

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

Putovanje vozom?



- Posao mašinovođe i san.
- Video: Fast Reaction and Coordination Test for Train Drivers by CareerVidz <https://youtu.be/q1LcEoYvd2Q>, June 19, 2013 (pristupljeno 20. marta 2019. godine).
- Slika: Riding the Rails (On the Alaska Railroad South of Anchorage) od CMY23; Flickr <https://www.flickr.com/photos/62741243@N04/37486002560/>; CC BY-NC 2.0

Data frame

```
> head(dat)
  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
> class(dat)
[1] "data.frame"
> dat$Subject <- NULL
> head(dat)
  X Reaction Days
1 1 249.5600    0
2 2 258.7047    1
3 3 250.8006    2
4 4 321.4398    3
5 5 356.8519    4
6 6 414.6901    5
> |
```

- Primeri brisanja kolona za podatke učitane iz studije spavanja su prikazani na slici.
- Više na:
<https://stackoverflow.com/questions/6286313/remove-an-entire-column-from-a-data-frame-in-r/6286336>
(pristupljeno 15. marta 2019. godine).

R update

```
Console ~/
> install.packages("installr")
Installing package into 'C:/Users/Nadica Miljkovic/Documents/R/win-library/3.4'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/installr_0.19.0.zip'
Content type 'application/zip' length 260735 bytes (254 KB)
downloaded 254 KB

package 'installr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\Nadica Miljkovic\AppData\Local\Temp\Rtmpwne5Yg\downloaded_packages
> library(installr)
Loading required package: stringr

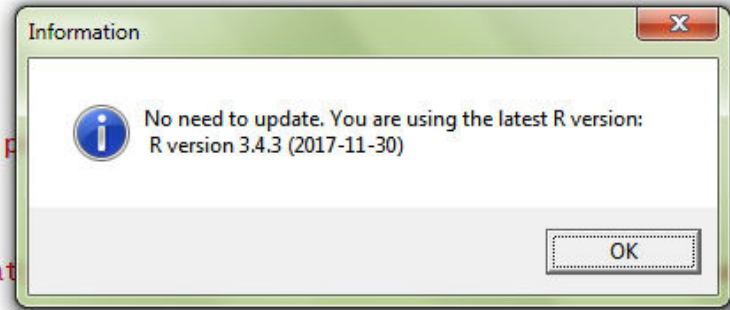
Welcome to installr version 0.19.0

More information is available on the installr page:
https://github.com/talgalili/installr/

Contact: <tal.galili@gmail.com>
Suggestions and bug-reports can be submitted at:
  https://github.com/talgalili/installr/issues

To suppress this message use:
  suppressPackageStartupMessages(library(installr))

> updateR()
[1] FALSE
> updateR()
```



- Instalirajte paket “installr” sa CRAN-a.
- Potom pokrenuti funkciju `updateR()`. Može, ali nekada ne rešava problem.

Greška kao na slici (učitavanje ggplot2 biblioteke)

```
> library(ggplot2)
Error: package or namespace load failed for 'ggplot2' in get(Info[i, 1], envir = env):
  lazy-load database 'C:/Users/Nadica Miljkovic/Documents/R/win-library/3.4/rlang/R/rlang.rdb' is corrupt
In addition: Warning message:
In get(Info[i, 1], envir = env) : internal error -3 in R_decompress1
> install.packages("ggplot2", dependencies=TRUE)
Installing package into 'C:/Users/Nadica Miljkovic/Documents/R/win-library/3.4'
(as 'lib' is unspecified)
also installing the dependencies 'checkmate', 'rstudioapi', 'zoo', 'backports', 'rex', 'Formula', 'latticeExtra', 'acepack', 'htmlTable', 'mvtnorm', 'TH.data', 'sandwich', 'SparseM', 'Matrix Models', 'rprojroot', 'gdtools', 'covr', 'ggplot2movies', 'Hmisc', 'maptools', 'multcomp', 'quantreg', 'rmarkdown', 'svglite'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/checkmate_1.8.5.zip'
Content type 'application/zip' length 628587 bytes (613 KB)
downloaded 613 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/rstudioapi_0.7.zip'
Content type 'application/zip' length 129886 bytes (126 KB)
downloaded 126 KB
```

- Najčešće nastane problem kada nisu instalirani drugi paketi na koje se željeni paket oslanja.
- Potrebno je ponovo instalirati paket koji nije mogao da se učita sa opcijom `dependencies = TRUE`.
- Više na: <https://github.com/swirldev/swirl/issues/371> (pristupljeno 15.03.2019.).

I konačano



- Iako korisno, prethodno iskustvo mi baš i nije bilo od pomoći.
- “My Documents” folder na računaru sam preimenovala u “Documents”, jer Path koji je R tražio je bio pogrešan.
- Ni to nije pomoglo u potpunosti (neki paketi su bili aktivni, a neki nisu).
- Znači, skoro pa konačno...
- Najverovatnije se mora podesiti i u <https://github.com/STAT545-UBC/Discussion/issues/250> (pristupljeno 15. marta 2019. godine).
- Obrisala sam sve Associated Folders i ponovo sve instalirala i onda je radilo!
- Razlog je što sam menjala broj korisnika na računaru. Sve radosti Windows OS-a i na drugim predavanjima...



Modifikovana slika *Visit to hell* by Mexican artist Mauricio García Vega - Painting and photograph of Mauricio García Vega, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=24243238>

Pisanje funkcija

- **Zašto je korisno pisati funkcije?** U slučajevima kada postoji:
 - kod koji se ponavlja,
 - deljenje koda (podela na manje celine),
 - sa ciljem organizacije koda i
 - zbog interfejsa (apstrakcija postojećeg koda).
- Funkcije u R-u su objekti prve klase (klasa funkcija) i mogu biti:
 - prosleđene drugim funkcijama i
 - ugnježdene.
- Probati *class(mean)*, *class(sum)* i sl.

Primeri funkcija

```
# prazna funkcija
f1 <- function() {

}

# funkcija bez argumenata
f2 <- function() {
  print("Ovo je primer funkcije.")
}

# funkcija sa jednim argumentom
f3 <- function(ar) {
  print("Ovo je primer funkcije sa argumentom.")
  print(ar)
}

# funkcija sa podrazumevanim argumentom
f4 <- function(ar = 5) {
  print("Ovo je primer funkcije sa podrazumevanim argumentom.")
  print(ar)
}
```

- Funkcija ima posebnu klasu. Šta će za rezultat dati poziv prazne funkcije *f1()* u komandnom prozoru? NULL vrednost.
- Može se odabrati vrednost koju treba vratiti odnosno izlaznu vrednost funkcije, onda je moguće koristiti komandu *return()* unutar funkcije.
- Šta će se desiti, ako se u konzoli otkuca *f3()*? Odštampaće se tekst i javiće se greška da nedostaje argument funkcije.
- Formalni argumenti funkcije su argumenti koji su uključeni u definiciju funkcije. Te argumente moguće je izlistati korišćenjem funkcije *formals()*.

