



Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

# Programiranje v R-ju

## 1. Uvod

Vladimir Batagelj

Univerza v Ljubljani, FMF, Matematika

Finančna matematika  
Ljubljana, februar 2009  
1. oktober 2012



Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

- 1 Projekt R
- 2 Okolje R
- 3 Spremenljivke
- 4 Prireditveni stavek
- 5 Osnovne zvrsti
- 6 Branje in izpis



# Programski jezik in okolje R

## Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis



Ross Ihaka in Robert Gentleman na DSC 2001

V tem tečaju se bomo seznanili z osnovami programiranja v programskem jeziku/okolju R.

R sta sredi devetdesetih let razvila Robert Gentleman in Ross Ihaka z univerze v Aucklandu v Novi Zelandiji. Nastal je kot odprt kodna izvedba programskega jezika za statistiko S, ki ga je leta 1976 razvil John Chambers s sodelavci z Bell Laboratories. Tržna izvedba jezika S je znana kot S-Plus.

Projektu R se je pridružila vrsta sodelavcev z vsega sveta in tako je R sčasoma postal jezik, v katerem se razvija in objavlja večina novih statističnih metod.



# Zakaj R ?

## Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

- R je prost – uporabljamo ga lahko brezplačno, namestimo ga lahko tudi na domačem računalniku. Teče pod OS: Windows, Linux/Unix in na Mac.
- R je odprtokodni program – za vse stvari lahko pogledamo, kako so narejene in jih po potrebi tudi spremenimo; pri razvoju sodeluje veliko sodelavcev iz vsega sveta in z različnih področij, ki so ustvarili že čez 2000 paketov – knjižnic za reševanje posebnih problemov (**R-Forge**).
- omogoča visokokakovostne slikovne prikaze podatkov in rezultatov.
- R je nastal kot programski jezik za analizo podatkov in statistiko; je pa uporaben tudi širše: podpora odločanja, biokemija, ...



# Viri

## Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

- Projekt R: <http://www.r-project.org/>
- CRAN (The Comprehensive R Archive Network):  
<http://cran.at.r-project.org/>
- spletna revija R-news: <http://cran.r-project.org/doc/Rnews/>
- viri podatkov: DASL (The Data and Story Library)  
<http://lib.stat.cmu.edu/DASL/>, The world factbook  
<https://www.cia.gov/library/publications/the-world-factbook/>  
The UCI Knowledge Discovery in Databases Archive  
<http://kdd.ics.uci.edu/>
- konferenca UseR! :  
<http://www.r-project.org/conferences.html>
- **Prikazi podatkov v R-ju**
- Ashlee Vance: **Data Analysts Captivated by R's Power**. The New York Times, 6. jan. 2009



# Programski jeziki

## Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

Algoritme pripravimo za uporabo na računalniku tako, da jih zapišemo v izbranem programskem jeziku. Ta mora omogočati nedvoumen in pravilen zapis algoritmov. Zato moramo pri izgradnji programskega jezika natančno določiti, katera zaporedja znakov so programi — določiti moramo *slovnico* ali sintakso jezika. Poleg tega moramo za posamezne sestavine jezika natančno določiti njihov *učinek/pomen* — semantika jezika. Računalnik 'razume' le svoj *strojni jezik*, ki je vgrajen v računsko-krmilni enoti.

Običajno je na posamezni vrsti računalnika dostopen še *zbirni jezik*, ki je zelo blizu strojnemu jeziku, le zapisan je v ljudem lažje berljivi obliki. Večina programov je zapisanih v *višjih* programskih jezikih (C, java, python, javascript, perl, R, mathematica, pascal, fortran, basic, logo, lisp, . . .).



## Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

Pomemben primer dvoumnosti je vrstni red izračuna vrednosti aritmetičnega izraza. Kaj pomenita izraza  $3 + 4 * 5$  in  $3 - 4 - 5$  ?

Ali smo mislili  $(3 + 4) * 5$  ali  $3 + (4 * 5)$  oziroma  $(3 - 4) - 5$  ali  $3 - (4 - 5)$  ?

