

In the name of God



استاد : دکتر تیموری

دانشجو : توحید حقیقی سیس

شماره دانشجویی : 830598021

موضوع : تمرین هشتم

1. For any set of numbers x_1, \dots, x_n , prove algebraically that

$$\sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n x_i^2 - n\bar{x}^2$$

where $\bar{x} = \sum_{i=1}^n x_i / n$.

①

$$\sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n x_i^2 - n\bar{x}^2$$

where $\bar{x} = \sum_{i=1}^n x_i / n \rightarrow \sum_{i=1}^n x_i = n\bar{x}$

$$\rightarrow \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n (x_i^2 - 2x_i\bar{x} + \bar{x}^2) =$$

$$= n\bar{x}^2 + \sum_{i=1}^n x_i^2 - 2\bar{x} \sum_{i=1}^n x_i =$$

$$\rightarrow \sum_{i=1}^n x_i^2 - n\bar{x}^2$$

اثبات شد.

تمرین دوم :

3. Write a program that uses the recursions given by Equations (8.6) and (8.7) to calculate the sample mean and sample variance of a data set.

در این سوال مقادیر واریانس و میانگین بصورت بازگشتی خواسته شده که با استفاده از روابط داده شده در همین فصل بصورت زیر مقادیر را بصورت بازگشتی محاسبه می نماییم ، با این تفاوت که شرط پایه میانگین بر گرداندن تنها عنصر آرایه (لیست یا دیتاست) است اما در واریانس با رسیدن به دو نمونه یعنی آرایه دو عنصری به محاسبه واریانس پرداخته و بازگشت میزنیم ، پاسخ ها بصورت عددی با نمونه داده شده تست شده و یکسان بودند.

```
a = [2,6,7,9,12,16]
# یافتن میانگین
def average(a):
    if len(a) == 1:
        return a[0]
    else:
        n = len(a)
        # یافتن میانگین به صورت بازگشتی
        return average(a[0:n-1]) + ((a[n-1]-average(a[0:n-1]))/n)

print('average using recursion for dataset a:' , a , 'is :' , average(a))

# یافتن واریانس
def variance(a):
    if len(a) == 2:
        return ( (a[0]-average(a))**2 + (a[1]-average(a))**2 )
    else:
        n = len(a)
        # یافتن واریانس به صورت بازگشتی
        return (1-(1/(n-1)))*variance(a[0:n-1]) + n*((average(a)-average(a[0:n-1]))**2)

# چاپ مقدار های ورودی و محاسبه واریانس
print('variance(s^2) using recursion for dataset a:' , a , 'is :' , variance(a))
```

خروجی به صورت زیر است :

```
average using recursion for dataset a: [2, 6, 7, 9, 12, 16] is : 8.666666666666668
variance(s^2) using recursion for dataset a: [2, 6, 7, 9, 12, 16] is : 23.866666666666669
```

4. Continue to generate standard normal random variables until you have generated n of them, where $n \geq 100$ is such that $S/\sqrt{n} < 0.1$, where S is the sample standard deviation of the n data values.

- (a) How many normals do you think will be generated?
- (b) How many normals did you generate?
- (c) What is the sample mean of all the normals generated?
- (d) What is the sample variance?
- (e) Comment on the results of (c) and (d). Were they surprising?

در این سوال میبایست مقادیر تصادفی نرمال استاندارد را ابتدا 111 نمونه تولید کرده و سپس تا جایی که شرط $s/\sqrt{n} < 0.1$ شود ادامه میدادیم که این مقدار در قطعه کد با پارامتر d مشخص شده است. همچنین برای پیاده سازی از الگوریتم زیر استفاده شده فقط با این تفاوت که در یکی از شرط ها تا 111 نمونه تولید نشود اصلا وارد شرط termination نخواهیم شد و لذا 111 نمونه اولیه حتما تولید خواهند شد. الگوریتم مورد استفاده:

1. Choose an acceptable value d for the standard deviation of the estimator.
2. Generate at least 100 data values.
3. Continue to generate additional data values, stopping when you have generated k values and $S/\sqrt{k} < d$, where S is the sample standard deviation based on those k values.
4. The estimate of θ is given by $\bar{X} = \sum_{i=1}^k X_i / k$.

