



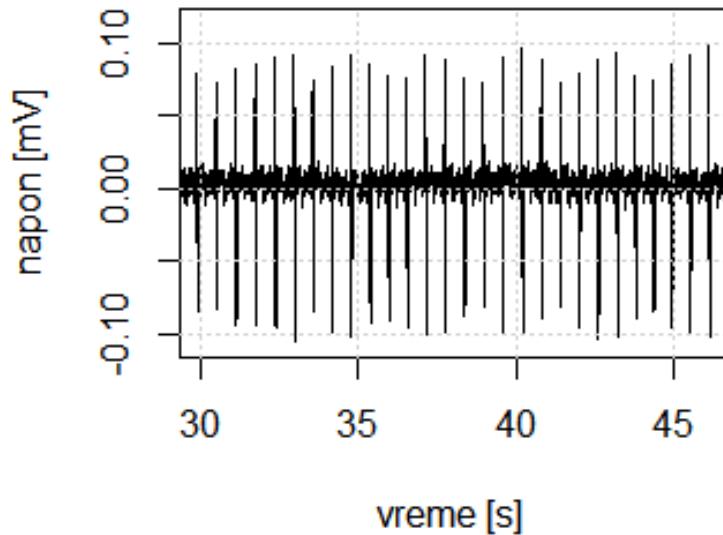
# Kontrolne funkcije, pretprocesiranje biomedicinskih signala i debugovanje

Dr Nadica Miljković, vanredni profesor  
kabinet 68, [nadica.miljkovic@etf.bg.ac.rs](mailto:nadica.miljkovic@etf.bg.ac.rs)

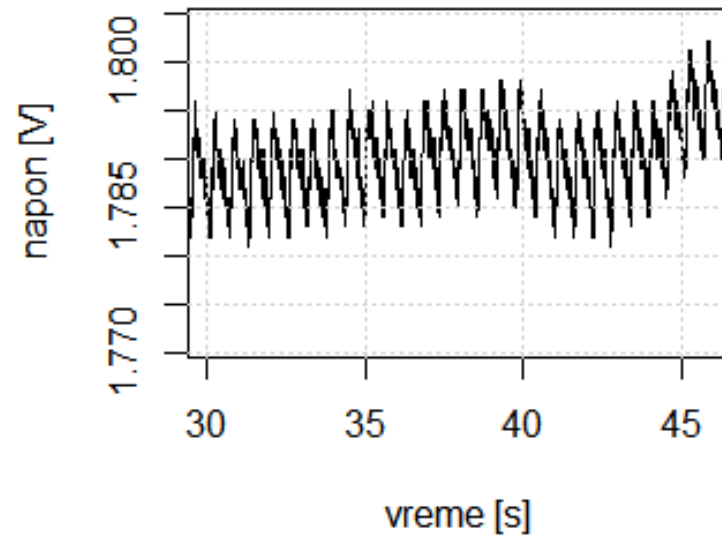


# EKG i IR

**EKG signal**



**IR signal**



- U ovoj vežbi se koriste signali merenja pulsa primenom EKG pojačavača i IR senzora.
- Signali su prikazani na slici.
- Za potrebe ove vežbe, signali su snimani u laboratoriji 69 na Elektrotehničkom fakultetu, Univerziteta u Beogradu 2017. godine i nalaze se u fajlu "puls.txt".

# Zadatak #1

- Učitati signale iz fajla.
- Pojačanje EKG signala (nalazi se u prvoj koloni) je bilo 5000, pa je potrebno ove odbirke pretvoriti u mV (snimljeni odbirci su izraženi u V).
- Formirati vremenski vektor (frekvencija odabiranja je bila podešena na 1000 odbiraka u sekundi) i prikazati oba signala u vremenskom domenu kao na prethodnom slajdu.
- Ograničiti prikaz signala od 30 s do 46 s i označiti sve ose i naslov grafika. Za prikaz grafika u istom prozoru koristiti funkciju *par(mfrow = c(1, 2))*.
- Za brisanje svega što postoji u konzoli koristi se prečica sa tastature CTRL+L.

