

Мастер академске студије
Информациони инжењеринг

Статистика у информационом инжењерингу

Основни концепти језика R

(материјали за предавања)

1. Језик R
2. Објекти
3. Функције
4. Контрола тока
5. Пакети
6. Синтакса
7. Извори и литература

Настанак

аутори

Рос Ихака и Роберт Џентлмен

Департман за статистику, Универзитет у Оукленду (Нови Зеланд)

кључне године

1992 – полазна идеја

1993 – подељене бинарне копије

1994 – завршена почетна верзија

1995 – отворен изворни кôд

2000 – објављена верзија 1.0

назив

омаж језику S

омаж ауторима

Настанак

публикације

Ihaka R, Gentleman R. R: A Language for Data Analysis and Graphics. Journal of Computational and Graphical Statistics. 1996;5(3):299-314.

Ihaka R. R: Past and future history. In Weisberg S, editor. Proceedings of the 30th Symposium on the Interface. The Interface Foundation of North America; 1998. p. 392-396.

Ihaka R. The R project: A brief history and thoughts about the future. [Lecture] University of Otago. 20th April 2011. Internet:
<https://www.stat.auckland.ac.nz/~ihaka/downloads/Otago.pdf>

Језик

језик за статистику

израчунавања

исцртавања

слободно доступан

<https://www.r-project.org/>

широко распрострањен

велики број корисника широм света

доприноси бројних стручњака

<https://www.r-project.org/contributors.html>

Језик

интерпретиран

заснован на више парадигми

функционалној, структурној, објектној, ...

корени

по узору на језике *Scheme* и *S*

почетак од интерпретера за језик *Scheme*

прилагођаван да личи на језик *S*

синтакса

векторски типови

пасивна евалуација аргумената функције (енгл. *lazy arguments*)

комбиновани резултат

синтакса од језика *S*

евалуација израза од језика *Scheme*

Појам R

у пракси често шире значење

језик R

интерпретер за језик R

статистички софтверски систем

интерпретер

конзола

дебагер

...

Софтверска подршка

основно софтверско окружење

пројекат *R*

GNU пројекат

бесплатно доступно

најновија верзија *R 4.5.1 (Great Square Root)*

<https://www.r-project.org/>

додатна радна окружења и едитори

RStudio

<https://posit.co/products/open-source/rstudio/>

R Commander

<https://socserv.socsci.mcmaster.ca/jfox/Misc/Rcmdr/>

Rattle

<https://togaware.com/projects/rattle/index.html>

...

Основни начин употребе

путем конзоле

наредба се задаје у оквиру конзоле

наредба се извршава

парсер обрађује наредбу

евалuator извршава обрађену наредбу

евалuator враћа вредност

Пример почетног одзива конзоле

```
R version 4.0.0 (2020-04-24) -- "Arbor Day"  
Copyright (C) 2020 The R Foundation for Statistical Computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.
```

```
  Natural language support but running in an English locale
```

```
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.
```

```
>
```

Примери наредби

```
> "Kia ora"  
[1] "Kia ora"  
> 1  
[1] 1  
> 1 + 2 / 3  
[1] 1.666667  
> exp(-pi*1i) + 1  
[1] 0-1.224606e-16i  
> x <- log(0)  
> x  
[1] -Inf  
>
```

КОНЗОЛА

Основни концепти

објекат

евалуација израза

функција

пакет

У језику R, „све“ је објекат!

Садржај

1. Језик R
- 2. Објекти**
3. Функције
4. Контрола тока
5. Пакети
6. Синтакса
7. Извори и литература

Објекат

наменска структура података

служи за приступ подацима

није предвиђено директно приступање меморији рачунара

врсте објеката

основни типови

атрибути

посебни сложени објекти

Основни типови

вектори

листе

објекти језика

изрази

функције

NULL

уграђени објекти и посебни облици

очекивани објекти

објекти три тачке

окружења

пар-листе

произвољни објекти

Атрибути

називи

димензије

називи димензија

класе

атрибути временских серија

...

Посебни сложени објекти

фактори

скупови података

Вектор

низ ћелија које садрже податке
постоји поредак међу ћелијама
ћелији приступа по индексу (позицији)
индекс прве ћелије је 1
сви подаци истог типа

ОСНОВНИ ТИПОВИ ВЕКТОРА

логички
целобројни
реални
комплексни
знаковни (стринг)
бајтовски (енгл. *raw*)

Вектор

појединачне константе евалуирају се као вектори

вектор дужине један

могуће је више вредности истог типа комбиновати у вектор

помоћу функције `c(...)`

```
> 1
[1] 1
> c(1)
[1] 1
> c(1, 2, 3)
[1] 1 2 3
> c(1, 2, c(3, c(4, 5)))
[1] 1 2 3 4 5
>
```

КОНЗОЛА

Вектор

операције над векторима
резултат вектор

```
> c(1, 3, 5) + c(2, 4, 6)
[1] 3 7 11
> c(3, 5, 7) / 2
[1] 1.5 2.5 3.5
> c(10, 20, 30) / c(5, 4)
[1] 2 5 6
Warning message:
In c(10, 20, 30)/c(5, 4) :
  longer object length is not a multiple of
shorter object length
>
```

КОНЗОЛА

Вектор

постоје специјалне вредности

NA (*Not Available*)

Inf, **-Inf** (*Positive Infinity, Negative Infinity*)

NaN (*Not a Number*)

```
> c(1, 2, NA)
[1] 1 2 NA
> 2^10000
[1] Inf
> -1 * Inf
[1] -Inf
> 0 / 0
[1] NaN
>
```

