

Univerzitet u Beogradu – Elektrotehnički fakultet (ETF)

Katedra za signale i sisteme



Tehnike obrade biomedicinskih signala 13M051TOBS

Dr Nadica Miljković, vanredna profesorka
Kabinet 68, nadica.miljkovic@etf.bg.ac.rs

Shiny

- *Shiny* je R paket / biblioteka se koristi za kreiranje korisničkih interaktivnih aplikacija. U obradi biomedicinskih, ali i drugih signala, *Shiny* omogućava dinamički način razmene i prikaza podataka („*The new black in data analysis*“).
- Detaljnije, interaktivni dokument tj. *Shiny* aplikacija omogućava pregledniji prikaz rezultata, ali i promenu rezultata u zavisnosti od parametara (dodatna i interaktivna EDA, eng. *Exploratory Data Analysis*).
- Ukratko, korišćenjem *Shiny* funkcija moguće je kreirati interaktivne veb aplikacije.
 - Napomena: *Shiny* aplikacije mogu imati *free web hosting* za ograničen broj aplikacija.
- U ovoj prezentaciji dat je samo kratak pregled mogućnosti ovog R paketa. Za više informacija posetite zvaničnu stranicu i *Tutorials* na: <https://shiny.posit.co/r/getstarted/shiny-basics/lesson1/index.html> (pristupljeno 26.05.2024) ili pogledajte knjigu: H. Wickham, *Mastering Shiny*: <https://mastering-shiny.org/index.html> (pristupljeno 26.05.2024.)

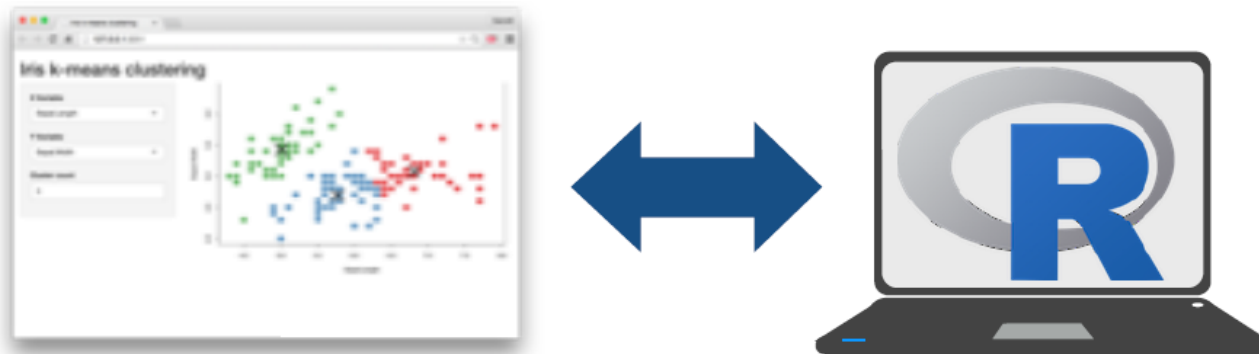
Shiny kolekcije

- Postoji kolekcija *Shiny* aplikacija koje se koriste u obrazovanju (Cal Poly University) na linku: <https://statistics.calpoly.edu/shiny> (pristupljeno 26.05.2024). Pogledajte neke od ovih aplikacija, posebno su zanimljive aplikacije koje demonstriraju primenu linearne regresije na podacima i promenu parametara.
- Druge kolekcije su dostupne preko *www*-a.

Modifikovana slika od Tim Mossholder na Unsplash.



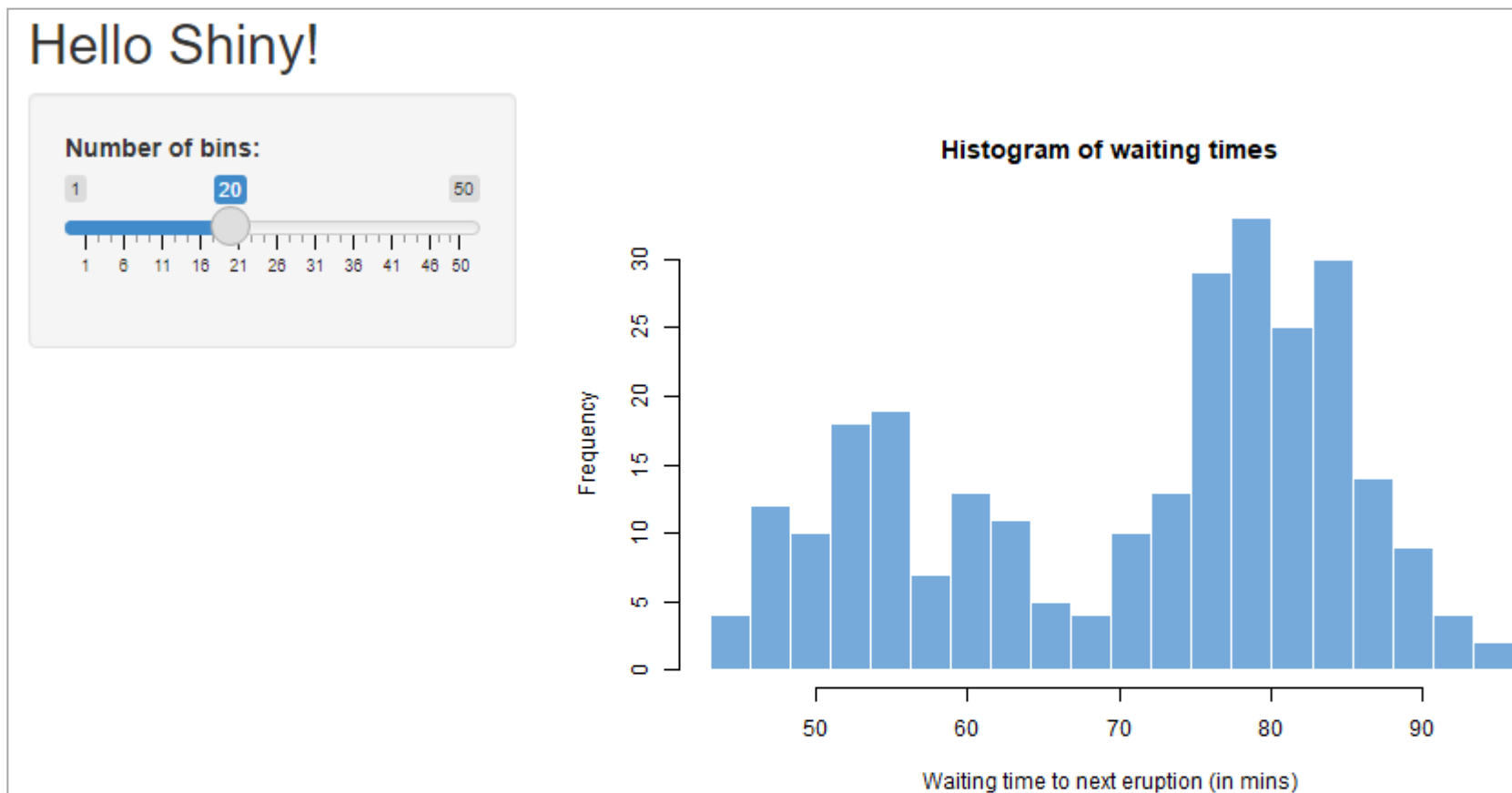
Shiny



Slika je preuzeta iz *Shiny Cheatsheet*-a: <https://shiny.posit.co/images/shiny-cheatsheet.pdf> (pristupljeno 2021. godine), *Fair Use*.

- *Shiny* aplikacija je veb stranica (korisnički interfejs) koja je povezana sa računarem koji može biti ili server ili personalni računar na kome je omogućeno pokretanje R sesije. Korisnički deo *Shiny* aplikacije je zapravo HTML dokument.
- Korisnici mogu manipulirati dostupnim funkcijama (*widgets*) i ta manipulacija posledično pokreće R kod na personalnom računaru ili serveru koji omogućava promenu na korisničkom interfejsu.
- Korisni linkovi za aktivaciju *Shiny* aplikacije:
 - Shiny Server: <https://github.com/rstudio/shiny-server> (pristupljeno 26.05.2024).
 - Shiny hosting na Rstudio / Posti sajtu: <https://www.shinyapps.io/> (pristupljeno 26.05.2024).

