Вероятности и статистика: Упражнения с R

Ангел Г. Ангелов
agangelov@fmi.uni-sofia.bg

20. XI. 2022 г.

Съдържание

1. Въведение в R

Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program.
Linus Torvalds

R е език и среда за статистически изчисления и анализ. Разпространява се свободно, съгласно условията на GNU General Public License. Създаден е първоначално през 1993 от Robert Gentleman и Ross Ihaka от Департамента по статистика на Университета Оукланд (University of Auckland) на основа на езика S. На сайта `https://cran.r-project.org/` може да се намери последната версия.

1.1. Математически функции

```
> (5+7)/(4-1)
[1] 4

> 9^2
[1] 81

> sqrt(25)
[1] 5

> log(exp(1))
[1] 1

> 28 %% 10
[1] 8

> 27/1000000
[1] 2.7e-05

> 5000*5000
[1] 2.5e+07

> options(scipen=999)
> 27/1000000
[1] 0.000027

> options(scipen=0)
> 27/1000000
[1] 2.7e-05
```

1.2. Вектори

```
> x <- c(5, 12, 11, 14, 2, 3, 14, 10, 3)

> x[3]
[1] 11
```

```
> x[1:5]
[1]  5 12 11 14  2

> x[c(2,5,9)]
[1] 12  2  3

> x[-4]
[1]  5 12 11  2  3 14 10  3

> x[-c(2,3)]
[1]  5 14  2  3 14 10  3

> x[x>10]
[1] 12 11 14 14

> length(x)
[1] 9

> min(x)
[1] 2

> max(x)
[1] 14

> head(x, 3)
[1]  5 12 11

> tail(x, 3)
[1] 14 10  3

> x>10
[1] FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE

> sum(x>10)
[1] 4

> which(x>10)
[1] 2 3 4 7

> diff(x)
[1]   7  -1   3 -12   1  11  -4  -7

> cumsum(x)
[1]  5 17 28 42 44 47 61 71 74

> sum(x)
[1] 74

> x^2
[1]  25 144 121 196   4   9 196 100   9

> sort(x)
[1]  2  3  3  5 10 11 12 14 14
```

```
> x[ order(x) ]
[1]  2  3  3  5 10 11 12 14 14

> rank(x)
[1] 4.0 7.0 6.0 8.5 1.0 2.5 8.5 5.0 2.5

> rm(x)
> x
Error: object 'x' not found

> x <- c(1,3,5,11,15)
> class(x)
[1] "numeric"

> x <- as.integer(c(1,3,5,11,15))
> class(x)
[1] "integer"

> y <- c("Y", "Y", "N")
> class(y)
[1] "character"

> z <- c(TRUE, TRUE, FALSE)
> class(z)
[1] "logical"

> x <- vector("logical", length=5)
> x
[1] FALSE FALSE FALSE FALSE FALSE

> y <- vector("numeric", length=5)
> y
[1] 0 0 0 0 0

> x <- c(5,5,5,7,7,7)
> y <- c(2,2,1)

> x+y
[1] 7 7 6 9 9 8

> y <- c(2,2,1,1)
> x+y
[1] 7 7 6 8 9 9
Warning message:
In x + y : longer object length is not a multiple of shorter object length
```

1.3. Генериране на редици

```
> rep( 5, times=8 )
[1] 5 5 5 5 5 5 5 5
> rep( c(1,2), times=5 )
```

```
[1] 1 2 1 2 1 2 1 2 1 2
> rep( c(1,2), each=5 )
[1] 1 1 1 1 1 2 2 2 2 2
> rep( c(1,2), length.out=7 )
[1] 1 2 1 2 1 2 1

> rep( c("a","b"), times=5 )
[1] "a" "b" "a" "b" "a" "b" "a" "b" "a" "b"
> rep( c("a","b"), each=3 )
[1] "a" "a" "a" "b" "b" "b"

> 5:12
[1]  5  6  7  8  9 10 11 12
> 10:1
 [1] 10  9  8  7  6  5  4  3  2  1
> seq( from=1, to=10, by=2 )
[1] 1 3 5 7 9
> seq( from=10, to=1, by=-2 )
[1] 10  8  6  4  2
> seq( from=0, to=1, by=0.2 )
[1] 0.0 0.2 0.4 0.6 0.8 1.0
> seq( from=0, to=1, length.out=11 )
[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
>
```

1.4. Матрици

```
> M <- rbind( c(5,3,5,6), c(8,3,7,4) )
> M
     [,1] [,2] [,3] [,4]
[1,]    5    3    5    6
[2,]    8    3    7    4

> M[2, 3]
[1] 7

> M[ ,3]
[1] 5 7

> M[2, ]
[1] 8 3 7 4

> M <- cbind( c(5,3,5,6), c(8,3,7,4) )
> M
     [,1] [,2]
[1,]    5    8
[2,]    3    3
[3,]    5    7
[4,]    6    4

> t(M)
     [,1] [,2] [,3] [,4]
```

```
[1,]    5    3    5    6
[2,]    8    3    7    4

> M[ c(3,1), ]
     [,1] [,2]
[1,]    5    7
[2,]    5    8

> M[ order( M[,1] ), ]
     [,1] [,2]
[1,]    3    3
[2,]    5    8
[3,]    5    7
[4,]    6    4

> M[ order( M[,1], M[,2] ), ]
     [,1] [,2]
[1,]    3    3
[2,]    5    7
[3,]    5    8
[4,]    6    4

> M <- matrix( c(1:12), nrow=3, ncol=4 )
> M
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

> M <- matrix( c(1:12), nrow=3, ncol=4, byrow=TRUE )
> M
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12

> head(M, 2)
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8

> tail(M, 2)
     [,1] [,2] [,3] [,4]
[2,]    5    6    7    8
[3,]    9   10   11   12

> sqrt(M)
         [,1]     [,2]     [,3]     [,4]
[1,] 1.000000 1.414214 1.732051 2.000000
[2,] 2.236068 2.449490 2.645751 2.828427
[3,] 3.000000 3.162278 3.316625 3.464102

> rownames(M) <- c("a", "b", "c")
```

```
> colnames(M) <- c("X1", "X2", "X3", "X4")
> M
  X1 X2 X3 X4
a  1  2  3  4
b  5  6  7  8
c  9 10 11 12
```

1.5. Data frame

```
> x <- c(5, 8, 11, 3, 2, 9, 4)
> y <- c("Y", "Y", "N", "Y", "N", "N", "Y")
> df <- data.frame(x,y)
> df
   x y
1  5 Y
2  8 Y
3 11 N
4  3 Y
5  2 N
6  9 N
7  4 Y

> str(df)
'data.frame':   7 obs. of  2 variables:
 $ x: num  5 8 11 3 2 9 4
 $ y: chr  "Y" "Y" "N" "Y" ...

> df$x
[1]  5  8 11  3  2  9  4

> df$y
[1] "Y" "Y" "N" "Y" "N" "N" "Y"

> df$x[4]
[1] 3

> df[ ,1]
[1]  5  8 11  3  2  9  4

> df[5, ]
  x y
5 2 N

> df[ , "x"]
[1]  5  8 11  3  2  9  4

> df$z <- seq(from=1, to=14, by=2)
> str(df)
'data.frame':   7 obs. of  3 variables:
 $ x: num  5 8 11 3 2 9 4
 $ y: chr  "Y" "Y" "N" "Y" ...
 $ z: num  1 3 5 7 9 11 13
```

```
> df[ 3, c("x","z") ]
   x z
3 11 5

> df[ c(5,7), c(2,3) ]
  y  z
5 N  9
7 Y 13

> df$x[ df$z <= 5 ]
[1]  5  8 11

> df$x[ df$y == "N" ]
[1] 11  2  9

> df[ df$z <= 5, c("x","z") ]
   x z
1  5 1
2  8 3
3 11 5
```

1.6. Полезни функции

```
getwd()
setwd(dir)
save(...)
save.image(...)
read.table(file)
write.table(x, file)
replace(x, list, values)
ifelse(test, yes, no)
any(...)
all(...)
unique(...)
duplicated(...)
is.element(x, y)
x %in% y
tabulate(...)
substr(x, start, stop)
```

2. Случайни експерименти

The most important questions of life are indeed,
for the most part, only problems of probability.

The theory of probability is only
common sense reduced to calculation.
Pierre Simon Laplace

Случаен експеримент наричаме експеримент, при който не знаем предварително какъв ще бъде резултата (изхода), но знаем какви са възможните изходи. Пример – при хвърляне на зар знаем, че ще се падне някоя от страните на зара, но не знаем коя.

Нека A е някакво събитие, което може да се случи при извършване на експеримента (или да не се случи). Например, при хвърляне на зар – пада се нечетно число.

На всяко събитие съпоставяме число между 0 и 1, което наричаме *вероятност* на събитието. Вероятността на събитието A означаваме с $\mathbf{P}(A)$.

Повтаряме експеримента n пъти, при едни и същи условия. Да означим с $c_n(A)$ броя случвания на събитието A при n повторения на експеримента, т.е. събитието A се е случило $c_n(A)$ пъти. Тогава за достатъчно големи n е изпълнено

$$\frac{c_n(A)}{n} \approx \mathbf{P}(A).$$

Това твърдение следва от т.нар. *закон за големите числа* в теорията на вероятностите. Например, ако хвърляме зар 1000 пъти и $A = \{$пада се нечетно число$\}$ и нечетно число се падне $c_n(A)$ пъти, $c_n(A)/1000$ е приблизително равно на вероятността на A.

Числото $\dfrac{c_n(A)}{n}$ наричаме *честота* на събитието A. Понякога се нарича относителна честота (*relative frequency*). Законът за големите числа твърди, че при достатъчно повторения на експеримента, честотата на случване на събитието A ще приближава вероятността на A. В известен смисъл вероятността $\mathbf{P}(A)$ е дефинирана така, че $\dfrac{c_n(A)}{n}$ да клони към $\mathbf{P}(A)$.

Нека B е друго събитие, което може да се случи при извършване на експеримента. Условна вероятност на A при условие B, т.е. вероятността да се случи A, ако имаме информация, че се е случило B, се дефинира така $\mathbf{P}(A \mid B) = \mathbf{P}(AB)/\mathbf{P}(B)$.

Означаваме с $c_n(AB)$ броя случвания на събитията A и B едновременно. Тогава за достатъчно големи n е изпълнено

$$\frac{c_n(AB)}{c_n(B)} \approx \mathbf{P}(A \mid B).$$

Числото $\dfrac{c_n(AB)}{c_n(B)}$ ни показва колко често се е случило събитието A, ако броим само експериментите, при които се е случило събитието B.

В следващите задачи ще намерим приближение за вероятността на дадено събитие като симулираме експеримента достатъчен брой пъти с помощта на R и използваме горните твърдения.

Функцията `sample(x, size, replace)` генерира определен брой (`size`) случайно избрани елементи от вектора `x`, с връщане (`replace=T`) или без връщане (`replace=F`).

Например, по следния начин генерираме 5 случайни числа от вектора $(1, 2, 3, \ldots, 10)$, с връщане:

```
> sample( c(1:10), 5, replace=T )
[1] 7 1 6 2 5
> sample( c(1:10), 5, replace=T )
[1] 5 1 1 8 5
```

По следния начин генерираме случайна пермутация на елементите на вектора $(1, 2, \ldots, 7)$:

```
> sample( c(1:7), 7, replace=F )
[1] 5 7 1 6 2 3 4
```

Задача 2.1. В отдел на фирма работят 20 човека. За Коледа те решават да си разменят подаръци. В кутия слагат 20 листчета, на всяко от които има едно име. Всеки тегли листче (без да го връща) и подарява на този, чието име е изтеглил. Каква е вероятността поне един да изтегли своето име?

```
sim.gifts <- function(k) {
  x <- sample( c(1:k), k, replace=F )
  d <- x - c(1:k)
  any(d==0)
}

prob.gifts <- function(Nrep, k) {
  rs <- replicate( Nrep, sim.gifts(k) )
  sum(rs)/length(rs)
}

prob.gifts(100000, 20)
```

Функцията `sim.gifts(k)` симулира един експеримент (за `k` човека) и връща `TRUE` ако се е случило събитието поне един да изтегли своето име.

Функцията `prob.gifts(Nrep, k)` повтаря експеримента `Nrep` пъти и връща честотата на случване на събитието (брой случвания разделен на брой повторения), която използваме като приближение на вероятността.

