

# Regularni izrazi

## Delo z nizi

- ▶ Skoraj vedno delamo s tekstovnim zapisom podatkov (datoteke, splet)
- ▶ Pogosta operacija: prepoznavanje zapisa v določenem formatu
- ▶ Katero število predstavlja niz "10202"?
- ▶ A je v nizu "102a" sploh število? Kaj pa v nizu "102"?
- ▶ Radi bi uporabili samo 2. indeks v nizu "matrika[2,34]". No včasih niz izgleda tako: "matrika[2, 34 ]"
- ▶ V stolpcu imamo podatke v nizu v obliki "2,32.1,0.4" in bi radi to razbili na 3 številske stolpce

# Vzorci

- ▶ Vizualno bi znali opisati vzorce
- ▶ Težje bi bilo napisati kodo, ki pregleduje niz znak po znak in se odloča o ujemanju, podnizih, delitvah, . . .
- ▶ Podobno, ponavljajoče se delo; veliko možnosti za napake
- ▶ Regularni izrazi
  - ▶ jezik za opis vzorcev
  - ▶ funkcije za opis manipulacij nad nizi, ki se ujemajo z vzorci

## Regularni izrazi

- ▶ Regularni vzorci so opisani v nizih
- ▶ Konkretna črka/številka/znak predstavlja vzorec za samo sebe

```
> niz <- "avto"  
> vzorec <- "avto"  
> grep(vzorec, niz)  
[1] 1
```

- ▶ grep - global regex print. Ali se v nizu nahaja vzorec

```
> nizi <- c("avto", "avtomobil", "avokado",  
           "avtokado", "navto")  
> vzorec <- "avto"  
> grep(vzorec, nizi)  
[1] 1 2 4 5
```

```
> grepl(vzorec, nizi)  
[1] TRUE TRUE FALSE TRUE TRUE
```

## Metaznaki in posebni znaki

- ▶ Določeni znaki so izjeme: . \ | ( ) [ { ^ \$ \* + ?
- ▶ \- ubežni znak
- ▶ . - kateri koli (eden) znak
- ▶ ^ - začetek niza
- ▶ \$ - konec niza

```
> grep("^avto", c("avto", "avtomobil", "avokado",  
  "avtokado", "navto"))  
[1] 1 2 4
```

```
> grep("^av..$", c("avto", "avtomobil", "avokado",  
  "avtokado", "navto"))  
[1] 1
```

# Posebni znaki

## ► Alfaničrni znak

```
> grep("\\w", c(" ", "a", "1", "A", "%", "\t"))  
[1] 2 3 4
```

## ► Ne alfanumerični znak

```
> grep("\\W", c(" ", "a", "1", "A", "%", "\t"))  
[1] 1 5 6
```

## ► Beli znak

```
> grep("\\s", c(" ", "a", "1", "A", "%", "\t"))  
[1] 1 6
```

## ► Nebeli znak

```
> grep("\\S", c(" ", "a", "1", "A", "%", "\t"))  
[1] 2 3 4 5
```

# Posebni znaki

## ► Številka

```
> grep("\\d", c(" ", "a", "1", "A", "%", "\t"))  
[1] 3
```

## ► Neštevilka

```
> grep("\\D", c(" ", "a", "1", "A", "%", "\t"))  
[1] 1 2 4 5 6
```

## Variante vzorcev

- ▶ Možnosti za en znak naštejemo v oglatem oklepju

```
> grep("^[abc]\\w\\w", c("avto", "bus", "ne", "vozi"))  
[1] 1 2
```

- ▶ Vse tričrkovne besede iz malih črk

```
> grep("[a-z][a-z][a-z]$", c("Čas", "teče", "nič",  
    "ne", "reče:", "tik", "tak"))  
[1] 6 7
```

- ▶ Vse tričrkovne besede, ki se začnejo z malo ali veliko črko in so sicer iz malih črk, tudi šumnikov

```
> grep("[a-zA-ZčšžČŠŽ][a-zčšž][a-zčšž]$", c("Čas",  
    "teče", "nič", "ne", "reče:", "tik", "tak"))  
[1] 1 3 6 7
```



## Variante vzorcev

- ▶ Ena ali dvomestne številke kjerkoli

```
> grep("((\\d)|([1-9]\\d))", c("1", "20", "0", "nič",  
    "to je 100%", "09"))  
[1] 1 2 3 5 6
```

- ▶ Okrogli oklepaji predstavljajo skupino. Morajo biti pravilno vgnezdjeni
- ▶ Natanko eno ali dvo mestne številke

```
> grep("^((\\d)|([1-9]\\d))$", c("1", "20", "0",  
    "nič", "to je 100%", "09"))  
[1] 1 2 3
```

- ▶ Pozor: okrogli oklepaji okrog možnosti navedenimi z operatorjem | so obvezni.

