Univerzitet u Beogradu – Elektrotehnički fakultet (ETF)
Katedra za signale i sisteme

Tehnike obrade biomedicinskih signala 19M051TOBS

Dr Nadica Miljković, vanredni profesor
kabinet 68, nadica.miljkovic@etf.bg.ac.rs

OPERATORI U R

Operatori u R-u

-	Minus, can be unary or binary
+	Plus, can be unary or binary
!	Unary not
~	Tilde, used for model formulae, can be either unary or binary
?	Help
:	Sequence, binary (in model formulae: interaction)
*	Multiplication, binary
/	Division, binary
^	Exponentiation, binary
%x%	Special binary operators, *x* can be replaced by any valid name
%%	Modulus, binary
%/%	Integer divide, binary
%*%	Matrix product, binary
%o%	Outer product, binary
%x%	Kronecker product, binary
%in%	Matching operator, binary (in model formulae: nesting)

<	Less than, binary
>	Greater than, binary
==	Equal to, binary
>=	Greater than or equal to, binary
<=	Less than or equal to, binary
&	And, binary, vectorized
&&	And, binary, not vectorized
|	Or, binary, vectorized
||	Or, binary, not vectorized
<-	Left assignment, binary
->	Right assignment, binary
$	List subset, binary

```
> y <- c(T, T, T, T)
> x <- c(T, F, T, F)
> x & y
[1]  TRUE FALSE  TRUE FALSE
> x && y
[1] TRUE
> x <- c(F, F, T, F)
> x && y
[1] FALSE
```

- Kompletna lista operatora u R-u se može naći na: https://cran.r-project.org/doc/manuals/R-lang.html#Operators (pristupljeno 14. marta 2019. godine).
- Na slici je prikazan i primer upotrebe logičkih operatora.

Operacije sa nizovima – dodatno

```
> EMGsignali <- read.table("EMG.txt")
> head(EMGsignali)
      V1     V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16
1 -0.060 -0.323  0  0  0  0  0  0  0   0   0   0   0   0   0   0
2 -0.060 -0.319  0  0  0  0  0  0  0   0   0   0   0   0   0   0
3 -0.097 -0.297  0  0  0  0  0  0  0   0   0   0   0   0   0   0
4 -0.129 -0.267  0  0  0  0  0  0  0   0   0   0   0   0   0   0
5 -0.130 -0.239  0  0  0  0  0  0  0   0   0   0   0   0   0   0
6 -0.130 -0.204  0  0  0  0  0  0  0   0   0   0   0   0   0   0
> fs <- 1000
> vreme <- seq(0, length(EMGsignali$V1)/fs - 1/fs, by = 1/fs)
> length(vreme)
[1] 39166
> length(EMGsignali$V1)
[1] 39166
>
```

- Za kreiranje određene sekvence brojeva mogu se koristiti funkcije *c()*, *vector()* i operator ':'.
- Međutim, posebno kod formiranja vremenskih osa, pogodno je da se koriste i funkcije tipa *seq()* i *seq_along()*, ali i *rep()* jer se inkrement može podesiti na željenu vrednost.
- Na slici je dat primer formiranja vremenske ose za signal snimljen u fajlu "EMG.txt".

DATUM I VREME

Datum

```
> x <- as.Date("1970-01-01")
> x
[1] "1970-01-01"
> unclass(x)
[1] 0
> x <- as.Date("01.01.1970")
Error in charToDate(x) :
  character string is not in a standard unambiguous format
> x <- as.Date("01.01.1970.")
Error in charToDate(x) :
  character string is not in a standard unambiguous format
> unclass(as.Date("1970-01-02"))
[1] 1
>
```

- U R-u postoji posebna reprezentacija datuma i vremena.
- Datumi su prezentovani kao *Date* klasa podataka.
- Interno, datumi su smešteni kao brojevi dana od 01.01.1970 (redni broj 1).
- Na primeru je pokazano kako je moguće predstaviti *string* kao datum primenom *as.Date()* funkcije. Primetiti da *string* ima odgovarajuću strukturu u R-u (kod bi javio grešku da nema).

Vreme

```
> x <- Sys.time()
> x
[1] "2017-02-18 17:16:37 CET"
> p <- as.POSIXlt(x)
> p
[1] "2017-02-18 17:16:37 CET"
> names(unclass(p))
 [1] "sec"    "min"    "hour"   "mday"   "mon"    "year"
 [7] "wday"   "yday"   "isdst"  "zone"   "gmtoff"
> p$sec
[1] 37.30038
> p$mon
[1] 1
> p$wday
[1] 6
>
```

```
> unclass(x)
[1] 1487434597
>
```

- Vreme je predstavljeno sa dve klase POSIXct i POSIXlt.
- POSIXct – *integer* vrednost
- POSIXlt – lista koja sadrži i druge dodatne informacije (dan u nedelji i sl.)
- Interno, vremena su smeštena kao redni broj sekunde od 01.01.1970.
- Funkcije koje vrše određene operacije sa ovim klasama su:
  - *weekdays()*,
  - *months()* i
  - *quarters()*.
- Funkcija *Sys.time()* daje trenutno vreme i njen rezultat je POSIXct.
- Šta se dobija ako se primeni *unclass()* funkcija na POSIXct podatak o vremenu? Broj sekundi od 01.01.1970.

strptime() funkcija

strptime {base} R Documentation

Date-time Conversion Functions to and from Character

Description

Functions to convert between character representations and objects of classes `"POSIXlt"` and `"POSIXct"` representing calendar dates and times.

Usage

```
## S3 method for class 'POSIXct'
format(x, format = "", tz = "", usetz = FALSE, ...)
## S3 method for class 'POSIXlt'
format(x, format = "", usetz = FALSE, ...)

## S3 method for class 'POSIXt'
as.character(x, ...)

strftime(x, format = "", tz = "", usetz = FALSE, ...)
strptime(x, format, tz = "")
```