Задача 2.2. Каква е вероятността в група от 25 човека поне двама да имат рожден ден на един и същи ден от годината?

```
sim.bday <- function(k) {
  x <- sample( c(1:365), k, replace=T )
  anyDuplicated(x) > 0
}

prob.bday <- function(Nrep, k) {
  rs <- replicate( Nrep, sim.bday(k) )
  sum(rs)/length(rs)
}

prob.bday(100000, 25)
```

Задача 2.3. Иван има 5 ключа, но не знае кой е за неговата стая. Той пробва последователно с всеки от тях, като помни кой ключ е пробвал. Каква е вероятността да отключи с петия ключ?

```
sim.keys <- function() {
  x <- sample( c(1:5), 5, replace=F )
  x[5]==1
}

prob.keys <- function(Nrep) {
  rs <- replicate( Nrep, sim.keys() )
  sum(rs)/length(rs)
}

prob.keys(100000)
```

Задача 2.4. На всеки от върховете на равностранен триъгълник има една мравка. Всяка мравка избира произволно един от другите два върха и тръгва към него. За единица време всяка мравка изминава разстоянието от един връх до друг. Две мравки могат да се разминат ако тръгнат една срещу друга. Каква е вероятността след единица време да има по една мравка на всеки връх?

```
sim.ants <- function() {
  a <- vector("numeric", 3)
  a[1] <- sample( c(2,3), 1 )
  a[2] <- sample( c(1,3), 1 )
  a[3] <- sample( c(1,2), 1 )
  all( c(1,2,3) %in% a )
}

prob.ants <- function(Nrep) {
  rs <- replicate( Nrep, sim.ants() )
  sum(rs)/length(rs)
}

prob.ants(100000)
```

Задача 2.5. Имаме 3 карти: първата е бяла от двете страни, втората е черна от двете страни, а третата е бяла от едната и черна от другата страна. Всяка карта е поставена в затворена кутия. Избираме произволна кутия, отваряме я и виждаме, че горната страна на картата в нея е бяла. Каква е вероятността другата страна на картата също да е бяла?

```
sim.bw <- function() {
  card <- sample( c("bb", "ww", "bw"), 1 )
  side <- sample( c(1,2), 1 )
  up <- substr( card, start=side, stop=side )
  c(up, card)
}
```

```
prob.bw <- function(Nrep) {
  rs <- replicate( Nrep, sim.bw() )
  sum(rs[2,]=="ww") / sum(rs[1,]=="w")
}

prob.bw(100000)
```

3. Случайни величини

The human mind treats a new idea the same way
the body treats a strange protein; it rejects it.
P.B. Medawar

Често с изхода на даден случаен експеримент свързваме някаква числова величина, например сумата от точките при хвърляне на два зара, броя дефектни продукти в дадена партида, времето на живот на батерия. Такава числова величина наричаме *случайна величина* (сл.в.). Предварително не знаем каква ще е стойността на случайната величина, но знаем какви са възможните ѝ стойности. Конкретната стойност на случайната величина при извършване на експеримента се определя еднозначно от изхода му, например ако хвърляме два зара и се падне $\boxed{::}$ и $\boxed{\cdot}$ стойността на сл.в. „сума от точките на двата зара“ ще е 5.

3.1. Дискретни сл.в.

Когато случайната величина може да приема краен брой стойности или стойности от изброимо множество (например целите числа $\{0, \pm 1 \pm 2, \ldots\}$ или естествените числа $\{1, 2, 3, \ldots\}$), се нарича *дискретна* случайна величина.

Нека $x_1, x_2, \ldots, x_k, \ldots$ са възможните стойности на дискретната сл.в. X. Вероятността да наблюдаваме стойност x_k при извършване на експеримента означаваме $\mathbf{P}(X = x_k)$. Сумата на вероятностите $\mathbf{P}(X = x_k)$ е единица: $\sum_k \mathbf{P}(X = x_k) = 1$.

Възможните стойности на дискретната случайна величина и техните вероятности обикновено се записват в таблица от вида:

x_1	x_2	$\ldots$	x_k	$\ldots$
$\mathbf{P}(X = x_1)$	$\mathbf{P}(X = x_2)$	$\ldots$	$\mathbf{P}(X = x_k)$	$\ldots$

Дискретната сл.в. е дефинирана, ако знаем възможните ѝ стойности и техните вероятности.

Математическо очакване (средно) на дискретната сл.в. X наричаме числото

$$\mathrm{E}(X) = \mu = \sum_k x_k \, \mathbf{P}(X = x_k).$$

Дисперсия на дискретната сл.в. X наричаме числото

$$\mathrm{Var}(X) = \sigma^2 = \mathrm{E}(X - \mu)^2 = \mathrm{E}(X^2) - \mu^2 \quad = \sum_k (x_k)^2 \, \mathbf{P}(X = x_k) - \mu^2.$$

3.1.1. Бернулиево разпределение

Разглеждаме експеримент (опит), при който може да се случи събитието A, което условно наричаме *успех*, или да не се случи събитието A, т.е. да се случи допълнението $A^c = \overline{A}$, което наричаме *неуспех*. Такъв опит наричаме *Бернулиев опит*. Например, при хвърляне на монета се интересуваме от събитието $A = \{\text{пада се ези}\}$ и го наричаме „успех“, падането на тура ще е „неуспех“. Ако при хвърляне на зар се интересуваме от това дали се е паднала шестица, за нас „успех“ ще бъде падането на шестица, а „неуспех“ – падането на число различно от шестица.

Дефинираме случайна величина X по следния начин:

$$X = \begin{cases} 1, & \text{при } \textit{успех} \\ 0, & \text{при } \textit{неуспех} \end{cases}$$

$$\mathbf{P}(X = 1) = p, \quad \mathbf{P}(X = 0) = 1 - p = q$$

Казваме, че случайната величина X има Бернулиево разпределение с параметър p.

3.1.2. Биномно разпределение

Да разгледаме поредица от n независими Бернулиеви опити с една и съща вероятност за *успех* p и нека $q = 1 - p$. Нека X е броя успехи в тази поредица опити.

$$\mathbf{P}(X = k) = \binom{n}{k} p^k q^{n-k}, \quad k = 0, 1, 2, \ldots, n.$$

Случайната величина X наричаме биномно разпределена с параметри n, p и означаваме $X \in \mathrm{Bi}(n, p)$ или $X \sim \mathrm{Bi}(n, p)$.

Средно и дисперсия:

$$\mathrm{E}(X) = np, \qquad \mathrm{Var}(X) = npq.$$

`dbinom(`k`, `n`, `p`)` $= \mathbf{P}(X = k) = \binom{n}{k} p^k q^{n-k}$.

`pbinom(`k`, `n`, `p`)` $= \mathbf{P}(X \leq k)$.

`rbinom(`N`, `n`, `p`)` генерира N случайни числа от биномно разпределение с параметри n, p.

Ако X_1 има Бернулиево разпределение с параметър p, то $X_1 \sim \mathrm{Bi}(1, p)$. Ако $X_1, X_2, \ldots, X_n$ са независими Бенулиеви сл.в. с параметър p, то сумата им има биномно разпределение: $X_1 + X_2 + \ldots + X_n \sim \mathrm{Bi}(n, p)$.

3.1.3. Геометрично разпределение

Разглеждаме поредица от независими Бернулиеви опити с вероятност за *успех* p и нека $q = 1 - p$. Нека X е броя опити до първия успех (включително), с други думи, първият успех е на X-тия опит.

$$\mathbf{P}(X = k) = q^{k-1} p, \quad k = 1, 2, 3, \ldots$$

Случайната величина X наричаме геометрично разпределена с параметър p и означаваме $X \sim \mathrm{Ge}(p)$.

Средно и дисперсия:

$$\mathrm{E}(X) = \frac{1}{p}, \qquad \mathrm{Var}(X) = \frac{q}{p^2}.$$

`dgeom(`$k-1$`, `p`)` $= \mathbf{P}(X = k) = q^{k-1} p$.

`pgeom(`$k-1$`, `p`)` $= \mathbf{P}(X \leq k)$.

`rgeom(`N`, `p`) + 1` генерира N случайни числа от $\mathrm{Ge}(p)$.

⟨!⟩ Понякога случайната величина $Y =$ брой неуспехи преди първия успех в поредица от независими Бернулиеви опити, също се нарича геометрично разпределена. Очевидно $X = Y + 1$. Ще използваме означението $Y \sim \mathrm{Ge}^*(p)$.

$$\mathbf{P}(Y = k) = \mathbf{P}(X = k + 1) = q^k p, \quad k = 0, 1, 2, \ldots$$

Средно и дисперсия:

$$\mathrm{E}(Y) = \mathrm{E}(X - 1) = \frac{1}{p} - 1 = \frac{q}{p},$$

$$\mathrm{Var}(Y) = \mathrm{Var}(X - 1) = \mathrm{Var}(X) = \frac{q}{p^2}.$$

`dgeom(`k`, p)` $= \mathbf{P}(Y = k) = q^k p$.

`pgeom(`k`, p)` $= \mathbf{P}(Y \le k)$.

`rgeom(`N`, p)` генерира N случайни числа от $\mathrm{Ge}^*(p)$.

3.1.4. Отрицателно биномно разпределение

Разглеждаме поредица от независими Бернулиеви опити с вероятност за *успех* p и нека $q = 1 - p$. Нека X е броя опити до r-тия успех (включително), с други думи, r-тият успех е на X-тия опит (r е фиксирано цяло число).

$$\mathbf{P}(X = k) = \binom{k-1}{r-1} p^r q^{k-r}, \quad k = r, r + 1, r + 2, \ldots$$

Случайната величина X наричаме отрицателно биномно разпределена с параметри r, p и означаваме $X \sim \mathrm{NB}(r, p)$.

Средно и дисперсия:

$$\mathrm{E}(X) = \frac{r}{p}, \qquad \mathrm{Var}(X) = \frac{rq}{p^2}.$$

`dnbinom(`$k - r$`, `r`, `p`)` $= \mathbf{P}(X = k) = \binom{k-1}{r-1} p^r q^{k-r}$.

`pnbinom(`$k - r$`, `r`, `p`)` $= \mathbf{P}(X \le k)$.

`rnbinom(`N`, `r`, `p`) + `r генерира N случайни числа от $\mathrm{NB}(r, p)$.

⟨!⟩ Понякога случайната величина $Y =$ брой неуспехи преди r-тия успех в поредица от независими Бернулиеви опити, също се нарича отрицателно биномно разпределена. Очевидно $X = Y + r$. Ще използваме означението $Y \sim \mathrm{NB}^*(r, p)$.

$$\mathbf{P}(Y = k) = \binom{r + k - 1}{r - 1} p^r q^k, \quad k = 0, 1, 2, \ldots$$

Средно и дисперсия:

$$\mathrm{E}(Y) = \frac{rq}{p}, \qquad \mathrm{Var}(Y) = \frac{rq}{p^2}.$$

`dnbinom(`k`, `r`, `p`)` $= \mathbf{P}(Y = k) = \binom{r+k-1}{r-1} p^r q^k$.

`pnbinom(`k`, `r`, `p`)` $= \mathbf{P}(Y \le k)$.

`rnbinom(`N`, `r`, `p`)` генерира N случайни числа от $\mathrm{NB}^*(r, p)$.

3.1.5. Хипергеометрично разпределение

В кутия има M бели и $N-M$ черни топки. Вадим n топки без да ги връщаме. Нека X е броя на извадените бели топки.

$$\mathbf{P}(X=k)=\frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}},\quad k=0,1,2,\ldots,n.$$

Случайната величина X наричаме хипергеометрично разпределена с параметри N, M, n и означаваме $X \sim \mathrm{HG}(N,M,n)$. В горната формула приемаме, че $\binom{n}{k}=0$ ако $k<0$ или $n<k$. Вероятностите $\mathbf{P}(X=k)$ са положителни за $\max(0,n-N+M)\le k\le \min(M,n)$.

Средно и дисперсия:

$$\mathrm{E}(X)=n\,\frac{M}{N},\qquad \mathrm{Var}(X)=n\,\frac{M}{N}\,\frac{N-M}{N}\,\frac{N-n}{N-1}.$$

`dhyper(`k`, `M`, `$N-M$`, `n`)` $=\mathbf{P}(X=k)=\dfrac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}}$.

`phyper(`k`, `M`, `$N-M$`, `n`)` $=\mathbf{P}(X\le k)$.

`rhyper(`R`, `M`, `$N-M$`, `n`)` генерира R случайни числа от хипергеометрично разпределение с параметри N, M, n.

3.1.6. Поасоново разпределение

Казваме, че случайната величина X има Поасоново разпределение с параметър $\lambda>0$ и означаваме $X\sim \mathrm{Po}(\lambda)$, ако

$$\mathbf{P}(X=k)=e^{-\lambda}\,\frac{\lambda^k}{k!},\quad k=0,1,2,\ldots$$

Средно и дисперсия:

$$\mathrm{E}(X)=\lambda,\qquad \mathrm{Var}(X)=\lambda.$$

`dpois(`k`, `λ`)` $=\mathbf{P}(X=k)=e^{-\lambda}\,\dfrac{\lambda^k}{k!}$.

`ppois(`k`, `λ`)` $=\mathbf{P}(X\le k)$.

`rpois(`N`, `λ`)` генерира N случайни числа от Поасоново разпределение с параметър λ.