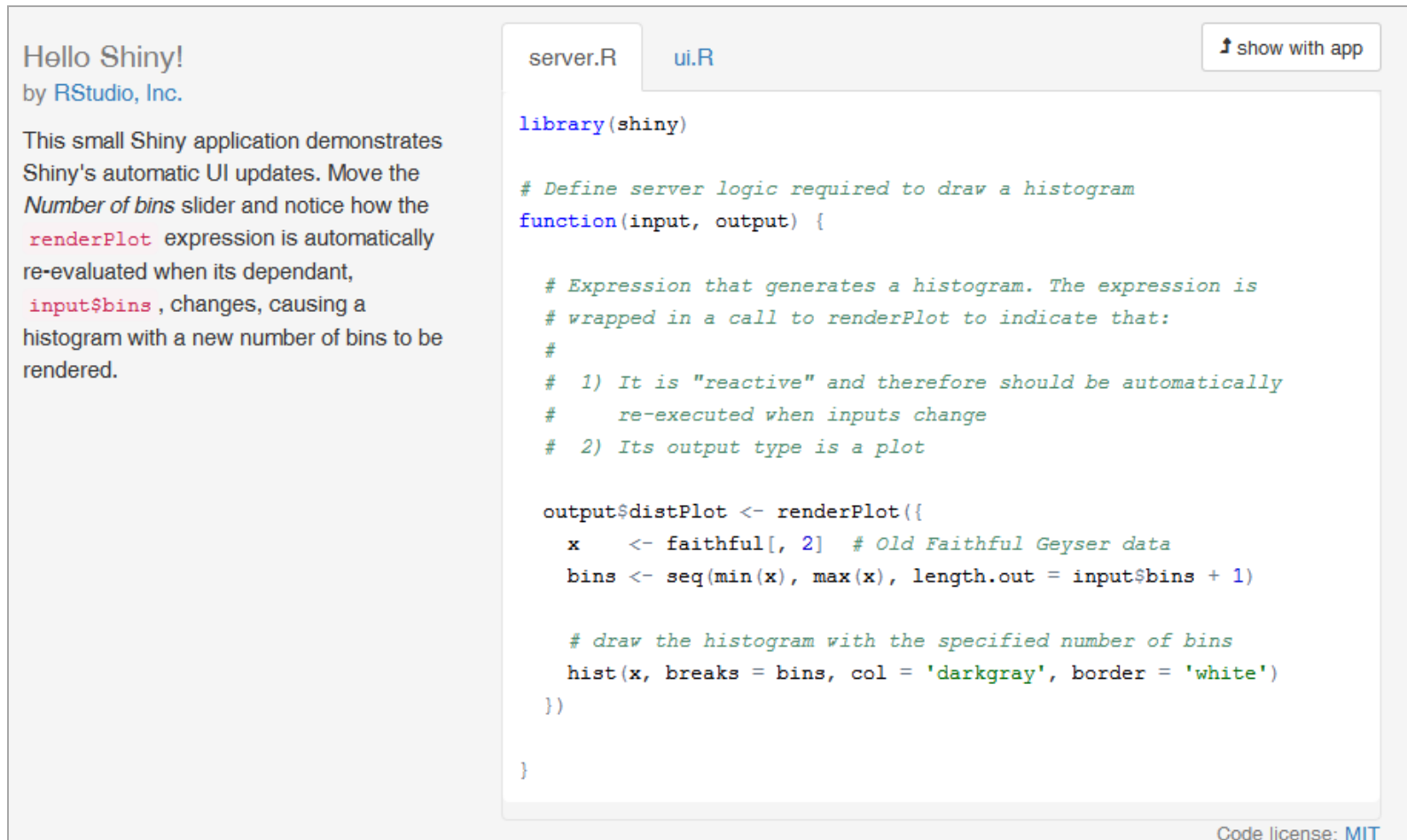
Jedan primer



- Aktiviranjem *Shiny* biblioteke (koja je prethodno instalirana u R-u) i pozivom funkcije `runExample("01_hello")`, otvara se prozor kao na slici.
- Koja je funkcija ovog programa?

Da testiramo rad ove aplikacije

Kod?



The screenshot shows a Shiny application window. On the left, there is a text area with the following content:

Hello Shiny!
by RStudio, Inc.

This small Shiny application demonstrates Shiny's automatic UI updates. Move the *Number of bins* slider and notice how the `renderPlot` expression is automatically re-evaluated when its dependant, `input$bins`, changes, causing a histogram with a new number of bins to be rendered.

On the right, there is a code editor showing the `server.R` file. The code is as follows:

```
server.R  ui.R  show with app

library(shiny)

# Define server logic required to draw a histogram
function(input, output) {

  # Expression that generates a histogram. The expression is
  # wrapped in a call to renderPlot to indicate that:
  #
  # 1) It is "reactive" and therefore should be automatically
  #    re-executed when inputs change
  # 2) Its output type is a plot

  output$distPlot <- renderPlot({
    x <- faithful[, 2] # Old Faithful Geyser data
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

    # draw the histogram with the specified number of bins
    hist(x, breaks = bins, col = 'darkgray', border = 'white')
  })
}
```

Code license: [MIT](#)

- Obratite pažnju da je kod za aplikaciju sa prethodnog slajda podeljen u dva .R dela: *server.R* i *ui.R*.
- Unutar *server.R* koda se nalazi funkcija `renderPlot()` koja za rezultat prikazuje na histogramu podatke *x* sa određenim brojem intervala (*bins*). Primetiti da se parametru *bins* pristupa preko `input$bins`.
- Primetiti da su ulazni parametri funkcije *server.R* `input` i `output`.

Nova verzija

This small Shiny application demonstrates Shiny's automatic UI updates.

Move the *Number of bins* slider and notice how the `renderPlot` expression is automatically re-evaluated when its dependant, `input$bins`, changes, causing a histogram with a new number of bins to be rendered.

app.R

↑ show with app

```
library(shiny)

# Define UI for app that draws a histogram ----
ui <- fluidPage(

  # App title ----
  titlePanel("Hello Shiny!"),

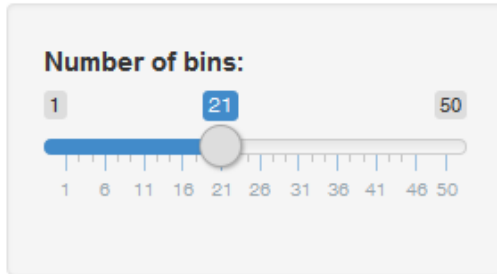
  # Sidebar layout with input and output definitions ----
  sidebarLayout(

    # Sidebar panel for inputs ----
    sidebarPanel(

      # Input: Slider for the number of bins ----
      sliderInput(inputId = "bins",
                  label = "Number of bins:",
                  min = 1,
                  max = 50,
                  value = 30)
```

Nema odvojenih delova ... već postoje funkcije.

Kod?



widget

```
server.R  ui.R  show with app

library(shiny)

# Define UI for application that draws a histogram
fluidPage(

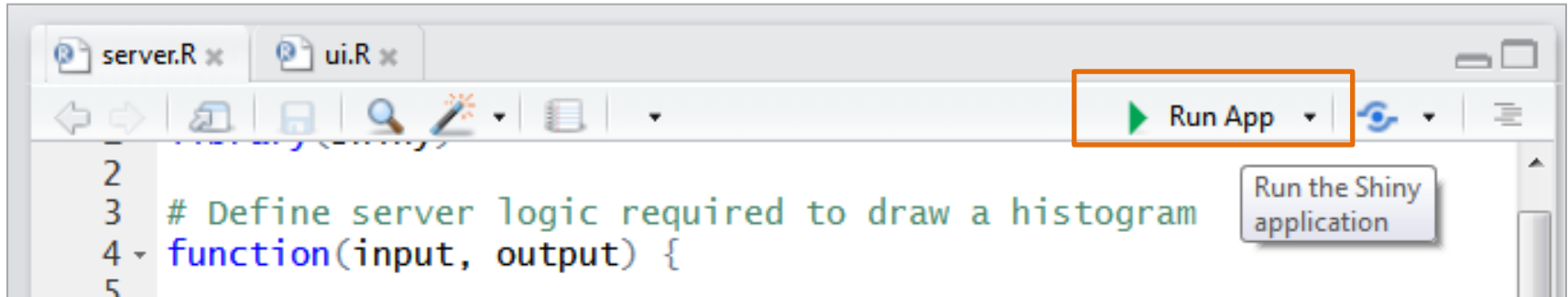
  # Application title
  titlePanel("Hello Shiny!"),

  # Sidebar with a slider input for the number of bins
  sidebarLayout(
    sidebarPanel(
      sliderInput("bins",
                  "Number of bins:",
                  min = 1,
                  max = 50,
                  value = 30)
    ),

    # Show a plot of the generated distribution
    mainPanel(
      plotOutput("distPlot")
    )
  )
)
```

- Vrednost promenljive `input$bins` korisnik unosi na interfejsu preko *widgets*-a.
- Datoteka/funkcija `server.R` sadrži kod koji upravlja ulaznim parametrima i prikazuje / menja izlaz na korisničkom interfejsu (u ovom slučaju to je prikaz histograma).
- Na slici je prikazan kod `ui.R`. Gde je definisano ime promenljive `bins`?