# Rešenje: zadatak #1

```
# učitavanje i prikaz podataka
dat <- read.table("puls.txt")
head(dat)

kanal1 <- dat$V1 / 5 # zbog pojačanja, da bi signal bio u mV
kanal2 <- dat$V2

fs <- 1000
vreme <- seq(0, (length(dat$V1)/fs - 1/fs), by = 1 / fs)

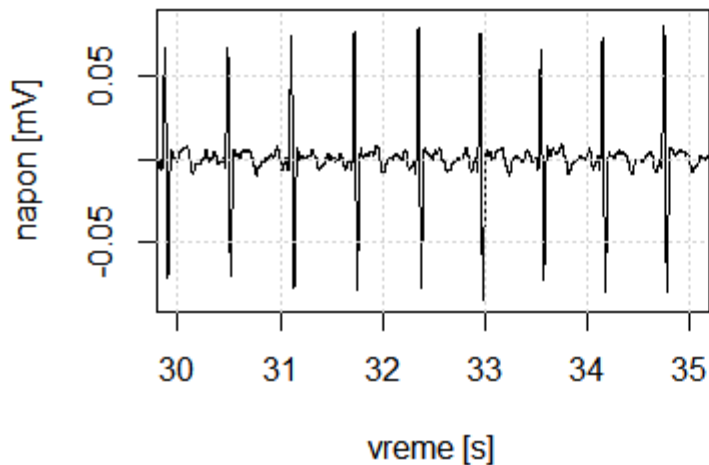
par(mfrow=c(1,2))
plot(vreme, kanal1, main = "EKG signal", xlab = "vreme [s]",
      ylab = "napon [mV]", type = "l", xlim = c(30, 46))
grid()
plot(vreme, kanal2, main = "IR signal", xlab = "vreme [s]",
      ylab = "napon [V]", type = "l", xlim = c(30, 46))
grid()
```

# Zadatak #2

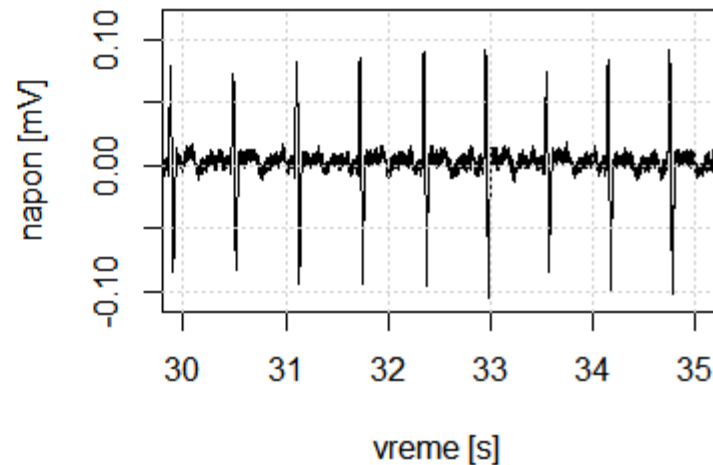
- Uočiti koji se šumovi javljaju.
- Filtrirati EKG signal *Butterworth*-ovim filtrima: 1) visokopropusnik 2. reda koji odseca frekvencije manje od 0.5 Hz i 2) niskopropusnik 3. reda čija je frekvencija odsecanja podešena na 30 Hz.
- Izmeriti (pomoću funkcije *system.time()*) *elapsed time* za filtriranje EKG signala. Koliko je vremena potrebno?
- Prikazati filtriran signal zajedno sa nefiltriranim pomoću *par()* funkcije.
  - Koje promene se uočavaju na signalu?