Поасоновото разпределение може да се използва като апроксимация на биномното за големи стойности на n и малки стойности на p, $np=\lambda$, т.е.

$$\binom{n}{k}p^k q^{n-k}\;\approx\; e^{-\lambda}\,\frac{\lambda^k}{k!}.$$

Разпределение	Функция в R	$\mathbf{P}(X=k)$	k
$\mathrm{Bi}(n,p)$	`dbinom(`k`, `n`, `p`)`	$\binom{n}{k}p^k q^{n-k}$	$0,1,2,\ldots,n$
$\mathrm{Ge}(p)$	`dgeom(`$k-1$`, `p`)`	$q^{k-1}p$	$1,2,3,\ldots$
$\mathrm{Ge}^*(p)$	`dgeom(`k`, `p`)`	$q^k p$	$0,1,2,\ldots$
$\mathrm{NB}(r,p)$	`dnbinom(`$k-r$`, `r`, `p`)`	$\binom{k-1}{r-1}p^r q^{k-r}$	$r,\ r+1,\ r+2,\ldots$
$\mathrm{NB}^*(r,p)$	`dnbinom(`k`, `r`, `p`)`	$\binom{r+k-1}{r-1}p^r q^k$	$0,1,2,\ldots$
$\mathrm{HG}(N,M,n)$	`dhyper(`k`, `M`, `$N-M$`, `n`)`	$\dfrac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}}$	$0,1,2,\ldots,n$
$\mathrm{Po}(\lambda)$	`dpois(`k`, `λ`)`	$e^{-\lambda}\,\dfrac{\lambda^k}{k!}$	$0,1,2,\ldots$

3.2. Непрекъснати сл.в.

Случайната величина X наричаме *непрекъсната*, ако съществува неотрицателна функция $f(x)$, такава че

$$\mathbf{P}(a \le X \le b) = \int_a^b f(x)dx.$$

Функцията $f(x)$ наричаме *плътност* на случайната величина X. С други думи, вероятността да наблюдаваме стойност в интервала $[a,b]$ е равна на интеграл от плътността в граници от a до b.

Непрекъснатата сл.в. може да приема произволни реални стойности от даден интервал.

Математическо очакване (средно) на непрекъснатата сл.в. X наричаме числото

$$\mathrm{E}(X) = \mu = \int_{-\infty}^{\infty} x\, f(x)dx.$$

Дисперсия на непрекъснатата сл.в. X наричаме числото

$$\mathrm{Var}(X) = \sigma^2 = \mathrm{E}(X-\mu)^2 = \mathrm{E}(X^2) - \mu^2 \quad = \int_{-\infty}^{\infty} x^2\, f(x)dx - \mu^2.$$

Стандартно отклонение на сл.в. X наричаме числото $\sigma = \sqrt{\mathrm{Var}(X)}$.

Функцията $F(q) = \mathbf{P}(X \le q)$ се нарича *функция на разпределение* на сл.в. X.

Обратната функция $Q(p) = F^{-1}(p)$ се нарича *квантилна функция* на сл.в. X. Ако $F(q)$ е строго растяща, $Q(p) = q \iff \mathbf{P}(Y \le q) = p$. В общия случай квантилната функция се дефинира така: $Q(p) = \inf\{q :\ F(q) \ge p\}, \quad p \in (0,1)$.

3.2.1. Равномерно разпределение

Случайната величина X наричаме равномерно разпределена в интервала (a, b) и означаваме $X \sim \mathrm{U}(a, b)$ ако нейната плътност има вида:

$$f(x) = \begin{cases} \dfrac{1}{b-a}, & x \in (a, b), \\ \\ 0, & x \notin (a, b). \end{cases}$$

Средно и дисперсия:

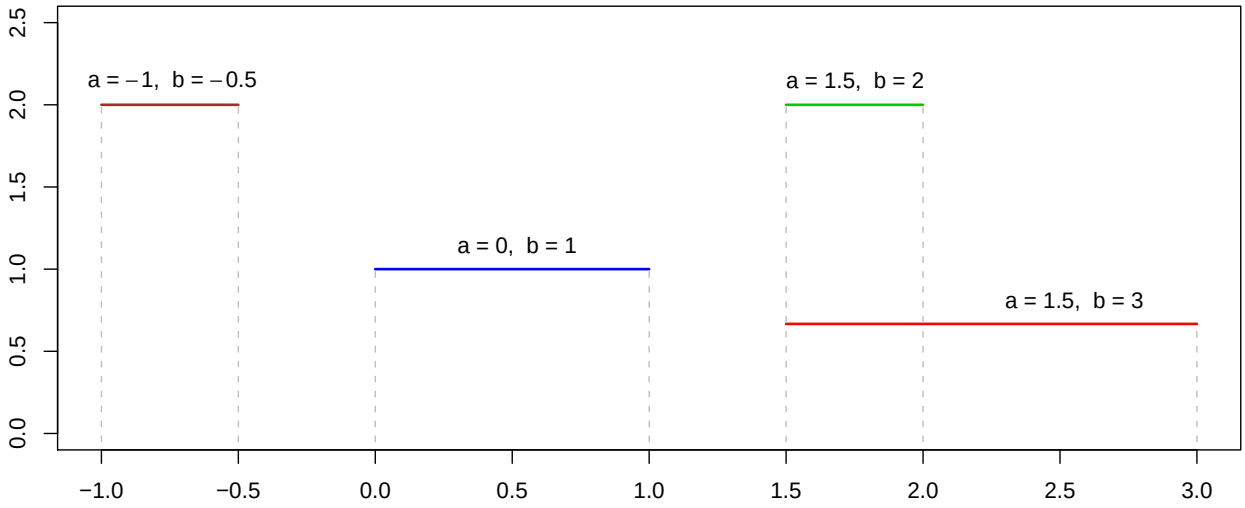
$$\mathrm{E}(X) = \frac{a+b}{2}, \qquad \mathrm{Var}(X) = \frac{(b-a)^2}{12}.$$

`dunif(`x`, `a`, `b`)` $= f(x)$.

`punif(`q`, `a`, `b`)` $= \mathbf{P}(X \leq q) = F(q)$.

`qunif(`p`, `a`, `b`)` $= Q(p) = F^{-1}(p)$.

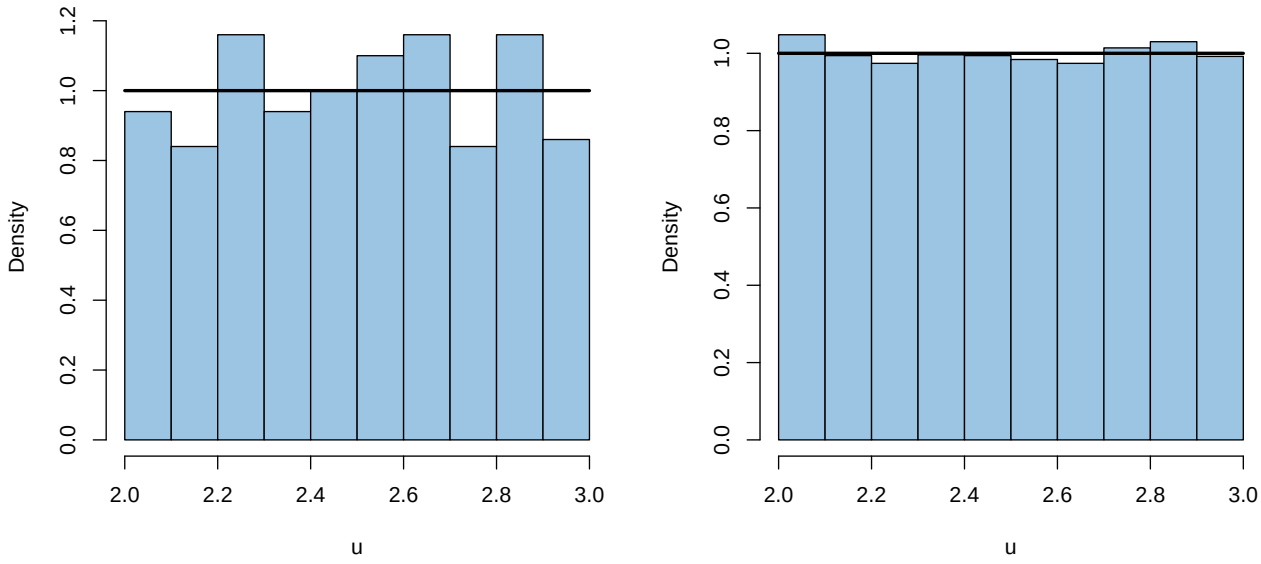
`runif(`N`, `a`, `b`)` генерира N случайни числа от равномерно разпределение в интервала (a, b).



Фигура 3.1. Равномерно разпределение: графики на плътността при различни стойности на параметрите.

Задача 3.1. Генерирайте 500 случайни числа от равномерно разпределение в интервала $(2, 3)$. Начертайте хистограма на генерираните числа и на същата картинка добавете графика на плътността $f(x)$. Повторете същото с 5000 случайни числа.

```
> u <- runif(500, 2, 3)
> hist( u, probability=T )
> curve( dunif(x, 2, 3), from=2, to=3, add=T, lwd=2.5 )
```



Фигура 3.2.

3.2.2. Експоненциално разпределение

Случайната величина X наричаме експоненциално разпределена с параметър $\lambda > 0$ и означаваме $X \sim \mathrm{Exp}(\lambda)$ ако нейната плътност има вида:

$$f(x) = \begin{cases} \lambda\, e^{-\lambda x}, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

Средно и дисперсия:

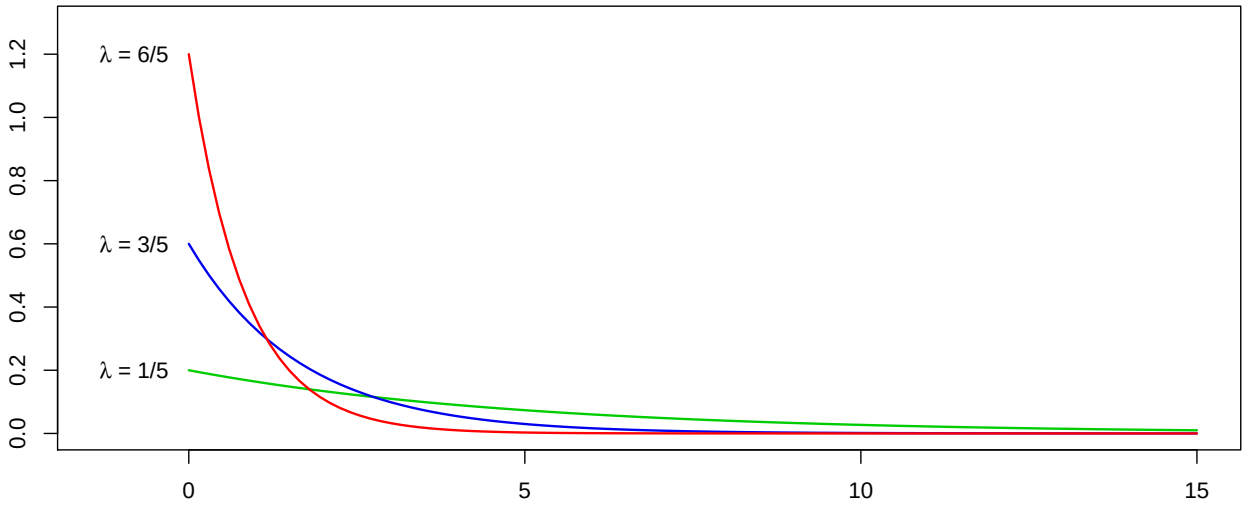
$$\mathrm{E}(X) = \frac{1}{\lambda}, \qquad \mathrm{Var}(X) = \frac{1}{\lambda^2}.$$

`dexp(`x`, `λ`)` $= f(x)$.

`pexp(`q`, `λ`)` $= \mathbf{P}(X \leq q) = F(q)$.

`qexp(`p`, `λ`)` $= Q(p) = F^{-1}(p)$.

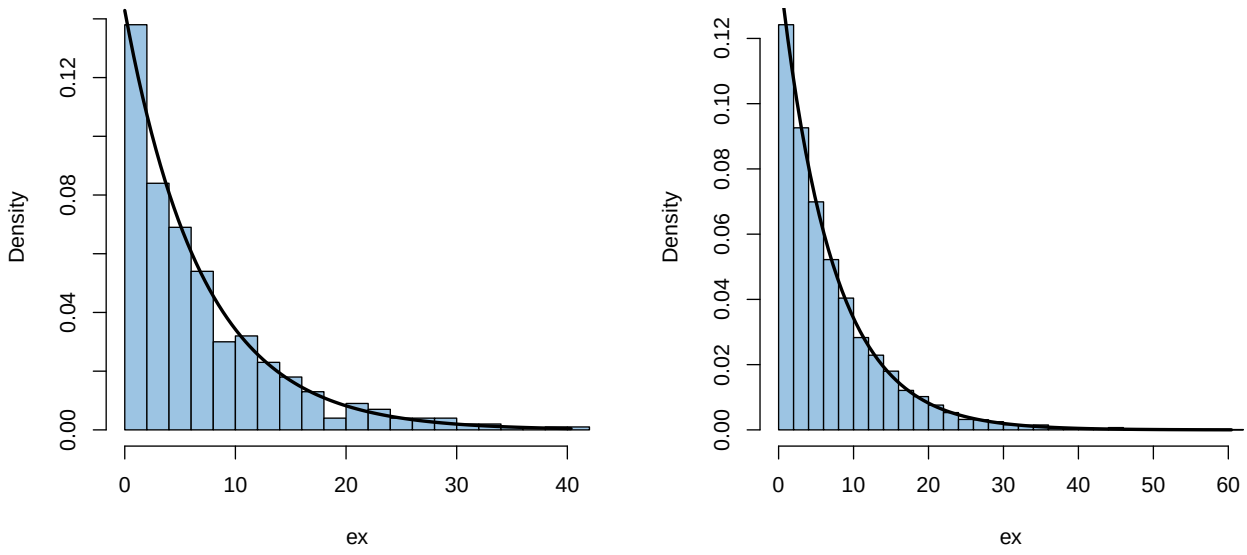
`rexp(`N`, `λ`)` генерира N случайни числа от експоненциално разпределение с параметър λ.