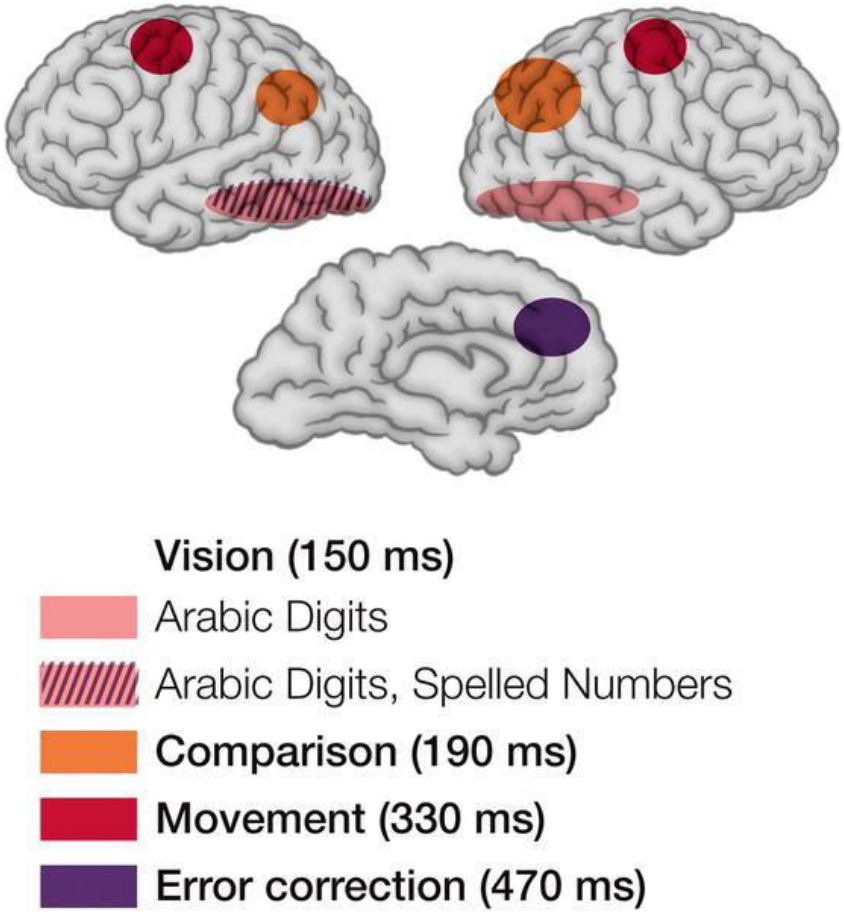
- Ova funkcija se koristi ako su datum i vreme napisani u drugačijem formatu.
- Za detalje korišćenja ove funkcije pogledati *?strptime*.
- Da li postoji nulta godina? Pogledati: https://en.wikipedia.org/wiki/Year_zero.

Operacija za klase datum i vreme

```
> datum1 <- as.Date("1997-05-12")
> datum2 <- as.Date("2017-02-20")
> datum2 - datum1
Time difference of 7224 days
>
```

- Nad podacima tipa datum i vreme mogu se koristiti aritmetičke i logičke operacije.
- Treba voditi računa da se sve operacije nad datumima i vremenima rade nad objektima koji pripadaju istim klasama.
- Ne treba voditi računa o vremenskim zonama i o prestupnim godinama, jer se to dešava automatski. Prestupna godina = *leap year* (eng).
- DODATNO: Datum iz 1997. godine predstavlja datum kada je R postao deo GNU projekta (https://en.wikipedia.org/wiki/R_(programming_language), a datum iz 2017. godine je datum početka letnjeg semestra u školskoj 2016/2017. godini kada je uveden TOBS predmet.

Vreme i mozak

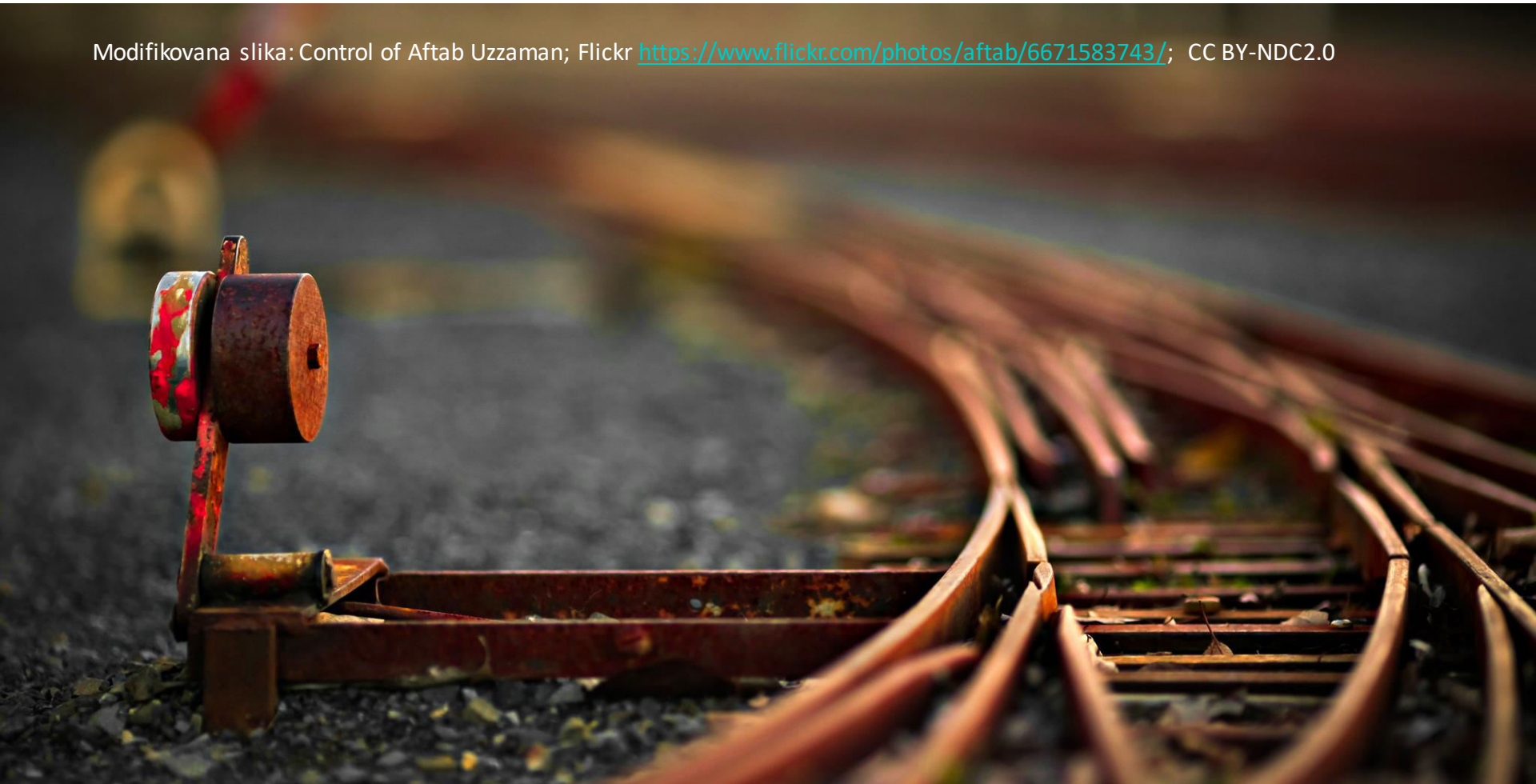


- Kad smo kod vremena, datuma i brojeva
- Na slici su prikazane regije u mozgu koje su aktivne tokom zadatka poređenja brojeva primenom EEG (elektroencefalografije) i fMRI (funkcionalne magnetske rezonance).
- Slika: By http://biology.plosjournals.org/perlserv/?request=get-document&doi=10.1371/journal.pbio.0030051, CC BY-SA 2.5, https://en.wikipedia.org/w/index.php?curid=4776115) odgovara istraživanju koje je predstavljeno u radu:
  - Posner, Michael I. "Timing the brain: Mental chronometry as a tool in neuroscience." *PLoS Biol* 3.2 (2005): e51. doi: 10.1371/journal.pbio.0030051
- Mentalna hronometrija

