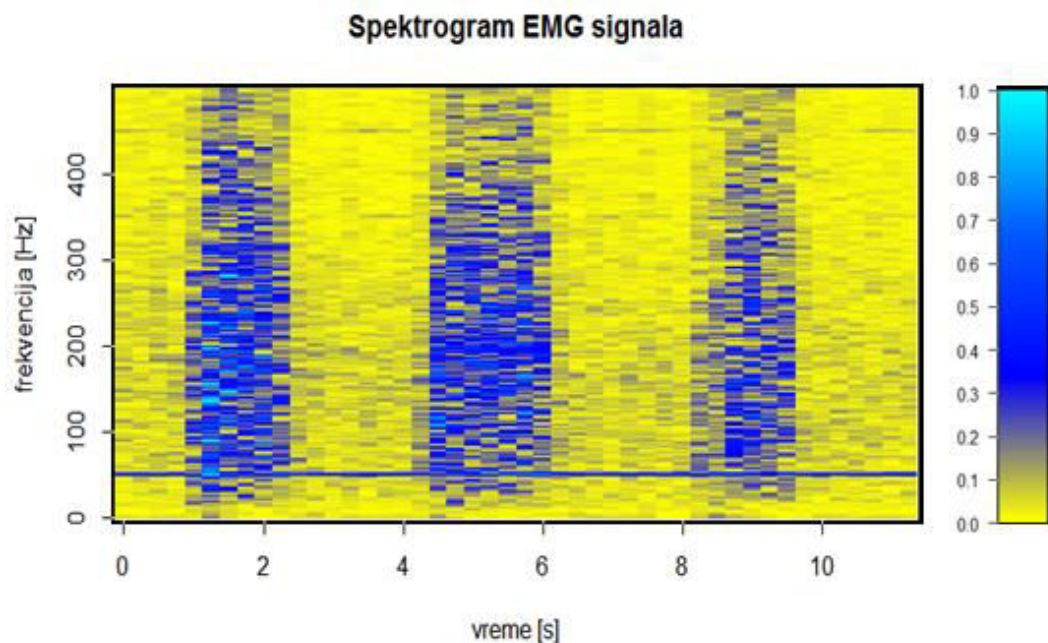
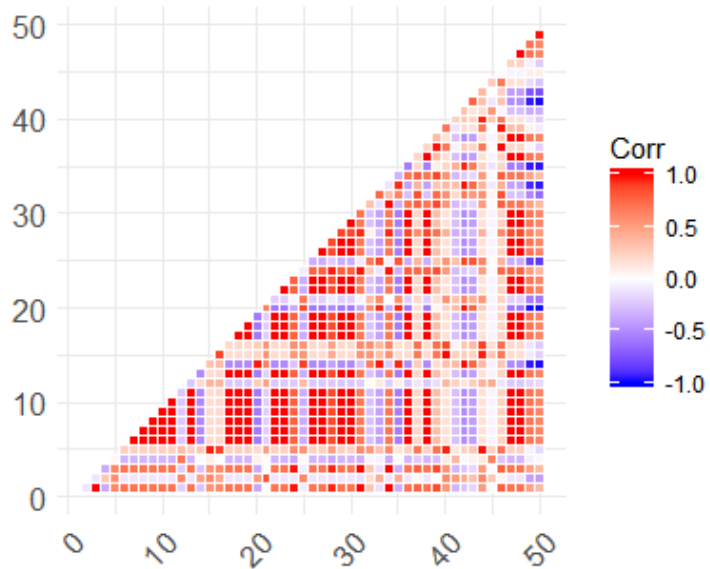


Tehnike obrade biomedicinskih signala 19M051TOBS



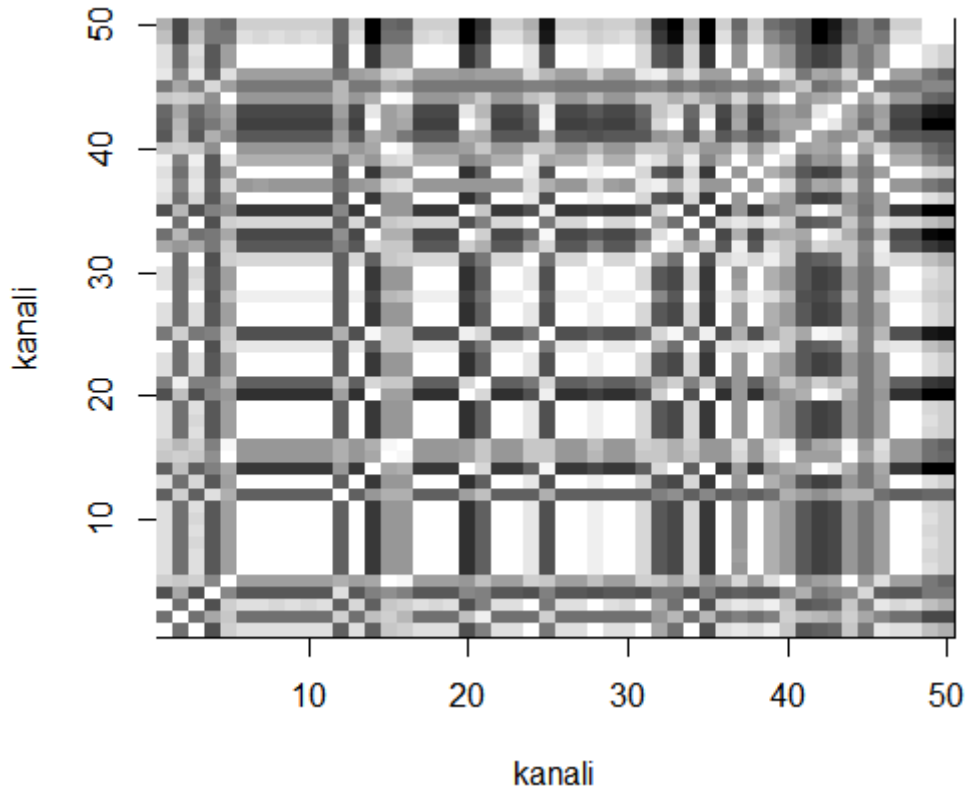
Dr Nadica Miljković, vanredni profesor
kabinet 68, nadica.miljkovic@etf.bg.ac.rs

2D prikaz (od prošle nedelje)

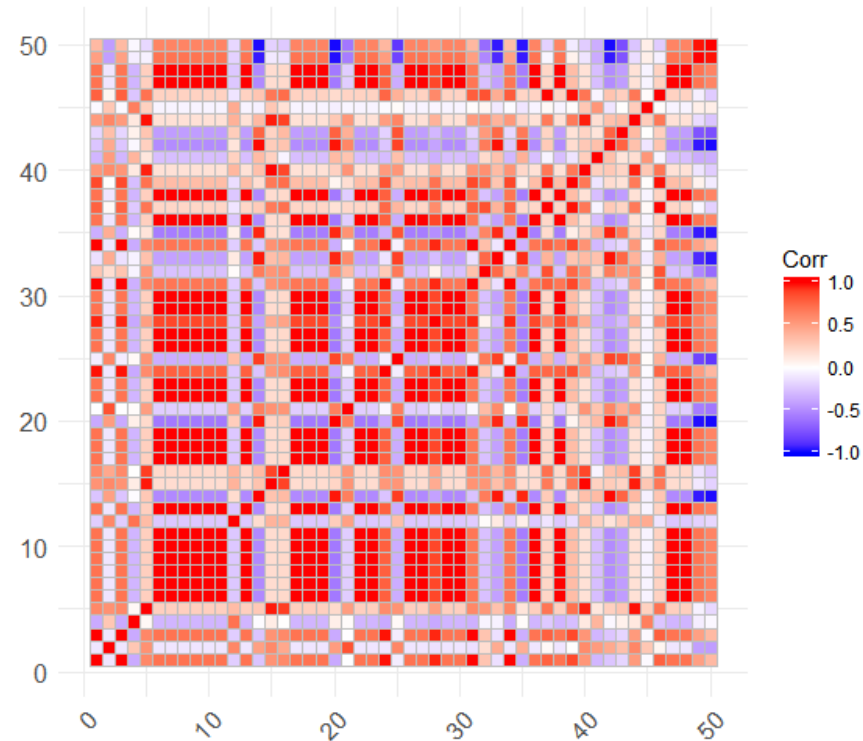
- Za meren relativno veliki broj podataka, potrebno je proveriti kakva je kroskorelacija između pojedinačnih kanala.
 - Vrlo koristan predkorak.
- To je važan predkorak u redukciji dimenzionalnosti podataka i najčešće predstavlja osnovnu informaciju o merenim signalima. Može da definiše korake koji će biti korišćeni u daljoj analizi.
- Neka je meren video signal i neka je na licu ispitanika izdvojeno 50 markera koji su snimljeni u fajlu “50.txt”.
 - Koji biosignal je moguće izdvojiti na ovaj način?
- Za tih 50 podataka izračunat je kroskorelacioni koeficijent.
- Potrebna je tehnika kojom će se prikazati 250 korelacionih koeficijenata. Tabela bi u ovom prikazu bila nepregledna.

Vizuelizacija

Kros-korelacija



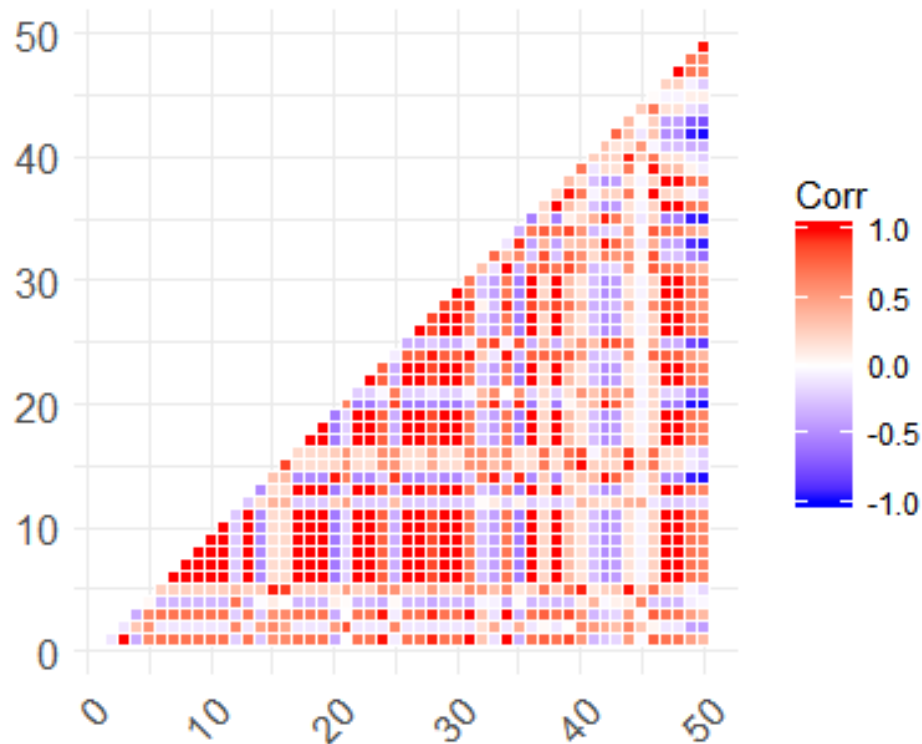
```
image(corM, x = 1:50, y = 1:50,  
      xlab = "kanali", ylab = "kanali",  
      main = "Kros-korelacija", col = gray((0:32)/32))
```



```
ddat <- read.table("50.txt")  
dim(ddat)  
  
corM <- cor(ddat)  
rownames(corM) <- factor(c(1:50))  
colnames(corM) <- factor(c(1:50))  
  
library(ggcorrplot)  
ggcorrplot(corM)
```

- Postoji paket koji se oslanja na ggplot2 i koji se zove ggcorrplot.
- Na slici su prikazane kros-korelacione matrice za 50 kanala koji su učitani iz fajla "50.txt" primenom osnovnih i ggcorrplot paketa, respektivno.

Kros-korelacione matrice su simetrične!



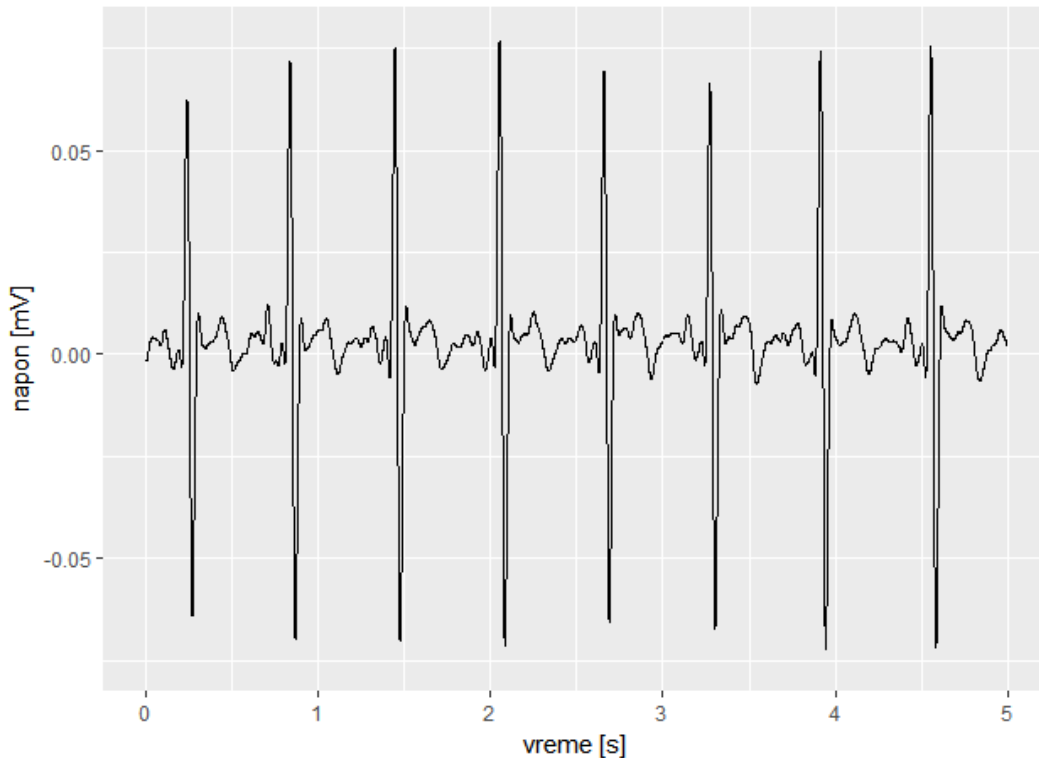
```
ggcorrplot(corrM, type = "lower",  
            outline.col = "white")
```

Više o primerima primene ggcorrplot paketa može se naći na:

<http://www.sthda.com/english/wiki/ggcorrplot-visualization-of-a-correlation-matrix-using-ggplot2>.

