

Interna IT izobraževanja KIS
Uporaba statističnega programa R

Modul A: Osnove R-a, R-studia
in pisanja kode

Jana Obšteter
16. 11. 2020

Vsebina


1. Osnove R-a in R-studia
2. Osnovni podatkovni tipi in strukture
3. Osnovne operacije
4. Osnove dela z datotekami
5. Preurejanje podatkov
6. Povzemanje podatkov
7. Pomoč za R in dobra praksa pisanja kode

Struktura predavanj

- Osnove (.pptx predstavitev)
- Vaje (.Rmd, .R, .pdf)
- Koda

1. Osnove R-a in R-studia (izvršitev kode, paketi)
2. Osnovni podatkovni tipi in strukture
3. Osnovne operacije
4. Osnove dela z datotekami
5. Preurejanje podatkov
6. Povzemanje podatkov
7. Pomoč za R in dobra praksa pisanja kode

Namestitev R-a in R-studia

-  <https://cloud.r-project.org/> (Mac, Windows, Linux)
 - Linux: `sudo apt -y install r-base`, or download deb/tar.gz
 - Windows

Download and Install R

Precompiled binary distributions of the base

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

This is what you want to install R for the first time.

[Download R 4.0.3 for Windows](#)


Namestitev R-a in R-studia

-  Studio®: <https://rstudio.com/products/rstudio/download/>

RStudio Desktop











Open Source License

Free



DOWNLOAD

[Learn more](#)

OS	Download
Windows 10/8/7	 RStudio-1.3.1093.exe 
macOS 10.13+	 RStudio-1.3.1093.dmg
Ubuntu 16	 rstudio-1.3.1093-amd64.deb
Ubuntu 18/Debian 10	 rstudio-1.3.1093-amd64.deb
Fedora 19/Red Hat 7	 rstudio-1.3.1093-x86_64.rpm
Fedora 28/Red Hat 8	 rstudio-1.3.1093-x86_64.rpm
Debian 9	 rstudio-1.3.1093-amd64.deb
SLES/OpenSUSE 12	 rstudio-1.3.1093-x86_64.rpm
OpenSUSE 15	 rstudio-1.3.1093-x86_64.rpm

Izberemo datoteko glede na operacijski sistem

Tu odpiramo /
pišemo /
urejamo
ukazne datoteke

Ukazne datotek v R-u = **.R**

Urejevalnik / podatki

Okolje (spremenljivke) /
zgodovina

Konzola

Datotke / grafi / paketi / pomoč

R-studio

The image shows the RStudio interface with three panels highlighted and labeled:

- Urejevalnik / podatki** (Editor / Data Viewer): The top-left panel, highlighted with a blue border, showing a script editor with a file named "Untitled1".
- Okolje (spremenljivke) / zgodovina** (Environment / History): The top-right panel, highlighted with a blue border, showing the "Environment" tab with a variable named "howdyMessage" having the value "Hello from R console!".
- Konzola** (Console): The bottom-left panel, highlighted with a blue border, showing the R console output:

```
> howdyMessage <- "Hello from R console!"  
> print(howdyMessage)  
[1] "Hello from R console!"  
>
```
- Datotke / grafi / paketi / pomoč** (Files / Plots / Packages / Help / Viewer): The bottom-right panel, highlighted with a green border, showing the "Help" tab with the documentation for "Data Frames".

A red speech bubble points to the console output, containing the text: "Tu izvršujemo ukaze / vidimo izpis" (Here we execute commands / see the output).

Prva vaja

R-studio

Če tega okna nimamo, kliknemo
in izberemo Rscript



Vpišimo 3+5
Pritisnemo Ctrl+Enter
ali



Okolje (spremenljivke) /
zgodovina

Files Plots Packages Help Viewer

R: Data Frames Find in Topic

data.frame {base} R Documentation

Data Frames

Description

Thi
ma
mc

Datoteke / grafi / paketi / pomoč

Usage

```
data.frame(..., row.names = NULL, check.rows = FALSE,  
            check.names = TRUE,  
            stringsAsFactors = default.stringsAsFactors())  
  
default.stringsAsFactors()
```

Arguments

... these arguments are of either the form value or tag = value.

Konzola

```
> howdyMessage <- "Hello from R console!"  
> print(howdyMessage)  
[1] "Hello from R console!"  
>
```

R-studio

The screenshot shows the R Studio interface with four panels:

- Source Editor (top-left):** Labeled "Urejevalnik / podatki" (Editor / data). It shows a file named "Untitled1" with a single line of code: `1`.
- Environment (top-right):** Labeled "Okolje (spremenljivke) / zgodovina" (Environment (variables) / history). It shows the "Global Environment" with a variable `howdyMessage` having the value `"Hello from R console!"`.
- Console (bottom-left):** Labeled "Vpišimo 3+5 in Enter" (We write 3+5 and press Enter). It shows the execution of the following R code:

```
> howdyMessage <- "Hello from R console!"  
> print(howdyMessage)  
[1] "Hello from R console!"  
>
```
- Help (bottom-right):** Labeled "Datotke / grafi / paketi / pomoč" (Files / plots / packages / help). It shows the "Data Frames" section of the R documentation, including a description and usage examples.

Ukazna koda

- Pisanje direktno v konzolo in Enter ali
- Pisanje v urejevalnik in izvršitev v kosu ali vrstica po vrstico [Ctrl + Enter]
- Pisanje v urejevalnik omogoča shranjevanje kode → **.R**
- V konzoli lahko prikličemo predhodne ukaze: ↑
- Izvršitev ukazne datotek v terminalu: `Rscript datoteka.R`

Ukazna koda

- Ctrl+C: prekine ukaz v konzoli
 - Esc: zbriše vrstico v konzoli
 - Ctrl + A: izbere vso kodo v urejevalniku
 - Ctrl+1: prestavi kurzor v urejevalnik
 - Ctrl+2: prestavi kurzor v konzolo
-
- Več bližnjic: <https://tinyurl.com/oqh3o3g>

Ukazna koda

- Pripis vrednosti: `<-` ali `=`

- Osnova sintaksa:

```
imeSpremenljivke <- vrednost
```

```
a <- 3
```

- Ime spremenljivke:

- dovoljene male, velike črke, številke, . in _;
- ne sme se **začeti** s številko ali _

- Klic spremenljivke: ime spremenljivke brez navednic

```
imeSpremenljivke
```

```
> a  
[1] 3
```

Ukazna koda

- Klic funkcij:

```
imeFunkcije(parametri)
```


- R je **case-sensitive!** (razlika med malimi in VELIKIMI črkami): Beseda \neq beseda
- Vrstice, ki se začnejo z # niso izvršene = komentar

Nameščanje paketov / knjižnic

- Viri paketov: CRAN, bioConductor, git
- Paket `base` vključen (osnovne funkcije)
- Funkcija `install.packages("ImePaketa")`
- Ali: Tools > Install packages
- Enkratna namestitev

Vaja: Podatkovni tipi in strukture

- Datoteke

- ModulA.Rmd: s pritiskom na ► izvršimo kodo
- ModulA.R: spuščamo kodo z Ctrl+Enter ali 
- ModulA.pdf: pdf s kodo

- Vse datoteke z ali brez rešitev!