КОНЗОЛА

Вектор

бројеви се начелно третирају као реални
с покретним зарезом

бројеви могу означити као цели
додавањем суфикса **L**

бројеви могу представити хексадецимално
додавањем префикса **0x**

```
> 1L
[1] 1
> 0xA
[1] 10
> c(0xA, 0xB, 12)
[1] 10 11 12
>
```

КОНЗОЛА

Вектор

комплексни бројеви задају преко компоненти

користи имагинарна јединица i

може задати и само имагинарна компонента

подржане операције над комплексним бројевима

```
> 1 + 2i
[1] 1+2i
> (1 + 2i) * (3 + 4i)
[1] -5+10i
> (-5 + 10i) / (3 + 4i)
[1] 1+2i
> sqrt(2i)
[1] 1+1i
>
```

КОНЗОЛА

Вектор

знаковне вредности

два начина означавања

помоћу наводника (")

помоћу апострофа (')

```
> "Gde?"  
[1] "Gde?"  
> 'Ovde'  
[1] "Ovde"  
> c("Levo", "Desno")  
[1] "Levo" "Desno"  
> paste("Levo", "Desno")  
[1] "Levo Desno"  
>
```

КОНЗОЛА

Вектор

знаковне вредности

посебни знакови обично означавају помоћу знака \

\\ \" \' \n \t итд.

могуће је уметати и знакове из стандарда *Unicode*

у облику `\uXXXX` или `\u{XXXX}`

XXXX – кôд знака у хексадецималном формату

```
> "\u{040b}"  
[1] "ћ"  
> "\u010ceka\u0107e \u011aci \u0161ume \u017eir!"  
[1] "Čekaće đaci šume žir!"  
>
```

КОНЗОЛА

Вектор

логичке вредности

TRUE FALSE

могуће је користити и скраћене ознаке

T F

```
> TRUE
[1] TRUE
> FALSE
[1] FALSE
> c(TRUE, FALSE, NA, T, F)
[1] TRUE FALSE NA TRUE FALSE
> T + F
[1] 1
>
```

КОНЗОЛА

Вектор

бајтовске вредности

за потребе бајтовске представе података
погодно за кодирање знакова
коришћење бинарног садржаја датотека

```
> charToRaw("Ah...")  
[1] 41 68 2e 2e 2e  
> rawToChar(as.raw(c(79, 105, 33)))  
[1] "oi!"  
>
```

КОНЗОЛА

Вектори – Индексирање

очитавање жељених вредности из вектора

настаје нови вектор који садржи очитане вредности

помоћу индексирајућег (индекс) вектора

вредности индексирајућег вектора одређују које вредности индексираног вектора треба очитати или прескочити при читавању

дозвољени облици индексирајућег вектора

вектор позитивних целих бројева

вектор негативних целих бројева

вектор логичких вредности

вектор знаковних вредности

Вектори – Индексирање

индексирајући вектор – вектор позитивних целих бројева

вредности индексирајућег вектора треба да буду у опсегу дозвољених индекса за индексирани вектор (од један до дужине индексираног вектора)

из индексираног вектора читавају вредности чији индекси одговарају вредностима индексирајућег вектора

```
> x <- c("A", "B", "C", "D", "E")
> x[c(1, 3, 5)]
[1] "A" "C" "E"
> x[c(1, 6)]
[1] "A" NA
> y <- c()
> y[1]
NULL
>
```

КОНЗОЛА

Вектори – Индексирање

индексирајући вектор – вектор негативних целих бројева

из индексираниог вектора читавају вредности чији индекси не одговарају апсолутним вредностима индексиралућег вектора

```
> x <- c("A", "B", "C", "D", "E")
> x[c(-2, -4)]
[1] "A" "C" "E"
> x[-c(2, 4)]
[1] "A" "C" "E"
> x[c(-6)]
[1] "A" "B" "C" "D" "E"
>
```

КОНЗОЛА

Вектори – Индексирање

индексирајући вектор – вектор логичких вредности

вредности индексираниог и индексирајућег вектора упарују по редоследу

за сваку вредност **TRUE** у индексирајућем вектору читава се њој одговарајућа вредност из индексираниог вектора

```
> w <- c(1, 2, 3, 4, 5)
> w[c(T, T, F, F, T)]
[1] 1 2 5
> w[c(T, T, F, F, T, T, T, F)]
[1] 1 2 5 NA NA
> w[c(T, F)]
[1] 1 3 5
> w[w >= 3]
[1] 3 4 5
>
```

КОНЗОЛА

Вектори – Индексирање

индексирајући вектор – вектор знаковних вредности

применљиво када индексирани вектор има придружене називе за садржане вредности

вредности индексирајућег вектора треба да садрже називе за вредности индексираног вектора које желе прочитати

```
> z <- c(59, 49, 49)
> names(z) <- c("P", "Q", "R")
> z[c("P", "Q")]
  P  Q
59 49
>
```

КОНЗОЛА

Вектори – Измене

помоћу оператора доделе

индексирање за потребе избора дела за измену

```
> u <- c(1, 3, 5, 3, 1)
> u[3] <- 7
> u[c(2, 4)] <- 5
> u
[1] 1 5 7 5 1
> u[u > 2] <- 2
> u
[1] 1 2 2 2 1
>
```

КОНЗОЛА

Вектори – Генерисање

формирање вектора помоћу оператора :

početna:krajnja

добија вектор уређених вредности

од вредности **početna** до вредности **krajnja**

```
> 1:5  
[1] 1 2 3 4 5  
> -1:-5  
[1] -1 -2 -3 -4 -5  
>
```