قطعه کد به صورت زیر است :

```

import random
import math
from statistics import mean

d = 0.1;theta_prv = 0;s2_prev = 0;theta = 0;s2 = 0;
k = 0;
while (1):
    #CreateNewSample
    k = k + 1;
    X = random.normalvariate(0,1)
    #Updatetheta
    theta = theta_prv + (X - theta_prv) / k;
    if k >= 2:
        s2 = (1 - 1 / (k - 1)) * s2_prev + k * (theta - theta_prv) ** 2;
        s2_prev = s2;
    theta_prv = theta;
    #CheckforTermination
    if math.sqrt(s2 / k) < d and s2!=0 and k >= 100:
        print('theta = ' , theta ,',k = ',k ,',s2 =:', s2);
        break;

```

نمونه خروجی:

- theta = 0.02058882478136268 ,k = 100 ,s2 =: 0.8765805653392336
- theta = -0.023049644169505522 ,k = 100 ,s2 =: 0.8470922585491878
- theta = 0.04985659686303402 ,k = 100 ,s2 =: 0.8747269035057608
- theta = -0.07576569059127186 ,k = 103 ,s2 =: 1.0288235926321931
- theta = 0.27296873510467873 ,k = 112 ,s2 =: 1.1119904973190224

تمرین چهارم :

6. Estimate $\int_0^1 \exp(x^2)dx$ by generating random numbers. Generate at least 100 values and stop when the standard deviation of your estimator is less than 0.01.

طی قضایای مطرح شده در صفحه ۴۱ و ۴۱ کتاب ، simulation با تولید مقادیر رندوم استاندارد نرمال و جایگذاری در تابع g تابعی که می‌خواهیم انتگرال بگیریم و میانگین گیری تخمینی از حواب بدست می آید لذا همین مراحل پیاده و اجرا شد

$$\theta = \int_0^1 g(x) dx$$

To compute the value of θ , note that if U is uniformly distributed over $(0, 1)$, then we can express θ as

$$\theta = E[g(U)]$$

```
import random
import math
from statistics import mean
d = 0.01; theta_prv = 0; s2_prev = 0; theta = 0;
s2 = 0;
k = 0;
res = []
while (1):
    #CreateNewSample
    k = k + 1;
    X = math.exp(random.uniform(0,1)**2)
    res.append(X)
    #Updatetheta
    theta = theta_prv + (X - theta_prv) / k;
    if k >= 2:
        s2 = (1 - 1 / (k - 1)) * s2_prev + k * (theta - theta_prv) ** 2;
        s2_prev = s2;
    theta_prv = theta;
    #CheckforTermination
    if math.sqrt(s2/k) < d and s2!=0 and k >= 100:
        print('Estimated answer for antegral:', mean(res))
        break;
```

مقدار $d=0.01$ و حداقل 111 متغیر در ابتدا تولید کرده و مقدار متغیر نهایی تولید شده برای g قرار

داده ایم.

```
X = math.exp(random.uniform(0,1)**2)
```

خروجی:

Estimated answer for antegral: 1.4709733083594805

که به مقدار واقعی نزدیک است

8. It can be shown that if we add random numbers until their sum exceeds 1, then the expected number added is equal to e . That is, if

$$N = \min \left\{ n: \sum_{i=1}^n U_i > 1 \right\}$$

then $E[N] = e$.

- Use this preceding to estimate e , using 1000 simulation runs.
- Estimate the variance of the estimator in (a) and give a 95 percent confidence interval estimate of e .

در این سوال برای قسمت a تابعی ساخته شده که شرط گفته شده در سوال را در بر دارد و در خروجی N را که تعداد است بر می گرداند و برای این قسمت سوال 1111 نمونه گیری و میانگین گیری شده که نتیجه تخمینی نیز به شرح زیر است

قسمت ۱)

```
import random
import math
from statistics import mean
def f():
    s=0
    c=0
    while (s<=1):
        c = c + 1
        r = random.random()
        s += r
    return c

res = 0
for i in range(1000):
    res += f()
print('Estimated e using 1000 simulation: ',res/1000)
```

خروجی نمونه:

Estimated e using 1000 simulation: 2.69

قسمت ۲)

در قسمت **b** خواسته شده تا مقدار 55 درصد به مقدار e نزدیک باشیم یعنی طبق برداشت من ، 1015 انحراف معیار داشته باشیم لذا مقدار d را برابر 1015 مقدار عدد e قرار داده و اجرا میکنیم ، در ضمن متغیر تصادفی در اینجا همان N است که از تابع قسمت قبلی کمک گرفتیم

```
import random
import math
from statistics import mean
d = 0.1359140; theta_prv = 0; s2_prev = 0; theta = 0; s2 = 0; k = 0;
res = []
def f():
    s=0
    c=0
    while (s<=1):
        c = c + 1
        r = random.random()
        s += r
    return c
while (1):
    # CreateNewSample
    k = k + 1;
    X = f()
    res.append(X)
    # Updatetheta
    theta = theta_prv + (X - theta_prv) / k;
    if k >= 2:
        s2 = (1 - 1 / (k - 1)) * s2_prev + k * (theta - theta_prv) ** 2;
        s2_prev = s2;
        theta_prv = theta;
    # CheckforTermination
    if math.sqrt(s2 / k) < d and s2 != 0 and k >= 100:
        print('Estimated e :', mean(res))
        break;
```

نمونه خروجی:

Estimated e : 2.73