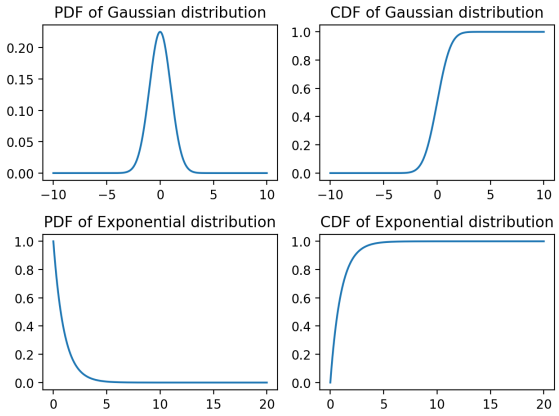
با توجه به

	PDF	CDF
$Gaussian\ distribution$	$\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$	$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{u^2}{2}}\, du$
$Exponential\ distribution$	$\lambda e^{-\lambda x} u(x)$	$(1 - e^{-\lambda x}) u(x)$

قطعه کد این بخش به صورت زیر می باشد:

```
fig,axs = plt.subplots(2, 2)

x1 = np.arange(-10,10,0.01)
y1 = []
for i in x1:
    y1.append((1/(math.sqrt(2)*math.pi))*(math.exp(-(i**2)/2)))

x2 = np.arange(-10,10,0.01)
y2 = []
for i in x2:
    y2.append(scipy.stats.norm.cdf(i))

x3 = np.arange(0,20,0.01)
y3 = []
for i in x3:
    y3.append(math.exp(-i))

x4 = np.arange(0,20,0.01)
y4 = []
for i in x4:
    y4.append(1-math.exp(-i))


axs[0,0].plot(x1,y1)
axs[0,0].set_title('PDF of Gaussian distribution')
axs[0,1].plot(x2,y2)
axs[0,1].set_title('CDF of Gaussian distribution')
axs[1,0].plot(x3,y3)
axs[1,0].set_title('PDF of Exponential distribution')
axs[1,1].plot(x4,y4)
axs[1,1].set_title('CDF of Exponential distribution')
fig.set_dpi(200)
fig.tight_layout()
```

خروجی این قطعه کد به صورت زیر می باشد:



