Osnove statistike z R

Nataša Kejžar

2019-11-15

Kazalo

Poglavje 1

Predgovor

Skripta je nastala v času predavanj in vaj pri predmetu Računalniška podpora statistike. Namenjena aktivnemu delu z računalnikom in vsebuje veliko nalog z rešitvami.

Za pomoč pri nastanku skripte se zahvaljujem vsem sodelavcem statistikom na Inštitutu za biostatistiko in medicinsko informatiko (Univerza v Ljubljani, Medicinska fakulteta).

Poglavje 2

Programska orodja za statistiko

- splošni programski jeziki, low-level, hitrejši (Fortran, C/C++, Java, Python,...)
- bolj matematično statistični, uporabniku prijaznejši (Mathematica, Matlab/Octave, R, Julia)

2.1 R

- uporablja vektorje, matrike in delo z njimi
- open-source
- najbolj uporabljan jezik v statistiki
- uporabniki dograjujejo njegovo funkcionalnost s programskimi paketi
- instalacija z osnovnim grafičnim vmesnikom
  - RStudio za dodatne funkcionalnosti
  - IDE (integrated development environment)
- dokumentacija dosegljiva prek RStudia, na spletu, forum *stackexchange* (omejite se z R)

2.2 RStudio, Rmarkdown

cheatsheet **rstudio-IDE**

- ukazna vrstica, R Markdown izpis
- help, knjižnice, pregled grafov, pregled datotek v mapi
- gobalno okolje (Global Environment), zgodovina ukazov

- datoteke (bližnjice <Ctrl> + <Enter>, <Ctrl> + <Shift> + <Enter>)

cheatsheet: **rmarkdown**

- .Rmd datoteka
- reproducible research (ponovljive raziskave)
- koda v R med tremi enojnimi narekovaji in crko `r`: “‘{r } <code> “‘
- R koda znotraj besedila med enima enojnima narekovajema in crko `r`: ‘{r <code>}‘
- enačbe kot v LaTeX-u (cheatsheet **LaTeX**)
- za kreiranje PDF dokumentov: `install.packages('tinytex'); tinytex::install_tinytex()`

Naloge

- Naredite nov .Rmd dokument kot je prikazan na sliki:

Moj .Rmd dokument

- Uvod
- Zaključek

Uvod

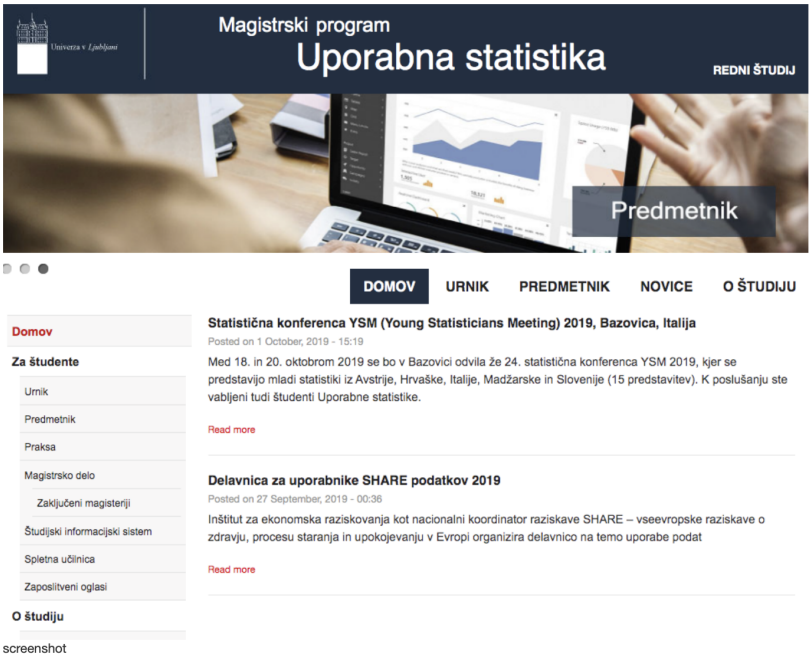
Danes začenjamo s predmetom Računalniška podpora statistike. Poleg tega predmeta se v tem semestru izvajajo še naslednji predmeti:

predmet	tip	število KT
Uvod v statistiko	obvezni	5
Matematika za statistike	strokovno-izbirni	10
Verjetnost	strokovno-izbirni	5
Bayesova statistika	strokovno-izbirni	5
Osnove teoretične statistike	obvezni	10
Linearni modeli	obvezni	5

Spletna stran študija

Spletno stran študija dosežete prek te povezave.

Izgled spletne strani vidite na sliki 1.



screenshot

Zaključek

V tem .Rmd dokumentu smo nastavili kazalo (le najvišja poglavja), vključili hiperpovezavo in sliko od zunaj. Vključevanje R se bomo naučili malce kasneje.

Slika 2.1: Prvi Rmd dokument

Poglavje 3

Osnovni gradniki v R

- znak pred komentarjem

? - znak za izpis pomoči (ali pa uporaba funkcije `help()`)

cheatsheet: **base-r**

3.1 Vrste stavkov

3.1.1 Izrazi

```
2+4
```

```
## [1] 6
```

3.1.2 Prirejanja

```
a1 = 1+6
a2 <- "vaje"
a3 = a1 == a2
```

`a1` je objekt, ki je shranjen v globalnem okolju.

Logične primerjave: `==, !=, <, >, is.na(), is.null()`

Operacije: `+, -, /, *, %%, %/%` (zadnja dva: deljenje po modulu in celoštevilsko deljenje)

3.2 (Osnovne) vrste objektov

3.2.1 Skalar

```
as1 = 1 # numeric
as2 = "Uporabna statistika" # character
as3 = TRUE # logical
str(as3)
```

```
##  logi TRUE
```

```
class(as3)
```

```
## [1] "logical"
```

```
is.numeric(a1)
```

```
## [1] TRUE
```

```
as.numeric(a3)
```

```
## [1] 0
```

```
ls() # list of objects
```

```
##   [1] "a"                     "a1"
##   [3] "a2"                    "a3"
##   [5] "aa"                    "aDelez"
##   [7] "af1"                   "af2"
##   [9] "af3"                   "af4"
##  [11] "af5"                   "al1"
##  [13] "am1"                   "am2"
##  [15] "am3"                   "as"
##  [17] "as1"                   "as2"
##  [19] "as3"                   "av1"
##  [21] "av2"                   "av3"
##  [23] "av4"                   "b"
##  [25] "barva"                 "bisekcija"
##  [27] "biseri"                "cbPalette"
##  [29] "celsius_to_fahrenheit" "d1"
##  [31] "d2"                    "df1"
##  [33] "drzave"                "funkcija"
##  [35] "galton"                "i"
##  [37] "interval"              "iris"
##  [39] "irisL"                 "izpisAB"
##  [41] "izpisSkupin"           "k"
##  [43] "kelvin_to_celsius"     "kelvin_to_fahrenheit"
##  [45] "limit025"              "limit975"
```

```
##  [47] "m1"                  "m2"
##  [49] "matrika"             "me1"
##  [51] "me2"                 "med"
##  [53] "moski"               "n"
##  [55] "nGroup"              "obeVar"
##  [57] "op"                  "opDelez"
##  [59] "opFrek"              "p"
##  [61] "p1"                  "p2"
##  [63] "p3"                  "pb"
##  [65] "pb1"                 "pb2"
##  [67] "pb3"                 "perms"
##  [69] "podatki"             "podatki2"
##  [71] "potop"               "povp"
##  [73] "povpN"               "povprecja"
##  [75] "povpTeze"            "powerX"
##  [77] "pp1"                 "pp2"
##  [79] "pricakovana"         "pricakovaneFrek"
##  [81] "pricakVred"          "pricStPotopov"
##  [83] "probabIn"            "resultA"
##  [85] "resultL"             "rezultat"
##  [87] "rezultatT"           "rezultatT2"
##  [89] "seznam"              "spLim"
##  [91] "square"              "st10"
##  [93] "starosti"            "stdo"
##  [95] "stdOdklon"           "stdOdklon2"
##  [97] "studenti"            "tabela"
##  [99] "tabela2"             "tabela3"
## [101] "teor"                "teorV"
## [103] "testna"              "tmp"
## [105] "USArrests"           "varianca"
## [107] "vsota"               "vsotaChi2"
## [109] "vsotaV"              "vzorci"
## [111] "vzorec"              "vzorec1"
## [113] "vzorec2"             "vzorec3"
## [115] "vzorec6"             "vzorecN"
## [117] "vzorecSum"           "x"
## [119] "Y"                   "z"
## [121] "zaIzris"             "zenske"
## [123] "zgLim"
```

Datume bomo obravnavali kasneje.

Naloge

- Kakšna je numerična vrednost `as3`?

```
as.numeric(as3)
```

```
## [1] 1
```

- Kako izbrišemo vse, kar je trenutno v globalnem okolju? Poglejte v help funkcije `rm()`.

```
rm(list=ls())
```

Posebne vrednosti:

- `NA` - not available
- `pi` - π
- `NaN` - not a number
- `Inf` - infinite value
- `NULL` - brez vrednosti, prazno
- `TRUE` in `FALSE` (okrajšavi `T` in `F`)

3.2.2 Vektor

```
av1 = c(1,2,3,4,5)
av2 = vector(mode="character",length=4)
av2
```

```
## [1] "" "" "" ""
```

```
class(av2)
```

```
## [1] "character"
```

```
av2[1] = "\u017Div\u00E9" # Žive
av2[2] = "naj"
av2[4] = "narodi"
av2
```

```
## [1] "Živé"   "naj"    ""       "narodi"
```

```
av2[-1]
```

```
## [1] "naj"    ""       "narodi"
```

```
av3 = 1:10
length(av3)
```

```
## [1] 10
```

Naloge

- Naredite vektor `av4`, v katerem so števila in znaki. Izpišite ga na zaslon. Kakšnega tipa je?
- Vektor `av3` skrajšajte:
  - na prve tri znake,
  - na zadnja dva znaka REZULTAT: 1, 2, 3,
  - izberite le vsak drugi znak,
  - vektor naj vsebuje samo lihe številke
- Vektor `av2` podaljšajte za naslednjo vrstico Zdravljice.
- Kaj se zgodi, če seštejemo `av1` in `av3[1:2]`? (krajši vektor se podvoji)

```
av4 = c("beseda",1,2,"stevilka")
av4
```

```
## [1] "beseda"   "1"        "2"        "stevilka"
```

```
str(av4)
```

```
##  chr [1:4] "beseda" "1" "2" "stevilka"
```

```
av3[(length(av3)-1):length(av3)]
```

```
## [1]  9 10
```

```
av3[seq(2,length(av3),by=2)]
```

```
## [1]  2  4  6  8 10
```

```
av3[av3%%2==1]
```

```
## [1] 1 3 5 7 9
```

```
av2[5:8] = c("ki","hrepene","dočakat","dan")
```

Seštevamo lahko tudi vektorje neenakih dolžin, vendar moramo biti pri tem **zelo pazljivi**! (npr. prištevamo skalar)

3.2.3 Matrika

```
am1 = matrix(c(1:6),nrow=2)
am1
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
am1[2,3] = 10
am1
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4   10
```

```
am1[2,] # second row
```

```
## [1]  2  4 10
```

```
is.matrix(am1)
```

```
## [1] TRUE
```

```
dim(am1)
```

```
## [1] 2 3
```

```
# dimensions can be added to the vector
dim(av3) = c(1,10)
# matrix as binded vectors
am2 = cbind(av1,av1)
class(am2)
```

```
## [1] "matrix"
```

```
str(am2)
```

```
##  num [1:5, 1:2] 1 2 3 4 5 1 2 3 4 5
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:2] "av1" "av1"
```

```
colnames(am2)
```

```
## [1] "av1" "av1"
```

```
am3 = rbind(av1,av1,rev(av1))
am1[1,]
```

```
## [1] 1 3 5
```

```
am1[1,,drop=FALSE] # preserve dimensions
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
```

Naloge

- Seštejte dve matriki `am1`.
- Zmnožite `am1`z 2.
- Kaj dobite z naslednjim izrazom: `am1*c(1,2)` ? Zakaj? (krajši vektor se podvoji)

- Kaj dobite z naslednjim izrazom: `am1*c(1,2,3)` ? Zakaj? (krajši vektor se podvoji)
- Kaj dobite, če po vrsticah skupaj združite `av1` in vektor `1:8`? Zakaj?
- Kaj dobite, če želite po stolpcih združiti `av1` in `av2`? Zakaj? (matrika znakovnih nizov)
- Kaj omogoča parameter `byrow` v funkciji `matrix`? Zapišite primer.

Vektorsko množenje in množenje matrik (`%*%`):

```
am1 %*% t(am1) # transpose
```

```
##      [,1] [,2]
## [1,]   35   64
## [2,]   64  120
```

```
av1 %*% t(av1)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    3    4    5
## [2,]    2    4    6    8   10
## [3,]    3    6    9   12   15
## [4,]    4    8   12   16   20
## [5,]    5   10   15   20   25
```

```
as.matrix(av1)
```

```
##      [,1]
## [1,]    1
## [2,]    2
## [3,]    3
## [4,]    4
## [5,]    5
```

```
av1 %*% av1
```

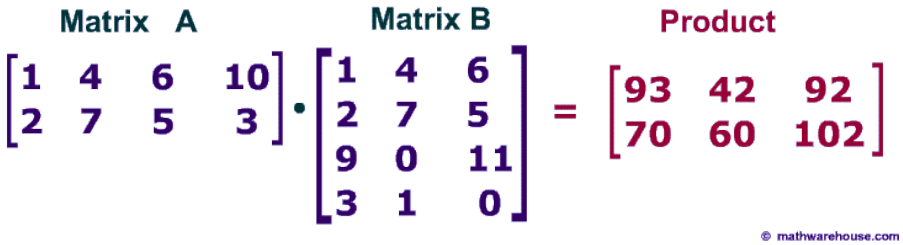
```
##      [,1]
## [1,]   55
```

```
t(av1) %*% av1
```

```
##      [,1]
## [1,]   55
```

Naloge

- Izračunajte naslednji produkt (preverite, ali je rezultat na desni pravilen)



+ Preverite v help-u, kaj naredi funkcija `solve`. (`?solve`) + Uporabite funkcijo `solve`, da boste dobili kvadratno matriko X dimenzije 4x4, za katero velja, da `Y %*% X = Y`. Matrika Y naj bo katerakoli matrika dimenzije 4x4. Zakaj dobite tak rezultat? (numerične metode, zaokroževanje)

```
set.seed(2019)
Y = matrix(runif(16),ncol=4)
solve(Y,Y)
```

```
##      [,1]          [,2] [,3] [,4]
## [1,]    1  0.000000e+00    0    0
## [2,]    0  1.000000e+00    0    0
## [3,]    0 -4.324218e-20    1    0
## [4,]    0 -7.102304e-18    0    1
```

```
round(solve(Y,Y))
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
```

3.2.4 Seznam

```
al1 = list(ime=c("Anton","Janez"),priimek=c("Novak","Trilar"),
           starost=c(67,58,34))
al1[1]
```

```
## $ime
## [1] "Anton" "Janez"
```

```
al1$priimek
```

```
## [1] "Novak"  "Trilar"
```

```
al1$starost[3]
```

```
## [1] 34
```

```
al1[[3]]
```

```
## [1] 67 58 34
```

```
al1[[3]][1]
```

```
## [1] 67
```

```
names(al1)
```

```
## [1] "ime"     "priimek" "starost"
```

Seznami so uprabni za združevanje več (različnih tipov) objektov v skupni objekt. Rezultati funkcij so ponavadi seznami različnih objektov.

3.2.5 Podatkovni okvir (osnovna statistična tabela)

```
#df1 = data.frame(ime=c("Anton","Janez"),priimek=c("Novak","Trilar"),
      #starost=c(67,58,34))
df1 = data.frame(ime=c("Anton","Janez"),priimek=c("Novak","Trilar"),
                 starost=c(67,58))
df1
```

```
##     ime priimek starost
## 1 Anton   Novak      67
## 2 Janez  Trilar      58
```

```
df1$starost
```

```
## [1] 67 58
```

```
df1[,"priimek"]
```

```
## [1] Novak  Trilar
## Levels: Novak Trilar
```

```
df1[[2]]
```

```
## [1] Novak  Trilar
## Levels: Novak Trilar
```

```
dim(df1)
```

```
## [1] 2 3
```

```
names(df1)
```

```
## [1] "ime"     "priimek" "starost"
```

```
rownames(df1) = c("oseba1","oseba2")
df1
```

```
##          ime priimek starost
## oseba1 Anton   Novak      67
## oseba2 Janez  Trilar      58
```

```
str(df1)
```

```
## 'data.frame':    2 obs. of  3 variables:
##  $ ime    : Factor w/ 2 levels "Anton","Janez": 1 2
##  $ priimek: Factor w/ 2 levels "Novak","Trilar": 1 2
##  $ starost: num  67 58
```

```
class(df1)
```

```
## [1] "data.frame"
```

```
class(df1$ime)
```

```
## [1] "factor"
```

Faktor je poseben tip podatka. Gre za opis kategorialnih podatkov, ki jim lahko priredimo opis posameznih kategorij in povemo, ali so urejeni.

```
af1 = c(0,0,0,0,1,1,1,1) # 4 men, 4 women, numeric
af2 = as.factor(af1)
af2
```

```
## [1] 0 0 0 0 1 1 1 1
## Levels: 0 1
```

```
as.numeric(af2) # factor  starts with 1, levels sorted
```

```
## [1] 1 1 1 1 2 2 2 2
```

```
af3 = factor(af1,levels = c(0,1),labels=c("M","F"))
# ordered=TRUE for ordered factors
af3
```

```
## [1] M M M M F F F F
## Levels: M F
```

Naloge

- Iz `af3` izbrišite vse moške in shranite rezultat v `af4`. Kakšen je izpis? Katere vrednosti ima lahko faktor?
- Ali lahko faktor dodamo novo kategorijo? V `af3` poskusite dodati kategorijo `0` - otrok. Kaj se zgodi? Kako boste dodali otroka, da bo to tudi nova kategorija?
- Poženite ukaz `data()`.
- Za podatkovje `USArrests` ugotovite, kaj so statistične enote in preverite, kakšne spremenljivke imamo na voljo (katerega tipa, kaj pomenijo).

- Prikažite le imena držav.
- Izberite iz podatkov vse države, ki imajo vsaj 70% populacije urbane. Koliko jih je?
- Iz podatkov izbrišite spremenljivko `Rape`.
- Kaj se zgodi, če `USArrests` spremenimo v matriko? Zakaj?

```
af4 = af3[af3!="M"] # se vedno ohranimo moznost za vrednost "M"
af5 = factor(af3,levels=c("M","F","O"))
af5[length(af5)+1] = "O"
rownames(USArrests)
```

```
##  [1] "Alabama"        "Alaska"         "Arizona"        "Arkansas"
##  [5] "California"     "Colorado"       "Connecticut"    "Delaware"
##  [9] "Florida"        "Georgia"        "Hawaii"         "Idaho"
## [13] "Illinois"       "Indiana"        "Iowa"           "Kansas"
## [17] "Kentucky"       "Louisiana"      "Maine"          "Maryland"
## [21] "Massachusetts"  "Michigan"       "Minnesota"      "Mississippi"
## [25] "Missouri"       "Montana"        "Nebraska"       "Nevada"
## [29] "New Hampshire"  "New Jersey"     "New Mexico"     "New York"
## [33] "North Carolina" "North Dakota"   "Ohio"           "Oklahoma"
## [37] "Oregon"         "Pennsylvania"   "Rhode Island"   "South Carolina"
## [41] "South Dakota"   "Tennessee"      "Texas"          "Utah"
## [45] "Vermont"        "Virginia"       "Washington"     "West Virginia"
## [49] "Wisconsin"      "Wyoming"
```

```
drzave = rownames(USArrests)[USArrests$UrbanPop>=70]
length(drzave)
```

```
## [1] 21
```

```
USArrests$Rape = NULL
```

3.2.6 Array (večdimenzionalna tabela)

```
aa = array(dim=c(3,5,2))
aa[1,,2] = "M"
aa[,,1] = 3
dimnames(aa) = list(c("oseba1","oseba2","oseba3"),
                    c("cas1","cas2","cas3","cas4","cas5"),
                    c("treatment","placebo"))
```

Poglavje 4

Osnove programiranja

4.1 If stavek

```
x = 3
if (x > 0){
  x = x + 1
} else {
  x = x - 1
}
x
```

```
## [1] 4
```

Pogoj mora biti zapisan v oklepajih. Pogoje lahko poljubno sestavljamo z: `!`, `&`, `&&`, `|`, `||`, `xor()`, `isTRUE(x)`, `isFALSE(x)`

Kratki if stavek (le, če v je v telesu if stavka samo eno prirejanje):

```
ifelse(x > 0, x+1, x-1)
```

```
## [1] 5
```

```
ifelse(x > 0, x+1, x-1)
```

```
## [1] 5
```

Naloge

- Napišite if stavek, ki bo izračunal `a/b` in vrnil izpis na standardni izhod. Če izračun ni mogoč, se na standardni izhod izpiše `Deljenje z 0 ni definirano.`.

- Zakaj je v drugem primeru (kratek if stavek) obakrat rezultat (`x`) enak? (REŠITEV: nove vrednosti x nismo shranili)
- Kaj vrne `isTRUE(x)`? Kaj pa `isTRUE(as.logical(x))`? Zakaj?
- Zapišite kratki if stavek, ki bo za vse države iz `USArrests` določil novo spremenljivko `Urban`, ki bo imela vrednost `yes`, če je v državi več kot 75% urbane populacije, sicer `no`.

```
# naloga 1
a=5
b=4
if(b!=0){
  a/b
}else{
  print("Deljenje z 0 ni definirano.")
}
```

```
## [1] 1.25
```

```
# naloga 4
USArrests$Urban = ifelse(USArrests$UrbanPop>75,"yes","no")
```

4.2 For zanka

```
for (i in 1:10){
  print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

Naloge

- Preverite, kaj se zgodi, če v telesu zanke zapišemo le `i`.
- Zapišite if stavek s pomočjo for zanke, ki bo za vse države iz `USArrests` določil novo spremenljivko `Urban2`, ki bo imela vrednost `yes`, če je v državo več kot 75% urbane populacije, sicer `no`.

- S pomočjo for zanke seštejte vse vrednosti v številskem vektorju `1:100`. Ko to naredite, izpišite še povprečje.
- V vektorju `av2` sta zapisani prvi dve kitici Zdravljice. Zapišite vsako besedo v svoji vrstici s pomočjo for zanke in poleg še dolžino besede, ki je izpisana (uporabite funkciji `strsplit()` in `paste()`).
- S pomočjo for zanke zapišite vsako kitico Zdravljice iz `av2` v svojo vrstico (kitica je dolga 4 besede).

```
# naloga 2
for(i in 1:dim(USArrests)[1]){
  if(USArrests$UrbanPop[i] > 75){
    USArrests$Urban2[i] = "yes"
  }else{
    USArrests$Urban2[i] = "no"
  }
}

# naloga 3
vsota = 0
for(i in 1:100){
  vsota = vsota + i
}
vsota
```

```
## [1] 5050
```

```
sum(1:100)
```

```
## [1] 5050
```

```
# naloga 4
for(i in av2){
  seznam = strsplit(i,split="")
  n = length(seznam[[1]])
  print(paste(i,n,"znakov"))
}
```

```
## [1] "Živé 4 znakov"
## [1] "naj 3 znakov"
## [1] " 0 znakov"
## [1] "narodi 6 znakov"
## [1] "ki 2 znakov"
## [1] "hrepene 7 znakov"
## [1] "dočakat 7 znakov"
## [1] "dan 3 znakov"
```

```
# naloga 5
for(i in seq(1,length(av2),by=4)){
  tmp = av2[i:(i+3)]
```

```
  print(paste(tmp,collapse=" "))
}
```

```
## [1] "Živé naj  narodi"
## [1] "ki hrepene dočakat dan"
```

4.3 While zanka

```
x = 2345
while (x > 100){
  x = x/2
  print(x)
}
```

```
## [1] 1172.5
## [1] 586.25
## [1] 293.125
## [1] 146.5625
## [1] 73.28125
```

4.4 Funkcija

```
# definition
square <- function(x){
  squared <- x*x
  return(squared)
}
# function call
square(3)
```

```
## [1] 9
```

```
# arguments
powerX <- function(x,p = 2){
  result = 1
  for(i in 1:p){
    result = result * x
  }
  return(result)
}
# function call
powerX(4)
```

```
## [1] 16
```

```
powerX(x=4,p=3)
```

```
## [1] 64
```

```
powerX(3,2)
```

```
## [1] 9
```

```
powerX(p=3,x=2)
```

```
## [1] 8
```

```
powerX <- function(x,p = 2){
  x^p
}
powerX(p=3,x=2)
```

```
## [1] 8
```

Naloge

- Zapišite funkcijo, ki izračuna vzorčni standardni odklon opazovanj, ki so shranjena v nekem številskem vektorju. Preverite, ali deluje pravilno.
- Zapišite funkcijo, ki na standardni izhod izpiše cela števila od neke številke `a` do `b`. To naj se izvrši le, če je mogoče in, če sta številki za največ 20 števil narazen. Preverite, ali funkcija deluje.