Фигура 3.3. Експоненциално разпределение: графики на плътността при стойности на параметъра $\lambda = 6/5,\ 3/5,\ 1/5$.

Задача 3.2. Генерирайте 500 случайни числа от експоненциално разпределение с параметър $\lambda = 1/7$. Начертайте хистограма на генерираните числа и на същата картинка добавете графика на плътността $f(x)$. Повторете същото с 5000 случайни числа.

```
> ex <- rexp(500, rate=1/7)
> hist( ex, probability=T )
> curve( dexp(x, rate=1/7), from=0, to=max(ex), add=T, lwd=2.5 )
```



Фигура 3.4.

3.2.3. Нормално разпределение

Случайната величина X наричаме нормално разпределна с параметри μ, σ^2 и означаваме $X \sim \mathcal{N}(\mu, \sigma^2)$ ако нейната плътност има вида:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-(x-\mu)^2/2\sigma^2}.$$

Средно и дисперсия:

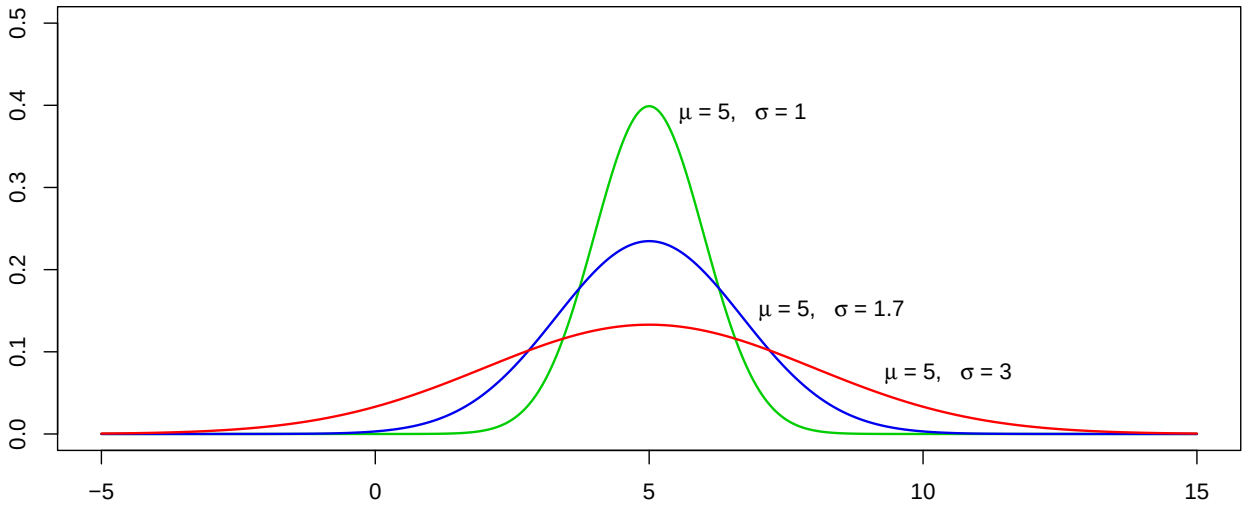
$$\mathrm{E}(X) = \mu, \qquad \mathrm{Var}(X) = \sigma^2.$$

`dnorm(`x`, `μ`, `σ`)` $= f(x)$.

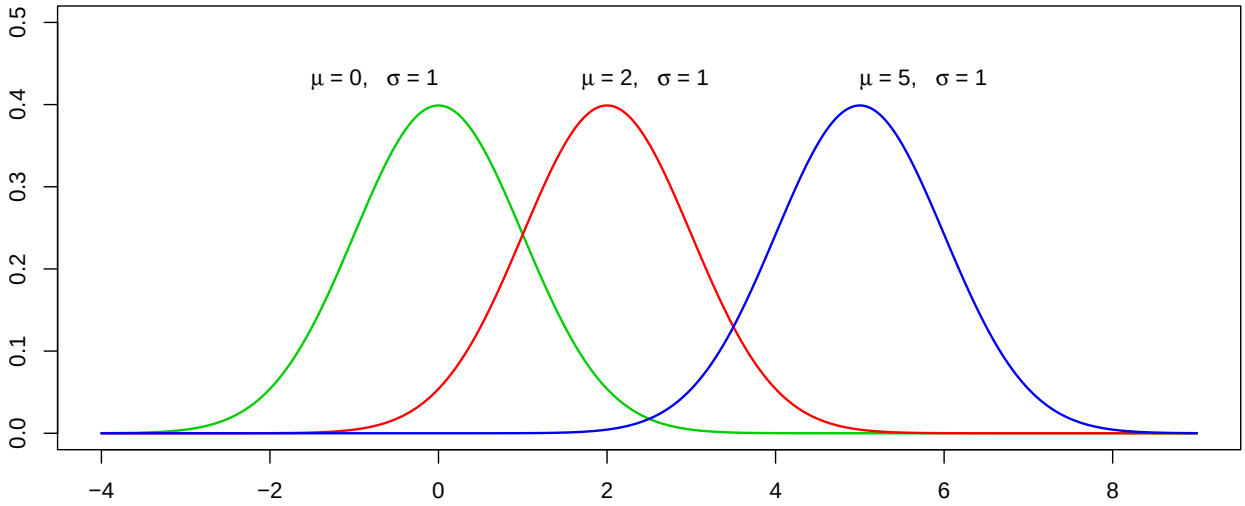
`pnorm(`q`, `μ`, `σ`)` $= \mathbf{P}(X \le q) = F(q)$.

`qnorm(`p`, `μ`, `σ`)` $= Q(p) = F^{-1}(p)$.

`rnorm(`N`, `μ`, `σ`)` генерира N случайни числа от нормално разпределение с параметри μ, σ.



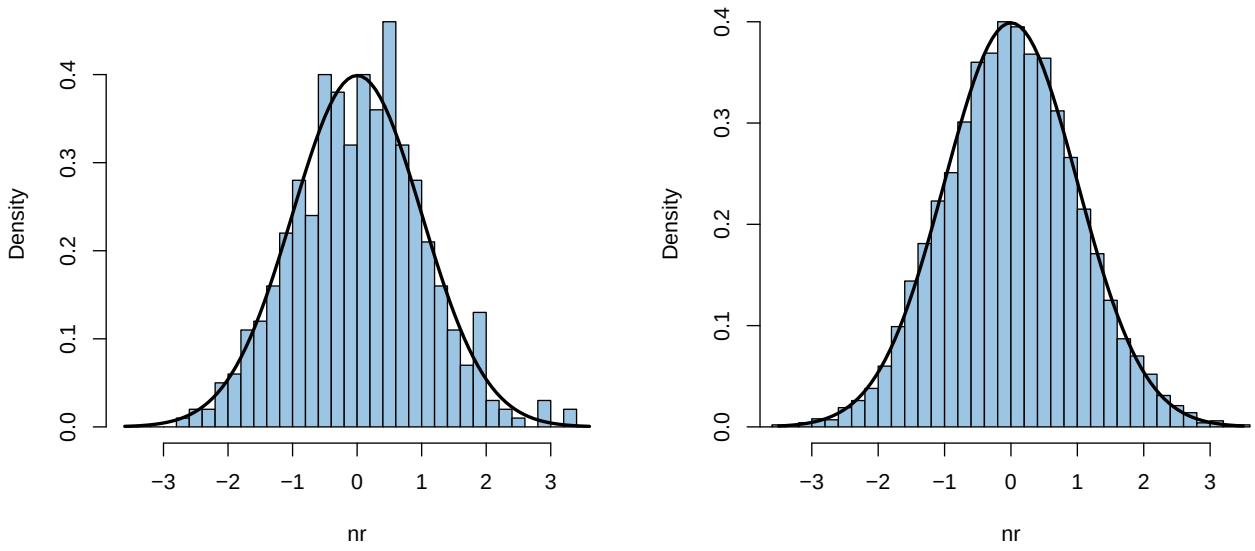
Фигура 3.5. Нормално разпределение: графики на плътността при $\mu = 5$ и няколко стойности на σ.



Фигура 3.6. Нормално разпределение: графики на плътността при няколко стойности на μ и $\sigma = 1$.

Задача 3.3. Генерирайте 500 случайни числа от нормално разпределение с параметри $\mu = 0$, $\sigma = 1$. Начертайте хистограма на генерираните числа и на същата картинка добавете графика на плътността $f(x)$. Повторете същото с 5000 случайни числа.

```
> nr <- rnorm(500, 0, 1)
> hist( nr, probability=T, xlim=c(-3.5,3.5) )
> curve( dnorm(x, 0, 1), add=T, lwd=2.5 )
```



Фигура 3.7.

Разпределение	Функция в R	$f(x)$	x
$\mathrm{U}(a, b)$	`dunif(`x`, `a`, `b`)`	$\dfrac{1}{b-a}$	(a, b)
$\mathrm{Exp}(\lambda)$	`dexp(`x`, `λ`)`	$\lambda\, e^{-\lambda x}$	$[0, \infty)$
$\mathcal{N}(\mu,\, \sigma^2)$	`dnorm(`x`, `μ`, `σ`)`	$\dfrac{1}{\sigma\sqrt{2\pi}}\, e^{-(x-\mu)^2/2\sigma^2}$	$(-\infty, \infty)$

4. Данни. Таблици и графики

"Data! Data! Data!" he cried impatiently.
"I can't make bricks without clay."
Sherlock Holmes
(Adventures of the Copper Breeches by A.C. Doyle)

Нека X е някаква променлива, от която се интересуваме, например пулса на човек при определена ситуация, времето на безотказна работа на машина, броя пътници в метрото за един ден, съдържанието на калий в един портокал и т.н. Записването на стойността на променливата X наричаме *наблюдение*. Обикновено, за да „изучим“ променливата X правим многократно наблюдения. Съвкупността от наблюдавани стойности на променливата X наричаме *данни* за X и ще означаваме: $x_1, x_2, \ldots, x_n$, като n е броя на наблюденията, които сме направили. Например, ако измерим съдържанието на калий в 30 портокала, ще имаме 30 наблюдения: $x_1, \ldots, x_{30}$ над променливата X = съдържание на калий в един портокал.

Множеството от всички възможни наблюдения, които можем да направим над една променлива наричаме *популация* или генерална съвкупност. Понякога популацията се отъждествява с множеството от обекти, които можем да наблюдаваме, например населението на България, персонала на дадена фирма, потребителите на даден продукт; тези популации имат краен брой елементи. Но ако правим лабораторен експеримент, който може да бъде повторен многократно, популацията е съвкупността от всички възможни експерименти и е безкрайна.

Тази част от популацията, която реално наблюдаваме, се нарича *извадка*. Например, ако не можем да наблюдаваме всички потребители на даден продукт, избираме по някакъв (случаен) начин част от тях и правим наблюдения само върху тази част. Когато правим лабораторен експеримент, го повтаряме само краен брой пъти и направените експерименти са нашата извадка. Извадката винаги има краен брой елементи.

Типове данни. Има два основни типа данни – числови (количествени) и категорни (неколичествени).

Числови данни – стойностите на наблюдаваната променлива са числа. Например брой пътници в метрото, температура на въздуха, пулс, брой продадени билети.

Категорни данни – стойностите на променливата (наричаме ги *категории* или *нива*) нямат никакви числови свойства. Например пол, цвят на очите, населено място, кръвна група, майчин език, марка телефон и т.н. Категориите на дадена променлива са взаимно изключващи се.

За удобство при обработка на данните, категориите обикновено се кодират с числа, например 0 = здрав, 1 = болен или 1 = кафяв, 2 = черен, 3 = син, 4 = зелен. Естествено, тези кодове са условни, нямат количествен смисъл.

След като са събрани, данните трябва да се представят в някакъв обобщен вид, за да се добие представа за основните им характеристики. Ще разгледаме често използваните таблични и графични техники за представяне на данни.