- ModulA_brezResitev.Rmd
- ModulA_brezResitev.R
- ModulA_brezResitev.pdf


Druga vaja

- Inštalirajmo pakete `reshape`, `tidyr`, `dplyr`
- V konzolo vpišemo:
 - `install.packages(c("reshape", "tidyr", "dplyr"),
dependencies = TRUE)`
(kopiranje ne bo delovalo zaradi napačnih navednic)

Nalaganje knjižnic

- Ob vsakem zagonu R-a
- Funkcija `library(ImePaketa)`
- Ali: Packages (spodnje desno okno) > obkljukamo paket

Samodejno dopolnjevanje

- `tab`  dopolni ime spremenljivke / funkcije
 - Vnesemo prvih par črk + `tab`: če je začetek unikaten – dopolni, drugače izpiše možnosti

1. Osnove R-a in R-studia
- 2. Osnovni podatkovni tipi in strukture**
3. Osnovne operacije
4. Osnove dela z datotekami
5. Preurejanje podatkov
6. Povzemanje podatkov
7. Pomoč za R in dobra praksa pisanja kode

Podatkovni tipi

- **Cela števila** (*integer*): 0, 1, 50, 1000000, -20, -550000 ...
- **Realna števila** (*numeric*): 1, 2.5, 900.23 ...
- **Znaki** (*character*): v navednicah, “a”, “beseda”, “Lahko tudi stavek”, “5”, “100” ...
- **Logični vector** (*logical*): TRUE / FALSE
- **Kategorične spremenljivke** (*factor*): določene vrednosti - ravni, S-J-V-Z, zjutraj-zvečer, pon-tor-sre, 2010-2011-2012 ...
- **Manjkajoča vrednost**: NA
- **Ni število** (*not a number*): NaN
- **Prazen element**: NULL

Podatkovne strukture

- **Vektor** (*vector*)
 - vrednosti: en podatkovni tip
 - funkcija `c(vrednosti)`
 - `c(1, 4, 5), c('a', 'b')` ali `1:10` (številsko zaporedje)
 - Izbira elementa z indeksom: `a[1]`

```
> a <- c(1, 4, 5)
> a
[1] 1 4 5
> a[1]
[1] 1
```


Podatkovne strukture

- **Faktor** (*factor*)
 - vrednosti: kategorične spremenljivke
 - Določeno število vrednosti (poletje-zima, pon-tor-sre, 2010-2011-2012, Slovenija-Avstrija-Hrvaška ...) = ravni
 - funkcija `factor(c(vrednosti))`
 - `factor(c('lok1', 'lok2'))`
 - Izbira elementa z indeksom: `a[1]`

```
> a <- factor(c("lok1",  
+               "lok2"))  
> a[1]  
[1] lok1  
Levels: lok1 lok2
```

Podatkovne strukture

- **Seznam** (*list*)
 - Različni podatkovni tipi
 - Funkcija **list**(elementi)
 - Izbira elementa z indeksom:

```
list(c(1,2,5),  
      c('a','b'),  
      5.23,  
      10)
```

```
a[[1]]
```

```
> a <- list(c(1,2,5), c("a","b"), 5.23, 10)  
> a  
[[1]]  
[1] 1 2 5  
  
[[2]]  
[1] "a" "b"  
  
[[3]]  
[1] 5.23  
  
[[4]]  
[1] 10  
  
> a[[1]]  
[1] 1 2 5
```

Podatkovne strukture

- **Podatkovni okvir / tabela** (*data frame*)

- dvodimenzionalna struktura
- stolpci in vrstice (lahko poimenovani), različni podatkovni tipi
- funkcija

```
data.frame(imeStolpca1 = vektor, ...)
```

- ```
data.frame(ID = c(1, 2, 3),
 visina = c(182, 183, 190))
```

# Podatkovne strukture

- **Podatkovni okvir / tabela** (*data frame*)

- Izbira elementa z indeksom vrstice in stolpca: `a[1, 1]`
- Izbira stolpca z indeksom stolpca: `a[, 1]`
- Izbira vrstice z indeksom vrstice: `a[1, ]`

```
> a <- data.frame(ID = c(1,2,3), visina = c(182, 183, 190))
> a
 ID visina
1 1 182
2 2 183
3 3 190
> a[1,1]
[1] 1
```

# Podatkovne strukture

- **Matrika** (*matrix*)
  - Dvodimenzionalna struktura
  - Matrično računanje: inverza (`solve()`), množenje (`%*%`) ...
  - Funkcija

```
matrix(podatki,
 nrow = #vrstic,
 ncol = #stolpcev,
 byrow = TRUE/FALSE)
```

# Podatkovne strukture

- **Matrika** (*matrix*)

- `matrix(1:10, nrow=5)`
- Izbira elementa z indeksom vrstice in stolpca:

`a[1, 1]`

- Izbira stolpca z indeksom stolpca:

`a[ , 1]`

- Izbira vrstice z indeksom vrstice:

`a[1, ]`

```
> a <- matrix(1:10, nrow=5)
> a
 [,1] [,2]
[1,] 1 6
[2,] 2 7
[3,] 3 8
[4,] 4 9
[5,] 5 10
> a[1,1]
[1] 1
```

# Podatkovne strukture

- **Polje** (*array*)

- Kot matrika, vendar poljubno število dimenzij
- funkcija

```
array(data = podatki,
 dim = vektorDimenzij,
 dimnames = imenaDimenzij)
```

# Podatkovne strukture

- **Polje** (*array*)

- `a <- array(c(1,2,3),  
dim=c(3,3,2))`
- Izbira elementa z indeksom vseh dimenzij: `a[1, 1, 1]`

```
> a <- array(c(1,2,3), dim=c(3,3,2))
> a
, , 1

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

, , 2

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

> a[1,1,1]
[1] 1
```



# Podatkovne strukture

- Preveri strukturo: `class (objekt)`
- Preveri dolžino: `length (objekt)`
- Pretvarjanje med strukturami: `as.struktura()`
  - `as.vector()`
  - `as.list()`
  - `as.data.frame()`
  - `as.matrix()`

# Vaja

- ModulA.Rmd / ModulA.R:  
Osnovni podatkovni tipi in strukture

1. Osnove R-a in R-studia (izvršitev kode, paketi)
2. Osnovni podatkovni tipi in strukture
- 3. Osnovne operacije**
4. Osnove dela z datotekami
5. Preurejanje podatkov
6. Povzemanje podatkov
7. Pomoč za R in dobra praksa pisanja kode

# Osnovni operatorji

- Aritmetični operatorji:
  - seštevanje/odštevanje/množenje/deljenje:  $+$   $-$   $*$   $/$
  - Potenca:  $^$
  - Ostanek:  $\% \%$ , celoštevilski kvocient  $\% / \%$
  - Matrično množenje:  $\% * \%$

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

odštevanje

```
> 3^3
[1] 27
```

potenca

```
> 10%%3
[1] 1
```

ostanek

```
> 10%/3
[1] 3
```

celoštevski  
kvocient

# Osnovni operatorji

- Primerjalni operatorji (vrnejo logično vrednost):
  - Večje / manjše: `>`, `>=`, `<`, `<=`
  - Preverjanje enakosti / neenakosti: `==` / `!=`
  - Preverjanje vsebnosti: `%in%`

```
> 3 > 5
[1] FALSE
```

Ali je 3  
večje od 5

```
> 3 == 3
[1] TRUE
```

Ali je 3  
enako 3

```
> 3 != 3
[1] FALSE
```

Ali 3 ni  
enako 3

```
> 3 %in% c(1,2,3)
[1] TRUE
```

Ali je 3 med  
vrednostmi c(1,2,3)

# Osnovni operatorji

- Logični operatorji (vrnejo logično vrednost):
  - Negacija: `!`
  - In: `&` (element po element), `&&` (za celoten objekt)
  - Ali: `|` (element po element), `||` (za celoten objekt)

```
> !(3 > 4)
[1] TRUE
```

Ali ni res, da je 3  
večje od 4

```
> (3 == 3) & (3 > 4)
[1] FALSE
```

Ali je 3 enako 3 IN  
3 večje od 4

```
> (3 == 3) | (3 > 4)
[1] TRUE
```

Ali je 3 enako 3 ALI  
3 večje od 4

# Osnovni operatorji

- Združevanje / lepljenje znakov
  - `paste(znak1, znak2, sep="ločilo")`
  - `paste0(znak1, znak2)`, ločilo = ""
  - `znak` je lahko tudi vektor → zlepi se prvi element vektorja 1 s prvim elementom vektorja 2 itd.

```
> paste("Kmetijski", "Institut", sep="_")
[1] "Kmetijski_Institut"
```

```
> paste0(c(1,2,3), c("a", "b", "c"))
[1] "1a" "2b" "3c"
```

# Osnovne vgrajene funkcije

- Vsota: `sum(vektor)`
- Povprečje: `mean(vektor)`
- Varianca: `var(vektor)`
- Dolžina: `length(element)`
- Minimum: `min(vektor)`
- Maximum: `max(vektor)`
- Izpiši: `print(element)`

**Pomoč:** `?ImeFunkcije`

```
> vrednosti <- c(1,2,3,5)
> sum(vrednosti)
[1] 11
> mean(vrednosti)
[1] 2.75
> max(vrednosti)
[1] 5
```



# Vaja

- ModulA.Rmd / ModulA.R:  
Osnovne operacije

1. Osnove R-a in R-studia (izvršitev kode, paketi)
2. Osnovni podatkovni tipi in strukture
3. Osnovne operacije
- 4. Osnove dela z datotekami**
5. Preurejanje podatkov
6. Povzemanje podatkov
7. Pomoč za R in dobra praksa pisanja kode

