به نام خدا

دانشکده ی برق دانشگاه صنعتی شریف

گزارش پروژه درس آمار و احتمال مهندسی دکتر آشتیانی

زهرا مجتهدين 99102167

تعيين حداقل نمونه:

با توجه به داده های مسئله خواهیم داشت:

$$n \geq \left[(z_{1-rac{lpha}{2}} + z_{eta}) \,
ight]^2 \left[\left(rac{\sigma}{d}
ight)
ight]^2$$
 تعیین حداقل نمونه

$$\alpha = 0.05$$
 , $1 - \frac{\alpha}{2} = 0.975 \implies z_{1 - \frac{\alpha}{2}} = 1.96$
 $\beta = 0.1$, $1 - \beta = 0.90 \implies z_{\beta} = 1.28$

لذا با توجه به مقادیر بدست آمده مقدار $z_{1-rac{lpha}{2}}$ را از داخل جدول بدست می آوریم . (منظور از جدول ، جدول جدول توزیع نرمال استاندارد می باشد.)

| مقدار ∝Z | Z.,1 | Z.,.o | Z.,. 40 | Z.,.1 | Z.,o |
|----------|-------|-------|---------|-------|-------|
| مقادیر Z | 1.777 | 1.540 | 1.98. | 7.774 | 7.078 |

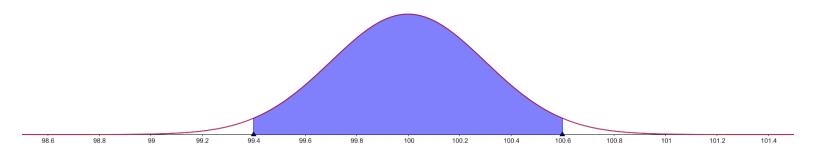
حال با توجه به داده های مسئله دو مقدار برای تعداد نمونه بدست می آوریم:

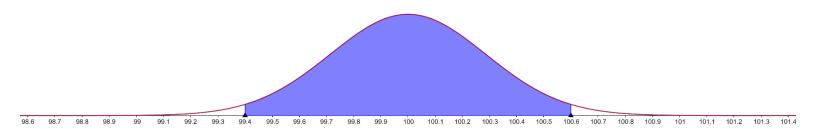
$$n \ge 10.49 * \frac{0.09}{1} = 0.94 \implies n_1 = 1$$
 $n \ge 10.49 * \frac{0.49}{1} = 5.14 \implies n_2 = 6$

حال با توجه به مقادير فوق بازه اطمينان 95٪ را براي اندازه گيري بدست مي آوريم :

با توجه به بازه اطمینان 95٪ و قدرت تست 90٪ (معادل با
$$\beta=0.1$$
) مقدار z_a را محاسبه میکنیم.
$$\eta_0\pm\sigma z_a=100\mp1.578$$

نمودار های توزیع احتمال را به ترتیب انحراف معیار های 0.7 و 0.7 مشاهده می کنیم :





در فایل کد، num همان عددی که بدست آوردیم می باشد. ما اگر این آزمایش را به تعداد دفعات بی شماری تکرار کنیم(یعنی به n تعداد زیادی مقاومت انتخاب کنیم) با احتمال 95 درصد در آن بازه قرار می گیرند. طبق روی سوال هم ما 5000 بار مقاومت با commend برابر n برمیداریم (عملا 30000 تا) که اگر تعداد این ها را افزایش دهیم به 95 می رسد. اعداد تولید شده در window به تعداد زیادی هستند.

100 نمونه مربوط به n=1 و sigma=0.3 نمونه مربوط به

```
Asus ▶ Documents ▶
                                                                                                                                        €
Editor - C:\Users\Asus\Documents\Part One_Project.m
   Part One_Project.m × +
       % for part I project
       sigma = 0.3;
 3
       %Sigma is Standard deviation
 5
       %Num is Calculated number
 6 -
       Counter = 0;
       upper = 100 + 1.96* sigma / sqrt(num);
       lower = 100 - 1.96* sigma / sqrt(num);
 8 -
9 -
     = for i = 1:5000
10 -
           rand=0;
11 - 🖨
            for j = 1:num
12 -
              rand = rand + (randn * sigma + 100)/num;
13 -
14 -
           if(lower < rand && rand < upper)</pre>
15 -
               Counter = Counter + 1;
16 -
17 -
      end
18 -
       disp(Counter * 100 / 5000);
      93.9800
      95.1000
      95.3400
      95.0800
      94.7200
      95.2600
```