4.1. Категорни данни

Попитали сме 20 души кой интернет браузър използват най-често (допитването е направено през юли 2011). Записали сме отговорите във файла `browsers.txt`. Прочитаме данните в R по следния начин:

```
> dt <- read.table("browsers.txt")
```

Записваме ги във вектора `brows`:

```
> brows <- dt$V1
> brows
 [1] "IE"      "Firefox" "Firefox" "IE"      "IE"      "Chrome"  "Firefox"
 [8] "Firefox" "IE"      "Chrome"  "Firefox" "IE"      "Chrome"  "IE"
[15] "IE"      "Chrome"  "Firefox" "IE"      "Safari"  "Opera"

> class(brows)
[1] "character"
```

В случая имаме 20 наблюдения над категорна променлива с 5 категории: *Chrome, Firefox, IE, Opera, Safari*. Може да представим данните в таблица, показваща колко пъти се среща всяка от категориите:

```
> table(brows)
brows
 Chrome Firefox      IE   Opera  Safari
      4       6       8       1       1
```

Виждаме, че *Chrome* ползват 4 човека, *Firefox* ползват 6 човека и т.н.

Ако разделим броя срещания на общия брой наблюдения ($n = 20$) ще получим каква част (процент) от хората са използат съответния браузър:

```
> table(brows)/length(brows)
brows
 Chrome Firefox      IE   Opera  Safari
   0.20    0.30    0.40    0.05    0.05
```

От този таблица разбираме, че *Chrome* ползват 20% от запитаните, *Firefox* ползват 30% от запитаните и т.н.

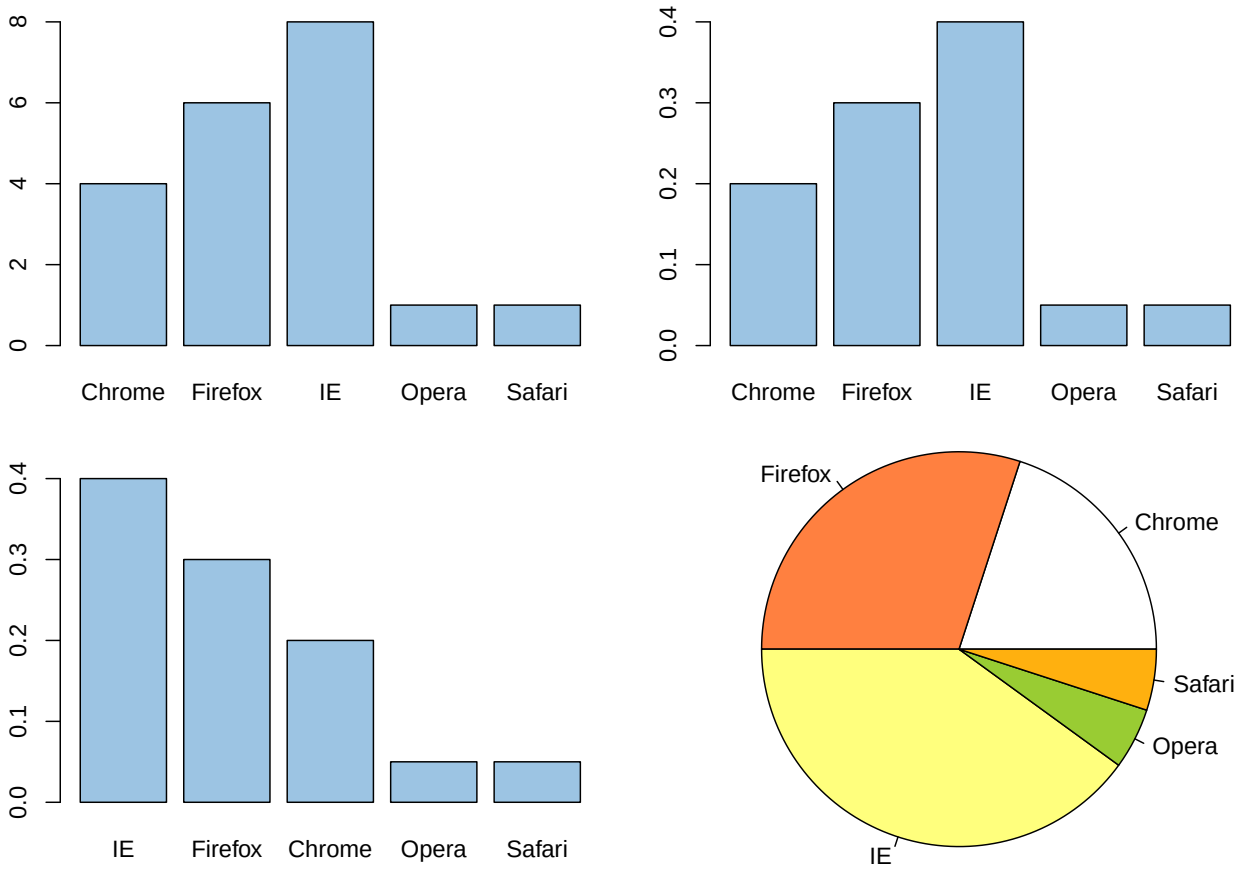
Можем да сортираме таблицата в намаляващ ред:

```
> sort( table(brows)/length(brows), decreasing=T )
brows
     IE Firefox  Chrome   Opera  Safari
   0.40    0.30    0.20    0.05    0.05
```

С помощта на функцията `barplot` представяме съответната таблица във вид на графика, в която всяка категория е представена със стълб с височина равна на съответната стойност от таблицата.

С функцията `pie` получаваме кръгова диаграма – всяка от категориите е представена като сектор от един кръг (с ъгъл, пропорционален на броя срещания).

```
> barplot( table(brows) )
> barplot( table(brows)/length(brows) )
> barplot( sort(table(brows)/length(brows), decreasing=T) )
> pie( table(brows) )
```



Фигура 4.1. Графично представяне на категорни данни

4.2. Числови данни

Иван си е отбелязвал времето (в минути) на чакане на автобуса всяка сутрин в продължение на 25 дни. Записваме данните във вектора `wait`:

```
> wait <- c(2,3,3,5,5,2,7,10,4,3,1,7,11,10,5,6,3,8,5,12,5,3,8,5,7)
```

Отново може да представим данните в таблица:

```
> table(wait)
wait
 1  2  3  4  5  6  7  8 10 11 12
 1  2  5  1  6  1  3  2  2  1  1
```

Таблицата показва, че една минута е чакал само веднъж, 2 минути – 2 пъти, 3 минути – 5 пъти и т.н.

Числовите данни обикновено могат да приемат много на брой стойности. Затова използването на подобна таблица не винаги е удачно. Вместо това, интервалът от възможни стойности се разбива на подинтервали (равни по дължина) и се прави таблица, показваща броя наблюдения във всеки подинтервал. В случая ще разделим интервала $(0, 12]$ на 6 подинтервала.

```
> wait.grp <- cut( wait, breaks=seq(0,12,2) )
> table(wait.grp)
```

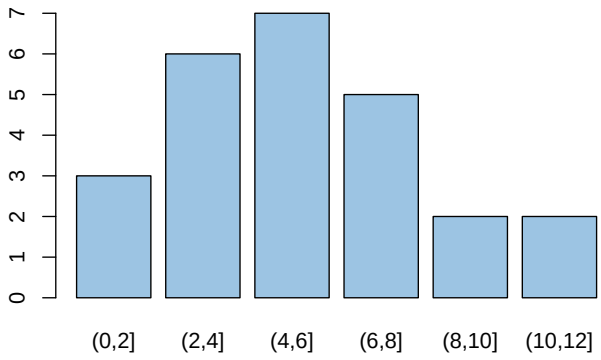
```
wait.grp
  (0,2]   (2,4]   (4,6]   (6,8]  (8,10] (10,12]
      3       6       7       5       2       2

> table(wait.grp)/length(wait)
wait.grp
  (0,2]   (2,4]   (4,6]   (6,8]  (8,10] (10,12]
   0.12    0.24    0.28    0.20    0.08    0.08
```

От първата таблица разбираме, че две или по-малко минути е чакал 3 пъти, от 2 до 4 минути е чакал 6 пъти и т.н. Втората таблица е с проценти: от 2 до 4 минути е чакал в 24% от дните, най-често е чакал между 4 и 6 минути – в 28% от дните.

Като използваме функцията `barplot` може да представим получената таблица във вид на графика, в която всеки подинтервал е представен със стълб с височина равна на броя наблюдения в подинтервала (фиг. 4.2).

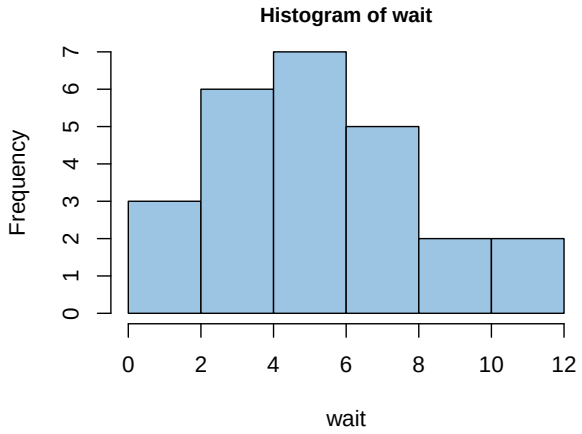
```
> barplot( table(wait.grp) )
```



Фигура 4.2.

Подобна графика получаваме и с фунцията `hist` – прилагаме я за вектора `wait`; тя разделя интервала на подинтервали и за всеки подинтервал рисува стълб с височина равна на броя наблюдения в подинтервала (фиг. 4.3). Такава графика се нарича *хистограма*.

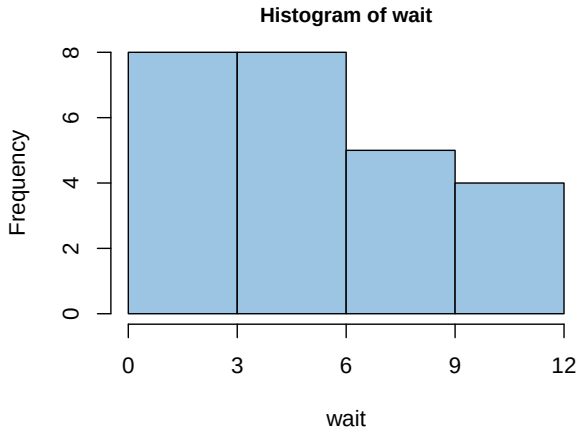
```
> hist(wait)
```



Фигура 4.3.

Фунцията `hist` сама определя какви да са подинтервалите. Но ако искаме може да зададем какви да бъдат, например може да разделим интервала $(0, 12]$ на 4 подинтервала:

```
> hist(wait, breaks=seq(0,12,3))
```



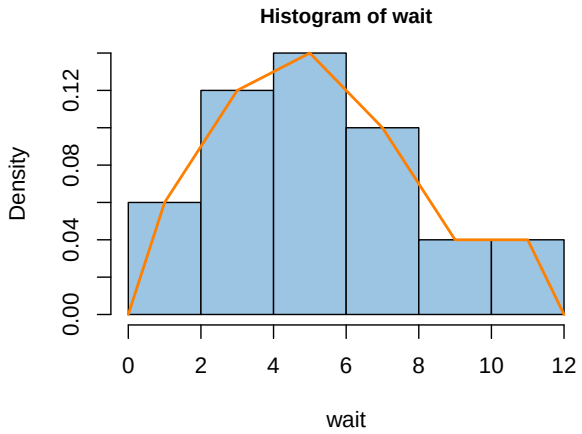
Фигура 4.4.

⟨!⟩ По подразбиране `hist` разделя интервала от стойности $[a, b]$ по следния начин: $[a, c_1]$ $(c_2, c_3]$ $(c_4, c_5]$ $\ldots (c_n, b]$, т.е. първият подинтервал е от вида [,], а останалите (,].

Командата `hist(..., probability=T)` чертае хистограма, така че сумата от лицата на всички стълбове (правоъгълници) е единица. Лицето на даден стълб е равно на честотата на данните в съответния интервал. Различава се от хистограмата `hist(...)` само по скалата на оста y.

Оранжевата линия на Фиг. 4.5 се нарича честотен полигон.

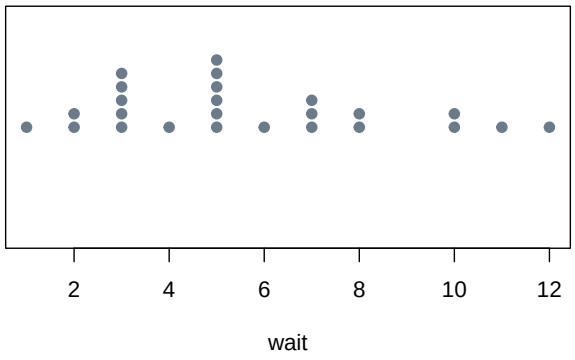
```
h <- hist(wait, probability=T)
lines( x=c( min(h$breaks), h$mids, max(h$breaks) ),
       y=c( 0, h$density, 0 ), type="l", lwd=2, col="darkorange1" )
```



Фигура 4.5.

Друг начин за графично представяне на числови данни е показан на Фиг. 4.6. На графиката всяко наблюдение е представено с кръгче.

```
stripchart(wait, method="stack", pch=20, cex=1.5)
```



Фигура 4.6.