# Rešenje: zadatak #2

Filtriran EKG signal



EKG signal



```
library(signal)
f1 <- butter(2, 0.5/(fs/2), "high")
f2 <- butter(3, 30/(fs/2), "low")

# filtriranje EKG signala
kanal1f <- filtfilt(f1$b, f1$a, filtfilt(f2$b, f2$a, kanal1))
# izmeriti vreme pomoću system.time za filtriranje prvog signala
s1 <- system.time(filtfilt(f1$b, f1$a, filtfilt(f2$b, f2$a, kanal1)))
s1

par(mfrow=c(1,2))
plot(vreme, kanal1f, main = "Filtriran EKG signal", xlab = "vreme [s]",
      ylab = "napon [mV]", type = "l", xlim = c(30, 35))
grid()
plot(vreme, kanal1, main = "EKG signal", xlab = "vreme [s]",
      ylab = "napon [mV]", type = "l", xlim = c(30, 35))
grid()
```

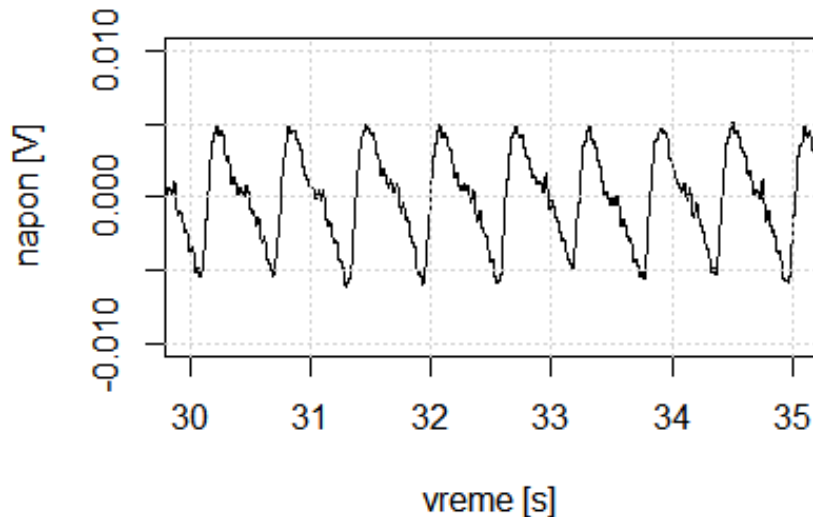
# Zadatak #3

- Istu metodu filtriranja (pretprocesiranja) primeniti na signale snimljene sa IR senzora.
- Prikazati filtiran i “sirov” (eng. *raw*) IR signal.
  - Koje promene se uočavaju na signalu?
- Pomoću funkcija *Rprof()* i *summaryRprof()* izmeriti *total time* potrebno za filtriranje ovog signala. Rezultat prikazati u konzoli.
  - Pre i posle pokretanja profajlera, proveriti koji se fajlovi nalaze u direktorijumu pozivom funkcije *dir()*.