КОНЗОЛА

Вектори – Генерисање

формирање вектора помоћу функције **seq(...)**

добија вектор од почетне до крајње задате вредности

може задати размак између суседних вредности

може задати број жељених вредности из задатог опсега

```
> seq(1, 5)
[1] 1 2 3 4 5
> seq(from=0, to=100, by=10)
[1] 0 10 20 30 40 50 60 70 80 90 100
> seq(0, 100, 10)
[1] 0 10 20 30 40 50 60 70 80 90 100
> seq(from=0, to=1, length.out=5)
[1] 0.00 0.25 0.50 0.75 1.00
>
```

КОНЗОЛА

Вектори – Генерисање

формирање вектора помоћу функције **rep(...)**

умножавање задатог вектора

вектора као целине одређени број пута

појединачних вредности из вектора редом одређени број пута

```
> (v <- seq(5, 25, 5))  
[1] 5 10 15 20 25  
> rep(v, times=2)  
[1] 5 10 15 20 25 5 10 15 20 25  
> rep(v, 2)  
[1] 5 10 15 20 25 5 10 15 20 25  
> rep(v, each=2)  
[1] 5 5 10 10 15 15 20 20 25 25  
>
```

КОНЗОЛА

Вектори – Помоћне радње

дужина вектора и функција **length(...)**

очитавање дужине вектора

промена дужине вектора

```
> (k <- c("abc", "def", "ghj"))
[1] "abc" "def" "ghj"
> length(k)
[1] 3
> length(k) <- 2
> k
[1] "abc" "def"
> length(k) <- 5
> k
[1] "abc" "def" NA    NA    NA
>
```

КОНЗОЛА

Вектори – Помоћне радње

рачунање збира и производа свих вредности у вектору

функције **sum(...)** и **prod(...)**

одређивање најмање и највеће вредности у вектору

функције **min(...)**, **max(...)** и **range(...)**

```
> (p <- c(3, 1, 4))
[1] 3 1 4
> sum(p); prod(p)
[1] 8
[1] 12
> min(p); max(p); range(p)
[1] 1
[1] 4
[1] 1 4
>
```

КОНЗОЛА

Вектори – Помоћне радње

рачунање средње вредности, медијане и варијансе

функције `mean(...)`, `median(...)` и `var(...)`

одређивање задатих квантила

функција `quantile(...)`

```
> t <- c(27, 28, 24, 27, 25, 25, 29, 30)
> mean(t); median(t); var(t)
[1] 26.875
[1] 27
[1] 4.410714
> quantile(t, c(0.25, 0.5, 0.75))
 25%   50%   75%
25.00 27.00 28.25
>
```

КОНЗОЛА

Вектори – Помоћне радње

утврђивање присуства недостајућих вредности

функција **is.na(...)**

провера за сваку појединачну вредност у изворном вектору да ли недостајућа

функција **anyNA(...)**

провера да ли постоји бар једна недостајућа вредност у изворном вектору

```
> x <- c(5.3, NA, 2+3.7i, 2)
> is.na(x)
[1] FALSE TRUE FALSE FALSE
> anyNA(x)
[1] TRUE
> y <- c(-Inf, -1+NaN*1i, NA+73i, NULL, NA, NaN)
> is.na(y)
[1] FALSE TRUE TRUE TRUE TRUE
>
```

КОНЗОЛА

Вектори – Помоћне радње

утврђивање присуства вредности **TRUE**

функција **any(...)**

провера да ли у изворном вектору постоји бар једна вредност **TRUE**

функција **all(...)**

провера да ли су у изворном вектору све вредности вредност **TRUE**

```
> any(c(TRUE, FALSE))
[1] TRUE
> any(c(NA, FALSE))
[1] NA
> all(c(TRUE, FALSE))
[1] FALSE
> all(c(NA, FALSE))
[1] FALSE
>
```

КОНЗОЛА

Вектори – Помоћне радње

утврђивање присуства вредности **TRUE**

може применити у сложенијим случајевима

```
> p <- c(2, 4, 6, NA)
> any(is.na(p))
[1] TRUE
> anyNA(p)
[1] TRUE
> any(p > 0)
[1] TRUE
> all(p > 0)
[1] NA
> all(p > 0, na.rm=TRUE)
[1] TRUE
>
```

КОНЗОЛА

Вектори – Матрице

вектор с две димензије као матрица

формирање матрице помоћу функције `matrix(...)`

очитавање вредност матрице индексирањем

```
> mat <- matrix(c(1, 2, 3, 4), ncol = 2, nrow = 2)
> mat
      [,1] [,2]
[1,]    1    3
[2,]    2    4
> mat[2, 2]
[1] 4
> mat[, 2]
[1] 3 4
>
```

КОНЗОЛА

Вектори – Матрице

за матрице постоји више основних описних података
дужина, димензионалност, број врста и број колона

```
> mat <- matrix(c(1, 2, 3, 4), ncol = 2, nrow = 2)
> length(mat)
[1] 4
> dim(mat)
[1] 2 2
> nrow(mat)
[1] 2
> ncol(mat)
[1] 2
>
```

КОНЗОЛА

Вектори – Матрице

множење елемент по елемент

```
> (ma <- matrix(c(1, 2, 3, 4), ncol=2, nrow=2))
      [,1] [,2]
[1,]    1    3
[2,]    2    4
> (mb <- matrix(c(5, 6, 7, 8), ncol=2, nrow=2))
      [,1] [,2]
[1,]    5    7
[2,]    6    8
> ma * mb
      [,1] [,2]
[1,]    5   21
[2,]   12   32
>
```