# Delovni imenik

- *Working directory*
- Privzeta lokacija za branje / pisanje datotek
- **setwd(potDoImenika)** ali  
Session > Set Working Directory > Choose Directory

# Branje datotek

- Različne funkcije za različne vrste datotek

- Osnovna sintaksa:

```
read.tipDatoteke(datoteka, parametri)
```

# Branje datotek

- **csv** (comma separated values)
  - `read.csv(potDoDatoteke/ImeDatoteke)`
  - Parametri (neobvezni) `?read.csv()`
    - `header`: ali datoteka vključuje imena stolpcev, TRUE / FALSE (**default = TRUE**)
    - `sep`: kaj razmejuje stolpce, “ločilo” (“,”, “.”, “ “, “\t”) (**default = “,”**)
    - `dec`: kateri znak označuje decimalko (**default = “.”**)
    - `row.names`: vektor imen vrstic (default: številka vrstice)
    - `col.names`: vektor imen stolpcev (default: “V” + št. Stolpca)
    - `skip`: število vrstic, ki jih preskoči pri branju (default = 0)

# Branje datotek

- **Evropski csv** (comma separated values)
  - `read.csv(potDoDatoteke/ImeDatoteke)`
  - Parametri (neobvezni) `?read.csv()`
    - `header`: ali datoteka vključuje imena stolpcev, TRUE / FALSE (**default = TRUE**)
    - `sep`: kaj razmejuje stolpce, “ločilo” (“,”, “.”, “ “, “\t”) (**default = “;”**)
    - `dec`: kateri znak označuje decimalko (**default = “,”**)
    - `row.names`: vektor imen vrstic (default: številka vrstice)
    - `col.names`: vektor imen stolpcev (default: “V” + št. Stolpca)
    - `skip`: število vrstic, ki jih preskoči pri branju (default = 0)

# Branje datotek

- **txt**

- `read.delim(potDoDatoteke/ImeDatoteke)`
- parametri (neobvezni) `?read.delim()`
  - header: ali datoteka vključuje imena stolpcev, TRUE / FALSE (**default = TRUE**)
  - sep: kaj razmejuje stolpce, "ločilo" (" ", ".", ",", "\t") (**default = "\t"**)
  - dec: kateri znak označuje decimalko (default = ".")
  - row.names: vektor imen vrstic (default: številka vrstice)
  - col.names: vektor imen stolpcev (default: "V" + št. Stolpca)
  - skip: število vrstic, ki jih preskoči pri branju (default = 0)

# Branje datotek

- **tabela**

- `read.table(potDoDatoteke/ImeDatoteke)`
- parametri (neobvezni) `?read.table()`
  - header: ali datoteka vključuje imena stolpcev, TRUE / FALSE (**default = FALSE**)
  - sep: kaj razmejuje stolpce, “ločilo” (“,”, “.”, “ ”, “\t”) (**default = “ ”**)
  - dec: kateri znak označuje decimalko (default = “.”)
  - row.names: vektor imen vrstic (default: številka vrstice)
  - col.names: vektor imen stolpcev (default: “V” + št. Stolpca)
  - skip: število vrstic, ki jih preskoči pri branju (default = 0)



# Branje datotek

- **excel dokument**

- knjižnica `library("readxl")`
- `read_xls / read_xlsx(potDoDatoteke/ImeDatoteke)`
- parametri (neobvezni) `?read_xls()`
  - `sheet`: številka/ime lista
  - `range`: obseg celic za branje ("B3:D87", "Budget!B2:G14")
  - `col_names`: ali dokument vsebuje imena stolpcev
  - `col_type`: vektor podatkovnih tipov v stolpcih ali NULL
  - `skip`: preskoči n število vrstic

# Branje datotek

- **iz odložišča (*clipboard*)**
  - `read.table("clipboard")`

# Branje datotek

|               | header | sep  | dec | comment.char |
|---------------|--------|------|-----|--------------|
| read.table()  | F      | ""   | "." | "#"          |
| read.csv()    | T      | ","  | "." | ""           |
| read.csv2()   | T      | ","  | "," | ""           |
| read.delim()  | T      | "\t" | "." | ""           |
| read.delim2() | T      | "\t" | "," | ""           |

# Branje večjih datotek

- readr: write\_delim / read\_table2
- data.table: fwrite / fread
- feather: write\_feather / read\_feather (binarno)
- vroom: vroom / vroom\_write

# Lastnosti prebranih podatkov

- Izpiši prvih par vrstic: `head(tabela, n=x)`, privzet  $n = 6$
- Izpiši zadnjih par vrstic: `tail(tabela, n=x)`, privzet  $n = 6$
- Struktura: `str(tabela)`
- Število vrstic: `nrow(tabela)`
- Število stolpcev: `ncol(tabela)`
- Povzemi vse spremenljivke: `summary(tabela)`

# Osnovne operacije na tabelah

- Izberi stolpec: `tabela$imeStolpca`, `tabela[, štStolpca]`

- Izberi vrstico: `tabela[štVrstice, ]`

- Izberi vrstice, ki zadostujejo pogoju v stolpcu (in vse stolpce):

```
tabela[tabela$imeStolpca == vrednost, :]
```

pogoj

vsi stolpci

```
tabela[tabela$imeStolpca > vrednost, :]
```

- Izberi vrstice, ki zadostujejo večim pogojem v stolpcu (in vse stolpce):

```
tabela[(tabela$imeStolpca1 > vrednost) & IN
(tabela$imeStolpca2 < vrednost),]
```

# Osnovne operacije na tabelah

- Alternativa za izbiranje stolpcev: funkcija `subset()`

```
subset(x = tabela,
 subset = pogoj,
 select = stolpci, ki obdržimo/odstranimo)
```

# Osnovne operacije na tabelah

- Spremeni ime stolpca/-ev:
  - `colnames(tabela) <- c(ImenaStolpcev)`
  - `colnames(tabela)[štStolpca] <- c(ImeStolpca)`



# Osnovne operacije na tabelah

- Povzemi stolpec:
  - Numeričen: `summary(objekt$imeStolpca)`
  - Kategoričen: `table(objekt$imeStolpca)`
- Ustvari nov stolpec:
  - `objekt$novStolpec <- vsebinaStolpca`  
→ dolžina nove vsebine = dolžina obstoječe vsebine
  - `objekt$novStolpec <- NA` (inicializiraj prazen stolpec)

# Primer branja tabele – ocene testa

```
Preberemo datoteko
```

```
> test <- read.csv("Test.csv")
```

```
Preverimo prve tri vrstice
```

```
> head(test)
```

|   | Ime   | Priimek | Starost | Tocke |
|---|-------|---------|---------|-------|
| 1 | Maja  | Novak   | 25      | 9.25  |
| 2 | Jure  | Kovač   | 46      | 9.02  |
| 3 | Luka  | Medved  | 33      | 2.21  |
| 4 | Eva   | Cankar  | 32      | 4.32  |
| 5 | Tina  | Zajc    | 22      | 3.28  |
| 6 | Marko | Petek   | 50      | 6.54  |

# Primer branja tabele

# Preverimo število vrstic / stolpcev

```
> nrow(test)
```

```
[1] 6
```

```
> ncol(test)
```

```
[1] 4 strukturo
```

```
> str(test)
```

```
'data.frame': 6 obs. of 4 variables:
```

```
$ Ime : chr "Maja" "Jure" "Luka" "Eva" ...
```

```
$ Priimek: chr "Novak" "Kovač" "Medved" "Cankar" ...
```

```
$ Starost: int 25 46 33 32 22 50
```

```
$ Tocke : num 9.25 9.02 2.21 4.32 3.28 6.54
```

# Primer branja tabele

# Izberemo drugo in tretjo vrstico

# Izberemo tretji stolpec

# Primer branja tabele

```
Ustvarimo nov stolpec
```

```
> test$Predmet <- "Matematika"
> head(test)
```

|   | Ime   | Priimek | Starost | Tocke | Predmet    |
|---|-------|---------|---------|-------|------------|
| 1 | Maja  | Novak   | 25      | 9.25  | Matematika |
| 2 | Jure  | Kovač   | 46      | 9.02  | Matematika |
| 3 | Luka  | Medved  | 33      | 2.21  | Matematika |
| 4 | Eva   | Cankar  | 32      | 4.32  | Matematika |
| 5 | Tina  | Zajc    | 22      | 3.28  | Matematika |
| 6 | Marko | Petek   | 50      | 6.54  | Matematika |