Rezultati poziva funkcija

```
> # prazna funkcija
> f1 <- function() {
+
+ }
> class(f1)
[1] "function"
> f1()
NULL
>
> # funkcija bez argumenata
> f2 <- function() {
+     print("ovo je primer funkcije.")
+ }
> f2()
[1] "ovo je primer funkcije."
> f2(5)
Error in f2(5) : unused argument (5)
>
> # funkcija sa jednim argumentom
> f3 <- function(ar) {
+     print("ovo je primer funkcije sa argumentom.")
+     print(ar)
+ }
> f3()
[1] "ovo je primer funkcije sa argumentom."
Error in print(ar) : argument "ar" is missing, with no default
>
> # funkcija sa podrazumevanim argumentom
> f4 <- function(ar = 5) {
+     print("ovo je primer funkcije sa podrazumevanim argumentom.")
+     return(ar*4)
+ }
> f4()
[1] "ovo je primer funkcije sa podrazumevanim argumentom."
[1] 20
> f4(3)
[1] "ovo je primer funkcije sa podrazumevanim argumentom."
[1] 12
```

- Na slici je prikazana konzola R-a za pozive prethodnih funkcija.
- Takođe, prikazan je način korišćenja komande *return()*.
- Da li je za funkciju *f3()* važno koji tip podataka je unet za *ar* promenljivu ili će raditi sa bilo kojim tipom podataka? Sa bilo kojim tipom, jer se koristi *print()* funkcija.

Slaganje argumenata

- Slaganje tj. podudaranje argumenata (eng. *Argument Matching*) je rešeno u R-u na dva načina: prema **poziciji** i prema **imenu**.
- Ako funkcija ima više argumenata, onda bi trebalo prilikom poziva funkcije navesti ime argumenta.
- U R-u argumenti mogu biti poziciono ili po imenu podudarani sa definisanim ulaznim parametrima (argumentima funkcije).

Slaganje argumenata – primer

```
> str(rnorm)
function (n, mean = 0, sd = 1)
> args(rnorm)
function (n, mean = 0, sd = 1)
NULL
> |
```

- Ugrađena funkcije *rnorm()* se koristi za generisanje slučajnih brojeva prema Gausovoj/normalnoj raspodeli.
- Argumenti ove funkcije su *rnorm(n, mean = 0, sd = 1)* – otkucati *?rnorm* u R konzoli za detaljne informacije. Koji od argumenata nije podrazumevani?
- Postoji i podudaranje argumenata gde se koriste oba tipa podudaranja i po poziciji i po imenu. R radi sledeće (redom):
 - proverava da li postoji potpuno podudaranje za neki od argumenata po imenu,
 - proverava da li postoji delimično podudaranje za neki od argumenata po imenu i
 - proverava da li postoji poziciono podudaranje
- Za listanje svih argumenata funkcije može se koristiti funkcija *args()*, a *str()* za strukturu objekta.
- **Preporuka je da se koristi podudaranje argumanta po imenima, a ne po poziciji posebno kada funkcija ima relativno veliki broj argumenata. Takođe, treba izbegavati kombinaciju ove dve metode podudaranja.**
- Nekada je zgodno i da neki od argumenata bude postavljen na NULL vrednost, jer u tom slučaju se može promeniti uloga funkcije.

set.seed() funkcija

```
> set.seed(1)
> rnorm(4)
[1] -0.6264538  0.1836433 -0.8356286  1.5952808
>
> set.seed(10)
> rnorm(4)
[1]  0.01874617 -0.18425254 -1.37133055 -0.59916772
>
> set.seed(1)
> rnorm(4)
[1] -0.6264538  0.1836433 -0.8356286  1.5952808
>
> rnorm(4)
[1]  0.3295078 -0.8204684  0.4874291  0.7383247
> rnorm(4)
[1]  0.5757814 -0.3053884  1.5117812  0.3898432
> rnorm(4)
[1] -0.62124058 -2.21469989  1.12493092 -0.04493361
> rnorm(4)
[1] -0.01619026  0.94383621  0.82122120  0.59390132
> |
```

- Prilikom generisanja slučajnih brojeva rezultat koda (ili izgled grafika za zavisnost SD od broja merenja) se menja.
- Kako bi se obezbedili kontrolisani uslovi za simulaciju slučajnih brojeva na računaru, koristi se *set.seed()* funkcija.
- Na slici su dati primeri upotrebe ove funkcije.

Slaganje argumenata – primer

```
> br <- 4
> set.seed(1)
> brojevi <- rnorm(br, mean = 0, sd = 1)
> brojevi
[1] -0.6264538  0.1836433 -0.8356286  1.5952808
> brojevi <- rnorm(br)
> brojevi
[1]  0.3295078 -0.8204684  0.4874291  0.7383247
> set.seed(1)
> brojevi <- rnorm(br)
> brojevi
[1] -0.6264538  0.1836433 -0.8356286  1.5952808
> set.seed(1)
> brojevi <- rnorm(br, sd = 1, mean = 0)
> brojevi
[1] -0.6264538  0.1836433 -0.8356286  1.5952808
> set.seed(1)
> brojevi <- rnorm(br, s = 1, me = 0)
> brojevi
[1] -0.6264538  0.1836433 -0.8356286  1.5952808
> set.seed(1)
> brojevi <- rnorm(s = 1, me = 0, br)
> brojevi
[1] -0.6264538  0.1836433 -0.8356286  1.5952808
> set.seed(1)
> brojevi <- rnorm(br, 0, 1)
> brojevi
[1] -0.6264538  0.1836433 -0.8356286  1.5952808
>
```

- Na slici je prikazana upotreba *rnorm()* funkcije.
- Obratiti pažnju da bez upotrebe *set.seed()* funkcije, slaganje argumenata ne bi moglo da se prikaže primenom *rnorm()* funkcije.
- Gde se u ovom primeru javlja slaganje po imenu, a gde po poziciji?

Pozivi funkcija ~ lenjost

```
> lenjPrimer <- function(ar1 = 5) {  
+   print("Ovo je primer za lazy evaluation.")  
+   print(ar1)  
+   print(ar2)  
+   return(ar1*4 + 6)  
+ }  
> lenjPrimer()  
[1] "Ovo je primer za lazy evaluation."  
[1] 5  
Error in print(ar2) : object 'ar2' not found  
> |
```

- Ako u pozivu funkcije postoji neka greška (na primer nije definisan ili nije prosleđen odgovarajući argument), R funkcija će se izvršiti sve do trenutka kada do greške dođe.
- Ova osobina R-a se naziva *lazy evaluation*.
- Primer je dat na slici.