Числови данни могат да бъдат преставени и чрез диаграмата *„клон с листа“* (*stem-and-leaf plot*). Подходяща е при малък обем на извадката (малко наблюдения). В известен смисъл е вариант на хистограмата от времената, когато графичните възможности на компютрите са били по-ограничени.

Да се върнем на данните `wait`. При разделяне на интервала на $[0, 4]$ $(4, 9]$ $(9, 12]$ се получава:

```
> stem(wait)

  The decimal point is 1 digit(s) to the right of the |

  0 | 122333334
  0 | 555555677788
  1 | 0012
```

С параметъра `scale` може да увеличим броя на подинтервалите (например, ако зададем `scale=2` интервалите ще са повече, но не е ясно колко ще са). На следващата диаграма са $[0, 1]$ $(1, 3]$ $(3, 5]$ $\ldots$ $(9, 11]$ $(11, 13]$, всяко наблюдение в съответния интервал е представено с `0`:

```
> stem(wait, scale=2)

  The decimal point is at the |

   0 | 0
   2 | 0000000
   4 | 0000000
   6 | 0000
   8 | 00
  10 | 000
  12 | 0
```

На следващата картинка, интервалите са $(0, 1]$ $(1, 2]$ $(2, 3]$ $\ldots$ $(11, 12]$, отново всяко наблюдение в съответния интервал е представено с `0`, т.е. показва ни, че 1 се среща един път, 2 – два пъти, 3 – пет пъти, 4 – един път и т.н.

```
> stem(wait, scale=3)
```

```
The decimal point is at the |

 1 | 0
 2 | 00
 3 | 00000
 4 | 0
 5 | 000000
 6 | 0
 7 | 000
 8 | 00
 9 |
10 | 00
11 | 0
12 | 0
```

От трите получени картинки, при първата и третата данните могат да бъдат възстановени еднозначно, докато при втората (`scale=2`) не могат – например в интервала $[2, 3]$ всички наблюдения са означени по един начин и не знаем колко от тях са '2' и колко '3'.

5. Числови характеристики на данните

Nothing in life is to be feared,
it is only to be understood.
Marie Curie

Нека $x_1, x_2, \ldots, x_n$ са наблюдения над някаква числова променлива X. Ще означаваме най-малкото по големина наблюдение с $x_{(1)}$, следващото по големина с $x_{(2)}$, и т.н., най-голямото с $x_{(n)}$, т.е. $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(n)}$.

Средна стойност (средно) на данните $x_1, x_2, \ldots, x_n$ наричаме числото

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad = \frac{x_1 + x_2 + \ldots + x_n}{n}.$$

Нарича се още извадъчно средно.

Медиана на данните $x_1, x_2, \ldots, x_n$ наричаме числото

$$\widehat{Me} = \begin{cases} x_{(\frac{n+1}{2})}, & \text{при нечетно } n \\ \\ \dfrac{x_{(\frac{n}{2})} + x_{(\frac{n}{2}+1)}}{2}, & \text{при четно } n \end{cases}$$

Нарича се още извадъчна медиана. По-малки или равни на медианата са поне половината от данните и по-големи или равни са също поне половината от данните. Грубо казано, медианата разделя данните на две равни части.

p-квантил на данните $x_1, x_2, \ldots, x_n$ наричаме числото, от което са по-малки или равни поне $100p\%$ от данните и са по-големи или равни поне $100(1-p)\%$. Грубо казано, p-квантилът разделя данните на две части, съответно от $100p\%$ и останалите $100(1-p)\%$. Нарича се още извадъчен p-квантил.

0.5-квантилът е всъщност медианата, 0.25-квантилът се нарича *първи квартил* (Q_1), а 0.75-квантилът се нарича *трети квартил* (Q_3). Разликата между третия и първия квартил ($Q_3 - Q_1$) се нарича *интерквартилен размах* (IQR).

Стандартно отклонение на данните $x_1, x_2, \ldots, x_n$ наричаме числото

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2} \quad = \sqrt{\frac{(x_1 - \overline{x})^2 + (x_2 - \overline{x})^2 + \ldots + (x_n - \overline{x})^2}{n-1}}.$$

Нарича се още извадъчно стандартно отклонение.

Неравенство на Чебишев. Нека данните $x_1, x_2, \ldots, x_n$ имат средна стойност $\overline{x}$ и стандартно отклонение s. За всяко $k > 0$ е вярно, че поне $100(1 - 1/k^2)$ процента от данните лежат в интервала $[\overline{x} - ks,\ \overline{x} + ks]$. При $k = 3$ получаваме, че поне 88.9% от данните са в интервала $[\overline{x} - 3s,\ \overline{x} + 3s]$.

Средната стойност и медианата показват центъра на данните в някакъв смисъл. Медианата е център на данните в смисъл, че е по средата на сортираните данни, т.е. приблизително половината от данните са по-малки и приблизително половината са по-големи от нея.

За да изясним в какъв смисъл средната стойност е център на данните, ще покажем, че $\sum_{i=1}^{n}(x_i - \overline{x}) = 0$. Наистина

$$\begin{aligned}\sum_{i=1}^{n}(x_i - \overline{x}) &= \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \overline{x} \\ &= n\,\frac{1}{n}\sum_{i=1}^{n} x_i - n\overline{x} \\ &= n\overline{x} - n\overline{x} = 0.\end{aligned}$$

Това тъждество означава, че ако сумираме на разликите на всяко наблюдение от средната стойност ще получим нула. С други думи, сумата на положителните разлики е равна на сумата на отрицателните разлики. В този смисъл средната стойност е център на данните – балансира сумата на положителните и сумата на отрицателните разлики.

Стандартното отклонение характеризира разпръскването на данните около средната стойност. То всъщност е корен от усреднения квадрат на разликата на всяко наблюдение от средната стойност (засега няма да изясняваме, защо усреднява се като се дели на $(n-1)$ на вместо на n). Друга мярка за разпръскването (разсейването) на данните е разликата между най-голямото и най-малкото наблюдение, $x_{(n)} - x_{(1)}$, нарича се *размах*.

Интервалът от неравенството на Чебишев ни дава представа доколко далече от средното може да се простират данните, ако са ни известни $\overline{x}$ и s.

Нека данните $x_1, x_2, \ldots, x_n$ са записани във вектора `x`. Горните числови характеристики се пресмятат в R със следните функции:

$\overline{x}$ = `mean(x)`

$\widehat{Me}$ = `median(x)`

p-квантил = `quantile(x, p)`

Q_1 = `quantile(x, 0.25)`

Q_3 = `quantile(x, 0.75)`

IQR = `IQR(x)`

s = `sd(x)`

Пример 1. Разглеждаме данните `airquality`. В променливата `airquality$Temp` има наблюдения за температурата (градуси по Фаренхайт) в Ню Йорк от май до септември 1973.

Пресмятаме медианата, средната стойност и стандартното отклонение:

```
> temp <- airquality$Temp
> median(temp)
[1] 79
> mean(temp)
[1] 77.88235
> sd(temp)
[1] 9.46527
```

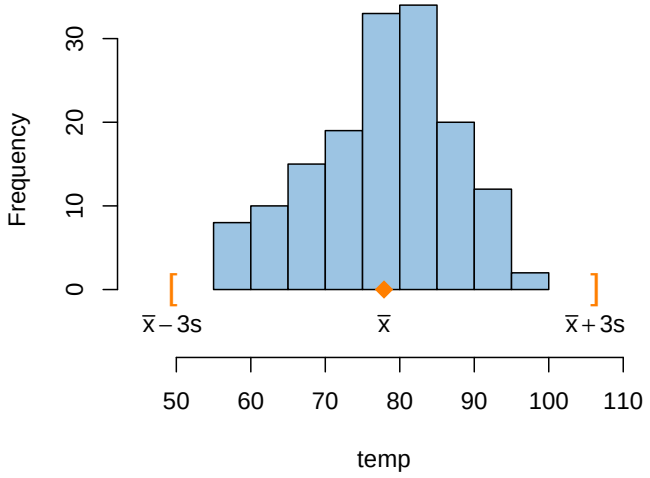
Функцията `summary` ни дава някои от разгледаните числови характеристики:

```
> summary(temp)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  56.00   72.00   79.00   77.88   85.00   97.00
```

Интервалът от неравенството на Чебишев е $[\overline{x} - 3s,\ \overline{x} + 3s] = [49.5,\ 106.3]$. Може да забележим, че всички данни са в този интервал.

```
> mean(temp) - 3*sd(temp)
[1] 49.48654
> mean(temp) + 3*sd(temp)
[1] 106.2782
```

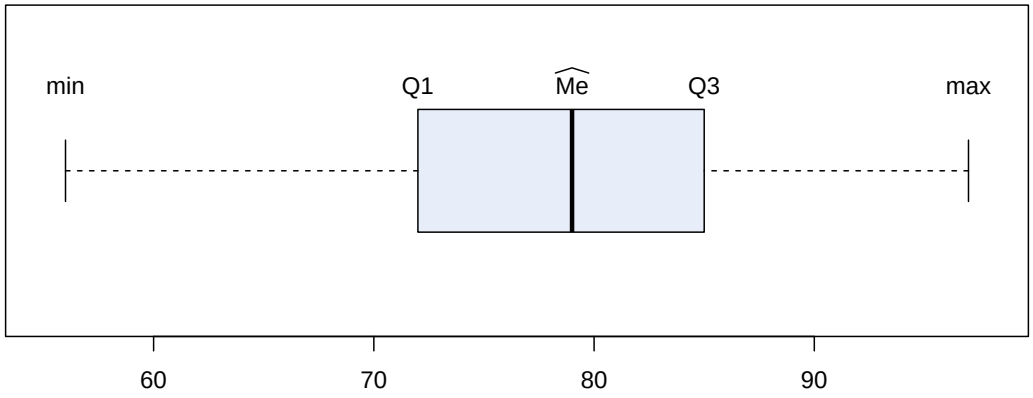
На следващата графика е показана хистограма на температурата и са нанесени средното $\overline{x}$ и интервалът от неравенството на Чебишев:



Фигура 5.1.

Първият квартил, медианата, третият квартил, както и най-малкото и най-голямото наблюдение се изобразяват на графика, наречена *кутия с мустаци*, с помощта на фунцията `boxplot`:

```
> boxplot(temp, horizontal=T)
```

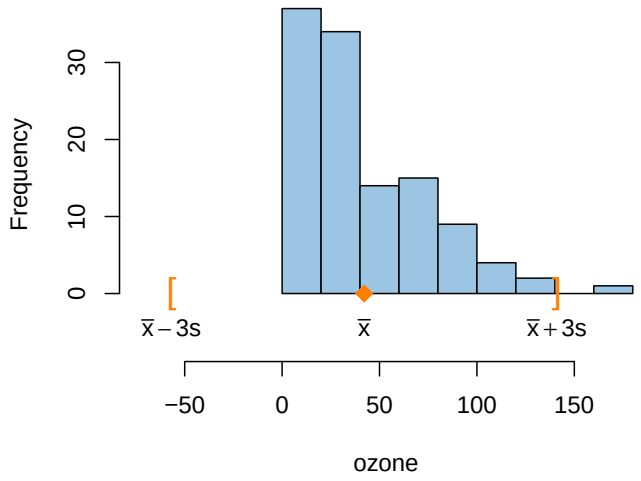


Фигура 5.2. Кутия с мустаци на temp

В данните `airquality` има и измервания на съдържанието на озон във въздуха (променливата `airquality$Ozone`).

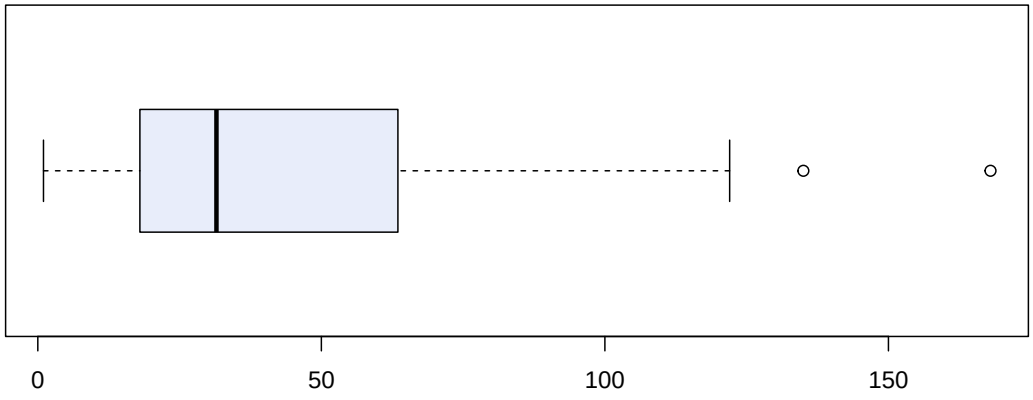
```
> ozone <- airquality$Ozone
> median(ozone, na.rm=T)
[1] 31.5
> mean(ozone, na.rm=T)
[1] 42.12931
> sd(ozone, na.rm=T)
[1] 32.98788
> summary(ozone)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   1.00   18.00   31.50   42.13   63.25  168.00      37
> mean(ozone, na.rm=T) - 3*sd(ozone, na.rm=T)
[1] -56.83434
> mean(ozone, na.rm=T) + 3*sd(ozone, na.rm=T)
[1] 141.093
```

На следващата графика е дадена хистограма на `ozone` и са нанесени средното $\overline{x}$ и интервалът от неравенството на Чебишев:



Фигура 5.3.