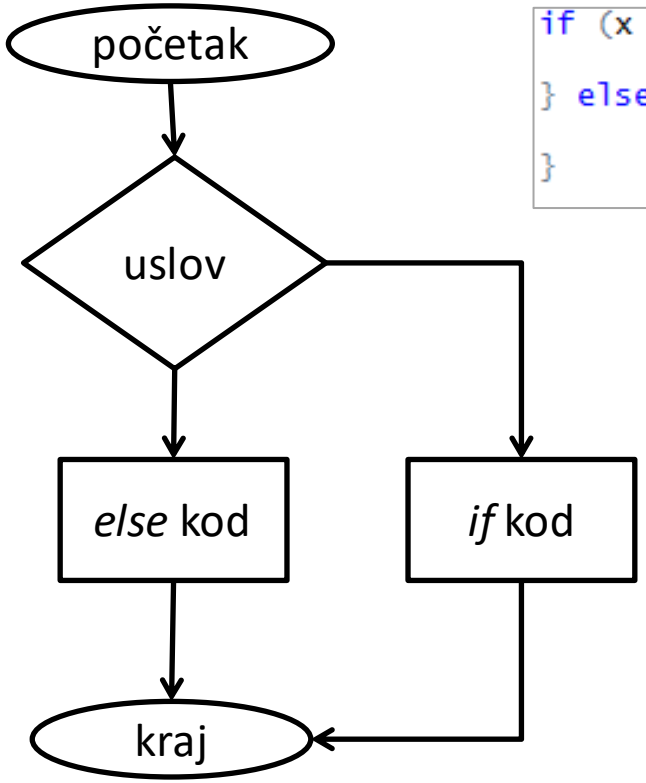
KONTROLNE STRUKTURE

Kontrolne strukture

- Kontrolne strukture u R-u su:
  - *If/else, for, while, repeat, break, next, return*



Kontrolne strukture




```
if (x > 3) {
        y <- "veceOd3"
} else {
        y <- "manjeIliJednako3"
}
```

```
y <- if (x > 3) {
        "veceOd3"
} else {
        "manjeIliJednako3"
}
```

- *if-else* struktura je najčešće korišćena struktura u svim programskim jezicima.
- U R-u postoji i nešto drugačiji zapis u odnosu na druge programske jezike – pogledati sliku gore.
- Zagrade { } se koriste kao početak i kraj, respektivno niza komandi koje je potrebno izvršiti unutar strukture.
- *else* deo nije neophodan – pogledati sliku levo.

for petlja

```
> for (i in 1:10) {
+         print(i)
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

```
x <- c("a", "u", "t", "o", "m", "a", "t", "i", "k", "a")

# prvi način
for (i in 1:10) {
        print(x[i])
}

# drugi način
for (i in seq_along(x)) {
        print(x[i])
}

# treći način
for (letter in x) {
        print(letter)
}

# skraćeno
for (i in 1:10) print(x[i])
```

- *for* petlja: najčešće korišćena petlja u R-u
- Iako zapis može biti jednostavan (u jednoj liniji) bez naznačenog početka i kraja tela petlje, preporuka je da se uvek koriste vitičaste zagrade { i }.
- Šta radi *seq_along()* funkcija?
- Koji je rezultat petlji koje su prikazane na slici desno? Šta bi se dobilo da umesto x[i] stoji samo x? Probajte sami!
- *For* petlje mogu biti ugnježdene (najviše 2-3 nivoa je preporučeno, zašto?).
- Da li iterator sme da bude *i*?

while petlja

```
iterator <- 0

while (iterator < 5) {
        iterator <- iterator + 1
        print(iterator)
}
```

```
iterator <- 0

while (iterator < 5 && iterator >= 0) {
        iterator <- iterator + 1
        print(iterator)
}
```

- *While* petlja – ređe se koristi od *for* petlje. Zašto? Zato što kao rezultat može dati beskonačnu petlju, što se ne može desiti sa *for* petljom.
  - Nekada je to i korisno. Kod namenskih računarskih sistema na primer.
- R nema koncept inkrement operatora, ali korisnik može sam da definiše takvu funkciju.
- Kako od *while* petlje koja je prikazana na slici "napraviti" beskonačnu petlju? *iterator <- Inf*
- *While* petlja može biti korisna kada se ne zna unapred broj iteracija.
- Ako se koristi *while* petlja sa većim brojem uslova, onda se uslovi proveravaju s leva na desno.

Petlje

```
x <- 0
repeat {
        print(x)
        x <- x + 1
        if (x > 10) {
                break
        }
}
```

- *repeat* petlja: je petlja koja se izvršava beskonačan broj puta, sve dok korisnik ne pozove funkciju *break()*.
- Zgodno je koristiti kod estimacije parametara: da li je neka vrednost "dovoljno blizu" definisanog praga, ali u tom slučaju je moguće koristiti i *while* petlju (ali i *for* petlju!).
- Kako bi se primer sa slike realizovao korišćenjem *for* petlje?
- Ako se konvergencija ne ispuni, onda postoji opasnost od beskonačne petlje. -> Trebalo bi postaviti bar još jedan uslov izlaska iz petlje.
- *break()* se može koristiti i sa bilo kojom drugom petljom.
- *next()* se može koristiti za preskakanje iteracije -> u kombinaciji sa *if-else* strukturom može se preskočiti i >1 iteracije.
- *return()* vraća konačnu vrednost i izlazi iz funkcije (prekida tok funkcije).

REZIME

Rezime

- U R-u postoje posebne vrste podataka koje se koriste za smeštanje podataka o vremenu i datumu. Takođe, nad tim podacima se mogu vršiti standardne funkcije (oduzimanje, sabiranje i sl.).
- Kontrolne strukture su korisne jer mogu da omoguće kontrolu nad izvršavanjem programskog koda.
- Poželjno je koristiti *for* petlju kad god je to moguće u kombinaciji sa *if-else* strukturom, kako bi se izbegla beskonačna petlja.

dplyr PAKET

dplyr paket

- Ovaj paket se koristi za rad sa *data frame* podacima.
- Pretpostavke koje bi trebalo da važe za podatke na kojima se primenjuju funkcije dplyr (od eng. *deep layer*) paketa su:
  - postoji jedna opservacija po vrsti (kao u podacima "EMG.txt" – jedna vrsta predstavlja izmerene podatke u jednom vremenskom trenutku)
  - svaka kolona predstavlja neku promeljivu (merni kanal u slučaju podataka u fajlu "EMG.txt")
  - i druge.
- Ovaj paket je razvio Hadley Wickham (Rstudio, https://cran.r-project.org/web/packages/dplyr/dplyr.pdf). Ne postoji nova funkcionalnost u odnosu na osnovne funkcije u R-u, ali su postojeće dodatno poboljšane. Mnoge funkcije su razvijene u C++, pa je izvršavanje tih funkcija relativno brzo, a kod efikasniji.

dplyr funkcije

- Osnovne funkcije koje se koriste u ovom paketu su:
  - *select()*: za odabir podskupa kolona
  - *filter()*: za odabir podskupa vrsta na osnovu logičke opreacije
  - *arrange()*: da se promeni redosled vrsta (oprezno!)
  - *rename()*: da se preimenuju imena promenljivih (preporučeno!)
  - *mutate()*: da se dodaju nove kolone ili da se promene postojeće (preporučeno!)
  - *summarise()*: automatsko generisanje statistike na nekom skupu podataka (srednja vrednost, standardna devijacija i sl.)
- Postoji i drugačija *print()* funkcija koja automatski "sprečava" korisnika da odštampa preveliki broj podataka.
  - U velikom broju slučajeva nema puno smisla da se štampaju svi mogući podaci u konzoli R-a.

dplyr instalacija