Šta ako je argument “...” ?

```
> noviRnorm <- function(sd = 2, ...) {  
+   rnorm(sd = 2, ...)  
+ }  
>  
> x <- noviRnorm(200, mean = 2, sd = 2)  
> sd(x)  
[1] 2.173227  
> length(x)  
[1] 200  
> |
```

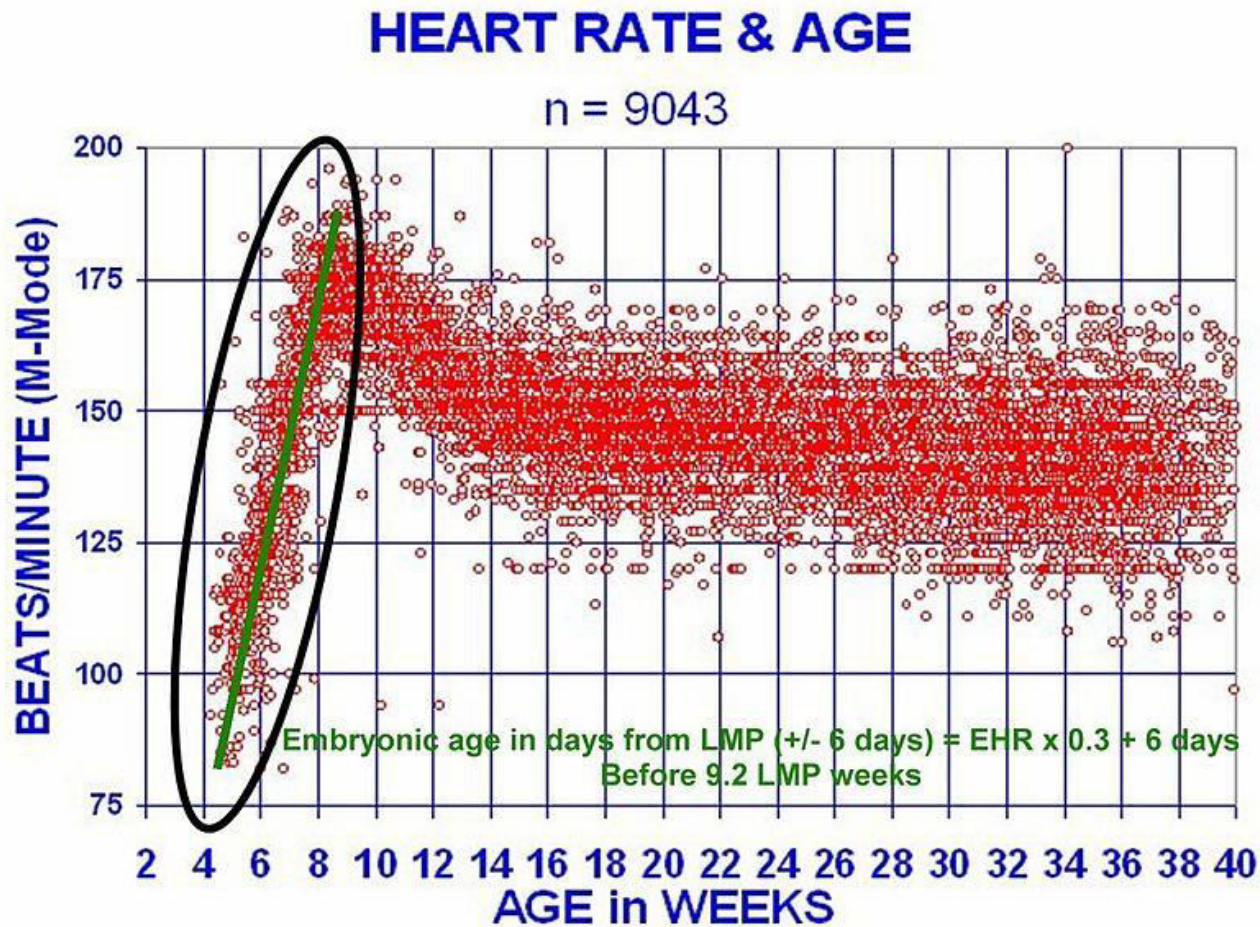
- Kada se koristi: ako se ugrađena funkcija i njeni argumenti prosleđuju novoj funkciji, onda se koristi argument “...” jer tako su svi argumenti ugrađene funkcije dostupni i u novoj.
- Korisno je menjati/preimenovati ugrađene funkcije posebno kada se sa novom funkcijom postiže različita funkcionalnost.
- Neke funkcije (kao što su *paste()* i *cut()*) imaju “...” argument, jer se ne zna unapred koliki će biti broj ulaznih argumenata.
- **NAPOMENA:** svi argumenti koji dolaze posle “...” ne mogu se pozivati sa delimičnim, već sa potpunim podudaranjem po imenu.

Srčani puls

SRČANI RITAM U MIROVANJU [bpm]					
novorođene bebe (0 - 3 meseci starosti)	bebe (3 - 6 meseci starosti)	bebe (6 - 12 meseci starosti)	deca (1 - 10 godina starosti)	odrasli i deca > 10 godina starosti	sportisti
100 - 150	90 - 120	80 - 120	70 - 130	60 - 100	40 - 60

- Srčani puls (eng. *heartbeat*) odgovara sekvenci mehaničkih i električnih događaja koji se ponavljaju, https://en.wikipedia.org/wiki/Cardiac_cycle.
- Frekvencija srčanog pulsa (srčani ritam) se najčešće izražava u bpm jedinicama (eng. *beats per minute*) i ona je određena brojem srčanih kontrakcija u minuti. Ova vrednost varira tokom fizičkih aktivnosti, sna, stresa, bolesti i može se menjati kao posledica upotrebe lekova.
- U tabeli su izražene “normalne” vrednosti pulsa za starosne grupe i sportiste. Podaci iz tabele su peuzeti sa sajta: <https://en.wikipedia.org/wiki/Pulse>.
- Svaka neregularnost srčanog ritma se naziva aritmija: **bradikardija** je usporen srčani ritam (< 60 bpm u mirovanju), a **tahikardija** je ubrzan (> 100 bpm u mirovanju). Videti detalje na: https://en.wikipedia.org/wiki/Heart_rate.