Кутия с мустаци на `ozone`:



Фигура 5.4.

Наблюденията, които не попадат в интервала $[Q_1 - 1.5 * IQR,\ Q_3 + 1.5 * IQR]$ се изобразяват с кръгче. В случая има две наблюдения извън този интервал. Такива наблюдения се считат за необичайни (*outliers*).

```
> quantile(ozone, 0.25, names=F, na.rm=T) - 1.5*IQR(ozone, na.rm=T)
[1] -49.875
> quantile(ozone, 0.75, names=F, na.rm=T) + 1.5*IQR(ozone, na.rm=T)
[1] 131.125
```

Задача 5.1. Да се намерят (на ръка и с R) средното, медианата, стандартното отклонение, първия и третия квартил на данните: 3, 6, 10, 0, 8, 3, 7, 2, 6.

Средното е:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{3+6+10+0+8+3+7+2+6}{9} = \frac{45}{9} = 5.$$

За да намерим s, от всяко наблюдение изваждаме $\overline{x}$ (третата колона) и повдигаме на квадрат (четвъртата колона):

i	x_i	$x_i - \overline{x}$	$(x_i - \overline{x})^2$
1	3	-2	4
2	6	1	1
3	10	5	25
4	0	-5	25
5	8	3	9
6	3	-2	4
7	7	2	4
8	2	-3	9
9	6	1	1
Σ	45	0	82

Стандартното отклонение е:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}} = \sqrt{\frac{82}{9-1}} = 3.20.$$

За да намерим медианата, подреждаме данните по големина:

$$0, 2, 3, 3, \mathbf{6}, 6, 7, 8, 10$$

В средата е числото 6, следователно $\widehat{Me} = 6$.

За да намерим Q_1 и Q_3 разделяме сортираните данни на две половини, като числото в средата се включва и в двете половини:

$$\underline{0, 2, 3, 3, \mathbf{6}}, 6, 7, 8, 10$$

$$0, 2, 3, 3, \underline{\mathbf{6}, 6, 7, 8, 10}$$

Медианата на първата половина е първия квартил, т.е. $Q_1 = 3$, а медианата на втората половина е третия квартил, т.е. $Q_3 = 7$.

6. Многомерни данни

The advanced reader who skips parts that appear to him too elementary may miss more than the less advanced reader who skips parts that appear to him too complex.
G. Polya

Когато наблюдаваме (измерваме) повече от една променлива, наричаме данните многомерни. Те обикновено се записват в таблица от вида:

	X	Y	Z	$\ldots$	W
1	x_1	y_1	z_1	$\ldots$	w_1
2	x_2	y_2	z_2	$\ldots$	w_2
3	x_3	y_3	z_3	$\ldots$	w_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
n	x_n	y_n	z_n	$\ldots$	w_n

където всеки ред отговаря на едно наблюдение (опит, измерване, участник в изследване и т.н.), а всяка колона отговаря на една променлива. Тук ще се запознаем с някои основни техники за боравене с многомерни данни.

Многомерните данни се представят в R чрез обект наречен *data frame*. Той е подобен на матрица, с тази разлика, че колоните могат да бъдат от различен тип.

Ще разгледаме данните `survey` от пакета `MASS`. Тези данни съдържат отговорите на няколко въпроса, зададени на 237 студенти от курса *Статистика I* в Университета на Аделаида (не е ясно кога е направена анкетата).

За да използваме обекти от даден пакет го зареждаме с командата `library(packageName)`.

```
> library(MASS)
> ?survey
> str(survey)
'data.frame':   237 obs. of  12 variables:
 $ Sex   : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 1 2 1 2 2 ...
 $ Wr.Hnd: num  18.5 19.5 18 18.8 20 18 17.7 17 20 18.5 ...
 $ NW.Hnd: num  18 20.5 13.3 18.9 20 17.7 17.7 17.3 19.5 18.5 ...
 $ W.Hnd : Factor w/ 2 levels "Left","Right": 2 1 2 2 2 2 2 2 2 2 ...
 $ Fold  : Factor w/ 3 levels "L on R","Neither",..: 3 3 1 3 2 1 1 3 3 3 ...
 $ Pulse : int  92 104 87 NA 35 64 83 74 72 90 ...
 $ Clap  : Factor w/ 3 levels "Left","Neither",..: 1 1 2 2 3 3 3 3 3 3 ...
 $ Exer  : Factor w/ 3 levels "Freq","None",..: 3 2 2 2 3 3 1 1 3 3 ...
 $ Smoke : Factor w/ 4 levels "Heavy","Never",..: 2 4 3 2 2 2 2 2 2 2 ...
 $ Height: num  173 178 NA 160 165 ...
 $ M.I   : Factor w/ 2 levels "Imperial","Metric": 2 1 NA 2 2 1 1 2 2 2 ...
 $ Age   : num  18.2 17.6 16.9 20.3 23.7 ...
```

⟨!⟩ По-нататък в тази глава, когато говорим за променлива, ще разбираме наблюденията над тази променлива (стойностите от съответната колона в таблицата по-горе).

`fix(survey)` – извежда данните като таблица;
`summary(survey)` – числови характеристики за всяка променлива;
`survey[ ,'Age']` – променливата `Age`;

`survey$Age` – променливата `Age` (друг начин);
`survey[ ,12]` – дванадесетата променлива;
`survey[5, ]` – петото наблюдение;

Може да се обръщаме към промелнивите от даден data frame директно (например `Age` вместо `survey$Age`), ако напишем `attach(dataFrameName)`.

```
> summary(Age)
Error in summary(Age) : object 'Age' not found
> summary(survey$Age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  16.75   17.67   18.58   20.37   20.17   73.00
> attach(survey)
> summary(Age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  16.75   17.67   18.58   20.37   20.17   73.00
```

Понякога има липсващи наблюдения за дадена променлива. Във R те се означават с `NA`. За да се пресметнат някои числови характеристики (например средно, медиана, стандартно отклонение) на променлива, в която има липсващи наблюдения, тези наблюдения трябва да се игнорират при пресмятането (например, при пресмятане на средно се пресмята средното на останалите). Това се задава с параметъра `na.rm=T`.

```
> mean(Age)
[1] 20.37451
> mean(Pulse)
[1] NA
> mean(Pulse, na.rm=T)
[1] 74.15104
> summary(Pulse)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  35.00   66.00   72.50   74.15   80.00  104.00   45.00
```

По подразбиране R показва 7 значещи цифри след десетичната точка. За следващите примери ще ги намалим на 3:

```
> options(digits=3)
```

Когато наблюдаваме няколко променливи, често се интересуваме от някакви връзки между тях. Да разгледаме променливите `Smoke` и `W.Hnd`. И двете са категорни. Във `Smoke` е записана честотата на пушене, има следните категории: *Heavy, Regul, Occas, Never*. Във `W.Hnd` е записано с коя ръка пише студентът (*Left, Right*). Може да представим двете променливи в двумерна таблица (крос-таблица):

```
> table(Smoke, W.Hnd)
       W.Hnd
Smoke   Left Right
  Heavy    1    10
  Never   13   175
  Occas    3    16
  Regul    1    16
```

От подобна таблица може да разберем, например, колко от студентите пишат с лявата ръка (*Left*) и не пушат (*Never*). В тази таблица са изключени липсващите наблюдения; те могат да бъдат показани с добавяне на `useNA="always"`.

```
> table(Smoke, W.Hnd, useNA="always")
       W.Hnd
Smoke   Left Right <NA>
  Heavy    1    10    0
  Never   13   175    1
  Occas    3    16    0
  Regul    1    16    0
  <NA>     0     1    0
```

Вместо броя срещания, двумерната таблица може да показва относителния дял (процент). Ще разгледаме три вида такива таблици. Ако разделим първата таблица на общия брой наблюдения, които са 235 (поради изключването на липсващите), ще получим таблицата:

```
> tab.smoke.hand  <- table(Smoke, W.Hnd)
> prop.table(tab.smoke.hand)
       W.Hnd
Smoke      Left   Right
  Heavy 0.00426 0.04255
  Never 0.05532 0.74468
  Occas 0.01277 0.06809
  Regul 0.00426 0.06809
```

От нея може да разберем, например, че 74.5% от студентите пишат с дясната ръка и не пушат.

Ако разделим всеки ред от първата таблица на сумата на реда, получаваме таблица с редови процент; за целта използваме командата `prop.table(tab.smoke.hand, 1)`. От тази таблица може да разберем, например, че 84.2% от пушещите „понякога“ (*Occas*) пишат с дясната ръка.

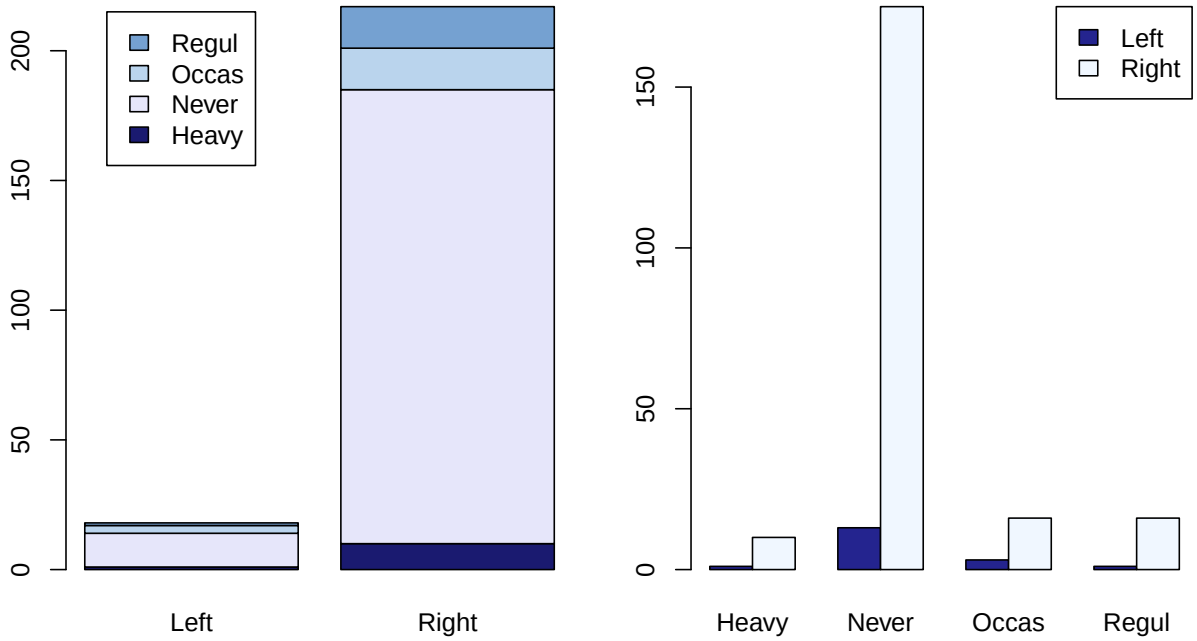
```
> prop.table(tab.smoke.hand, 1)
       W.Hnd
Smoke     Left  Right
  Heavy 0.0909 0.9091
  Never 0.0691 0.9309
  Occas 0.1579 0.8421
  Regul 0.0588 0.9412
```

Ако разделим всяка колона от първата таблица на сумата на колоната, получаваме таблица с колонен процент; за целта пишем `prop.table(tab.smoke.hand, 2)`. От тази таблица може да видим, например, че 7.37% от пишещите с дясната ръка пушат „понякога“ (*Occas*).

```
> prop.table(tab.smoke.hand, 2)
       W.Hnd
Smoke     Left  Right
  Heavy 0.0556 0.0461
  Never 0.7222 0.8065
  Occas 0.1667 0.0737
  Regul 0.0556 0.0737
```