```
> install.packages("dplyr")
Installing package into 'C:/Users/Nadica Miljkovic/Documents/R/win-library/3.2'
(as 'lib' is unspecified)
also installing the dependency 'BH'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/BH_1.60.0-2.zip'
Content type 'application/zip' length 15529294 bytes (14.8 MB)
downloaded 14.8 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/dplyr_0.5.0.zip'
Content type 'application/zip' length 2520739 bytes (2.4 MB)
downloaded 2.4 MB

package 'BH' successfully unpacked and MD5 sums checked
package 'dplyr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Nadica Miljkovic\AppData\Local\Temp\RtmpOclU8N\downloaded_packages
> library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

>
```

- Na slici je prikazano kako se instalira dplyr paket.
- Prilikom korišćenja funkcionalnosti nekog od R paketa (koji nije osnovni paket) potrebno je pozvati funkciju *library()*.
- Paket može da se instalira i sa GitHub-a pozivanjem komande *install_github("hadley/dplyr")*.

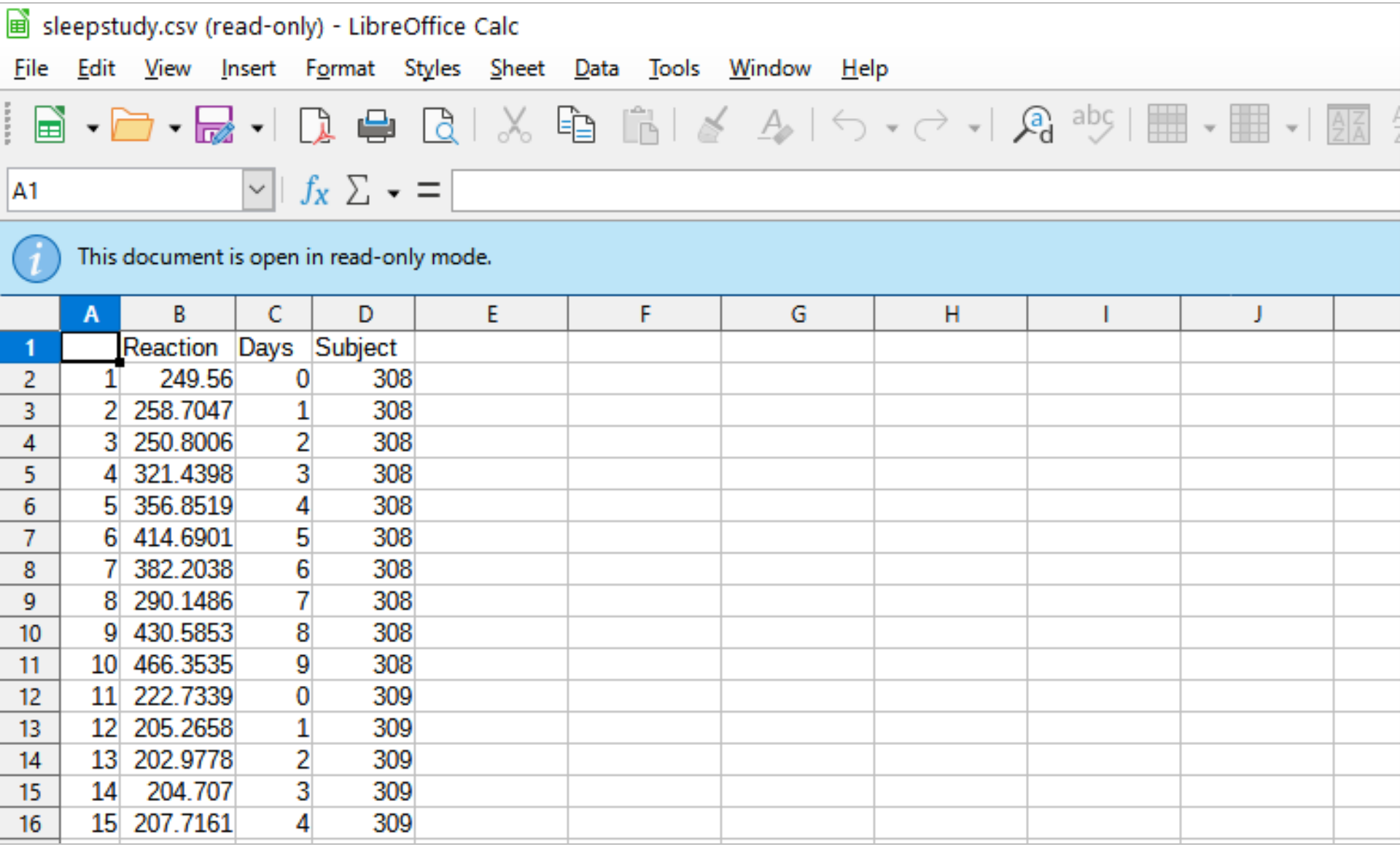
GitHub?

- GitHub je repizitorijum koji se nalazi na www, https://github.com/ (slika *Fair use*).
- Nudi opciju slobodnih (eng. *free*) repizitorijuma na kojima se nalazi *open-source* softverski projekti.
- U aprilu 2016. GitHub je imao 14 miliona korisnika i 35 miliona repizitorijuma što ga čini najvećim mestom za skladištenje kodova (izvor: https://en.wikipedia.org/wiki/GitHub), a u junu 2018. je imao 28 miliona korisnika i 57 miliona repozitorijuma (od kojih je 28 miliona javnih repozitorijuma).
- Savim je moguće da ćete koristiti ovu tehnologiju na poslu.
- Verzionisanje koda i kolaborativni rad su samo neke od prednosti.

Podaci

- Upotreba dplyr paketa će biti prikazana na primeru studije spavanja.
- Podaci ove studije pod nazivom “Reaction times in a sleep deprivation study” su dostupni na: https://vincentarelbundock.github.io/Rdatasets/datasets.html.
  - Opis dostupnih podataka iz ove studije se može naći na: https://vincentarelbundock.github.io/Rdatasets/doc/lme4/sleepstudy.html.
  - Rad u kome su opisani rezultati ove studije: Belenky, Gregory, et al. "Patterns of performance degradation and restoration during sleep restriction and subsequent recovery: A sleep dose-response study." *Journal of sleep research* 12.1 (2003): 1-12, [Online] http://onlinelibrary.wiley.com/doi/10.1046/j.1365-2869.2003.00337.x/full, Assessed March 14, 2019.

Podaci


sleepstudy.csv (read-only) - LibreOffice Calc

File Edit View Insert Format Styles Sheet Data Tools Window Help

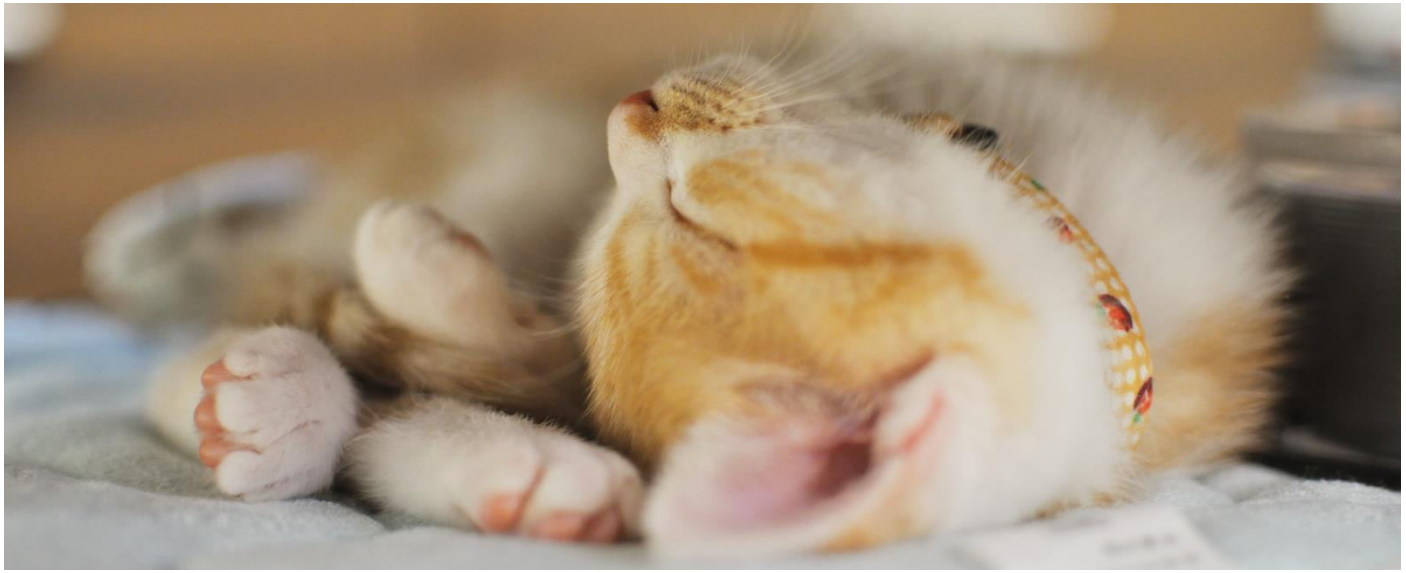
A1

This document is open in read-only mode.

	A	B	C	D	E	F	G	H	I	J
1		Reaction	Days	Subject						
2	1	249.56	0	308						
3	2	258.7047	1	308						
4	3	250.8006	2	308						
