Домашна работа №2

на
Денис Тихогло
Софтуерно инженерство, 3 курс, 3 група, ф.н. 855256

Щом моят факултетен номер е 855256, **W = 5, A = 2, S = 5, D = 6 = = 0 mod 2**

- **Задача 1. Направете функция, демонстрираща Централната гранична теорема, за случайна величина X:**
  - **X ~ U (min{S, W } − 1, max{S, W } + 1)**

min{S, W} = max{S, W} = 5,

X ~ U (4, 6) – това е разпределението, което ми е дадено

Ще напишем функция f, която приема като параметри размер на извадката n и брой итерации k и връща вектор от извадкови средни.

```
> f = function(n, k){
+   a = 4; b = 6
+   results = c()
+   for(i in 1:k) {
+     X = runif(n = n, min = 4, max = 6)
+     results[i] = mean(X)
+   }
+   return(results)
+ }
```

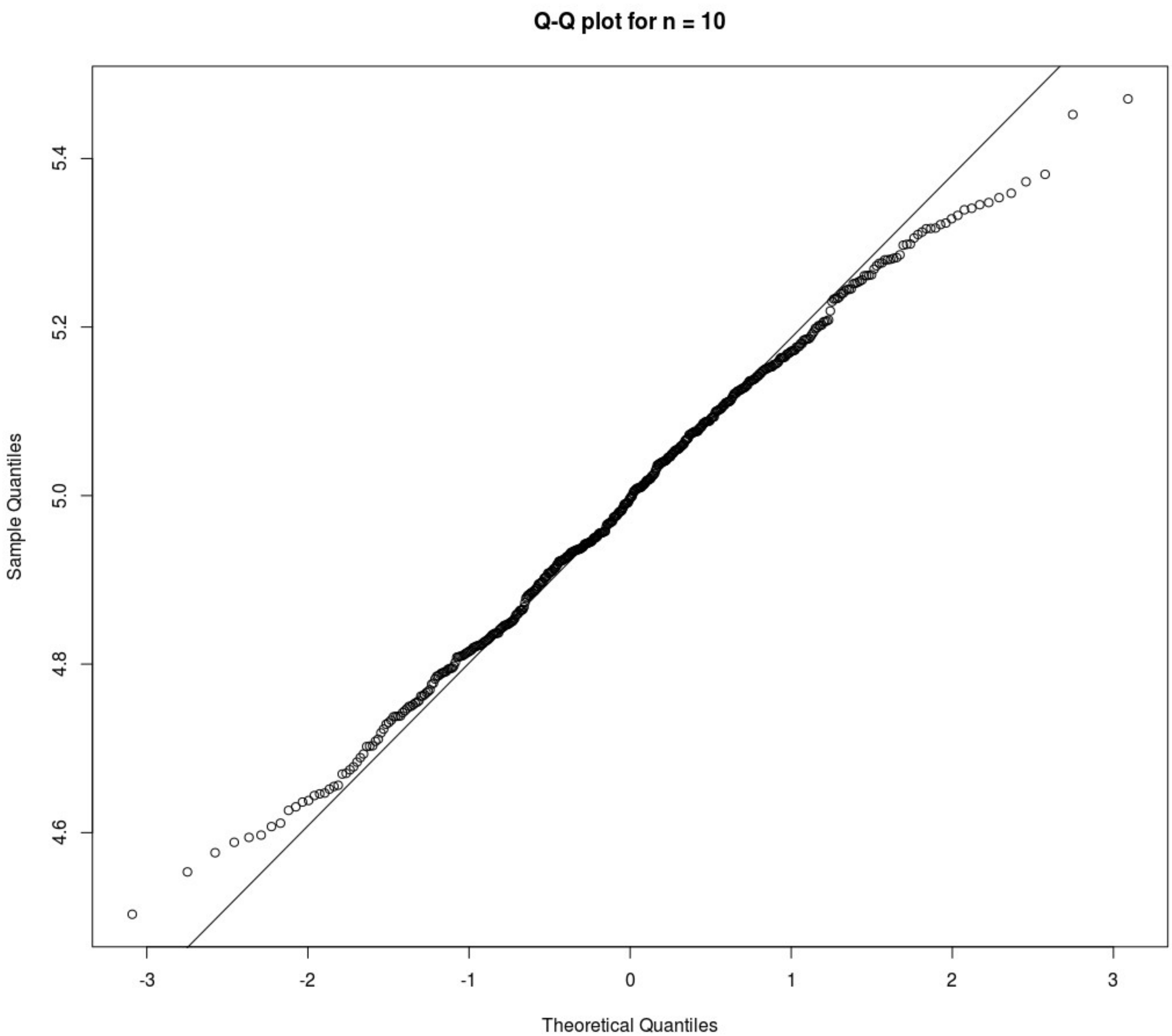
Вътре във функцията цикълт for записва k пъти извадкови средни на равномерното разпределение (което е с параметри a = 4 и b = 6 и с n на брой елементи) във вектор result. Накрая връщам получения вектор results.