Srčani ritam ~ nedelje starosti embriona



By Original uploader was DuBose at en.wikipedia - Transferred from en.wikipedia, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=12467228>

Studija spavanja i srčani ritam

- Polisomnografija (eng. *polysomnography*) je metoda koja se koristi u dijagnostici sna. Koristi se multi-parametarska metoda koja omogućava ocenu različitih faza sna. Ova merenja uključuju, pored ostalih, i merenje pulsa.
- U toku sna, najčešće se meri EKG sa 2 - 3 elektrode (klasična klinička metoda uključuje 10 elektroda). Pored srčanog ritma određuju se i drugi parametri.
- Očekivano je da će tokom spavanja srčani ritam biti sporiji, uobičajeno ~ 40 – 50 bpm (non REM (eng. *Rapid Eye Movement*) faza koja čini ~ 80 % sna).
- U toku REM faze srčani ritam može biti i veći. **Zašto?**

Sleep apnea

- *Sleep apnea* (https://en.wikipedia.org/wiki/Sleep_apnea) je poremećaj koji uzrokuje prestanak disanja (od par s do par min) tokom sna.
- Ovakav prestanak disanja može biti opasan po život.
- Tokom epizoda u kojima pacijentkinja/pacijent prestaje da diše dolazi do povećanja srčanog ritma.
- Pokazano je da je povećanje srčanog ritma u korelaciji sa AHI (eng. *Apnea-Hypopnea Index*). AHI je indeks koji se koristi u oceni *sleep apnee* i predstavlja odnos broja pojave *sleep apnee* tokom sna i broja sati provedenih u snu (https://en.wikipedia.org/wiki/Apnea%20%93hypopnea_index).
- Literatura:
 - Andreas, S., et al. "Changes in heart rate during obstructive sleep apnoea." *European Respiratory Journal* 5.7 (1992): 853-857.

Drugi faktori koji utiču na srčani ritam



- Kofein i nikotin su stimulansi nervnog sistema i izazivaju povećan srčani ritam.
- Osim fizičke aktivnosti koja ubrzava srčani ritam, temperatura tela može da utiče na srčani ritam:
 - hipertermija (povećana telesna temperatura) može dovesti do povećanja srčanog ritma, a
 - hipotermija (smanjena telesna temperatura) usporava srčani ritam, ali i smanjuje amplitudu srčanih kontrakcija.
- BMI (eng. *body mass index*, https://en.wikipedia.org/wiki/Body_mass_index) se izražava u jedinicama kg/m^2 i pokazano je da se srčani ritam povećava sa povećanjem BMI. Pogledati:
 - Martins, D., et al. "The relationship between body mass index, blood pressure and pulse rate among normotensive and hypertensive participants in the third National Health and Nutrition Examination Survey (NHANES)." *Cellular and molecular biology (Noisy-le-Grand, France)* 49.8 (2003): 1305-1309.
 - Mohesh, Glad, and Ajith Prasath. "Correlation of body mass index and cardiovascular reactivity to exercise-induced stress in young adults." *International Journal of Medical Science and Public Health* 4.5 (2015): 717-720. Na slici je prikazan grafik iz ovog rada.

Oporavak srčanog ritma

- HR_{recovery} (eng. *heart rate recovery*, https://en.wikipedia.org/wiki/Heart_rate#Heart_rate_recovery) predstavlja smanjenje srčanog ritma tokom fizičke aktivnosti i srčanog ritma koji je izmeren tokom određenog perioda nakon fizičke aktivnosti (vreme oporavka).
- Što je veća razlika između srčanog ritma u trenutku fizičke aktivnosti i nakon fizičke aktivnosti to je ispitanik u boljoj formi.
- Patološkim nalazom se smatra srčani ritam koji se ne promeni za više od 12 bpm nakon 1 min. Postoje standardno usvojene vrednosti (tabele) za različite ispitanike i rizične grupe. Jedan deo komercijalnih uređaja uključuje ova računanja.

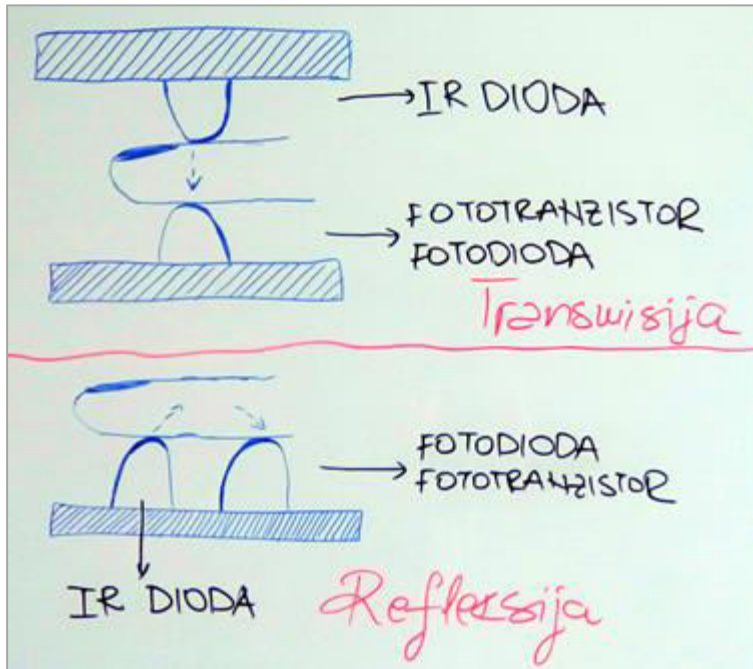


Merenje srčanog ritma

- Manuelna metoda.
- Elektrofiziološka merenja.
- IR (eng. *infrared*) merenja.
- Merenja primenom optičkih vlakana.
- Video kamerom.
- Druge ideje?
- Kako se meri puls primenom mobilnog telefona?