# Primer branja tabele

```
Zlepimo dva stolpca v nov stolpec
```

```
> test$ImePriimek <- paste(test$Ime, test$Priimek, sep="_")
> head(test)
```

|   | Ime   | Priimek | Starost | Tocke | Predmet    | ImePriimek  |
|---|-------|---------|---------|-------|------------|-------------|
| 1 | Maja  | Novak   | 25      | 9.25  | Matematika | Maja_Novak  |
| 2 | Jure  | Kovač   | 46      | 9.02  | Matematika | Jure_Kovač  |
| 3 | Luka  | Medved  | 33      | 2.21  | Matematika | Luka_Medved |
| 4 | Eva   | Cankar  | 32      | 4.32  | Matematika | Eva_Cankar  |
| 5 | Tina  | Zajc    | 22      | 3.28  | Matematika | Tina_Zajc   |
| 6 | Marko | Petek   | 50      | 6.54  | Matematika | Marko_Petek |

# Primer branja tabele

# Izberemo kandidate, ki so dosegli nad 5 točk

```
> test[test$Tocke > 5,]
```

|   | Ime   | Priimek | Starost | Tocke | Predmet    | ImePriimek  |
|---|-------|---------|---------|-------|------------|-------------|
| 1 | Maja  | Novak   | 25      | 9.25  | Matematika | Maja_Novak  |
| 2 | Jure  | Kovač   | 46      | 9.02  | Matematika | Jure_Kovač  |
| 6 | Marko | Petek   | 50      | 6.54  | Matematika | Marko_Petek |

"-----", ki so dosegli nad 5 točk in so mlajši od 30

```
> test[(test$Tocke > 5) & (test$Starost < 30),]
```

|   | Ime  | Priimek | Starost | Tocke | Predmet    | ImePriimek |
|---|------|---------|---------|-------|------------|------------|
| 1 | Maja | Novak   | 25      | 9.25  | Matematika | Maja_Novak |

# Primer branja tabele

# Izračunamo povprečno starost

# Povzamemo stolpec Tocke



# Pisanje datotek

- **tabela**

- `write.table(objekt, potDoDatoteke/ImeDatoteke)`
- parametri (neobvezni) `?write.table()`
  - `quote`: zapiši znake z navednicami, TRUE / FALSE (**default=TRUE**)
  - `sep`: kaj razmejuje stolpce, “ločilo” (“,”, “.”, “ “, “\t”) (**default=“ ”**)
  - `append`: ali dodati objekt v datoteko, TRUE / FALSE (**default=FALSE**)
  - `row.names`: zapiši imena vrstic, TRUE/FALSE (**default=TRUE**)
  - `col.names`: zapiši imena stolpcev, TRUE/FALSE (**default=TRUE**)

# Pisanje datotek

- **csv**

- `write.csv(objekt, potDoDatoteke/ImeDatoteke)`
- parametri (neobvezni) `?write.table()`
  - `quote`: zapiši znake z navednicami, TRUE / FALSE (**default=TRUE**)
  - `sep`: kaj razmejuje stolpce, “ločilo” (“,”, “.”, “ “, “\t”) (**default=“,”**)
  - `append`: ali dodati objekt v datoteko, TRUE / FALSE (default=FALSE)
  - `row.names`: zapiši imena vrstic, TRUE/FALSE (default=TRUE)
  - `col.names`: zapiši imena stolpcev, TRUE/FALSE (default=TRUE)

# Pisanje datotek

- **Evropski csv**

- `write.csv2(objekt, potDoDatoteke/ImeDatoteke)`
- parametri (neobvezni) `?write.table()`
  - `quote`: zapiši znake z navednicami, TRUE / FALSE (**default=TRUE**)
  - `sep`: kaj razmejuje stolpce, “ločilo” (“,”, “.”, “ “, “\t”) (**default=“;”**)
  - `append`: ali dodati objekt v datoteko, TRUE / FALSE (default=FALSE)
  - `row.names`: zapiši imena vrstic, TRUE/FALSE (default=TRUE)
  - `col.names`: zapiši imena stolpcev, TRUE/FALSE (default=TRUE)

# Primer pisanja tabele

```
Shranimo kandidate z več kot 5 točkami v novo tabelo
```

```
> test0pravljen <- test[test$Tocke > 5,]
```

```
Zapišemo tabelo kot csv datoteko
```

```
> write.csv(test0pravljen,
```

```
+ "Test0pravljen.csv", —————▶ Ime tabele
```

```
+ row.names=FALSE, —————▶ Nočemo shraniti imena vrstic
```

```
+ quote=FALSE) —————▶
 Nočemo shraniti znakov z navednicami
```

# Vaja

- ModulA.Rmd / ModulA.R:  
Osnove dela z datotekami

1. Osnove R-a in R-studia (izvršitev kode, paketi)
2. Osnovni podatkovni tipi in strukture
3. Osnovne operacije
4. Osnove dela z datotekami
- 5. Preurejanje podatkov**
6. Povzemanje podatkov
7. Pomoč za R in dobra praksa pisanja kode

# Preurejanje podatkov

- Združevanje / delitev podatkov
- Preurejanje stolpcev / vrstic (“vrtenje” tabel)

# Spajanje podatkov

- Spajanje vektorjev (stolpcev) v tabelo:
  - funkcija **cbind**(vektor1, vektor2 ...) = column bind
  - Output: matrika

```
> Ime
[1] "Maja" "Jure" "Luka"
> Priimek
[1] "Novak" "Kovač" "Medved"
```



```
> cbind(Ime, Priimek)
 Ime Priimek
[1,] "Maja" "Novak"
[2,] "Jure" "Kovač"
[3,] "Luka" "Medved"
```



# Spajanje podatkov

- Spajanje stolpcev tabel z enakim številom vrstic  
funkcija **`cbind(tabela1, tabela2 ...)`**

```
> tabela1
```

|   | ID | Ime  |
|---|----|------|
| 1 | 1  | Maja |
| 2 | 2  | Jure |
| 3 | 3  | Luka |

```
> tabela2
```

|   | Priimek | Starost |
|---|---------|---------|
| 1 | Novak   | 31      |
| 2 | Kovač   | 40      |
| 3 | Medved  | 35      |



```
> cbind(tabela1, tabela2)
```

|   | ID | Ime  | Priimek | Starost |
|---|----|------|---------|---------|
| 1 | 1  | Maja | Novak   | 31      |
| 2 | 2  | Jure | Kovač   | 40      |
| 3 | 3  | Luka | Medved  | 35      |

# Spajanje podatkov

- Spajanje vrstic tabel z enakim številom in imeni stolpcev  
funkcija **rbind**(tabela1, tabela2 ...) = row bind

```
> tabela1
```

|   | ID | Ime  |
|---|----|------|
| 1 | 1  | Maja |
| 2 | 2  | Jure |
| 3 | 3  | Luka |

```
> tabela2
```

|   | ID | Ime   |
|---|----|-------|
| 1 | 4  | Eva   |
| 2 | 5  | Tina  |
| 3 | 6  | Marko |



```
> rbind(tabela1, tabela2)
```

|   | ID | Ime   |
|---|----|-------|
| 1 | 1  | Maja  |
| 2 | 2  | Jure  |
| 3 | 3  | Luka  |
| 4 | 4  | Eva   |
| 5 | 5  | Tina  |
| 6 | 6  | Marko |

# Uparjanje podatkov

- Združevanje tabel na podlagi ključa / -ev

funkcija **merge** (tabela1, tabela2, by="ključ")

- Ključ = ime/-na stolpca/-ev za združevanje
- by → enako ime stolpca v obeh tabelah
- by.x in by.y → različna imena stolpcev v tabelah

# Uparjanje podatkov

- Združevanje tabel na podlagi ključa / -ev

funkcija **merge** (tabela1, tabela2, by="ključ")

> tabela1

|   | ID | Ime  |
|---|----|------|
| 1 | 1  | Maja |
| 2 | 2  | Jure |
| 3 | 3  | Luka |

> tabela2

|   | ID | Priimek |
|---|----|---------|
| 1 | 1  | Novak   |
| 2 | 2  | Kovač   |
| 3 | 3  | Medved  |



> merge(tabela1, tabela2, by="ID")

|   | ID | Ime  | Priimek |
|---|----|------|---------|
| 1 | 1  | Maja | Novak   |
| 2 | 2  | Jure | Kovač   |
| 3 | 3  | Luka | Medved  |