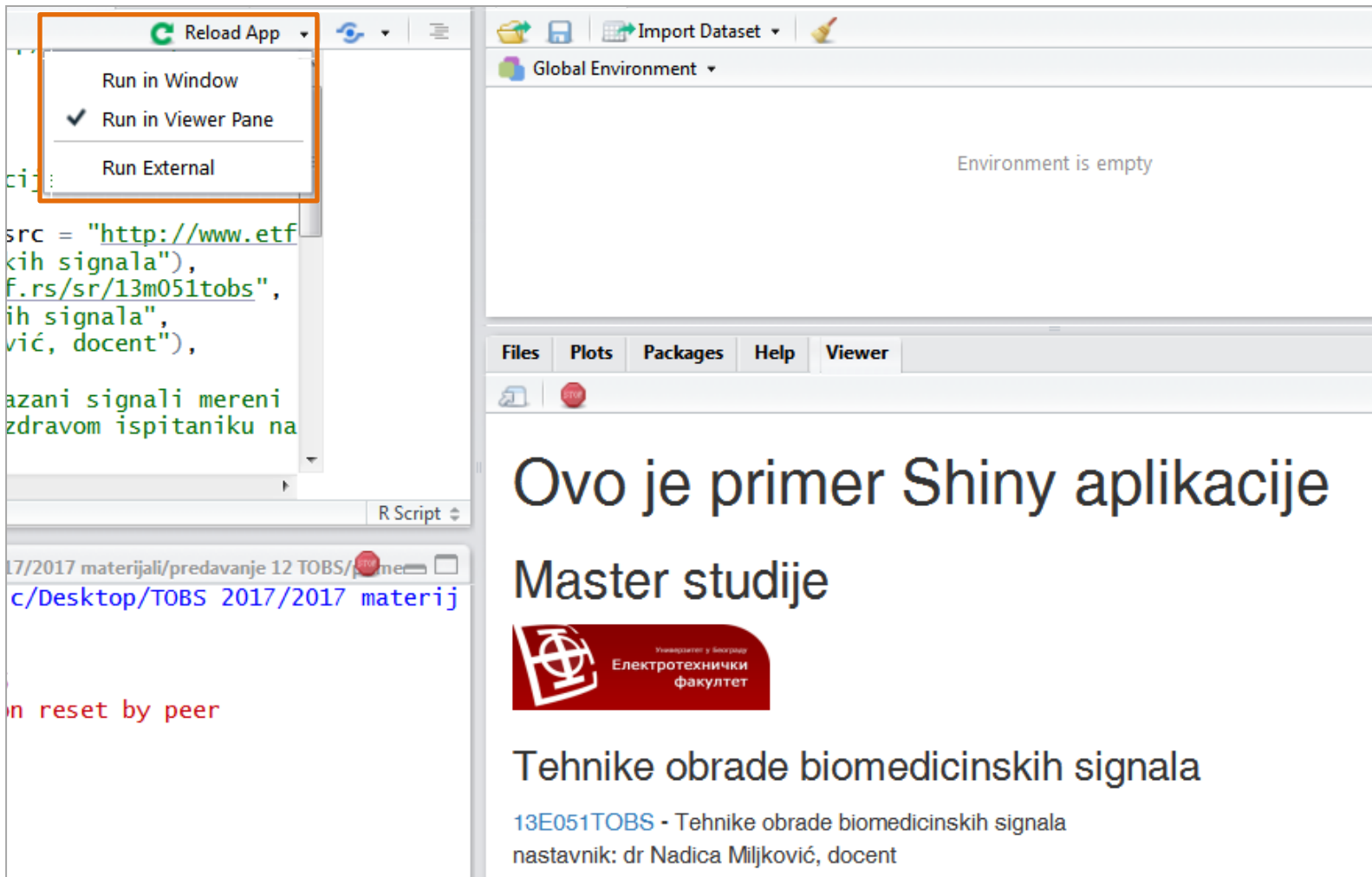
Struktura *Shiny* koda



```
2  
3 # Define server logic required to draw a histogram  
4 function(input, output) {  
5
```

- Postoje dve posebne funkcije (prikazana je ranija verzija sa dve datoteke).
- Ako postoje dve datoteke trebalo bi da se nalaze u istoj fascikli/folderu da bi se pokrenula *Shiny* aplikacija:
 - *server.R* (u ovom kodu definisan je način prikaza izlaznih parametara, odnosno parametara koji se prikazuju u korisničkom interfejsu, u prethodnom primeru definisan je prikaz histograma, odnosno u ovom delu se nalaze instrukcije za manipulaciju podacima) i
 - *ui.R* (u njemu su definisani svi ulazi i izlazi na ekranu tj. izgled korisničkog interfejsa, kao što su grafik i slajder za unos broja binova u ovom primeru)
- Za pokretanje *Shiny* aplikacije postoji opcija (na slici): *Run App*.

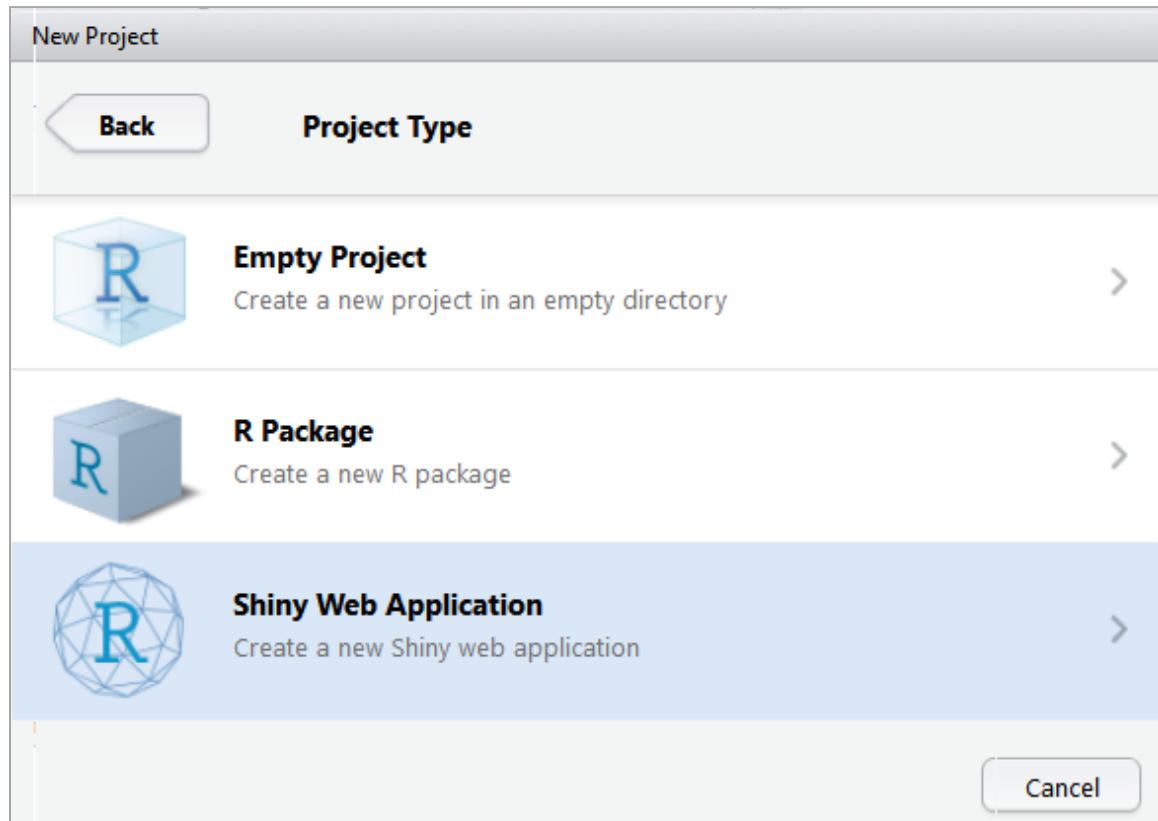
Pokretanje i testiranje na računaru



The screenshot displays the R Studio environment. On the left, the R Script editor shows code for a Shiny application. A dropdown menu is open under the 'Reload App' button, with three options: 'Run in Window', 'Run in Viewer Pane' (which is selected with a checkmark), and 'Run External'. The right pane shows the 'Global Environment' and the 'Viewer' pane. The Viewer pane displays the rendered Shiny application, which includes the text 'Ovo je primer Shiny aplikacije Master studije' and a logo for the Faculty of Electrical Engineering at the University of Belgrade. Below the logo, it says 'Tehnike obrade biomedicinskih signala' and '13E051TOBS - Tehnike obrade biomedicinskih signala nastavnik: dr Nadica Miljković, docent'.