> res1 = f(n = 10, k = 500) – изпълнявам функцията за n = 10 и k = 500 и записвам отговора в res1

Рисувам Q-Q графикът за сравнение на разпределението на получения вектор res1 с нормалното

```
> qqnorm(res1, main = "Q-Q plot for n = 10")
```

> qqline(res1) – за базовата линия (reference line)

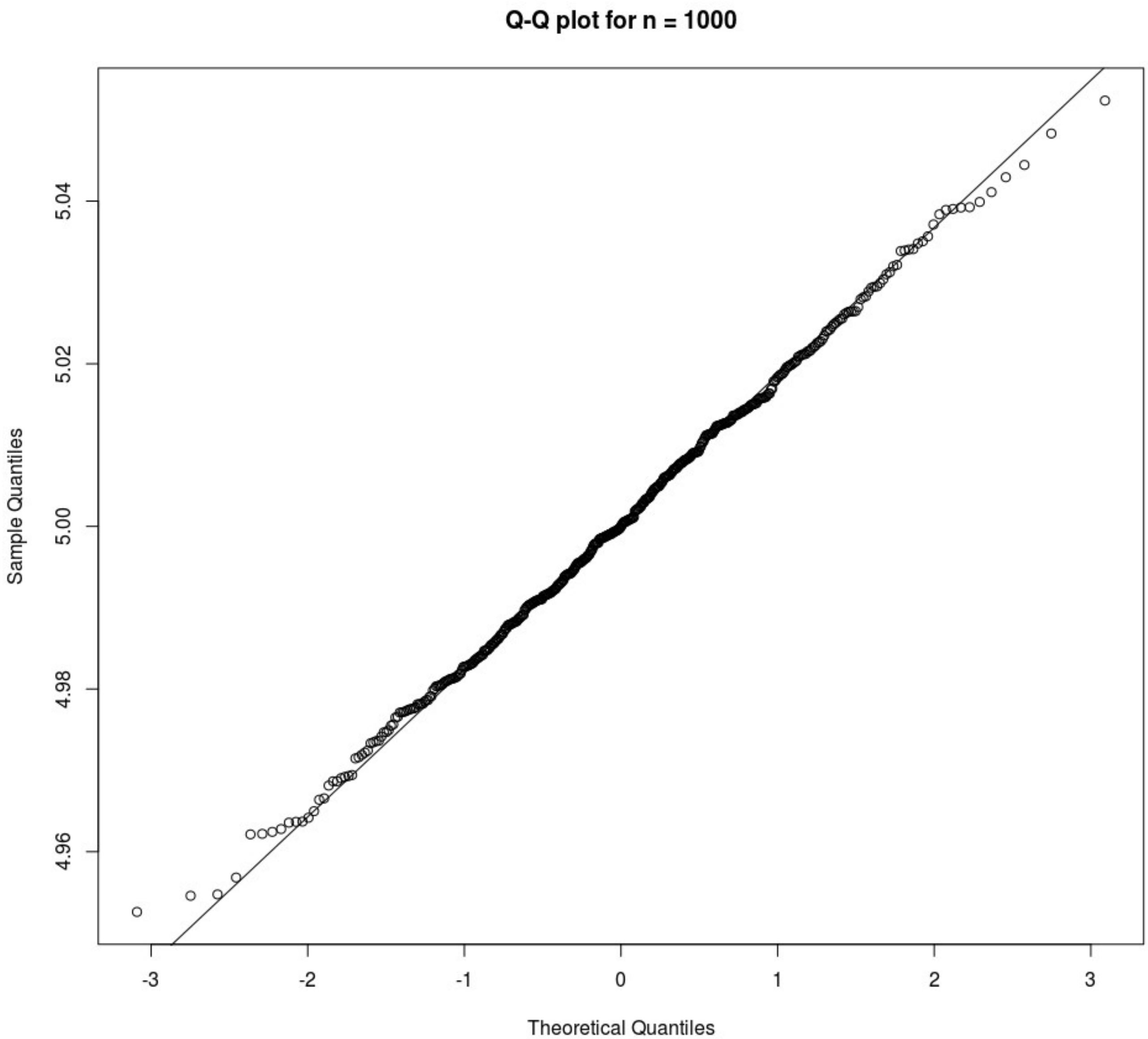


Виждаме, че точките от вектора приблизително съвпадат с линията. От това се прави извод, че разпределението на извадкови средни на даденото равномерно разпределение приблизително (до някаква степен) съвпада с теоретическото нормлно разпределение.

Ще повторя същата процедура за `n = 1000`

`> res2 = f(n = 1000, k = 500)` - изпълнявам функцията за `n = 1000` и `k = 500` и записвам отговора в `res2`

```
> qqnorm(res2, main = "Q-Q plot for n = 1000")
> qqline(res2)
```



В този случай се вижда, че точките са още по-близо към базовата линия.Затова разпределението на извадкови средни за случая с n = 1000 по-хубаво съвпада с теоретическо нормално разпределение в сравнение със същото разпределение с дължина на извадка равна на 10. Това се съгласува с централната гранична теорема: $\frac{\bar{X}-\mu}{\frac{\sigma}{\sqrt{n}}} \rightarrow N(0,1)$ (или $\bar{X} \rightarrow N(\mu, \frac{\sigma^2}{n})$) при $n \rightarrow \infty$. Тъй като в случая n се увеличава, разпределението на средните все повече асимптотично клони към нормалното разпределение.