5	4	321.4398	3	308						
6	5	356.8519	4	308						
7	6	414.6901	5	308						
8	7	382.2038	6	308						
9	8	290.1486	7	308						
10	9	430.5853	8	308						
11	10	466.3535	9	308						
12	11	222.7339	0	309						
13	12	205.2658	1	309						
14	13	202.9778	2	309						
15	14	204.707	3	309						
16	15	207.7161	4	309						

Studija spavanja

- Ova studija je realizovana na 66 zdravih ispitanika.
- Ispitanici su podeljeni u grupe koje su 7 dana imale 3, 5, 7 i 9 sati sna. Nakon toga, svi ispitanici su tri dana spavali po 8 sati.
- Ocena uticaja sna ispitanike je bila PVT (eng. *Psychomotor Vigilance Task Performanse*). PVT meri vreme reakcije na vizuelni stimulus. Ovaj test je trajao 10 min i podaci koji su dostupni predstavljaju usrednjene vrednosti za jedno merenje za jednog ispitanika.
- Na raspolaganju za obradu je 180 merenja na 18 ispitanika.
- U studiji koja je publikovana, osim PVT mereni su i polisomnografski podaci (EEG, EMG, EKG i EOG).



PVT

- PVT je standardizovan test koji se koristi za merenje vremena reakcije subjekta na vizuelni stimulus (https://en.wikipedia.org/wiki/Psychomotor_vigilance_task).
- U studiji koja je ovde predstavljena, vizuelni stimulus je bio LED indikator. Može se relativno jednostavno projektovati i aplikacija na računaru, ali i na mobilnom telefonu za *self-assessment*.
- Vrlo često se koristi u svemirskim stanicama, jedan je od testova koje NASA (eng. *National Aueronautics and Space Administration*) redovno primenjuje.
- Primer testa: https://youtu.be/KT7A4e6hUf0, Psychomotor Vigilance Test (PVT)- Joggle Research for iPad by Joggle Research, Published on July 29, 2013, Assessed on March 14, 2019.

Podaci u studiji

```
> dat <- read.csv("sleepstudy.csv")
> dim(dat)
[1] 180   4
> str(dat)
'data.frame':   180 obs. of  4 variables:
 $ X       : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Reaction: num  250 259 251 321 357 ...
 $ Days    : int  0 1 2 3 4 5 6 7 8 9 ...
 $ Subject : int  308 308 308 308 308 308 308 308 308 308 ...
> names(dat)
[1] "X"        "Reaction" "Days"     "Subject"
> class(dat$X)
[1] "integer"
> class(dat$Reaction)
[1] "numeric"
> class(dat$Days)
[1] "integer"
> class(dat$Subject)
[1] "integer"
>
```

- Podaci su smešteni u fajl “sleepstudy.csv”.
- Sastoje se iz 4 kolone i 180 merenja. Prva kolona odgovara rednom broju merenja, druga kolona vremenu reakcije (u ms), treća kolona broju dana tokom kojih je ispitanicima bio smanjen broj sati provedenih u snu i u četvrtoj koloni se nalazi broj subjekta.
- NAPOMENA: Ovde su dostupni podaci o ispitanicima čiji je san trajao 0-tog dana 8 sati, a kasnije 3 sata. *Reaction* se odnosi na usrednjenu vrednost svih dnevnih testova. Ovaj deo studije sadrži ukupno 18 subjekata.

dplyr: funkcija *select()*