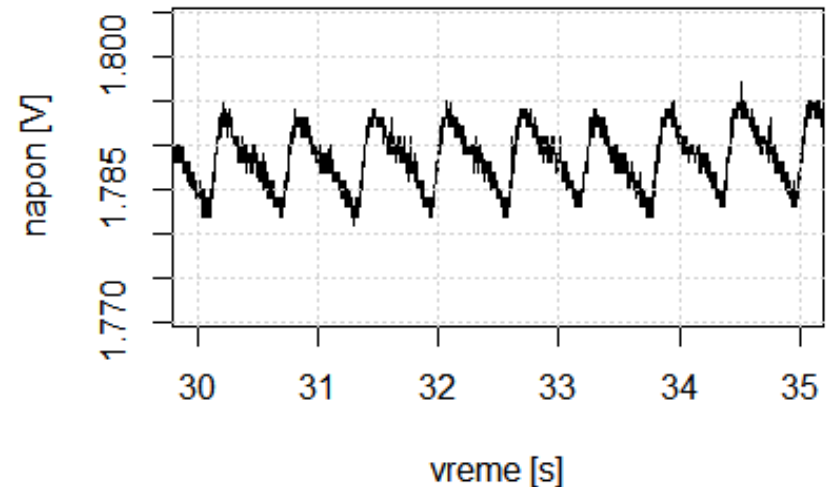


# Rešenje: zadatak #3

Filtriran IR signal



IR signal



```
# za filtriranje IR koriste se isti filtri kao za EKG
dir()
Rprof()
kanal2f <- filtfilt(f1$b, f1$a, filtfilt(f2$b, f2$a, kanal2))
Rprof(NULL)
summaryRprof()
dir()

par(mfrow=c(1,2))
plot(vreme, kanal2f, main = "Filtriran IR signal", xlab = "vreme [s]",
     ylab = "napon [V]", type = "l", xlim = c(30, 35), ylim = c(-0.01, 0.01))
grid()
plot(vreme, kanal2, main = "IR signal", xlab = "vreme [s]",
     ylab = "napon [V]", type = "l", xlim = c(30, 35))
grid()
```



# Zadatak #4

- Primenom *lapply()* funkcije na podatke učitane u ovoj vežbi izračunati:
  - srednje vrednosti svih kolona,
  - standardne devijacije svih kolona i
  - opsege svih kolona (koristiti funkciju *range()*).
- Rezultat prikazati u konzoli i proveriti kojoj klasi pripadaju.
- Potom, istu proceduru ponoviti primenom funkcije *sapply()*.
  - Koja je razlika između ove dve funkcije?

# Rešenje: zadatak #4

```
# lapply i sapply
datMean <- lapply(dat, mean)
datMean
datSd <- lapply(dat, sd)
datSd
datRange <- lapply(dat, range)
datRange
class(datRange)
as.character(datRange)
class(datRange)

datMean <- sapply(dat, mean)
datMean
datSd <- sapply(dat, sd)
datSd
datRange <- sapply(dat, range)
datRange
class(datRange)
```



# Zadatak #5

- Prikazati Furijeovu transformaciju EKG i IR signala.
- Primeniti *fft()* funkciju.
- DODATNI ZADACI: Izračunati sa prikaza FFT-a IR signala koliki je puls ispitanika i izraziti vrednost u jedinicama bpm (eng. *beats per minute*). Prikazati Furijeovu transformaciju filtriranih signala.