КОНЗОЛА

Вектори – Матрице

множење матрица

```
> (ma <- matrix(c(1, 2, 3, 4), ncol=2, nrow=2))
      [,1] [,2]
[1,]    1    3
[2,]    2    4
> (mb <- matrix(c(5, 6, 7, 8), ncol=2, nrow=2))
      [,1] [,2]
[1,]    5    7
[2,]    6    8
> ma %*% mb
      [,1] [,2]
[1,]   23   31
[2,]   34   46
>
```

КОНЗОЛА

Вектори – Матрице

решавање система линеарних једначина

помоћу функције `solve(...)`

```
> # x - 2y = 0
> # x + y = 3
> (A <- matrix(c(1, 1, -2, 1), ncol=2, nrow=2))
      [,1] [,2]
[1,]    1  -2
[2,]    1   1
> (c <- c(0, 3))
[1] 0 3
> solve(A, c)
[1] 2 1
>
```

КОНЗОЛА

Вектори – Матрице

спајање матрица по колонама помоћу функције **`cbind(...)`**

```
> (ma <- matrix(c(1, 2, 3, 4), ncol=2, nrow=2))
      [,1] [,2]
[1,]    1    3
[2,]    2    4
> (mb <- matrix(c(5, 6, 7, 8), ncol=2, nrow=2))
      [,1] [,2]
[1,]    5    7
[2,]    6    8
> cbind(ma, mb)
      [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8
>
```

КОНЗОЛА

Вектори – Матрице

спајање матрица по врстама помоћу функције **rbind(...)**

```
> (ma <- matrix(c(1, 2), ncol=2, nrow=1))
      [,1] [,2]
[1,]    1    2
> (mb <- matrix(c(3, 4), ncol=2, nrow=1))
      [,1] [,2]
[1,]    3    4
> rbind(ma, mb)
      [,1] [,2]
[1,]    1    2
[2,]    3    4
>
```

КОНЗОЛА

Вектори – Вишедимензионални низови

вектор с више од две димензије као вишедимензионални низ

формирање вишедимензионалног низа помоћу функције **array(...)**

```
> (vdn <- array(1:6, dim=c(1, 3, 2)))
```

```
, , 1
```

```
      [,1] [,2] [,3]
[1,]    1    2    3
```

```
, , 2
```

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

```
>
```

КОНЗОЛА

Вектори – Вишедимензионални низови

очитавање вредности вишедимензионалног низа индексирањем
вишедимензионални низ поседује дужину и димензионалност

```
> vdn <- array(1:6, dim=c(1, 3, 2))
> vdn[1, 3, 2]
[1] 6
> vdn[, 2, ]
[1] 2 5
> length(vdn)
[1] 6
> dim(vdn)
[1] 1 3 2
>
```

КОНЗОЛА

Листа

уређена колекција компоненти

компоненте су објекти

компоненте не морају бити истог типа

компоненте могу имати називе

формирање листе

функција **list(...)**

наводе компоненте листе

могу задати називи за компоненте листе

```
> p <- list("Novi Sad", 27, 1010)
> r <- list(grad="Novi Sad", temp=27, prit=1010)
>
```

КОНЗОЛА

Листа

приступање деловима листе

на основу позиције или назива компоненте

може користити пуни или скраћени назив компоненте

употреба оператора

помоћу оператора `[]`

приступ компоненти на основу позиције или назива компоненте

помоћу оператора `[]`

приступ подлисти на основу позиције или називе компоненте

помоћу оператора `$`

приступ компоненти на основу назива компоненте

Листа

```
> ns <- list(naz="Novi Sad", geo=c(45.255,19.845))
> ns[[1]]
[1] "Novi Sad"
> ns[[2]][1]
[1] 45.255
> ns[["geo"]]
[1] 45.255 19.845
> ns$geo[1]
[1] 45.255
> ns[2]
$geo
[1] 45.255 19.845
>
```

КОНЗОЛА

Листа – Измене

```
> t <- list(tim="Veseli veprovi", pob=10)
> t$pob <- t$pob + 1
> t$por <- 5
> t
$tim
[1] "Veseli veprovi"

$pob
[1] 11

$por
[1] 5

>
```

КОНЗОЛА

Листа – Спајање

помоћу функције `c(...)`

компоненте свих задатих листа смештају у једну листу

```
> kr <- list(real=2.4)
> ki <- list(imag=8.1)
> k <- c(kr, ki)
> k
$real
[1] 2.4

$imag
[1] 8.1

>
```

КОНЗОЛА

NULL

посебан објекат

означава недостајући објекат

постоји само једна инстанца

иста инстанца се свуда користи

Фактор

подршка за категоријска обележја

дозвољене вредности само оне које већ постоје или које унапред задате
основне функције за факторе

формирање фактора помоћу функције **factor(...)**

очитавање дозвољених вредности помоћу функције **levels(...)**

провера постојања поретка помоћу функције **is.ordered(...)**

```
> (f <- factor(c("levo", "dole", "x")))
[1] levo dole x
Levels: dole levo x
> levels(f)
[1] "dole" "levo" "x"
> is.ordered(f)
[1] FALSE
>
```

КОНЗОЛА

Фактор – Уређени фактори

могућност увођења поретка између дозвољених вредности

функција **ordered(...)**

поредак дозвољених вредности имплицитно по алфавитном уређењу
поредак дозвољених вредности експлицитно по задатом редоследу

```
> of <- ordered(c("mozda", "ne", "da", "da"))
> of
[1] mozda ne    da    da
Levels: da < mozda < ne
> of <- ordered(c("mozda", "ne", "da", "da"),
levels=c("ne", "mozda", "da"))
> of
[1] mozda ne    da    da
Levels: ne < mozda < da
>
```