Prikaz vremenskih serija signala

Snimljen EKG signal



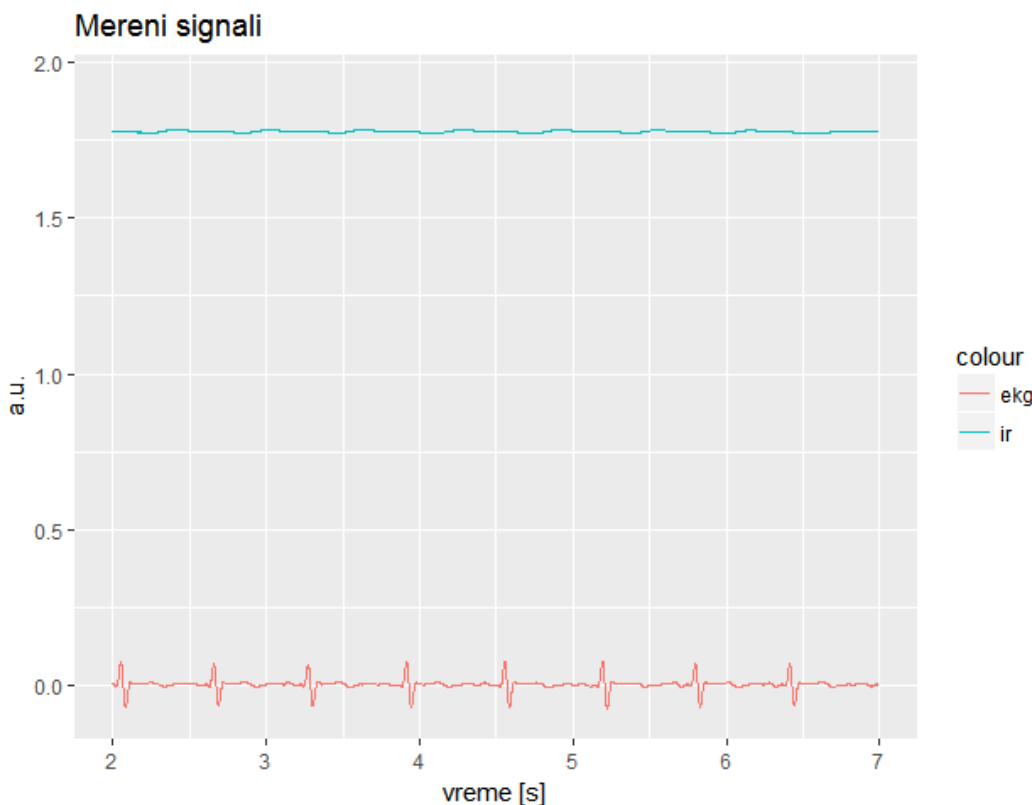
```
dat <- read.table("puls.txt")
fs <- 1000

library(signal)
pod <- list("numeric")
f1 <- butter(3, 100/fs/2, 'low')
pod$ekgf <- filtfilt(f1, dat$V1[1:10000] / 5)
pod$irf <- filtfilt(f1, dat$V2[1:10000])
pod$vreme <- seq(0, (length(pod$irf)/fs - 1/fs), by = 1 / fs)
pod <- data.frame(pod)

library(ggplot2)
ggplot(pod, aes(x = vreme, y = ekgf)) +
  geom_line() +
  ggtitle("Snimljen EKG signal") +
  ylab("napon [mV]") +
  xlab("vreme [s]") +
  xlim(0, 5)
```

- Korišćeni su signali iz fajla “puls.txt” (iz vežbe 6) u kome se nalazi puls ispitanika sniman sa EKG-om i sa IR senzorom.
- Signali su prikazani primenom ggplot2 funkcija.

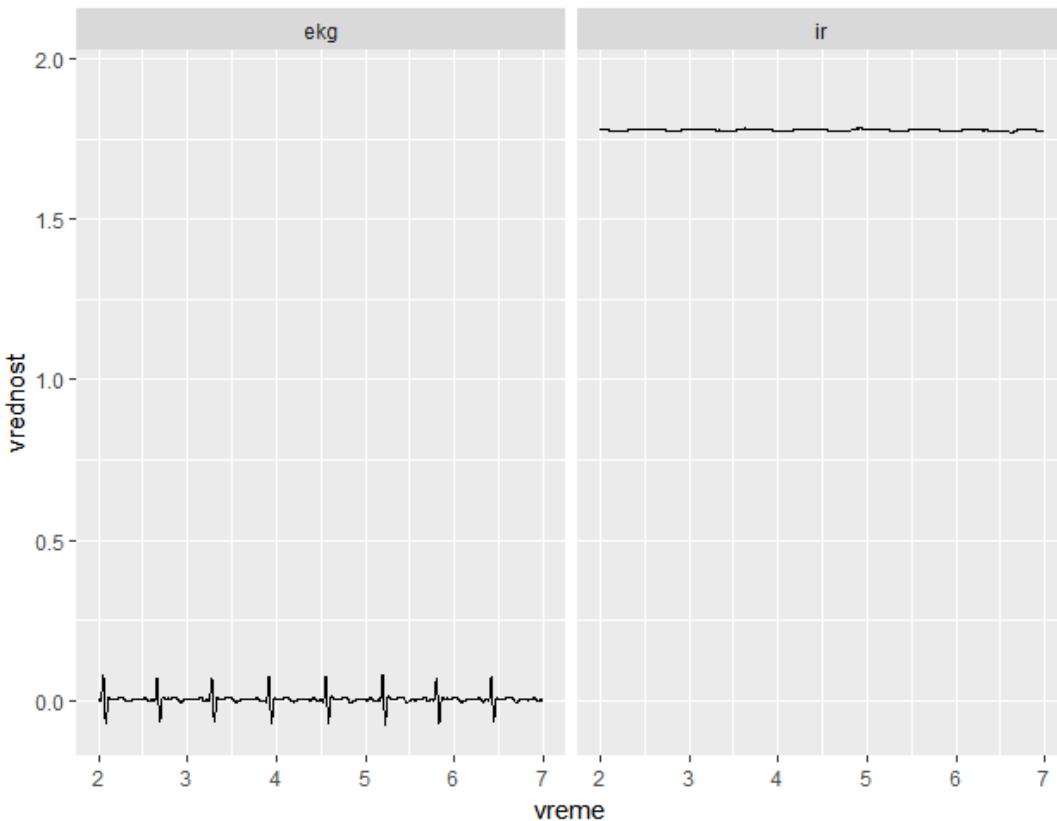
Više vremenskih serija odjednom?



```
ggplot(pod, aes(vreme)) +  
  geom_line(aes(y = ekgf, colour = "ekg")) +  
  geom_line(aes(y = irf, colour = "ir")) +  
  xlab("vreme [s]") +  
  ylab("a.u.") +  
  ggtitle("Mereni signali") +  
  xlim(2, 7)
```

- Za prikaz oba signala na grafiku (kao na slici levo) prikazan je kod na slici desno.
- Šta nije u redu sa ovim grafikom? Kako popraviti? Šta znači a.u.?

Drugačiji prikaz?

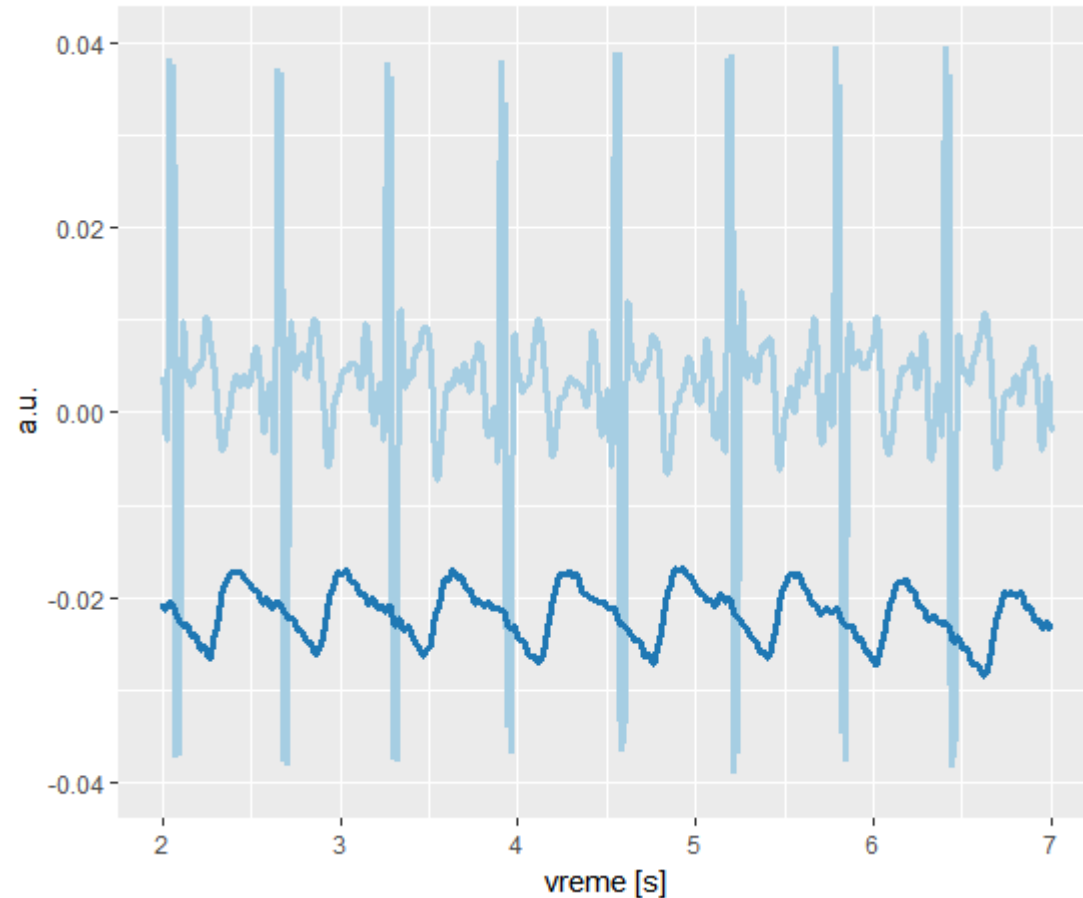


- *Faceting* se u ggplot2 paketu odnosi na ...
 - *lay out panels*
- Za prikaz kao sa slike transformisan je *data frame*. To je najčešće jedan od najvažnijih koraka u ggplot2 vizuelizaciji.
- Obratiti pažnju na funkcije koje omogućavaju prikaz signala kao na slici levo.

```
podN <- data.frame(vrednost = c(pod$ekgf, pod$irf),  
                  vreme = c(pod$vrema, pod$vrema),  
                  signal = c(rep("ekg", length(pod$ekgf)),  
                             rep("ir", length(pod$irf))))  
  
ggplot(data = podN,  
       aes(x = vreme, y = vrednost)) +  
  geom_line() +  
  facet_wrap(~signal) +  
  xlim(2, 7)
```

Ako je cilj prikaza poređenje ...

Filtrirani signali

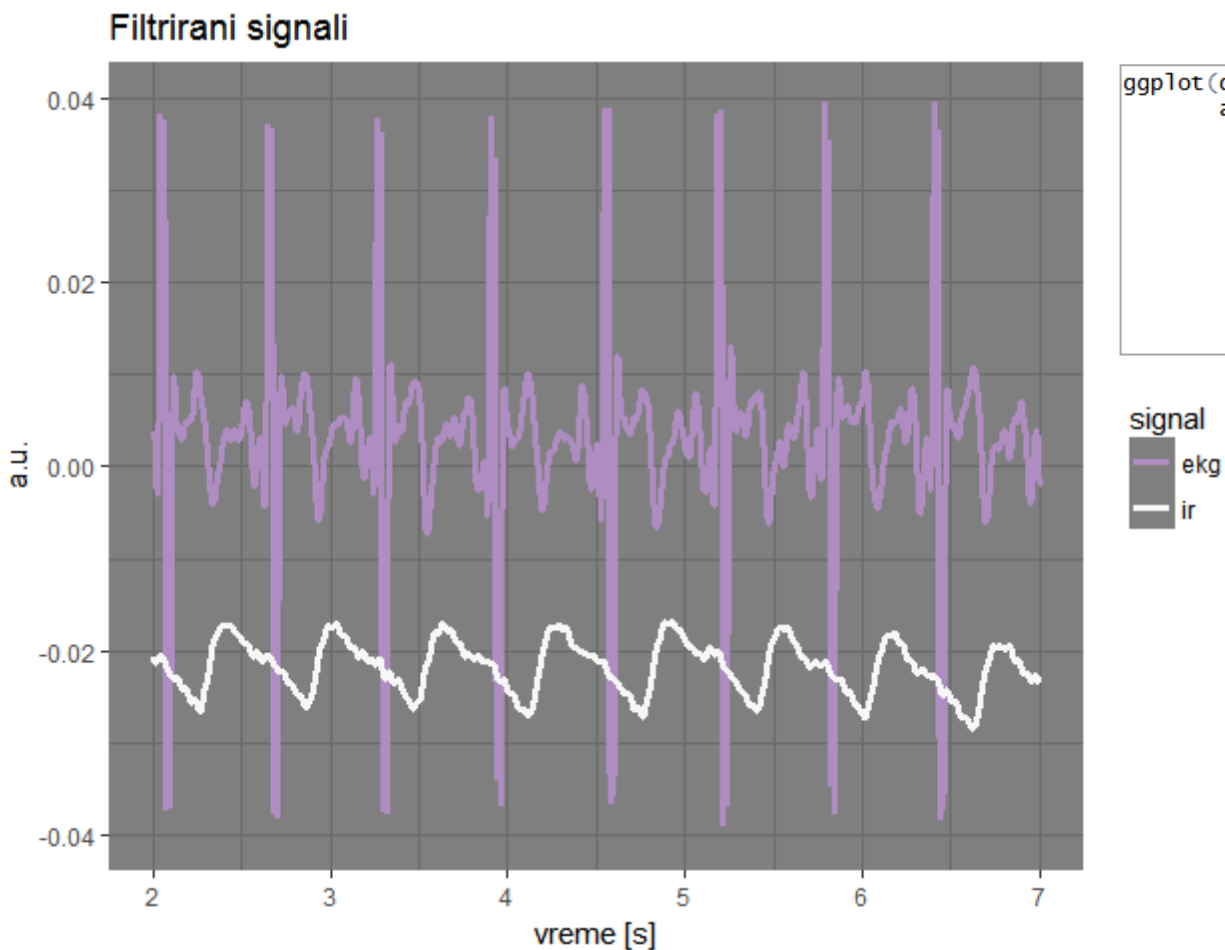


```
podN$vrednost[podN$signal == "ir"] <-  
  podN$vrednost[podN$signal == "ir"] - 1.8  
  
ggplot(data = podN,  
  aes(x = vreme, y = vrednost, color = signal)) +  
  geom_line(size = 1.2) +  
  xlim(2, 7) +  
  ylim(-0.04, 0.04) +  
  ggtitle("Filtrirani signali") +  
  ylab("a.u.") +  
  xlab("vreme [s]") +  
  #scale_color_brewer(palette = "PRGn") +  
  scale_color_brewer(palette = "Paired")# +  
  #theme_dark()
```

signal
— ekg
— ir

- onda je poželjno da signali budu prikazani na istom grafiku i u istoj skali.
- Kako je odabrana transformacija sa slike? Kako je trebalo pravilno izvesti ovu transformaciju?