```
# naloga 1
stdOdklon <- function(x){
  povp = mean(x)
  vsotaKO = 0 # vsota kvadriranih odklonov od povprecja
  for(xi in x){
    vsotaKO = vsotaKO + (xi-povp)^2
  }
  n = length(x)
  varianca = vsotaKO/(n-1)
  return(sqrt(varianca))
}
stdOdklon(1:10)
```

```
## [1] 3.02765
```

```
sd(1:10) # funkcija v R, ki nam to izracuna
```

```
## [1] 3.02765
```

```
stdOdklon2 <- function(x){
  povp = mean(x)
```

```
  vektorKO = (x-povp)^2
  vsotaKO = sum(vektorKO)
  n = length(x)
  varianca = vsotaKO/(n-1)
  return(sqrt(varianca))
}
stdOdklon2(1:10)
```

```
## [1] 3.02765
```

```
# naloga 2
izpisAB <- function(a,b){
  if(b<a){
    warning("izpis ni mogoc")
  }
  if(b-a>20){
    warning(paste("stevili",a,"in",b,"sta predalec narazen"))
  }
  print(a:b)
}
izpisAB(2,4)
```

```
## [1] 2 3 4
```

```
izpisAB(2,40)
```

```
##  [1]  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
## [24] 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
```

```
izpisAB(10,2)
```

```
## [1] 10  9  8  7  6  5  4  3  2
```

```
izpisAB(b=10,a=4)
```

```
## [1]  4  5  6  7  8  9 10
```

- Naredite 3 funkcije za pretvorbo med temperaturnimi lestvicami:
  - Kelvin_to_Celsius ($cel = kel - 273.15$)
  - Celsius_to_Fahrenheit ($fahr = cel * 9/5 + 32$)
  - Kelvin_to_Fahrenheit (tako, da uporabite zgornji funkciji)

```
# This function converts input temperatures in Kelvin to Celsius.
kelvin_to_celsius <- function(temp_K) {
  temp_C <- temp_K - 273.15
  temp_C
}

# This function converts input temperatures in Celsius to Fahrenheit.
```

```
celsius_to_fahrenheit <- function(temp_C) {
  temp_F <- (temp_C * 9/5) + 32
  temp_F
}

# This function converts input temperatures in Kelvin to Fahrenheit.
kelvin_to_fahrenheit <- function(temp_K) {
  temp_C <- kelvin_to_celsius(temp_K)
  temp_F <- celsius_to_fahrenheit(temp_C)
  temp_F
}
```

- Z bisekcijo bi radi dobili x, kjer je $f(x) \approx 0$, za funkcijo $f(x) = x^3 - x - 2$ med točkama 1 in 2.

  ALGORITHM: The input for the method is a continuous function f, an interval $[a, b]$, and the function values $f(a)$ and $f(b)$. The function values are of opposite sign (there is at least one zero crossing within the interval). Each iteration performs these steps:

  - Calculate c, the midpoint of the interval, $c = (a + b)/2$.
  - Calculate the function value at the midpoint, f(c).
  - If convergence is satisfactory (that is, $c - a < 10^{-5}$ is sufficiently small, or $|f(c)| < 10^{-5}$ is sufficiently small), return c and stop iterating.
  - Examine the sign of $f(c)$ and replace either $(a, f(a))$ or $(b, f(b))$ with $(c, f(c))$ so that there is a zero crossing within the new interval.

  NAVODILO:

  - Najprej zapišite funkcijo f(x) kot funkcijo v R
  - Naredite program, ki bo izračunal ničlo funkcije dovolj natančno.
  - Program spremenite v funkcijo v R. Funkcija naj vsebuje 3 vhodne argumente: a, b in ime funkcije (ime funkcije je legalen vhodni argument).

```
funkcija <- function(x){
  return(x^3 - x - 2)
}

bisekcija <- function(a,b,FUN){
  razmik = 1
  absF = 1
  while((razmik >= 10^(-5)) & (absF >= 10^(-5))){
    c = (a+b)/2 # calculate midpoint
    fc = FUN(c) # calculate f(c)
    razmik = c - a
    absF = abs(fc)
    if(fc<0){ # make new interval
```

```
      a = c
    }else{
      b = c
    }
  }
  return(c)
}

bisekcija(a=1,b=2,FUN=funkcija)
```

```
## [1] 1.521385
```

Obstaja ogromno funkcij, ki so v R že definirane. Za vse lahko preverite, kakšni so njihovi vhodni parametri, kateri so *obvezni* in kateri *neobvezni*.

- `print`
- `length`
- `factor`
- `sum`
- `mean`
- `var`
- `seq` zaporedje
- `rep` za hitro pisanje daljših vektorjev (times, each, length)
- `cut` za kategoriziranje spremenljivk
- `table` za povzemanje kategorialnih spremenljivk, tudi več kot ene
- `round`, `floor`, `ceiling`, `format` za zaokroževanje števil
- `which` za pridobivanje indeksov

Naloge (primeri za uporabo funkcij)

- Ugotovite, kako deluje funkcija `which`. Z njeno pomočjo dobite države, ki imajo vsaj polovico urbanega prebivalstva. (REŠITEV: pridobivanje indeksov v objektu)
- Izračunajte povprečno število umorov (`USArrests`), dodajte mu vzorčno varianco.
- Zapišite zaporedje sodih naravnih števil do 100.
- Izpišite vektor, ki ima 10 enic, 5 dvojk, 6 trojk in 20 štiric (v tem vrstnem redu).
- Dodajte podatkovju `USArrests` še spremenljivko `State`, kjer bodo zapisana imena držav.
- Dodajte podatkovju `USArrests` spremenljivko `Group`, ki bo države razdelila na 5 skupin (prvih 10 bo v 1. skupini itn.). Spremenite to spremenljivko v faktor, vrednosti naj bodo od `group1` do `group5`.

- Spremenljivko `USArrests$UrbanPop` kategorizirajte v naslednje kategorije:
  - 0-39%
  - 40-65%
  - 66-80%
  - nad 80%

  Naredite faktor, ki bo urejen, imena kategorij si izmislite sami. Kot novo spremenljivko `UrbanCat` jo dodajte v `USArrests`.
- Na spremenljivki iz prejšnje naloge uporabite funkcijo `table`.
- Ali so skupine (spremenljivke `Group`) približno enakovredno porazdeljene glede na spremenljivko `UrbanCat`? Preverite s kontingenčno tabelo.

```
mean(USArrests$Murder)
```

```
## [1] 7.788
```

```
sd(USArrests$Murder)
```

```
## [1] 4.35551
```

```
seq(from=2,to=100,by=2)
```

```
##  [1]   2   4   6   8  10  12  14  16  18  20  22  24  26  28  30  32  34
## [18]  36  38  40  42  44  46  48  50  52  54  56  58  60  62  64  66  68
## [35]  70  72  74  76  78  80  82  84  86  88  90  92  94  96  98 100
```

```
USArrests$State = rownames(USArrests)

n = dim(USArrests)[1]
nGroup = rep(1:5,each=10)
USArrests$Group = paste(rep("group",n),nGroup,sep="")

USArrests$UrbanCat = cut(USArrests$UrbanPop,breaks = c(0,39,65,80,100))

table(USArrests$UrbanCat)
```

```
##
##   (0,39]  (39,65]  (65,80] (80,100]
##        2       20       20        8
```

```
table(USArrests$UrbanCat,USArrests$Group)
```

```
##
##             group1 group2 group3 group4 group5
##   (0,39]         0      0      0      0      2
##   (39,65]        4      5      4      3      4
##   (65,80]        5      3      3      5      4
##   (80,100]       1      2      3      2      0
```

Poglavje 5

Delo z datotekami

Branje in pisanje datotek

```
# for .txt data
studenti = read.table("data/studenti2012.txt",sep="\t")
str(studenti)
```

```
## 'data.frame':    44 obs. of  12 variables:
##  $ V1 : Factor w/ 6 levels "20","21","22",..: 6 5 2 2 2 2 2 2 1 3 ...
##  $ V2 : Factor w/ 13 levels "0","1","10","11",..: 13 10 2 10 11 7 6 10 4 9 ...
##  $ V3 : Factor w/ 3 levels "F","M","spol": 3 2 1 1 1 1 2 1 1 1 ...
##  $ V4 : Factor w/ 26 levels "50","51","52",..: 26 25 11 6 17 15 23 3 4 13 ...
##  $ V5 : Factor w/ 24 levels "156","157","158",..: 24 18 16 18 11 15 15 7 6 12 ...
##  $ V6 : Factor w/ 27 levels "","154","156",..: 27 24 18 19 10 13 17 9 6 9 ...
##  $ V7 : Factor w/ 12 levels "36","37","38",..: 12 9 8 4 4 5 6 4 3 6 ...
##  $ V8 : Factor w/ 3 levels "lasje","S","T": 1 3 3 3 2 3 3 3 3 3 ...
##  $ V9 : Factor w/ 3 levels "oci","S","T": 1 2 3 3 3 2 3 3 3 2 ...
##  $ V10: Factor w/ 20 levels "","155","157",..: 20 2 6 14 5 13 9 5 4 14 ...
##  $ V11: Factor w/ 20 levels "","170","172",..: 20 11 14 11 19 7 12 2 11 15 ...
##  $ V12: Factor w/ 6 levels "L","M","majica",..: 3 1 4 4 4 2 5 4 4 2 ...
```

```
dimnames(studenti)
```

```
## [[1]]
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14"
## [15] "15" "16" "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27" "28"
## [29] "29" "30" "31" "32" "33" "34" "35" "36" "37" "38" "39" "40" "41" "42"
## [43] "43" "44"
##
## [[2]]
##  [1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11"
## [12] "V12"
```

```
studenti = read.table("data/studenti2012.txt",sep="\t",header=TRUE)
str(studenti)
```

```
## 'data.frame':    43 obs. of  12 variables:
##  $ starost: int  59 21 21 21 21 21 21 20 22 23 ...
##  $ mesec  : int  7 1 7 8 4 3 7 11 6 10 ...
##  $ spol   : Factor w/ 2 levels "F","M": 2 1 1 1 1 2 1 1 1 1 ...
##  $ masa   : int  91 60 55 70 65 88 52 53 62 59 ...
##  $ visina : int  178 173 178 167 171 171 162 161 168 169 ...
##  $ roke   : num  189 176 178 165 168 173 164 160 164 168 ...
##  $ cevelj : int  44 43 39 39 40 41 39 38 41 38 ...
##  $ lasje  : Factor w/ 2 levels "S","T": 2 2 2 1 2 2 2 2 2 1 ...
##  $ oci    : Factor w/ 2 levels "S","T": 1 2 2 2 1 2 2 2 1 1 ...
##  $ mati   : int  155 162 170 160 169 165 160 158 170 178 ...
##  $ oce    : int  180 184 180 190 176 182 170 180 185 180 ...
##  $ majica : Factor w/ 5 levels "L","M","S","XL",..: 1 3 3 3 2 4 3 3 2 2 ...
```

```
dimnames(studenti)
```

```
## [[1]]
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14"
## [15] "15" "16" "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27" "28"
## [29] "29" "30" "31" "32" "33" "34" "35" "36" "37" "38" "39" "40" "41" "42"
## [43] "43"
##
## [[2]]
##  [1] "starost" "mesec"   "spol"    "masa"    "visina"  "roke"    "cevelj"
##  [8] "lasje"   "oci"     "mati"    "oce"     "majica"
```

```
summary(studenti)
```

```
##     starost          mesec        spol        masa            visina
##  Min.   :20.00   Min.   : 0.000   F:33   Min.   : 50.00   Min.   :156.0
##  1st Qu.:21.00   1st Qu.: 5.000   M:10   1st Qu.: 55.50   1st Qu.:164.0
##  Median :21.00   Median : 7.000          Median : 61.00   Median :170.0
##  Mean   :22.07   Mean   : 6.814          Mean   : 78.07   Mean   :169.9
##  3rd Qu.:22.00   3rd Qu.: 9.500          3rd Qu.: 70.00   3rd Qu.:173.5
##  Max.   :59.00   Max.   :11.000          Max.   :700.00   Max.   :189.0
##
##       roke           cevelj      lasje  oci         mati
##  Min.   :154.0   Min.   :36.00   S:19   S:24   Min.   :155.0
##  1st Qu.:163.2   1st Qu.:38.00   T:24   T:19   1st Qu.:160.0
##  Median :167.8   Median :39.00                 Median :165.0
##  Mean   :169.3   Mean   :40.02                 Mean   :165.4
##  3rd Qu.:172.5   3rd Qu.:41.50                 3rd Qu.:168.0
##  Max.   :193.0   Max.   :48.00                 Max.   :180.0
##  NA's   :5                                     NA's   :5
```

```
##       oce        majica
##  Min.   :170.0   L : 5
##  1st Qu.:174.2   M :19
##  Median :179.5   S :16
##  Mean   :179.1   XL: 1
##  3rd Qu.:182.0   XS: 2
##  Max.   :190.0
##  NA's   :5
```

Neobvezni parameter `stringsAsFactors` vsak znakovni stolpec predstavi kot faktor. Rezultat branja je `data.frame`.

Naloge

- Kaj pri podatkih študentov - v povzetkih - opazite? Opazovanja, ki očitno niso prava, si poglejte bolj natančno in se odločite, kaj boste z njimi naredili:
  - izbrisali celo statistično enoto
  - popravili (če je smiselno)
  - izbrisali le vrednost pri spremenljivki (vrednost `NA`)
- Izberite pravi podatkovni tip za `mesec` in ga bolj natančno določite.
- Preglejte, kaj je s spremenljivko `majica`. Zakaj je vrstni red vrednosti tak, kot je? Popravite ga, da bo spremenljivka urejena.

```
# studenta s 700 kg spremenimo v 70 kg
studenti[studenti$masa == 700,"masa"] = 70
# izbrisemo 59-letnika
studenti = studenti[-which(studenti$starost == 59),]
# mesec 0 spremenimo v NA
studenti$mesec[studenti$mesec==0] = NA
# mesec postane faktor, ker imamo 12 mesecev je potrebno zapisati tudi "levels"
studenti$mesec2 = factor(studenti$mesec,levels=1:12,
                         labels=c("jan","feb","mar","apr","maj","jun",
                                  "jul","avg","sep","okt","nov","dec"))
# majica naj bo urejen faktor
summary(studenti$majica)
```

```
##  L  M  S XL XS
##  4 19 16  1  2
```

```
studenti$majica2 = factor(studenti$majica,
                          levels=c("XS","S","M","L","XL"),ordered=TRUE)
summary(studenti$majica2)
```

```
## XS  S  M  L XL
##  2 16 19  4  1
```

- Izpišite opisne statistike za težo moških in težo žensk posebej.

```
moski = studenti$visina[studenti$spol=="M"]
round(mean(moski,na.rm=TRUE),2)
```

```
## [1] 180.22
```

```
round(sd(moski,na.rm=TRUE),2)
```

```
## [1] 5.54
```

```
zenske = studenti$visina[studenti$spol=="F"]
round(mean(zenske,na.rm=TRUE),2)
```

```
## [1] 166.82
```

```
round(sd(zenske,na.rm=TRUE),2)
```

```
## [1] 5.42
```

- Naredite funkcijo za izpis povprečja in standardnega odklona za neko številsko spremenljivko. Kot neobvezni vhodni parameter naj bo podan še en vektor (faktor), ki določa skupine, za katere boste povprečja in standardni odklon izpisali. Preverite to na primeru višin za moške in ženske in na primeru višin za faktor `majica`.
- Preverite zgornjo funkcijo tudi za primere, ki ne bi smeli dati izpisov:
  - skupina ni faktor
  - skupina je drugačne dimenzije
  - osnovna spremenljivka ni številska

```
#skupina = studenti$majica
#x = studenti$masa
izpisSkupin <- function(x,skupina=NULL){
  # preverjanje x, skupina
  if(!is.numeric(x)){
    stop("x mora biti stevilska spremenljivka")
  }
  if(is.null(skupina)){
    skupina=factor(rep("all",length(x)))
  }
  if(!is.factor(skupina)){
    stop("skupina mora biti faktorska spremenljivka")
  }
  if(length(x)!= length(skupina)){
    stop("x in skupina morata biti enake dolzine")
  }
  groups = levels(skupina)
  for(g in groups){
    m1 = mean(x[skupina == g],na.rm=TRUE)
    s1 = sd(x[skupina == g],na.rm=TRUE)
```

```
    print(paste(g,":",m1,"(",s1,")"))
  }
}
#studenti[studenti$majica=="XS",]
izpisSkupin(studenti$starost,studenti$spol)
```

```
## [1] "F : 21.2121212121212 ( 0.819968587720581 )"
## [1] "M : 21.1111111111111 ( 0.781735959970572 )"
```

```
izpisSkupin(studenti$starost)
```

```
## [1] "all : 21.1904761904762 ( 0.803592398745607 )"
```

```
#izpisSkupin(studenti$spol)
#izpisSkupin(studenti$spol,studenti$starost)
#izpisSkupin(studenti$starost,studenti$starost)
```

- Preberite help za `write.table` in izpišite končno verzijo podatkov v datoteko `data/studenti2012_v2.txt`. Preverite novonastalo datoteko. To funkcijo uporabljamo npr., če želimo nekomu, ki ne pozna R pokazati podatke ...
- Naredite nov `vaja.Rmd` dokument v katerem bi radi iz datoteke `studenti` izpisali samo povzetek (`summary`). Na kaj je potrebno pri tem paziti (tj. da sploh lahko pretvorimo `.Rmd` dokument v npr. `html`)? (REŠITEV: da imamo v .Rmd vse na novo definirano, tj. vse knjižnice, vse uvoze podatkov ipd. Datoteka .Rmd se ob prevajanju (gumb `Knit`) obnaša kot nova R-jeva seja.)
- Shranite spremenjen `data.frame` studenti s pomočjo funkcije `save` in ga naložite z `load` v `vaja.Rmd`. Funkcija `save` shrani objekt, ki smo ga kreirali v R v nek zunanji dokument (ponavadi mu damo končnico `.RData`) in ni enostavno berljiv zunaj R-ja.

```
save(studenti,file="studenti.RData")
load("studenti.RData")
```

- Shranite studente z `dump` in jih preberite s `source` v `vaja.Rmd`. S funkcijo `dump` kreiramo novo datoteko, ki jo lahko preberemo kot R Script. Torej bi lahko svoje funkcije shranili v svoj R dokument npr. `funkcije.R` in jih prebrali v novo R-jevo sejo z `source("pot-do/funkcije.R")`.

```
dump("studenti",file="studenti.R")
source("studenti.R")
```

- Uporabite funkcijo `read.csv` da dobite novo podatkovje (podatki.xlsx) v Rjevo trenutno sejo. Naj bo v spremenljivki `podatki`.

```
# najprej prek Excela shranite "podatki.xlsx" v "podatki.csv"
# (comma separated value)
podatki = read.csv2("data/podatki.csv",sep=";")
```

```
# v primeru slovenskega Excela je locilo med stolpci ";"
```

5.1 Delo z datumi

V novi datoteki imamo dve datumski spremenljivki. Datum bi radi tudi v R zapisali pravilno.

Naloge

- Kakšni so trenutni tipi podatkov v datoteki?
- Spremenite jih tako, da bodo smiselni. Datumski spremenljivki pustite pri miru.
- Preverite razred `Date` in funkcijo `strptime`.

```
format(Sys.time(), "%a %b %d %X %Y %Z")
```

```
## [1] "Fri Nov 15 08:42:27 2019 CET"
```

```
x <- c("1jan1960", "2jan1960", "31mar1960", "30jul1960")
z <- as.Date(x, "%d%b%Y") # odvisno od lokalnih nastavitev sistema!!!
# v učilnici je bilo malce drugače kot na domačem računalniku
class(z)
```

```
## [1] "Date"
```

```
str(z)
```

```
##  Date[1:4], format: "1960-01-01" "1960-01-02" "1960-03-31" "1960-07-30"
```

```
as.numeric(z)
```

```
## [1] -3653 -3652 -3563 -3442
```

```
as.Date(0,origin="1970-01-01") # how much time from origin date
```

```
## [1] "1970-01-01"
```

Z datumi lahko tudi računamo: npr. jih odštevamo.

Naloge

- Spremenite obe datumski spremenljivki v razred (tip) `Date`. V primeru, da zapis ni povsod enak, razmislite, kaj bi bilo to najbolj smiselno narediti in to izvedite, da se podatki ujemajo.

```
# spremeni podatke
podatki = read.csv2("data/podatki.csv")
# v primeru slovenskega Excela je locilo med stolpci ";"
podatki$ID = as.factor(podatki$ID)
podatki$skupina = factor(podatki$skupina,labels=c("s1","s2","s3"))
podatki$terapija = factor(podatki$terapija,labels=c("brez","terapija"))
podatki$datum.testiranja = as.Date(x=podatki$datum.testiranja,
                                   format = "%d/%m/%Y")
podatki$datum.okužbe = as.Date(x=podatki$datum.okužbe, format = "%d/%m/%Y") # strptime
#podatki$datum.testiranja = as.Date(podatki$datum.testiranja,"%d.%m.%Y")
# to je delovalo v učilnici
#podatki$datum.okužbe = as.Date(podatki$datum.okužbe,"%d.%m.%Y")
# to je delovalo v učilnici
```

- Naredite novo spremenljivko `razlika`, kjer izračunate razliko med datumom testiranja in datumom okužbe. Preverite in izpišite vse enote, kjer pride do neskladja (tj. da je datum testiranja pred datumom okužbe). Kaj boste naredili s temi enotami?
- Naredite spremenljivko `razlikaLeta`, ki naj ima izraženo razliko v letih.
- Izračunajte povprečno razliko v letih za obe skupini (tiste na terapiji in tiste brez).

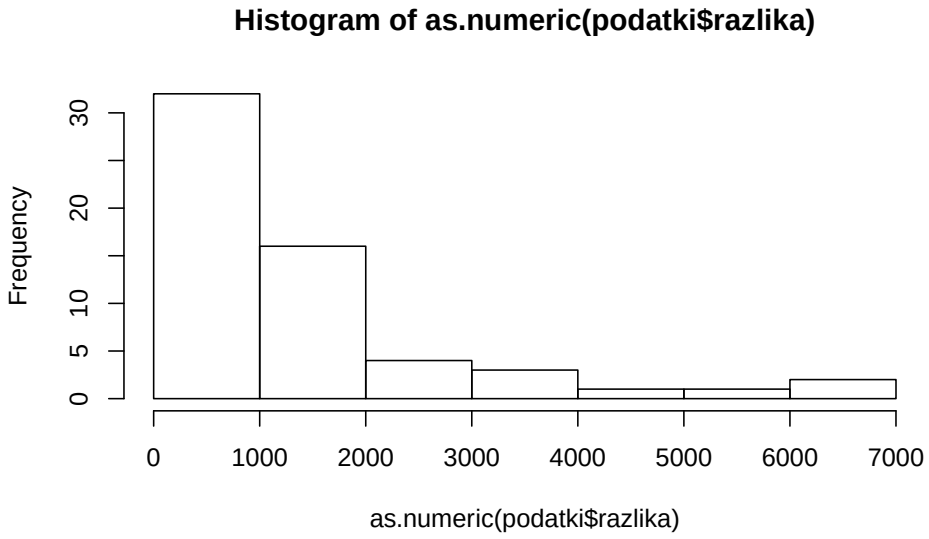
```
podatki$razlika = podatki$datum.testiranja - podatki$datum.okužbe
str(podatki$razlika)
```

```
##  'difftime' num [1:70] 1981 0 1463 76 ...
##  - attr(*, "units")= chr "days"
```

```
summary(as.numeric(podatki$razlika))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  -153.0   189.8   870.0  1281.7  1724.0  6939.0      10
```

```
podatki$datum.okužbe[podatki$razlika <0] = NA
podatki$datum.testiranja[podatki$razlika <0] = NA
podatki$razlika[podatki$razlika <0] = NA
podatki = podatki[-dim(podatki)[1],] # zadnja vrstica je prazna, jo izlocimo
hist(as.numeric(podatki$razlika))
```