КОНЗОЛА

Скуп података

колекција повезаних обележја и њихових вредности

енгл. *data frame*

подршка за рад над подацима који могу табеларно представити појаве описане преко вредности обележја

колоне одговарају обележјима и вредностима обележја

редови одговарају појавама и одговарајућим вредностима обележја

колона може бити

вектор (бројчани, знаковни или логички)

фактор

бројчана матрица

листа

скуп података

скуп података је прилагођење листе

колоне су компоненте листе

Скуп података – Пример

скуп података **bugarska**

подаци о броју становника за највеће градове у Бугарској

Попис становништва у Бугарској из 2011. године

<https://www.nsi.bg/census2011/indexen.html>

<https://www.nsi.bg/census2011/pageen29a73.html?p2=179&sp2=209>

главни резултати пописа (од 1. 2. 2011)

https://www.nsi.bg/census2011/PDOCS2/Census2011final_en.pdf

страница 14

grad	stan
Sofija	1204685
Plovdiv	338153
Varna	334870
Burgas	200271
Ruse	149642

Објекти

Скуп података

формирање помоћу функције `data.frame(...)`

```
> grad <- c("Sofija", "Plovdiv", "Varna",  
"Burgas", "Ruse")  
> stan <- c(1204685, 338153, 334870, 200271,  
149642)  
> sp <- data.frame(grad, stan)  
> sp
```

	grad	stan
1	Sofija	1204685
2	Plovdiv	338153
3	Varna	334870
4	Burgas	200271
5	Ruse	149642

```
>
```

КОНЗОЛА

Скуп података

очитавање података

слично као код матрице или листе

```
> sp <- data.frame(grad=c("Sofija", "Plovdiv",  
"Varna", "Burgas", "Ruse"), stan=c(1204685,  
338153, 334870, 200271, 149642))  
> sp[1, 2]  
[1] 1204685  
> sp[3, ]  
  grad  stan  
3 Varna 334870  
> sp$grad  
[1] Sofija Plovdiv Varna Burgas Ruse  
Levels: Burgas Plovdiv Ruse Sofija Varna  
>
```

КОНЗОЛА

Скуп података – Измена

```
> sp <- data.frame(grad=c("Sofija", "Plovdiv",  
"Varna", "Burgas", "Ruse"), stan=c(1204685,  
338153, 334870, 200271, 149642))  
> sp <- sp[-5, ]  
> sp$more <- factor(c("ne", "ne", "da", "da"))  
> sp$stan[1] <- NA  
> sp$grad <- NULL  
> sp  
      stan more  
1      NA   ne  
2 338153   ne  
3 334870   da  
4 200271   da  
>
```

КОНЗОЛА

Типови објеката

подаци о типу

тип објекта

тип у оквиру језика R

облик објекта

облик сличан типу

углавном исти

компатибилност с језиком S

облик складиштења објекта

облик складиштења сличан облику

углавном исти

компатибилност с језиком S и другим језицима

Типови објеката

очитавање података о типу објекта

тип објекта помоћу функције **typeof(...)**

облик објекта помоћу функције **mode(...)**

облик складиштења објекта помоћу функције **storage.mode(...)**

```
> v <- c(1L, 2L)
> typeof(v)
[1] "integer"
> mode(v)
[1] "numeric"
> storage.mode(v)
[1] "integer"
>
```

КОНЗОЛА

Садржај

1. Језик R
2. Објекти
- 3. Функције**
4. Контрола тока
5. Пакети
6. Синтакса
7. Извори и литература

Функција

сврстава се у објекте

ОСНОВНЕ КОМПОНЕНТЕ

листа аргумената

тело

окружење

```
> function(x) x^3  
function(x) x^3  
> koren <- function(x) {sqrt(x)}  
>
```

КОНЗОЛА

Функција

листа аргумената

аргумент поседује назив

аргумент може имати подразумевану вредност

вредности аргумената доступне у телу функције преко назива аргумената

листа аргумената не мора бити потпуна

по потреби функцији могу проследити додатни аргументи

како би то било могуће у листу аргумената додаје посебан објекат

... (три тачке)

```
> f <- function(x, y=2) {x + y}
> function(a, b, ...) {a + b}
function(a, b, ...) {a + b}
>
```

КОНЗОЛА

Функције

Функција

тело функције

садржи изразе

омеђено витичастим заградама, није неопходно ако садржан један израз

```
> function(x) 3 * x
function(x) 3 * x
> function(x) {3 * x; 2 * x}
function(x) {3 * x; 2 * x}
> function(x) {
+ 3 * x
+ }
function(x) {
3 * x
}
>
```

КОНЗОЛА

Функција

вредност функције

повратна вредност

може експлицитно назначити

користи позив **return** путем којег се поставља вредност функције

без позива **return** вредност функције је вредност последњег евалуираног израза

ако нема евалуираног израза вредност функције је **NULL**

```
> function(x) {return(x^3)}  
function(x) {return(x^3)}  
> function(x) {x^3}  
function(x) {x^3}  
> function(x) {}  
function(x) {}  
>
```

КОНЗОЛА

Функција

постављање аргумената при позиву функције

по називу аргумента

по скраћеном називу аргумента

по очекиваном редоследу аргумената

```
> f <- function(osn, ste=2) {osn^ste}
> f(osn=2)
[1] 4
> f(osn=3, ste=3)
[1] 27
> f(o=4, s=4)
[1] 256
> f(5, 5)
[1] 3125
>
```