# “Vrtenje” tabel

- Če imamo isti atribut v različnih stolpcih in ga želimo zbrati v en stolpec:

**messy**

| id | trt       | work.T1    | home.T1   | work.T2   | home.T2    |
|----|-----------|------------|-----------|-----------|------------|
| 1  | treatment | 0.08513597 | 0.6158293 | 0.1135090 | 0.05190332 |
| 2  | control   | 0.22543662 | 0.4296715 | 0.5959253 | 0.26417767 |
| 3  | treatment | 0.27453052 | 0.6516557 | 0.3580500 | 0.39879073 |
| 4  | control   | 0.27230507 | 0.5677378 | 0.4288094 | 0.83613414 |

**tidier**

| id | trt       | key     | time       |
|----|-----------|---------|------------|
| 1  | treatment | work.T1 | 0.08513597 |
| 2  | control   | work.T1 | 0.22543662 |
| 3  | treatment | work.T1 | 0.27453052 |
| 4  | control   | work.T1 | 0.27230507 |
| 1  | treatment | home.T1 | 0.61582931 |
| 2  | control   | home.T1 | 0.42967153 |
| 3  | treatment | home.T1 | 0.65165567 |
| 4  | control   | home.T1 | 0.56773775 |
| 1  | treatment | work.T2 | 0.11350898 |
| 2  | control   | work.T2 | 0.59592531 |
| 3  | treatment | work.T2 | 0.35804998 |
| 4  | control   | work.T2 | 0.42880942 |
| 1  | treatment | home.T2 | 0.05190332 |
| 2  | control   | home.T2 | 0.26417767 |
| 3  | treatment | home.T2 | 0.39879073 |
| 4  | control   | home.T2 | 0.83613414 |

# “Vrtenje” tabel

- Knjižnica `reshape` in funkcija `melt()`
- Imena stolpcev v spremenljivko

funkcija `melt` (`data`,  
                  `id.vars`,  
                  `measure.vars`)

- `id.vars` = ime stolpca ali vektor imen stolpcev z identifikatorji
- `measure.vars` = ime stolpca ali vektor imen z merjenimi vrednostmi

# “Vrtenje” tabel

```
> temp
```

|   | Mesec  | Ljubljana | Maribor |
|---|--------|-----------|---------|
| 1 | Julij  | 34        | 32      |
| 2 | Julij  | 28        | 30      |
| 3 | Avgust | 30        | 26      |
| 4 | Avgust | 31        | 26      |



```
> melt(data = temp, id.vars = "Mesec")
```

|   | Mesec  | variable  | value |
|---|--------|-----------|-------|
| 1 | Julij  | Ljubljana | 34    |
| 2 | Julij  | Ljubljana | 28    |
| 3 | Avgust | Ljubljana | 30    |
| 4 | Avgust | Ljubljana | 31    |
| 5 | Julij  | Maribor   | 32    |
| 6 | Julij  | Maribor   | 30    |
| 7 | Avgust | Maribor   | 26    |
| 8 | Avgust | Maribor   | 26    |

variable in value sta privzeti imeni za novonastala stolpca

Ime za variable stolpec nastavimo z variable\_name parametrom

# “Vrtenje” tabel

```
> temp
```

|   | Mesec  | Ljubljana | Maribor |
|---|--------|-----------|---------|
| 1 | Julij  | 34        | 32      |
| 2 | Julij  | 28        | 30      |
| 3 | Avgust | 30        | 26      |
| 4 | Avgust | 31        | 26      |



```
> melt(data = temp,
+ measure.vars = c("Ljubljana", "Maribor"))
```

|   | Mesec  | variable  | value |
|---|--------|-----------|-------|
| 1 | Julij  | Ljubljana | 34    |
| 2 | Julij  | Ljubljana | 28    |
| 3 | Avgust | Ljubljana | 30    |
| 4 | Avgust | Ljubljana | 31    |
| 5 | Julij  | Maribor   | 32    |
| 6 | Julij  | Maribor   | 30    |
| 7 | Avgust | Maribor   | 26    |
| 8 | Avgust | Maribor   | 26    |

variable in value sta privzeti imeni za novonastala stolpca

Ime za value stolpec nastavimo z value\_name parametrom



# Naprednejše funkcije za preurejanje - `tidyr`

- Knjižnica `tidyr`
- `tidyr` operira s podatkovno strukturo `tibble`
  - Poenostavljen `data.frame`
  - Omogoča manj, več se pritožuje → prej odkrijemo napako in poenostavimo kodo

# Naprednejše funkcije za preurejanje

- `pivot_longer`: imena stolpcev v spremenljivko → daljšanje tabele (manj stolpcev, več vrstic)

```
pivot_longer(data = tabela,
 cols = stolpci za združevanje,
 names_to= ime stolpca spremenljivke,
 values_to=ime stolpca vrednosti)
```

# Naprednejše funkcije za preurejanje

```
> temp
```

|   | Mesec  | Ljubljana | Maribor |
|---|--------|-----------|---------|
| 1 | Julij  | 34        | 32      |
| 2 | Julij  | 28        | 30      |
| 3 | Avgust | 30        | 26      |
| 4 | Avgust | 31        | 26      |



```
> pivot_longer(temp,
+ cols = c(Ljubljana, Maribor),
+ names_to = "Mesto",
+ values_to = "Temp")
```

```
A tibble: 8 x 3
```

|   | Mesec  | Mesto     | Temp  |
|---|--------|-----------|-------|
|   | <chr>  | <chr>     | <dbl> |
| 1 | Julij  | Ljubljana | 34    |
| 2 | Julij  | Maribor   | 32    |
| 3 | Julij  | Ljubljana | 28    |
| 4 | Julij  | Maribor   | 30    |
| 5 | Avgust | Ljubljana | 30    |
| 6 | Avgust | Maribor   | 26    |
| 7 | Avgust | Ljubljana | 31    |
| 8 | Avgust | Maribor   | 26    |

# Naprednejše funkcije za preurejanje

- `pivot_wider`: spremenljivka v ime stolpca → širjenje tabele (več stolpcev, manj vrstic)

```
pivot_wider(data = tabela,
 id_cols = unikaten identifikatorjev,
 names_from=ime stolpca spremenljivke,
 values_from=ime stolpca vrednosti)
```

# Naprednejše funkcije za preurejanje

```
> tempD
```

```
A tibble: 8 x 4
```

|   | Mesec<br><chr> | Datum<br><dbl> | Mesto<br><chr> | Temp<br><dbl> |
|---|----------------|----------------|----------------|---------------|
| 1 | Julij          | 1              | Ljubljana      | 34            |
| 2 | Julij          | 1              | Maribor        | 32            |
| 3 | Julij          | 15             | Ljubljana      | 28            |
| 4 | Julij          | 15             | Maribor        | 30            |
| 5 | Avgust         | 1              | Ljubljana      | 30            |
| 6 | Avgust         | 1              | Maribor        | 26            |
| 7 | Avgust         | 15             | Ljubljana      | 31            |
| 8 | Avgust         | 15             | Maribor        | 26            |



```
> pivot_wider(tempD,
```

```
+ id_cols = c(Mesec, Datum),
+ names_from = Mesto,
+ values_from = Temp)
```

```
A tibble: 4 x 4
```

|   | Mesec<br><chr> | Datum<br><dbl> | Ljubljana<br><dbl> | Maribor<br><dbl> |
|---|----------------|----------------|--------------------|------------------|
| 1 | Julij          | 1              | 34                 | 32               |
| 2 | Julij          | 15             | 28                 | 30               |
| 3 | Avgust         | 1              | 30                 | 26               |
| 4 | Avgust         | 15             | 31                 | 26               |

Unikaten ID

# Naprednejše funkcije za preurejanje

- unite: zlepi stolpce

```
unite(data = tabela,
 col = ime novega stolpca,
 ... = stolpci za združevanje,
 sep = ločilo,
 remove = ali odstranim ... stolpce)
```

> temp

|   | Mesec  | Ljubljana | Maribor | Datum |
|---|--------|-----------|---------|-------|
| 1 | Julij  | 34        | 32      | 1     |
| 2 | Julij  | 28        | 30      | 15    |
| 3 | Avgust | 30        | 26      | 1     |
| 4 | Avgust | 31        | 26      | 15    |