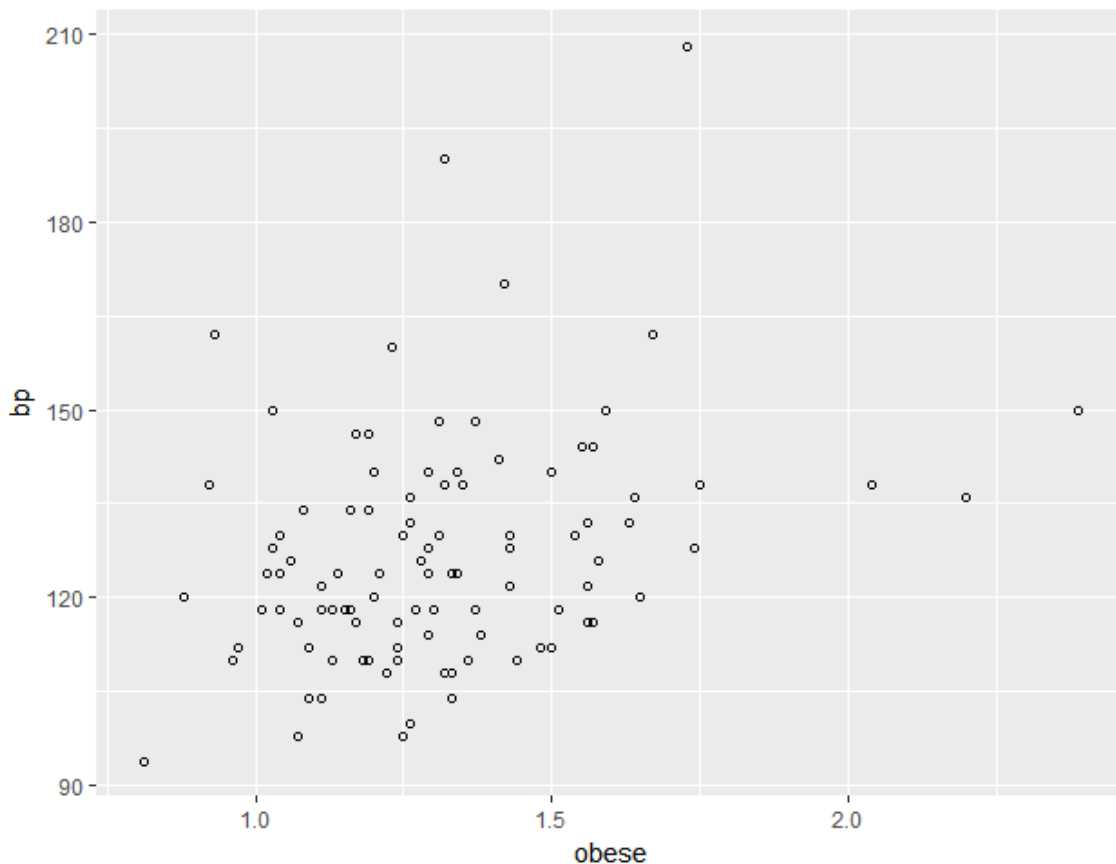
Koliko su boje važne?



```
ggplot(data = podN,  
  aes(x = vreme, y = vrednost, color = signal)) +  
  geom_line(size = 1.2) +  
  xlim(2, 7) +  
  ylim(-0.04, 0.04) +  
  ggtitle("Filtrirani signali") +  
  ylab("a.u.") +  
  xlab("vreme [s]") +  
  scale_color_brewer(palette = "PRGn") +  
  #scale_color_brewer(palette = "Paired")# +  
  theme_dark()
```

- Šta je promenjeno u odnosu na prethodni grafik?
- Da li ovakav prikaz više ili manje može da se koristi poređenje dva grafika?

Zavisnost dva signala (*scatter plot*)

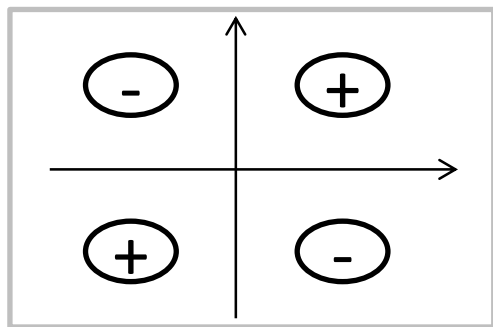
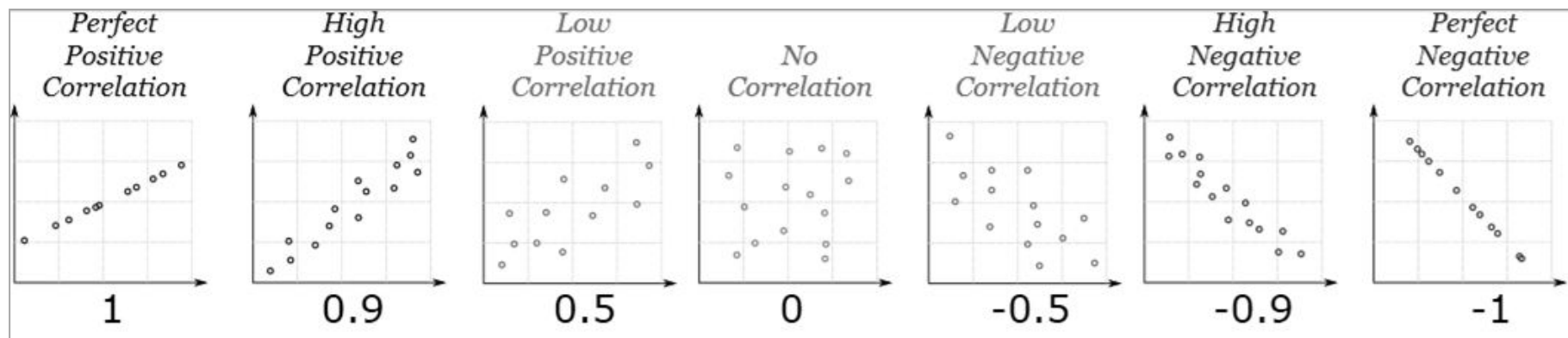


```
library(ISwR)
dat <- bp.obese
head(dat)

ggplot(dat, aes(x = obese, y = bp)) +
  geom_point(shape=1)
```

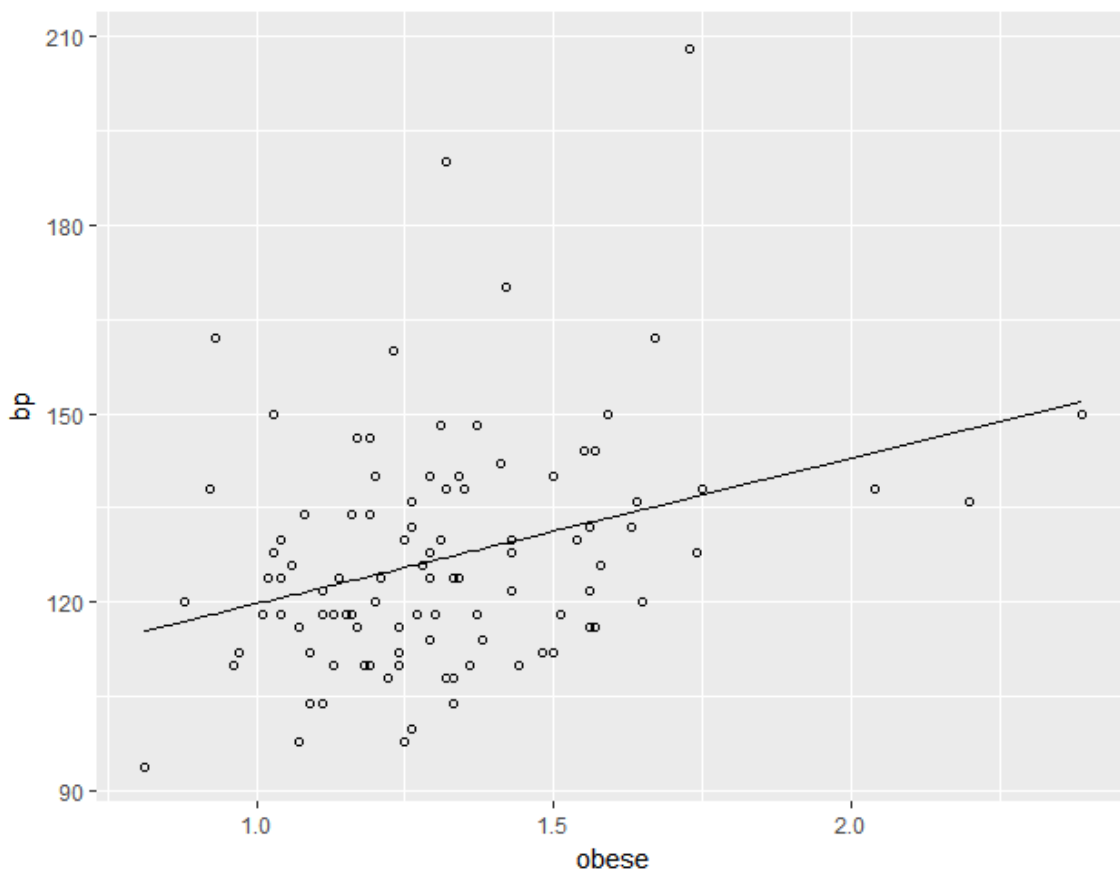
- Prikazana je zavisnost BMI i sistolnog krvnog pritiska (“bp.obese” podaci iz paketa ISwR).
- Grafik prikazan na slici zove se *scatter plot*.
- Služi da se vizuelizuje zavisnost dva signala/parametra (https://en.wikipedia.org/wiki/Scatter_plot) i može biti odličan predkorak u daljoj analizi signala. U neku ruku, na prethodnom grafiku su data dva *scatter plot*-a: IR i EKG u zavisnosti od vremena.

Anatomija *scatter plot*-a



- Slika na gornjem panelu je preuzeta sa sajta: <https://www.mathsisfun.com/data/images/correlation-examples.svg>. Prikazani su primeri *scatter plot* grafika (zavisnost dve promenljive) i njihova korelacija (tj. linearna zavisnost).
- Na donjem panelu (levo) je prikazana primer grafičke predstave korelacije.
- Na ovom dijagramu se može videti i nelinearna zavisnost i delimična korelacija podataka. Kako?

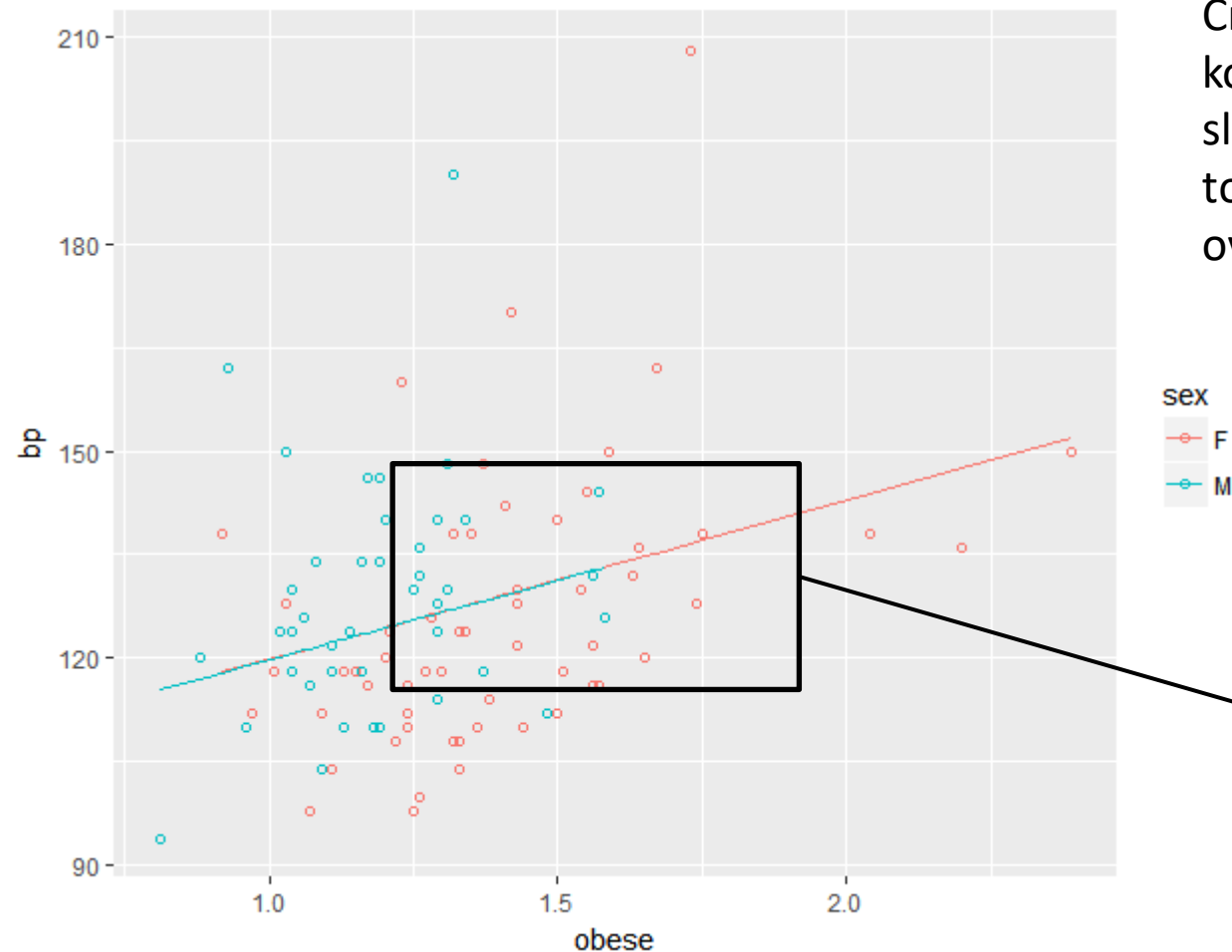
Zavisnost dva signala (*scatter plot*)



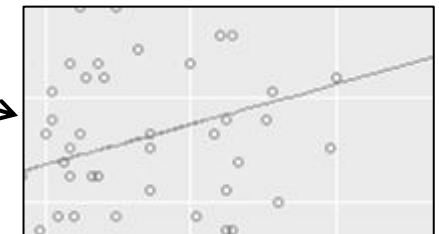
```
dat$pred <- predict(lm(bp ~ obese, data = dat))  
ggplot(dat, aes(x = obese, y = bp)) +  
  geom_point(shape=1) +  
  geom_line(aes(y = pred))
```

- Sa grafika se “naslućuje” linearna zavisnost ova dva signala.
- Mogu se koristiti *lm()* i *predict()* funkcije da se pronađe linearni model za ova dva parametra.
- Ovo sve bi trebalo uraditi automatski u *ggplot2* paketu – on je i kreiran sa tim ciljem.