# Rešenje: zadatak #5

```
# Furijeova transformacija signala
sig1FFT <- fft(dat$V1)
sig2FFT <- fft(dat$V2)
class(sig1FFT)
sig1Mag <- Mod(sig1FFT)
sig1Mag <- sig1Mag[1:(length(sig1Mag)/2)]
sig2Mag <- Mod(sig2FFT)
sig2Mag <- sig2Mag[1:(length(sig2Mag)/2)]

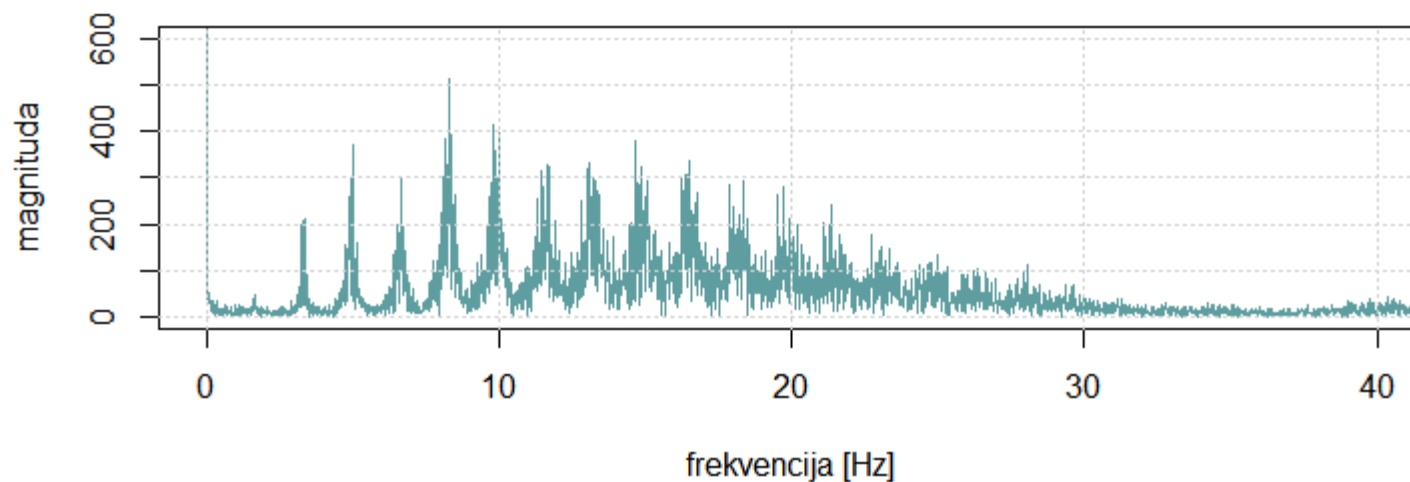
fosa <- 1:length(sig1Mag)/max(vreme)
```

```
# FFT EKG i IR signala
# x11() # za prikaz signala u novom prozoru
par(mfrow = c(1,1))
plot(fosa, sig1Mag, type = "l",
     ylim = c(0, 600), xlim = c(0, 40),
     xlab = "frekvencija [Hz]", ylab = "magnituda",
     main = "FFT EKG signala", col = "cadetblue")
grid(col = "lightgrey")

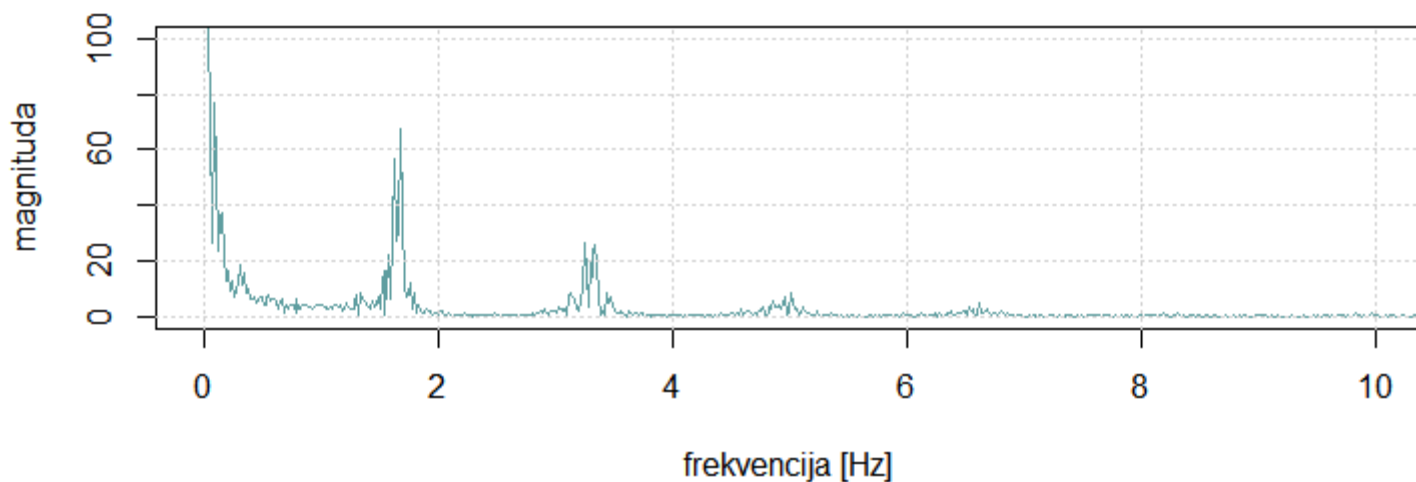
plot(fosa, sig2Mag, type = "l",
     ylim = c(0, 100), xlim = c(0, 10),
     xlab = "frekvencija [Hz]", ylab = "magnituda",
     main = "FFT IR signala", col = "cadetblue")
grid(col = "lightgrey")
```