> unite(temp, DatumMesec, Datum, Mesec, sep="\_")

|   | DatumMesec | Ljubljana | Maribor |
|---|------------|-----------|---------|
| 1 | 1_Julij    | 34        | 32      |
| 2 | 15_Julij   | 28        | 30      |
| 3 | 1_Avgust   | 30        | 26      |
| 4 | 15_Avgust  | 31        | 26      |

# Naprednejše funkcije za preurejanje

- `separate`: razdruži stolpec (naprednejša verzija z regex: `extract`)

```
separate(data = tabela,
 col = stolpec za razdružitev,
 into = imena ustvarjenih stolpcev,
 sep = ločilo)
```

```
> tempU
```

|   | DatumMesec | Ljubljana | Maribor |
|---|------------|-----------|---------|
| 1 | 1_Julij    | 34        | 32      |
| 2 | 15_Julij   | 28        | 30      |
| 3 | 1_Avgust   | 30        | 26      |
| 4 | 15_Avgust  | 31        | 26      |



```
> separate(tempU,
+ col = DatumMesec,
+ into = c("Datum", "Mesec"),
+ sep="_")
```


|   | Datum | Mesec  | Ljubljana | Maribor |
|---|-------|--------|-----------|---------|
| 1 | 1     | Julij  | 34        | 32      |
| 2 | 15    | Julij  | 28        | 30      |
| 3 | 1     | Avgust | 30        | 26      |
| 4 | 15    | Avgust | 31        | 26      |

# Naprednejše funkcije za preurejanje

- `expand`: vse možne kombinacije navedenih spremenljivk

```
expand(data = tabela,
 ... = stolpci za primerjavo)
```

```
> tempP
A tibble: 4 x 5
 Mesec Datum Ljubljana Maribor Cas
 <chr> <dbl> <dbl> <dbl> <chr>
1 Julij 1 34 32 Zjutraj
2 Julij 15 28 30 Zvecer
3 Avgust 1 30 26 Zvecer
4 Avgust 15 31 26 Zvecer
```



```
> tidyr::expand(tempP, Mesec, Cas)
A tibble: 4 x 2
 Mesec Cas
 <chr> <chr>
1 Avgust Zjutraj
2 Avgust Zvecer
3 Julij Zjutraj
4 Julij Zvecer
```



# Naprednejše funkcije za preurejanje

- `chop`: združevanje podatkov v sezname glede na unikatne kombinacije spremenljivk

```
chop(data = tabela,
 cols = stolpci za združevanje)
```

```
> tempS
```

|   | Mesec  | Ljubljana | Maribor |
|---|--------|-----------|---------|
| 1 | Julij  | 34        | 32      |
| 2 | Julij  | 28        | 30      |
| 3 | Avgust | 30        | 26      |
| 4 | Avgust | 31        | 26      |



```
> chop(tempS, c(Ljubljana, Maribor))
```

|   | Mesec  | Ljubljana | Maribor |
|---|--------|-----------|---------|
| 1 | Julij  | 34, 28    | 32, 30  |
| 2 | Avgust | 30, 31    | 26, 26  |

Različno obnašanje, če je vhodni podatkovni tip  
tibble ali data.frame

# Naprednejše funkcije za preurejanje

- `chop`: združevanje podatkov v sezname glede na unikatne kombinacije spremenljivk

```
chop(data = tabela,
 cols = stolpci)
```

Funkcija `nest` ima podobno obnašanje;  
Le da namesto zbiranje v sezname  
podatke zbere v posamične tabele

```
> tempS
```

|   | Mesec  | Ljubljana | Maribor |
|---|--------|-----------|---------|
| 1 | Julij  | 34        | 32      |
| 2 | Julij  | 28        | 30      |
| 3 | Avgust | 30        | 26      |
| 4 | Avgust | 31        | 26      |



```
> chop(tempS, c(Ljubljana, Maribor))
```

|   | Mesec  | Ljubljana | Maribor |
|---|--------|-----------|---------|
| 1 | Julij  | 34, 28    | 32, 30  |
| 2 | Avgust | 30, 31    | 26, 26  |

Različno obnašanje, če je vhodni podatkovni tip  
`tibble` ali `data.frame`

# Naprednejše funkcije za preurejanje

- `unchop`: razdruževanje podatkov iz seznamov

```
unchop(data = tabela,
 cols = stolpci za razdruževanje)
```

```
> tempSC
```

|   | Mesec  | Ljubljana | Maribor |
|---|--------|-----------|---------|
| 1 | Julij  | 34, 28    | 32, 30  |
| 2 | Avgust | 30, 31    | 26, 26  |



```
> unchop(tempSC, c(Ljubljana, Maribor))
```

|   | Mesec  | Ljubljana | Maribor |
|---|--------|-----------|---------|
| 1 | Julij  | 34        | 32      |
| 2 | Julij  | 28        | 30      |
| 3 | Avgust | 30        | 26      |
| 4 | Avgust | 31        | 26      |

# Naprednejše funkcije za preurejanje

- `pack`: “sesedanje” večih stolpcev v eno stolpec-tabelo

```
pack(data = tabela,
 cols = stolpci za “sesedanje”)
```

unpack

```
> tempS
```

|   | Mesec  | Ljubljana | Maribor |
|---|--------|-----------|---------|
| 1 | Julij  | 34        | 32      |
| 2 | Julij  | 28        | 30      |
| 3 | Avgust | 30        | 26      |
| 4 | Avgust | 31        | 26      |



```
> pack(tempS, Mesto=c(Ljubljana, Maribor))
```

|   | Mesec  | Mesto.Ljubljana | Mesto.Maribor |
|---|--------|-----------------|---------------|
| 1 | Julij  | 34              | 32            |
| 2 | Julij  | 28              | 30            |
| 3 | Avgust | 30              | 26            |
| 4 | Avgust | 31              | 26            |

Prvi stolpec

Drugi stolpec-tabela

# “Cevovodna” obdelava podatkov

- Paket `magittr`, ampak vključen v `tidyr`
- Direktno “podajanje” podatkov naslednji funkciji
- Operator `%>%` (Ctrl+M)
- `podatki %>% funkcija1(parametri) %>% funkcija2(parametri) ...`

# “Cevovodna” obdelava podatkov

```
> temp %>% unite(DatumMesec, Datum, Mesec) %>%
+ pivot_longer(c(Ljubljana, Maribor))
```

```
A tibble: 8 x 3
```

|   | DatumMesec<br><chr> | name<br><chr> | value<br><dbl> |
|---|---------------------|---------------|----------------|
| 1 | 1_Julij             | Ljubljana     | 34             |
| 2 | 1_Julij             | Maribor       | 32             |
| 3 | 15_Julij            | Ljubljana     | 28             |
| 4 | 15_Julij            | Maribor       | 30             |
| 5 | 1_Avgust            | Ljubljana     | 30             |
| 6 | 1_Avgust            | Maribor       | 26             |
| 7 | 15_Avgust           | Ljubljana     | 31             |
| 8 | 15_Avgust           | Maribor       | 26             |

# Vaja

- ModulA.Rmd / ModulA.R:  
Preurejanje podatkov

1. Osnove R-a in R-studia (izvršitev kode, paketi)
2. Osnovni podatkovni tipi in strukture
3. Osnovne operacije
4. Osnove dela z datotekami
5. Preurejanje podatkov
- 6. Povzemanje podatkov**
7. Pomoč za R in dobra praksa pisanja kode



# Povzemanje podatkov

- Srednje vrednosti in razpršenost podatkov
  - Povprečje, mediana, modus, razpon, variance, standardni odklon, kvantili
- Povzemanje po ravni neke spremenljivke

# Srednje vrednosti

- Vsota: `sum(vektor)`
  - Vsota stolpcev tabele:  
`colSums(data, na.rm = TRUE/FALSE)`
  - Vsota vrstic tabele:  
`rowSums(data, na.rm = TRUE/FALSE)`
  - Funkciji povzameta vse vrstice/stolpce razen prve(ga)
  - stolpci/vrstice morajo imeti numerične/logične vrednosti

# Srednje vrednosti

- Povprečje: `mean(vektor)`
  - Povprečje stolpcev tabele:  
`colMeans(data, na.rm = TRUE/FALSE)`
  - Vsota vrstic tabele:  
`rowMeans(data, na.rm = TRUE/FALSE)`
  - Funkciji povzameta vse vrstice/stolpce razen prve(ga)
  - stolpci/vrstice morajo imeti numerične/logične vrednosti