КОНЗОЛА

Функције

Функција

функција може бити анонимна

функција може да користи друге функције

аргумент функције може бити функција

```
> (function(x) {x^x})(-1)
[1] -1
> f <- function(x, y) {y(abs(x))}
> f(-3, sqrt)
[1] 1.732051
> f(5, function(x) {-x*x})
[1] -25
>
```

КОНЗОЛА

Функција

додатни аргументи

у листи аргумената функције означени помоћу резервисане речи `...`

број додатних аргумената доступан помоћу функције `...length()`

додатни аргументи доступни помоћу резервисане речи `..X`

X – редни број додатног аргумента, $X \in \{1, 2, \dots\}$

```
> f <- function(x, ...) {print(...length()); ..1}
> f("a")
[1] 0
Error in f("a") : the ... list does not contain
any elements
> f("a", "b")
[1] 1
[1] "b"
>
```

КОНЗОЛА

Функција

додатни аргументи

додатни аргументи могу бити прихваћени у листу

```
> f <- function(x, ...) {ar <- list(...)}
```

```
> (f(1, 2, 3))
```

```
[[1]]
```

```
[1] 2
```

```
[[2]]
```

```
[1] 3
```

```
>
```

КОНЗОЛА

Функције

Функција

додатни аргументи

додатни аргументи могу прослеђени новом позиву функције

```
> f <- function(x, ...) (print(...))  
> g <- function(x, ...) {print(x); f(...)}  
> g(1, 2, 3)  
[1] 1  
[1] 3  
[1] 3  
>
```

КОНЗОЛА

Оператор

функција која може на посебан начин позивати

преко посебних ознака

може бити унаран или бинаран

основне групе оператора

аритметички

логички

релациони

доделе

секвенце

разрешења досега

индексирања

позива функције

...

Оператор

уграђени оператор је у основи функција
може позивати као функција у префиксној нотацији
``operator` (lista-argumenata)`

```
> `+`  
function (e1, e2) .Primitive("+")  
> `==`  
function (e1, e2) .Primitive("==")  
> `==` (1, 11)  
[1] FALSE  
>
```

КОНЗОЛА

Оператори – Проширења

могућност увођења нових бинарних оператора

користи посебан оператор

у облику **%operator%**

део **operator** може заменити знаковима који карактеришу нови оператор

потребно дефинисати понашање новог оператора

не треба преклапати уграђене операторе

може имати негативне последице

```
> "%+^%" <- function(x, y){(x + y)^(x + y)}  
> 1 %+^% 2  
[1] 27  
>
```

КОНЗОЛА

Оператори – Правила

правила груписања

- груписање слева надесно

 - велика већина оператора

- груписање здесна налево

 - оператори доделе, степеновања

правила приоритета

- посебно одређен ниво приоритета за сваки оператор

Оператори – Основни приоритети

Ниво	Оператори	Ниво	Оператори
1 (највиши)	::	9	> >= < <= == !=
2	\$ @	10	!
3	^	11	& &&
4	- + (унарни)	12	
5	:	13	~
6	% <u>abc</u> %	14	-> ->>
7	* /	15	=
8	+ - (бинарни)	16 (најнижи)	<- <<-

Садржај

1. Језик R
2. Објекти
3. Функције
- 4. Контрола тока**
5. Пакети
6. Синтакса
7. Извори и литература

Гранања

врсте гранања

гранање **if**

гранање **ifelse**

гранање **switch**

Гранања

гранање `if`

основне варијанте

```
if (izraz-da-li-jeste) izraz-ako-da
```

```
if (izraz-da-li-jeste) izraz-ako-da else izraz-ako-ne
```

вредност за израз `izraz-da-li-jeste` мора бити или **TRUE** или **FALSE** може бити и вектор логичких вредности али тада узима у обзир само прва вредност

```
> if(TRUE) "jeste"
[1] "jeste"
> if(TRUE) {"jeste"; "zaista"}
[1] "zaista"
> if (5 %% 2 == 1) "neparan" else "paran"
[1] "neparan"
>
```

КОНЗОЛА

Гранања

гранање `ifelse`

користи се функција `ifelse(...)`

основни облик

`ifelse(izraz-da-li-jeste, izraz-ako-da, izraz-ako-ne)`

израз `izraz-da-li-jeste` је вектор логичких вредности

вредност функције је вектор

за сваку вредност `TRUE` у `izraz-da-li-jeste`

умеће се једна вредност израза `izraz-ako-da`

за сваку вредност `FALSE` у `izraz-da-li-jeste`

умеће се једна вредност израза `izraz-ako-ne`

```
> ifelse(c(T, F, T), "stvarno", "nestvarno")  
[1] "stvarno" "nestvarno" "stvarno"  
>
```

КОНЗОЛА

Гранања

гранање `switch`

користи се функција `switch(...)`

`switch (izraz, ...)`

могуће вредности функције

за знаковну вредност израза `izraz`

вредност аргумента чији назив одговара вредности израза `izraz`

за бројчану вредност израза `izraz`

вредност аргумента на позицији за један већој од вредности израза `izraz`

ако није одабрана ниједна вредност међу аргументима

објекат `NULL`

```
> switch("a", x=1, y=2, z=3, 0)
```

```
[1] 0
```

```
> switch(5-3, "a", "b", "c")
```

```
[1] "b"
```

```
>
```

КОНЗОЛА

Петље

врсте петљи

петља **for**

петља **while**

петља **repeat**

Петље

додатна контрола извршавања петље

break

прекида извршавање матичне петље

може користити у било којој врсти петље

неопходно да ограничи број понављања у петљи **repeat**

next

прекида извршавање тренутног понављања матичне петље

након тога прелази се на ново понављање

Петље

петља **for**

основни облик

```
for(naziv in izraz-vrednosti) izraz-telo
```

понашање

за **naziv** једна по једна редом вежу вредности вектора **izraz-vrednosti**
за сваку од вредности везаних за **naziv** евалуира се израз **izraz-telo**
izraz-vrednosti може бити и листа

```
> for(i in 1:3) {  
+ print(i)  
+ }  
[1] 1  
[1] 2  
[1] 3  
>
```