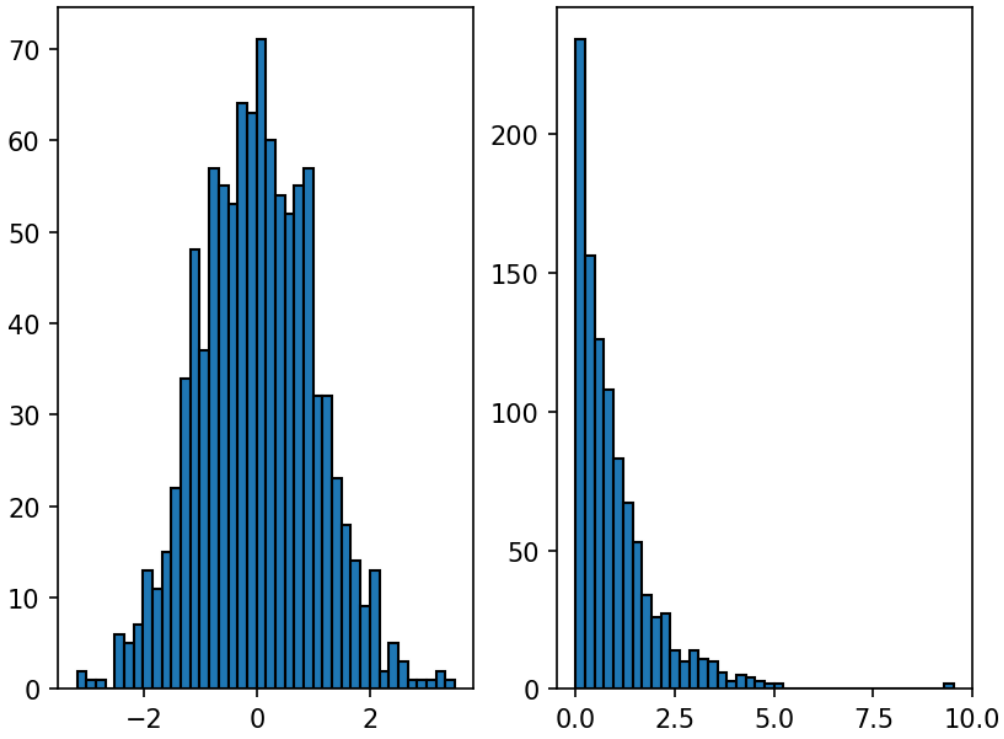
پرسش شبیه سازی ۱۱

ه ازای هرکدام از توزیع های بالا، ۱۰۰۰ داده ی تصادفی تولید کرده و هیستوگرام آن ها را در یک پنجره رسم کنید.

قطعه کد این بخش به صورت زیر می باشد:

```
fig, (ax1, ax2) = plt.subplots(1, 2)

random_data1 = np.random.normal(loc=0, scale=1, size=1000)
ax1.hist(random_data1,density=False, bins=40,edgecolor='black')

random_data2 = expon.rvs(scale=1, size=1000)
ax2.hist(random_data2,density=False, bins=40,edgecolor='black')

fig.set_dpi(150)
plt.show()
```

خروجی (نمونه) به صورت زیر خواهد بود:



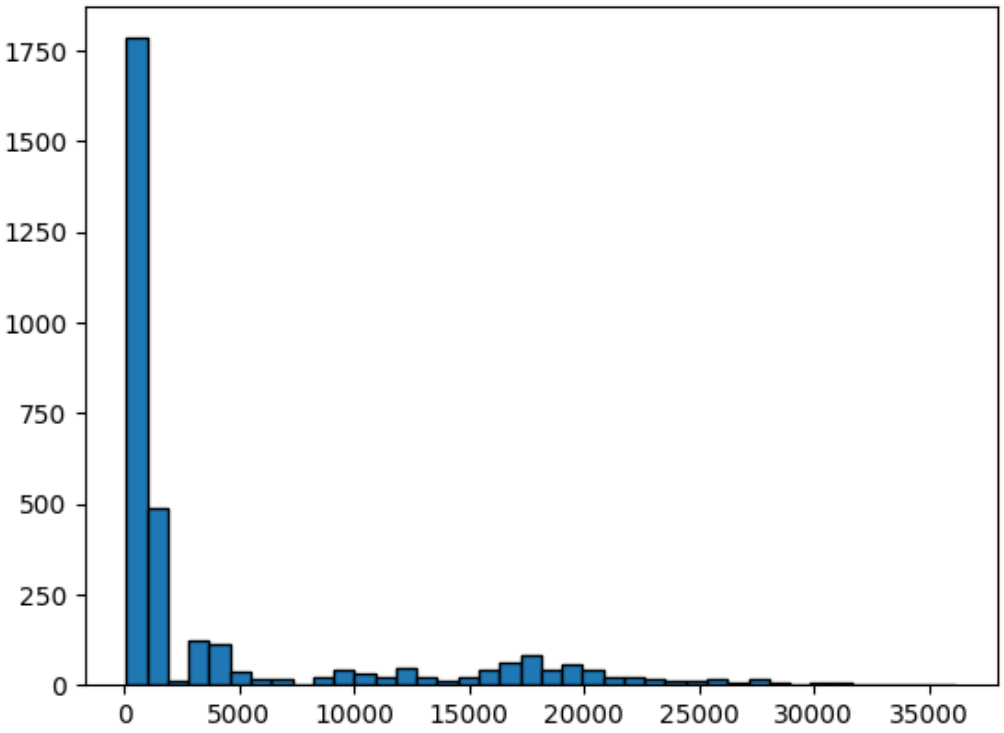
پرسش شبیه سازی ۱۲

فایل "$data.xls$" حاوی اطلاعات بورسی از یک شرکت خاص میباشد. یکی از ستون های آن به نام "قیمت پایانی" میباشد. داده های این ستون را بررسی کرده و با ذکر دلیل بگویید که این داده ها از کدام یک از توزیع های بیان شده میتوانند تولید شوند؟ (برای بررسی شهودی بهتر است هیستوگرام داده ها را رسم کرده و نمودار آن را با نمودار های قسمت قبلی مقایسه کنید!)