```
podatki$razlikaLeta = as.numeric(podatki$razlika)/365.24

# funkcijo izpisSkupin smo definirali zgoraj
izpisSkupin(podatki$razlikaLeta,podatki$terapija)
```

```
## [1] "brez : 2.74626687007969 ( 3.16415787375177 )"
## [1] "terapija : 5.80525818639798 ( 5.49911794249053 )"
```

Poglavje 6

Risanje podatkov (opisna statistika)

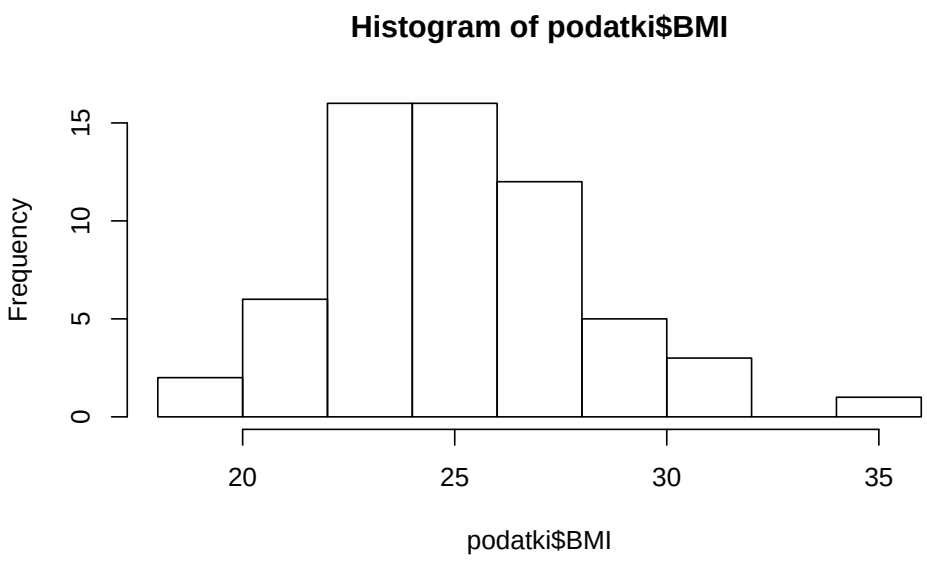
Podatke, ki smo jih začeli obdelovati, običajno tudi grafično pregledamo, preden začnemo z odgovarjanjem na raziskovalna vprašanja.

6.1 Številske spremenljivke

Funkcije `hist` za histogram, `boxplot` za okvir z ročaji, `plot` za razsevni grafikon. Za dodatne izrise:

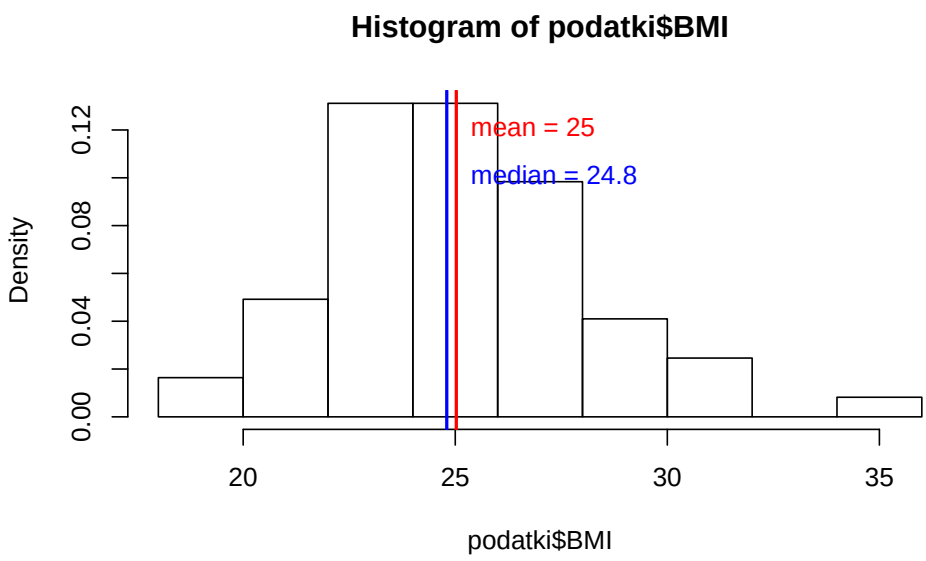
- `points`
- `lines`
- `segments`
- `text`
- `legend`

```
#source("podatki.R")
hist(podatki$BMI)
```

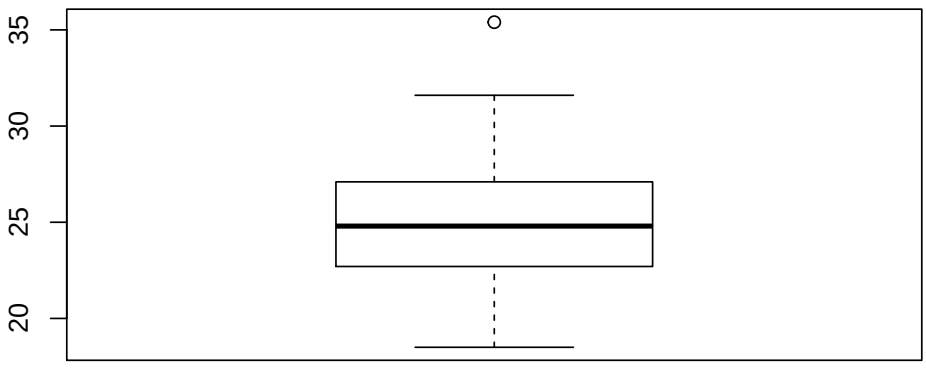



```
hist(podatki$BMI,xlab="indeks telesne teže",ylab="frekvenca",main="histogram")
```

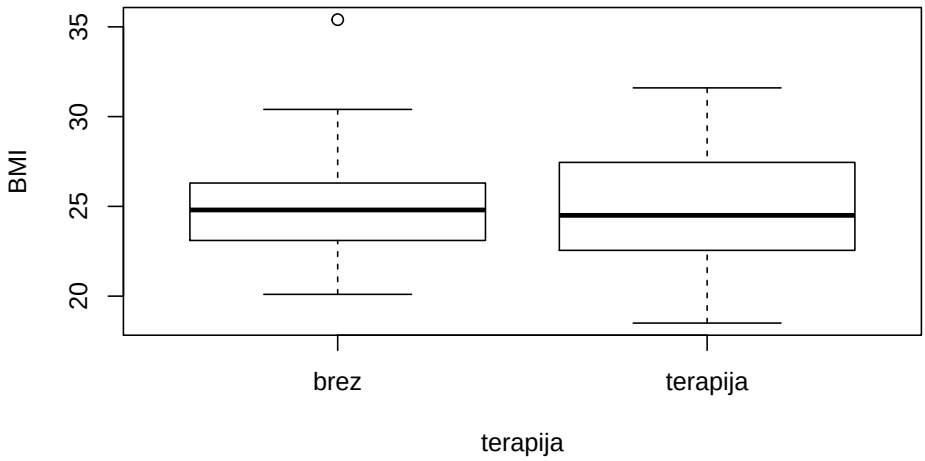

```
hist(podatki$BMI,freq=FALSE)
povp=mean(podatki$BMI,na.rm=TRUE)
abline(v=povp,col="red",lwd=2)
text(x=25,y=0.12,pos=4,labels=paste("mean =",round(povp,1)),col="red")
med=median(podatki$BMI,na.rm=TRUE)
abline(v=med,col="blue",lwd=2)
text(x=25,y=0.10,pos=4,labels=paste("median =",round(med,1)),col="blue")
```



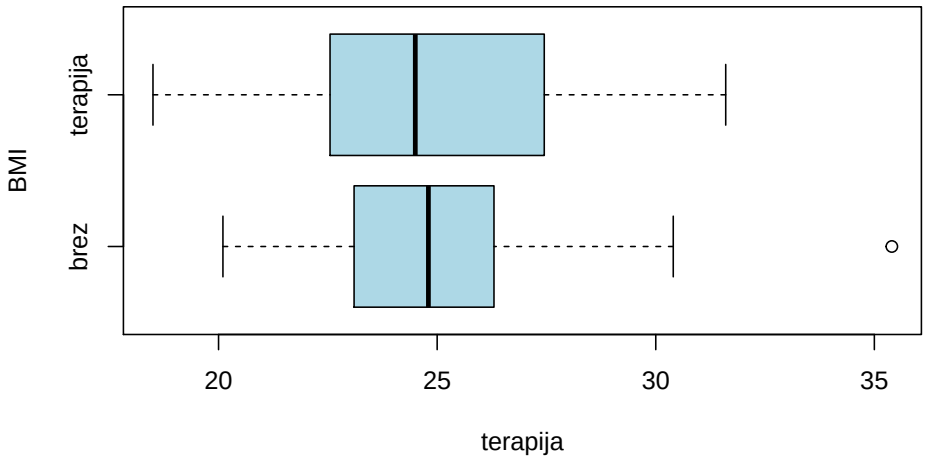

```
boxplot(podatki$BMI)
```



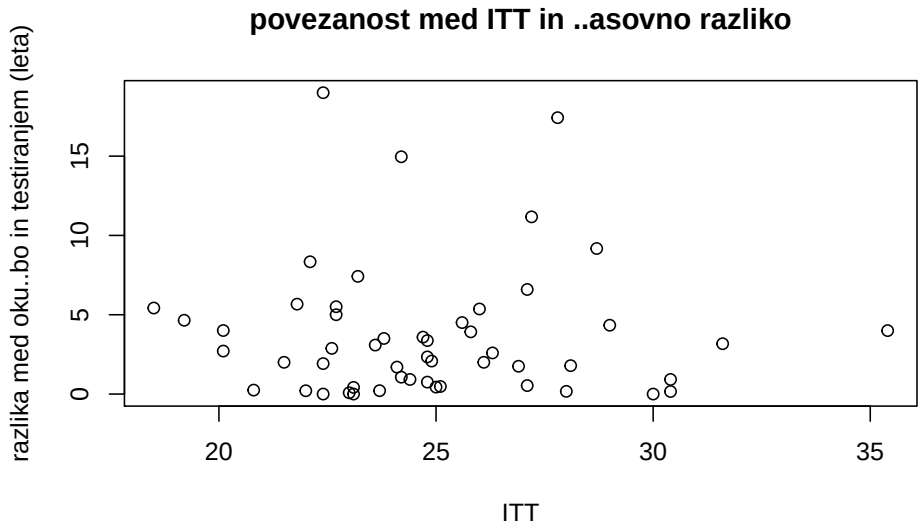

```
boxplot(formula=BMI~terapija,data=podatki)
```



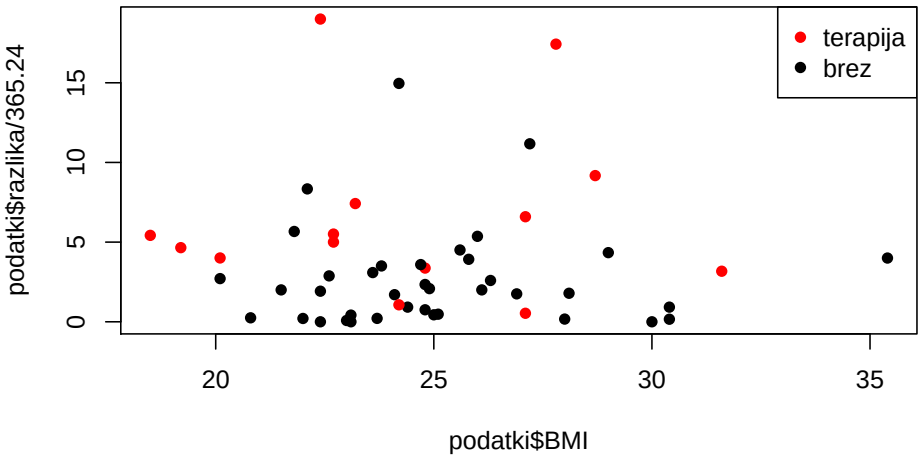

```
boxplot(formula=BMI~terapija,data=podatki,horizontal = TRUE,
        col="lightblue")
```



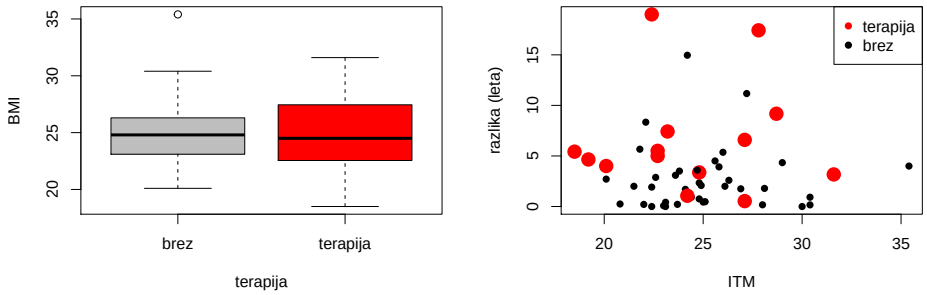

```
plot(podatki$BMI,podatki$razlika/365.24,xlab="ITT",
     ylab="razlika med okužbo in testiranjem (leta)",
     main="povezanost med ITT in časovno razliko")
```




```
plot(podatki$BMI,podatki$razlika/365.24,pch=16,col=podatki$terapija)
legend("topright",legend = c("terapija","brez"),col=c("red","black"),
       pch = c(16,16))
```




```
# plot 2 plots on the same plot
op = par(mfcol=c(1,2)) # two plots in a row
boxplot(formula=BMI~terapija,data=podatki,col=c("gray","red"))
plot(podatki$BMI,podatki$razlika/365.24,pch=16,
     col=podatki$terapija,cex=as.numeric(podatki$terapija),
     xlab="ITM",ylab="razlika (leta)")
legend("topright",legend = c("terapija","brez"),
       col=c("red","black"),pch = c(16,16))
```

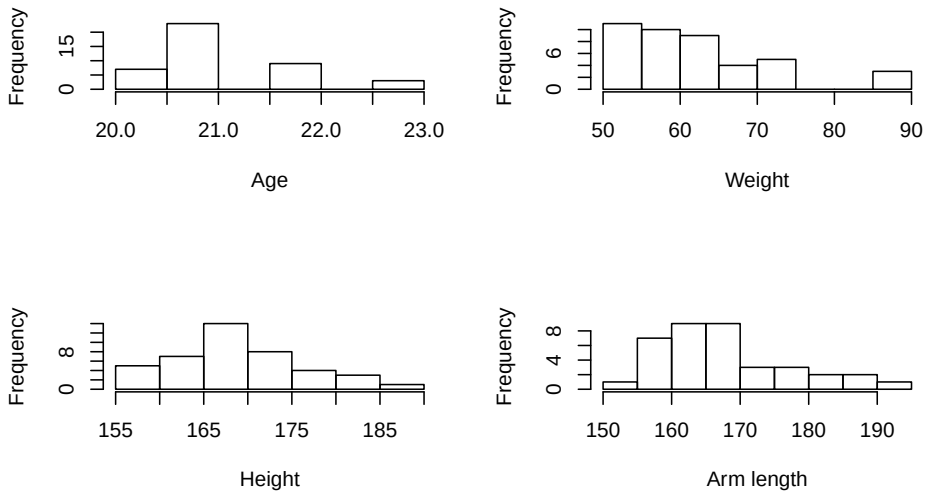



```
par(op)
```

Naloge

- Narišite histograme za 4 številske spremenljivke pri študentih (starost, visina, masa, roke).

```
op = par(mfcol=c(2,2)) # two plots in a row
hist(studenti$starost,main="",xlab="Age")
hist(studenti$visina,main="",xlab="Height")
hist(studenti$masa,main="",xlab="Weight")
hist(studenti$roke,main="",xlab="Arm length")
```

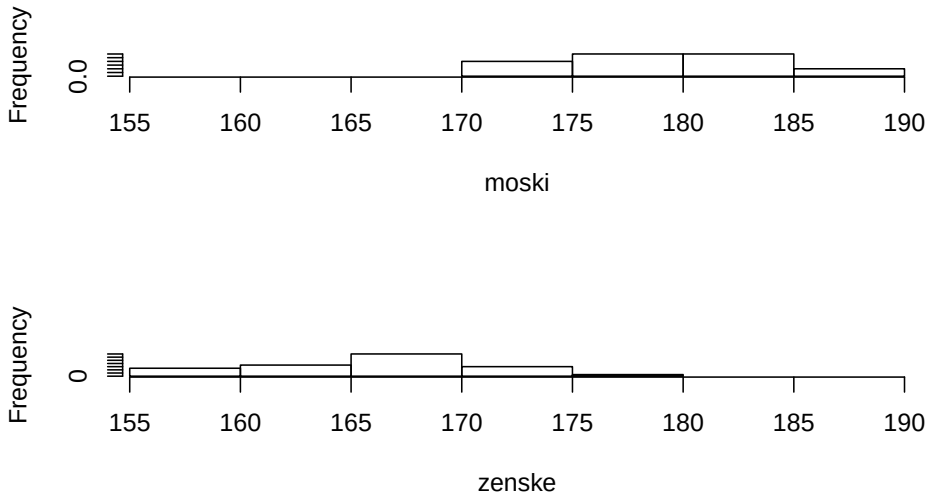



```
par(op)
```

- Narišite histograma za višino študentk in študentov posebej (enega pod drugim, da ju lahko primerjate).

```
spLim = min(moski,zenske)
zgLim = max(moski,zenske)
```

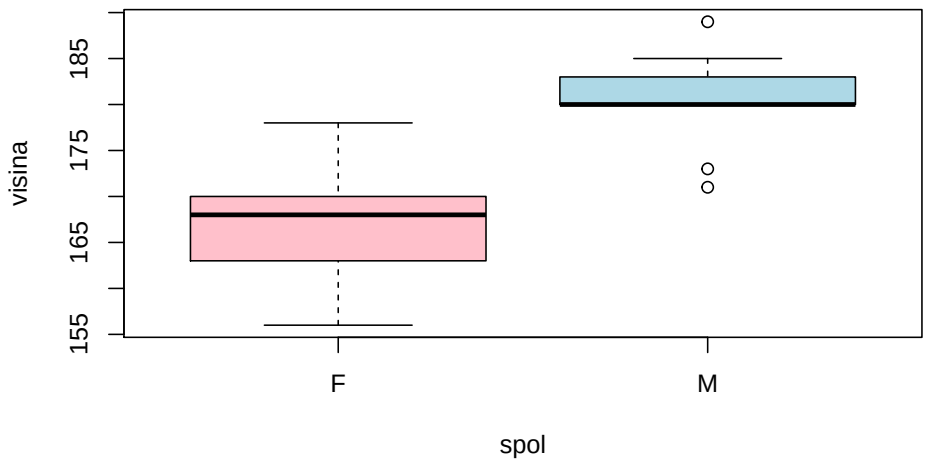
```
op = par(mfcol=c(2,1)) # two plots in a row
hist(moski,main="",xlim=c(spLim,zgLim))
hist(zenske,main="",xlim=c(spLim,zgLim))
```




```
par(op)
```

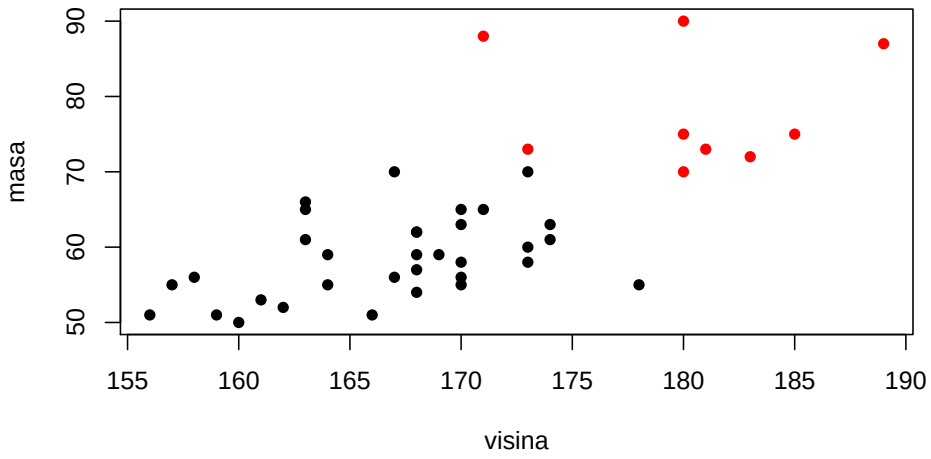
- Narišite porazdelitev višine študentk in študentov še z okvirjem z ročaji.

```
boxplot(visina~spol,data=studenti,col=c("pink","lightblue"))
```

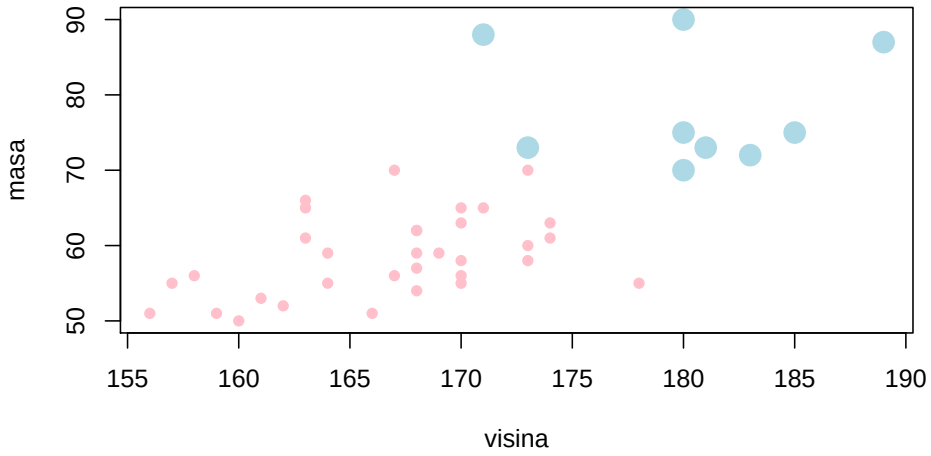


- Narišite povezanost med višino in težo. Označite spol.

```
plot(masa~visina,data=studenti,pch=16,col=spol)
```




```
barva = ifelse(studenti$spol=="F","pink","lightblue")
plot(masa~visina,data=studenti,pch=16,col=barva,cex=as.numeric(spol))
```

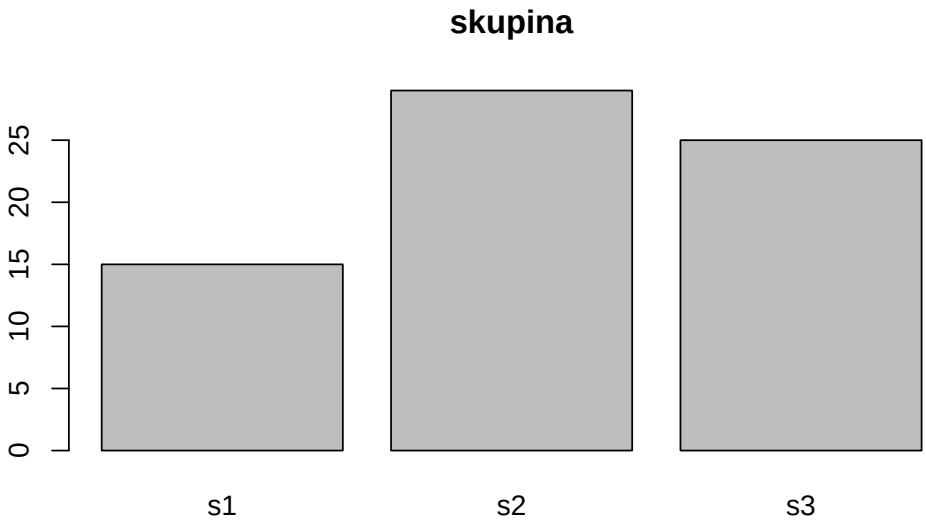


- Narišite povezanost med višino otrok in staršev. Označite spol.
- Narišite porazdelitev višine glede na velikost majice. Na sliki označite tudi povprečje vsake skupine (uporabite funkcijo `points` ali `abline`).
- Poglejte, kaj vse lahko določate pri izrisu s pomočjo `?par`.

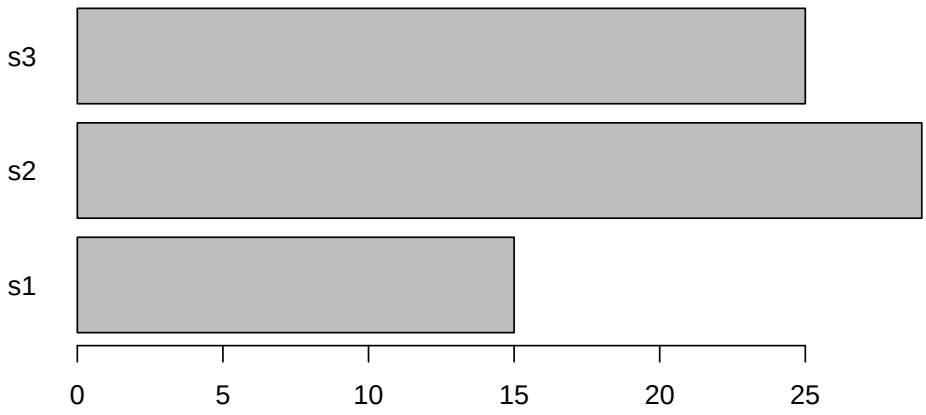
6.2 Opisne spremenljivke

Funkcije `barplot` za stolpični diagram, `pie` za okvir z ročaji, `mosaicplot`.

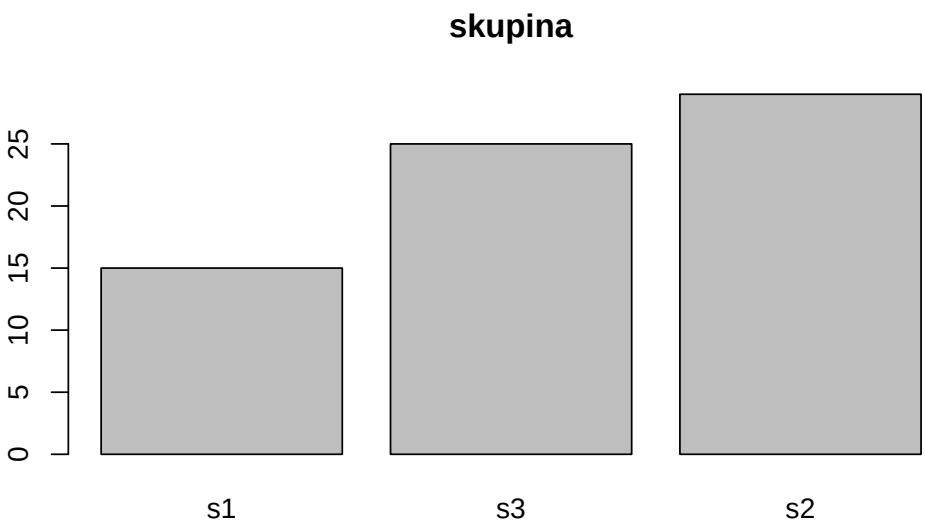
```
tabela = table(podatki$skupina)
barplot(tabela,main="skupina")
```



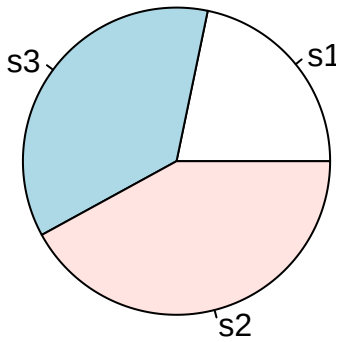