КОНЗОЛА

Петље

петља **while**

основни облик

```
while (izraz-kontrola) izraz-telo
```

понашање

евалуира се израз **izraz-telo**

докле год израз **izraz-kontrola** има вредност **TRUE**

```
> while (TRUE != FALSE) {print("istina"); break}  
[1] "istina"  
>
```

КОНЗОЛА

Петље

петља **repeat**

основни облик

```
repeat izraz-telo
```

понашање

понавља се евалуирање израза **izraz-telo**

може искористи **break** ради прекидања понављања

```
> repeat {if (TRUE) {print("istina"); break} else  
print("neistina")}  
[1] "istina"  
>
```

КОНЗОЛА

Садржај

1. Језик R
2. Објекти
3. Функције
4. Контрола тока
- 5. Пакети**
6. Синтакса
7. Извори и литература

Пакет

обједињује додатне функције и скупове података, помоћне датотеке и пратећу документацију

обједињавање у пакет по неком основу

најчешће имплементација онога што је неопходно за одређене врсте анализе

једноставно се дели, преноси, инсталира и учитава

пакет као једна архивска датотека

пакет може захтевати друге пакете

када је пакет учитан, садржај пакета је на располагању за употребу

пре учитавања пакет мора бити инсталиран

Основни пакети

обухваћени системом R и редовном инсталацијом
није потребно додатно их инсталирати

30 основних пакета

пакет **base** садржи основне елементе потребне за језик R
употреба основних пакета

поједини основни пакети се аутоматски читавају
могу користити без додатних припрема
пакет **base** и шест подразумеваних пакета

преглед основних пакета помоћу позива `getOption("defaultPackages")`
поједини основни пакети захтевају експлицитно читавање

```
> getOption("defaultPackages")  
[1] "datasets"  "utils"      "grDevices" "graphics"  
"stats"      "methods"
```

КОНЗОЛА

Преглед пакета

тренутно учитани пакети

помоћу позива функције **.packages()**

тренутно инсталирани пакети

помоћу позива **.packages(all.available=TRUE)**

```
> (.packages())  
[1] "stats"      "graphics"   "grDevices"  "utils"  
    "datasets"  "methods"  
[7] "base"  
>
```

КОНЗОЛА

Преглед пакета

```
> .packages(all.available=TRUE)
 [1] "base"           "boot"           "class"
"cluster"        "codetools"
 [6] "compiler"       "datasets"       "foreign"
"graphics"       "grDevices"
[11] "grid"           "KernSmooth"    "lattice"
"MASS"           "Matrix"
[16] "methods"        "mgcv"           "nlme"
"nnet"           "parallel"
[21] "rpart"          "spatial"        "splines"
"stats"          "stats4"
[26] "survival"       "tcltk"          "tools"
"translations"  "utils"
```

```
>
```

КОНЗОЛА

Учитавање пакета

стављање садржаја инсталираног пакета на располагање

помоћу функције `library(naziv-paketa)` или
помоћу функције `require(naziv-paketa)`

```
> library(rpart)
> require(stats3)
Loading required package: stats3
Warning message:
In library(package, lib.loc = lib.loc,
character.only = TRUE, logical.return = TRUE,  :
  there is no package called 'stats3'
>
```

КОНЗОЛА

Додатни пакети

захтевају инсталацију

пакети начелно инсталирају у локалну библиотеку

место на локалном систему где налазе датотеке инсталираних пакета

пакети начелно доступни у репозиторијумима

главни репозиторијум

CRAN (Comprehensive R Archive Network)

<https://cran.r-project.org/web/packages/index.html>

22323 пакета доступна (на дан 18. 6. 2025)

додатни репозиторијуми

Bioconductor

<https://www.bioconductor.org/>

GitHub

<https://github.com/>

...

Инсталација пакета

инсталација пакета из репозиторијума

помоћу **`install.packages("naziv-paketa")`**

подразумева преузимање пакета из репозиторијума

подразумева инсталацију пакета у локалну библиотеку

подразумева употребу репозиторијума *CRAN*

употреба одабраног *мирора*

деинсталација пакета

помоћу **`remove.packages("naziv-paketa")`**

деинсталација пакета из локалне библиотеке

Пакет – Помоћне информације

информације на жељену тему

help(naziv) или **?naziv**

приказ одговарајућег дела документације

naziv може бити назив пакета, функције, ...

?"naziv" или **?`naziv`**

приказ одговарајућег дела документације

naziv може бити резервисана реч или оператор

```
> help(stats)
> help(var)
> ?typeof
> ?"next"
> ?`$`
>
```

КОНЗОЛА

Пакет – Помоћне информације

додатне информације о пакету или функцији

`help.search("naziv")` или **`??naziv`**

претраживање документације по појму **`naziv`**

`library(help="naziv")`

основне информације и преглед садржаја пакета чији је назив **`naziv`**

`example(naziv)`

учитавање примера из пратеће документације

`vignette(naziv)`

приказ посебног дела из пратеће документације који садржи упутства и примере

```
> help.search("correlation")
> ??chisquare
> ??"fisher test"
> library(help="stats")
>
```