Tudi v programskih jezikih običajno (z zelo redkimi izjemami) uporabljamo ista pravila kot v matematiki:

Če vrstni red izračuna izraza ni določen z *oklepaji*, upoštevamo dogovor o *prednostnem vrstnem redu*; če ta odpove — imamo opravka z operacijami enake prednosti — pa opravimo izračun *z leve proti desni*.

Torej izraza pomenita  $3 + (4 * 5)$  in  $(3 - 4) - 5$ .



# Prevajanje in tolmačenje

## Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

Obstajata dva osnovna načina, kako lahko dani program izvedemo na računalniku:

**Prevajanje:** Za izbrani programski jezik je bil napisan poseben program — prevajalnik. Ta dani program prevede v enakovreden program v strojnem jeziku. Prevod shrani na posebno datoteko (EXE). Operacijski sistem računalnika zna tako datoteko naložiti v pomnilnik in sprožiti njeno izvajanje. Odslej uporabljamo prevod.

Pri prevajanju prevajalnik preveri slovnično pravilnost programa — prevede ga le, če je pravilen; sicer nas obvesti o napakah.

**Tolmačenje:** Za izvajanje programov potrebujemo poseben program — tolmač. Ta ponavlja naslednje: razčleni tekoči stavek in preveri njegovo pravilnost. Če je pravilen, izvede zahtevana opravila; sicer sporoči napako.

- prevajanje omogoča hitrejše izvajanje programov;
- tolmačenje omogoča lažje učenje in razvoj programov;
- v tolmačenih programskih jezikih je meja med podatki in programom prehodna.





# Okolje R

Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

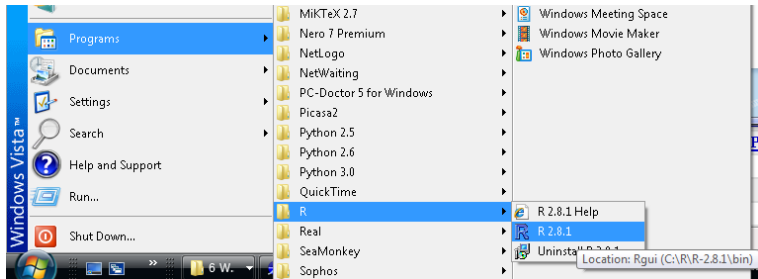
Osnovne zvrsti

Branje in izpis

R je interaktivno okolje za statistične izračune. Njegovo jedro je *tolmač* za programski jezik R, ki omogoča zapis korakov izračuna.

Jezik R temelji na *izrazih*. Tolmač za R *bere* izraze, jih *izračuna* in *izpiše/prikaže* njihove vrednosti.

Poženimo tolmač za R !





# Okolje R

## Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

Pojavi se nam pogovorno okno `R Console` z osnovnimi podatki o različici okolja R in kratkimi navodili. Konča se s *pozivnikom* `>`, ki nam sporoča, da tolmač pričakuje naslednji izraz.

Tolmač zapustimo z izrazom `q()`, znakom `<Ctrl>Z`, izbiro `File/Exit` ali klikom na gumb `x` programskega okna.

Podrobneje bomo zgradbo izrazov (slovnico jezika R) spoznali v nadaljevanju. Primeri preprostih izrazov so matematični izrazi, ki jih srečamo v raznih računih in obrazcih – tolmač za R lahko uporabljamo kot računalo. Poglejmo si nekaj primerov.



# Okolje R kot računalo

Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

```
> 1+1
[1] 2
> 3^2 + 4^2
[1] 25
> 3 + 4 * 5
[1] 23
> (3 + 4)*5
[1] 35
> pi
[1] 3.141593
> pi * 2.5^2
[1] 19.63495
> 4*atan(1)
[1] 3.141593
> (1+sqrt(5))/2
[1] 1.618034
```