```
barplot(tabela,horiz=TRUE,las=1)
```




```
# ordered
podatki$skupina0 = reorder(podatki$skupina,podatki$skupina,length)
tabela = table(podatki$skupina0)
barplot(tabela,main="skupina")
```



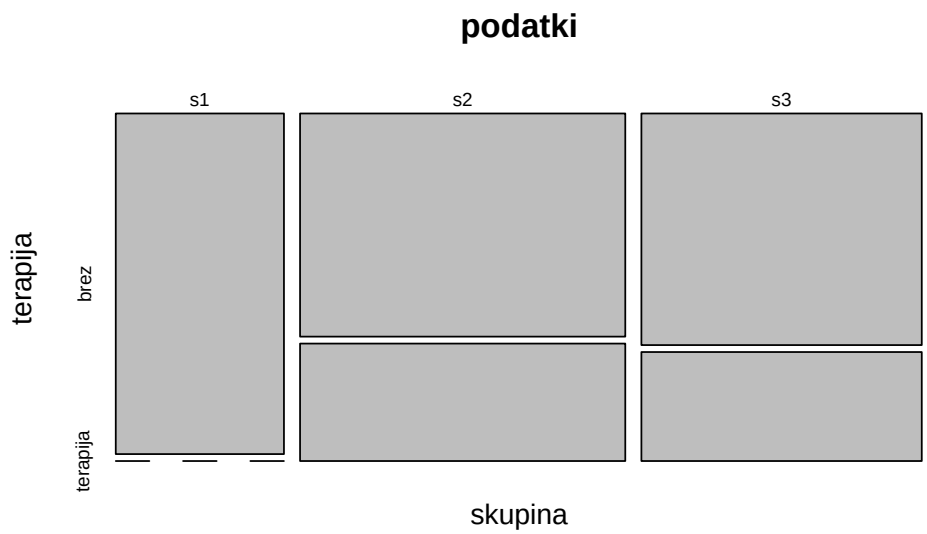

```
pie(tabela)
```




```
tabela2 = table(podatki$skupina,podatki$terapija)
tabela2
```

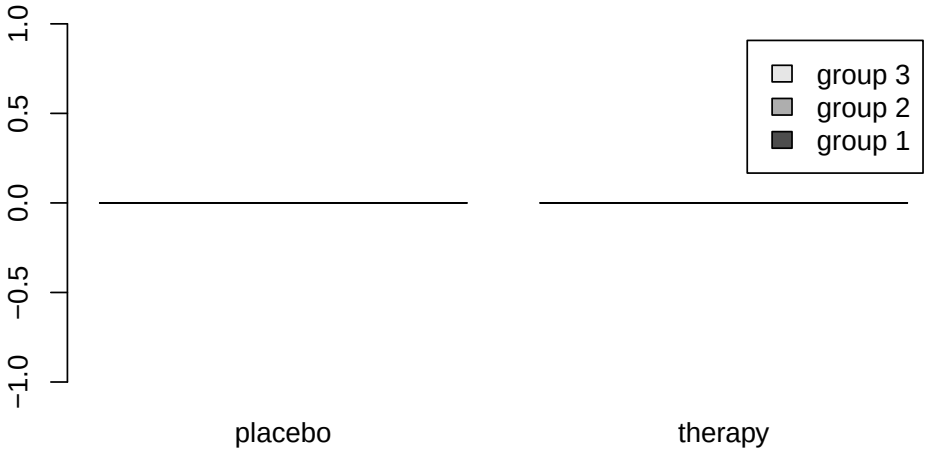
```
##
##      brez terapija
##   s1   15        0
##   s2   19       10
##   s3   17        8
```

```
mosaicplot(skupina~terapija,data=podatki)
```




```
barplot(tabela2,beside=TRUE,legend.text = TRUE)
```

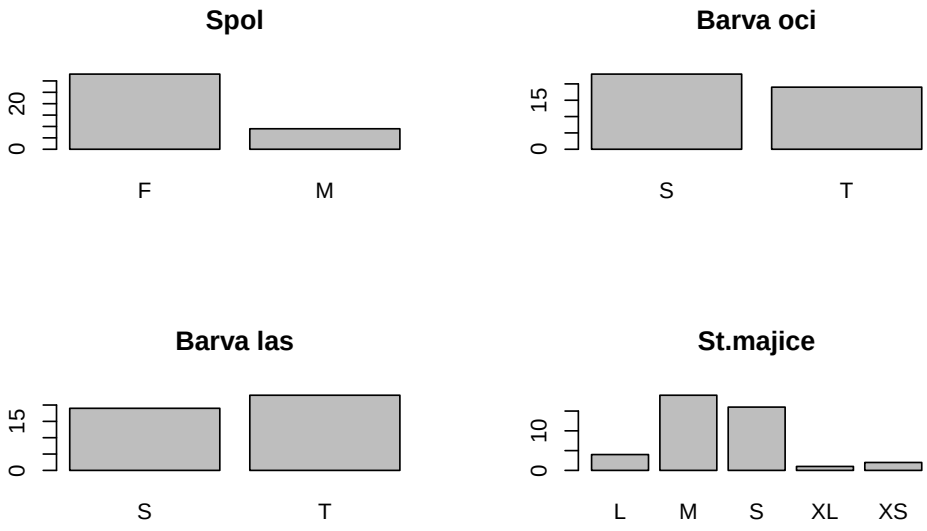

```
# use better names of values
podatki$skupinaN = factor(podatki$skupina,levels = 1:3,
                          labels=paste(rep("group",3),1:3))
podatki$terapijaN = factor(podatki$terapija,levels=0:1,
                           labels=c("placebo","therapy"))
tabela3 = table(podatki$skupinaN,podatki$terapijaN)
barplot(tabela3,beside=FALSE,legend.text = TRUE)
```



Naloge

- Prikažite kategorialne podatke pri študentih na najboljši možni grafični način.

```
op = par(mfcol=c(2,2))
barplot(table(studenti$spol),main = "Spol")
barplot(table(studenti$lasje),main="Barva las")
barplot(table(studenti$oci),main="Barva oci")
barplot(table(studenti$majica),main="St.majice")
```

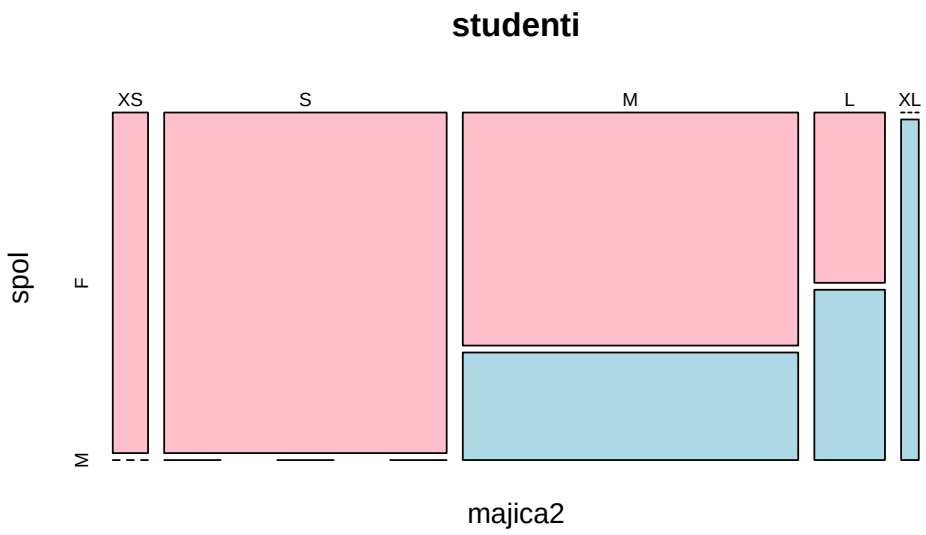



```
par(op)
```

- Z `mosaicplot` prikažite odvisnost spola od velikosti majice. Uporabite

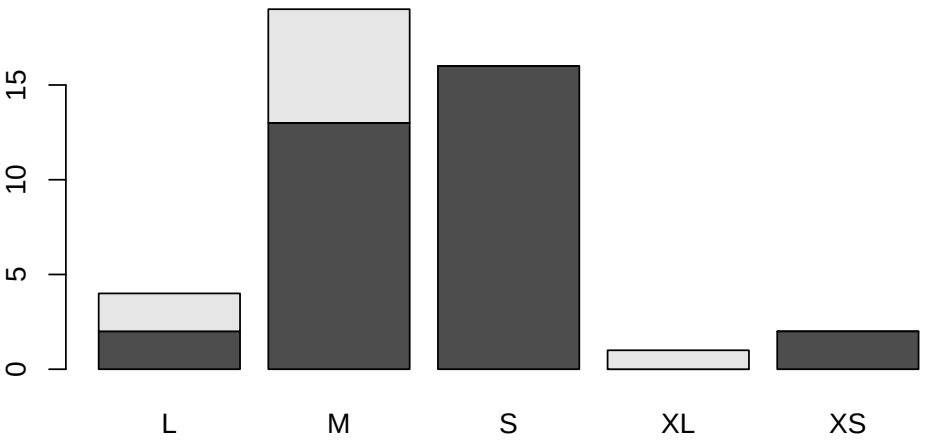
barve za intuicijo.

```
mosaicplot(majica2~spol,data=studenti,col=c("pink","lightblue"))
```

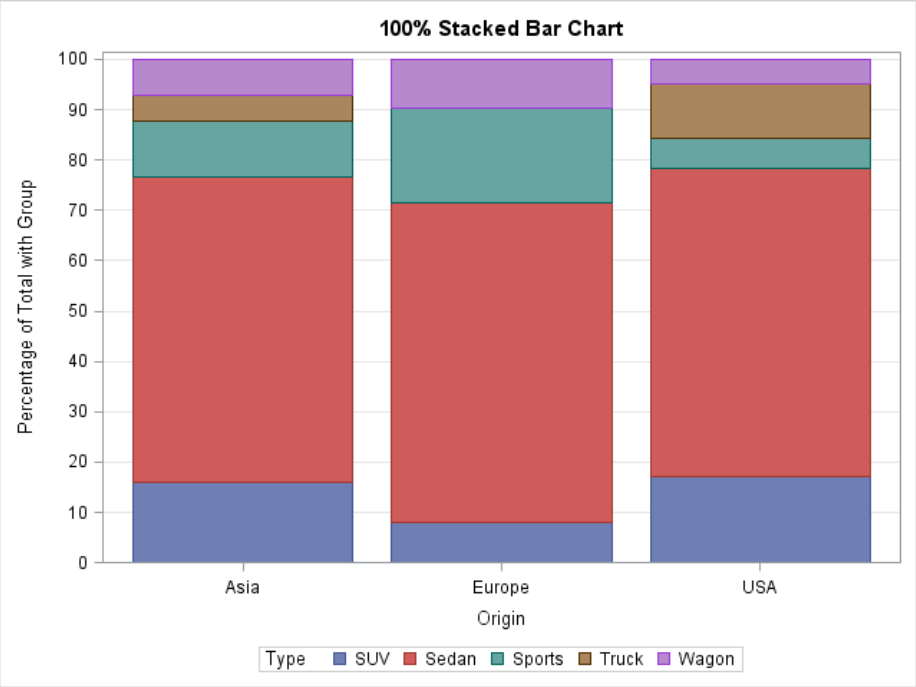


- Narišite stolpični diagram, spol/velikost majice, ki bo prikazoval deleže ljudi posameznega spola, ki nosijo določeno velikost majice. Poglejte npr. primer prodaja različnih tipov avtomobilov po kontinentih):

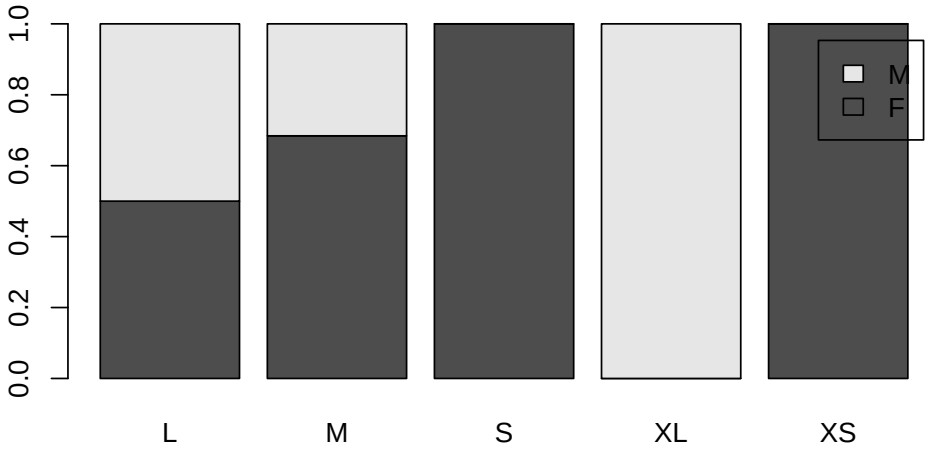
```
barplot(table(studenti$spol,studenti$majica))
```




```
a = table(studenti$spol,studenti$majica)
as = a[1,] + a[2,]
aDelez = t(t(a)/as)
barplot(aDelez,legend.text = TRUE)
```

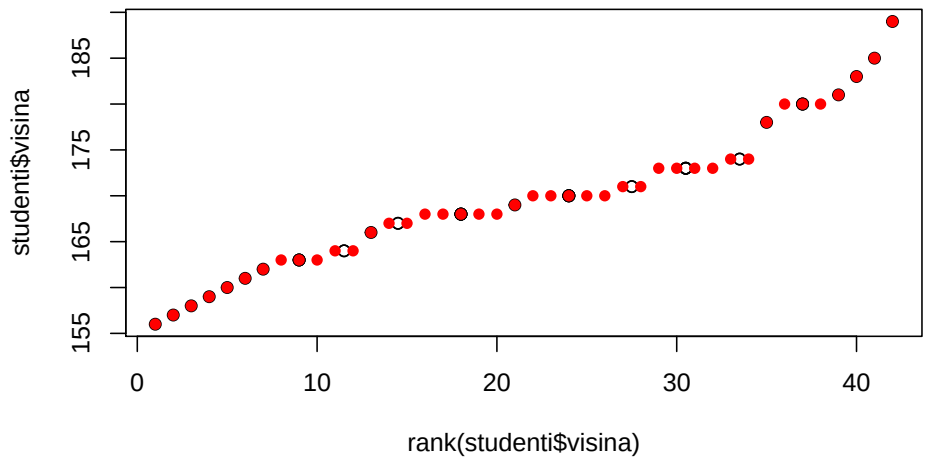


Slika 6.1: Stacked bar, percentages



- S pomočjo uporabe funkcije `rank()` dobite ranke višin študentov in jih prikažite na razsevnem grafikonu (x = rank, y= visina). Čez ta graf narišite še urejene višine (funkcija `sort()`). Kje je razlika?

```
plot(rank(studenti$visina),studenti$visina)
points(1:length(studenti$visina),sort(studenti$visina),col="red",pch=16)
```



Poglavje 7

Porazdelitve

7.1 Kategorialne porazdelitve

Funkcija `sample`. Glede na spodnjo tabelo bi želeli simulirati vzorec naslednjih 30 kupcev sladoledov.

Flavor	Number of Customers
Chocolate	16
Strawberry	5
Vanilla	9

```
set.seed(2019) # set random seed to enable exact repetition
vzorec1 = sample(x = c("Chocolate","Strawberry","Vanilla"),
                 size =30,replace=TRUE,prob=c(16,5,9))
vzorec1
```

```
##  [1] "Vanilla"    "Vanilla"    "Chocolate"  "Vanilla"    "Chocolate"
##  [6] "Chocolate"  "Vanilla"    "Chocolate"  "Chocolate"  "Vanilla"
## [11] "Vanilla"    "Vanilla"    "Chocolate"  "Chocolate"  "Vanilla"
## [16] "Vanilla"    "Chocolate"  "Vanilla"    "Chocolate"  "Chocolate"
## [21] "Chocolate"  "Vanilla"    "Strawberry" "Chocolate"  "Chocolate"
## [26] "Chocolate"  "Vanilla"    "Strawberry" "Vanilla"    "Chocolate"
```

```
table(vzorec1)
```

```
## vzorec1
##  Chocolate Strawberry    Vanilla
##         15          2         13
```

```r
# replace = TRUE for permutations
vzorec2 = sample(x = 1:20,size =20,replace=TRUE)
vzorec2
```

```
##  [1]  5  1 14  7  6  6 17 16  6  5 17  6 15 14 12 14 17 16  2 17
```

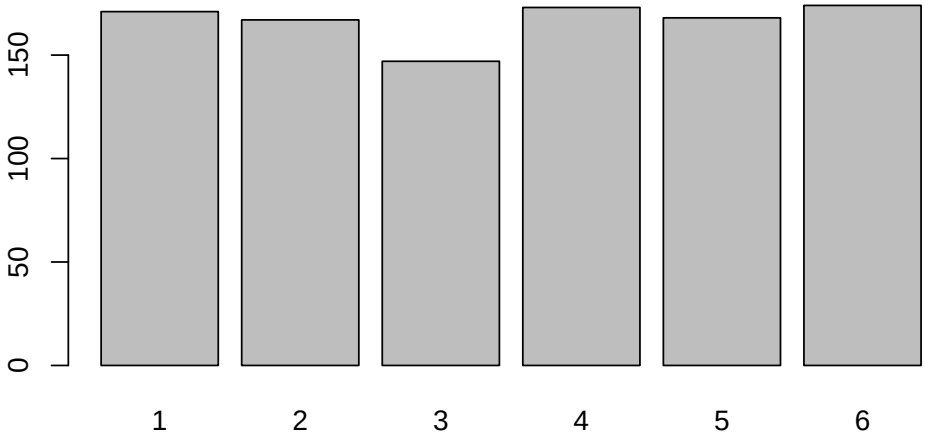
```r
# or for sampling without replacement
vzorec3 = sample(x = 1:20,size =10,replace=TRUE)
vzorec3
```

```
##  [1] 13  1  9 12 15  9 16 19 17  8
```

Naloge

- S pomočjo `sample` pridobite rezultate 1000 metov kocke.
  - Prikažite vzorec grafično.
  - Kako je porazdeljena spremenljivka, katere 1000 realizacij ste naredili?
  - Izračunajte delež padlih šestic.
  - Zanima nas samo, ali je v enem metu padla šestica. Spremenite vaš vzorec, da boste dobili ta podatek. Novi vzorec spet narišite. Kako je porazdeljena spremenljivka?

```r
vzorec = sample(x=1:6,size=1000,replace=TRUE)
barplot(table(vzorec))
```



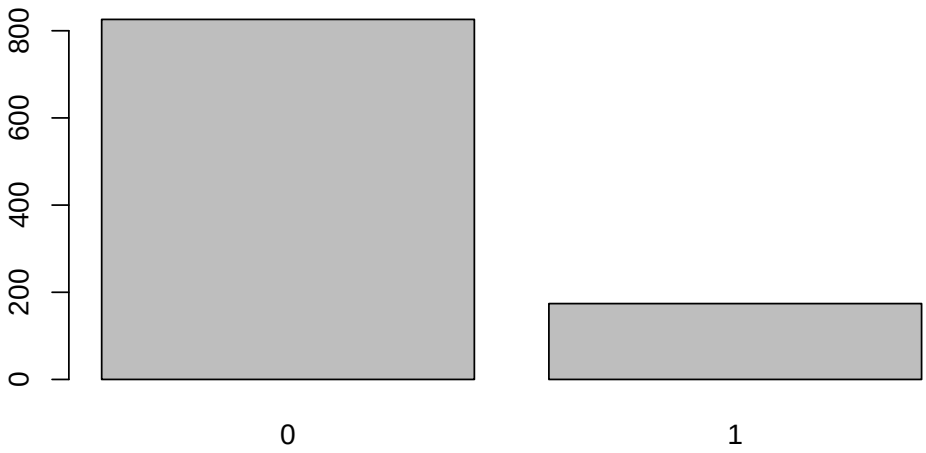

```r
# delez sestic
sum(vzorec == 6)/length(vzorec)
```

```
## [1] 0.174
```

```r
mean(vzorec == 6)
```

```
## [1] 0.174
```

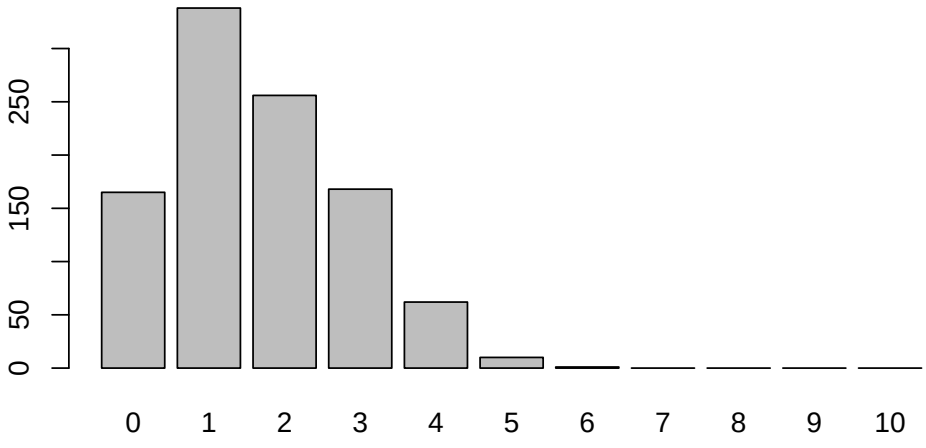
```
vzorec6 = ifelse(vzorec==6,1,0)
barplot(table(vzorec6)) # Bernoulli(1/6)
```



- S pomočjo `sample` pridobite 1000 opazovanj, kjer nas zanima število padlih šestic v desetih metih kocke.
  - Prikažite vzorec grafično.
  - Kako je porazdeljena spremenljivka, katere realizacija je *število šestic v 10 metih*?

```
vzorec = NULL
for(i in 1:1000){
  # stevilo 6 v 10 metih
  st10 = sum(sample(c(0,1),size=10,replace=TRUE,prob=c(5/6,1/6)))
  vzorec[i] = st10
}

barplot(table(factor(vzorec,levels=0:10)))
```

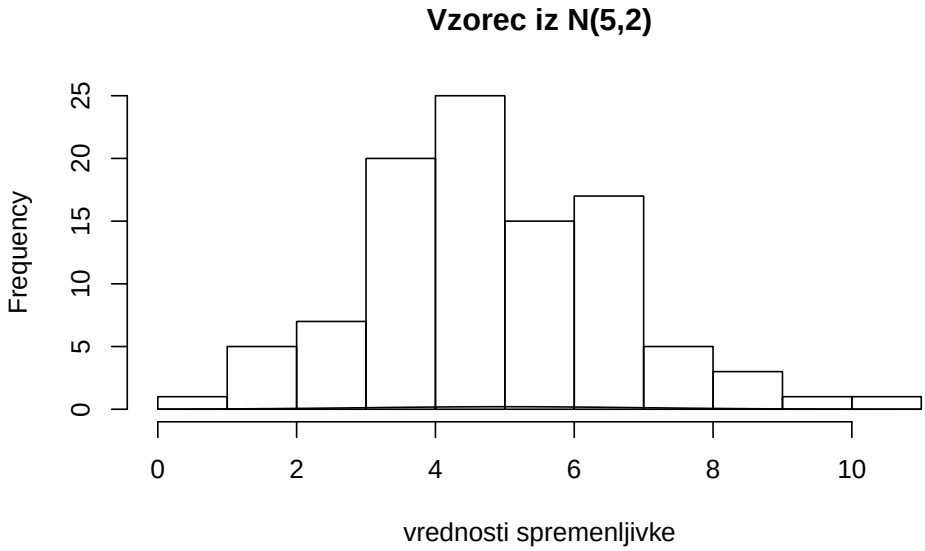


7.2 Distributions

Poglejte si pomoč za `Distributions`. V osnovnem R je veliko funkcij za generiranje/pridobivanje vrednosti, povezanih s porazdelitvami. Pomembno:

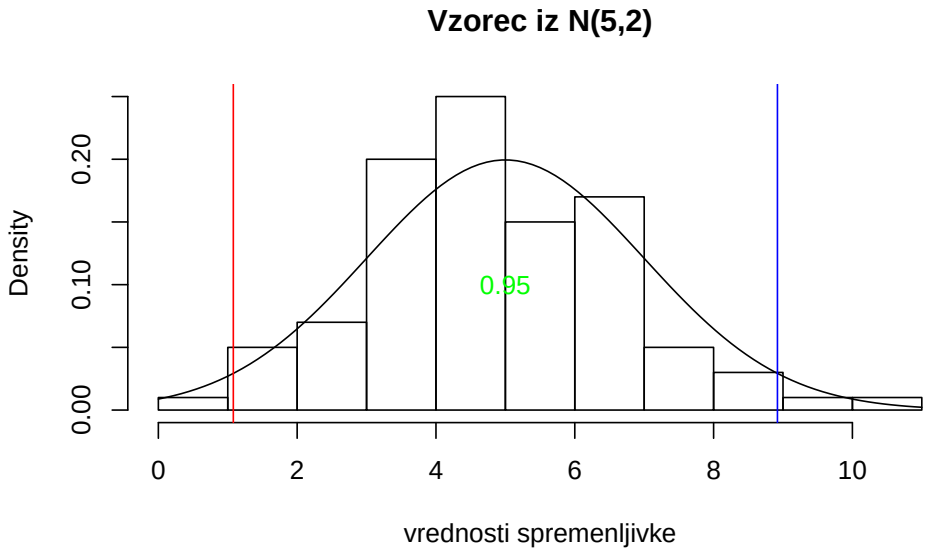
The functions for the density/mass function, cumulative distribution function, quantile function and random variate generation are named in the form dxxx, pxxx, qxxx and rxxx respectively.