```
> ?select
> # ako želim da odaberem (u ovom slučaju da prikažem) samo
> # podatke od Days do Subject
> tail(select(dat, Days:Subject))
    Days Subject
175    4     372
176    5     372
177    6     372
178    7     372
179    8     372
180    9     372
> # ili bez ovih kolona
> tail(select(dat, -(Days:Subject)))
      X Reaction
175 175 287.1726
176 176 329.6076
177 177 334.4818
178 178 343.2199
179 179 369.1417
180 180 364.1236
> # ili dve bilo koje kolone
> tail(select(dat, Reaction, Subject))
    Reaction Subject
175 287.1726     372
176 329.6076     372
177 334.4818     372
178 343.2199     372
179 369.1417     372
180 364.1236     372
```

- Funkcija *select()* omogućava da se odabere određeni podskup podataka sa kojim želimo da radimo (na kome se primenjuje željena analiza podataka).
- Ova funkcija se može koristiti i za “izbacivanje” podskupa podataka.Postoji niz argumenata ove funkcije koje bi trebalo proučiti za jednostavno korišćenje, na primer *ends_with* ili *starts_with* i sl.

dplyr: funkcija *filter()*

```
> ?filter
> ?subset
> # može da se napravi podskup podataka
> # tako da je reactionTime > 350 ms
> datf <- filter(dat, Reaction > 350)
> head(datf)
   X Reaction Days Subject
1  5 356.8519    4     308
2  6 414.6901    5     308
3  7 382.2038    6     308
4  9 430.5853    8     308
5 10 466.3535    9     308
6 40 354.0487    9     330
> # ako imam dva uslova, pa na primer
> datf1 <- filter(dat, Reaction > 350
+                 & Reaction < 400)
> head(datf1)
   X Reaction Days Subject
1  5 356.8519    4     308
2  7 382.2038    6     308
3 40 354.0487    9     330
4 50 371.5811    9     331
5 70 362.0428    9     333
6 80 377.2990    9     334
```

- Na slici je prikazano kako se koristi *filter()* funkcija.
- Ova funkcija je slična *subset()* funkciji u osnovnom R paketu.
- Unutar ove funkcije mogu da postoje i logički uslovi (jedan ili više).

dplyr: funkcija *arrange()*

```
> ?arrange
> # ako se podaci rasporedjuju
> # prema broju dana
> # days of sleep deprivation
> datNew <- arrange(dat, Days)
> head(dat, 4L)
  X Reaction Days Subject
1 1 249.5600    0     308
2 2 258.7047    1     308
3 3 250.8006    2     308
4 4 321.4398    3     308
> head(datNew, 4)
   X Reaction Days Subject
1  1 249.5600    0     308
2 11 222.7339    0     309
3 21 199.0539    0     310
4 31 321.5426    0     330
> tail(datNew, 4)
      X Reaction Days Subject
177 150 366.5131    9     369
178 160 372.2288    9     370
179 170 369.4692    9     371
180 180 364.1236    9     372
> # u obrnutom redosledu
> # descending
> datNew1 <- arrange(dat, desc(Days))
> tail(datNew1, 3)
      X Reaction Days Subject
178 151 225.2640    0     370
179 161 269.8804    0     371
180 171 269.4117    0     372
```

- Na slici je prikazano kako se koristi *arrange()* funkcija.
- Mogu se pojedini podaci rasporediti po rastućem ili opadajućem redosledu.
- Ovo nije poželjno primenjivati na podacima koji su mereni kao npr. EMG signali, ali na podacima o pacijentima (da se preurede po polu, težini, visini, određenim kliničkim parametrima ima smisla primeniti ovu funkciju).

dplyr: funkcija *rename()*

```
> ?rename
> # X je samo redni broj merenja
> # nema smisla da ostane pod tim
> # imenom
> datNew2 <- rename(dat, Num = X)
> head(dat, 4)
  X Reaction Days Subject
1 1 249.5600    0     308
2 2 258.7047    1     308
3 3 250.8006    2     308
4 4 321.4398    3     308
> head(datNew2, 4)
  Num Reaction Days Subject
1   1 249.5600    0     308
2   2 258.7047    1     308
3   3 250.8006    2     308
4   4 321.4398    3     308
```

- Na slici je prikazano kako se koristi *rename()* funkcija.
- Kao i sa ostalim dplyr funkcijama, ovo se relativno jednostavno radi ako za to postoji posebna funkcija.
- Poželjno je da promenljive imaju što je moguće logičnija i opisna imena, kako bi se omogućilo jednostavno šerovanje tj. deljenje podataka i koda, ali i manipulacija sa podacima i dalja analiza.

dplyr: funkcija *mutate()*

```
> ?mutate
> # kako normalizovati podatke u odnosu na srednju vrednost
> # da li je to ovde potrebno?
> datNew3 <- mutate(dat, Reaction = Reaction - mean(Reaction, na.rm = T))
> head(dat, 3)
  X Reaction Days Subject
1 1 249.5600    0     308
2 2 258.7047    1     308
3 3 250.8006    2     308
> head(datNew3, 3)
  X  Reaction Days Subject
1 1 -48.94789    0     308
2 2 -39.80319    1     308
3 3 -47.70729    2     308
>
> # Reaction je u ms, ali može biti i u sekundama
> datNew4 <- mutate(dat, Reaction = Reaction / 1000)
> head(dat, 3)
  X Reaction Days Subject
1 1 249.5600    0     308
2 2 258.7047    1     308
3 3 250.8006    2     308
> head(datNew4, 3)
  X  Reaction Days Subject
1 1 0.2495600    0     308
2 2 0.2587047    1     308
3 3 0.2508006    2     308
> # može i
> head(select(datNew4, Reaction), 3)
   Reaction
1 0.2495600
2 0.2587047
3 0.2508006
```

- Na slici je prikazano kako se koristi *mutate()* funkcija.
- Oduzimanje srednje vrednost od nekog merenja (centriranje promenljive)
- Šta znači argument *na.rm*?
- Postoji i *transmute()* funkcija koja omogućava da se iz niza promenljivih izostave netransformisane.

dplyr: funkcija *group_by()*

```
> # definisanje nove promenljive Reaction
> sum(is.na(dat$Reaction))
[1] 0
> mean(dat$Reaction)
[1] 298.5079
> ?group_by
> datNew5 <- mutate(dat,
+                   Reaction = factor(1*(Reaction > mean(Reaction, na.rm = T))
+                   labels = c("short", "long")))
> head(dat, 4)
  X Reaction Days Subject
1 1 249.5600    0     308
2 2 258.7047    1     308
3 3 250.8006    2     308
4 4 321.4398    3     308
> head(datNew5, 4)
  X Reaction Days Subject
1 1    short    0     308
2 2    short    1     308
3 3    short    2     308
4 4     long    3     308
> # grupisanje prema novoj promenljivoj
> datGroup <- group_by(datNew5, Reaction)
> head(datGroup)
Source: local data frame [6 x 4]
Groups: Reaction [2]