Ili?

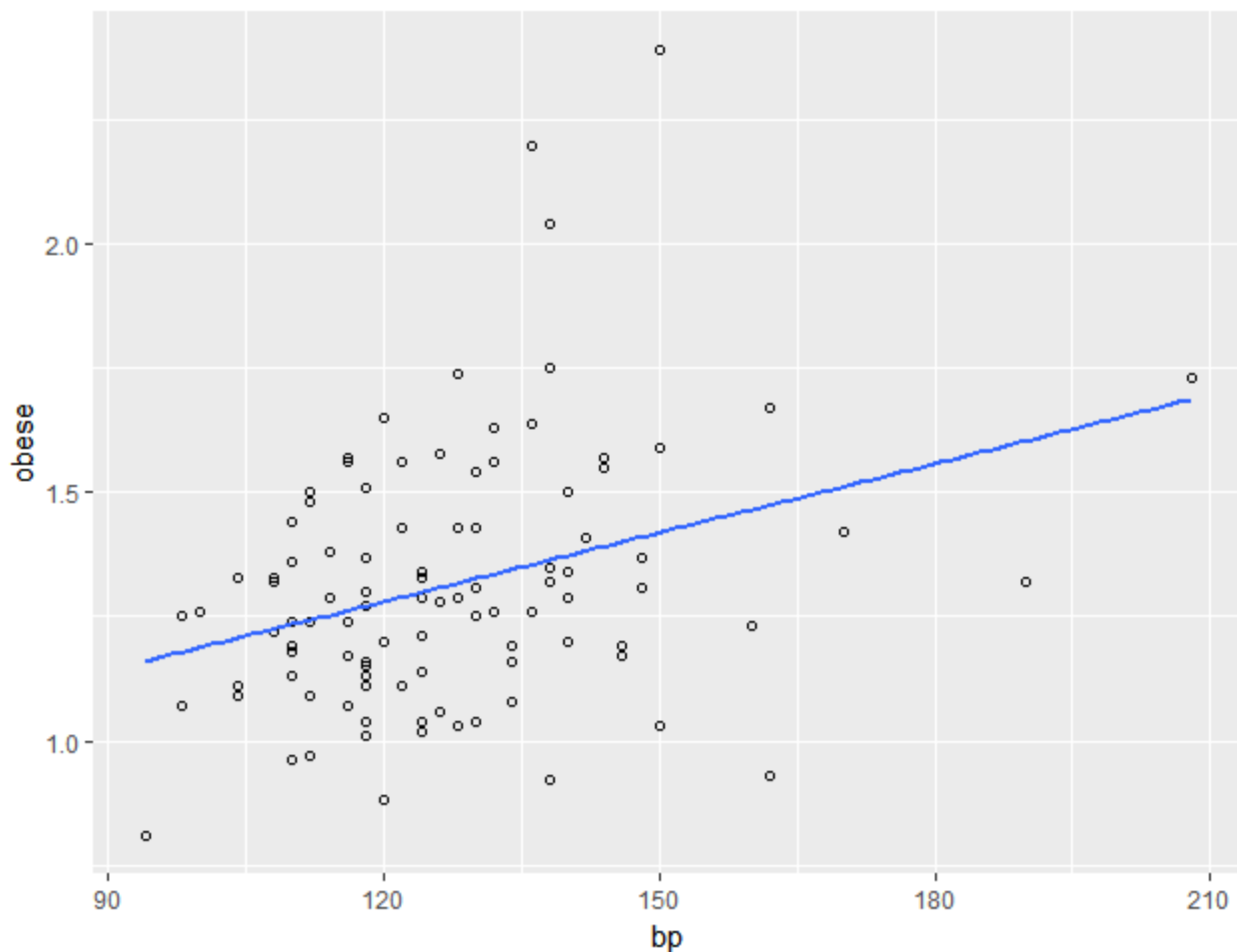


Crvenu i plavu boju trebalo bi koristiti u kombinaciji samo u slučaju kada postoji razlog za to, jer u sivoj skali se najčešće ove dve boje ne razlikuju.



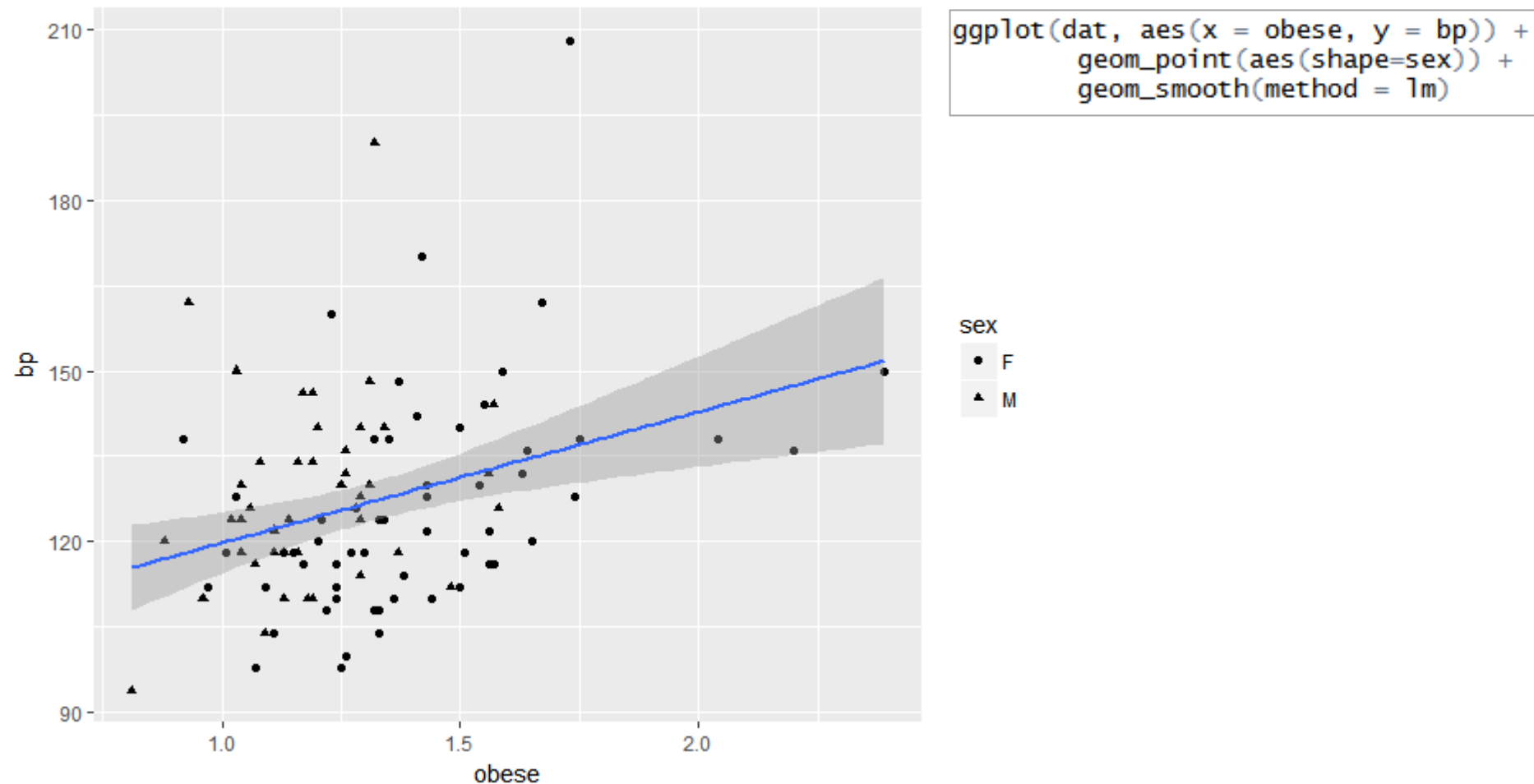
```
dat$sex <- ifelse(dat$sex==1, "F", "M")
ggplot(dat, aes(x = obese, y = bp, col = sex)) +
  geom_point(shape=1) +
  geom_line(aes(y = pred))
```

Linearan model ggplot2



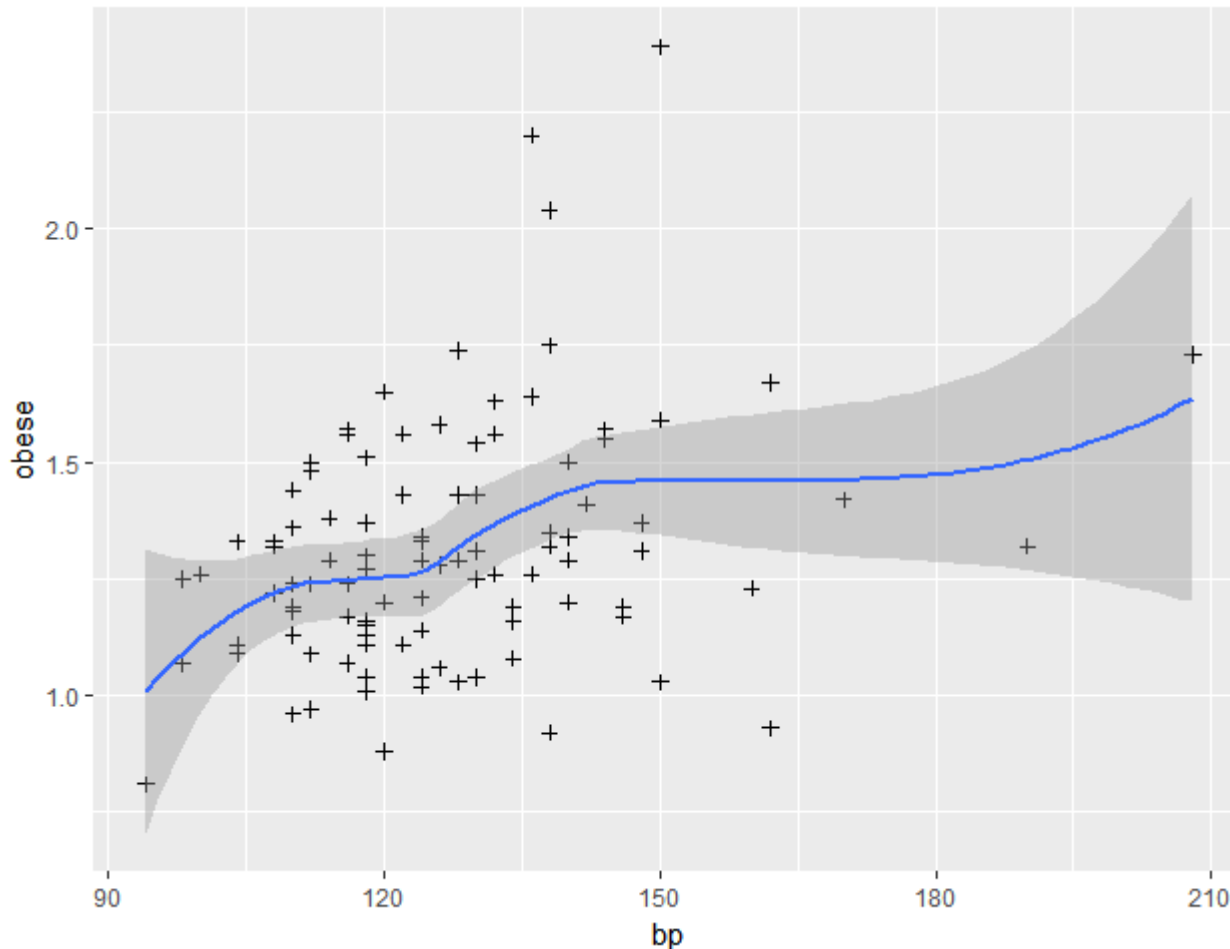
```
ggplot(dat, aes(x = bp, y = obese)) +  
  geom_point(shape = 1) + # koriste se kružići  
  geom_smooth(method = 'lm', se = FALSE) # lin. regresija
```


Linearan model – kompletan prikaz



- Primetiti da je umesto boja moguće koristiti i druge oblike za prikaz vrednosti na grafiku.
- Interval koji je prikazan na slici odgovara intervalu od 95% verovatnoće u linearnom modelu.

A drugi modeli?



```
ggplot(dat, aes(x = bp, y = obese)) +  
  geom_point(shape = 3) +  
  geom_smooth()
```

- Na grafiku se koristi loess (eng. *Local Polynomial Regression Fitting*) model.
- Koji model “više” odgovara ovim podacima “lm” ili “loess”?
- Da li postoji deo ovih podataka koje nije moguće modelirati na odgovarajući način? Zašto?

ggplot2 još?