```
set.seed(2019) # set random seed to enable exact repetition
povp=5
stdo = 2
# random sample
vzorecN = rnorm(100,mean=povp,sd=stdo)
hist(vzorecN,xlab="vrednosti spremenljivke",
     main=paste0("Vzorec iz N(",povp,",",stdo,")"))
curve(dnorm(x,mean=povp,sd=stdo),add=TRUE)
```




```
# density plot
hist(vzorecN,xlab="vrednosti spremenljivke",
     main=paste0("Vzorec iz N(",povp,",",stdo,")"),freq=FALSE)
curve(dnorm(x,mean=povp,sd=stdo),add=TRUE)
# quantile values
limit025 = qnorm(0.025,mean=povp,sd=stdo)
limit975 = qnorm(0.975,mean=povp,sd=stdo)
abline(v = limit025,col="red")
abline(v = limit975,col="blue")
# probabilities
probabIn = pnorm(limit975,mean=povp,sd=stdo) -
```

```
  pnorm(limit025,mean=povp,sd=stdo)
text(5,0.1,labels = probabIn,col="green")
```



Naloge

Glej naloge pod poglavjem Tabele!

Poglavje 8

Funkcije apply, sapply, replicate

R je programski jezik, kjer je računanje z vektorji zelo zaželjeno. Zato obstajajo funkcije, ki omogočijo, da neko (netrivialno) funkcijo izvedemo na celem vektorju naenkrat (brez pisanja zank).

```
?apply
n = 10
# apply
matrika = matrix(runif(n),nrow=10)
matrika = cbind(matrika,runif(n,min=-1,max=1),rnorm(n))
apply(matrika,2,mean) # calculate mean column-wise
```

```
## [1]  0.50834082  0.01561734 -0.56157981
```

```
apply(matrika,2,sd)
```

```
## [1] 0.3291284 0.5828940 0.8513620
```

```
# replicate (instead of for loop)
perms = replicate(10,sample(1:20,size=20,replace=FALSE))
# get 10 permutations
dim(perms)
```

```
## [1] 20 10
```

```
# sapply
data(iris)
irisL = list(dolzina = iris$Petal.Length,sirina = iris$Petal.Width)
sapply(irisL,FUN=quantile)
```

```
##      dolzina sirina
```

```
## 0%      1.00    0.1
## 25%     1.60    0.3
## 50%     4.35    1.3
## 75%     5.10    1.8
## 100%    6.90    2.5
```

```
tabela = sapply(irisL,FUN=quantile)
```

Naloge

Glej naloge pod poglavjem Tabele!

Poglavje 9

Tabele v .Rmd

Uporaba funkcije `kable` iz knjižnice `knitr`. Za lepše tabele lahko uporabljate tudi knjižnico `kableExtra` (npr. spletna stran)

```
library(knitr)
kable(tabela)
```

	dolzina	sirina
0%	1.00	0.1
25%	1.60	0.3
50%	4.35	1.3
75%	5.10	1.8
100%	6.90	2.5

```
kable(tabela,caption="Naslov tabele.")
```

```
tabela2 = apply(iris[1:4],2,quantile)
colnames(tabela2) = rep(c("dolžina","širina"),2)
library(kableExtra)
kable(tabela2) %>%
  kable_styling("striped",full_width = F) %>%
  add_header_above(c("","venčni listi" = 2, "časni listi" = 2))
```

	venčni listi		časni listi	
	dolžina	širina	dolžina	širina
0%	4.3	2.0	1.00	0.1
25%	5.1	2.8	1.60	0.3
50%	5.8	3.0	4.35	1.3
75%	6.4	3.3	5.10	1.8
100%	7.9	4.4	6.90	2.5

Tabela 9.1: Naslov tabele.

	dolzina	sirina
0%	1.00	0.1
25%	1.60	0.3
50%	4.35	1.3
75%	5.10	1.8
100%	6.90	2.5

Naloge

- Pridobite 1000 opazovanj števila padlih petic v desetih metih kocke. Uporabite eno izmed porazdelitvenih funkcij.
  - Predstavite vzorec z grafičnim prikazom.
  - Izračunajte teoretične verjetnosti za vsako padlo število petic v 10 metih. S pomočjo verjetnosti izračunajte še pričakovano število opazovanj z določenim številom petic.
  - Prikažite teoretične in opazovane deleže v tabeli.
  - Čez grafični prikaz iz prejšnje točke dodajte frekvenčni poligon teoretične porazdelitve (gl. prejšnjo točko - dodajte pričakovano število opazovanj za vsako dobljeno število petic).

```
vzorec = rbinom(1000,size=10,prob=1/6)
hist(vzorec,breaks = seq(from=-0.25,to=10.25,by=0.5),col="gray")
#teoreticno
pbinom(0:10,size=10,prob=1/6)
```

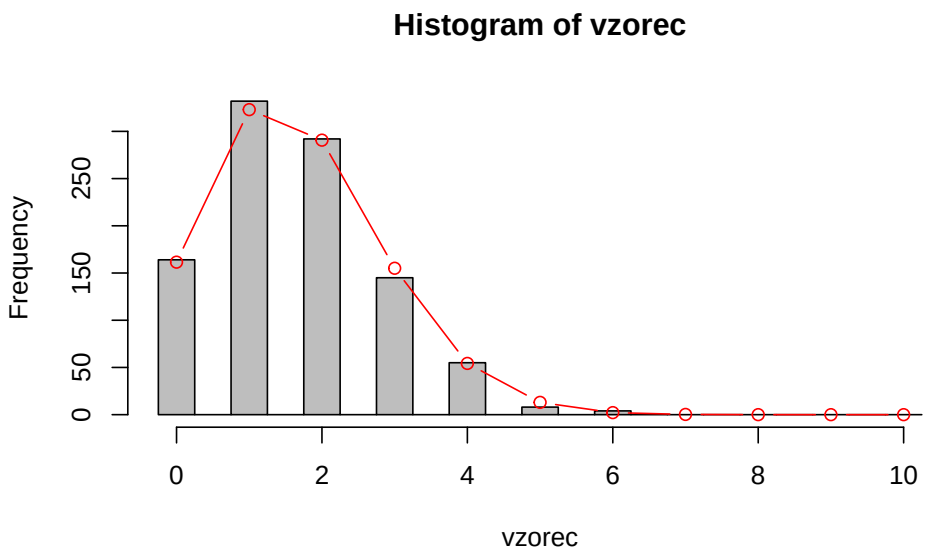
```
##  [1] 0.1615056 0.4845167 0.7752268 0.9302722 0.9845380 0.9975618 0.9997325
##  [8] 0.9999806 0.9999992 1.0000000 1.0000000
```

```
teorV = dbinom(0:10,size=10,prob=1/6)
opFrek = table(factor(vzorec,levels=0:10)) #use opazovane frekvence
opDelez = opFrek/1000 #usi opazovani delezi

# izpis tabele
tabela = rbind(teorV,opDelez)
kable(tabela,digits=3,format="markdown")
```

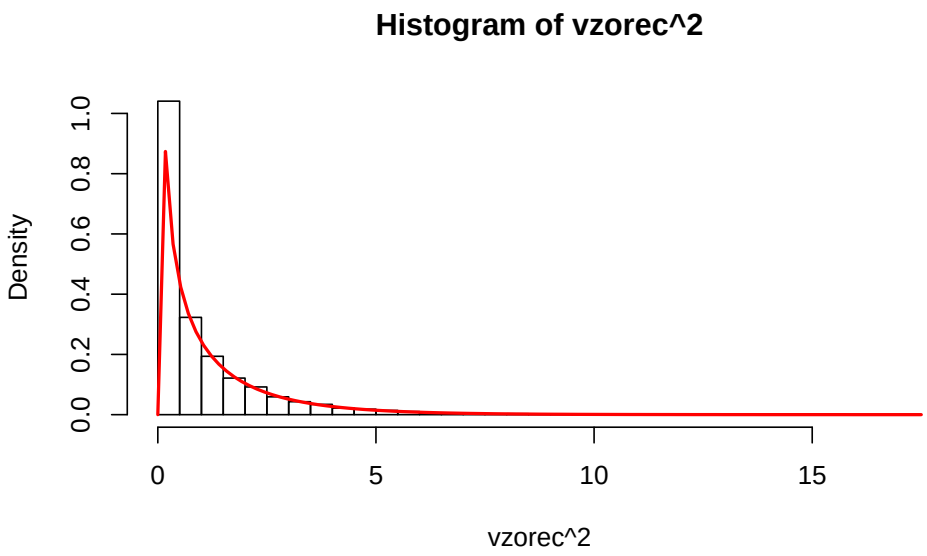
	0	1	2	3	4	5	6	7	8	9	10
teorV	0.162	0.323	0.291	0.155	0.054	0.013	0.002	0	0	0	0
opDelez	0.164	0.332	0.292	0.145	0.055	0.008	0.004	0	0	0	0

```
hist(vzorec,breaks = seq(from=-0.25,to=10.25,by=0.5),col="gray")
pricakVred = teorV * 1000 # pi * n
lines(0:10,pricakVred,type="b",col="red")
```

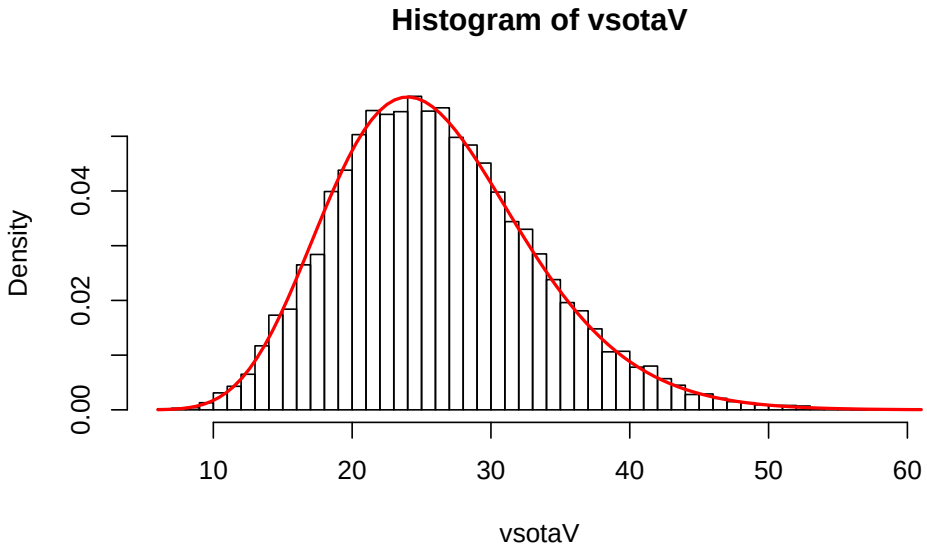


- Generirajte velik vzorec n = 10000 iz standardne normalne porazdelitve.
  - Predstavite grafično kvadrirane vrednosti.
  - Po kateri porazdelitvi je porazdeljena kvadrirana standardno normalna spremenljivka?
  - Grafično se prepričajte, da je to res (narišite krivuljo čez grafični prikaz).
  - Grafično pokažite, da velja, da je vsota kvadriranih k neodvisnih standardno normalno porazdeljenih spremenljivk porazdeljena po porazdelitvi χ^2_k.
    - Rezultate izračunajte s pomočjo zanke.
    - Za izračune uporabite `replicate` in `apply`.
  - Vsota dveh neodvisnih porazdelitev χ^2 je spet porazdeljena po χ^2 porazdelitvi s stopinjami prostosti, ki so seštevek stopinj prostosti osnovnih dveh porazdelitev χ^2. Preverite s tem, da napišete funkcijo in lahko poizkusite za različne stopinje prostosti.
  - Kaj pa razlika? Je razlika spet porazdeljena po porazdelitvi χ^2? Zakaj da/ne?

```
vzorec=rnorm(10000)
hist(vzorec^2,freq=FALSE,breaks=50)
curve(dchisq(x,df=1),add=TRUE,col="red",lwd=2)
```




```
k = 26
vsotaV = rep(0,10000)
for(i in 1:k){
  vzorec=rnorm(10000)
  vsotaV = vsotaV + vzorec^2
}
hist(vsotaV,freq=FALSE,breaks=50)
curve(dchisq(x,df=k),add=TRUE,col="red",lwd=2)
```




```
# brez zank
vzorec = replicate(k,rnorm(10000))
```

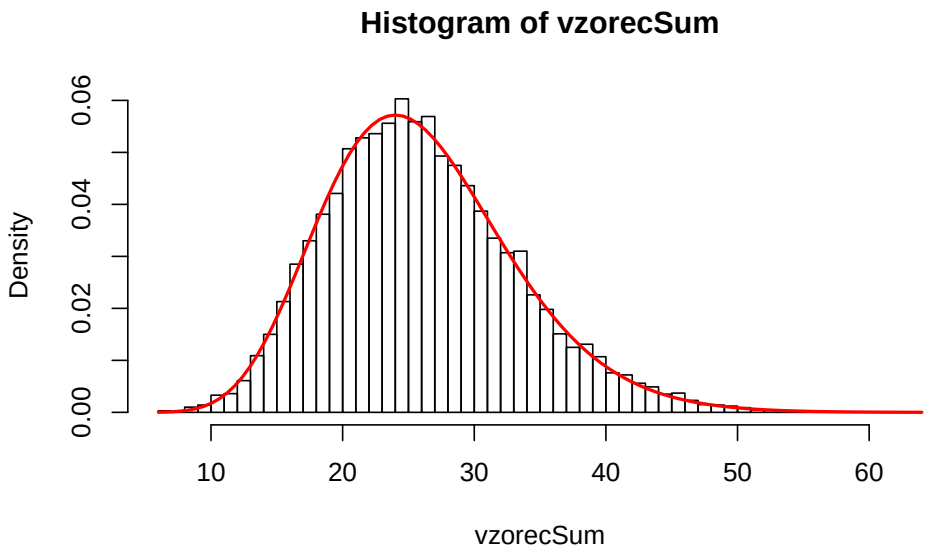
```r
dim(vzorec)
```

```
## [1] 10000    26
```

```r
vzorec2 = vzorec^2
vzorecSum = apply(vzorec2,1,sum)
length(vzorecSum)
```

```
## [1] 10000
```

```r
hist(vzorecSum,freq=FALSE,breaks=50)
curve(dchisq(x,df=k),add=TRUE,col="red",lwd=2)
```



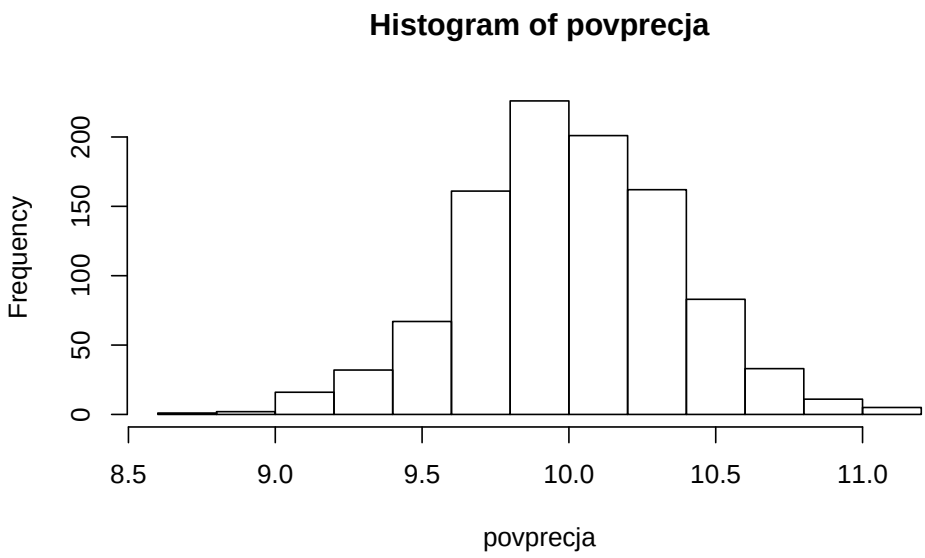

```r
vsotaChi2 = function(df1,df2,n){
  vz1 = rchisq(n,df=df1)
  vz2 = rchisq(n,df=df2)
  hist(vz1-vz2,freq=FALSE,breaks=50)
  curve(dchisq(x,df=df1-df2),add=TRUE,col="red",lwd=2)
}
vsotaChi2(df1 = 4,df2 = 1,n = 10000)
```

```
![](_main_files/figure-latex/unnamed-chunk-59-1.pdf)<!-- -->
```

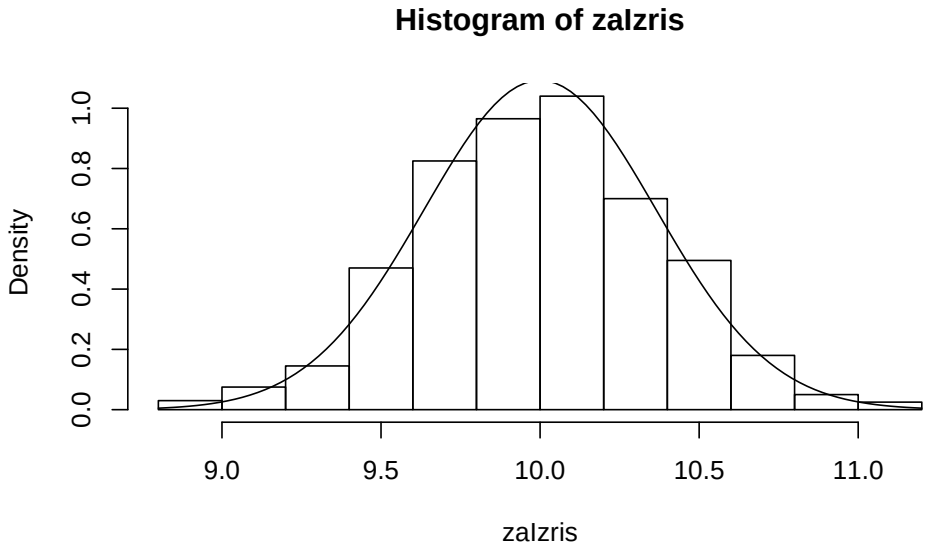
- Simulirajte 1000 vzorčnih povprečij velikosti n = 30 iz N(10,2) (temperatura vode iz pipe).
  - Narišite dobljen vzorec vzorčnih povprečij.
    - Rezultate izračunajte s pomočjo zanke.
    - Za izračune uporabite `replicate` in `apply`.

– Čezenj narišite porazdelitev, ki bi po centralnem limitnem izreku morala teoretično veljati za to spremenljivko.

```
povprecja = NULL
for(i in 1:1000){
  vzorec = rnorm(n=30,mean=10,sd=2)
  povprecja[i] = mean(vzorec)
}
hist(povprecja)
```




```
povpN = function(n,mean,sd){
  return(mean(rnorm(n=n,mean=mean,sd=sd)))
}
zaIzris = replicate(1000,povpN(n=30,mean=10,sd=2))
hist(zaIzris,freq=FALSE)
curve(dnorm(x,mean=10,sd=2/sqrt(30)),add=TRUE)
```



- S simulacijami iz normalne porazdelitve pokažite, da je izračun variance vzorca na način

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

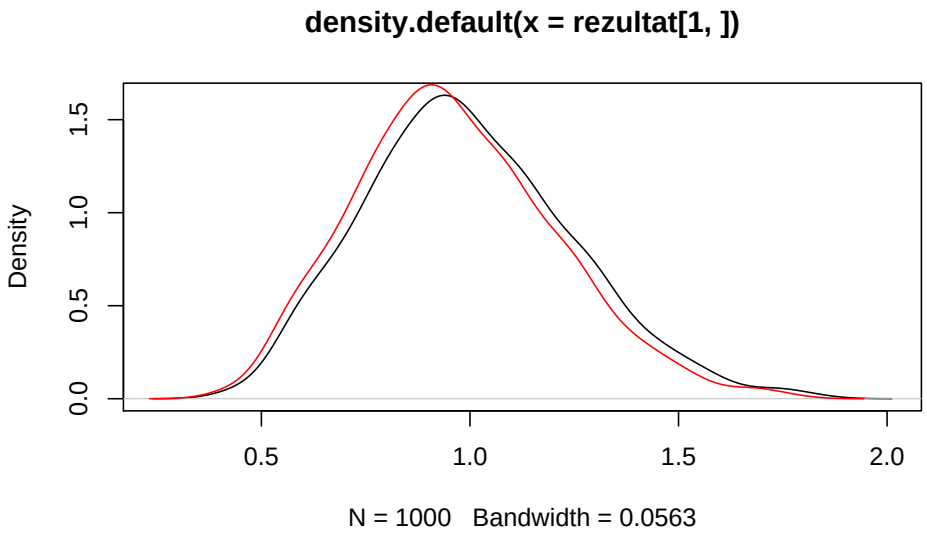
slabši približek prave (teoretične) variance kot vzorčna varianca s^2, ki jo v R dobimo s funkcijo `var`.
  - naredite funkcijo za zgoraj izraženo varianco
  - simulirajte velik vzorec obeh varianc
  - narišite vzorca varianc (uporabite `plot(density(x))`)
  - izračunajte povprečji, mediani varianc
  - izračunajte delež varianc pod teoretično vrednostjo
  - rezutate predstavite s tabelo (funkcija `kable`, `format = "markdown"`)

```
varianca = function(x){
 n = length(x)
 return(var(x)*(n-1)/n)
}

obeVar = function(n){
 vzorec = rnorm(n)
 return(c(var(vzorec),varianca(vzorec)))
}
rezultat = replicate(1000,obeVar(30))
dim(rezultat)
```

```
## [1]    2 1000
```

```
plot(density(rezultat[1,]))
lines(density(rezultat[2,]),col="red")
```




```
# povprecji
m1 = mean(rezultat[1,])
m2 = mean(rezultat[2,])
# mediani
me1 = median(rezultat[1,])
me2 = median(rezultat[2,])
# delez pod teoreticno vrednostjo
d1 = mean(rezultat[1,]<1)
d2 = mean(rezultat[2,]<1)

tabela = cbind(c(m1,m2),c(me1,me2),c(d1,d2))
colnames(tabela) = c("povprecje","mediana","delez pod 1")
rownames(tabela) = c("var iz R","nasa varianca")
kable(tabela,digits = 3,format="markdown")
```

	povprecje	mediana	delez pod 1
var iz R	0.998	0.978	0.537
nasa varianca	0.964	0.945	0.598

Poglavje 10

Statistični projekt - nabiranje biserov

Japonska nabiralka biserov *ama* se v sezoni potaplja vsak dan. Verjetnost, da na en potop dobi biser, je 20% (in zanemarimo možnost, da ama v enem potopu dobi več kot en biser).

1. Izračunajte verjetnost, da bo *ama* v največ 350 potopih našla 80 biserov.

   - Uporabite negativno binomsko porazdelitev v R $NegBinom(n, p)$, ki meri število *neuspehov* na n *uspehov*. Za to porazdelitev velja naslednje. Parametra sta n - število *uspehov* in p verjetnost za *uspeh*. Računamo torej verjetnost, da se vmes (med n *uspehi*) zgodi x *neuspehov*

     $$P(X = x) = \binom{n + x - 1}{n - 1} p^n (1 - p)^x.$$

     V zadnjem poskusu se bo *uspeh* nujno zgodil.

   - Kaj bo v našem primeru *uspeh* in kaj *neuspeh*?

   - Narišite teoretično verjetnostno porazdelitev **števila potopov** za *amo*, ki nabere 80 biserov.

   - Označite grafično, kako velika bo verjetnost za $P(st.potopov \leq 350)$.

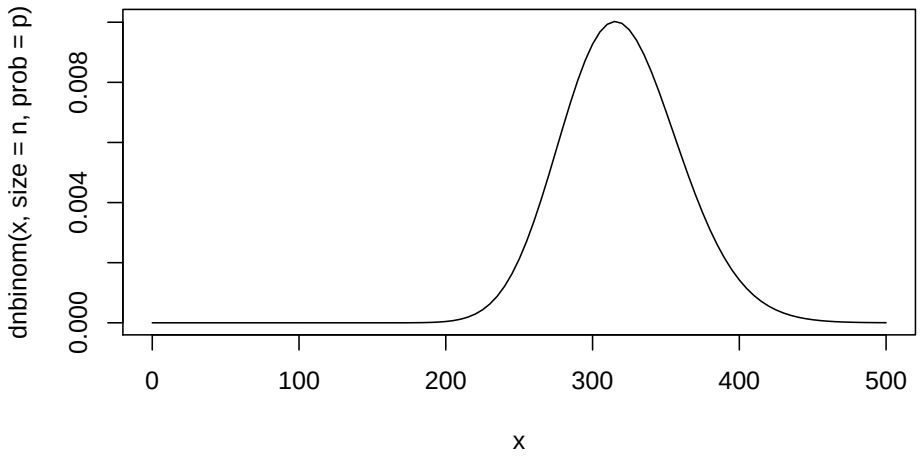
   - Ugotovite, kakšna je pričakovana vrednost in kakšna varianca negativne binomske porazdelitve za 80 nabranih biserov (`help(dnbinom)`)?

   - Kaj nam ta pričakovana vrednost govori? Kakšna je pričakovano število potopov za 80 biserov?

   - Na podlagi zgornje alineje - izračunajte z normalno aproksimacijo, kakšna je verjetnost za največ 350 potopov pri 80 nabranih biserih.

- Izračunajte še verjetnost na podlagi negativne binomske porazdelitve.