      X Reaction  Days Subject
  <int>   <fctr> <int>   <int>
1     1    short     0     308
2     2    short     1     308
3     3    short     2     308
4     4     long     3     308
5     5     long     4     308
6     6     long     5     308
```

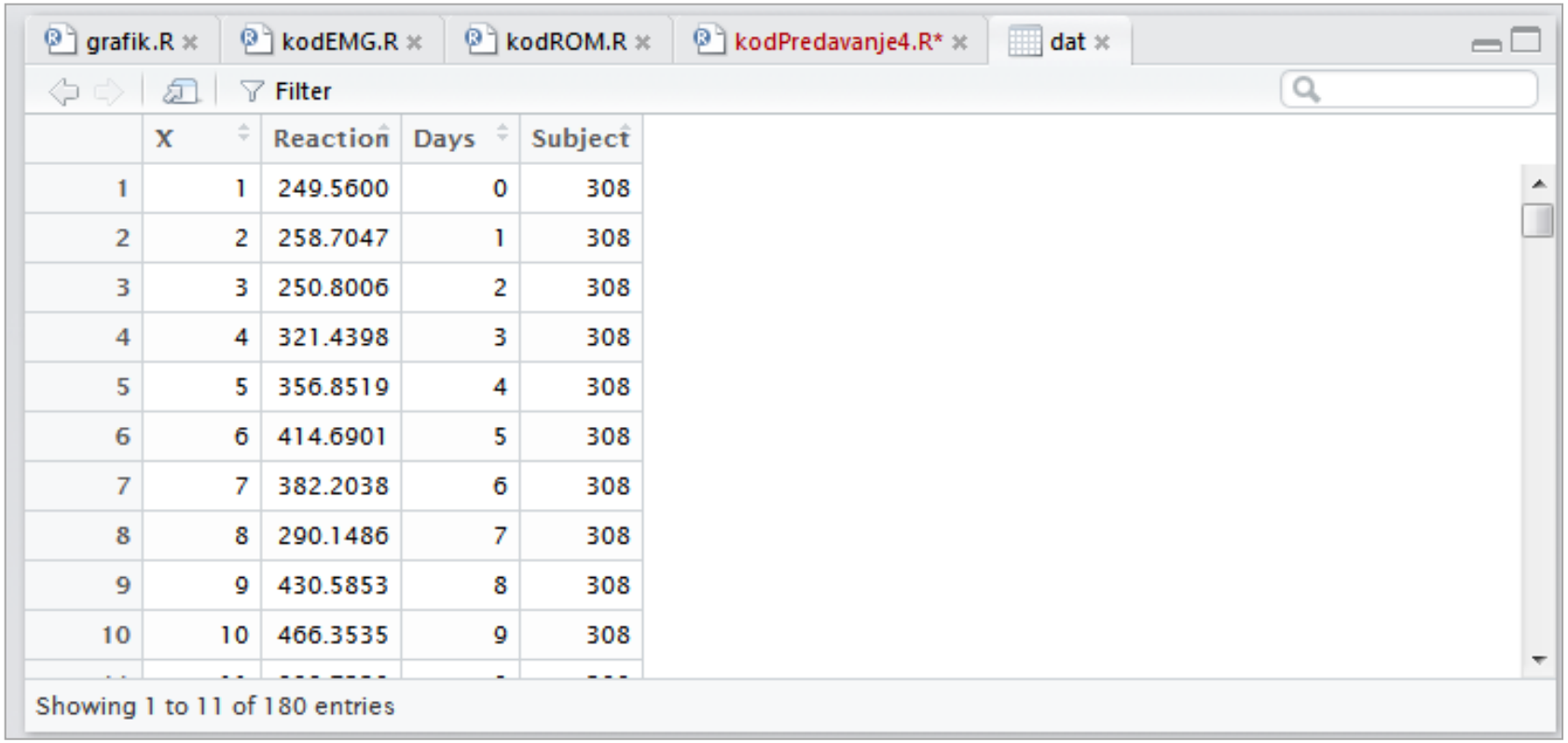
- Na slici je prikazano kako se koristi *group_by()* funkcija.
- Ako postoji potreba da se određena analiza primeni na podskupu podataka, na primer *summarise()* da se primeni na podskupu, onda je zgodno napraviti najpre odgovarajući podskup sa *group_by()* funkcijom koju treba kombinovati sa nekom drugom funkcijom koja se bavi analizom signala.

dplyr: funkcija *summarise()*

```
> ?summarise
> summarise(datGroup, Days = mean(Days))
# A tibble: 2 x 2
  Reaction     Days
    <fctr>    <dbl>
1    short 3.277228
2     long 6.063291
> summarise(datGroup, Days = median(Days))
# A tibble: 2 x 2
  Reaction  Days
    <fctr> <int>
1    short     3
2     long     7
>
```

- Na slici je prikazano kako se koristi *summarize()* funkcija.
- Primetiti da se ova funkcija koristi u kombinaciji sa izlazom iz *group_by()* funkcije.
- Šta znače podaci koji su dobijeni primenom ove funkcije na studiju spavanja?

Funkcija *View()*



	X	Reaction	Days	Subject
1	1	249.5600	0	308
2	2	258.7047	1	308
3	3	250.8006	2	308
4	4	321.4398	3	308
5	5	356.8519	4	308
6	6	414.6901	5	308
7	7	382.2038	6	308
8	8	290.1486	7	308
9	9	430.5853	8	308
10	10	466.3535	9	308

Showing 1 to 11 of 180 entries

- Omogućava da se podaci pregledaju tabelarno.
- Primer kako izgleda rezultat *View(dat)* komande je prikazan na slici.
- Funkcija je primenjena na podacima iz studije spavanja. Ova funkcija ne priprada dplyr paketu, ali se često koristi u kombinaciji sa drugim funkcijama iz dplyr paketa.

PIPELINE

Pipeline

- Pipeline operator %>% omogućava da se veći broj funkcija realizuje unutar jedne sekvence
- Na primer, ako postoji poziv ugnježdenih funkcija kao što je:
  - **third(second(first(x)))**
- onda se te funkcije mogu pozvati na sledeći način:
  - **first(x) %>% second %>% third**
- Pogodno je koristiti Pipeline, iz sledećih razloga:
  - ne koriste se nikakve pomoćne promenljive,
  - zapis je elegantniji (mogu se izostaviti pojedini argumenti) i
  - zapis je razumljiviji.

Uvoz i manipulacija podacima

- Najjednostavnije je koristiti dplyr paket (slika paketa u gornjem desnom uglu je preuzeta sa https://rstudioblog.files.wordpress.com/2014/10/datacamp-dplyr.png?w=350&h=200&crop=1, pristupljeno 14. marta 2019. godine).
- U računarskoj tehnici pipeline se koristi kao termin za skup procesirajućih elemenata koji su povezani redno (tako da je izlaz prvog elementa ulaz drugog itd.), izvor https://en.wikipedia.org/wiki/Pipeline_(software).

Pipeline primer