# Številске operacije in funkcije

Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

$x + y$

$x - y$

$x * y$

$x / y$

$x \wedge y$

$x \% y$  =  $x \bmod y$

$x \% \% y$  =  $x \operatorname{div} y$

`sign(x)`

`abs(x)`

`sqrt(x)`

`log(x, base = exp(1))`

`log10(x)`

`log2(x)`

`log1p(x) = log(1+x)`

`exp(x)`

`expm1(x) = exp(x) - 1`

`ceiling(x)`

`floor(x)`

`trunc(x, ...)`

`round(x, digits = 0)`

`signif(x, digits = 6)`

`zapsmall(x, digits = getOption("digits"))`

`min(x,y)`

`max(x,y)`

`cos(x)`

`sin(x)`

`tan(x)`

`acos(x)`

`asin(x)`

`atan(x)`

`atan2(y,x) = atan(y/x)`

Za podrobnosti o posameznih funkcijah pogledjte

`help(Arithmetic)`, `help(Trig)`, `help(complex)`,

`help(log)`, `help(Round)`, `help(sqrt)`, ...



# Spremenljivke

## Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

Naloga razmestitve količin v pomnilniku je v večini višjih programskih jezikov prepuščena prevajalniku/tolmaču. S tem je povezan pojem *spremenljivke*, ki združuje štiri sestavine: *ime*, *zvrst* ali tip, *prostor* in *vrednost*.

Če hočemo v programu s spremenljivko kaj početi, uporabimo za to njeno ime. Ime spremenljivke prevajalnik/tolmač poveže z njej pripadajočim prostorom v pomnilniku. Vsebina tega prostora je vrednost spremenljivke. Zvrst spremenljivke pa določa: velikost spremenljivki pripadajočega prostora; vrednosti, ki jih lahko vsebuje; kako je potrebno razumeti vsebino prostora; in katere operacije so nad spremenljivko dovoljene. Seveda v danem trenutku izvajanja programa, zaradi nedvoumnosti, v danem okolju več spremenljivk ne more imeti istega imena; v nekaterih jezikih pa lahko več spremenljivkam (imenom) pripada isti prostor.



# Imena

## Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

V R-ju so *imena* zaporedja črk, števk, pik . in podčrtajev \_.  
Ime se ne sme začeti s števkjo ali podčrtajem \_, niti s piko ., ki  
ji sledi števka.

*Rezervirane besede*: if, else, repeat, while, function,  
for, in, next, break, TRUE, FALSE, NULL, Inf, NaN,  
NA, NA\_integer\_, NA\_real\_, NA\_complex\_,  
NA\_character\_ ter ..., ..1, ..2, ..3, ... niso imena.  
NaN – *neštevilo* (Not a number); NA – *nedoločeno* neznana,  
manjkajoča vrednost (not available); Inf – *neskončno*  
(Infinity).

V imenih R razlikuje med velikimi in malimi črkami.

Desetiške številke pišemo običajno; '10 na' označimo s črko E.  
Npr. 72.6E3 . R pozna tudi šestnajstiške številke, ki začenjajo  
z 0x.

Glejte `help(Quotes)` in `help(NumericConstants)`.



# Prireditveni stavek

Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavki

Osnovne zvrsti

Branje in izpis

S spremenljivkami je tesno povezan prireditveni stavek. Ta ima obliko

$$\text{ime\_spremenljivke} \leftarrow \text{aritmetični\_izraz}$$

Pojem aritmetičnega izraza bomo podrobno razdelali v nadaljevanju. Zaenkrat naj pomeni v izbrani programski jezik prepisan izraz, kakršnega srečamo na levi ali desni strani enačaja v matematičnih obrazcih.

V matematiki izraz oblike

$$\text{ime} = \text{izraz}$$

pomeni: vrednost količine ime je enaka vrednosti izraza.

Prireditveni stavek

$$\text{ime} \leftarrow \text{izraz}$$

pa pomeni: v prostor, ki pripada spremenljivki ime shrani vrednost izraza. Po izvedbi prireditvenega stavka stara vrednost spremenljivke izgine.



# Prireditveni stavek

## Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

$i = i + 1$  v matematiki nima smisla;  $i \leftarrow i + 1$  pa pomeni: vrednost spremenljivke  $i$  povečaj za 1.

Zamenjajmo vrednosti spremenljivk  $a$  in  $b$ . Standardna pot je z uvedbo dodatne spremenljivke  $c$ :

$c \leftarrow a$

$a \leftarrow b$

$b \leftarrow c$

V R-ju uporabljamo za znak prirejanja  $\leftarrow$ ; oziroma  $\rightarrow$ , če je ime spremenljivke na desni strani.