Забелязваме, че и в двата случая извадкови средни значения се разполагат около 5 – математическто очакване на U(4, 6). Но на втория график се вижда повишаване на плътността на точките близо до средата (практически всички точки са в интервала [4.96, 5.04], докато в първия график точките са примерно в интервала [4.6, 5.4]) . Това е свързано със намаляването на значение на стандартно отклонение ($\frac{\sigma}{\sqrt{n}}$) с увеличаване на n.

- **Задача 2. С помощта на R пресметнете:**
    - **а) P(W + 5 > X ≥ min{2, A}) за сл. в. X ~ Ge((S + D + 9)/90);**
    - **б) P(min{3, A} < Y ≤ W + 4) за сл. в. Y ~ Po(D + 1.5);**
    - **в) стойността z * ,така че P(−z * < Z ≤ z *) = (W + A + S + D + 11)/111 за сл. в. Z ~ N (0, 1)**
    - **г) стойността x, така че P(−1.5 < T ≤ x) = (55 + W + A)/100 за T ~ t(33). Изчислете кой квантил на T е x**
    -

а) S + D + 9 = 20, W + 5 = 10, min{2, A} = 2

X ~ Ge(2/9)

Търси се P(10 > X ≥ 2) = P(2 ≤ X ≤ 9) = P(X ≤ 9) − P(X ≤ 1)

За целта ще се възползвам от функцията pgeom:

```
> pgeom(9, prob=2/9) - pgeom(1, prob=2/9)
```

Отговорът, който се връща, е **0.5239252**

б) min{3, A} = 2, W + 4 = 9, D + 1.5 = 7.5

Y ~ Po(7.5)

Търси се P(2 < Y ≤ 9) = P(3 ≤ Y ≤ 9) = P(Y ≤ 9) − P(Y ≤ 2)

За целта ще се възползвам от функцията ppois:

```
> ppois(9, lambda=7.5) - ppois(2, lambda=7.5)
```

Отговорът, който се връща, е **0.7561509**

в) W + A + S + D = 18

Z ~ N (0, 1)

Търси се z* ,така че P(−z* < Z ≤ z*) = 29/111

Стандартното нормално разпределение е симетрично относно ординатната ос, затова $P(0 < Z \le z*) = 29/222$

$P(0 < Z \le z*) = P(Z \le z*) - P(Z \le 0)$

$P(Z \le 0) = 0.5$ по същата причина (симетричност относно ординатната ос)

Следователно, $P(Z \le z*) = 29/222 + 0.5 = 0,630630631$ – номерът на квантила $z*$.