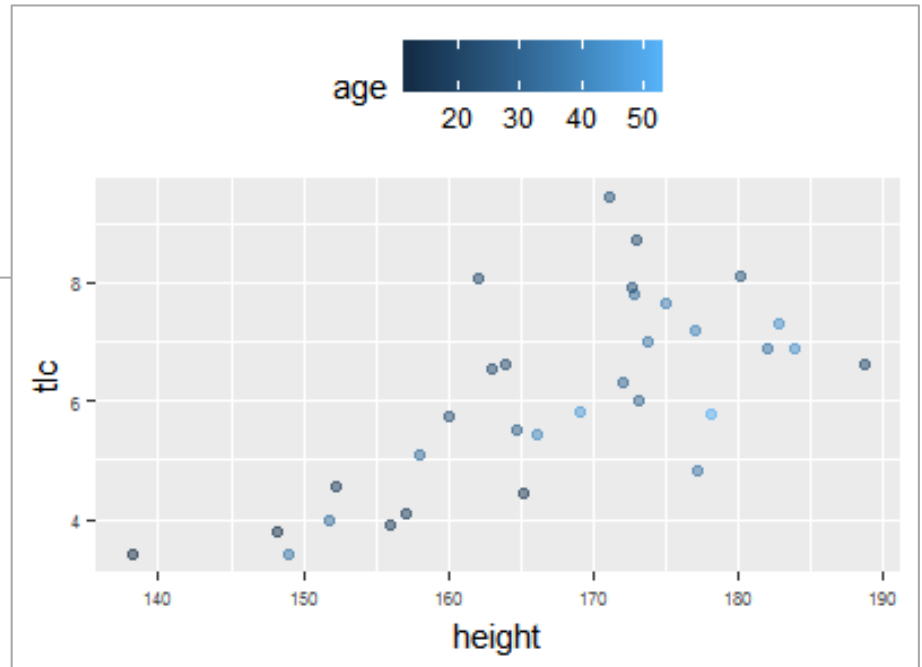
- Više na:
 - [http://www.cookbook-r.com/Graphs/Scatterplots_\(ggplot2\)/](http://www.cookbook-r.com/Graphs/Scatterplots_(ggplot2)/) i
 - <http://www.sthda.com/english/wiki/ggplot2-scatter-plots-quick-start-guide-r-software-and-data-visualization>
 - <http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html>
- Promenljive se mapiraju pomoću *aes()* funkcije, dok se estetski elementi koji su fiksirani podešavaju izvan *aes()* funkcije. Voditi računa da ne dođe do greške!
- Neki grafici ne zahtevaju transformacije podataka kada se podaci prikazuju (npr. *scatterplot*), ali drugi grafici zahtevaju transformacije podataka (npr. *boxplot*).
- Svaki *geom* ima podrazumevanu statistiku, ali je moguće ovu statistiku promeniti (transformisati).

Skaliranje

- Skaliranje u ggplot2 paketu uključuje:
 - poziciju,
 - boju (eng. *color, fill*),
 - veličinu,
 - oblik i
 - tip linije.
- Postoji niz funkcija koji se koristi za skaliranje i cela serija tih funkcija počinje sa *scale_()*.
- Uobičajeni argumenti ovih funkcija su:
 - ime (naslov ose ili legende),
 - granice (minimalna i maksimalna vrednost skale),
 - *breaks* (tačke na skali na kojima bi trebalo postaviti oznake, anotacije),
 - labele (nazivi koji se pojavljuju kod svake anotacije) i
 - postoje i drugi argumenti za neke funkcije (*scale_color_continuous* je argument koji omogućava da se boja postavi na vrednost *low* i *high*).

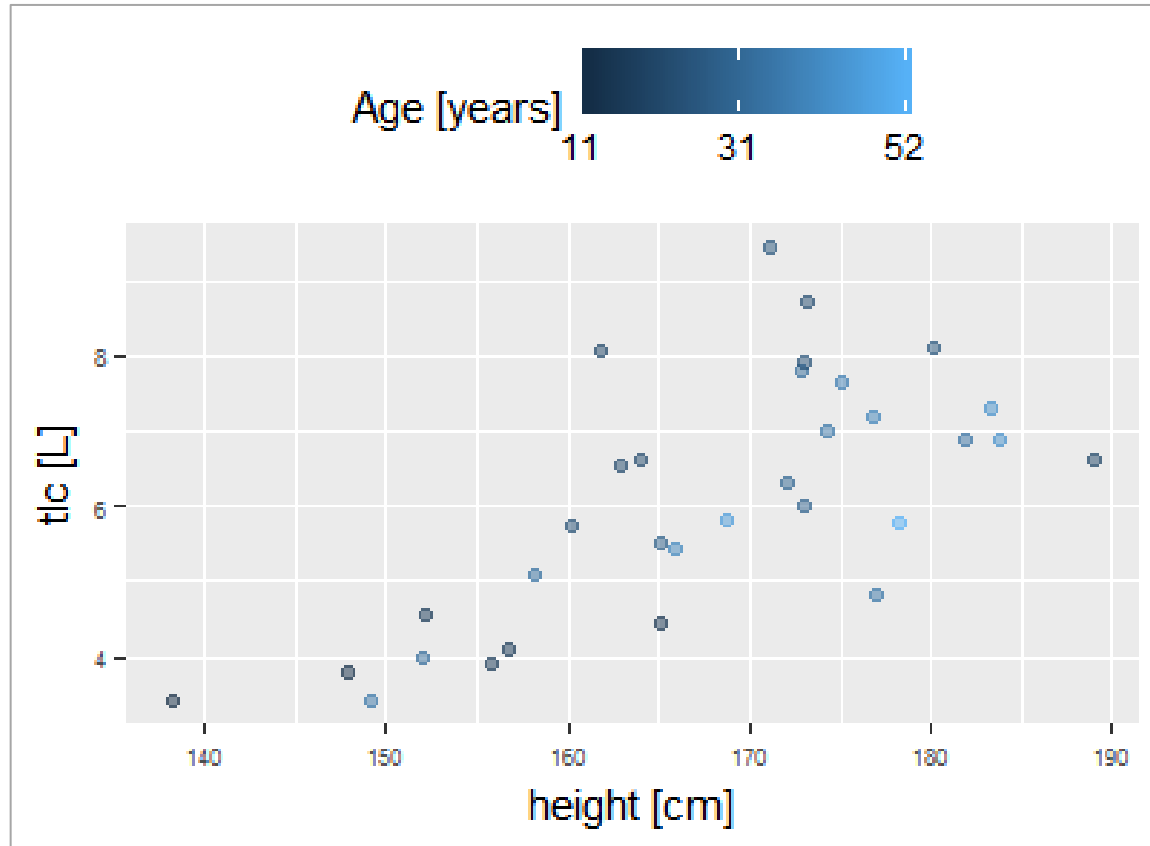
Primer grafika

```
> library(ISwR)
> dat <- tlc
> head(tlc)
  age sex height  tlc
1  35  1   149  3.40
2  11  1   138  3.41
3  12  2   148  3.80
4  16  1   156  3.90
5  32  1   152  4.00
6  16  1   157  4.10
>
> gr <- ggplot(dat, aes(x = height,
+                       y = tlc)) +
+   theme(legend.position = "top",
+         axis.text = element_text(size = 6))
> gr1 <- gr + geom_point(aes(color = age),
+                         alpha = 0.5,
+                         size = 1.5,
+                         position = position_jitter(width = 0.25, height = 0))
> print(gr1)
```



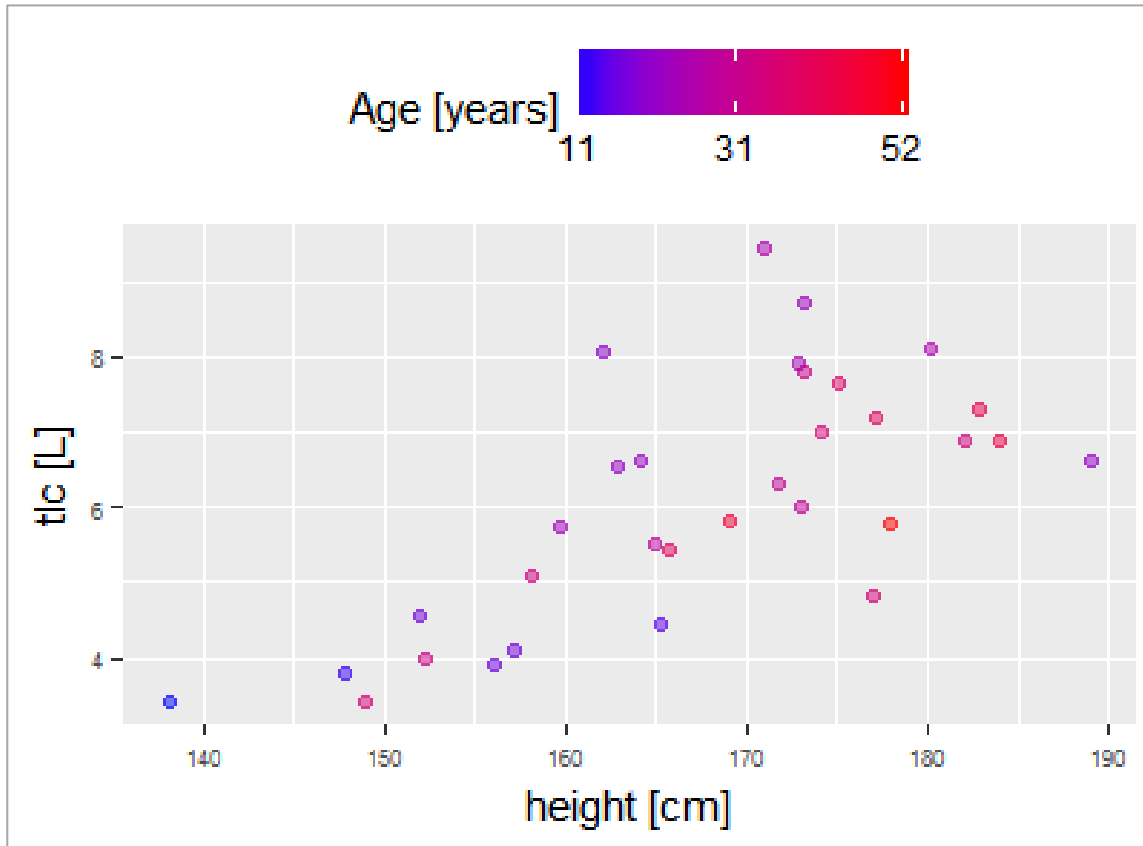
Korišćeni su podaci iz ISwR paketa "tlc" (eng. *Total Lung Capacity*).

Skaliranje boje



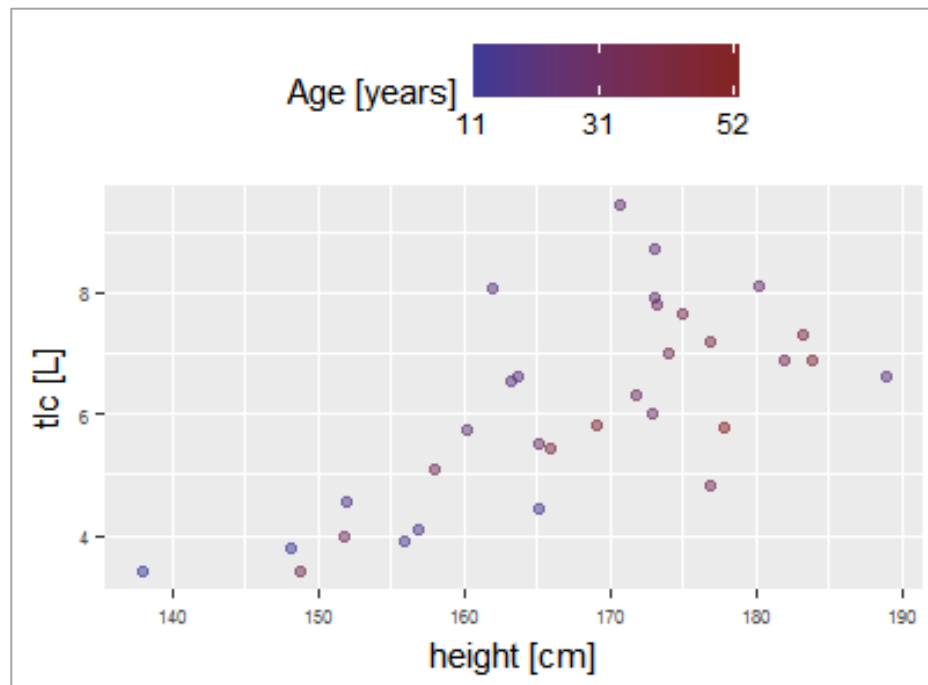
```
> gr1 + scale_color_continuous(name = "Age [years]",  
+                               breaks = c(11, 31, 52),  
+                               labels = c(11, 31, 52)) +  
+   xlab("height [cm]") +  
+   ylab("tlc [L]")
```

Još boja?



```
> gr1 + scale_color_continuous(name = "Age [years]",  
+                               breaks = c(11, 31, 52),  
+                               labels = c(11, 31, 52),  
+                               low = "blue", high = "red") +  
+                               xlab("height [cm]") +  
+                               ylab("tlc [L]")
```

scales paket (CRAN)



```
> library(scales)
> gr1 + scale_color_continuous(name = "Age [years]",
+                             breaks = c(11, 31, 52),
+                             labels = c(11, 31, 52),
+                             low = muted("blue"), high = muted("red")) +
+   xlab("height [cm]") +
+   ylab("t1c [L]")
>
>
> show_col(c("red", "blue", muted("red"), muted("blue")))
```

- Ovaj paket omogućava modifikaciju i skaliranje boja po želji korisnika.
- Na slici je prikazana funkcija `scales()` i `show_col()` primena ovih boja na grafik sa prethodnog slajda.
- Za veći broj ispitanika ove promene bi bile izražajnije.

Muted boje



By Cpl. Megan L. Stiner, modified by Jacob Rus - modified from Image:Surfing in Hawaii.jpg, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=4609704>.



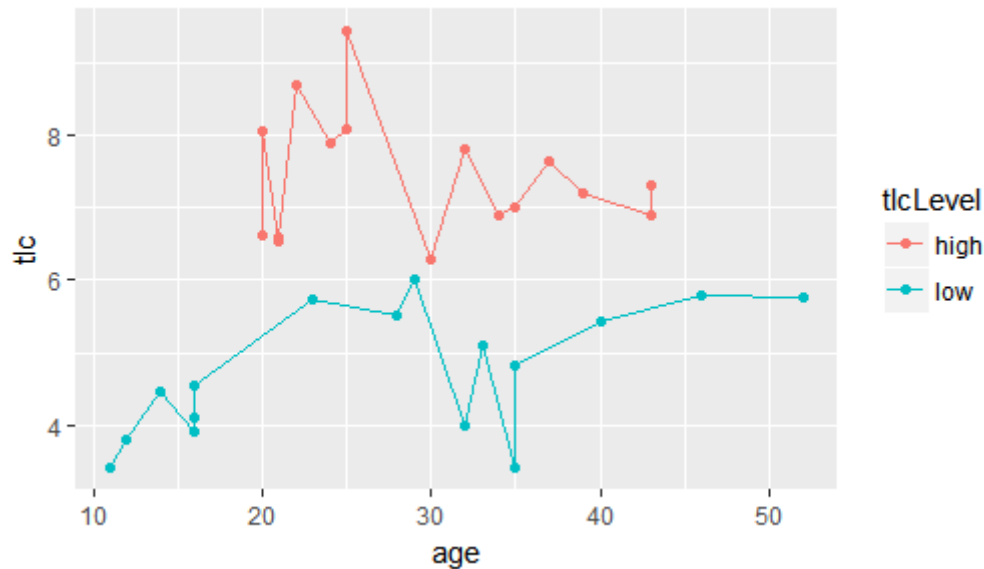
By Cpl. Megan L. Stiner, sRGB profile added by Jacob Rus - modified from File:Surfing in Hawaii.jpg, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=4609599>.

