```r
# Вариация


team1 <- c(72, 73, 76, 76, 78)

team2 <- c(67, 72, 76, 76, 84)

basketball_teams <- data.frame(team1, team2)



# 3. Оценка на вариацията на разпределение



# Преди да започнем с изследването на разсейването, първо ще видим какво е очакването

colMeans(basketball_teams) # Взимаме средните на стойностите по колони

apply(basketball_teams, 2, mean) # Еквивалентно на горния ред



# Като цело се избягват заключенията само на база средните стойности или други оценки за

# локацията на разпределенията (ще наблегнем на това при оценките на хипотезите).

# И при двета отбора средните стойности са равни - 75. това не означава, че двата отбора

# са близки относно разпределението на височината им. В частност, височината при играчите на

# втория отбор варира много повече от тази на първия. В следващата секция ще разгледаме как

# можем да оженим вариацията.



# 3.1. Обхват (Range) - максималната стойност - минималната стойност


rangeFunction <- function(x) {

max(x) - min(x)

}
```

```
rangeFunction(basketball_teams$team1)

rangeFunction(basketball_teams$team2)

# както можем да видим от резултатите, при първия отбор имаме разлика от 6 инча, докато при втория - 17




# 3.2. Вариация (дисперсия) и стандартно отклонение




# За разлика от обхвата, стандартното отклонение взема под внимамнеи всички наблюдения.

# Стандартното отклонение е оценка на вариацията, която показва колко далече са наблюденията от очакването



# Стандартното отклонение е предпочитана оценка за вариацията, когато средното се използва за оценка

# на локацията (центъра) на разпределението.



# Вариацията (дисперсията) се изчислява по формулата по-долу:

variationFunction <- function(x) {

x_mean <- sum(x)/length(x)

x_minus_xMean <- x - x_mean

x_minus_xMean_2 <- x_minus_xMean^2


sum(x_minus_xMean_2) / (length(x) - 1)

}




# Вариацията има функция в базовия пакет на R

variationFunction(basketball_teams$team1)

var(basketball_teams$team1)

var(basketball_teams$team2)
```

```
# При тестването на хипотези и при определянето на доверителните интервали се използва

# стандардартното отклонение. Стандартното отклонение е производно на вариацяита и представлява

# корен квадратен от дисперсията.


sqrt(variationFunction(basketball_teams$team1))

sd(basketball_teams$team1)

sd(basketball_teams$team2)


# Правилото на Чебишев, което е валидно за всички множества, ни казва, че 89% от наблюденията

# лежат в интервала (X_mean - 3*X_std; X_mean + 3*X_std), където X_mean - средната стойност и

# X_std - стандартното отклонение.

# При камбановидна форма на разпределението, този процент достига до 99.7


N <- 10^4

set.seed(94171)

dist1 <- rnorm(N)

dist2 <- rgamma(N, 3)


par(mfrow = c(1, 2))

plot(density(dist1), lwd = 2, main = "Плътност", xlab = "Нормално разпределение",

ylab = "Плътност", col = "lightblue", xlim = range(dist1))

mu <- mean(dist1); sigma <- sd(dist1)

abline(v = c(mu - 3*sigma, mu, mu + 3*sigma), lwd = 2, col = c("black", "red", "black"))


plot(density(dist2), lwd = 2, main = "Плътност", xlab = "Гама разпределение",

ylab = "Плътност", col = "lightblue", xlim = range(dist2))

mu <- mean(dist2); sigma <- sd(dist2)
```

```
abline(v = mu + c(-3, 0, 3)*sigma, lwd = 2, col = c("black", "red", "black"))

par(mfrow = c(1, 1))


mu <- mean(dist2); sigma <- sd(dist2)

round(sum(dist2 >= mu - 3*sigma & dist2 <= mu + 3*sigma)*100/N, 2)



# 3.3. The five number summary

# тази статтистика най-често показва минималната стойност, 1-ви квартил, медиана (2-ри квартил),

# 3-ти квартил и максималната стойност

# В R използваме фунцкциите summary() и fivenum()



summary(dist1)

fivenum(dist1)



par(mfrow = c(1, 2))

plot(density(dist1), lwd = 2, main = "Плътност", xlab = "Нормално разпределение",

ylab = "Плътност", col = "lightblue", xlim = range(dist1))

abline(v = fivenum(dist1), lwd = c(1.5, rep(2, 3), 1.5), col = c("black", "red", "red", "red", "black"),

lty = c(1, rep(3, 3), 1))


plot(density(dist2), lwd = 2, main = "Плътност", xlab = "Гама разпределение",

ylab = "Плътност", col = "lightblue", xlim = range(dist2))

abline(v = fivenum(dist2), lwd = c(1.5, rep(2, 3), 1.5), col = c("black", "red", "red", "red", "black"),

lty = c(1, rep(3, 3), 1))
```

```
par(mfrow = c(1, 1))




# 3.4. Interquartile Range и MAD

# Тези два вида оценки на дисперсията е препоръчително да се използват, когато за оценка на центъра

# на разпределението се използва медианата. И двете оценки се водят "стабилни" към екстремалните стойности