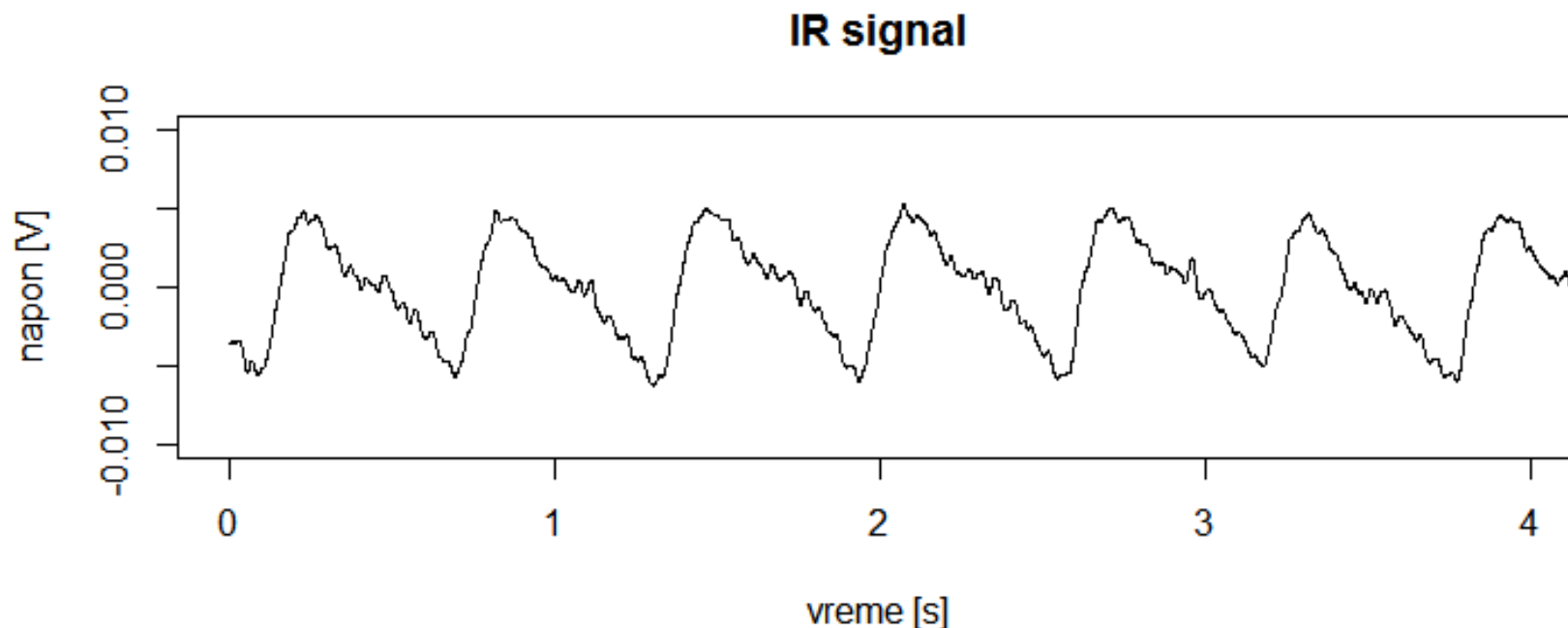


IR merenje



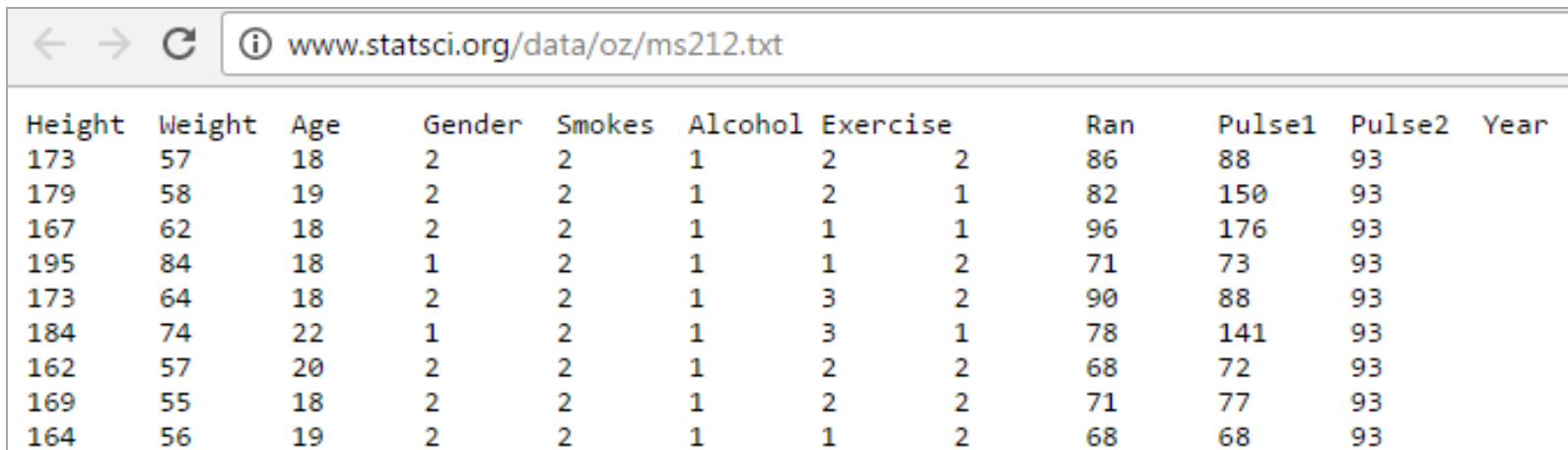
- Postoje dva tipa IR metoda za merenje pulsa. Ovi uređaji se najčešće postavljaju na prst ili čelo ispitanika.
- Na slici su prikazani transmisioni i reflektivni tip (slika je fotografisana u laboratoriji 69 na Elektrotehničkom fakultetu u Beogradu, februar 2017.).
- Ova metoda se često naziva fotopletizmografija (<https://en.wikipedia.org/wiki/Photoplethysmogram>).
- Koje su prednosti, ako je su mane jednog i drugog merenja?

Refleksivno IR merenje



- Na slici je prikazan signal koji je dobijen primenom refleksivne IR metode na kažiprstu leve ruke.
- Signal je meren u laboratoriji 69 na Elektrotehničkom fakultetu u Beogradu, februar 2017.
- Koliki je srčani ritam na signalu sa slike? Otprilike, naravno.

Studija merenja pulsa



The image shows a screenshot of a web browser displaying a text file from the URL www.statsci.org/data/oz/ms212.txt. The browser's address bar shows navigation icons and the URL. Below the address bar, a table of data is displayed with the following columns: Height, Weight, Age, Gender, Smokes, Alcohol, Exercise, Ran, Pulse1, Pulse2, and Year. The data consists of 10 rows of numerical values.

Height	Weight	Age	Gender	Smokes	Alcohol	Exercise	Ran	Pulse1	Pulse2	Year
173	57	18	2	2	1	2	2	86	88	93
179	58	19	2	2	1	2	1	82	150	93
167	62	18	2	2	1	1	1	96	176	93
195	84	18	1	2	1	1	2	71	73	93
173	64	18	2	2	1	3	2	90	88	93
184	74	22	1	2	1	3	1	78	141	93
162	57	20	2	2	1	2	2	68	72	93
169	55	18	2	2	1	2	2	71	77	93
164	56	19	2	2	1	1	2	68	68	93

- Na vežbama će biti korišćeni podaci “Pulse Dataset Information” od Dr Richard J. Wilson, Department of Mathematics, University of Queensland koji je dostupan na sajtu: <http://www.statsci.org/data/oz/ms212.html>.
- Podaci se nalaze u tekstualnoj datoteci. Izgled datoteke sa osnovnim podacima dat je na slici.