За намиране на $z*$ ще се възползвам от функцията pnorm:

```
> qnorm(0,630630631, mean=0, sd=1)
```

Отговорът, който се връща, е 0.333524

Следователно **z* = 0.333524**

г) $W + A = 7$

$T \sim t(33)$

Търси се x, така че $P(-1.5 < T \le x)=31/50$ и номерът на квантил x.

$P(-1.5 < T \le x) = P(T \le x) - P(T \le -1.5)$

$P(T \le x) = 31/50 + P(T \le -1.5)$

За намиране на $P(T \le -1.5)$ ще се възползвам от функцията pt:

```
> pt(-1.5, df = 33)
```

Резултатът е 0.07156093

Следователно, $P(T \le x) = 31/50 + 0.07156093 =$ **0.6915609** – номерът на квантила x

За намиране на x ще се възползвам от функцията qt:

```
> qt(0.6915609, df = 33)
```

Отговорът, който се връща, е 0.5050528

Следователно **x = 0.5050528**

- **Задача 3. Нека имаме произволна пермутация на числата от 1 до 100. С помощта на симулации пресметнете приближено вероятността числото 12 да се намира след числото 20 и преди числото 16.**

Нека X е случайната величина – „12 се намира след числото 20 и преди числото 16". Очевидно, това разпределение е от Бернулиевия тип: приема само 2 значения – TRUE (1) и FALSE(0). Вероятността p, което е параметърът на случайна величина X, ще намеря приближено с помощта на симулации.

Първо ще напиша функция, която моделира условието и връща TRUE или FALSE в зависимост от изхода на експеримента.

```
> getResult = function(){
+    s = sample(1:100, 100, replace=FALSE)
+    ind12 = which(s==12)
+    ind16 = which(s==16)
+    ind20 = which(s==20)
+    if (ind12 > ind20 && ind12 < ind16){
+       return(TRUE )
+    }
+    else {
+       return(FALSE )
+    }
+ }
```

Първо с помощта на функцията sample, получавам пермутация на числа от 0 до 100 (replace=FALSE, защото пермутация не допуска повторения на стойности). Записвам получения вектор в променливата s. След това намирам индексите на числа 12, 16 и 20 и ги записвам съответно в променливите ind12, ind16 и ind20. Накрая правя проверка, дали индекс на число 12 се намира между ind20 и ind16. Ако това е изпълнено, връщам TRUE, в противния случай връщам FALSE.

За да намерим приближено вероятността опит да е успешен (getResult да върне TRUE), трябва да го изпълним голям брой пъти и да изчислим отношението на брой „успешни" опити върху общ брой опити.

За целта ще напиша функция getProbability, приемаща параметър iteraions – общия брой експерименти:

```
> getProbability = function(iterations){
+    results = c()
+    for (i in 1:iterations){
```

```
+         results[i] =  getResult()
+     }
+     return(sum(results==TRUE)/iterations)
+ }
```