# Nielsen Company е публикувала информация колко часа седмично американците прекарват пред телевизора.

# Това е извадка от 20 човека

tv_viewing_times <- c(25, 41, 27, 32, 43, 66, 35, 31, 15, 5, 34, 26, 32, 38, 16, 30, 38, 30, 20, 21)



# За да покажем как екстремалните стойности влияят върху част от оценките ще добавим голяма стойност,

# например 240 часа

tv_viewing_times_new <- c(tv_viewing_times, 240)





# Интерквартилния обхват се изчислява като разлика между 3-ти квартил и 1-ви квартил

summary(tv_viewing_times)

summary(tv_viewing_times)[c(2, 5)]

diff(summary(tv_viewing_times)[c(2, 5)])



# Базовата функция в R се казва IQR()

IQR(tv_viewing_times)





IQR(tv_viewing_times)
```

```
IQR(tv_viewing_times_new)

# Както се вижда няма кой знае колко голяма промяна след добаяването на екстремалната стойност



# Какво обаче би станало, ако използваме стандартното отклонение?

sd(tv_viewing_times)

sd(tv_viewing_times_new)

# Разликата скача в пъти



# Ето защо при наличието на екстремуми е по-разумно да използваме медианата за оценка на центъра и

# IQR или mad за оценка на дисперсията



# MAD

# Оценката MAD представлява медианата на вектора с абсолютните стойности от разлики от стойността и

# медианата на самия вектор. Резултатът е умножен по 1.4826

# Формулата е записана по-долу

X_median <- median(tv_viewing_times_new)

X_median

X_diff <- abs(tv_viewing_times_new - X_median)

X_diff

median(X_diff)*1.4826



mad(tv_viewing_times_new)





# - Добре, при оценката на вариацията имаме значима промяна. Как ли стоят нещата с оценките за центъра?

# - Екстремумите оказват влияние и при оценката за центъра. Ето защо, при наличие на такива стойности,

# предпочитаме да използваме медианата, вместо средната стойност.



mean(tv_viewing_times)
```

```
mean(tv_viewing_times_new)

median(tv_viewing_times)

median(tv_viewing_times_new)

# Разликата е очевидна

# - Между другото имаме и други опции при наличието на екстремални стойности - bootstrap метод и

# trimmed mean. За съжаление, няма да можем да се запознаем в курса с bootstrap, но ви го препоръчвам.

# Какво прави trim опцията? Тя премахва по част от най-големите и най-малките стойности.

# В нашия случай, ние сме посочили, че искаме да махмен 5% от най-големите и най-малките стойности.

