

СТАТИСТИКА И ЕМПИРИЧНИ МЕТОДИ

Домашно 2

Антоан Венциславов Стефанов

Ф.Н.: 61797

гр.5

Софтуерно инженерство

21.12.2016

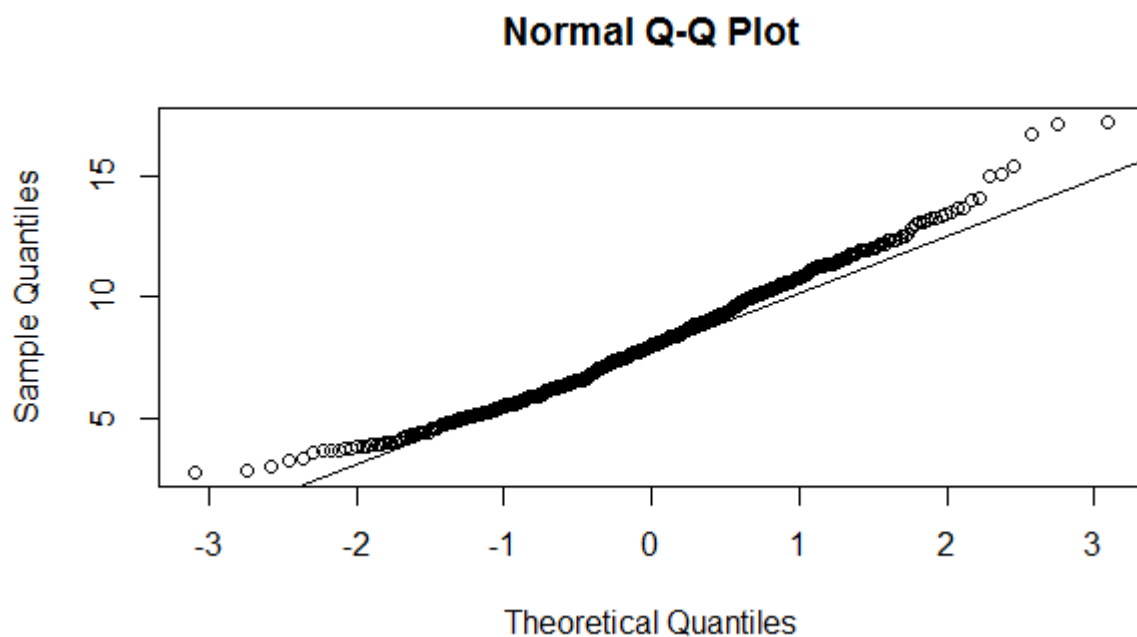
Задача 1.

Направете функция, демонстрираща Централната гранична теорема, за случайна величина X :

