

به نام خدا



درس : آمار و احتمال مهندسی

نام : محسن کمال آبادی فراهانی

شماره دانشجویی : ۹۹۱۰۲۰۸۳

گزارش تمرین کامپیوتری چهارم

۱.
کد:

```
In [15]: from scipy.stats import cauchy
from scipy.stats import pareto
def CauchySampling(x0, lmbda, size):
    sample = [0]*size
    sample = cauchy.rvs(x0, lmbda, size)
    return sum(sample)/size
cs=[0]*1000
cs=[CauchySampling(100,10,100) for _ in range(1000)]
var=0
meanCs=sum(cs)/1000
for i in range(1000):
    var=var+(cs[i]-meanCs)**2
print(var/1000)
```

248700.6582016738

```
In [16]: def ParetoSampling(xm, alpha, size):
    sample=[0]*size
    sample=pareto.rvs(xm, alpha, size=size)
    mean=sum(sample)/size
    return mean
cs=[0]*1000
cs=[ParetoSampling(1,0.5,100) for _ in range(1000)]
var=0
meanCs=sum(cs)/1000
for i in range(1000):
    var=var+(cs[i]-meanCs)**2
print(var/1000)
```

8570.215399089888

با استفاده از کتابخانه های پیشنهادی و دستور های لازمه نمونه ها را تولید می کنیم و میانگینشان را از تابع میگیریم سپس با ۱۰۰۰ بار نمونه گیری از هر کدام واریانسشان را حساب می کنیم.

۲.
کد:

```
In [7]: import numpy as np
def SampleBinomial(p,n,size):
    samples=[0]*size
    samples=np.random.binomial(n,p,size)
    return samples
X=SampleBinomial(0.5,20,10000)
meanX= np.mean(X)
VarX= np.var(X)
print(meanX)
print(VarX)
```

```
10.0294
4.967935639999999
```

```
In [8]: def FindProb(samples,l,u):
X=(sample<=u)*(sample>=l)
return(sum(X)/sample.size)
sample=np.arange(1000)
print(FindProb(samples,0,299))
```

```
0.3
```

```
In [9]: import math
from scipy.stats import norm
def EstProb(p,n,l,u):
    mean = p
    var = p*(1-p)
    x = (l-n*p)/(math.sqrt(n*var))
    y = (u-n*p)/(math.sqrt(n*var))
    return norm.cdf(y)-norm.cdf(x)
print(EstProb(0.5,20,8,10))
```

```
0.3144533152386512
```

```
In [10]: def CorEstProb(p,n,l,u):
return (EstProb(p,n,l-0.5,u+0.5))
print(CorEstProb(0.5,20,8,10))
```

```
0.45669212447945295
```

$$\text{تجربی: } P(8 \leq Y \leq 10) = \sum_{k=8}^{10} \binom{n}{k} p^k (1-p)^{n-k}$$

$$= \left[\binom{20}{8} + \binom{20}{9} + \binom{20}{10} \right] \left(\frac{1}{2} \right)^{20} = 0.456$$

همانگونه که انتظار می رفت نتیجه بخش دوم به نتیجه تجربی نزدیک تر بود. زیرا ما در بخش یک توزیع گسسته را با یک توزیع پیوسته تخمین می زدیم که برای جبران این تخمین l و u را به علاوه و منهای 0.5 کردیم.

۳.
کد:

```
In [4]: import pandas as pd
df = pd.read_csv("heart.csv")
print(df.head())
```

```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0    63   1   3     145    233   1         0     150      0      2.3      0
1    37   1   2     130    250   0         1     187      0      3.5      0
2    41   0   1     130    204   0         0     172      0      1.4      2
3    56   1   1     120    236   0         1     178      0      0.8      2
4    57   0   0     120    354   0         1     163      1      0.6      2

   ca  thal  target
0   0     1        1
1   0     2        1
2   0     2        1
3   0     2        1
4   0     2        1
```

```
In [5]: print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
---  -
0   age         303 non-null    int64
1   sex         303 non-null    int64
2   cp          303 non-null    int64
3   trestbps    303 non-null    int64
4   chol        303 non-null    int64
5   fbs         303 non-null    int64
6   restecg     303 non-null    int64
7   thalach     303 non-null    int64
8   exang       303 non-null    int64
9   oldpeak     303 non-null    float64
10  slope       303 non-null    int64
11  ca          303 non-null    int64
12  thal        303 non-null    int64
13  target      303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
None
```

```
In [6]: df_chol=df[['chol']]
print(df_chol.describe())
```

```
count    303.000000
mean     246.264026
std       51.830751
min      126.000000
25%      211.000000
50%      240.000000
75%      274.500000
max       564.000000
Name: chol, dtype: float64
```

```
In [8]: import matplotlib.pyplot as plt
myplot = plt.hist(df_chol, bins=100)
plt.axvline(df_chol.mean(), color='g')
plt.show()
```