- *Muted* (zatamnjene) boje su boje kojima je dodata siva nijansa (<https://en.wikipedia.org/wiki/Colorfulness>).
- Na slici je prikazana razlika između slike koja ima zatamenjene i koja nema ove nijanse.
- Još jedan važan parametar kod odabira boja je *flat design* (https://en.wikipedia.org/wiki/Flat_design) koji se koristi za dizajn GUI-ja i predstavlja poseban minimalistički stil.

Faceting – po čemu je ggplot2 poznat

- Odnosi se na podelu prozora grafika na manje prozore i prikaz većeg broja signala ili podskupa nekog signala.
- Koriste se dve funkcije u ovu svrhu: *facet_wrap()* i *facet_grid()*.
- Poželjno je koristiti *faceting* jer može dati korisnu informaciju pogotovu kod poređenja više signala.
- Primer signala koji su predstavljeni u jednom i u više prozora dat je na sledećim slajdovima.

Faceting – primer

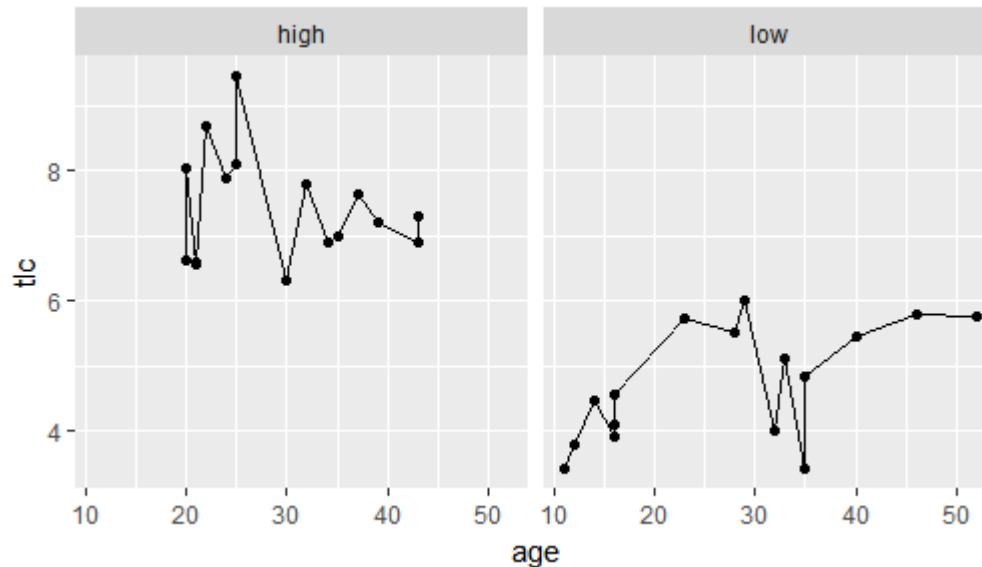


```
library(ISwR)
dat <- t1c
dat$t1cLevel = ifelse(dat$t1c > 6, "high", "low")

graf1 <- ggplot(dat, aes(x = age, y = t1c))

graf1 + geom_point(aes(color = t1cLevel)) +
  geom_line(aes(color = t1cLevel))

graf1 + geom_point() +
  geom_line() +
  facet_wrap(~ t1cLevel)
```



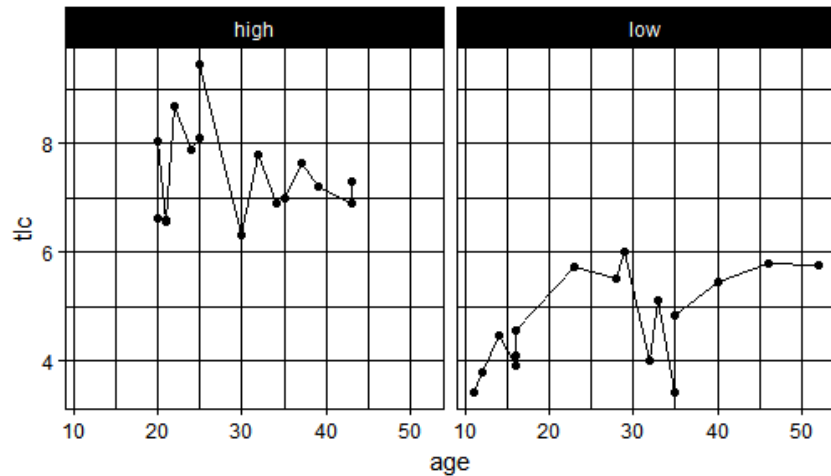
Moguće je koristiti i funkciju *facet_grid()*.

Themes

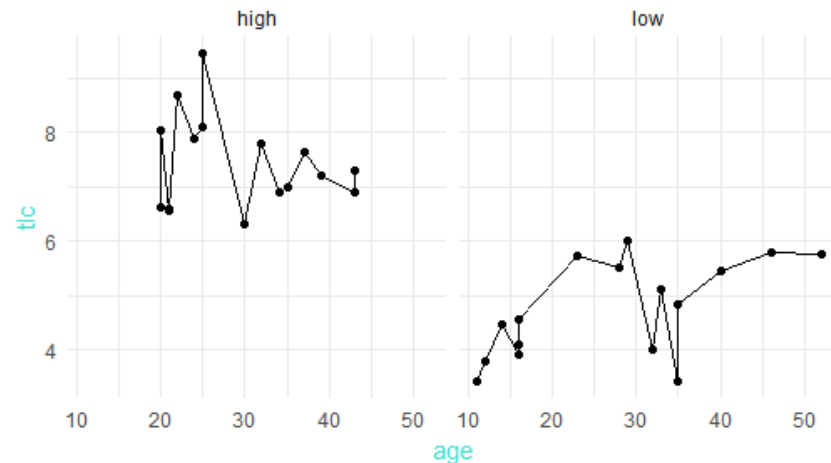
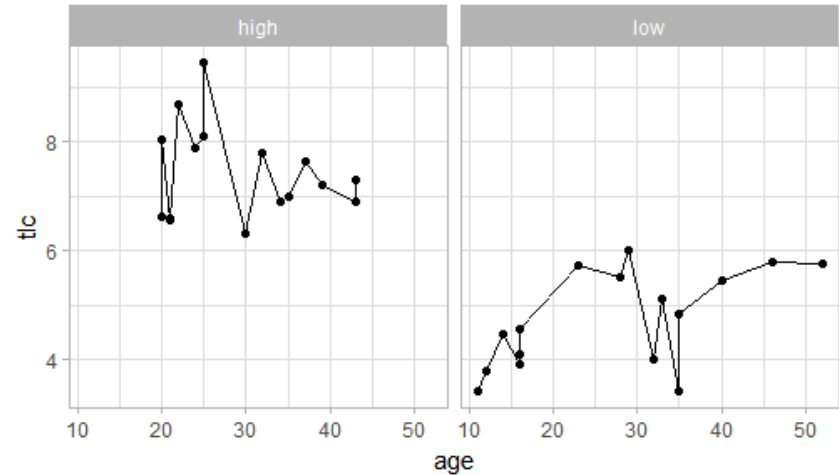
- Postoje različite teme koje je moguće koristiti u R-u, a odnose se na:
 - labele/oznake osa,
 - pozadinu *plot*-a,
 - pozadinu labela/oznaka i
 - na formatiranje legende.
- Postoje ugrađene teme kao što su: *theme_gray()*, *theme_bw()*, *theme_classic()*.
- Mogu se kreirati i nove *themes*, ali za sada dovoljno.

Themes – primeri

```
graf1 + geom_point() +  
  geom_line() +  
  facet_wrap(~ tlcLevel) +  
  theme_linedraw()
```



```
graf1 + geom_point() +  
  geom_line() +  
  facet_wrap(~ tlcLevel) +  
  theme_light()
```



Bolje?

Sve informacije se nalaze na stranici *?theme*.

```
graf1 + geom_point() +  
  geom_line() +  
  facet_wrap(~ tlcLevel) +  
  theme_minimal() + theme(text = element_text(color = "turquoise"))
```


Osnovno pravilo obrade signala



- Prvo vizuelizacija, pa tek onda obrada! Još od osnovnihi studija i MAS predmeta.
- Najvažnija je vizuelizacija. Ono što se ne vidi na grafiku neće se ni izdvojiti primenom tehnike obrade signala / podataka.
- Nekada se obrada signala završava vizuelizacijom.
- Slika je sa: By Randri87 - <https://www.triopticaonline.com/comprar/gafas-de-vista/>, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=52841451>.

Ostala pravila

- *De gustibus non disputandum est*, ali ...
- Cilj vizuelizacije je da se na odgovarajući način prikaže neka informacija. I trebalo bi se pridržavati bar nekih osnovnih pravila.
- Zato nikada, ali baš nikada, ne treba:
 - koristiti *pie chart*,
 - boje koje nemaju dobar kontrast,
 - 3D prikaz ako postoji 2D grafik koji opisuje istu zavisnost,
 - boje za informaciju koja je već prikazana na drugi način i
 - koristiti grafik koji ne prikazuje ili delimično prikazuje informaciju od interesa.
- Za svaku preporuku je da se pogleda video prof. Rafael-a Irizarry-ja pod nazivom “Statistics for Genomics: Useful plots and bad plots” na linku: <https://www.youtube.com/watch?v=46-t2jOYsyY>. (od 8:07 – *common mistakes*), ako već niste gledali.



EAGLE
HILLS

Još neka pravila

- Generalno za dizajn izveštaja, sajta, grafika:
 - Najvažniji elementi bi trebalo da su na vrhu strane.
 - Ako želite da neko obrati pažnju na detalj (da ga “kupi”) istaknite taj detalj.
 - Većina ljudi gleda levo (većina čita sa leva na desno): svi moji grafici u prezentacijama su sa leve strane ... Da li ste приметили?
 - Ljudi čitaju u F formi, odnosno početna slova redova znače.
 - Ako neki deo Vašeg izveštaja, slajda, grafika liči na logo ili reklamu, neće biti primećen. Izbegavajte *banner look*.
 - Uključite *summary*, odnosno kratku informaciju o onome što pričate, neće svako pregledati izveštaj od početka do kraja.
 - Istaknite razlike, npr. “Stara metoda ima uspešnost 78 %, a nova 92 %” i slično.
 - Vaše objašnjenje rezultata treba da bude dopadljivo, ali na prvom mestu tačno i profesionalno.
 - I druga pravila (za više informacija pogledajte i <https://conversionxl.com/19-things-we-can-learn-from-numerous-heatmap-tests/>).

Kada je dozvoljen *pie chart*?



By Scott Bauer - <http://www.ars.usda.gov/is/graphics/photos/mar01/k7633-3.htm>, Јавно власништво,
<https://commons.wikimedia.org/w/index.php?curid=79501>

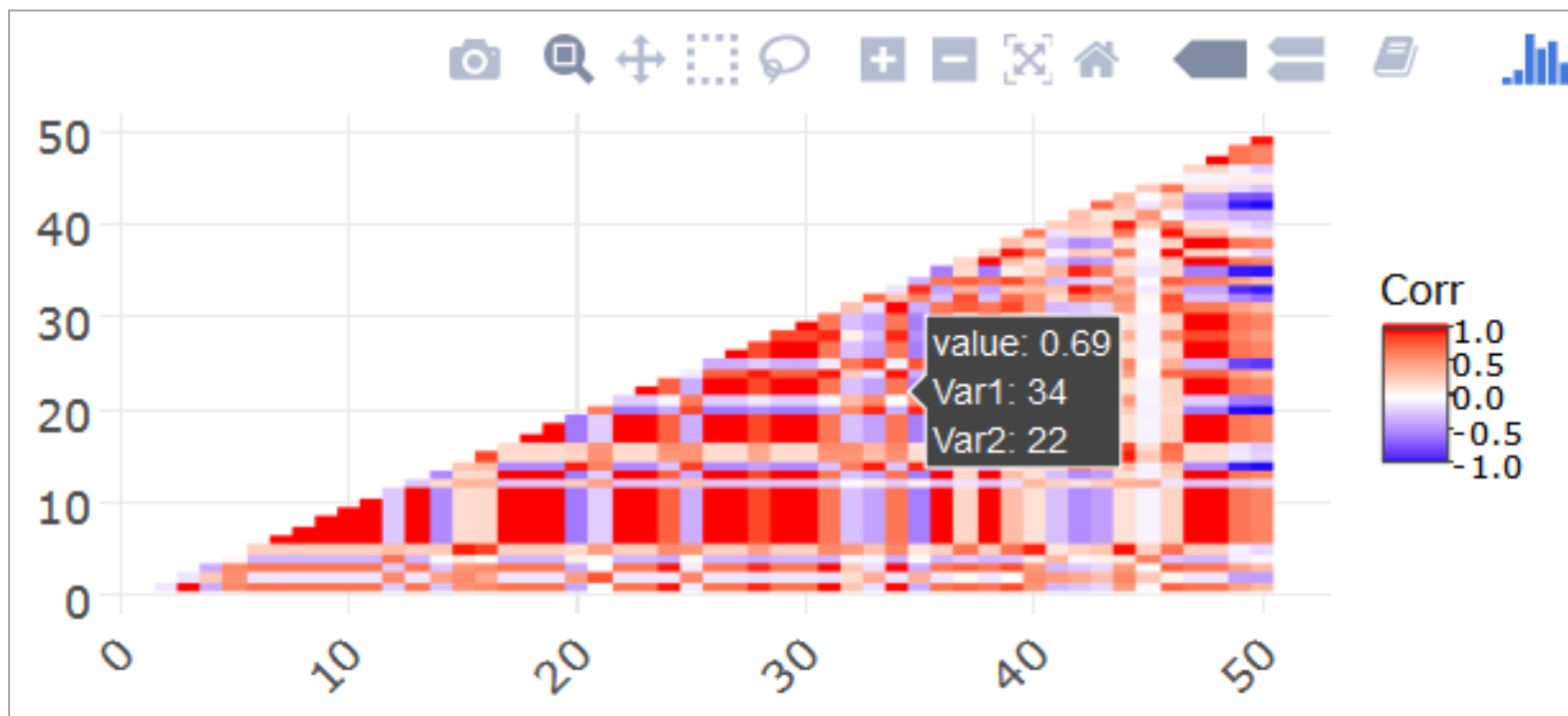
Šta je gore od jednog *pie chart*-a?