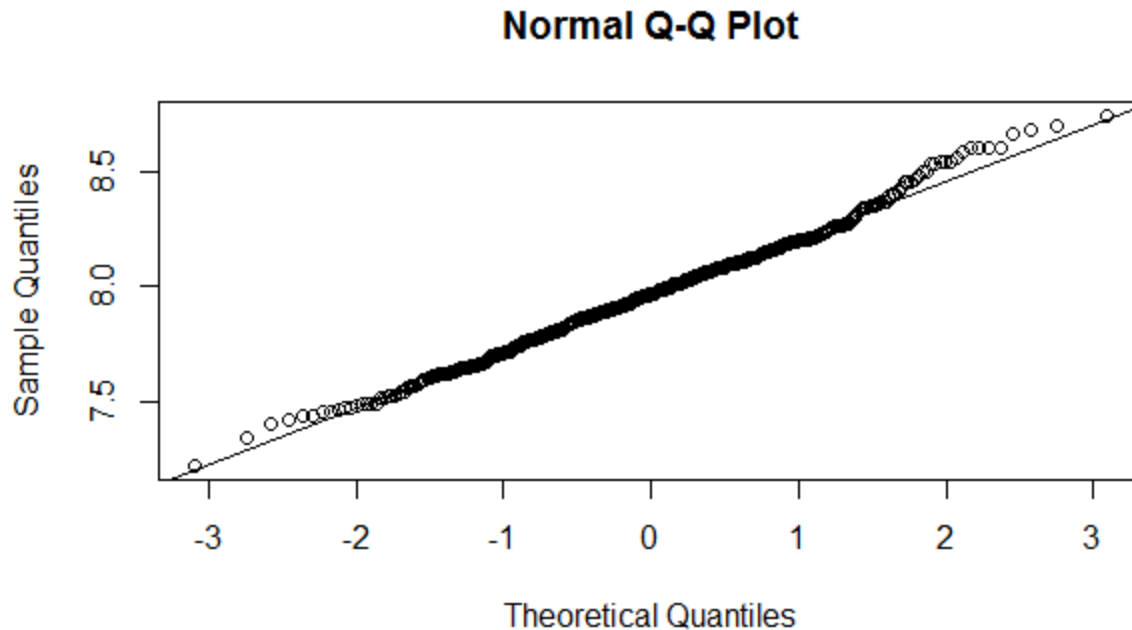
$$X \sim \text{Exp}(1/8)$$

```
theoremFunc <- function(n, k) {  
  vec = c();  
  for(i in 1:k) {  
    avrNum = mean(sample(x = data, size = n))  
    vec <- c(vec, avrNum);  
  }  
  return (vec);  
}
```

```
qqnorm(theoremFunc(10, 500))  
qqline(theoremFunc(10, 500))
```



```
qqnorm(theoremFunc(1000, 500))
qqline(theoremFunc(1000, 500))
```



Очевидно разпределението на резултатите при $n=1000$ е по-близо до нормалното, отколкото при $n=10$. Това е така, защото приближението на едно разпределение до нормалното става по-точно с увеличаването на големината на извадката (Според Теоремата за .

Задача 2.

С помощта на R пресметнете:

2.а) $P(6 > X \geq 2)$, за сл. в. $X \sim \text{Ge}(25/90)$

```
task2A <- function () {
  param = 25/90
  return (dnbinom(2, 1, param) + dnbinom(3, 1, param) + dnbinom(4, 1, param) +
  dnbinom(5, 1, param));
}
```

```
[1] 0.379691
```

2.6)

```
task2B <- function () {  
  return (dpois(4, 7.5) + dpois(5, 7.5));  
}
```

```
[1] 0.182291
```

2.B)

```
task2V<- function (mejdinnaPlosht) {  
  area = 37 / 111;  
  leftQuantile = (1-area)/2;  
  minusZ = qnorm(leftQuantile, mean = 0, sd = 1);  
  rightQuantile = leftQuantile + area;  
  z = qnorm(rightQuantile, mean = 0, sd = 1);  
  return (z);} 
```

```
[1] 0.4307273
```

2.r)

```
task2G <- function () {  
  middleArea = 63/100  
  leftQuantile = -1.5  
  areaToLeftQuantile = pt(leftQuantile, 33);  
  areaToX = areaToLeftQuantile + middleArea;  
  x = qt(areaToX, 33);  
  return (x);  
}  
[1] 0.5340623
```

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.

```
simulationFunction <- function(n) {  
  counter = 0;  
  for (i in {1:n}) {  
    numbers = sample({1:100}, 100);  
    indexTwelve = which(numbers == 12);  
    indexTwenty = which(numbers == 20);  
    indexSixteen = which(numbers == 16);  
    if (indexTwelve > indexTwenty && indexTwelve < indexSixteen) {  
      counter = counter + 1;  
    }  
  }  
  return (counter/n);  
}
```

```
simulationFunction(100)
```

```
[1] 0.18
```

```
simulationFunction(10000)
```

```
[1] 0.16645
```

```
simulationFunction(1000000)
```

```
[1] 0.16682
```

Виждаме, че с увеличаване на повторенията, резултатът се приближава повече и повече до 0.166, което е вероятността 12 да е след 20 и преди 16.