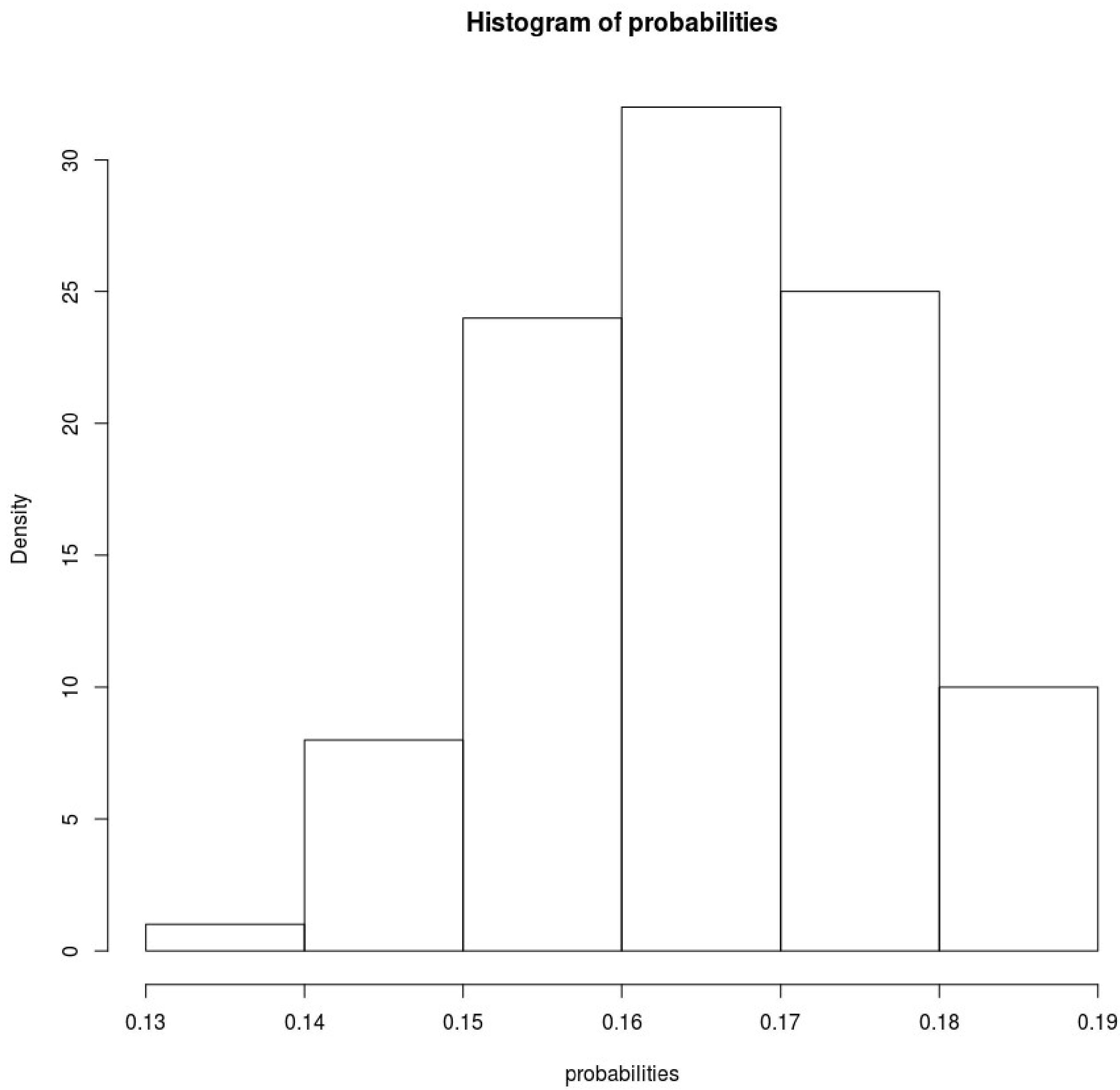
Вътре във функцията векторът results се попълва със стойности, върнати от getResult(). Наркая той съдържа iterations на брой резултати. Функцията връща отношение на брой значения TRUE на вектора results към дължина на вектора.

Сега, изпълнявайки функцията getProbability, можем да получим търсената приближена вероятност. Вдигайки значениетона iterations получаваме по-прецизни резултати. Този факт следва от централната гранична теорема. С повишаване на брой итерации, стандартното отклонение на извадковото средно намалява и стойностите се разполагат по-плътно към математическото очакване.

Можем да построим хистограмма на средните значения и да се убедим, че те приличат на нормалното разпределение:

```
> probabilities = c()
> for (j in 1:100){
+     probabilities[j] = getProbability(1000)
+ }
> hist(probailities)
```

Записвам приближените вероятности във вектора probabilities и рисувам хистограмата:



Намирам финалната приближена вероятност усреднявайки получените приближени вероятности:

```
> mean(probabilities)
```

Резултатът е примерно равен на 0.17.

То ест, вероятността числото 12 да се намира след числото 20 и преди числото 16 примерно е равна на **17%**