100.8 نمونه مربوط به $\sin a=0.3$ و n=1 و مقدار

*تفاوت هر کد با کد قبلی در مقدار دهی مقادیر اسمی می باشد.

```
Editor - C:\Users\Asus\Documents\Untitled3.m
 Part One_Project.m × Untitled3.m × +
       % for part I project
1
2 -
       sigma = 0.3;
       %Sigma is Standard deviation
3
 4 -
      num = 1;
 5
       %Num is Calculated number
      Counter = 0;
7 -
      upper = 100.8 + 1.96* sigma / sqrt(num);
       lower = 100.8 - 1.96* sigma / sqrt(num);
8 -
9 - \Box \text{ for } i = 1:5000
10 -
           rand=0;
11 -
           for j = 1:num
12 -
              rand = rand + (randn * sigma + 100.8)/num;
13 -
14 -
          if(lower < rand && rand < upper)
              Counter = Counter + 1;
16 -
17 -
      end
       disp(Counter * 100 / 5000);
18 -
Command Window
  >> Untitled3
      94.7000
  >> Untitled3
      95.0800
  >> Untitled3
     95.0600
  >> Untitled3
      94.7600
f_{x} >>  Untitled3
```

نمونه مربوط به sigma=0.3 و n=1 و مقدار 101.1

```
Part One_Project.m × Untitled3.m × +
1
       % for part I project
       sigma = 0.3;
2 -
3
       %Sigma is Standard deviation
       num = 1;
4 -
 5
       %Num is Calculated number
 6 -
       Counter = 0;
 7 -
       upper = 101.1 + 1.96* sigma / sqrt(num);
8 -
       lower = 101.1 - 1.96* sigma / sqrt(num);
9 - \Box \text{ for } i = 1:5000
10 -
           rand=0;
         for j = 1:num
11 - 🖨
12 -
             rand = rand + (randn * sigma + 101.1)/num;
13 -
14 -
         if(lower < rand && rand < upper)
15 -
              Counter = Counter + 1;
16 -
17 -
       disp(Counter * 100 / 5000);
18 -
Command Window
  >> Untitled3
     95.5200
  >> Untitled3
     95.1000
  >> Untitled3
     94.5000
  >> Untitled3
     95.1200
f_{x} >>  Untitled3
```

Z Editor - C:\Users\Asus\Documents\Untitled3.m

```
Z Editor - Untitled4*
  Part One_Project.m × Untitled3.m × Untitled4* × +
1
       % for part I project
      sigma = 0.7;
2
3
      %Sigma is Standard deviation
      num = 6;
5
      %Num is Calculated number
      Counter = 0;
 6
      upper = 100 + 1.96* sigma / sqrt(num);
      lower = 100 - 1.96* sigma / sqrt(num);
8
9
    rand=0;
11
         for j = 1:num
12
            rand = rand + (randn * sigma + 100)/num;
13
14
          if(lower < rand && rand < upper)
15
             Counter = Counter + 1;
16
17
      end
18
       disp(Counter * 100 / 5000);
Command Window
     95.1000
     95.1400
     95.1400
     94.6200
     94.8400
     94.4200
     94.4800
```

100.8 و مقدار n=6 و sigma=0.7 نمونه مربوط به

```
Z Editor - C:\Users\Asus\Documents\Untitled4.m
Part One_Project.m × Untitled3.m × Untitled4.m × +
      % for part I project
1
2 -
      sigma = 0.7;
      %Sigma is Standard deviation
3
4 -
      num = 6;
 5
       %Num is Calculated number
 6 -
      Counter = 0;
7 -
      upper = 100.8 + 1.96* sigma / sqrt(num);
8 -
      lower = 100.8 - 1.96* sigma / sqrt(num);
9 - \Box \text{ for } i = 1:5000
10 -
           rand=0;
11 -
          for j = 1:num
            rand = rand + (randn * sigma + 100.8)/num;
13 -
14 -
          if(lower < rand && rand < upper)
15 -
              Counter = Counter + 1;
16 -
         end
17 -
       disp(Counter * 100 / 5000);
Command Window
     95.1600
  >> Untitled4
     95.1200
  >> Untitled4
     95.3800
  >> Untitled4
     94.6400
  >> Untitled4
   94 9200
```

```
Z Editor - C:\Users\Asus\Documents\Untitled4.m
Part One_Project.m × Untitled3.m × Untitled4.m × +
        % for part I project
 3
       %Sigma is Standard deviation
        num = 6;
        %Num is Calculated number
        upper = 101.1 + 1.96* sigma / sqrt(num);
       lower = 101.1 - 1.96* sigma / sqrt(num);
      \neg for i = 1:5000
            rand=0;
11 -
            for j = 1:num
12 -
               rand = rand + (randn * sigma + 101.1)/num;
13 -
14 -
           if(lower < rand && rand < upper)
15 -
               Counter = Counter + 1;
16 -
17 -
        disp(Counter * 100 / 5000);
      95.0800
   >> Untitled4
   >> Untitled4
      95.0200
   >> Untitled4
      94.5000
f_{X} >>  Untitled4
```

در هر شش حالت سطح اطمینان 95 درصد به ازای n های مذکور برای دو حالت بدست آمده تایید می شود.