КОНЗОЛА

Пакет – Помоћне информације

```
> example(chisq.test)
```

```
chsq.t> ## From Agresti(2007) p.39
```

```
chsq.t> M <- as.table(rbind(c(762, 327, 468),  
c(484, 239, 477)))
```

```
chsq.t> dimnames(M) <- list(gender = c("F", "M"),  
chsq.t+           party =  
c("Democrat", "Independent", "Republican"))
```

```
chsq.t> (Xsq <- chisq.test(M)) # Prints test  
summary
```

```
Pearson's Chi-squared test
```

КОНЗОЛА

Садржај

1. Језик R
2. Објекти
3. Функције
4. Контрола тока
5. Пакети
- 6. Синтакса**
7. Извори и литература

Синтакса језика R

релативно једноставна

потребно водити рачуна о разлици између малих и великих слова

дозвољени скуп знакова

зависи од подешавања локала

начелно обухваћени

обични алфанумерички знакови

тачка

доња црта

правила за називе

назив почиње или словом или тачком

ако назив почиње тачком онда други знак у називу не сме бити цифра

дужина назива није посебно ограничена

не користити резервисане речи као називе

Резервисане речи

if	else	repeat	while
function	for	in	next
break	TRUE	FALSE	NULL
Inf	NaN	NA	NA_integer_
NA_real_	NA_complex_	NA_character_	...
..1	..2	<i>(и тако редом)</i>	

Синтакса језика R

употреба коментара

коментари задају на нивоу једног реда

коментар целокупног реда

коментар од произвољног места у реду до краја реда

почетак коментара обележава знаком **#**

Racunanje reziduala

Синтакса језика R

евалуација израза у више редова

интерпретер начелно покушава да пронађе исправан завршетак израза

прво у тренутном реду

ако није могуће онда у наредном

знаковне вредности у више редова

очекује исправан затварајући знак који подудара с отварајућим знаком

```
> if(TRUE)
+ print("jeste")
[1] "jeste"
> "Sto
+ posto!"
[1] "Sto\nposto!"
>
```

КОНЗОЛА

Стил кодирања

The tidyverse Style Guide

<https://style.tidyverse.org/>

Google's R Style Guide

<https://google.github.io/styleguide/Rguide.html>

Advanced R. Style Guide (Hadley Wickham)

<http://adv-r.had.co.nz/Style.html>

R. Style. An Rchaeological Commentary (Paul E. Johnson)

<https://cran.r-project.org/web/packages/rockchalk/vignettes/Rstyle.pdf>

...

Стил кодирања – *The tidyverse Style Guide* (извод)

датотека изворног кода

користити `.R` као екстензију

дати смислен назив

дужина реда

требало би да буде највише 80 знакова

називи у програмском коду

називи променљивих и функција

употреба искључиво малих слова, цифара и доње црте

употреба доње црте за раздвајање више речи у називу

пример за назив променљиве

sezona_let

називи променљивих да буду именице

називи функција да буду глаголи

тежити концизним и смисленим називима

Стил кодирања – *The tidyverse Style Guide* (извод)

увлачење програмског кода у контроли тока

два размака

употреба размака

већина инфиксних оператора да буде окружена размацима

оператори високог приоритета да не буду окружени размацима

примери оператора високог приоритета

:: \$ @ - (унарни) + (унарни) :

употреба витичастих заграда

лева витичаста заграда да буде последњи знак у реду

десна витичаста заграда да буде први знак у реду

садржај између витичастих заграда да буде увучен за два размака

употреба тачке са запетом (;)

не стављати тачку са запетом на крај реда

не користити тачку са запетом за потребе смештања више наредби у један ред

Стил кодирања – *The tidyverse Style Guide* (извод)

означавање знаковних вредности

употреба знака " а не знака '

осим када знаковни садржај обухвата знакове " и не обухвата знакове '

додела вредности

користити оператор <-

не користити оператор =

коментари

сваки ред коментара почети знаком за коментар # и једним размаком

Важна је доследност!

Садржај

1. Језик R
2. Објекти
3. Функције
4. Контрола тока
5. Пакети
6. Синтакса
- 7. Извори и литература**

Основни извори и литература

- ◆ R Project. An introduction to R – Notes on R: A programming environment for data analysis and graphics version 4.5.1 (2025-06-13). Internet: <https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>
- ◆ R Project. R language definition – Version 4.5.1 (2025-06-13) DRAFT. Internet: <https://cran.r-project.org/doc/manuals/r-release/R-lang.pdf>
- ◆ R Project. R: A language and environment for statistical computing – Reference index – The R Core Team – Version 4.5.1 (2025-06-13). Internet: <https://cran.r-project.org/doc/manuals/r-release/fullrefman.pdf>
- ◆ Adler J. R in a nutshell: A desktop quick reference. 2nd edition. O'Reilly; 2012.
- ◆ Wickham H. The tidyverse style guide. Internet: <https://style.tidyverse.org/>

Додатни извори и литература

- ◆ Wickham H. Advanced R. 2nd edition. Chapman & Hall / CRC Press; 2019. Internet: <https://adv-r.hadley.nz/index.html>

Извори података

- ◆ скуп података **bugarska**
 - ◆ Попис становништва у Бугарској из 2011. године
 - ◆ <https://www.nsi.bg/census2011/indexen.html>
 - ◆ <https://www.nsi.bg/census2011/pageen29a73.html?p2=179&sp2=209>
 - ◆ главни резултати пописа (од 1. 2. 2011)
 - ◆ https://www.nsi.bg/census2011/PDOCS2/Census2011final_en.pdf
 - ◆ подаци о броју становника за највеће градове у Бугарској (страна 14)

Мастер академске студије
Информациони инжењеринг

Статистика у информационом инжењерингу

Основни концепти језика R

(материјали за предавања)