# Rešenje: zadatak #5

**FFT EKG signala**

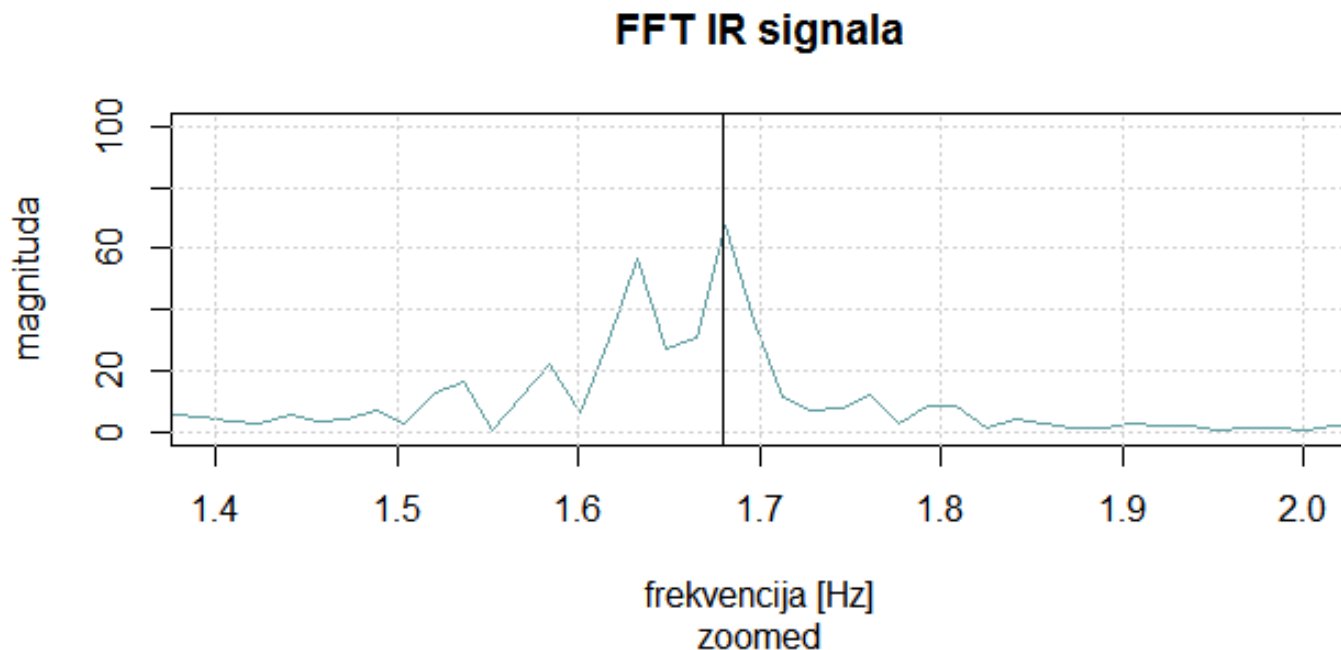


**FFT IR signala**





# Puls ispitanika?



```
> # puls?  
> plot(fosa, sig2Mag, type = "l",  
+       ylim = c(0, 100), xlim = c(1.4, 2),  
+       xlab = "frekvencija [Hz]", ylab = "magnituda",  
+       main = "FFT IR signala", col = "cadetblue", sub = "zoomed")  
> abline(v = 1.68)  
> grid(col = "lightgrey")  
>  
> # pulse IR  
> 60 / (1 / 1.68)  
[1] 100.8  
> |
```



# Funkcije i komande za danas

- `read.table()`
- `head()`
- `seq()`
- `par()`
- `plot()`
- `grid()`
- `butter()`
- `filtfilt()`
- `system.time()`
- `dir()`
- `Rprof()`
- `summaryRprof()`
- `lapply()`
- `sapply()`
- `as.character()`
- `class()`
- `fft()`
- `Mod()`
- `length()`
- `max()`
- `abline()`

# Do sledećeg časa ...



- Za domaći: Pokrenuti SWIRL i uraditi lekcije pod nazivom *Missing values* i *Matrices and Data Frames*.
  - NAPOMENA: Pre pokretanja *swirl()* komande potrebno je učitati i biblioteku/paket komandom *library(swirl)*.