سپس در این قسمت قصد داریم شبیه سازی برای 20 تحقق از اندازه گیری مقاومت با توزیع مثلثی با توزیع نرمال ، یونیفرم و کوشی برای خطا را بدست آوریم.

در واقع در قسمت بعدی ما واریانس یک مفدار نامشخصی بوده و از این نتیجه می شود که حد n نیز یک مقدار نا مشخص دارد.

(عواهد بود) وریانس خواهد بود) و غیار (توان دوی آن برابر واریانس خواهد بود) با فرمول انحراف معیار
$$\sigma=\sqrt{rac{\sqrt{\Sigma(x_i-\mu)^2}}{n-1}}$$
 $n=\left[\left(z_{1-rac{lpha}{2}}+Z_{eta}
ight)
ight]^2\left[\left(rac{\sigma}{d}
ight)
ight]^2$ برای توزیع نرمال نیز داریم :

آزمون و بررسی اعداد را تا زمانی پیش می بریم که به یک مقدار n معینی برسیم. در پیاده سازی کد آن هم باید n را پیدا کنیم که در عبارت بالا صدق کند. بعد از اینکه n را پیدا کردیم مانند بخش اول باید 5000 بار تست کرده و در 95 قرار دهیم.

روش کد نویسی:

برای ادامه حل مسئله یک توزیع مثلثی با حد پایین 100 و حد بالای 101.1 و پیک 100.8 می سازیم برای این کار از دستور makedist استفاده میکنیم بعد از این کار و وارد کردن توزیع ساخته شده برای تعداد ورودی ها (20 عدد) توزیع ساخته شده را با دستور random تولید میکنم و آن را در متغیر Re (مخفف resistor) ذخیره میکنیم .

| 100.8527 | 100.9237 | 100.3343 | 100.9309 | 100.7460 |
|----------|----------|----------|----------|----------|
| 100.2930 | 100.4951 | 100.6937 | 100.9816 | 100.9924 |
| 100.3724 | 101.0015 | 100.9811 | 100.6536 | 100.8433 |
| 100.3534 | 100.6092 | 100.9332 | 100.8381 | 100.9844 |

جدول تحقق مقادير مقاومت ها از توزيع مثلثي

از روابطی که در این قسمت باید استفاده کنیم:

توزیع نرمال
$$n=rac{\sum_{i=1}^n(x_i-\mu)^2}{n-1}(z_{rac{lpha}{2}}+z_{eta})^2$$
 $n=\sum_{i=1}^n(x_i-\mu)^2(n-1)(z_{rac{lpha}{2}}+z_{eta})^2$ t-student توزیع

ابتدا برای توزیع غیر نرمال:

با توجه به میانگین خطای صفر و واریانس 0.3 خواهیم داشت :

$$\sigma^2 = \frac{(b-a)^2}{12} = 0.3 \implies \mu = 0 \implies |a| = |b| = \frac{\sqrt{3.6}}{2}$$

برای واریانس 0.7 نیز داریم:

$$\sigma^2 = \frac{(b-a)^2}{12} = 0.7 \implies \mu = 0 \implies |a| = |b| = \frac{\sqrt{8.4}}{2}$$

: ابتدا برای واریانس 0.7 قدرت تست را بدست می آوریم

حال برای قدرت تست از دستور ztest استفاده میکنیم لذا چون احتمال رد شدن مقاومت های بالا تر 101 برابر 0.9 است لذا مقدار $\beta=0.1$ است پس با 5000 بار تکرار در آزمایش میتوان میزان قدرت تست را آن را بدست می آوریم .

| 0.9000 | 1.0000 | 1.0000 | 1.0000 | 0.2118 |
|--------|--------|--------|--------|--------|
| 1.0000 | 1.0000 | 0.6778 | 1.0000 | 1.0000 |
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.8032 |
| 1.0000 | 1.0000 | 1.0000 | 0.7410 | 1.0000 |

حال می توان عملیات بالا را با دستور ttest نیز انجام داد ، این آزمون برای مواقعی مورد استفاده قرار میگیرد که میزان واریانس معلوم نیست لذا با تعیین نمودن حداقل نمونه تست که در این حالت برابر 3 است مشابه حالت قبل تست را انجام میدهیم و مقادیر زیر حاصل این تست را نشان میدهد :

| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0 |
|--------|--------|--------|--------|--------|
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

: برای واریانس 0.3 نیز مراحل بالا را تکرار و مقادیر زیر بدست می آید

| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.5214 |
|--------|--------|--------|--------|--------|
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

قدرت z test طراحی شده

| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0 |
|--------|--------|--------|--------|--------|
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