Protokol merenja pulsa

- U fajlu su dati osnovni podaci o ispitanicima (visina, težina, godine života, pol, podaci o konzumaciji cigareta i alkohola i fizička aktivnost ispitanika).
- Zadatak koji su studentkinje i studenti imali je bio da manuelno izmere svoj puls u stanju mirovanja (sedeći položaj). Ti podaci se nalaze u koloni "Pulse1".
- Potom, "slučajno" odabrane studentkinje i studenti su imali zadatak da trče u roku od 1 min, a ostale studentkinje i studenti da nastave da sede. Nakon toga, izmeren je srčani ritam svima (manuelnom metodom) i podaci su smešteni u kolonu "Pulse2".
 - Svi studenti su merili "Pulse2".
- U promenljivoj/koloni "Ran" su smeštene informacije o tome koje studentkinje u studenti su "trčali u mestu", a koji su ostali da sede pre drugog računanja pulsa.

Studija merenja pulsa

Variable	Description	Summary Statistics				
HEIGHT	Height (cm)	Mean = 173.3 cm		Std Dev = 10.17		
WEIGHT	Weight (kg)	Mean = 66.8 kg		Std Dev = 14.88		
AGE	Age (years)	Mean = 20.6 yrs		Std Dev = 3.97		
GENDER	Sex (1 = Male, 2 =Female)	Males: 54.2%		Females: 45.8%		
SMOKES	Regular smoker? (1 = Yes, 2 = No)	Yes: 10.3%		No: 89.7%		
ALCOHOL	Regular drinker? (1 = Yes, 2 = No)	Yes: 62.6%		No: 37.4%		
EXERCISE	Frequency of exercise (1 = High, 2 = Moderate, 3 = Low)	High: 13.1%	Mod: 53.3%	Low: 33.6%		
TRT	Whether the student ran or sat (1 = Ran, 2 = Sat)	Ran: 42.1%		Sat: 57.9%		
PULSE1	First pulse measurement (rate per minute)	Mean = 75.2		Std Dev = 12.68		
PULSE2	Second pulse measurement (rate per minute)	Mean = 96.2		Std Dev = 31.55		
YEAR	Year of class (93 - 98) – values are counts ⇨	93: 26	95: 22	96: 21	97: 22	98: 16

- Dodatno, podaci o ovoj studiji i signalu se mogu naći na linkovima:
 - <http://media.news.health.ufl.edu/misc/bolt/Intro/SPSS/PulseDataInformation.pdf>
 - <http://www.statsci.org/data/oz/ms212.html>
 - <http://www.statsci.org/data/oz/ms212.txt>
- Na slici je predstavljena tabela sa preglednom statistikom tj. *Summary Statistics* podacima za ovu studiju.

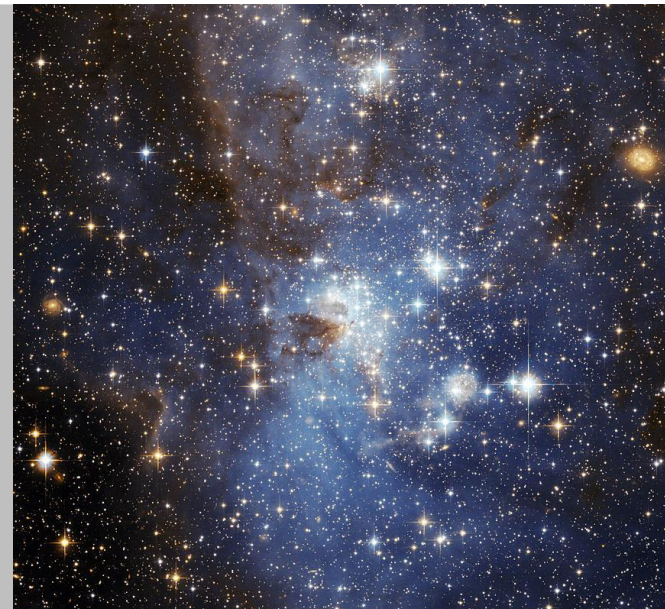
Dodeljivanje vrednosti

```
> mean <- function(x) {x*x}
> mean
function(x) {x*x}
> search()
[1] ".GlobalEnv"          "tools:rstudio"      "package:stats"      "package:graphics"
[5] "package:grDevices"    "package:utils"      "package:datasets"   "package:methods"
[9] "Autoloads"           "package:base"
>
```

- Ako se definiše nova funkcija koja ima isto ime kao postojeća funkcija kako R zna o čemu je reč?
- R radi sledeće:
 - Prvo se proverava globalno okruženje u kome je potrebno tražiti objekat pod tim imenom (pogledati redosled pretrage nakon poziva *search()* funkcije).
 - Potom se proveravaju svi paketi (po abecedi).
 - Na kraju se proverava *base* paket. I u dobru i u zlu.
- U R-u se objekat i funkcija mogu isto zvati, jer je rezervisan različit prostor za oba. U globalnom okruženju može postojati samo jedan naziv.
- Redosled po kom se paketi pretražuju je moguće menjati ... redosledom aktivacije paketa u R-u.
- Nadam se da Vam ovo neće biti potrebno.

Pravila pretrage argumenata

- R koristi leksička pravila pretrage ili statička pravila pretrage (eng. *Lexical / Static Scoping*).
- Više informacija na Wikipediji:
[https://en.wikipedia.org/wiki/Scope_\(computer_science\)#Lexical_scope_vs._dynamic_scope](https://en.wikipedia.org/wiki/Scope_(computer_science)#Lexical_scope_vs._dynamic_scope).





Kada se primenjuju pravila pretrage?

```
> f <- function(x, y) {  
+   x + y + z  
+ }  
>
```

```
> f(2, 3)  
Error in f(2, 3) : object 'z' not found  
> z <- 1  
> f(2, 3)  
[1] 6  
>
```

- Funkcija sa slike ima dva formalna argumenta x i y, ali i slobodnu promenljivu z.
- Leksička pravila pretrage argumenata određuju koja će vrednost biti dodeljena promenljivoj z.
- U R-u ona će biti pretražena u okruženju u kome je funkcija definisana.
- Pretraga se obavlja u R-u na sledeći način:
 - Ako vrednost promenljive nije pronađena u okruženju u kome je definisana funkcija, onda se ona traži u *parent* okruženju.
 - Pretraga se nastavlja do *top-level* okruženja (obično globalno okruženje).
 - Pretraga dalje nastavlja do *down-level* okruženja dok ne stigne do praznog (*empty*) okruženja.
- Ako se ne nađe promenljiva, onda dolazi do greške.
- **SAVET:** Izbegavati definisanje funkcija na ovaj način. Za sada.