# Srednje vrednosti

- Mediana: `median(vektor)`
- Modus: `mfv()` v paketu `modeest` ali s tabeliranje podatkov `table()`
- Varianca: `var(vektor)`
- Standardni odklon: `sd(vektor)`
- Kvantili: `quantile(vektor)`
  - Specifični kvantil: dodamo parameter `probs = c(kvantili))`

```
> temp
```

|   | Mesec  | Ljubljana | Maribor |
|---|--------|-----------|---------|
| 1 | Julij  | 34        | 32      |
| 2 | Avgust | 28        | 30      |
| 3 | Julij  | 30        | 26      |
| 4 | Avgust | 31        | 26      |

Samo 2. in 3.  
stolpec sta  
numerična

Vsota vrstic

```
> rowSums(temp[,2:3])
[1] 66 58 56 57
```

Povprečje stolpcev

```
> colMeans(temp[,2:3])
Ljubljana Maribor
30.75 28.50
```

Kvantili temperatur v Ljubljani

```
> quantile(temp$Ljubljana,
+ probs = c(0.05, 0.95))
5% 95%
28.30 33.55
```

Varianca stolpcev

```
> var(temp[,2:3])
Ljubljana Maribor
Ljubljana 6.250000 2.833333
Maribor 2.833333 9.000000
```

varianca

kovarianca

# Povzemanje podatkov po skupinah

- Povzemanje po ravni neke spremenljivke (npr. povprečja različnih skupin)
- Povzemanje s trivialno funkcijo
- Funkcija aggregate

**aggregate(formula, FUN="funkcija")**

formula:

odvisna\_spremenljivka ~ neodvisna\_spremenljivka1 +  
neodvisna\_spremenljivka2 + ...

funkcija: ime poljubne funkcije

# Povzemanje podatkov po skupinah



Agregiranje po eni spremenljivki

```
> tempM
```

|   | Mesec  | Mesto     | value |
|---|--------|-----------|-------|
| 1 | Julij  | Ljubljana | 34    |
| 2 | Avgust | Ljubljana | 28    |
| 3 | Julij  | Ljubljana | 30    |
| 4 | Avgust | Ljubljana | 31    |
| 5 | Julij  | Maribor   | 32    |
| 6 | Avgust | Maribor   | 30    |
| 7 | Julij  | Maribor   | 26    |
| 8 | Avgust | Maribor   | 26    |

```
> aggregate(tempM$value ~ tempM$Mesec, FUN="mean")
```

|   | tempM\$Mesec | tempM\$value |
|---|--------------|--------------|
| 1 | Avgust       | 28.75        |
| 2 | Julij        | 30.50        |



Agregiranje po več spremenljivkah

```
> aggregate(tempM$value ~ tempM$Mesec + tempM$Mesto, FUN="mean")
```

|   | tempM\$Mesec | tempM\$Mesto | tempM\$value |
|---|--------------|--------------|--------------|
| 1 | Avgust       | Ljubljana    | 29.5         |
| 2 | Julij        | Ljubljana    | 32.0         |
| 3 | Avgust       | Maribor      | 28.0         |
| 4 | Julij        | Maribor      | 29.0         |

# Povzemanje podatkov: paket dplyr

- `library(dplyr)`
- Ustvarimo skupine: funkcija `group_by()`

```
group_by(tabela,
 spremenljivke za združevanje)
```



# Povzemanje podatkov: paket dplyr

```
> tempM
```

|   | Mesec  | Mesto     | value |
|---|--------|-----------|-------|
| 1 | Julij  | Ljubljana | 34    |
| 2 | Avgust | Ljubljana | 28    |
| 3 | Julij  | Ljubljana | 30    |
| 4 | Avgust | Ljubljana | 31    |
| 5 | Julij  | Maribor   | 32    |
| 6 | Avgust | Maribor   | 30    |
| 7 | Julij  | Maribor   | 26    |
| 8 | Avgust | Maribor   | 26    |



```
> group_by(tempM, Mesto, Mesec)
```

```
A tibble: 8 x 3
```

```
Groups: Mesto, Mesec [4]
```

|   | Mesec  | Mesto     | value |
|---|--------|-----------|-------|
|   | <chr>  | <fct>     | <dbl> |
| 1 | Julij  | Ljubljana | 34    |
| 2 | Avgust | Ljubljana | 28    |
| 3 | Julij  | Ljubljana | 30    |
| 4 | Avgust | Ljubljana | 31    |
| 5 | Julij  | Maribor   | 32    |
| 6 | Avgust | Maribor   | 30    |
| 7 | Julij  | Maribor   | 26    |
| 8 | Avgust | Maribor   | 26    |

Funkcija ne spremeni podatkov – samo ustvari skupine

# Povzemanje podatkov: paket dplyr

- Povzemanje podatkov: funkcija `summarize()`

```
summarize(tabela,
 imePovzetihPodatkov = funkcija(podatki),
 imePovzetihPodatkov2 = funkcija(podatki),
 ...)
```

# Povzemanje podatkov: paket dplyr

```
> tempM
```

|   | Mesec  | Mesto     | value |
|---|--------|-----------|-------|
| 1 | Julij  | Ljubljana | 34    |
| 2 | Avgust | Ljubljana | 28    |
| 3 | Julij  | Ljubljana | 30    |
| 4 | Avgust | Ljubljana | 31    |
| 5 | Julij  | Maribor   | 32    |
| 6 | Avgust | Maribor   | 30    |
| 7 | Julij  | Maribor   | 26    |
| 8 | Avgust | Maribor   | 26    |



```
> summarize(tempM, povpTemp = mean(value))
 povpTemp
1 29.625
```

POZOR: Funkcija `summarize()` se nahaja tudi v paketu `plyr` in ima drugačno obnašanje! Eksplicitno lahko funkcijo iz paketa pokličemo `paket::funkcija()` → v tem primeru `dplyr::summarize()`

# Povzemanje podatkov: paket dplyr

- `group_by()` in `summarize()` sami po sebi nista najbolj uporabni funkciji
- AMPAK dplyr omogoča “cevvodno obdelavo podatkov” (kot tidy) z operatorjem `%>%`
  - lahko podajamo podatke med funkcijami: izhodni podatki ene funkcije so vhodni podatki za drugo

```
podatki %>% group_by() %>% summarize()
```

# Povzemanje podatkov: paket dplyr

```
> tempM
```

|   | Mesec  | Mesto     | value |
|---|--------|-----------|-------|
| 1 | Julij  | Ljubljana | 34    |
| 2 | Avgust | Ljubljana | 28    |
| 3 | Julij  | Ljubljana | 30    |
| 4 | Avgust | Ljubljana | 31    |
| 5 | Julij  | Maribor   | 32    |
| 6 | Avgust | Maribor   | 30    |
| 7 | Julij  | Maribor   | 26    |
| 8 | Avgust | Maribor   | 26    |

```
> tempM %>% group_by(Mesec, Mesto) %>%
+ summarize(povpTemp = mean(value),
+ varTemp = var(value))
'summarise()' regrouping output by 'Mesec' (ov
A tibble: 4 x 4
Groups: Mesec [2]
 Mesec Mesto povpTemp varTemp
 <chr> <fct> <dbl> <dbl>
1 Avgust Ljubljana 29.5 4.5
2 Avgust Maribor 28 8
3 Julij Ljubljana 32 8
4 Julij Maribor 29 18
```



# Povzemanje podatkov

- V “cevovod” lahko združimo različne funkcije iz različnih paketov

```
podatki %>%
```

```
pivot_longer() %>%
```

```
unite() %>%
```

```
group_by() %>%
```

```
summarize()
```

# Povzemanje podatkov

## Vhodni podatki

> temp

|   | Mesec  | Ljubljana | Maribor | Datum |
|---|--------|-----------|---------|-------|
| 1 | Julij  | 34        | 32      | 1     |
| 2 | Avgust | 28        | 30      | 15    |
| 3 | Julij  | 30        | 26      | 1     |
| 4 | Avgust | 31        | 26      | 15    |

# Povzemanje podatkov

```
> temp %>% pivot_longer(cols = c(Ljubljana, Maribor), names_to="Mesto") %>%
+ unite("MesecDatum", Mesec, Datum, sep="_", remove = F) %>%
+ group_by(MesecDatum, Mesto) %>%
+ summarize(povpTemp = mean(value))
`summarise()` regrouping output by 'MesecDatum' (override with `groups` arg)
A tibble: 4 x 3
Groups: MesecDatum [2]
 MesecDatum Mesto povpTemp
 <chr> <chr> <dbl>
1 Avgust_15 Ljubljana 29.5
2 Avgust_15 Maribor 28
3 Julij_1 Ljubljana 32
4 Julij_1 Maribor 29
```