قدرت t test طراحی شده

توزيع نرمال:

حال برای توزیع نرمال باید از دستور normrnd استفاده کنیم با میانگین 0 و واریانس 0.7 و 0.7 لذا با توجه با مطالب بالا نتایج زیر بدست خواهد آمد . ابتدا برای واریانس 0.3 قدرت تست را بدست می آوریم :

| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
|--------|--------|--------|--------|--------|
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

قدرت t test طراحی شده

| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.8826 |
|--------|--------|--------|--------|--------|
| 1.0000 | 1.0000 | 0.9998 | 1.0000 | 1.0000 |
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

قدرت z test طراحی شده

برای واریانس 0.7 نیز مراحل بالا را تکرار و مقادیر زیر بدست می آید :

| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
|--------|--------|--------|--------|--------|
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 1.0000 | 1.0000 | 1.0000 | 0 | 1.0000 |
| 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

قدرت t test طراحی شده

| 0.9782 | 1.0000 | 1.0000 | 1.0000 | 0.9138 |
|--------|--------|--------|--------|--------|
| 1.0000 | 1.0000 | 0.9178 | 1.0000 | 1.0000 |
| 1.0000 | 1.0000 | 1.0000 | 0.9926 | 0.9504 |

| 1.0000 | 0.9998 | 1.0000 | 0.9398 | 1.0000 |
|--------|--------|--------|--------|--------|
| | | | | |

قدرت z test طراحی شده

همانطور که مشاهده گردید با توجه به مقدار مقاومت ها و حدودی از بازه اطمینان انتظار داشتیم ، قدرت تست برای هر تحقق قابل قبول بوده و انتظارات ما را تایید میکند .

کد و قسمت شبیه سازی شده توزیع نرمال به صورت زیر است:

```
1 -
      s = abs(randn * 3);
                                                                              Command Window
2 -
      num = 2;
3 -
      x(1) = (rand -0.5)*s*2;
                                                                                  >> Untitled6
 4 -
      x(2) = (rand -0.5)*s*2;
5 -
      sigmatwo = ((x(1) - mean(x))^2) / (num-1) + ((x(2)-mean(x))^2)/(num-1);
                                                                                      94.9200
6 -
    while num <= ceil(sigmatwo * 10.51)
7 -
          num = num + 1;
 8 -
          x(num) = (rand -0.5)*s*2;
                                                                                  >> Untitled6
9 -
          sigmatwo = 0;
                                                                                      94.9400
10 -
          for j = 1:num
11 -
             12 -
                                                                                  >> Untitled6
13 -
14 -
      Counter = 0;
                                                                                      94.7000
      upper = 100 + 1.96* sigmatwo / sqrt(num);
15 -
16 -
      lower = 100 - 1.96* sigmatwo / sqrt(num);
17 -
                                                                                  >> Untitled6
18 -
          rand=0;
19 -
          for i = 1:num
                                                                                      94.6600
20 -
            rand = rand + (randn * sigmatwo + 100)/num;
21 -
22 -
         if(lower < rand && rand < upper)
                                                                                  >> Untitled6
23 -
            Counter = Counter + 1;
                                                                                      95.2600
24 -
        end
25 -
      disp(Counter * 100 / 5000);
26 -
```

کد و قسمت شبیه سازی شده توزیع t-student به صورت زیر است:

```
1
       s = abs(randn * 3);
                                                                                      Command Window
 2
       num = 2;
       x(1) = (rand -0.5)*s*2;
 3
                                                                                               95,1200
 4
       x(2) = (rand -0.5)*s*2;
 5
       sigmatwo = ((x(1) - mean(x))^2) / (num-1) + ((x(2)-mean(x))^2) / (num-1);
     \exists while num <= ceil(sigmatwo * (tinv(0.975, n) + norminv(0.9))^2)
 6
                                                                                               94.7400
           num = num + 1;
 8
           x(num) = (rand -0.5)*s*2;
 9
           sigmatwo = 0;
                                                                                               95.1000
10
           for j = 1:num
               sigmatwo = sigmatwo + ((x(j)-mean(x))^2)/(num-1);
11
12
13
                                                                                               94.7200
14
       Counter = 0;
       upper = 100 + 1.96* sigmatwo / sqrt(num);
15
       lower = 100 - 1.96* sigmatwo / sqrt(num);
16
                                                                                               94.8200
17

\Box
 for i = 1:5000
18
19
           for j = 1:num
                                                                                               95.1800
             rand = rand + (randn * sigmatwo + 100)/num;
21
22
          if(lower < rand && rand < upper)</pre>
23
              Counter = Counter + 1;
24
25
       disp(Counter * 100 / 5000);
26
```

در خروجی های فوق د درصدهای سطح اطمینان حول 95 است.