```
potop = 350
n = 80
p = 0.2
# neuspeh - brez bisera, uspeh - biser, n=80, p=0.2
curve(dnbinom(x,size=n,prob=p),from=0,to=500)
```




```
# teoretična porazdelitev NB za st. neuspehov
# stevilo potopov (+80)
teor = dnbinom(0:500,size=n,prob=p)
plot(80:580,teor,type="p")
curve(dnbinom(x-n,size=n,prob=p),from=80, to=580,add=TRUE,
      col="red",lwd=4)
abline(v=350,col="blue")
pnbinom(350-n,size=n,prob=p)
```

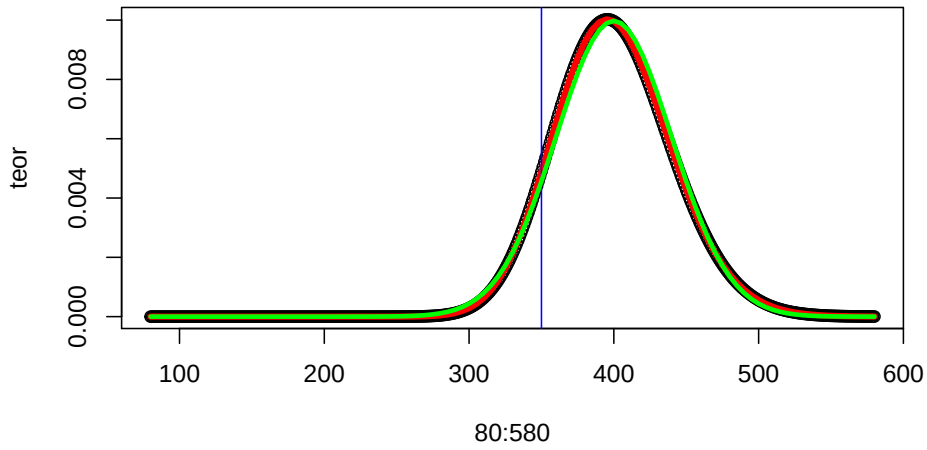
```
## [1] 0.1034488
```

```
# pricakovana vrednost in varianca
pricakovana = n*(1-p)/p # pricakovano st. neuspehov
pricStPotopov = pricakovana +n
varianca = n*(1-p)/p^2

# norm. aprox.
pnorm(350,mean=pricStPotopov,sd=sqrt(varianca))
```

```
## [1] 0.1056498
```

```
curve(dnorm(x,mean=pricStPotopov,sd=sqrt(varianca)),
      add=TRUE,col="green",lwd=3)
```



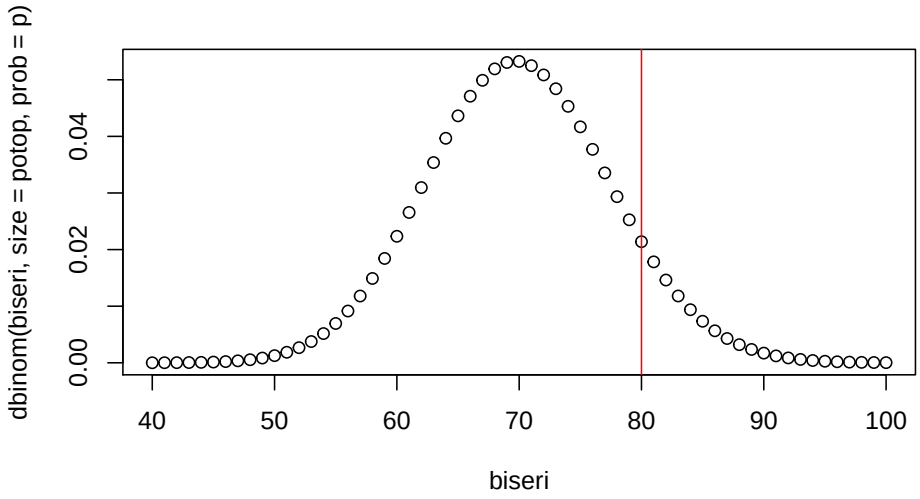

```
# neg.binomska
pnbinom(350-n,size=n,prob=p)
```

```
## [1] 0.1034488
```

2. Predstavite porazdelitev števila biserov v 350 potopih.

   - Z binomsko porazdelitvijo grafično predstavite porazdelitev števila biserov v 350 potopih. Omejite se na tisti del grafa, kjer so verjetnosti relativno velike.
   - Označite grafično, kako verjetno je, da v 350 potopih ama nabere najmanj 80 biserov.
   - Izračunajte teoretično verjetnost, da je v 350 potopih nabrano najmanj 80 biserov.
   - Simulirajte porazdelitev števila biserov brez eksplicitne uporabe zank:
     - število vzorcev naj bo 1000
     - narišite histogram in pravilno označite osi
     - čez histogram narišite teoretični frekvenčni poligon
     - čez narišite še aproksimativno zvezno normalno porazdelitev

```
# točka 2
biseri = 40:100
plot(biseri,dbinom(biseri,size=potop,prob = p),type="b")
abline(v=n,col="red")
```




```
sum(dbinom(80:350,size=350,prob=p))
```

```
## [1] 0.1034488
```

```
pbinom(n-1,size=350,prob=p,lower.tail=FALSE)
```

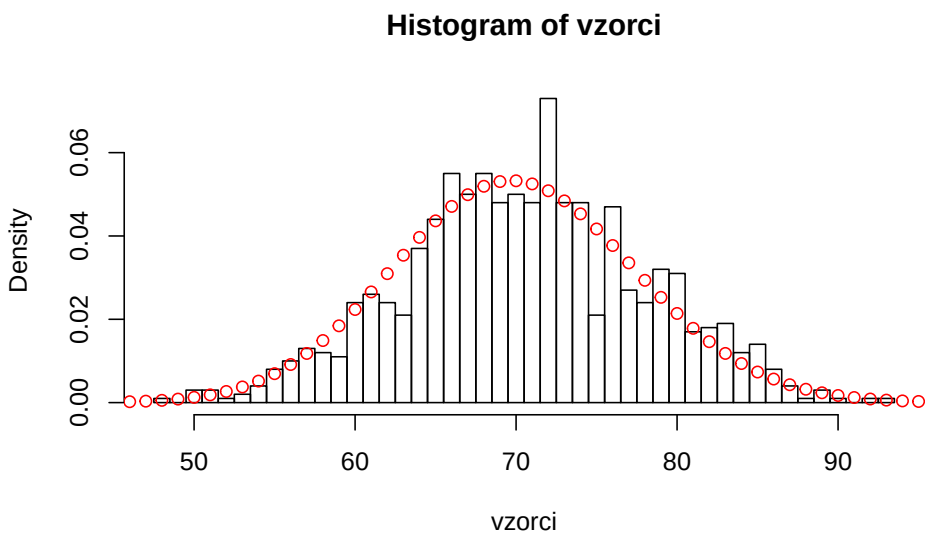
```
## [1] 0.1034488
```

```
1-pbinom(n-1,size=350,prob=p)
```

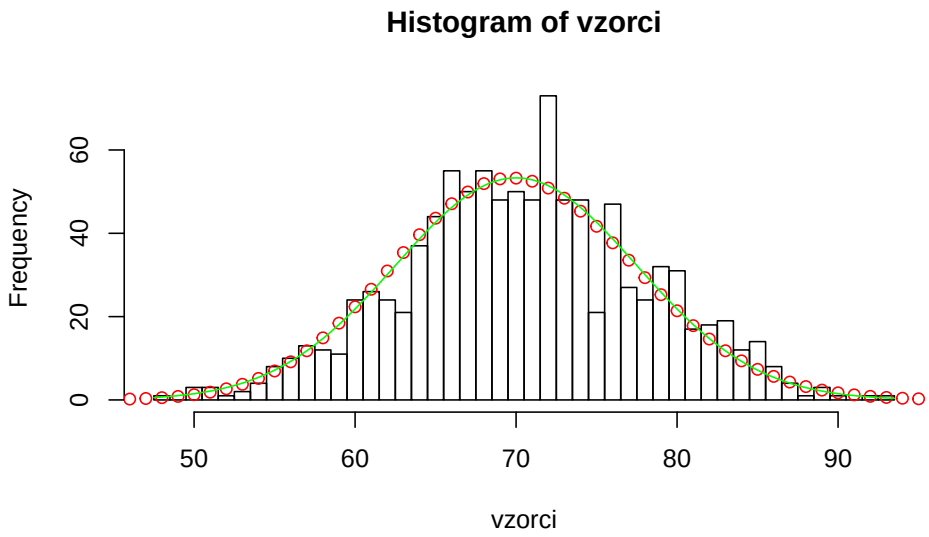
```
## [1] 0.1034488
```

```
vzorci = rbinom(1000,size=350,prob=p)
interval = min(vzorci):(max(vzorci)+1) - 0.5

# relativne frekvence
hist(vzorci,breaks=interval,freq=FALSE)
teor = dbinom(0:350,size=350,prob=p)
points(0:350,teor,col="red")
```




```
# absolutne frekvence
hist(vzorci,breaks=interval)
teor = dbinom(0:potop,size=potop,prob=p)
pricakovaneFrek = teor * 1000
points(0:potop,pricakovaneFrek,col="red")
# zvezna aproksimacija z normalno !!
curve(dnorm(x,mean=potop*p,sd=sqrt(potop*p*(1-p)))*1000,
      add=TRUE,col="green")
```



3. Predstavite porazdelitev števila potopov za 80 biserov.

   - Z geometrijsko porazdelitvijo lahko simuliramo število potopov do

prvega bisera. Geometrijska porazdelitev ima en parameter, in sicer je to verjetnost za uspešen dogodek:

$$P(X = x) = p \cdot (1 - p)^x.$$

  - S pomočjo R naredite eno opazovanje za število potopov do (vključno) 1.bisera.
  - Simulirajte število potopov za 80 biserov.
- Simulacijo iz zgornje alineje ponovite 1000 x:
  - narišite histogram števila potopov in pravilno označite osi
  - čez histogram narišite teoretični frekvenčni poligon (uporabite negativno binomsko porazdelitev)
  - iz simulacije izračunajte opazovano verjetnost za največ 350 potopov.

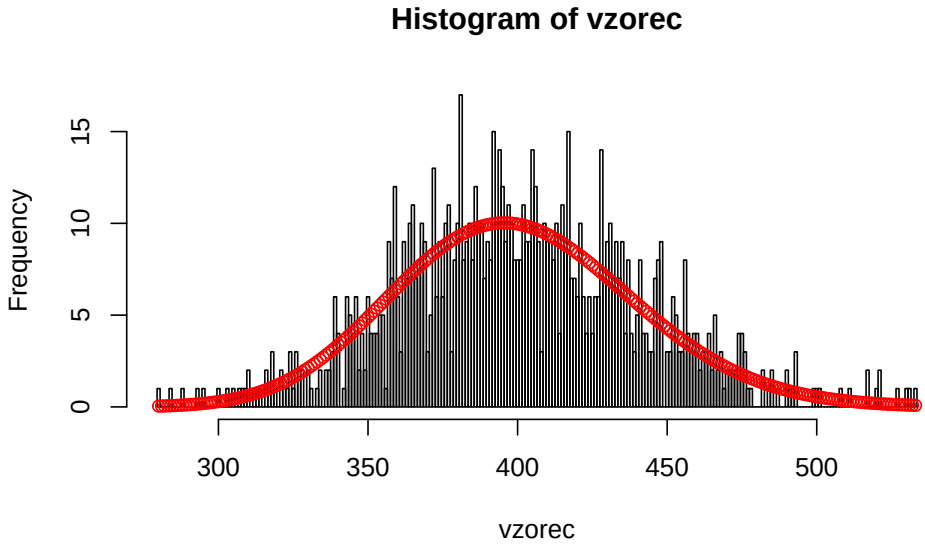
```
set.seed(1)
rgeom(1,prob =p) +1
```

```
## [1] 3
```

```
set.seed(1)
sum(rgeom(80,prob =p) +1)
```

```
## [1] 341
```

```
vzorec = replicate(1000,sum(rgeom(80,prob =p) +1))
interval = min(vzorec):(max(vzorec)+1) - 0.5
hist(vzorec,breaks=interval)
teor = dnbinom(min(vzorec):max(vzorec)-80,size=80,prob=0.2)
points(min(vzorec):max(vzorec),teor*1000,col="red")
```



```r
mean(vzorec <=350)
```

```
## [1] 0.097
```

Poglavje 11

Sintaksa pri statističnih testih

Podatki, ki jih bomo uporabljali pri tej vaji, so podatki `studenti2012.txt`, ki smo jih srečali pri Risanju podatkov.

```
studenti = read.table("data/studenti2012.txt",sep="\t",header=TRUE)
# popravimo napacne vnose in dolocimo faktorje
studenti[studenti$masa == 700,"masa"] = 70
studenti = studenti[-which(studenti$starost == 59),]
studenti$mesec[studenti$mesec==0] = NA
studenti$mesec = factor(studenti$mesec,levels=1:12,
                        labels=c("jan","feb","mar","apr","maj","jun",
                                 "jul","avg","sep","okt","nov","dec"))
studenti$majica = factor(studenti$majica,
                         levels=c("XS","S","M","L","XL"),ordered=TRUE)
```

V R obstaja ogromno funkcij za izračun statističnih testov. Osnovni:

- `t.test`
- `wilcoxon.test`
- `chisq.test`
- `fisher.test`
- `binom.test`
- `aov` (ANOVA)
- `kruskal.test`
- `lm` (linearna regresija)
- `glm` (posplošena linearna regresija - logistična regresija)

Naloge

- Preverite delovno hipotezo, da so v povprečju fantje višji od deklet. Zapišite ukaz
  - s pomočjo formule
  - brez formule
  - naredite enostranski test, saj veste v katero smer pričakujete odmik, $\alpha = 0.01$

```
rezultatT = t.test(visina~spol,data=studenti)
t.test(x=studenti$visina[studenti$spol=="F"],
       y=studenti$visina[studenti$spol=="M"])
```

```
##
##  Welch Two Sample t-test
##
## data:  studenti$visina[studenti$spol == "F"] and studenti$visina[studenti$spol == "M
## t = -6.4643, df = 12.502, p-value = 2.55e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -17.901862  -8.906219
## sample estimates:
## mean of x mean of y
##  166.8182  180.2222
```

```
rezultatT2 = t.test(visina~spol,data=studenti,alternative="less",
                    conf.level=0.95)
```

- Je delež svetlolasih deklet večji od deleža svetlolasih fantov? (se dekleta raje barvajo na svetlo barvo)
  - Izpišite opazovane in pričakovane frekvence.
  - Uporabite test, ki ima izpolnjene predpostavke.
    - Za katero mero je izpisan interval zaupanja?
    - Izračunajte to mero še sami (brez intervala zaupanja).

```
chisq.test(x=studenti$lasje,y=studenti$spol)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  studenti$lasje and studenti$spol
## X-squared = 1.4096, df = 1, p-value = 0.2351
```

```
chisq.test(x=table(studenti$lasje,studenti$spol))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(studenti$lasje, studenti$spol)
```

```
## X-squared = 1.4096, df = 1, p-value = 0.2351
```

```
fisher.test(x=studenti$lasje,y=studenti$spol)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  studenti$lasje and studenti$spol
## p-value = 0.1494
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##   0.5699288 40.7046845
## sample estimates:
## odds ratio
##   3.609737
```

```
p1 = 17/33
p2 = 2/9
p1/(1-p1)/(p2/(1-p2)) # 3.71875
```

```
## [1] 3.71875
```

- Odgovorite na raziskovalno vprašanje: *Ali velja, da se največ ljudi rodi januarja?* Uporabite primeren statistični test. (naj bodo predpostavke izpolnjene)

```
chisq.test(x=table(studenti$mesec),simulate.p.value = TRUE)
```

```
##
##  Chi-squared test for given probabilities with simulated p-value
##  (based on 2000 replicates)
##
## data:  table(studenti$mesec)
## X-squared = 13.732, df = NA, p-value = 0.2499
```

- Raziskovalno vprašanje: *Ali se teža študentov glede na njihovo starost kaj razlikuje?* pretvorite v dva smiselna statistična testa in odgovorite na vprašanje.
  - V tabeli izpišite še povprečno težo glede na starost.
  - Ker vemo, da je teža povezana z višino posameznika, dodajte v model kot prediktor (neodvisno spremenljivko) tudi višino študentov in ponovno vsebinsko interpretirajte rezultat.

```
starosti = sort(unique(studenti$starost))
povpTeze =sapply(starosti,
                 FUN=function(x){mean(studenti$masa[studenti$starost==x],
                                      na.rm=TRUE)})

library(knitr)
```

```
kable(cbind(starosti,povpTeze))
```

starosti	povpTeze
20	62.85714
21	62.60870
22	64.66667
23	58.00000

```
?aov
resultA =aov(masa~as.factor(starost),data=studenti)
summary(resultA)
```

```
##                    Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(starost)  3    101   33.76   0.319  0.812
## Residuals          38   4022  105.85
```

```
resultL = lm(masa~starost + visina,data=studenti)
summary(resultL)
```

```
##
## Call:
## lm(formula = masa ~ starost + visina, data = studenti)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.352  -4.141  -1.779   3.581  24.044
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -92.48814   40.14232  -2.304   0.0266 *
## starost       0.00986    1.41940   0.007   0.9945
## visina        0.91367    0.14740   6.198 2.75e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.293 on 39 degrees of freedom
## Multiple R-squared:  0.4969, Adjusted R-squared:  0.4711
## F-statistic: 19.26 on 2 and 39 DF,  p-value: 1.52e-06
```

```
# primer za primer Galton spodaj
# za testiranje domneve, da je reg.koeficient neka neničelna konstanta
resultL = lm(visina~oce,data=studenti)
x = summary(resultL)
confint(resultL)
```

```
##                  2.5 %     97.5 %
## (Intercept) 3.89955894 162.925959
## oce         0.03352379   0.921262
```

```
testna = (x$coefficients[2,1] - 1)/x$coefficients[2,2]
pt(testna,df=40)*2
```

```
## [1] 0.02163737
```

11.1 Regresija k povprečju – poročilo analize podatkov

Galton je ugotavljal korelacijo med velikostjo staršev in potomcev. Uvedel je pojem regresija, ki izvira iz njegove ugotovitve, da sta višina starša in potomca v posebnem razmerju, ki zagotavlja 'regresijo' k povprečju. Sam je to imenoval tudi *reversion to mediocrity*.

Velja, da je višina delno gensko pogojena, torej bodo potomci višjih oseb tudi sami višji. Regresija k povprečju pa v tem primeru pomeni, da bomo vseeno vedno opazili, da so potomci zelo visokih staršev v povprečju nižji od njih, in podobno potomci zelo nizkih staršev v povprečju višji.

Njegova biološka razlaga te ugotovitve, je sicer napačna, matematični konstrukti linearne regresije pa so temelj nadaljnjemu razvoju regresijskih modelov.

Naloga

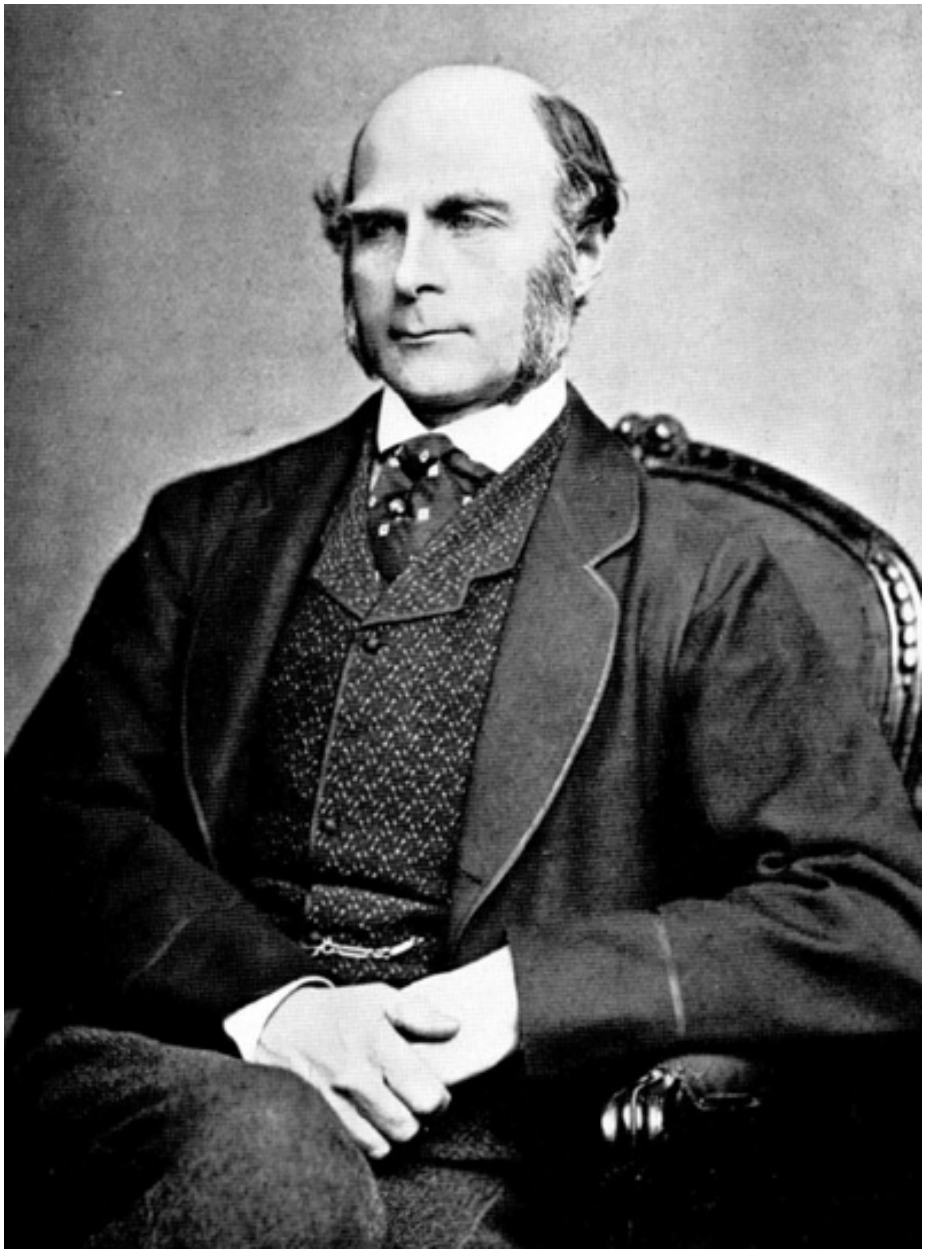
Naj bo naša raziskovalna naloga poskusiti dokazati, da gre pri višini staršev in potomcev res za neke vrste regresijo k povprečju. Za ta namen uporabimo podatke `studenti`. Spremenljivke, ki jih imamo na voljo, naj bodo podmnožica vseh spremenljivk iz podatkovja, in sicer:

```
galton = studenti[c("spol","visina","mati","oce")]
```

Naredite poročilo analize podatkov (datoteko `html` ali `pdf`) za to raziskovalno nalogo. Upoštevajte pravila za izdelavo poročila. Vsako poročilo naj bi vsebovalo:

1. Kratek opis ozadja podatkov in raziskovalno vprašanje.
2. Poročilo o čiščenju podatkov in opis ostalih težav, na katere ste pri tem naleteli.
3. Predstavitev vzorca s pomočjo opisnih statistik (v tabeli in grafično).
4. Izbira primernega statističnega testa za odgovor na raziskovalno vprašanje. Predstavitev rezultatov in interpretacija končnih rezultatov, ki je primerna tudi za nestatistike.

Pri poročilu je potrebno vedno premisliti, komu je namenjeno in kako tehnično je lahko poročilo. Vedno se izogibajte direktnim izpisom iz programskega paketa, ki ga uporabljate, poročajte le tiste rezultate, ki so za primer smiselni!



Slika 11.1: Sir Francis Galton (1822 - 1911)

Poglavje 12

Sestavljanje grafa in risanje z ggplot

- podatki (`data.frame`)
- kateri podatki bodo prikazani in kako: `aes()`
  - spremenljivke na grafu `mapping=aes(x=Sex, y=Height, ...)`
  - dodatni parametri: `color` (zunanje barve), `fill` (notranje barve), `position`, `shape` (oblika točk), `size`, `linetype`, ...
  - `group`: če želimo podatke prikazati po skupinah na istem grafu
  - vrednosti, ki so odvisne od kakšne spremenljivke podatkov, so določene znotraj `aes()`, fiksne vrednosti izven `aes()`! Na primer `aes(x=Height, color=Sex)` je pravilno, `aes(x=Height, color="black")` ni pravilno, saj je barva konstanta.
  - `aes` določimo v osnovni funkciji `ggplot()`, lahko pa ga določamo tudi kasneje (še posebno, če želimo z različnimi dodatnimi elementi vrednosti malce spremeniti)
- geometrijski elementi: `geom_*` (določamo lahko tudi: barve, velikosti, položaje); vsak graf mora imeti vsaj en `geom`. Znotraj `geom` je lahko tudi `aes()` Primeri `geom_*`
  - `geom_histogram()`: histogram
  - `geom_bar()`: stolpični diagram
  - `geom_boxplot()`: okvir z ročaji
  - `geom_points()`: točke, razsevni diagram
  - `geom_line()`: premica
  - `geom_text()`: tekstovne oznake
  - ...
  - položaj (`position`): kako urediti geometrijske elemente, ki bi se nahajali na istem mestu, je ponavadi specificirano znotraj `geom`
  - `geom_point(position = "jitter")`: točke slučajno malo premakni
  - `geom_bar(position = štack")`: stolpec naloži

  - `geom_bar(position = "dodge")`: stolpec položi poleg
  - `geom_bar(position = "fill")`: stolpec naloži in standardiziraj
  - `geom_point(position="identity")`: ohrani, kot je (tudi, če se prekrivajo)
- skale (scales)
  - `scale_x_continuous()`, `scale_y_continuous()`: zvezne osi
  - `scale_*_discrete()`: diskretne osi
  - `scale_*_log10()`: logaritemska transformacija
  - `scale_color_*`
  - `scale_size_*`
  - `scale_shape`
- `coord_*` koordinatni sistem
  - `coor_cartesian()`: kartezični koordinatni sistem
  - `coor_flip()`: obrne x in y osi v kartezičnem koordinatnem sistemu
- faceting: risanje po podskupinah (delih osnovnega podatkovja)
  - `facet_grid(.~.)`
  - `facet_wrap(.~.)`
- tema: oblika grafikona
  - `theme_classic()`
  - `theme_bw()`
  - spreminjanje teme samo za del kode v R: `previous_theme <- theme_set(theme_bw())` in na koncu `theme_set(previous_theme)`

Dodatne uporabne možnosti

- statistične transformacije `stat()` za prikaz povzetkov na grafu
- imenovanje `labs(x="oznaka na x osi", y="oznaka na y osi", title="naslov")`

- legende: znotraj dela `guides()` ali kot argument `legend.*` znotraj ostalih elementov - npr. `scale_*_*`, `theme`
- risanje čez prejšnjo sliko (overplotting) `color=alpha("black",0.2)`: uporaba transparentne barve
- `ggsave` za shranjevanje grafa kot slike (.pdf, .png, .jpg itn.)

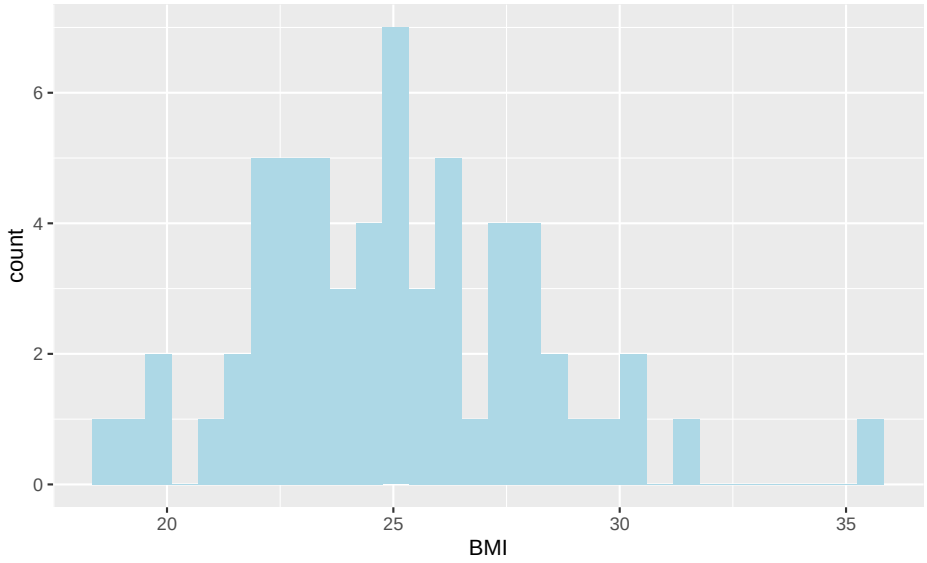
Poskušajmo grafe iz vaje osnovnega risanja (Risanje podatkov) narisati še v notaciji `ggplot`a.

12.1 Histogram, okvir z ročaji

```
library(ggplot2)
# get "podatki"
source("data/podatki.R")
p1 = ggplot(data=podatki,mapping=aes(x=BMI)) +
```

```
  geom_histogram(fill="lightblue")
p1
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
summary(p1)
```

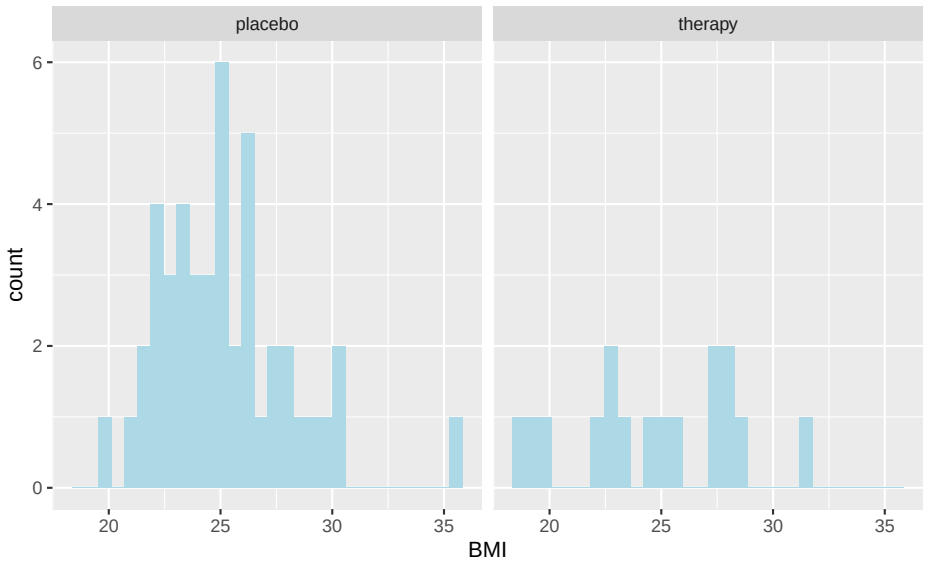
```
## data: ID, skupina, BMI, datum.testiranja, datum.okuzbe, terapija,
##   razlika, skupinaN, terapijaN [69x9]
## mapping:  x = ~BMI
## faceting: <ggproto object: Class FacetNull, Facet, gg>
##     compute_layout: function
##     draw_back: function
##     draw_front: function
##     draw_labels: function
##     draw_panels: function
##     finish_data: function
##     init_scales: function
##     map_data: function
##     params: list
##     setup_data: function
##     setup_params: function
##     shrink: TRUE
##     train_scales: function
##     vars: function
##     super:  <ggproto object: Class FacetNull, Facet, gg>
## -----------------------------------
## geom_bar: na.rm = FALSE
```

```
## stat_bin: binwidth = NULL, bins = NULL, na.rm = FALSE, pad = FALSE
## position_stack
```

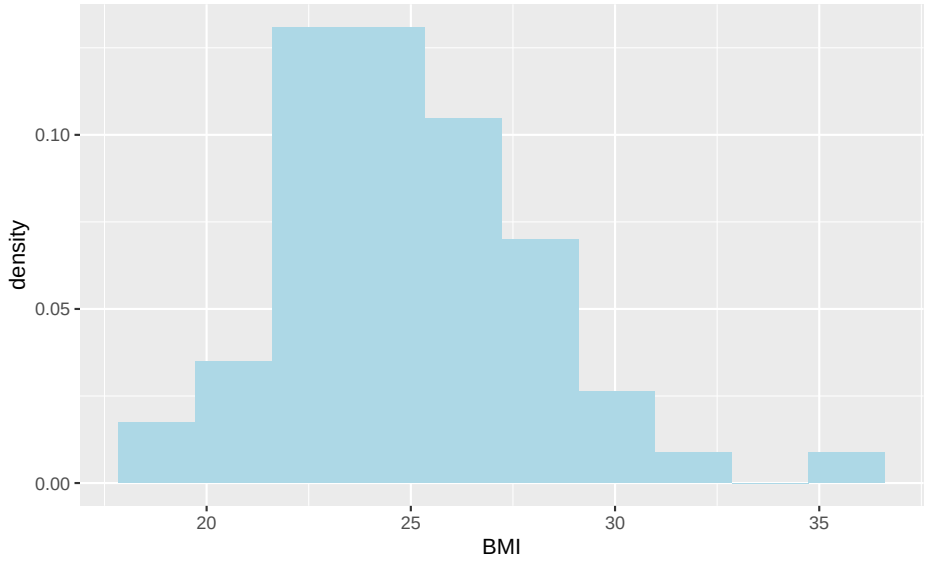
Histogram BMI glede na terapijo

```
p1 + facet_grid(.~terapijaN)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



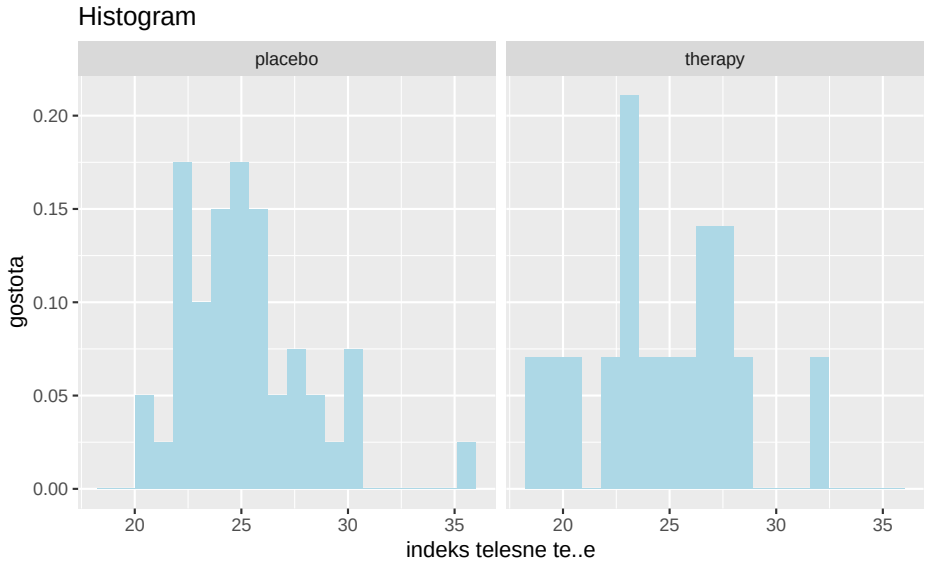

```
# switch to density instead of frequency
ggplot(data=podatki,mapping=aes(x=BMI,y=..density..)) +
  geom_histogram(fill="lightblue",bins=10)
```



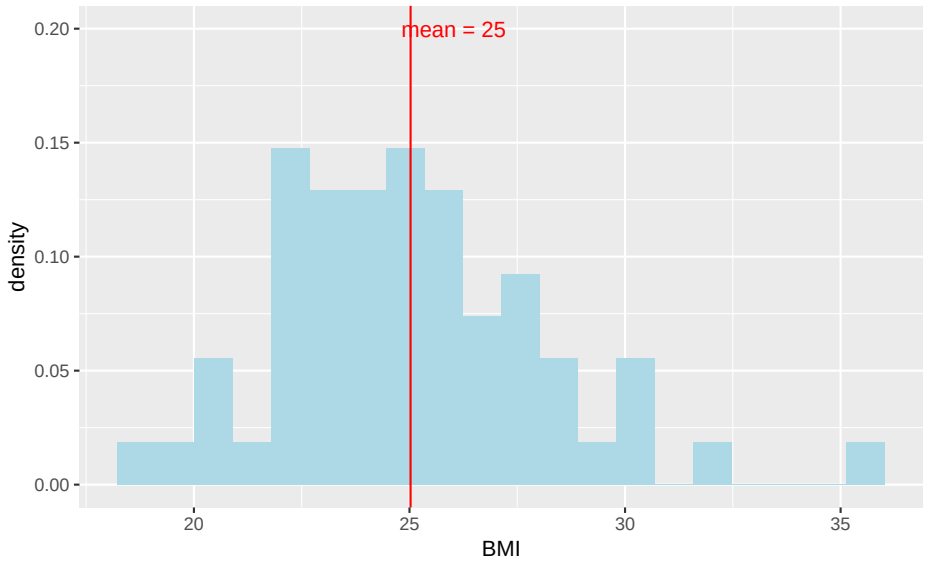
V objemu `..` se zapisuje povzetek, ki se prek ggplota generira iz spremenljivk in ga želimo imeti izrisanega. To so lahko npr. tudi: * `..identity..` * `..count..` * `..sum..` * ...

Spreminjanje oznak

```
p2 = ggplot(data=podatki,mapping=aes(x=BMI,y=..density..)) +
  geom_histogram(fill="lightblue",bins=20)
p2 + facet_grid(.~terapijaN) +
  labs(x="indeks telesne teže",y="gostota",title="Histogram")
```



```
povp = mean(podatki$BMI,na.rm=TRUE)
p2 + geom_vline(xintercept = povp,col="red") +
  annotate(geom="text",x=(povp+1),y=0.2,
           label=paste("mean =",round(povp,1)),color="red")
```



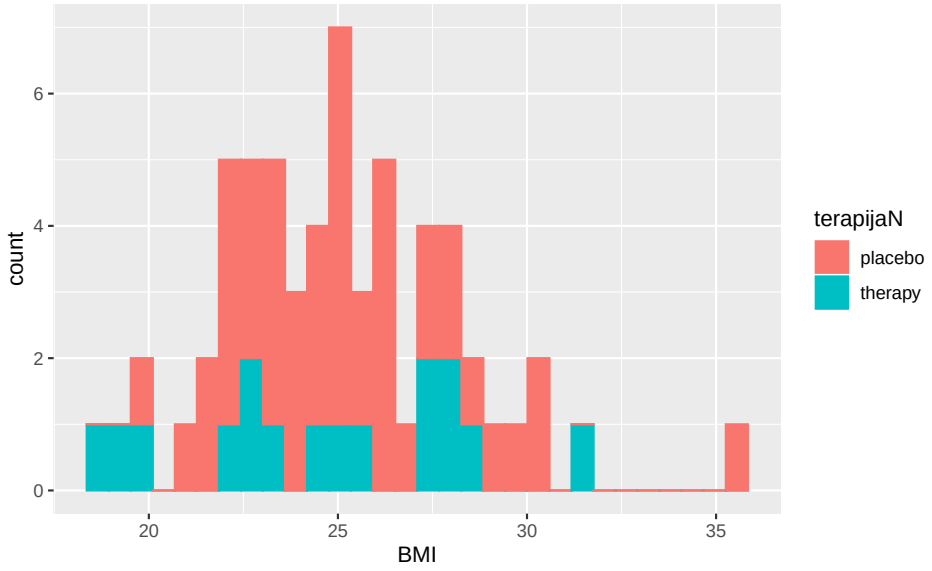
Naloge

- Dodajte zadnjemu grafu še mediano in dopišite vrednosti povprečja in mediane na graf (element `annotate`).
- Spremenite graf tako, da bo porazdelitev BMI prikazana kot okvir z ročaji (na kateri osi so vrednosti spremenljivke?)
- Graf razdelite tako, da boste imeli porazdelitev razdeljeno glede na skupino. Uporabite:
  - `facet_grid`
  - definirajte dodatni `aes` element
- Prikažite porazdelitev glede na skupino in na terapijo.

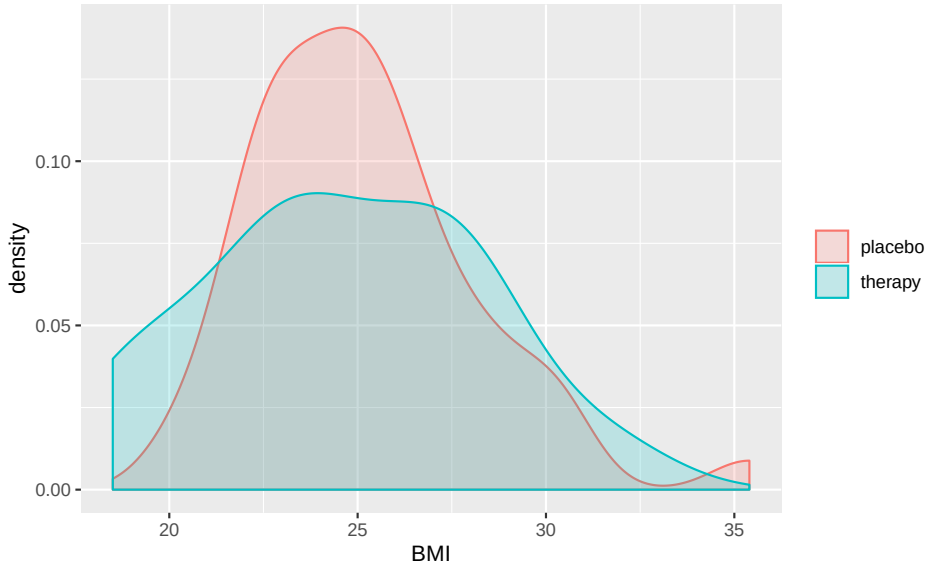
Prekrivanje grafov

```
ggplot(podatki, aes(x=BMI, fill=terapijaN, color=terapijaN))+
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



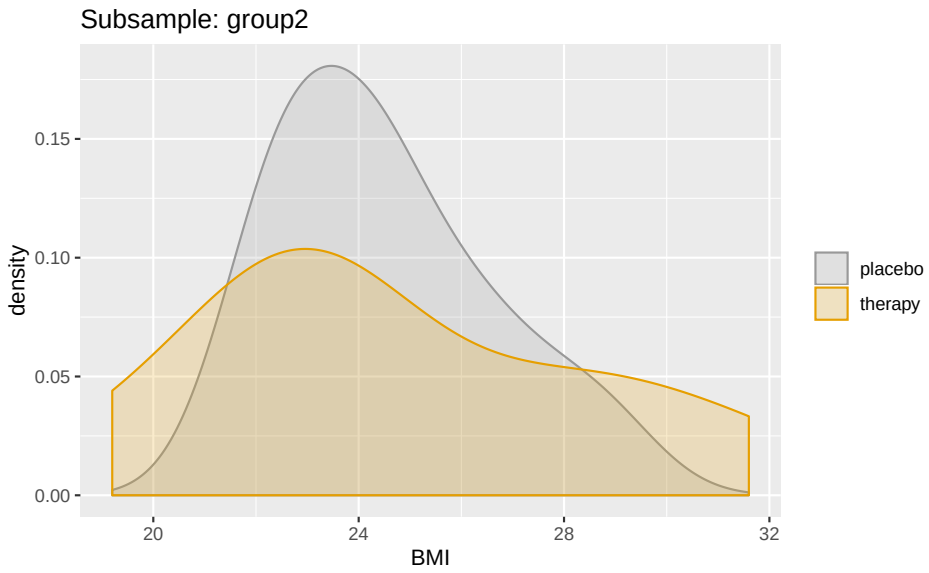

```
# density
p3 = ggplot(podatki, aes(x=BMI, fill=terapijaN, color=terapijaN))+
  geom_density(alpha=0.2) + theme(legend.title=element_blank())
p3
```



Isti graf na podmnožici podatkov (le skupina 2)

```
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73",
               "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
podatki2 = podatki[podatki$skupina==2,]
```

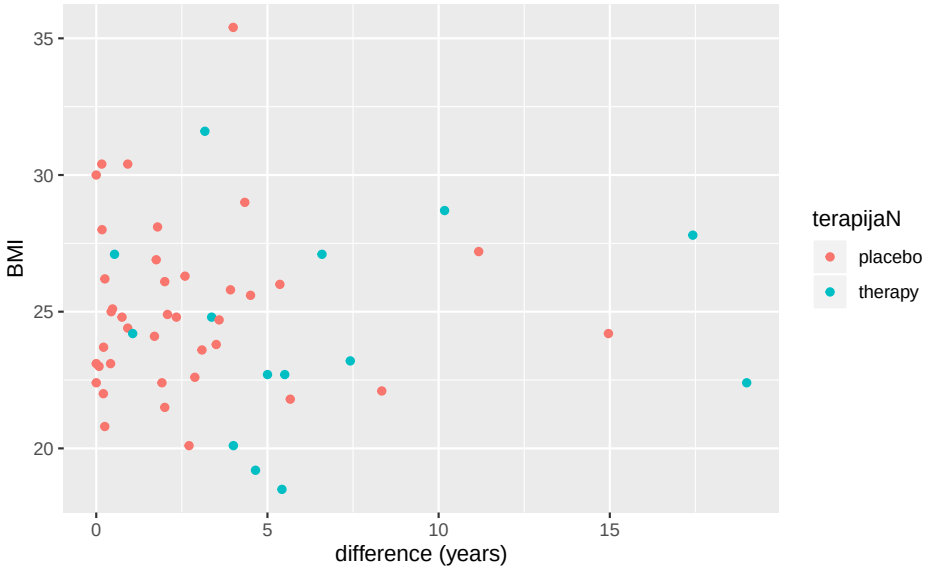
```
p3 %+% podatki2 + labs(title="Subsample: group2") +
  scale_color_manual(values=cbPalette) +
  scale_fill_manual(values=cbPalette)
```



12.2 Razsevni diagram

```
pp1 = ggplot(podatki,aes(x=razlika/365.24,y=BMI,col=terapijaN)) +
  geom_point() + labs(x="difference (years)")
pp1
```

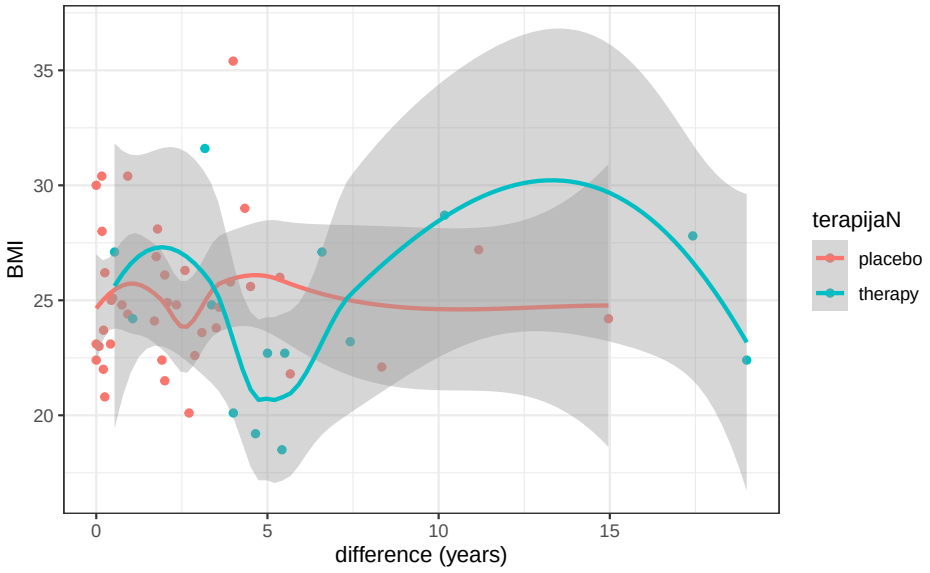
```
## Don't know how to automatically pick scale for object of type difftime. Defaulting t
```




```
pp1 + theme_bw() + stat_smooth()
```

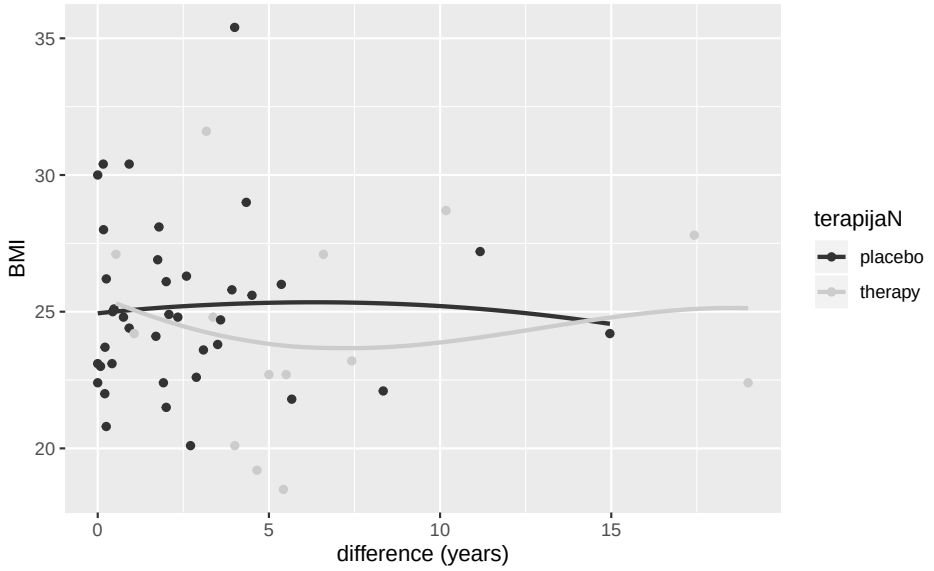
```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuo
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