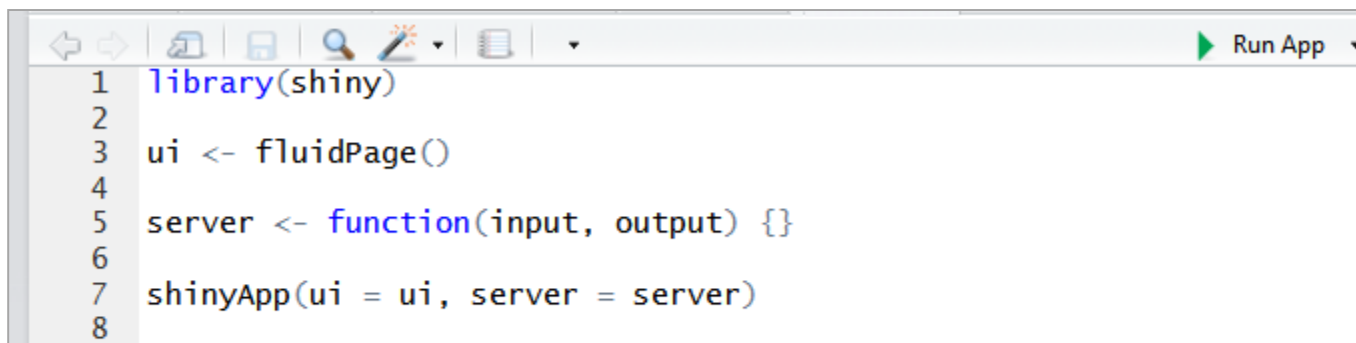
- Moguće je iz padajućeg menija sa slike odabrati opciju po izboru.
- Jedna od često korišćenih opcija je da se interaktivna *Shiny* aplikacija prikazuje unutar *Viewer* prozora (R Studio).

Kako početi?



- Ako se odaberu sledeće opcije iz padajućeg *File* menija: *New Project / New Directory / Shiny Web Application* otvoriće se ugrađen primer koji je prikazan ranije, odnosno otvoriće se *ui.R* i *server.R*. U novijoj verziji će se pojaviti jedinstvena datoteka.
- Potrebno je poznavanje odgovarajućih komandi koje omogućavaju funkcionalnost *Shiny* aplikacije. Na sledećim slajdovima će biti prikazane te funkcije.

Jedna ili dve datoteke?

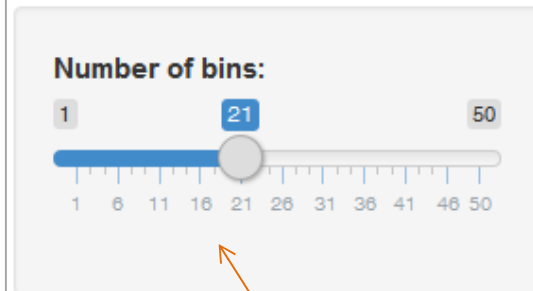


```
1 library(shiny)
2
3 ui <- fluidPage()
4
5 server <- function(input, output) {}
6
7 shinyApp(ui = ui, server = server)
8
```

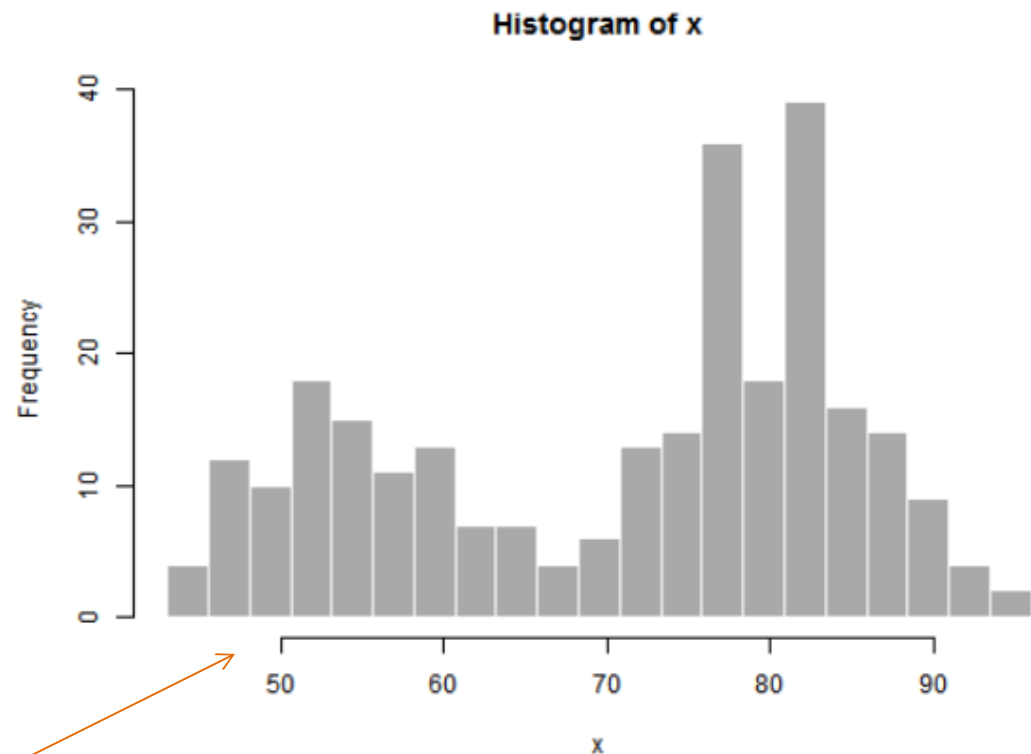
- Novije verzije *Shiny* aplikacija ne sadrže dve, već jednu datoteku. Sve aplikacije koje sadrže dve datoteke (*ui.R* i *server.R*) su i dalje funkcionalne i često u upotrebi.
- Ako postoji jedna datoteka, onda je sav kod smešten u njoj, ali odvojen u dve *ui()* i *server()* funkcije. Na kraju te datoteke potrebno je dodati deo koda *shinyApp(ui = ui, server = server)* za pokretanje *Shiny* aplikacije. (**ova linija koda uvek mora biti poslednja**)
- Na slici je prikazana “prazna” *Shiny* aplikacija smeštena u jednoj .R datoteci.
- NAPOMENA: Ove funkcije i kodovi se uvek moraju zvati istim imenima ako se radi sa dve datoteke i uvek moraju biti smeštene u istom folderu. Ako se radi sa jednom datotekom uvek nosi ime *app.R*. R automatski prepoznaje ime *app.R* i tada je omogućena opcija *Run App* u gornjem desnom uglu R Studio okruženja.

Interaktivni R *Shiny* objekti

Hello Shiny!



Widgets – omogućavaju korisnicima da unose vrednosti / odluke upisom teksta, klikom i sl.



Rendered – menjaju se u skladu sa promenama koje korisnik unosi preko *widgets*-a.

Primeri *widgets*?

The image displays a gallery of nine different Shiny widget types, arranged in a 3x3 grid. Each widget is presented in a light gray box with a title, a visual representation of the widget, a 'Current Value' field, and a 'See Code' button.

- Action button:** Shows a button labeled 'Action'. The current value is 0. The code is `shinyActionButtonValue`.
- Single checkbox:** Shows a checked checkbox labeled 'Choice A'. The current value is TRUE.
- Checkbox group:** Shows three checkboxes, with 'Choice 1' checked. The current value is '1'.
- Date input:** Shows a date input field with '2014-01-01'. The current value is '2014-01-01'.
- Date range:** Shows a date range input with '2017-05-10' to '2017-05-10'. The current value is '2017-05-10' '2017-05-10'.
- File input:** Shows a file input field with 'Browse...' and 'No file selected'. The current value is NULL.
- Numeric input:** Shows a numeric input field with '1'. The current value is 1.
- Radio buttons:** Shows three radio buttons, with 'Choice 1' selected. The current value is '1'.
- Select box:** Shows a select box with 'Choice 1' selected. The current value is '1'.