- Rotiran *pie chart*
- Dva *pie chart*-a
- ...

TRENDOVI

Interactive is the “new black” for visualization.

plotly paket



- Kako bi grafici bili interaktivni, moguće je koristiti plotly paket sa CRAN-a (<https://cran.r-project.org/web/packages/plotly/index.html>) i funkciju “ggplotly()” iz ovog paketa.
- Primer jednog neinteraktivnog grafika sa ranijih slajdova je iskorišćen za prikaz njegove moderne interaktivne verzije na slici. Obratiti pažnju da je zumiranje grafika omogućeno.
- Kome se ovo sviđa, postoji i online knjiga pod nazivom “plotly for R” koju je napisao Carson Sievert: https://cpsievert.github.io/plotly_book/.

Čarobne reční? Tj. kod?

```
> ggcorrplot(corrM, type = "lower",  
+           outline.col = "white")  
> ggplotly(ggplot2::last_plot())  
> |
```

Još interaktivnih grafika?

Ngram Viewer, [link](#).

Google Books Ngram Viewer

data science,visualization

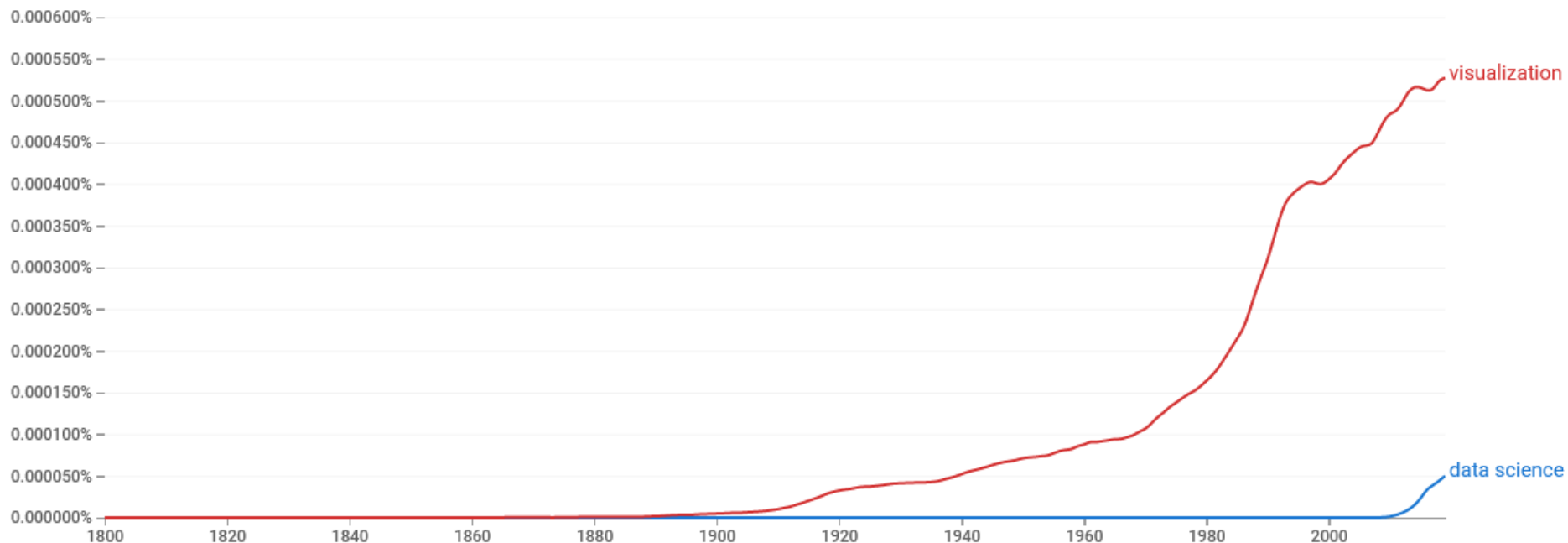


1800 - 2019

English (2019)

Case-Insensitive

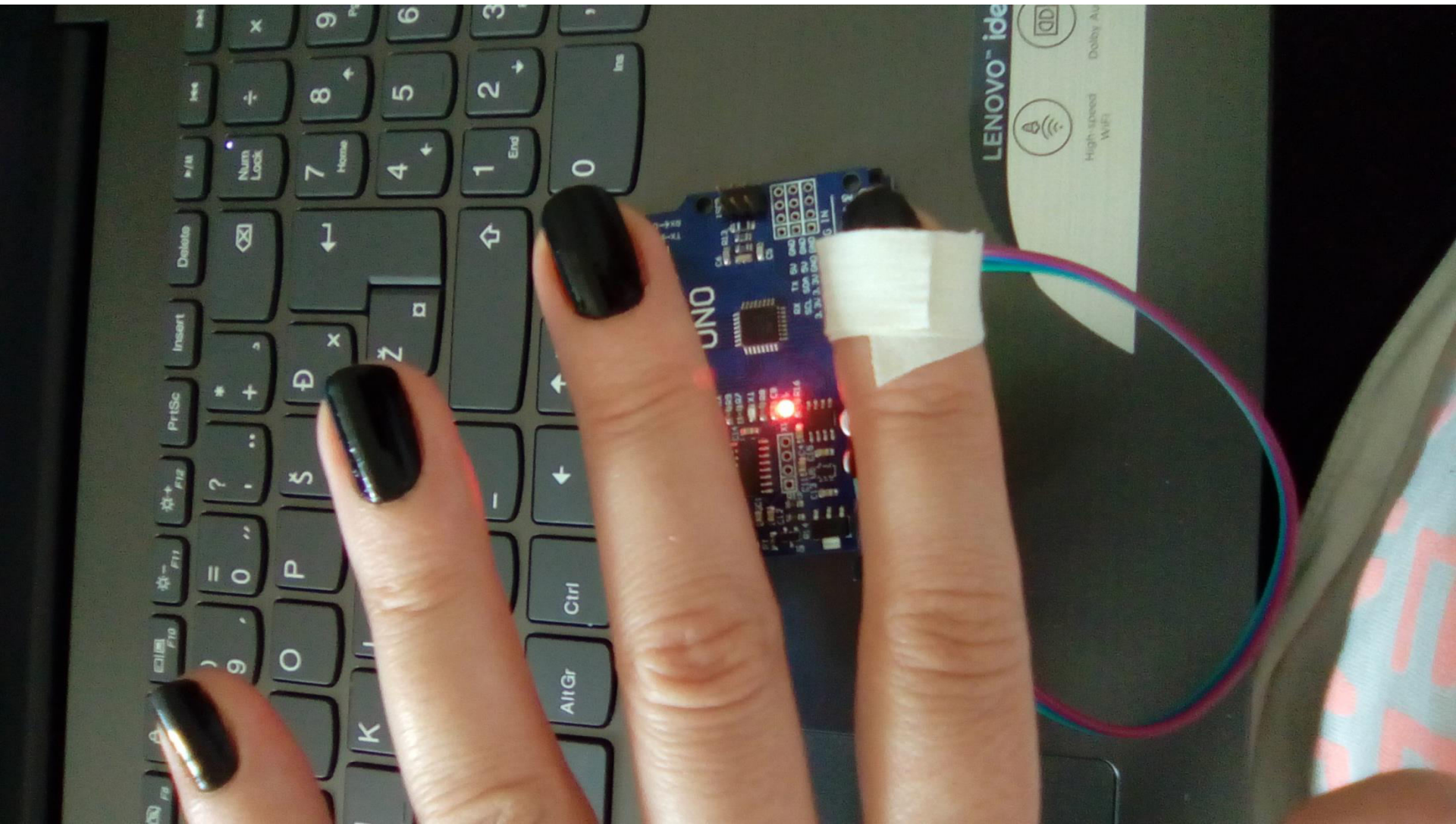
Smoothing



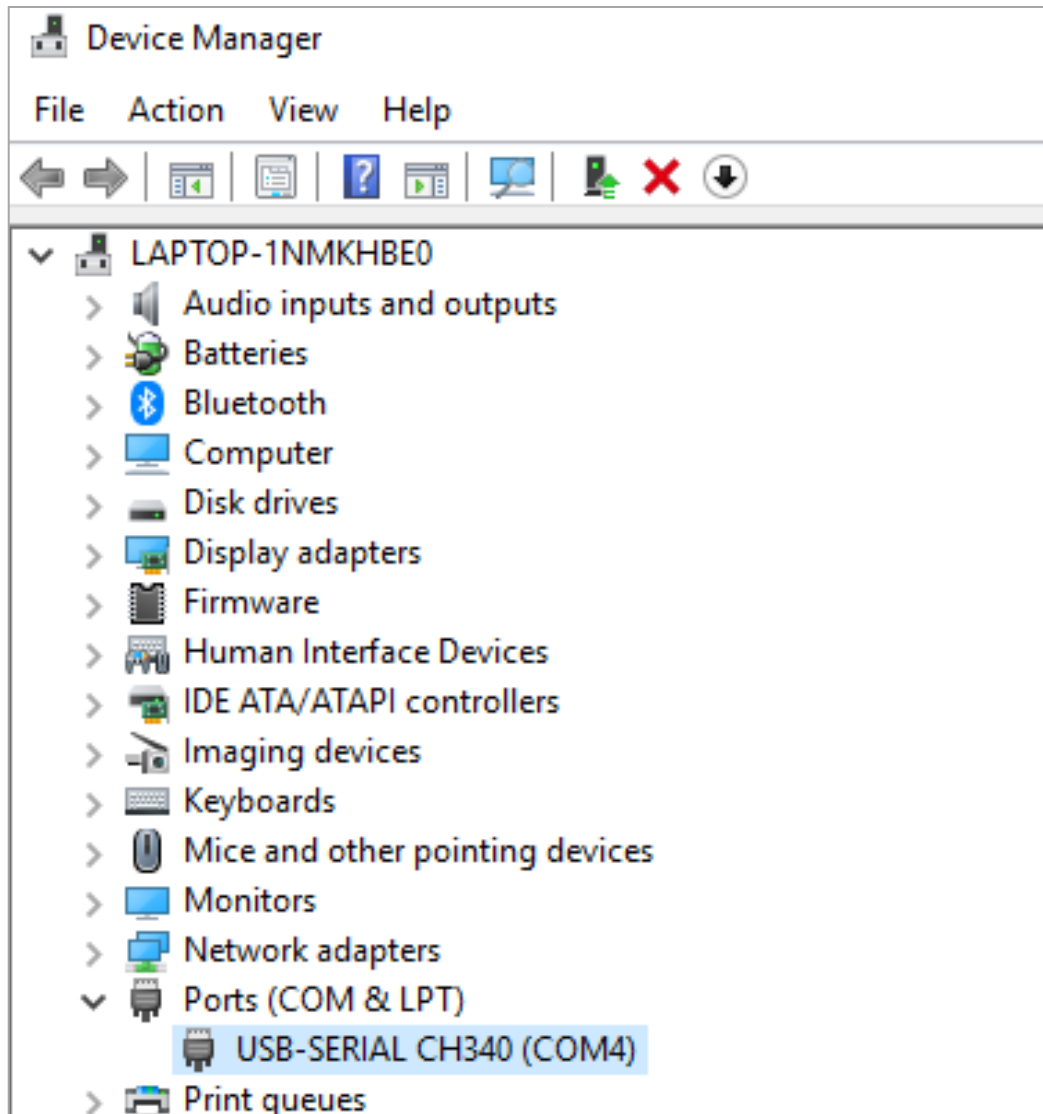
serial biblioteka

- Na CRAN-u postoji biblioteka serial autora Martina Selimajera (<https://cran.r-project.org/web/packages/serial/index.html>) koja se koristi za merenje signala sa hardvera koji koristi RS232/RS422/RS485 ili bilo koji drugi virtuelni serijski interfejs.
- Da vidimo kako radi sa npr. Arduino softverom i UNO R3 hardverom...
- Koga zanima više detalja i povezivanje Arduina i R-a može pogledati Rduino autora Suha i Hofa (<https://cran.r-project.org/web/packages/Rduino/index.html>).