```
> dat %>% mutate(Reaction = factor(1*(Reaction > mean(Reaction)),
+                                  labels = c("short", "long"))) %>%
+          group_by(Reaction) %>% summarise(Days = median(Days))
# A tibble: 2 x 2
  Reaction  Days
    <fctr> <int>
1    short     3
2     long     7
> 
```

- Na osnovu prethodnih podataka koji su dobijeni u studiji spavanja i na osnovu prethodnih operacija, realizovan je Pipeline kao na slici.
- Rezultat Pipeline-a prikazanog na slici može se i dodeliti promenljivoj po želji.
- Primetiti da postoji manji broj ulaznih parametara nego u primerima koji su pokazani ranije.
- Šta je rezultat ovog Pipeline-a?

Šta je *tibble*?

- To je tip podataka sličan *data frame*-u, samo sa nešto drugačijim osobinama.
- Često se kaže da je *tibble* moderan *data frame*.
- Više na: https://cran.r-project.org/web/packages/tibble/vignettes/tibble.html i na http://www.sthda.com/english/wiki/tibble-data-format-in-r-best-and-modern-way-to-work-with-your-data (pristupljeno 14. marta 2019. godine).
- Takođe, razlike između ovih tipova podataka mogu se pogledati na CRAN-u: posebno se odnose na podskup i recikliranje, https://cran.r-project.org/web/packages/tibble/vignettes/tibble.html
- Ovde se ne bavimo posebno raznim skupovima podataka.
- Zgodno je da znate da postoje.

Pipeline primer 2

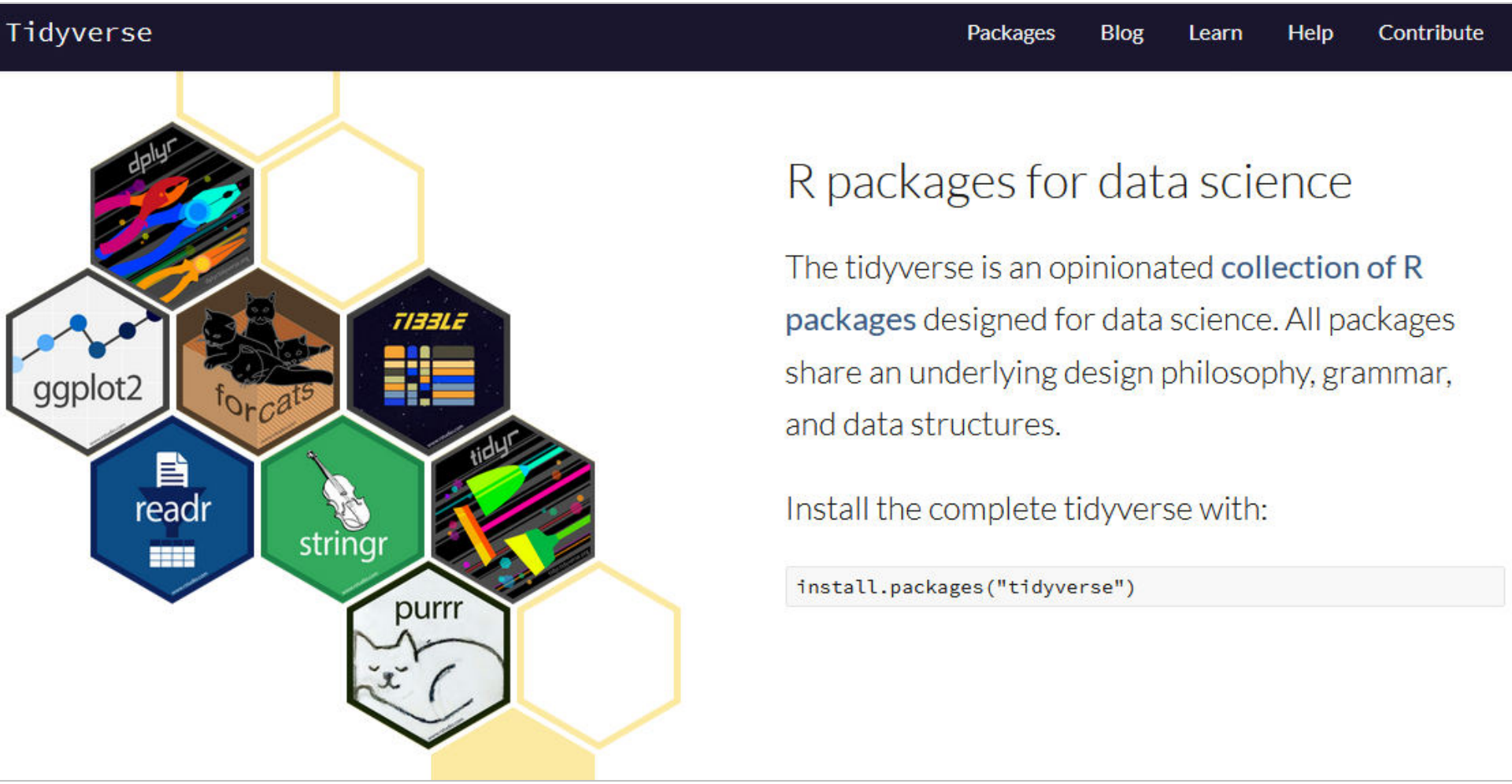
```
> # Pipeline primer 2
> dat %>% mutate(Days = factor(1*(Days > 5),
+                              labels = c("short", "long"))) %>%
+         group_by(Days) %>% summarise(Reaction = mean(Reaction))
# A tibble: 2 x 2
    Days Reaction
  <fctr>    <dbl>
1  short 277.7782
2   long 329.6024
> 
```

- Šta je rezultat ovog Pipeline-a?
- Šta to znači za PVT i za subjekte koji su učestvovali u ovoj studiji?
- Da li se brzina kojom mozak reaguje promenila sa brojem dana koje je subjekat proveo sa smanjenim snom?

dplyr dodatno

- Uz dplyr gramatiku, moguće je:
  - postići dodatnu funkcionalnost koja ovde nije prikazana (više na https://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html, pristupljeno u martu 2017. godine).
  - moguće je koristiti data.table paket u kombinaciji sa dplyr.
- Pored ovog paketa, postoje i drugi paketi koje vredi istražiti.
- SQL je programski jezik (od eng. *Structures Query Language*) koji se koristi za rad sa bazama podataka (više na: https://en.wikipedia.org/wiki/SQL). U dplyr paketu postoji funkcija *translate_sql()* za prevod R koda u SQL kod.

dplyr je deo tidyverse skupa paketa



https://www.tidyverse.org/, *Fair Use*.

REZIME

Rezime

- dplyr paket omogućava relativno jednostavnu manipulaciju podacima na relativno efikasan i brz način.
- Ta efikasnost se može poboljšati pomoću Pipeline operatora %>%.
- Studije spavanja koja je u ovoj lekciji korišćena kao primer biomedicinskih podataka je realizovana sa ciljem ocene promena u psihomotornom sistemu usled neredovnog spavanja kod zdravih ispitanika. Rezultati ove studije su pokazali da se mozak adaptira na uslove smanjenog sna. Takođe, period oporavka je duži što je vreme sna u kraće.