# Vaja

- ModulA.Rmd / ModulA.R:  
Povzemanje podatkov

1. Osnove R-a in R-studia (izvršitev kode, paketi)
2. Osnovni podatkovni tipi in strukture
3. Osnovne operacije
4. Osnove dela z datotekami
5. Preurejanje podatkov
6. Povzemanje podatkov
7. Pomoč za R in dobra praksa pisanja kode

# Pomoč za R

- ?imeFunkcije
- CRAN: priročniki paketov
-  - problema ali napake
- google za R: <https://rseek.org/> 
-  **stackoverflow** - <https://stackoverflow.com/>

# Pomoč za učenje R

- Vgrajena pomoč: `demo()`
- Interaktivne vaje: paket `discover`
- Spletni interaktivni tečaji:
  - DataCamp:  
<https://learn.datacamp.com/skill-tracks/r-programming>  
<https://learn.datacamp.com/career-tracks/data-scientist-with-r>  
<https://learn.datacamp.com/career-tracks/data-analyst-with-r>
  - Codecademy: <https://www.codecademy.com/learn/learn-r>
- Spletne knjige z nalogami
  - <https://jilymackay.github.io/RatRDSVS/index.html>
  - R za data science: <https://r4ds.had.co.nz/>

# Najpogostejše napake

- Pozabimo oklepaj / zaklepaj
- Pozabimo navednice

```
> rezultat <- (4 + 5 * 7
+
```

```
> seznam <- c("banane", "mleko", "jajca")
Error: unexpected symbol in "seznam <- c("banane", "mleko", "jajca"
```

- Napačno ime spremenljivke

```
> visina <- 180
> teza <- 80
> viaina * teza
Error: object 'viaina' not found
```

# Najpogostejše napake

- Napačno ime funkcije

```
> means(c(1,2,3))
Error in means(c(1, 2, 3)) : could not find function "means"
```

- Napačno ime imenika / mape / datoteke

```
> tabela <- read.table("/home/jana/tabela.csv")
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
cannot open file '/home/jana/tabela.csv': No such file or directory
```

# Najpogostejše napake

- Napačno izbrani elementi tabele

```
Ime Priimek Starost Tocke Predmet ImePriimek
1 Maja Novak 25 9.25 Matematika Maja_Novak
2 Jure Kovač 46 9.02 Matematika Jure_Kovač
> test$Ime[test$Tocke > 5,]
Error in test$Ime[test$Tocke > 5,] : incorrect number of dimensions
```

- Aplikacija funkcije na napačen podatkovni tip

```
> mean(test$Ime)
[1] NA
Warning message:
In mean.default(test$Ime) :
 argument is not numeric or logical: returning NA
```

# Dobra praksa pisanja kode

- Opis programa na začetku datoteke
- Naloži potrebne knjižnice na začetku
- Previdno z relativnimi potmi do datotek
- Komentiraj kodo!!! → kaj koda počne, kaj so spremenljivke, za označitev sekcij kode
- Definiraj funkcije na začetku datoteke (ali v posebni datoteki)



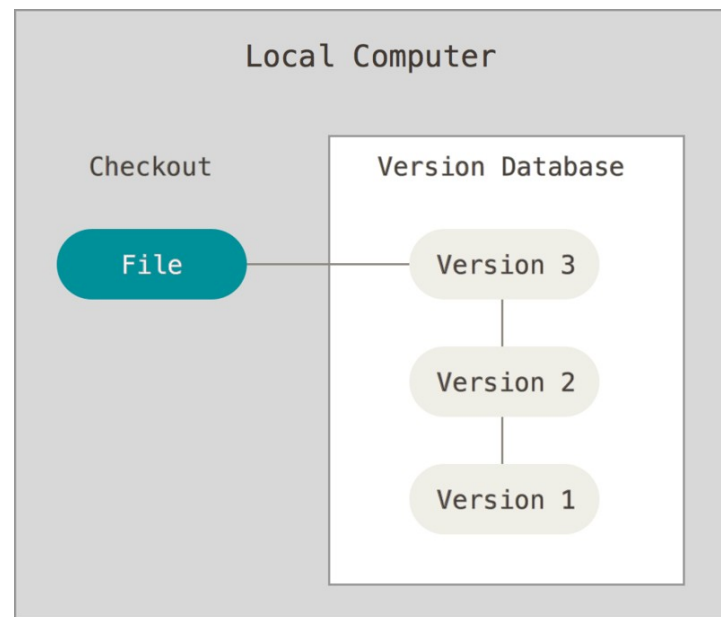


# Dobra praksa pisanja kode

- Dosledno poimenuj spremenljivke in oblikuj kodo
- Pišimo kodo za večkratno uporabo
  - Razbij kodo v majhne, samostojne dele
  - Ne ponavljaj pogostih operacij → funkcije
- Uporabljaljaj nadzor različic (“version control”)

# Nadzor različic: Git

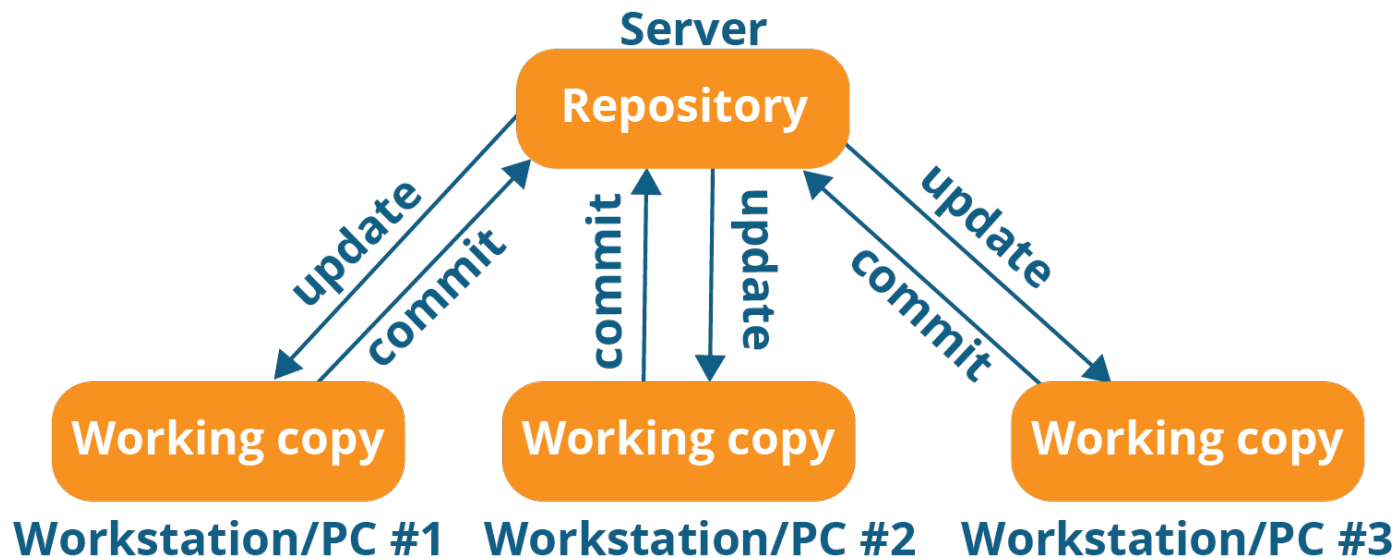
- github / gitlab / bitbucket
- Programi za nadzor različic:
  - Zapis vseh spremembe v datoteki tekom časa
  - Priklic katerokoli izmed prejšnjih verzij datoteke
  - Povrnitev v prejšnje stanje
  - Primenjava sprememb med različicami ...



# Github / Gitlab / Bitbucket

- Programi za nadzor različic:
  - Sočasno urejanje datoteke s strani večih uporabnikov

## Centralized version control system



# Github / Gitlab / Bitbucket

- 1) Ustvarimo “remote” repozitorij (projekt)
- 2) Kloniramo repozitorij na lokalni računalnik
- 3) Dodamo in pošljemo spremembe
- 4) Potisnemo spremembe na remote repozitorij
- 5) Urejanje nesoglasij v kodi
- 6) Naslednjič: povlečemo spremembe in nadaljujemo z delom

<https://www.atlassian.com/git/tutorials/comparing-workflows>