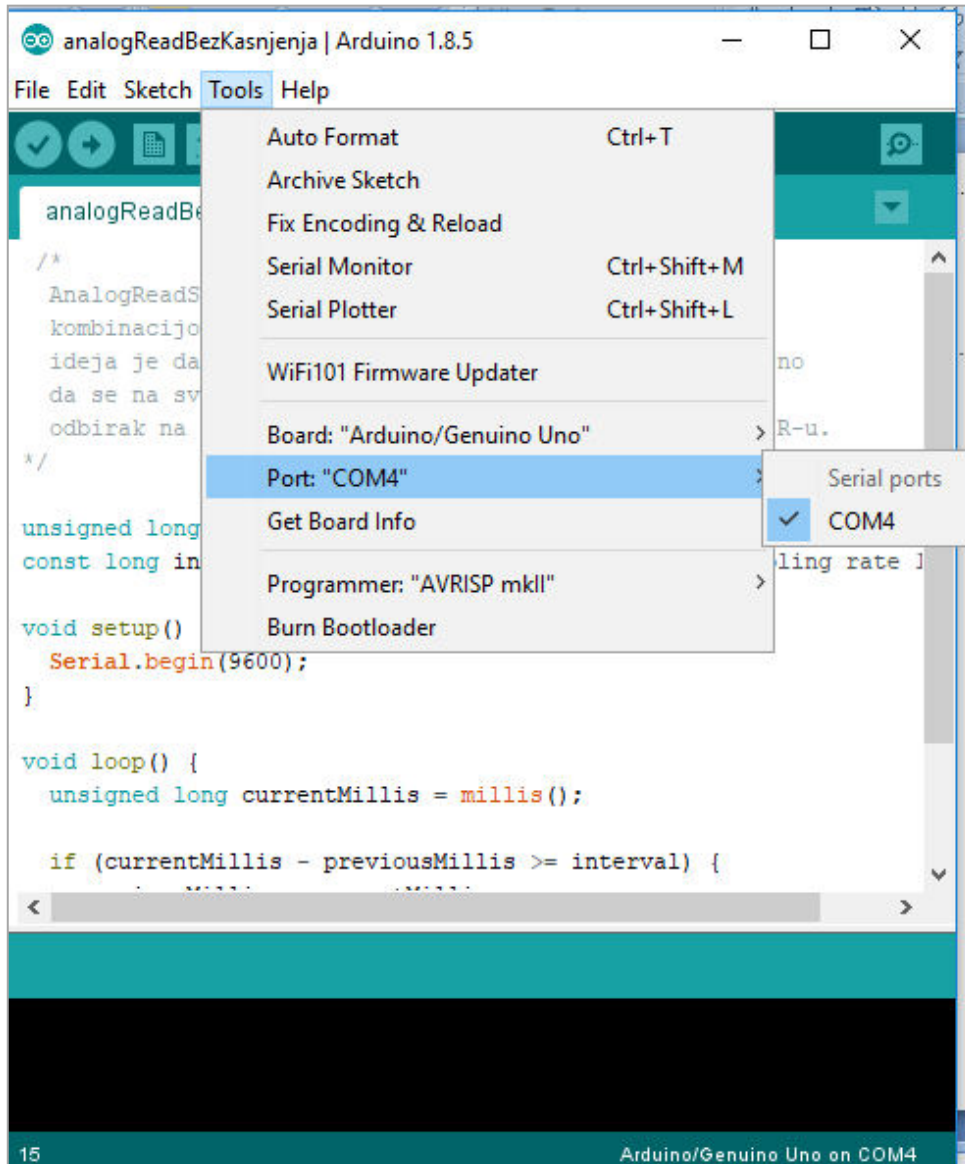
DEMO: senzor za merenje pulsa



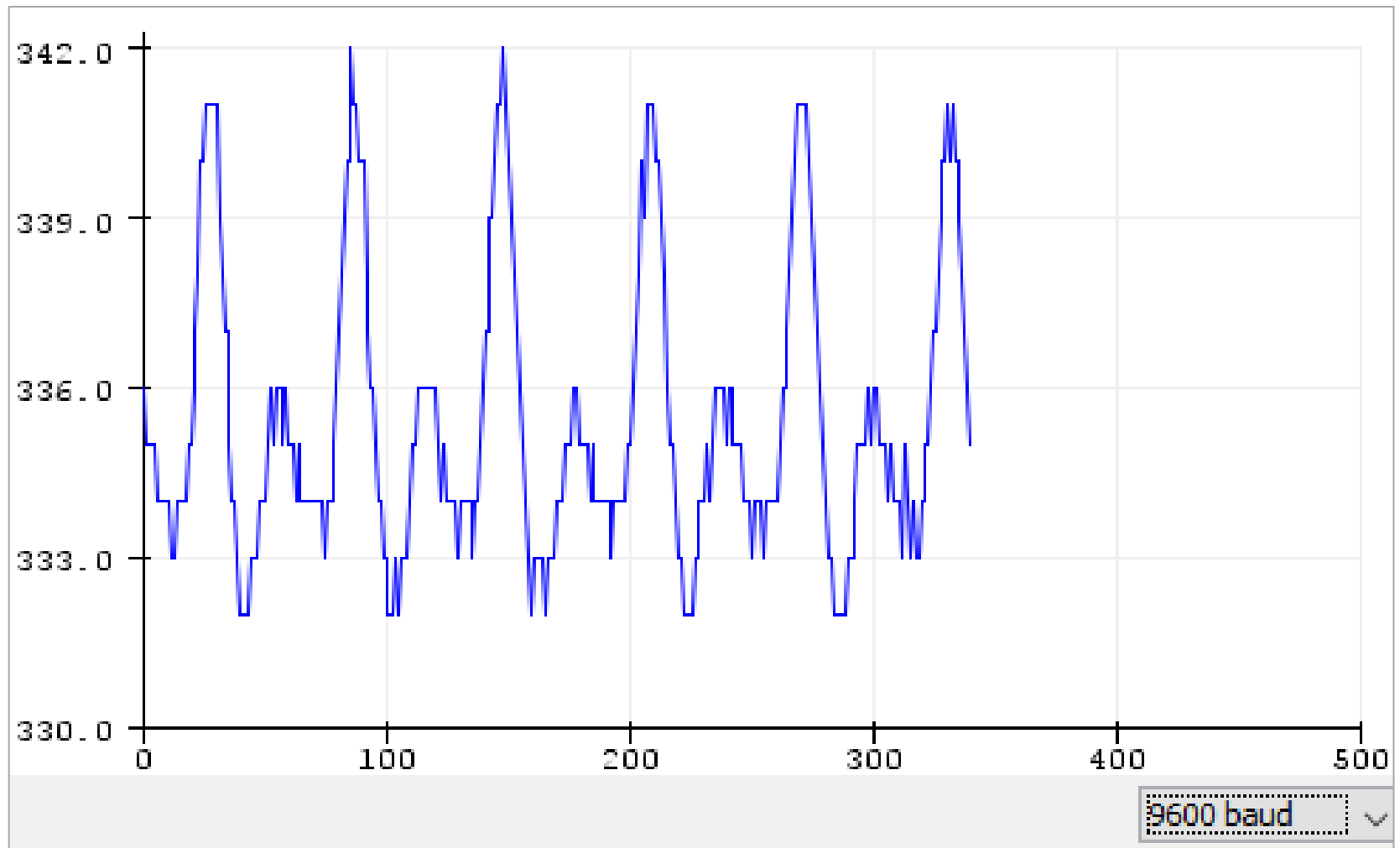
DEMO: COM port za OS Windows



DEMO: Arduino port



DEMO: merenje i slanje na serijski port



```
library(serial)
```

```
# serijska konekcija, ove detalje mozete da pogledate na  
# Control Panel / Device Manager / Ports, pa kliknete na  
# port na kome se nalazi Arduino
```

```
# Moze i na Devices and Printers, pa dupli klik na Arduino
```

```
con <- serialConnection(name = "test_con",  
                        port = "COM4",  
                        mode = "9600,n,8,1",  
                        newline = 1,  
                        translation = "cr")
```

```
# otvaranje konekcije (uvek je prati close)  
open(con)
```

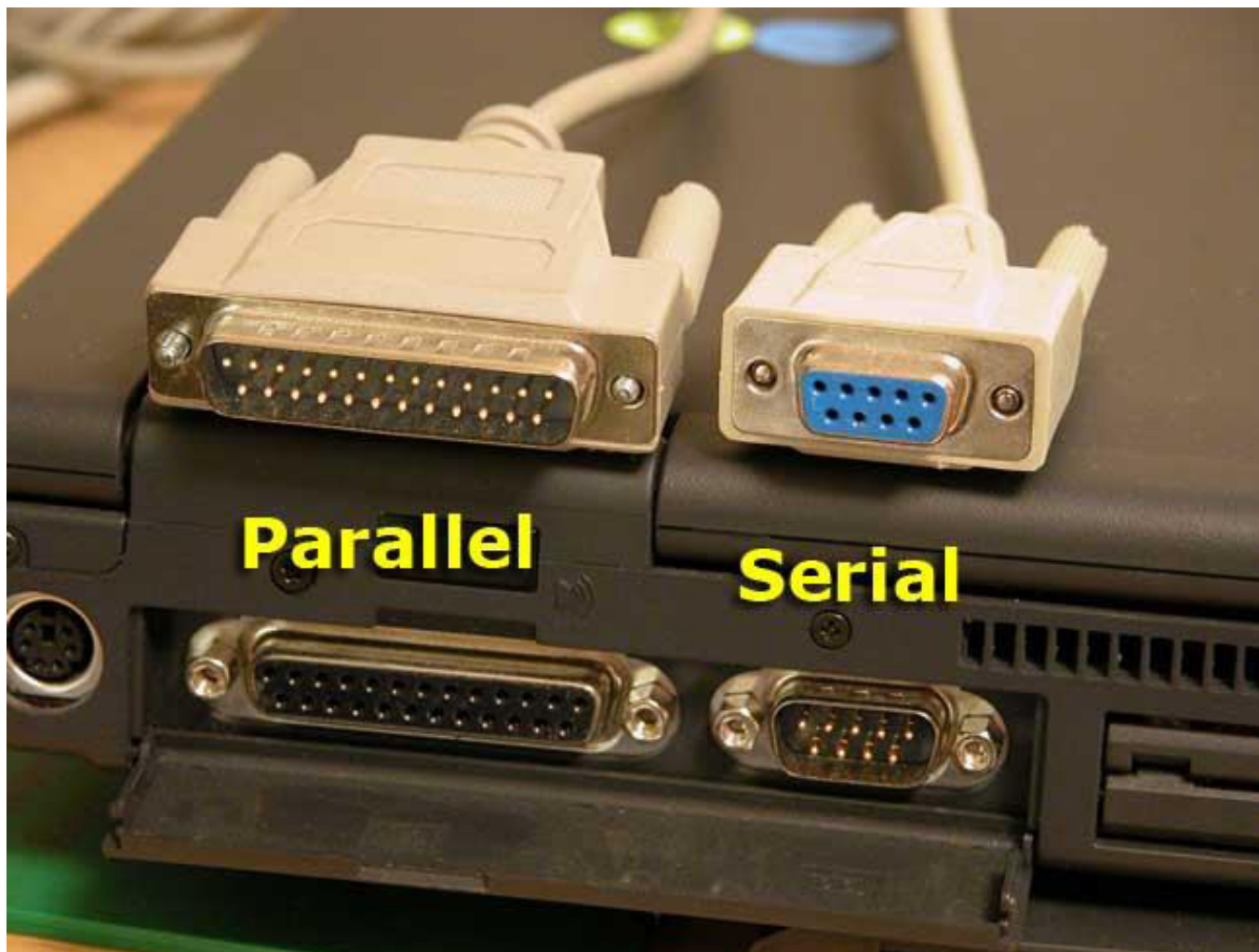
```
n <- 1  
foo <- ""  
textSize <- 0  
while(n < 1004) {  
  # citanje sa serijskog porta  
  newText <- read.serialConnection(con)  
  if(0 < nchar(newText)) {  
    # smestanje u promenljivu foo  
    foo <- paste(foo, newText)  
    n <- n + 1  
  }  
}
```

```
# stampanje u konzoli i snimanje podataka u fajl po izboru  
cat("\r\n", foo, "\r\n", file = "podatak3.txt")
```

```
# zatvaranje konekcije  
close(con)
```

DEMO: R kod

Serijski port i serijska komunikacija



Slika: <https://www.unm.edu/~tbeach/terms/figures/oldports.html>, *Fair Use*.

Primer merenja

- Prikazano je merenje u R-u sa analognog kanala.
- Procedura je sledeća:
 - Senzor za merenje pulsa (<https://pulsesensor.com/>) se poveže sa UNO R3 hardverom sa računarom (USB port)
 - Pokrene se Arduino softver i *upload*-uje se kod u UNO R3 mikrokontrolersku pločicu
 - U kodu bi trebalo da je iskorišćena neka od funkcija za slanje/ispis merenih odbiraka na serijskom portu (*Serial.println*)
 - Potom se pokrene R kod koji omogućava pristup i čitanje sa serijskog porta
 - Podaci koji se čitaju su tekstualni
 - Obavezno je zatvoriti konekciju na kraju čitanja (ako to ne uradite čeka Vas puno ponovnog pokretanja i R-a i Arduino sofvera)
- NAPOMENE: Umesto UNO R3 mikrokontrolerske pločice moguće je koristiti bilo koji hardver/softver koji omogućava ispis na serijskom portu.

Arduino softver i UNO R3 hardver



- Preporučena literatura za osnove programiranja u Arduino programskom okruženju je knjiga u kojoj se koristi hardver UNO R3:
 - Alan G. Smith, Introduction to Arduino: A piece of cake!, online: <https://www.introtoarduino.com/downloads/IntroArduinoBook.pdf>, 2011.
- Arduino softver je napisan u Java, C i C++. Operativni sistemi na kojima je moguće instalirati Arduino softver i programirati u njemu su: Windows, macOS i Linux, <https://en.wikipedia.org/wiki/Arduino>.
 - Arduino je veoma sličan C++ programskom jeziku.
- NAPOMENA: Nisu sve funkcije dostupne za UNO R3 hardver. Neke funkcije iako postoje u softveru su specifične za hardver. Potrebno je pogledati uputstvo za odgovarajuću funkciju.

```

6  import processing.serial.*;
7  import org.rosuda.REngine.Rserve.*;
8  import org.rosuda.REngine.*;
9
10 int mySerialPort = 7; // Change this to the port number of th
11 Serial myPort;      // The serial port
12 int xPos = 1;       // horizontal position of the graph
13 RConnection c;     // R connection variable
14 int ht = 512;       // Height of the graphic window
15 int wd = 512;       // Width of the graphic window
16
17 void setup () {
18     // set the window size:
19     size(ht, wd);
20     smooth();
21
22     // List all the available serial ports
23     println(Serial.list());
24     // I know that the 7th port in the serial list on my Mac
25     // is always my Arduino, so I open Serial.list()[7].
26     // Open whatever port is the one you're using.
27     myPort = new Serial(this, Serial.list()[mySerialPort], 9600
28     // don't generate a serialEvent() unless you get a newline
29     myPort.bufferUntil('\n');

```

Obrnut primer?

- Moguće je u Arduino softveru pristupati R-u.
- Za te potrebe se koristi R server
(<http://www.rforge.net/Rserve/index.html>).
- Primer korišćenja R servera u Arduino programskom jeziku je dat na slici.
- Slika je preuzeta sa:
<https://www.r-bloggers.com/connecting-the-real-world-to-r-with-an-arduino/>, *Fair Use*.

Rezime

- Funkcije u ggplot2 paketu, ali i druge funkcije u R-u su korisne za pravilnu vizuelizaciju podataka odnosno prikaz rezultata.
- Koga zanima više o vizuelizaciji, preporučujem da pogleda:
 - <https://www.edwardtufte.com/tufte/> (profesor emeritus i naučnik i umetnik, njegov rad je posvećen vizuelizaciji i prezentaciji)
 - Su YS. It's easy to produce chartjunk using Microsoft® Excel 2007 but hard to make good graphs. *Computational Statistics & Data Analysis*. 2008 Jun 15;52(10):4594-601. (ima odličnih primera šta ne treba raditi generalno)
 - <http://kbroman.org/> (blog univerzitetskog profesora sa odličnim preporukama za prikaz grafika i vizuelizaciju)
 - <https://www.perceptualedge.com/> (o *Visual Business Intelligence*)
 - <https://www.amazon.com/Elements-Graphing-Data-William-Cleveland/dp/0963488414> (knjiga sa odličnim uputstvima o vizuelizaciji)
- Deo ovog predavanja inspirisan je predavanjem Tala Galilija sa linka: <https://www.youtube.com/watch?v=g80DzGuwSuY> (nije dostupan u 2021).
- U R-u postoji biblioteka za konekciju sa serijskim portom. Moguće ju je koristiti i za merenje signala.