## Ponavljanje vzorcev

- ▶ Operatorji za ponavljanje delujejo na prejšnji znak ali skupino
- ▶ ? - kvečjemu ena ponovitev
- ▶ \* - nič ali več ponovitev
- ▶ + - ena ali več ponovitev
- ▶ {m} – natanko m ponovitev
- ▶ {m, n} – m do n ponovitev
- ▶ {m, } – vsaj m ponovitev
- ▶ Besede sestavljene iz malih črk

```
> grep("^[a-z]*$", c("slika", "je", "vredna",  
    "1000", "besed."))
```

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

- ▶ Brez prazne besede

```
> grep("^[a-z]+$", c("", "slika", "je", "vredna",  
    "1000", "besed."))
```

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

# Ponavljjanja

- ▶ Zaporedje besed iz malih črk ločenih s presledki

```
> grep("^([a-z]+ )*[a-z]+$", c("besede",  
    "ali pa stavki", "123 ni"))  
[1] 1 2
```

- ▶ Besede dolžine 3-5 črk

```
> grep("[a-z]{3,5}$", c("ta", "beseda", "nima",  
    "pomena"))  
[1] 3
```

- ▶ Predznačena cela števila

```
> grep("[+-]?(0|[1-9][0-9]*)$",  
    c("0", "+1", "01", "-99"))  
[1] 1 2 4
```

# Podvzorci

- ▶ Podvzorke označimo s skupinami (okrogli oklepaji)
- ▶ Skupine identificiramo po številki

(... (... (...)) ... (... (...)) ... (...))  
1        2    3                4    5                6

- ▶ Cela števila iz vektorja

(1, 2)

( -2, 7)

(       -3       ,       45)

## Podvzorci

```
> nizi <- c("(1,2)", "(-2, 7)", "(-3, 45)",  
            "(a, 3)")  
> vzorec <- paste0("\\(\\s*([+-]?(0|[1-9][0-9]*))\\s*,",  
                    "\\s*([+-]?(0|[1-9][0-9]*))\\s*\\)")  
> lidx <- !grepl(vzorec, nizi)  
> komp1 <- sub(vzorec, "\\1", nizi)  
> komp1[lidx] <- NA  
> komp2 <- sub(vzorec, "\\3", nizi)  
> komp2[lidx] <- NA  
> as.integer(komp1)  
[1] 1 -2 -3 NA  
> as.integer(komp2)  
[1] 2 7 45 NA
```

## Podvzorci s *strapplyc*

```
nizi <- c("(1,2)", "( -2, 7)", "(    -3    ,    45)",  
          "(a, 3)")  
vzorec <- paste0("\\(\\s*([+-]?(0|[1-9][0-9]*))\\s*,",  
                  "\\s*([+-]?(0|[1-9][0-9]*))\\s*\\)")  
require(gsubfn, quietly = TRUE)  
skupine <- strapplyc(nizi, vzorec)
```

## Podvzorci s *strapplyc*

```
print(skupine)
```

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

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

```
##
```

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

```
## [1] "-2" "2"  "7"  "7"
```

```
##
```

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

```
## [1] "-3" "3"  "45" "45"
```

```
##
```

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

```
## character(0)
```

## Pridobivanje podvzorcev

- ▶ Pomožna funkcija za ekstrahiranje i-tega elementa

```
indeks <- function(x, i) {  
  if(length(x) >= i) x[[i]] else ""  
}
```

- ▶ Izpeljana funkcija s prednastavljenimi parametri

```
> f <- . %>% indeks(1)  
> f(c("a", "b", "c"))  
[1] "a"
```

- ▶ Pozor: operator %>% (ang. pipe) “živi” v paketu magrittr (ki ga posredno uvozimo preko paketa dplyr)



## Pridobivanje podvzorcev

- Funkcija `sapply` uporabi podano funkcijo na elementih zaporedja (seznama) in rezultat vrne v vektorju

```
> x <- sapply(skupine, . %>% indeks(1)) %>% as.integer  
> y <- sapply(skupine, . %>% indeks(3)) %>% as.integer  
> data.frame(x=x, y=y)
```

	x	y
1	1	2
2	-2	7
3	-3	45
4	NA	NA

## Razcepljanje nizov

- ▶ V stolpcu je več podatkov, ločenih bodisi z vejico ali podpičjem

```
> cudenStolpec <- c("12, 3; 8", "6,1,2")
```

- ▶ Razcep po separatorju opisanem z regularnim izrazom

```
> spl <- strsplit(cudenStolpec, "[\\,\\;]")
```

- ▶ Zadnji stolpec

```
> sapply(spl, function(x) as.integer(x[[3]]))  
[1] 8 2
```

## Pridobivanje tabel iz razdelitev

- Sestavimo data.frame

```
> unlist(spl)
[1] "12" " 3" " 8" "6"  "1"  "2"
```

```
> as.integer(unlist(spl))
[1] 12  3  8  6  1  2
```

```
> mat <- matrix(as.integer(unlist(spl)), ncol=3,
                 byrow=TRUE)
```

```
> print(mat)
      [,1] [,2] [,3]
[1,]   12   3   8
[2,]    6   1   2
```

```
> df <- data.frame(mat)
```

```
> print(df)
  X1 X2 X3
1 12  3  8
2  6  1  2
```

## Pridobivanje tabel iz razdelitev

```
> colnames(df) <- c("Dolžina", "Širina", "Višina")
```

```
> print(df)
```

	Dolžina	Širina	Višina
1	12	3	8
2	6	1	2

```
> df %>% mutate(Volumen=Višina*Širina*Dolžina)
```

	Dolžina	Širina	Višina	Volumen
1	12	3	8	288
2	6	1	2	12