Leksička pravila pretrage

```
> sd
function (x, na.rm = FALSE)
sqrt(var(if (is.vector(x) || is.factor(x)) x else as.double(x),
  na.rm = na.rm))
<bytecode: 0x0000000014279b98>
<environment: namespace:stats>
> select
Error: object 'select' not found
> 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

warning message:
package 'dplyr' was built under R version 3.3.2
> select
function (.data, ...)
{
  select_(.data, .dots = lazyeval::lazy_dots(...))
}
<environment: namespace:dplyr>
> |
```

- Da razjasnimo:
 - Okruženje (eng. *environment*) je kolekcija simbola i njihovih vrednosti. Svako okruženje ima svoje *parent* okruženje i može imati više okruženja kojima je to okruženje *parent*. Jedino okruženje bez *parent* okruženja je prazno (eng. *empty environment*).
- Uobičajeno, funkcije i ostali R objekti/promenljive se definišu u globalnom okruženju i mogu se izlistati u radnom prostoru (eng. *global environment*).
- Kako je u R-u moguće definisati funkciju unutar funkcije ili kao izlaz jedne može se dobiti druga funkcija, onda je okruženje ugnježdene funkcije druga funkcija, a ne globalno i sl. ...
- Pomoću funkcija *ls()* i *get()* moguće je proveriti koji su objekti definisani u kom okruženju.
- Kucanjem samo imena funkcije u konzoli dobija se informacija o okruženju u kome je funkcija definisana.

Npr. u C-u ne može da se definiše funkcija unutar funkcije, a u R-u može!

Koliko je *funkcija1*(10)?

```
> x <- 2
>
> funkcija1 <- function(arg1) {
+     x <- 5
+     x * funkcija2(arg1)
+ }
>
> funkcija2 <- function(arg1) {
+     arg1 * x
+ }
```

Leksička vs. dinamička pravila pretrage

```
> x <- 2
>
> funkcija1 <- function(arg1) {
+     x <- 5
+     x * funkcija2(arg1)
+ }
>
> funkcija2 <- function(arg1) {
+     arg1 * x
+ }
>
> funkcija1(10)
[1] 100
> |
```

- Na slici je dat primer koda i rešenje.
- Leksička pretraga: vrednost x se traži u *parent* okruženju u kome je funkcija definisana, pa je tražen rezultat $funkcija1(10) = 100$.
- Za dinamičku pretragu, to je *calling environment* i vrednost bi bila drugačija pa je $funkcija1(10) = 250$.

Leksičko pretraživanje

- Uobičajeno je i u drugim programskim jezicima kao što su: Scheme, Perl, Python i Common Lisp da se koristi leksičko pretraživanje.
- Kao posledica leksičkog okruženja, svi objekti moraju biti smešteni u memoriji računara.
- Sve funkcije sa sobom imaju i pokazivače ka okruženju u kome su definisane. **Kako je moguće doći do tih pokazivača?**
 - Tako što se ukuca ime funkcije u konzoli.
- Najčešće se ovo pravilo primenjuje kod funkcija koje vrše optimizaciju i one za argumente imaju druge funkcije (kojima traže minimum ili maksimum).

Ukratko ...

```
> fun <- function(x) {  
+     y <- 10  
+     return(z * x^y)  
+ }  
> |
```

- Svaka funkcija ima tri dela:
 - formalne parametre (argumenti koje korisnik definiše prilikom poziva funkcije),
 - telo funkcije (sekvenca komandi i izraza koje bi trebalo pokrenuti prilikom poziva funkcije) i
 - okruženje (u kome “živi” funkcija).
- Kada su funkcije u pitanju, postoje 3 vrste promenljivih:
 - formalne promenljive,
 - lokalne promenljive i
 - slobodne promenljive (na njih se primenjuju pravila leksičkog pretraživanja).
- U primeru sa slike, koje promenljive su formalne, lokalne, a koje slobodne?
- Više informacija o leksičkom pretraživanju i primerima na: <https://cran.r-project.org/doc/misc/lexical.tex>.

Stil programiranja

- Iako je individualan, mora da zadovolji određene standarde.
- Neke smernice su:
 - Uvek pisati kod u tekstualnom editoru, nikada u konzoli (Ctrl/Enter, RStudio)
 - Reorganizacija blokova koda (razmaci obavezno) – preglednost koda je povećana (automatski? Tools/Global Options/Code/Editing ...)
 - Ograničiti “širinu” koda (~80 kolona)
 - Dati opisne nazive promenljivim (*frekvencija_odabiranja*, *frekvencijaOdabiranja*, *fOd*, *fs*, *fod*, ...)
 - Pisati komentare (#)
 - Ne preterivati (“Manje je više.”)
 - Ograničiti “dužinu” funkcija (veličina prozora)
 - Logički naziv i uloga funkcija
- Pogledati:
 - https://en.wikipedia.org/wiki/Programming_style
 - https://en.wikipedia.org/wiki/Coding_conventions
 - [https://en.wikipedia.org/wiki/Naming_convention_\(programming\)](https://en.wikipedia.org/wiki/Naming_convention_(programming))

- Install Packages...
- Check for Package Updates...
- Version Control
- Shell...
- Addins
- Keyboard Shortcuts Help Alt+Shift+K
- Modify Keyboard Shortcuts...
- Project Options...
- Global Options...

- General
- Code
- Appearance
- Pane Layout
- Packages
- R Markdown
- Sweave
- Spelling
- Git/SVN
- Publishing

General

- Insert spaces for tab
Tab width
- Insert matching parens/quotes
- Auto-indent code after paste
- Vertically align arguments in auto-indent
- Soft-wrap R source files
- Continue comment when inserting new line
- Surround selection on text insertion:
- Keybindings:

Execution

- Always save R scripts before sourcing
- Focus console after executing from source
- Execute all lines in a statement