Prirejanje lahko uporabljamo tudi v podizrazih.

S funkcijo `objects()` ali `ls()` dobimo seznam trenutnih objektov (spremenljivk in funkcij).

S funkcijo `rm(seznam spremenljivk)` zahtevamo, da se spremenljivke iz seznama odstranijo iz pomnilnika.





# Prireditveni stavek

Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni stavek

Osnovne zvrsti

Branje in izpis

```
> a <- 14
> a
[1] 14
> (7 -> a)
[1] 7
> a
[1] 7
> (a <- a+1)
[1] 8
> b <- c <- a
> b
[1] 8
> c
[1] 8
```

```
> d
Error: object "d" not found
> rm(b)
> b
Error: object "b" not found
> (b <- a * (d <- 3))
[1] 24
> b
[1] 24
> d
[1] 3
> B
Error: object "B" not found
> (B <-2009)
[1] 2009
```



# Osnovne zvrsti podatkov

Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

Podatkovne zvrsti delimo na *osnovne* in *sestavljene*. V R-ju so osnovne zvrsti: `numeric`, `integer`, `complex`, `logical`, `character`, `raw`. Obstaja tudi več načinov, kako iz osnovnih zvrsti sestavimo sestavljene: `vector`, `matrix`, `array`, `list`, ...

Za preverjanje in pretvarjanje med zvrstmi so na voljo funkcije:  
`is.zvrst(x)` – preverja ali je `x` izbrane zvrsti  
`as.zvrst(x)` – ustvari podatek izbrane zvrsti ali poskusi pretvoriti `x` v izbrano zvrst  
`typeof(x)` – vrne zvrst `x`-a; skoraj isto vlogo imata tudi `class(x)` in `mode(x)`



# Števila

Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

`help(numeric)`, `help(double)`, `help(integer)`, `help(single)`,  
`help(raw)`

Zvrsti `numeric` ustrezajo realna števila dvojne natančnosti. Ostale zvrsti postanejo pomembne, ko povezujemo R s podprogrami napisanimi v drugih programskih jezikih (C, Fortran). R upošteva za realna števila **IEEE 754 Arithmetic Standard (Goldberg)**.

V 64-bitni predstavitvi so števila predstavljena na okrog 15 mest natančno. Največje  $1.797693 \cdot 10^{308}$ , najmanjše  $2.225074 \cdot 10^{-308}$ .

Cela števila so podmnožica realnih. Operacije ( / le deloma) s celimi števili ohranjajo celost, če rezultat ne preseže obseg na računalniku predstavljenih celih števil – med  $-2^{53}$  in  $2^{53}$  oziroma nekje med  $-10^{17}$  in  $10^{17}$ .

Zvrst `raw` so 8-bitni zlogi z vrednostmi (0 .. 255) zapisanimi z dvema šestnajstiškima števkom.

Še nekaj matematičnih funkcij najdemo v `help(Hyperbolic)`, `help(Special)`, `help(Bessel)`.



# Računanje z Inf, NaN, NA

## Uvod

### V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

```
> 1^2009
```

```
[1] 1
```

```
> 2009^0
```

```
[1] 1
```

```
> 0^0
```

```
[1] 1
```

```
> 1/0
```

```
[1] Inf
```

```
> -3/0
```

```
[1] -Inf
```

```
> 5/Inf
```

```
[1] 0
```

```
> Inf+Inf
```

```
[1] Inf
```

```
> 2009*Inf
```

```
[1] Inf
```

```
> Inf*Inf
```

```
[1] Inf
```

```
> sqrt(Inf)
```

```
[1] Inf
```

```
> Inf-Inf
```

```
[1] NaN
```

```
> 0/0
```

```
[1] NaN
```

```
> 0*Inf
```

```
[1] NaN
```

```
> Inf/Inf
```

```
[1] NaN
```

```
> 3%%0
```

```
[1] NaN
```

```
> sqrt(-2)
```

```
[1] NaN
```

```
Warning message:
```

```
In sqrt(-2) : NaNs produced
```

```
> 3+sqrt(NA)
```

```
[1] NA
```

```
> 1e308
```

```
[1] 1e+308
```

```
> 1e309
```

```
[1] Inf
```

```
> 1e-323
```

```
[1] 9.881313e-324
```

```
> 1e-324
```

```
[1] 0
```