داده های مربوط به ستون "قیمت پایانی" را به صورت زیر رسم میکنیم:

```
df = pd.read_excel('data.xls')

values = df.iloc[:3288,5]
plt.hist(values.values,density=False, bins=40,edgecolor='black')
plt.show()
```

با توجه به خروجی زیر می توان گفت که این داده ها میتوانند با استفاده از توزیع نمایی تولید شده باشند و توزیع نمایی بهتر از توزیع نمرال روی آن فیت می شود:



بعد از انتخاب توزیع مناسب، به دنبال یافتن تخمینی از پارامتر توزیع موردنظر هستیم. یکی از تخمین های معروف که در این قسمت استفاده خواهیم کرد تخمین زننده ی $Maximum\ Likelihood\ Estimator$ است. در آینده با این تخمین زننده بیشتر آشنا خواهید شد. دیتاهای ستون "قیمت پایانی" را در نظر بگیرید. اعدادی که کوچکتر از ۲۵۰۰ هستند را در یک $vector$ به نام X ذخیره کرده و عضو i ام آن را x_i مینامیم. بنابراین بردار X به صورت زیر در می آید:

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}$$

چیزی که ما فعلا در اینجا از $Maximum\ Likelihood\ Estimator$ نیاز داریم روابط زیر هستند:

$$\hat{\mu}_{MLE} = \bar{x} \quad , \quad \hat{\sigma}^2_{MLE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \quad , \quad \hat{\lambda}_{MLE} = \frac{1}{\bar{x}}$$

پرسش شبیه سازی ۱۳

با تعریف پارامتر های جدید بالا، از دو توزیع معرفی شده تعداد n داده ی تصادفی تولید کنید و نمودار هیستوگرام آن ها را رسم کرده و با نمودار بخش آزمایش قبلی مقایسه کنید .(n طول بردار X است که برابر تعداد داده های کوچکتر از ۲۵۰۰ است!) دلیل تفاوت زیاد بین نمودار هیستوگرام دیتاها در آزمایش قبلی و نمودارهای این بخش را توجیه کنید. (به این توجه کنید که برای مدلسازی دیتاها گزینه های دیگری غیر از توزیع نرمال و نمایی میتوان استفاده کرد!)