Snippets

- Enable code snippets

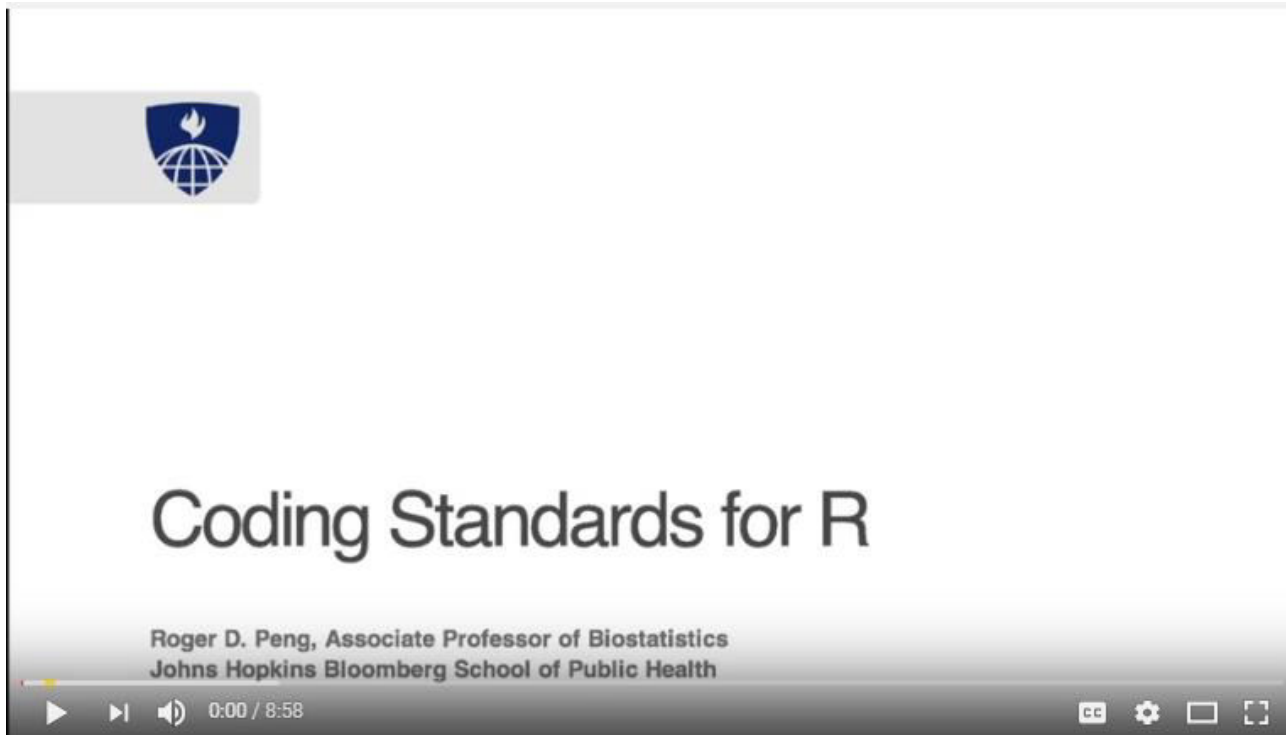
```

+ scale_colour_gradient(low = "blue")
+ scale_shape(solid = FALSE)
geom_point(colour = "red", size = 3)
price))

```

contributors.
tion and
ackages in publications.
for on-line help, or
erface to help.

Saveti



- od nekoga ko o standardima za pisanje koda zna puno.
- Pogledati Coding Standards for R, Rodger D. Peng, Professor in Biostatistics, John Hopkins Bloomberg School of Public Health
- Video se nalazi na linku:
<https://www.youtube.com/watch?v=MSPKE1y3cyQ&feature=youtu.be>.

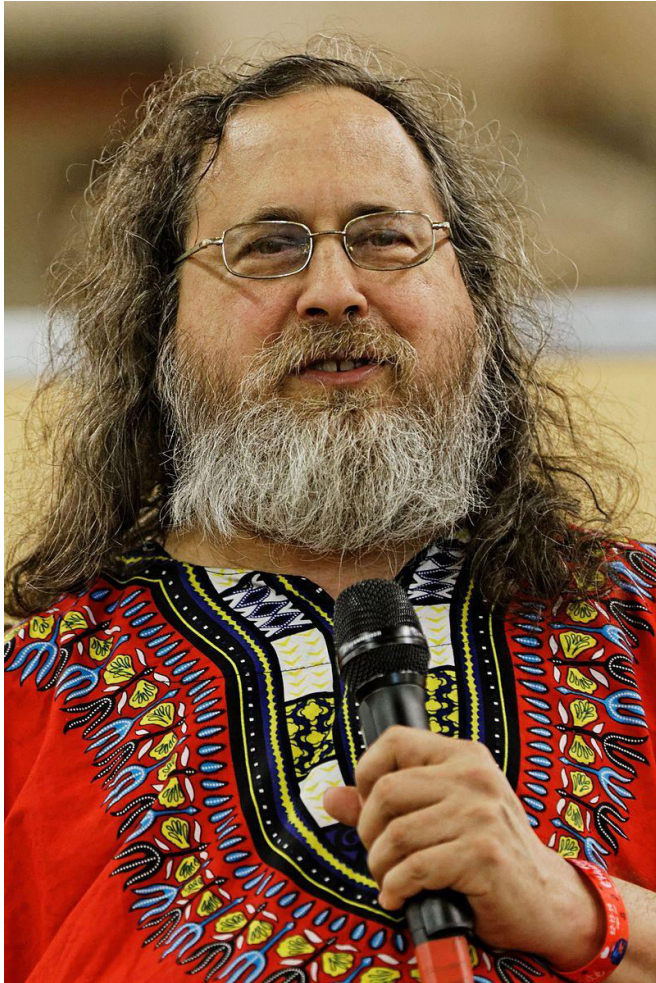


Rezime

Take Stay home messages

- Dplyr paket je koristan i to je prikazano na primeru studije spavanja.
- Spavati je važno!
- Funkcije se mogu pisati korišćenjem *function* naredbe i one predstavljaju vrstu R objekata.
- Funkcije mogu biti pozivane sa argumentima (postavljeni, izostavljeni, podrazumevani, podudarani i preuzeti). Postoji poseban “...” argument.
- Funkcije kao rezultat uvek daju rezultat poslednjeg izraza koji postoji u telu funkcije ili prikazuju sve što se nalazi unutar svih funkcija *print()*. Može se koristiti i naredba *return()*.
- Objektivne funkcije koje učestvuju u definisanju druge funkcije mogu se koristiti: značajno pojednostavljuje kod (potrebno je znanje iz oblasti *Lexical Scopings*).
- Pregledan i jednostavan kod je odlika profesionalca koji u svojoj praksi koristi programski jezik bez obzira na oblast kojom se bavi.
- Postoji niz metoda za merenje pulsa, a najčešće korišćene su primenom elektrofizioloških pojačavača i IR metoda.

Dodatno, o *free software*-u



- od osnivača GNU projekta Ričarda Stalmana (https://en.wikipedia.org/wiki/Richard_Stallman).
- TEDxGeneva 2014: Free software, free society: Richard Stallman, https://www.youtube.com/watch?v=Ag1AKII_2GM.
- Postoji i knjiga za one koje to zanima:
 - Free as in Freedom (2.0): Richard Stallman and the Free Software Revolution Sam Williams Second edition revisions by Richard M. Stallman, <https://sagitter.fedorapeople.org/faif-2.0.pdf>.