- Za interakciju sa korisnikom mogu se koristiti različiti *widgets*. To su ulazi korisničkog interfejsa.
- Oni omogućavaju korisniku da bira parametre koji se prikazuju na korisničkoj aplikaciji.
- Slika je preuzeta sa sajta: <https://shiny.posit.co/r/gallery/widgets/widget-gallery/>, *Fair Use*, (pristupljeno 26.05.2024).

Funkcije (*widgets*) u *ui.R*

```
sliderInput("bins",  
           "Number of bins:",  
           min = 1,  
           max = 50,  
           value = 30)
```

- U ovom delu koda, mogu se koristiti funkcije koje su definisane za svaki *widget* posebno:
 - ulazne funkcije koje u većini slučajeva imaju ključnu reč *input*:
 - *actionButton()*
 - *checkboxGroupIn()* – grupa *checkbox*-ova
 - *checkboxInput()* – jedan *checkbox*
 - *dateInput()* – kalendar koji omogućava unos datuma
 - *dateRangeInput()* – dva kalendara koji omogućavaju da se odabere vremenski opseg
 - *fileInput()* – ovom funkcijom se otvara *wizard* koji omogućava učitavanje iz datoteke
 - *helpText()* – tekst koji se dodaje ostalim *widgets*-ima sa uputstvima korisniku
 - *numericInput()* – polja u kojima je moguće uneti numeričke / brojne vrednosti
 - *radioButtons()* – grupa *radio buttons* komandi
 - *selectInput()* – skup opcija kako bi korisnik mogao da odabere odgovarajuće opcije
 - *sliderInput()* – *slide bar* (korišćen je u prikazanom primeru i pokazan na slici)
 - *submitButton()*
 - *textInput()* – polje u koje korisnik unosi tekst
 - Vrednosti dobijene *pomoću widgets*-a se najčešće nazivaju reaktivnim vrednostima.
 - VAŽNO: reaktivne vrednosti se koriste sa reaktivnim funkcijama. Biće detaljno objašnjene kasnije.

Funkcije u *ui.R* kodu

```
# Show a plot of the generated distribution
mainPanel(
  plotOutput("distPlot")
)
```

- U ovom delu koda mogu se, dalje, koristiti:
 - izlazne funkcije koje imaju ključnu reč *output*:
 - *plotOutput(outputID)* na primer: *plotOutput("grafik")* i koje se koriste da se označi naziv izlaza i funkcionalnosti (prikaz grafika) koja se dodeljuje tom izlazu. Za prethodni primer prikazano je korišćenje ove funkcije na slici na slici.
 - *imageOutput()* – služi za prikaz slike u aplikaciji
 - *htmlOutput()* – služi za prikaz HTML koda u aplikaciji
 - *tableOutput()* – služi za prikaz tabele u aplikaciji
 - *textOutput()* – služi za prikaz teksta u aplikaciji
 - *uiOutput()* – služi za dodavanje *Shiny UI* elemenata
 - *verbatimText()* – služi za prikaz teksta u aplikaciji

server.R funkcije

```
output$distPlot <- renderPlot({  
  x <- faithful[, 2] # Old Faithful Geyser data  
  bins <- seq(min(x), max(x), length.out = input$bins + 1)  
  
  # draw the histogram with the specified number of bins  
  hist(x, breaks = bins, col = 'darkgray', border = 'white')  
})
```

- Najčešće se u *server.R* kodu nalazi neka iz familije funkcija *render* koje koriste *widgets* (*slide bar* u prethodnom primeru) i uloga im je da ažuriraju izlaz (histogram u prethodnom primeru).
- Ova funkcija (misli se na *render* familiju funkcija) nema ulazne parametre već ulazni kod između oznaka `{}`. Na slici je prikazana *renderPlot()* funkcija iz prethodnog primera.
- Postoje i druge *render* funkcije:
 - *renderDataTable()* – interaktivna tabela (npr. iz *data frame*-a)
 - *renderImage()* – slika koja se snimljena kao link ka originalnoj datoteci
 - *renderPrint()*
 - *renderText()*
 - *renderPlot()*
 - *renderTable()*
 - *renderUI()*
- Ove funkcije se koriste za prikaz izlaza u *Shiny* aplikaciji.

Pravila povezivanja ulaza i izlaza

Reactivity 101

- *Reactivity* 101: znači da kada se vrednost jedne promenljive promeni (*bins*) onda se menjaju svi rezultati funkcija/izraza koji koriste tu promenljivu (*histogram plot*). Ta promenljiva se naziva reaktivna promenljiva. Pisanje ovakvog koda se naziva reaktivno programiranje.
- Direktna posledica reaktivnosti je što se standardne R funkcije ne mogu koristiti sa vrednostima koje su učitane pomoću *widgets* (u ovom primeru sa *bins*).
 - Na primer, nije moguće pozvati funkciju *class(input\$bins)* u *Shiny* kodu.
- Da bi se koristila ulazna vrednost *Shiny* aplikacije koja je dobijena iz *widgets*, moraju se koristiti posebne familije funkcija.
- Jedna od tih familija je ***render*** familija. Postoje i ***observe***, ***reactive*** i ***isolate*** familije funkcija.

Familije funkcija

- *Isolate* familija funkcija omogućava da se izlazni objekat ne menja kada se promeni ulazni parametar *widget*-a.
 - Na primer `renderText(isolate{ # neki kod})`.
 - Ove funkcije omogućavaju da *Shiny* reaktivne funkcije postanu klasične R funkcije.
- *Observe* funkcije omogućavaju ponovno pokretanje R linija koda svaki put kada se promeni vrednost preko *widget*-a.
- Dodatno, `eventReactive()` funkcija iz familije *reactive*, omogućava da se promena preko *widget*-a izvrši u trenutku kada korisnik to želi (npr. dugme “iscrtaj”).

HTML

- Izgled korisničkog interfejsa se zasniva na HTML (eng. *Hypertext Markup Language*) kodu.
- HTML se u korisničkom interfejsu koristi za statički izgled tj. estetiku aplikacije.
- Ko zna HTML može i njega da koristi, a ko ne zna može koristiti odgovarajuće *Shiny* R funkcije.
- **Ovaj deo koda se nalazi u *ui.R* datoteci.**

Osnovne HTML funkcije: Statički sadržaj

- Na primer, *h1()*, *h2()*, ..., *h(6)* se koriste za definisanje prvog do 6. nivoa naslova i podnaslova (*h* od eng. *header*).
- *a()* funkcija se koristi za definisanje hiperlinkova u *Shiny* aplikaciji.
- Tekst se može pisati slobodno unutar *fluidPage()* funkcije.
- Funkcija *p()* se koristi za novi paragraf.
- Prelazak u novi red se postiže funkcijom *br()*.
- Funkcija *em()* se koristi za tekst koji je ispisan kurzivom.
- Za podebljana slova (eng. *bold*) koristi se funkcija *strong()*.
- Horizontalna linija (tzv. lenjir) se može dodati pomoću funkcije *hr()*.
- Dodavanje slika se može vršiti pomoću funkcije *img()*. Ako se dodaje slika sa računara ona mora biti sačuvana u fascikli / folderu unutar *Current Folder*-a koji nosi naziv "www".
- NAPOMENA: Primititi da je na kraju poziva svih ovih funkcija potrebno pisati zarez " ;".
- Ko zna HTML, neka piše samo HTML kod unutar *HTML(' # neki kod ')* funkcije u *fluidPage()* delu *Shiny* koda.
 - Sve funkcije na ovom slajdu uključujući i npr. *h1()* odgovaraju HTML *tag* funkcijama kao npr. `<h1></h1>`.