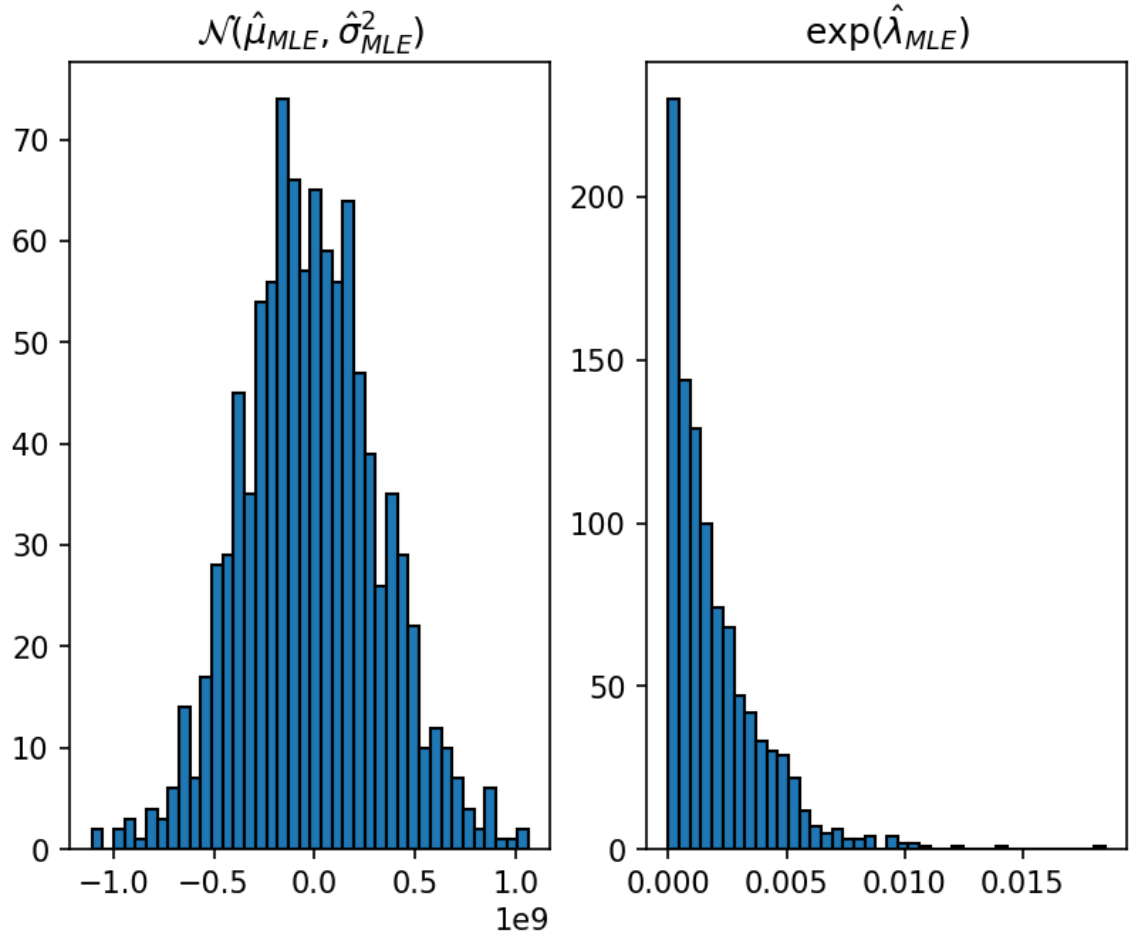
قطعه کد این بخش به صورت زیر می باشد:

```
df = pd.read_excel('data.xls')

values = df.iloc[:3288,5]
X=[]
for i in range(3288):
    if values[i]<2500:
        X.append(values[i])

Xbar = round(sum(X),2)/len(X)
muhat = Xbar
lambdahat = 1/Xbar

sigma=0
for i in range(len(X)):
    sigma+=(X[i]-Xbar)**2
sigmahat = sigma/n

```

```
fig, (ax1, ax2) = plt.subplots(1, 2)

        random_data1 = np.random.normal(loc=muhat, scale=sigmahat, size=1000)
        ax1.hist(random_data1,density=False, bins=40,edgecolor='black')
        ax1.set_title("$\mathcal{N} (\hat{\mu}_{MLE},\hat{\sigma}_{MLE}^2)$")

        random_data2 = expon.rvs(scale=lambdahat, size=1000)
        ax2.hist(random_data2,density=False, bins=40,edgecolor='black')
        ax2.set_title("$\exp (\hat{\lambda}_{MLE})$")

        fig.set_dpi(150)
        plt.show()
```

به ازای داده های کوچک تر از ۲۵۰۰ ، مقادیر پارامتر ها به صورت زیر است:

$$\hat{\mu}_{MLE} \approx 493.87 \quad , \quad \hat{\sigma}^2_{MLE} \approx 327666685.92 \quad , \quad \hat{\lambda}_{MLE} \approx 0.0020248$$



همانطور که مشخص است، تفاوت زیادی در نمودار ها وجود دارد که ناشی ازاین است که داده ها، برخلاف حدس ما از توزیع نمایی پیروی نمیکنند و ممکن است توزیع دیگری غیر از نمایی و نرمال داشته باشند.