# Kompleksna števila

Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

```
> (z <- 3-4i)
[1] 3-4i
> Re(z)
[1] 3
> Im(z)
[1] -4
> Mod(z)
[1] 5
> Arg(z)
[1] -0.9272952
> Conj(z)
[1] 3+4i
> z^2
[1] -7-24i
> w <- -2+3i
> z+w
[1] 1-1i
> sqrt(z)
[1] 2-1i
> sqrt(-2)
[1] NaN
Warning message:
In sqrt(-2) : NaNs produced
> sqrt(as.complex(-2))
[1] 0+1.414214i
```

$$z = x + iy$$
$$r = \sqrt{x^2 + y^2}$$
$$\varphi = \arctg(y/x)$$
$$z = r(\cos \varphi + i \sin \varphi)$$
$$\operatorname{Re}(z) \equiv x, \operatorname{Im}(z) \equiv y,$$
$$\operatorname{Mod}(z) \equiv r, \operatorname{Arg}(z) \equiv \varphi$$
$$\text{help(complex)}$$



# Logične vrednosti

Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni stavek

Osnovne zvrsti

Branje in izpis

*Logični vrednosti* sta TRUE (resnično) in FALSE (lažno), ali okrajšano T in F, ter NA. Število 0 ima vlogo FALSE, različna od 0 pa vlogo TRUE.

! x	x & y	x   y
xor(x, y)	x && y	x    y
isTRUE(x)		

	NA	F	T	&	NA	F	T	xor	NA	F	T
NA	NA	NA	T	NA	NA	F	NA	NA	NA	NA	NA
F	NA	F	T	F	F	F	F	F	NA	F	T
T	T	T	T	T	NA	F	T	T	NA	T	F

Logične vrednosti dobimo s primerjavami

x < y	x <= y	x == y
x > y	x >= y	x != y

in raznimi predikati, kot so `is.zvrst`, `is.finite`, `is.infinite`, `is.na`, `is.nan`, `is.null`, ...

`ifelse(pogoj, izrazT, izrazF)`

`switch(izraz, izraz1, izraz2, izraz3, ..., izrazk)`

`help(logical)`, `help(Logic)`, `help(Comparison)`



# Nizi znakov

## Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

*Niz znakov* zapišemo tako, da ga oklenemo z enojnima ' ali " navednicama. `help(character)`, `help(Quotes)`, `help(Encoding)`

Znak `\` je *ubežni znak* in nam omogoča v nizih zapisati znake, ki jih sicer ne moremo zapisati:

`\n` newline, `\r` carriage return, `\t` tab, `\b` backspace, `\a` alert (bell), `\f` form feed, `\v` vertical tab, `\\` backslash, `\nnn` osmiška koda, `\xnn` šestnajstiška koda, `\unnnn` Unicode (do 4 šestnajstiške številke), `\Unnnnnnnn` Unicode (do 8 šestnajstiških števk)

`paste`, `substr`, `substring`, `strsplit`, `strtrim`, `tolower`, `toupper`, `sub`, `cat`, `print`  
`chartr`, `grep`, `parse`, `deparse`, `format`, `invisible`



# Kodi ASCII in Unicode

Uvod

V. Batagelj

Projekt R

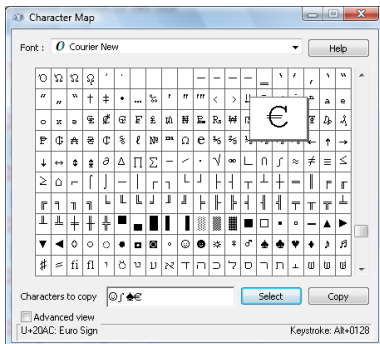
Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis



Za zapis znakov se pretežno uporablja 8-bitna koda **ASCII** in njene izpeljanke **ISO 8859**. Na spletu je standardna koda **Unicode** (17 16-bitnih ravni), ki pozna skoraj vse pisave sveta (tudi kitajske pismenke). Za varčen zapis besedil, ki vsebujejo pretežno znake latinske (angleške) abecede, se uporablja zapis **UTF-8**.

```
help(rawConversion),  
help(utf8Conversion)
```

```
> (t <- "\u0409\u0443\u0431\u043b\u044a\u043d\u0430")  
[1] "Љубљана"  
> (s <- "\u00a9\u03b1\u20ac\u2642")  
[1] "©ª§€¤¢"  
> (w <- "žarišče")  
[1] "žarišče"  
> (čas <- 27)  
[1] 27
```





# Računanje z logičnimi vrednostmi in nizi

## Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

```
> 5 & 0
[1] FALSE
> x <- -3
> ifelse(x>=0,"pozitiven","negativen")
[1] "negativen"
> sqrt(ifelse(x >= 0, x, NA))
[1] NA
> ifelse(x >= 0, sqrt(x), NA)
[1] NA
> k <- 4
> s <- 3
> switch(k,k+s,s*k-5,3*k-s,2^k,2*k+s)
[1] 16
> k <- 5
> switch(k,k+s,s*k-5,3*k-s,2^k,2*k+s)
[1] 13
> a <- "Novo"
> b <- "Mesto"
> paste(a,b)
[1] "Novo Mesto"
> paste(a,b,sep="-")
[1] "Novo-Mesto"

> s <- "lJUBLjAna"
> paste(toupper(substr(s,1,1)),tolower(substr(s,2,nchar(s))),sep='')
[1] "Ljubljana"

> c <- "Ljubljana"
> nchar(c)
[1] 9
> substr(c,6,9)
[1] "jana"
> strsplit(c,"j")
[[1]]
[1] "L" "ubl" "ana"
> sub("a","*",c)
[1] "Ljublj*na"
> gsub("a","*",c)
[1] "Ljublj*n*"
> tolower(c)
[1] "ljubljana"
> toupper(c)
[1] "LJUBLJANA"
> (ar <- charToRaw("a"))
[1] 61
> as.integer(ar)
[1] 97
> rawToChar(as.raw(65))
[1] "A"
```



# Prednostni vrstni red

## Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni stavek

Osnovne zvrsti

Branje in izpis

[	[[	indeksi		
::	:::	doseganje spremenljivk v imenskih prostorih		
\$	@	del podatka		
^		potenciranje (z desne)		
-	+	predznak		
:		zaporedje		
%any%		posebne operacije		
*	/	množenje, deljenje		
+	-	seštevanje, odštevanje		
<	>	primerjave		
<=	>=	==	!=	
!		zanikanje		
&	&&	in		
		ali		
~		model		
->	->>	prirejanje na desno		
=		prirejanje (z desne)		
<-	<<-	prirejanje (z desne)		
?		pomoč		

V izrazih se za enakovredne operacije uporablja pravilo z leve proti desni, razen za tiste, pri katerih je zapisano (z desne).



# Branje in izpis

## Uvod

V. Batagelj

Projekt R

Okolje R

Spremenljivke

Prireditveni  
stavek

Osnovne zvrsti

Branje in izpis

Branje in izpis podatkov bomo podrobneje obdelali pri datotekah. Zaenkrat le najosnovnejše. Branje s tipkovnice nam omogoča funkcija `scan`, izpis pa funkciji `cat` in `print`. Funkcija `print` izpiše en sam podatek in je v bistvu enakovredna temu, da ta podatek vtipkamo v interaktivnem načinu dela. Funkcija `cat` omogoča združen izpis večih podatkov, ki jih predela v en niz znakov in izpiše.

```
> a <- scan(what=character(0),n=1)          > z <- 3-4i
1: "abra"                                   > z
Read 1 item                                 [1] 3-4i
> a                                          > print(z)
[1] "abra"                                  [1] 3-4i
> a <- 3; b <- 4
> cat(a, "+", b, "=", a+b, "\n", sep="")
3+4=7
```