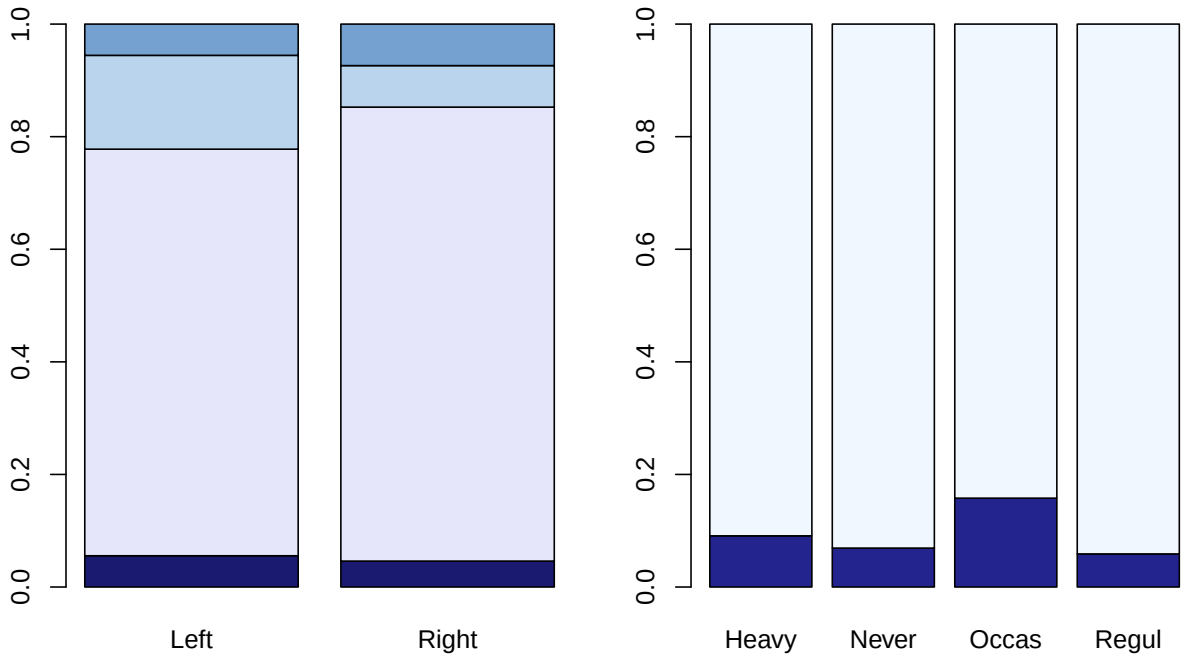
Като използваме функцията `barplot` може да представим горните таблици графично по различни начини:

```
barplot( table(Smoke, W.Hnd), legend=T,
         args.legend=list(x="topleft", inset=0.05) )
barplot( table(W.Hnd, Smoke), beside=T, legend=T,
         args.legend=list(x="topright", inset=0.05) )
```



Фигура 6.1.

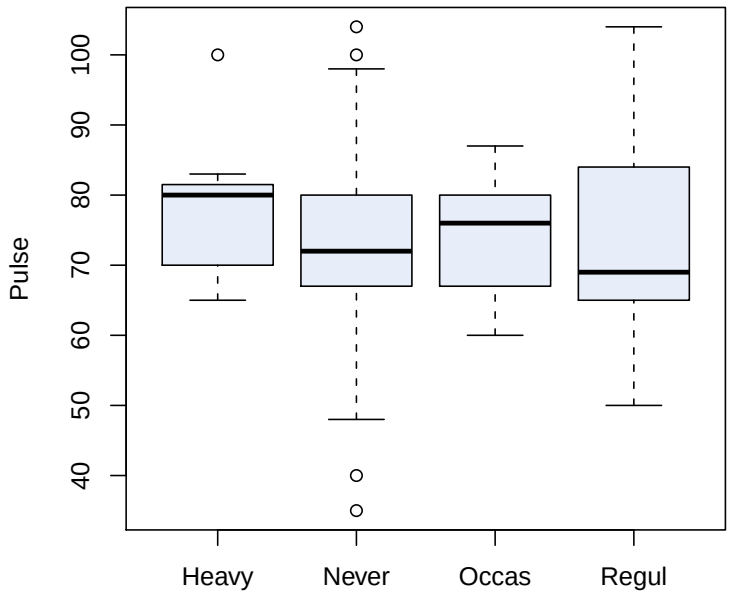
```
barplot( prop.table( tab.smoke.hand, 2 ) )
barplot( t( prop.table( tab.smoke.hand, 1 ) ) )
```



Фигура 6.2.

Може да се интересуваме доколко стойностите на дадена числова променлива се различават при различните нива (категории) на някаква категорна променлива. Да разгледаме числовата променливата `Pulse`, в която е записан пулсът на студента и категорната променлива `Smoke`. Със следната команда получаваме кутия с мустаци на `Pulse` за всяка от категориите на `Smoke` (променливата `Pulse` се разбива по категориите на `Smoke` и за всяка категория се рисува кутия с мустаци).

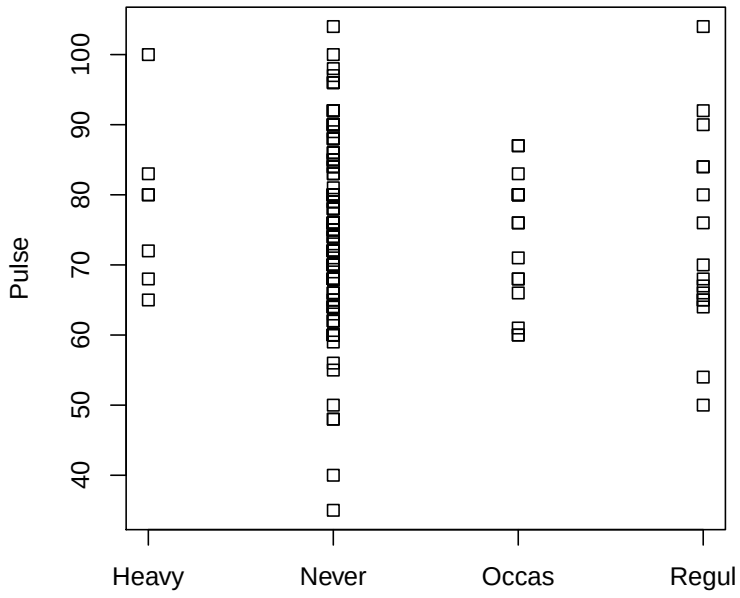
```
> boxplot(Pulse ~ Smoke)
```



Фигура 6.3.

Следната команда представя променливата `Pulse` разбита по категориите на `Smoke`, като всяко наблюдение е изобразено с квадратче:

```
> stripchart(Pulse ~ Smoke, vertical=T)
```



Фигура 6.4.

Нека във вектора `x` са записани наблюденията $x_1, \ldots, x_n$ над някаква числова променлива, а във вектора `y` – наблюденията $y_1, \ldots, y_n$ над друга числова променлива. Командата `plot(x,y)` изобразява точките $(x_1, y_1), \ldots, (x_n, y_n)$ в координатната система Oxy. Изобразяването на две променливи на такава графика, може да ни подскаже за някаква зависимост между тях. Ще изобразим по двойки някои от следните променливи:

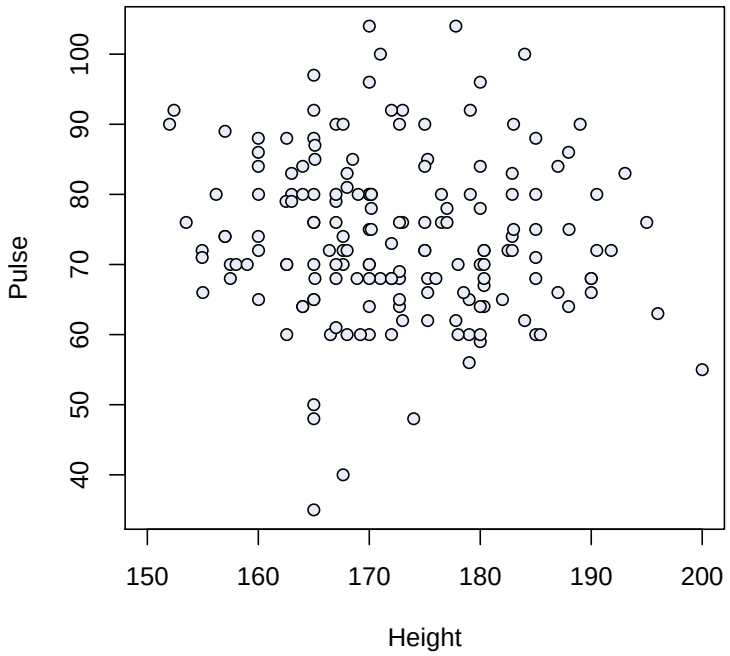
`Height` – височина на студента (в сантиметри);

`Pulse` – пулс на студента;

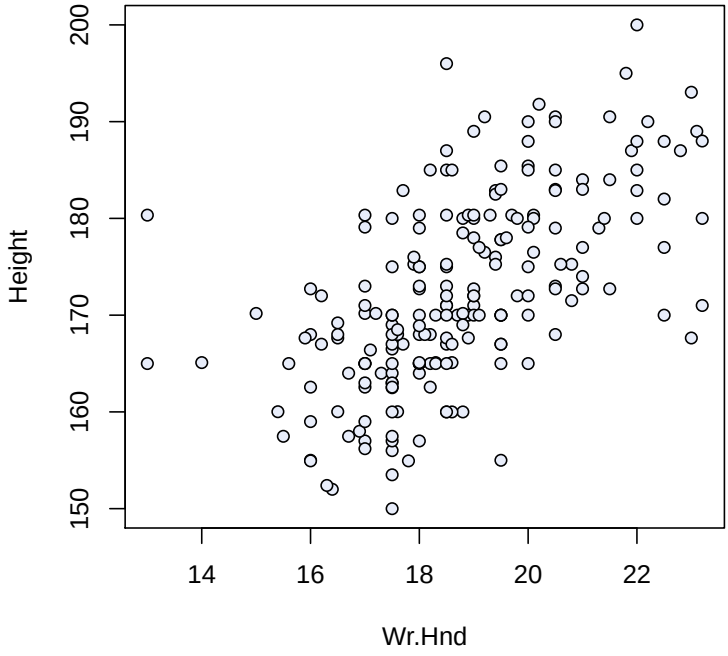
`Wr.Hnd` – дължина на педята на ръката, с която студентът пише (в сантиметри);

`NW.Hnd` – дължина на педята на ръката, с която студентът не пише (в сантиметри);

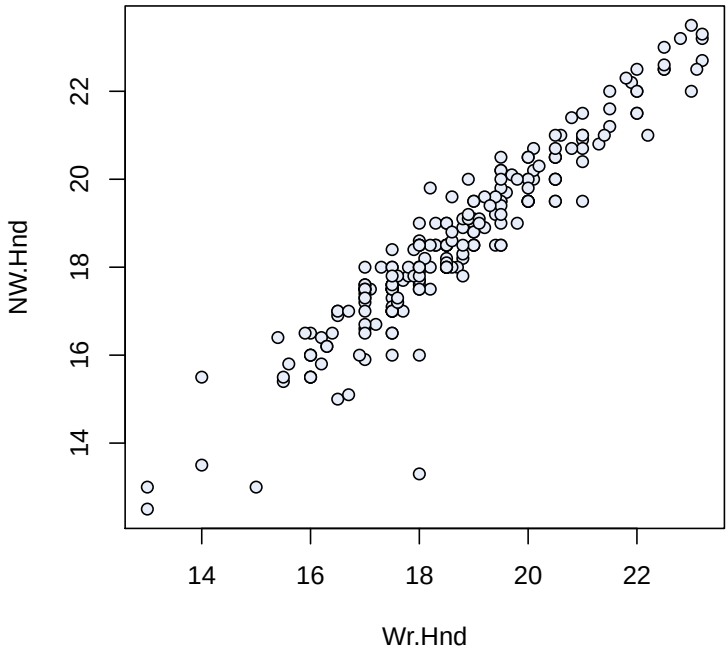
```
> plot(Height, Pulse)
> plot(Wr.Hnd, Height)
> plot(Wr.Hnd, NW.Hnd)
```



Фигура 6.5.



Фигура 6.6.



Фигура 6.7.

Нека $x_1, \ldots, x_n$ са наблюдения над някаква числова променлива X, а $y_1, \ldots, y_n$ са наблюдения над друга числова променлива Y. Числото

$$r = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2 \sum_i (y_i - \overline{y})^2}}$$

се нарича *извадъчна корелация* на променливите X и Y. То е мярка за линейната зависимост между тях. Пресмята се с `cor(x,y)`.

Ако точките $(x_1, y_1), \ldots, (x_n, y_n)$ са близо да права линия, то r е близо до 1 или -1 и казваме, че променливите са положително или отрицателно корелирани, съответно. Ако r е близо до 0, казваме, че променливите са некорелирани.

⟨!⟩ Ако точките $(x_1, y_1), \ldots, (x_n, y_n)$ са близо до прави от вида $x = \text{const}$ или $y = \text{const}$, извадъчната корелация r ще е близка до 0. (Защо?)

Ще намерим корелациите на двойките променливи, които изобразихме на графики по-горе. Тъй като има липсващи наблюдения, за да се игнорират при пресмятането, добавяме `use="complete.obs"`.

```
> cor(Height, Pulse, use="complete.obs")
[1] -0.0839
> cor(Wr.Hnd, Height, use="complete.obs")
[1] 0.601
> cor(Wr.Hnd, NW.Hnd, use="complete.obs")
[1] 0.948
```

Корелацията между `Height` и `Pulse` е близка до 0. Тази между `Wr.Hnd` и `Height` е по-близо до 1, а между `Wr.Hnd` и `NW.Hnd` е почти 1.