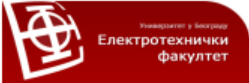
Primer

```
library(shiny)

ui <- fluidPage(
  h1("Ovo je primer Shiny aplikacije"),
  h2("Master studije"),
  img(height = 50, width = 150,
       src = "http://www.etf.bg.ac.rs/templates/etf2/images/logo.png"),
  h3("Tehnike obrade biomedicinskih signala"),
  a(href = "http://automatika.etf.rs/sr/13m051tobs", "13E051TOBS"),
  p("- Tehnike obrade biomedicinskih signala",
    p("nastavnik: dr Nadica Miljković, docent"),
    hr(),
    p("U ovom primeru će biti prikazani signali mereni na mišiću",
      em("Trapezius"), "(lat.) na zdravom ispitaniku na Elektrotehničkom
      fakultetu u Beogradu."),
    hr(),
```

Ovo je primer Shiny aplikacije

Master studije



Tehnike obrade biomedicinskih signala

[13E051TOBS](#) - Tehnike obrade biomedicinskih signala
nastavnik: dr Nadica Miljković, docent

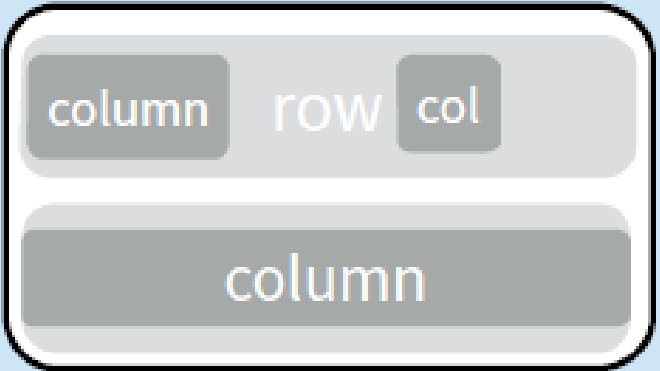
U ovom primeru će biti prikazani signali mereni na mišiću *Trapezius* (lat.) na zdravom ispitaniku na Elektrotehničkom fakultetu u Beogradu.

Skica interfejsa 1

- Prvi korak u kreiranju ovih aplikacija je da se napravi skica korisničkog interfejsa, odnosno interaktivnog izveštaja.
- Podela po panelima je uobičajena kod kreiranja veb aplikacija. Koriste se funkcije za svaki panel ponaosob i unutar panela se postavljaju željeni ulazi i izlazi.
- Neke od funkcija koje mogu biti od koristi su:
 - *fluidRow()* – za deljenje panela po vrstama
 - *column(width, offset)* – za definisanje širine i pomeraja udesno kod kreiranja kolona

Primer korišćenja *fluidRow()* i *column()* funkcija

fluidRow()



```
ui <- fluidPage(  
  fluidRow(column(width = 4),  
           column(width = 2, offset = 3)),  
  fluidRow(column(width = 12))  
)
```

Slika je preuzeta iz: <https://shiny.posit.co/images/shiny-cheatsheet.pdf>, pristupljeno 26.05.2024, *Fair Use*.

Maksimalna širina kolone je 12.

Skica interfejsa 2

- Postoje i *panel* familija funkcija koje se koriste za grupisanje objekata na ekranu korisničkog interfejsa.
- Primeri tih funkcija su:
 - *wellPanel()* – elementi koji se nalaze unutar ove funkcije se grupišu u odvojenom panelu (sive boje)
 - *absolutePanel()* – kreira se panel sa apsolutnim pozicijama
 - *headerPanel()* – kreira se panel koji sadrži naslov
 - *tabPanel()* – kreiraju se paneli u slojevima (često je potrebno ako se prikazuje npr. relativno veliki broj signala unutar jedne aplikacije)
 - *tabsetPanel()* – unutar ove funkcije se nalazi više *tabPanel()*-a radi grupisanja po slojevima
 - *navlistPanel()* – slično kao *tabsetPanel()* samo se razlikuje po grupisanju na interfejsu
 - *inputPanel()* – odnosi se na poseban raspored (*flowLayout*) sa grupisanjem bojama i ivicama sive boje
 - *mainPanel()* – odnosi se na poseban raspored (*sidebarLayout*)
 - *titlePanel()* – kreira se naslov za korisničku aplikaciju
 - ...
- Proučiti *Help* stranice za ove funkcije kada krenete u realizaciju *Shiny* aplikacije. Za više informacija, postoji i članak o različitim rasporedima ulaza i izlaza na korisničkom interfejsu na linku: <https://shiny.posit.co/r/articles/build/layout-guide/>, pristupljeno 26.05.2024.

Primer panela 1

```
library(shiny)

ui <- fluidPage(
  titlePanel("Primer Shiny aplikacije"),
  sidebarLayout(
    sidebarPanel(
      "DEFINISANJE PARAMETARA ZA PRIKAZ SIGNALA",
      textInput(inputId = "naslov",
                label = "Unesite željeni naslov",
                value = "Beli šum")
    ),
    mainPanel("PRIKAZ SIGNALA",
              plotOutput(outputId = "plt"))
  )
)

server <- function(input, output) {
  output$plt <- renderPlot(
    plot(rnorm(1000),
         main = input$naslov, type = "l")
  )
}

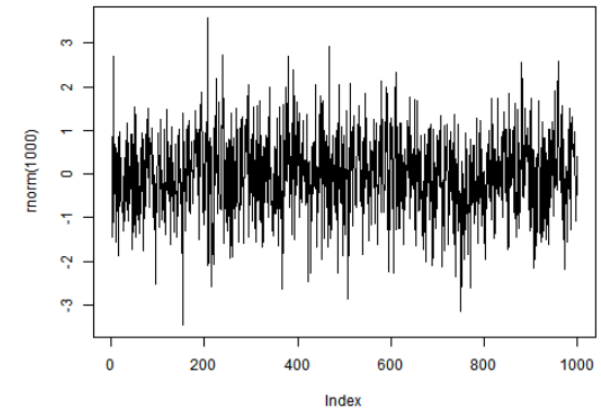
shinyApp(ui = ui, server = server)
```

Primer Shiny aplikacije

DEFINISANJE PARAMETARA
ZA PRIKAZ SIGNALA
Unesite željeni naslov

PRIKAZ SIGNALA

Beli šum



- *SidebarLayout* je ugrađen obrazac i vrlo često se koristi.
- Ovaj obrazac ima ugrađenu podelu ekrana u dve kolone.

Primer panela 2

```
library(shiny)

ui <- fluidPage(
  titlePanel("Primer Shiny aplikacije"),
  navbarPage(
    title = "Primer navbar panela",
    tabPanel("ULAZNI PANEL",
      numericInput(inputId = "br",
        label = "Unesite broj
        generisanih odbiraka",
        value = 10)
    ),
    tabPanel("IZLAZNI PANEL",
      plotOutput(outputId = "plt")
    )
  )
)

server <- function(input, output) {
  output$plt <- renderPlot(
    plot(rnorm(input$br),
      main = "Normalna raspodela")
  )
}

shinyApp(ui = ui, server = server)
```