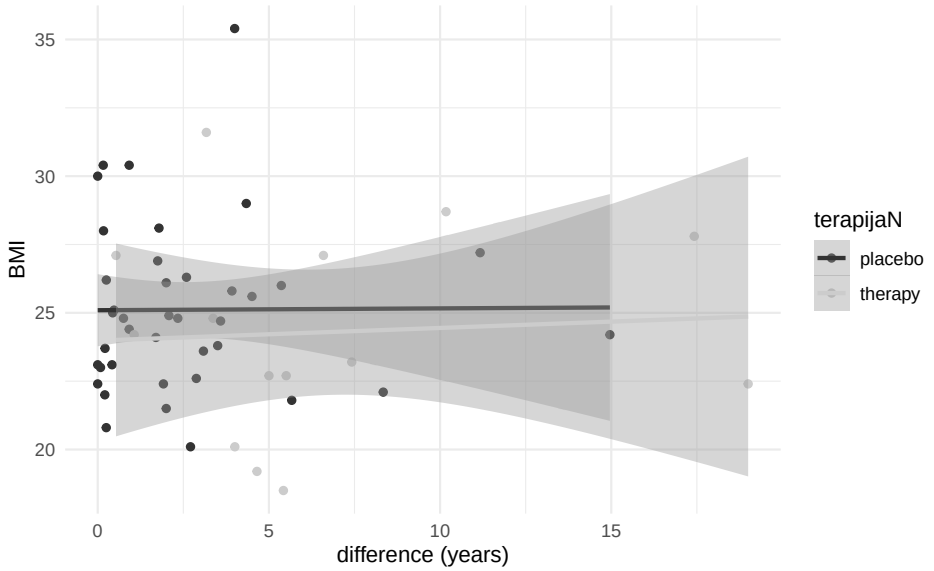

```
pp1 + stat_smooth(span=2,se=FALSE) + scale_color_grey()# less overfit
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuo
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
pp1 + stat_smooth(method="lm")+ scale_color_grey() +
  theme_minimal() # linear regression fit
```

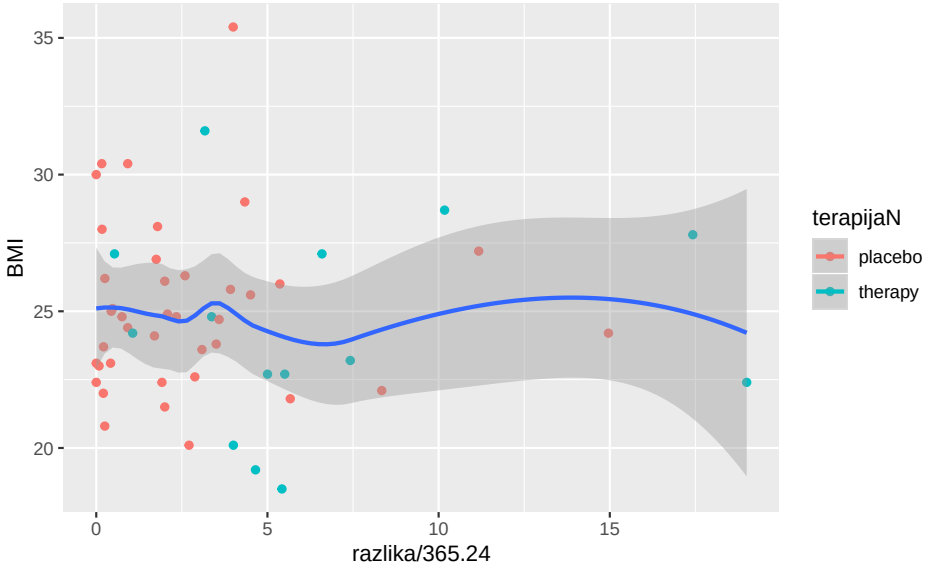
```
## Don't know how to automatically pick scale for object of type difftime. Defaulting t
```




```
pp2 = ggplot(podatki,aes(x=razlika/365.24,y=BMI,col=terapijaN)) +
  geom_point() + stat_smooth(aes(group=1)) # one fit for both groups
pp2
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting t
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Naloge

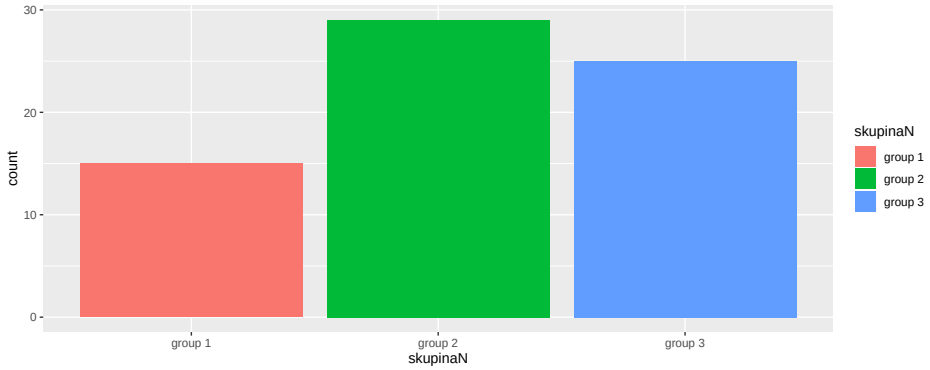
```
studenti = read.table("data/studenti2012.txt",sep="\t",header=TRUE)
# popravimo napacne vnose in dolocimo faktorje
studenti[studenti$masa == 700,"masa"] = 70
studenti = studenti[-which(studenti$starost == 59),]
studenti$mesec[studenti$mesec==0] = NA
studenti$mesec = factor(studenti$mesec,levels=1:12,
                        labels=c("jan","feb","mar","apr","maj","jun",
                                 "jul","avg","sep","okt","nov","dec"))
studenti$majica = factor(studenti$majica,
                         levels=c("XS","S","M","L","XL"),ordered=TRUE)
```

- Narišite histograma za višino študentk in študentov posebej (enega pod drugim, da ju lahko primerjate).
- Čez dva grafa iz prve točke dodajte še gostoto (pazite, da boste imeli y na enaki skali)!
- Narišite porazdelitev višine študentk in študentov z okvirjem z ročaji.
- Na prejšnji graf dodajte še točke, ki jih ločite z `geom_jitter`.
- Narišite povezanost med višino in težo. Označite spol.
- Dodajte krivuljo, ki se najbolj prilega podatkom (`loess`), nato pa še premico, ki se najbolje prilega podatkom.
  - najprej naj bo to ena krivulja za VSE podatke
  - nato dve krivulji po spolu

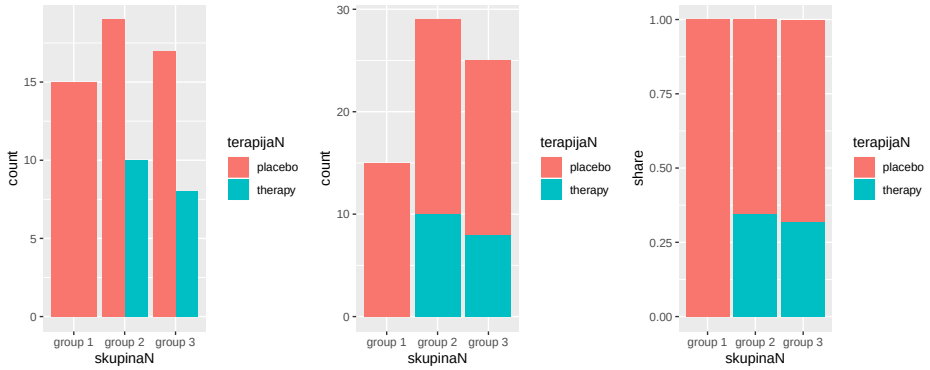
- Narišite porazdelitev razpona rok glede na velikost majice. Na sliki označite tudi povprečje vsake skupine
  - s histogrami
  - z okvirji z ročaji

12.3 Stolpični diagram

```
library(gridExtra)
pb = ggplot(podatki,aes(x=skupinaN))
pb + geom_bar(aes(fill=skupinaN))
```

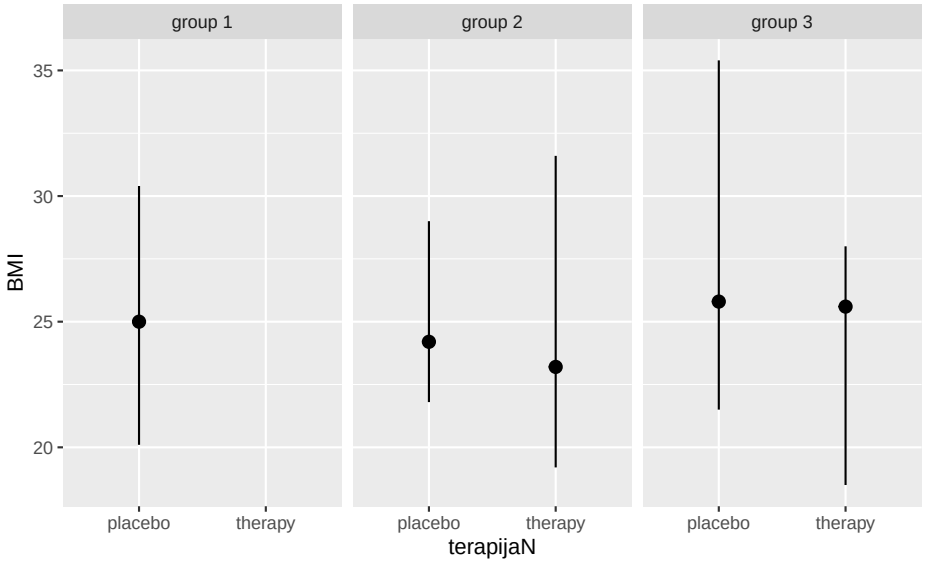



```
pb1 = pb + geom_bar(aes(fill=terapijaN),position="dodge")
pb2 = pb + geom_bar(aes(fill=terapijaN),position="stack")
pb3 = pb + geom_bar(aes(fill=terapijaN),position="fill") + labs(y="share")
grid.arrange(pb1, pb2, pb3, nrow=1)
```



12.4 Povzetki spremenljivk po skupinah

```
ggplot(podatki)+stat_summary(mapping = aes(x = terapijaN, y = BMI),
                             fun.ymin = min, fun.ymax = max,
                             fun.y = median)+ facet_wrap(skupinaN~.)
```



Naloge

- Prikažite kategorialne podatke pri študentih na najboljši možni grafični način.
- Narišite stolpični diagram, spol/velikost majice, ki bo prikazoval deleže ljudi posameznega spola, ki nosijo določeno velikost majice.
- Prikažite povzetek višine glede na spol (tj. minimum, maksimum in mediano).
- Prikažite referenčna intervala razpona rok za vsak spol posebej. Na koncu jih ločite še glede na barvo las. Ali pričakujete, da se bosta referenčna intervala razlikovala? Zakaj?
- Poglejte tabelo naslednjih podatkov (izraženo v milijonih oseb):

Tabela 12.1: Uporaba elektronskih naprav

računalnik	TV	telefon	radio	Ipod	spletna kamera
4.3	6	8	3.5	1	1.4

- Podatke shranite v `data.frame`.

- Narišite stolpični diagram (bodite pozorni, kaj boste uporabili za `stat` (statistiko), saj je v tabeli že frekvenca)
- Obrnite osi (naj bodo stolpci vodoravni).

Poglavje 13

Domače naloge

K poglavju 2

Naloga – Rmd

Doma si uredite `R`, `RStudio` in naredite iz predavanj svoj PDF dokument. Dokument predavanj je na spletni učilnici objavljen kot datoteka `Rmd`. Dodajte mu rešitve obravnavanih nalog.

K poglavju 4

Naloga – Funkcije

Naredite Rjev dokument z naslednjimi funkcijami:

- **izpisStevilska(x,stDec)**

  Rezultat naj bo znakovni niz, ki bo na lep način prikazal povprečje in standardni odklon številske spremenljivke `x` na `stDec` mest natancno (za zaokroževanje uporabljajte funkcijo `round()`).

- **izpisStevilskaAsim(x,stDec)**

  Rezultat naj bo znakovni niz, ki bo na lep način prikazal mediano in interkvartilni razmik številske spremenljivke `x` na `stDec` mest natancno. Pri tem si pomagajte s funkcijo `quantile()`.

- **izpisOpisna(x)** Rezultat naj bo znakovni niz, ki bo na lep način prikazal frekvenco in delež vseh kategorij opisne spremenljivke `x`. Pazite: spremenljivka ima lahko več kot 2 kategoriji.

Table 1 Baseline characteristics of NOAC-associated and warfarin-associated ICH

Variables	NOAC-ICH (n = 11)	Warfarin-ICH (n = 52)	*p* Value
Age, y, median (IQR)	81 (76-83)	80 (72-85)	0.45
Female, n (%)	9 (82)	20 (38)	0.009
Event-scan time, d, median (IQR)	0 (0-0)	0 (0-0)	0.29[a]
Hypertension, n (%)	10 (91)	35 (67)	0.12
Hypercholesterolemia, n (%)	76 (55)	30 (59)	0.80
Diabetes, n (%)	2 (18)	15 (29)	0.47
Smoking (never), n (%)	5 (50); n = 10	26 (50)	0.64
Alcohol units/wk, median (IQR)	0 (0-0)	1 (0-4); n = 50[b]	0.12[a]
IHD, n (%)	2 (18)	10 (20); n = 51[b]	0.91
Previous IS, n (%)	6 (54)	13 (25); n = 51[b]	0.06
Previous ICH, n (%)	2 (19)	12 (24); n = 51[b]	0.70
Previous TIA, n (%)	2 (19)	12 (24); n = 49[b]	0.66
Concurrent antiplatelet, n (%)	1 (9)	5 (9)	0.96
Premorbid modified Rankin Scale score, median (IQR)	3 (1-3)	1 (0-2); n = 48[b]	0.04[a]
Anticoagulant reversal (3 or 4 factor prothrombin complex) n (%)	6 (55)	45 (87)	0.44
SVD, Fazekas score median (IQR)	0 (0-2)	2 (0-2)	0.22[a]
Non lobar ICH location, n (%)	8 (73)	30 (59)	0.39
IVH, n (%)	3 (27)	15 (29)	0.89
AF as reason for anticoagulation n (%)	8 (73)	39 (75)	0.42

Abbreviations: AF = atrial fibrillation; ICH = intracerebral hemorrhage; IHD = ischemic heart disease; IQR = interquartile range; IS = ischemic stroke; IVH = intraventricular hemorrhage; NOAC = non-vitamin K oral anticoagulant; SVD = small vessel disease.
[a] Nonparametric Mann-Whitney or Fisher exact test used (*t* test or χ^2 tests were used otherwise, as appropriate).
[b] Number of patients for whom data were available in variable with incomplete/missing data.

Slika 13.1: Tabela opisnih statistik (Wilson et al (2016): Volume and functional outcome of intracerebral hemorrhage according to oral anticoagulant type)

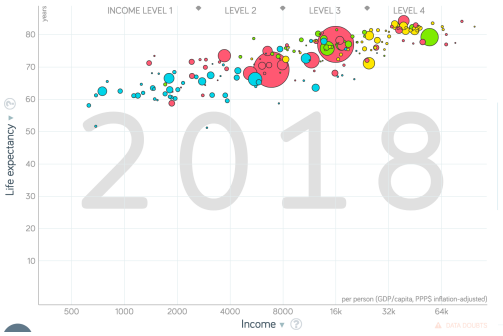
Poleg funkcij naj bosta za vsako funkcijo pripravljena tudi vsaj 2 primera uporabe.

Te funkcije bodo omogočale pripravo izpisov za spremenljivke, ki jih bomo kasneje enostavno sestavili v tabelo, kot je npr. spodnja:

K poglavju 5, 6

Naloga – Delo z datotekami in risanje

Oglejte si spletno stran https://www.gapminder.org/. Gre za spletno stran, ki jo je ustanovil švedski statistik Hans Rosling za popularizacijo statistike. Oglejte



Slika 13.2: Prvi Gapminderjev graf - Hans Rosling

si naslednja zavihka:

- `Tools` (orodje za risanje zanimivih grafov)
- `Data` (za pridobivanje podatkov)

S pomočjo zavihka `Data` pridobite podatke, da boste lahko v Rju izrisali prvi graf, ki se vam pojavi, ko kliknete na zavihek `Tools` (gl. sliko na koncu te domače naloge).

Torej, radi bi narisali razsevni diagram za leto 2018 (lahko si izberete tudi kakšno drugo leto, ki ima dovolj podatkov). Narišite torej povezavo povprečnega dohodka s pričakovano življenjsko dobo. Pri tem upoštevajte (predstavite) tudi velikost populacije v posamezni državi. Barv ni potrebno izrisovati.

Kot rezultat pošljite .html dokument (narejen prek .Rmd) z izpisanimi vsemi ukazi v R, ki ste jih pri tem uporabili. V nalogi se zahteva, da

- podatke s spletne strani Gapminder naložite na računalnik
- datoteke iz spomina računalnika preberete v R
- izberete le tiste spremenljivke, ki vas zanimajo
- jih povežete v skupen `data.frame` - pri tem si poglejte funkcijo `merge()`
- narišete razsevni diagram, ki bo podoben Gapminderjevemu (z lepim izpisom imen)

Naloga – Risanje vzorcev

1. Oglejte si funkcijo `sample`. S pomočjo te funkcije generirajte 20 vzorcev spremenljivke X velikosti $n = 15$. Spremenljivka X lahko zavzame naravna števila od 1 do 5, verjetnosti za posamezno vrednost pa so navedene spodaj:

vrednost $X = x$	1	2	3	4	5
verjetnost $P(X = x)$	0,15	0,4	0,3	0,1	0,05

2. Vseh 20 vzorcev narišite s primernim grafičnim prikazom (izris vzorcev brez uporabe zanke ne zadostuje za opravljeno domačo nalogo). Vsi grafi naj bodo na eni sliki. Pazite, da bodo imeli vsi grafi enako lestvico tako na x kot na y osi. Končno sliko shranite v PDF dokument. Višino in širino slike nastavite tako, da se bodo vsi elementi grafov lepo videli (tj. lestvice, številke ipd.).

3. Nato naredite enako še za vzorce velikosti $n = 100$. Novega grafa ni potrebno shraniti, ga pa preglejte in komentirajte ugotovitve.

   - Kateri grafi so bolj variabilni in zakaj?

PDF sliko iz točke 2. Naložite, prosim, oba dokumenta posebej in ne v stisnjeni različici.

Domačo nalogo oddajte v html z imenom **dn5_priimek.html** (kjer namesto besede *priimek* uporabite vaš priimek). Naloga naj vsebuje vso kodo v R in izris obeh grafov.

Naloga – Prikaz spremljanja pacientov

Iz datoteke `dn5_data.txt`, ki je priložena domači nalogi, preberite podatke. Podatke te vrste ponavadi statistično analizira z metodami analize zgodovine dogodkov (angl. event history analysis, survival analysis).

S pomočjo funkcije `as.Date` pretvorite datume v datumski format `Date` in narišite graf, kjer bo za vsako statistično enoto (na y osi) prikazan čas njenega spremljanja (spremenljivka čas naj bo na x osi). Na koncu daljice, ki kaže na čas spremljanja vsake enote, naj bo označeno, ali je bila krnjena (angl. censored - ponavadi krogec) ali pa je spremljanje končano, ker je prišlo do dogodka (angl. event - ponavadi $\times$). Za daljice lahko uporabite npr. funkcijo `segments`.

Za primer si poglejte Sliko 1 (spodaj) iz znanstvenega članka *A. Brembilla in sod. – Use of the Cox regression analysis in thoracic surgical research.* Naslov pri sliki je bil:

Illustration of time to event and time to censoring of 6 subjects. The red dot represents the entry of the subject in the study. The vertical blue lines indicate the start of the study and the end of the study. The subjects 1, 2 and 5 have experienced the event by the end of the study. The subject 3 has experienced the event after the end of the study: he is right-censored. The subject 4 is lost to follow-up during the study period: he is also right-censored. The subject 6 is left-censored.

Narišite 2 grafa. Enote naj bodo urejene

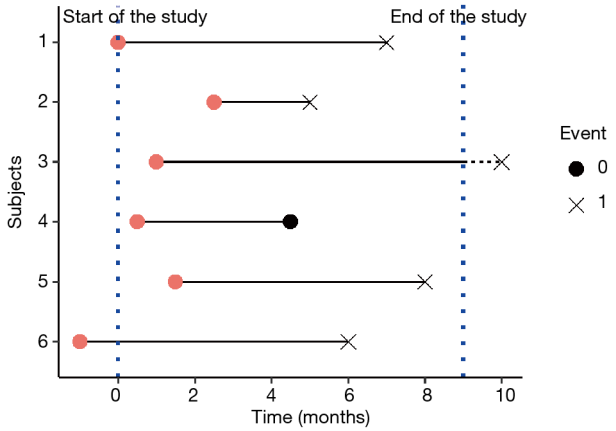
1. po njihovi številki
2. po njihovem času začetka spremljanja (entry of the subject)

K poglavju 7, 8, 9

Naloga – Bootstrapping (samovzorčenje)

Glavna ideja samovzorčenja (angl. bootstrapping) je, da lahko sklepamo na populacijo iz vzorca tako, da iz vzorca, ki ga imamo na voljo, naredimo nove vzorce (s ponavljanjem) in na podlagi vrednosti teh vzorcev sklepamo na populacijo.

Uporabo samovzorčenja bomo prikazali za sklepanje o pričakovani vrednosti.



Slika 13.3: Slika iz članka

Naredite naslednje:

1. Vzorec iz populacije naj bo naključni vzorec iz gama porazdelitve s parametroma ($\alpha = 2$ in $\beta = 0.5$):

```
set.seed(2019)
vzorec1 = rgamma(15,shape=2,rate=1/2)
```

2. Naredite 1000 novih vzorcev iz `vzorec1` z vzorčenjem s ponavljanjem. Zanje izračunajte povprečje. Izračunajte skupno povprečje in *95% interval zaupanja* tako, da iz vseh povprečij izberete 2.5 in 97.5 percentil.

3. Enako kot zgoraj naredite še za naslednji vzorec:

```
set.seed(2019)
vzorec2 = rgamma(100,shape=2,rate=1/2)
```

Pomembno: * Točka 2 mora biti izvedena brez eksplicitne uporabe zank. Pomagate si lahko npr. s funkcijami `replicate`, `apply` in `sapply`.

- Rezultate (povprečje povprečij in interval zaupanja) za oba vzorca prikažite v tabeli.

- Rezultate tabele kratko komentirajte:

  - širina intervala,
  - simetričnost,
  - odmik od prave populacijske pričakovane vrednosti za primer $\Gamma(\alpha = 2, \beta = 0.5)$).

Naloga – Problem Monty-ja Hall-a

Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice?"Vos Savant's response was that the contestant should switch to the other door (vos Savant 1990a). Under the standard assumptions, contestants who switch have a 2/3 chance of winning the car, while contestants who stick to their initial choice have only a 1/3 chance.

Paul Erdős, one of the most prolific mathematicians in history, remained unconvinced until he was shown a computer simulation demonstrating the predicted result.

Torej - radi bi naredili simulacijo za ta problem, in sicer bomo 10.000 x ponovili poskus. Lahko sledite spodnjim točkam, ali pa simulacijo naredite povsem po svoje.

Ponavljaj naslednji poskus:

1. generiraj naključno, za katerimi vrati je avto in katera vrata izberemo
2. generiraj, katera od preostalih vrat pokaže gostitelj (tu je potreben manjši razmislek)
3. zapomni si, ali si zadel avto (za oba primera: da zamenjaš vrata ali pa, da ne)

Primerjaj verjetnost zadetka za oba primera in ju izpiši v tabeli.

Simulacijo poskušajte zapisati čimbolj elegantno (npr. z uporabo funkcij, brez zank).

Naloga – Regresijska premica za napovedovanje maloprodajne cene starih avtomobilov

Starost avta v letih in njegova maloprodajna cena na trgu (v evrih) naj bosta v populaciji linearno povezani. Velja naj:

- starost avta naj bo med 2-14 let in
- starost naj bo porazdeljena po enakomerni porazdelitvi

Naj bo populacijska premica med njima naslednja:

$$cena = 8000 - 500 \cdot starost + \epsilon.$$

In naj velja, da je regresijska napaka porazdeljena kot: $\epsilon \sim N(0, 1000^2)$.

1. Na podlagi zgornje regresijske premice generirajte en vzorec avtomobilov (*starost* in *ceno*) velikosti $n = 100$. Narišite razsevni diagram za ta vzorec;

nanj narišite populacijsko premico (z modro) in ocenjeno premico (z rdečo) in komentirajte sliko.
2. Simulirajte 1000 vzorcev (kot v 1. točki) velikosti $n = 100$. Vsakokrat ocenite premico, in poročajte, v kolikšnem deležu zavrnete ničelno domnevo, da je naklon premice enak naklonu populacijske premice. S tem boste ocenili velikost statističnega testa (gl. več pri predmetu Osnove teoretične statistike).

K poglavju 11

Naloga – Risanje z ggplot

Podatke generirajte iz linearnega modela (kot v domači nalogi Regresijska premica za napovedovanje maloprodajne cene starih avtomobilov):

Starost avta v letih in njegova maloprodajna cena na trgu (v evrih) naj bosta v populaciji linearno povezani. Velja naj, da je starost avta med 2-14 let naj bo porazdeljena po enakomerni porazdelitvi. Naj bo populacijska premica med njima naslednja:

$$cena = 8000 - 500 \cdot starost + \epsilon.$$

Regresijska je napaka porazdeljena kot: $\epsilon \sim N(0, 1000^2)$.

Generirajte **6** vzorcev $n = 40$ iz zgoraj opisanega linearnega modela za maloprodajno ceno starih avtov. Vsak vzorec prikažite (z `ggplot`) s svojim razsevnim diagramom. Avtomobili naj bodo različno obarvani glede na starost:

- skupina 1: 2-4 leta
- skupina 2: 5-7 let
- skupina 3: 8-10 let
- skupina 4: 11-14 let

Na (isti) razsevni diagram vsakega vzorca dodajte prilegajoče premice (s 95% intervali zaupanja):

- za vsako skupino (glede na starost) posebej
- za cel vzorec

Vaš grafični prikaz bo torej vseboval 6 razsevnih diagramov s pripadajočimi premicami.

Komentirajte prikaze:

- širino intervalov zaupanja
- smerne koeficiente premic