# Тоест ще вземем 5/2 = 2.5% от най-малките стойности и 2.5 от най-големите.

mean(tv_viewing_times, trim = 0.05)

mean(tv_viewing_times_new, trim = 0.05)

# Както виждаме стойностите са близки

# 4. Графично представяне на разпределение

# 4.1. Barplot

# Използваме barplot, когато искаме да представим честотното разпределение на категорийни променливи

set.seed(4012)

fruits <- sample(x = c("Apple", "Banana", "Blackberry", "Peach"), size = 40, replace = T,

prob = c(0.4, 0.1, 0.3, 0.2))
```

```
tt <- table(fruits); tt

barplot(height = tt, col = "seagreen3", main = "Barplot")

?barplot

# height - приема вектор или матрица с числови стойности като вход. Стойностите могат да бъдат и отрицателни

# main - заглавие на графиката

# col - цвят на стълбовете

# Тези параметри са основни и ги има и при другите графики


barplot(prop.table(tt))





# 4.2. Хистограма

# Използваме хистограма, когато искаме да представим разпределението на непрекъснати променливи

set.seed(7821)

r1 <- rnorm(n = 10^3, mean = 4, sd = 3)

hist(r1, main = "Хистограма (честотно разпределение)", xlab = "Нормално разпределение", ylab = "Честота",

col = "tomato3")

hist(r1, main = "Хистограма (вероятностно разпределение)", xlab = "Нормално разпределение", ylab = "Честота",

col = "tomato3", prob = T)


colors() # различни видове цветове, които се подържат от базовия пакет в R





# 4.3. Piechart

# Използваме piechart-, когато боравим с категорийни променливи и искаме да

# изобразим процентното им разпределение

cities <- c(rep("London", 14), rep("New York", 49), rep("Singapore", 28), rep("Mumbai", 36))
```

```
cities.table <- table(cities)

pie(cities.table, main = "City pie chart", col = rainbow(length(cities.table)))

# Броя на цветовете е хубаво да бъде равен на броя на категориите. В противен случай два сигмента
# ще бъдат оцветени в един и същи цвят.

piepercent<- round(100*cities.table/sum(cities.table), 1)

pie(cities.table, labels = piepercent, main = "City pie chart", col = rainbow(n = length(cities.table)))
# rainbow(n) - връща n на брой цветове, произтичащи от дъгата

legend(x = "topright", legend = c("London","New York","Singapore","Mumbai"), cex = 0.8,
fill = rainbow(length(cities.table)))
?legend
# x - разположение на графиката - може да слагате както координати, така и да описвате позицията
# legend - имената на категориите
# cex - големината на текста
# fill - цвета, на който отговаря текста в легендата

# 4.4. Boxplot
# При едномерния анализ, boxplot-а се използва, за да откриване на потенциални outlier-и.
tv_viewing_times <- c(25, 41, 27, 32, 43, 66, 35, 31, 15, 5, 34, 26, 32, 38, 16, 30, 38, 30, 20, 21)
tv_viewing_times_new <- c(tv_viewing_times, 240)

par(mfrow = c(1, 2))
```

```
boxplot(tv_viewing_times, col = "powderblue", main = "Boxplot", xlab = "TV
viewing")

boxplot(tv_viewing_times_new, horizontal = T, col = "palevioletred", main =
"Boxplot",

xlab = "TV viewing + outlier")

par(mfrow = c(1, 1))




# 4.5. Q-Q plot

# Проверяваме дали стойностите на наблюдаваната променлива се доближават до
теоретичните

# стойности на някое разпределение.


emp <- c(19.14, 6.29, 17.02, 6.13, 1.63, 18.78, 9.43, 11.21, 2.89, 9.52, 9.49,
4.83, 13.26, -0.96,

5.12, 1.39, 6.76, 2.1, 4.32, 1.38, 10.7, 9.01, 4.73, 11.59, 7.22, 1.53, 8.36,
10.91, 6.49,

3.69, 2.06, 15.92, 16.76, 18.13, 10.22, 19.25, 9.65, 17.75, 2.52, 1.24, 18.51,
11.52, 14.67,

12.65, 11.22, 27.78, 1.76, 9.64, 11.42, 12.29)


d1 <- rnorm(n = 10^2, mean = mean(emp), sd = sd(emp))

d2 <- rcauchy(n = 10^2, location = mean(emp), scale = sd(emp))



par(mfrow = c(1, 2))

qqplot(emp, d1, ylab = "theoretical distribution", main = "Check for normal
distr")

abline(a = 0, b = 1)


qqplot(emp, d2, ylab = "theoretical distribution", main = "Check for cauchy
distr")

abline(a = 0, b = 1)

par(mfrow = c(1, 1))
```

```
# abline() - чертае права линия

# a - изместване по X, b - ъгъл на правата, v - вертикална линия, h -
хоризонтална линия
```