Primer Shiny aplikacije

Primer navbar panela ULAZNI PANEL IZLAZNI PANEL

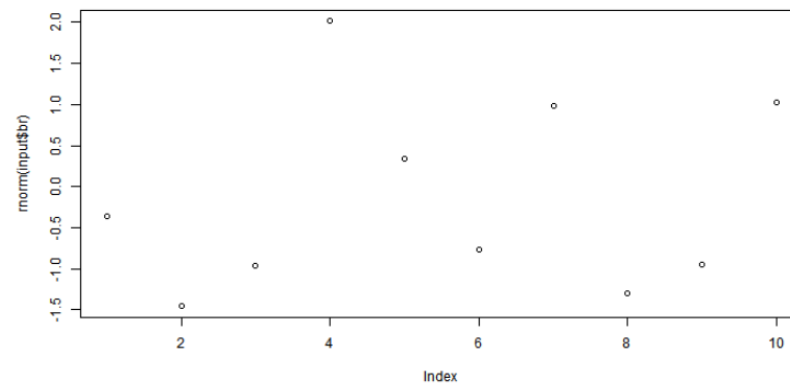
Unesite broj generisanih odbiraka

10

Primer Shiny aplikacije

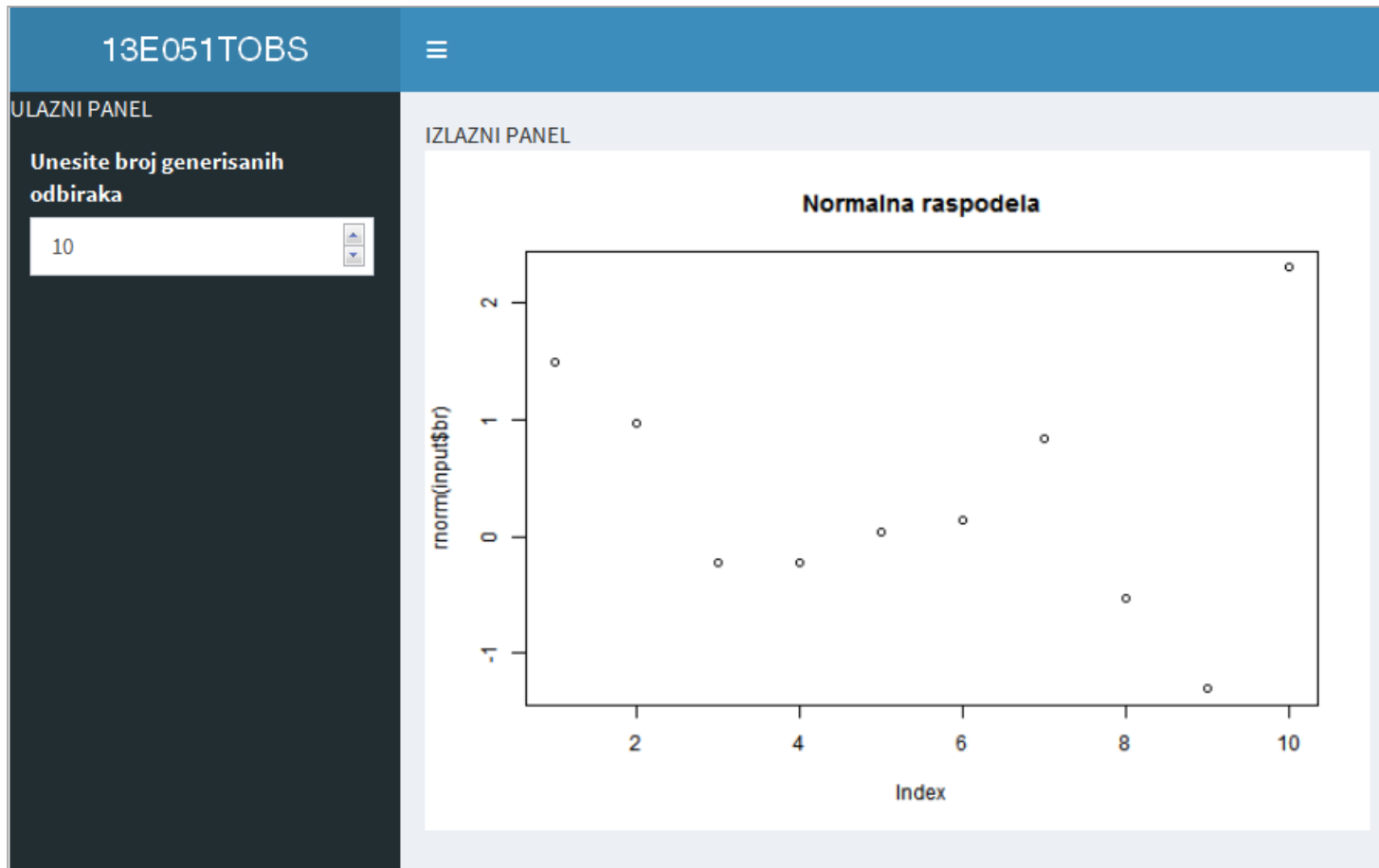
Primer navbar panela ULAZNI PANEL IZLAZNI PANEL

Normalna raspodela



- Na slici je prikazan *navbar* panel.
- Najčešće se koristi kada nema dovoljno mesta da se u jednom prozoru prikažu svi rezultati ili za izdavanje pojedinih sekcija.
- Ako se unutar funkcije *navbarMenu()* uključi više *tab* panela onda će se oni prikazati u formi padajućeg menija.

Primer panela 3



- Postoji poseban paket "shinydashboard".
- Primenom ovog paketa, moguće je realizovati korisnički interfejs kao na slici.

Panel 3, kod

```
library(shiny)
library(shinydashboard)

ui <- fluidPage(
  dashboardPage(
    title = "Primer korišćenja dashboard paketa",
    dashboardHeader(title = "13E051TOBS"),
    dashboardSidebar("ULAZNI PANEL",
      numericInput(inputId = "br",
        label = "Unesite broj
        generisanih odbiraka",
        value = 10)
    ),
    dashboardBody("IZLAZNI PANEL",
      plotOutput(outputId = "plt"))
  )
)

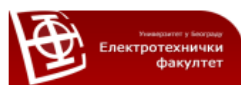
server <- function(input, output) {
  output$plt <- renderPlot(
    plot(rnorm(input$br),
      main = "Normalna raspodela")
  )
}

shinyApp(ui = ui, server = server)
```


Jedan primer

Ovo je primer Shiny aplikacije

Master studije, Elektrotehnički fakultet u Beogradu



Tehnike obrade biomedicinskih signala

13E051TOBS - Tehnike obrade biomedicinskih signala
nastavnik: dr Nadica Miljković, docent

U ovom primeru će biti prikazani signali mereni na mišiću *Trapezius* (lat.) na zdravom ispitaniku na Elektrotehničkom fakultetu u Beogradu.

Učitajte signal

Browse... No file selected

1

Crtaj

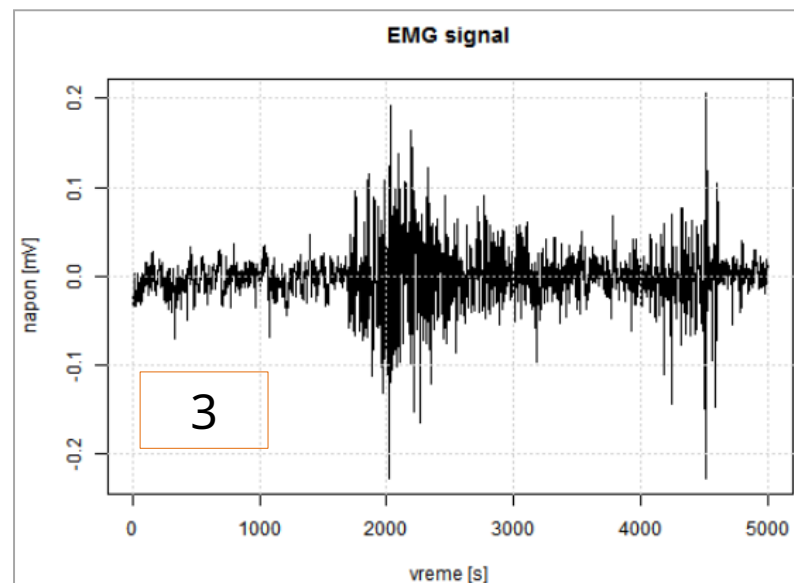
Učitajte signal

Browse... emg2.txt

Upload complete

Crtaj

2



- Korisnička aplikacija sa slike omogućava da se učita signal iz datoteke po izboru.
- Potom, korisnik pritiskom na dugme "Crtaj" aktivira prikaz učitanoog signala u vremenskom domenu.

Kod (*ui*)?

```
library(shiny)

ui <- fluidPage(
  h1("Ovo je primer Shiny aplikacije"),
  h2("Master studije, Elektrotehnički fakultet u Beogradu"),
  img(height = 50, width = 150,
       src = "http://www.etf.bg.ac.rs/templates/etf2/images/logo.png"),
  h3("Tehnike obrade biomedicinskih signala"),
  a(href = "http://automatika.etf.rs/sr/13m051tobs", "13E051TOBS"),
  "- Tehnike obrade biomedicinskih signala",
  p("nastavnik: dr Nadica Miljković, docent"),
  hr(),
  p("U ovom primeru će biti prikazani signali mereni na mišiću",
     em("Trapezius"), "(lat.) na zdravom ispitaniku na Elektrotehničkom
     fakultetu u Beogradu."),
  hr(),

  fluidRow(column(4,
                 fileInput(inputId = "file", label = "Učitajte signal")
                ),
           fluidRow(column(2, actionButton(inputId = "akcija",
                                           label = "Crtaj")
                    ),
                    fluidRow(column(8, plotOutput("plt"))))
  )
)
```

Kod (*server*)?

```
server <- function(input, output) {  
  data <- eventReactive(input$akcija,  
                        {inFile <- input$file  
                        dat <- read.table(inFile$datapath, dec = ",")  
                        dat$V1}  
  )  
  output$plt <- renderPlot(  
    {fs <- 1000  
    vreme = seq(seq(0, length(data())/fs - 1/fs, by = 1/fs))  
    plot(vreme, data(), type = "l",  
         main = "EMG signal",  
         xlab = "vreme [s]",  
         ylab = "napon [mV]")  
    grid()  
  }  
  )  
}  
  
shinyApp(ui = ui, server = server)
```

Još primera

Emergency Department Simulation

Home

Customize

Statistics

Patient Data

Plots

Case Study

Future Work

About Us

Help

Welcome to the Online ED Simulator!

What We Do

Overview

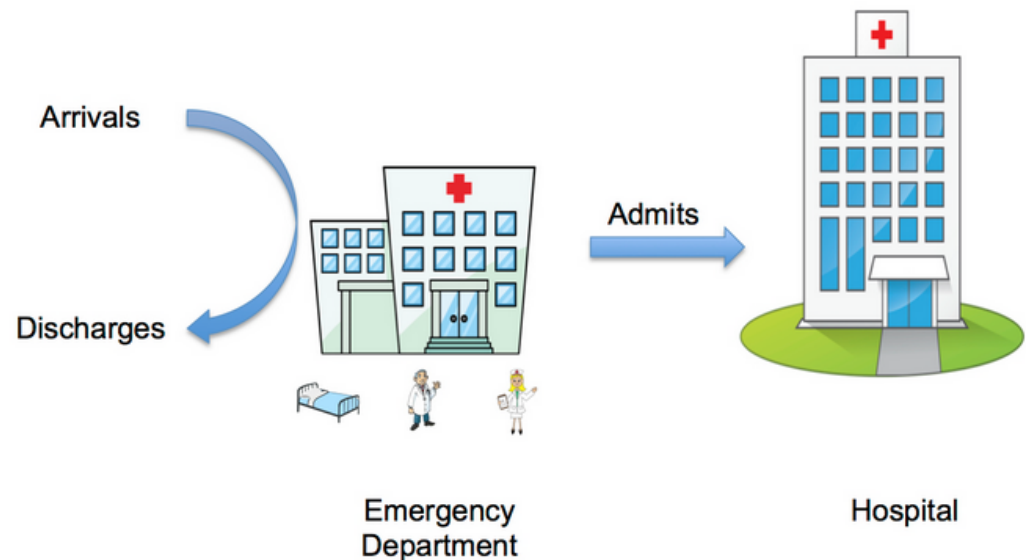
Get Started

The ED Simulator is a comprehensive simulation tool to quantify and predict site-specific emergency department crowding. Designed by applied mathematicians, statisticians, and emergency department physicians, the ED Simulator can accurately model patient flow dynamics in a wide variety of user-specified ED settings. Our objective is to collaborate with ED physicians and managers to design custom simulation tools to improve patient throughput.

What's Here

In this online version, we put the power and flexibility of our model in your hands. Our online version includes:

- Intuitive and accesible interfaces
- Highly customizable ED environments
- Traditional throughput metrics
- Advanced utilization statistics

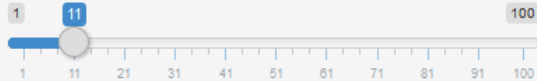


- Ovaj primer se nalazi na sajtu: <https://gallery.shinyapps.io/Edsimulation/>, pristupljeno 26.05.2024, *Fair Use*.
- Koje elemente do sada predstavljenih *Shiny* aplikacija prepoznajete?

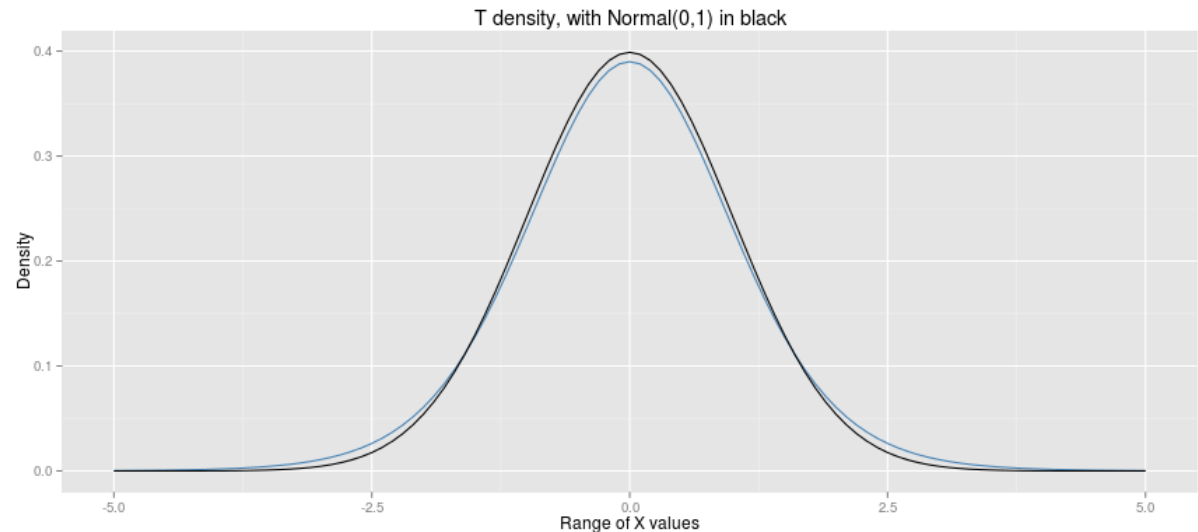
Još primera

Interactive Student's T Density

Move the slider to change the degrees of freedom



Though the t with one degree of freedom looks different from the $\text{Normal}(0,1)$, as the degrees of freedom increase, it becomes very close to a standard Normal



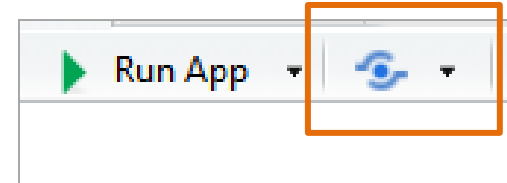
In all circumstances, the tails of the T distribution are "fatter," which means outliers are more likely under the T distribution.

- Ovaj primer se nalazi na sajtu: <https://gallery.shinyapps.io/tdist/>, pristupljeno 26.05.2024, *Fair Use*.
- Koje elemente do sada predstavljenih *Shiny* aplikacija prepoznajete?

Preporuke za *Shiny* aplikacije

- Smanjiti ponavljanje koda: ako se aplikacija bude koristila preko *www*, može doći do sporijeg odziva u slučaju više korisnika.
- Kod koji je izvan *server* dela će se izvršiti jednom po sesiji, ali kod koji je unutar *server* dela će se izvršavati onoliko puta koliko konekcija (korisnika aplikacije) ima.
- Kod unutar reaktivne funkcije će biti pokrenut svaki put prilikom rekacije, odnosno “veliki” broj puta.
- ZAKLJUČAK: sprečiti ponavljanje koda i smanjiti kod unutar reaktivne funkcije.
- Budete kreativni. Kombinujte elemente koje ste do sada naučili. Na primer “*plotly*” i “*shiny*” pakete.

Host Shiny aplikacija



- Najjednostavniji način za deljenje (eng. *sharing*) Shiny aplikacija je da se ona *host*-uje na: <https://www.shinyapps.io/> (pristupljeno 26.05.2024).
- Da bi ste to uradili, potrebno je da otvorite nalog na ovom sajtu.
- Potom je potrebno da iskoristite *Publish* ikonicu u R Studio interfejsu (na slici).
- Za kraj, možete i da realizujete i sami svoj server ili da ga kupite.
- Za više detalja, pogledajte: <https://shiny.posit.co/r/getstarted/shiny-basics/lesson7/> (pristupljeno 26.05.2024).

Rezime i korisna literatura

- “Shiny” je R paket koji se koristi za razvoj interaktivnih *web* aplikacija u R-u.
- Literatura:
 - Luyts M., Sichien J. Tutorial in R Shiny package: Developing Web Applications in the area of Biostatistics & Data Science, Interuniversity Institute for Biostatistics and statistical Bioinformatics (I-BioStat), Katholieke Universiteit Leuven, Belgium, 2016,
<https://ibiostat.be/seminar/uploads/introdcution-r-shiny-package-20160330.pdf>
(pristupljeno 26.05.2024).
 - H. Wickham, Mastering Shiny, 2021,
<https://mastering-shiny.org/index.html> (pristupljeno 26.05.2024).
- VAŽNO: *Shiny* interaktivne funkcije i *widgets* se mogu koristiti i u kombinaciji sa *.Rmd* dokumentom.

Join the team



- PSSOH konferencija na ETF-u, <https://pssoh.etf.bg.ac.rs/>
- R-Ladies (Global & local = Belgrade), <https://rladies.org/>
- Lista R konfrencija, <https://www.r-project.org/conferences/>
- I druga mesta gde možete da se povežete!
- Srećno!!!

Sloboda ili ne?



Slika: https://fsfe.org/contribute/promopics/ilovefs-sticker_thumb.png, pristupljeno 26.05.2024, *Fair Use*.

- Nadam se da niste razvili zavisnost ni za jedan programski jezik ili paket, ali da jeste za analizu, biomedicinsko inženjerstvo i programiranje.
- I da ćete sada moći i želeti da učite dalje.