**PAKET SHINY V RSTUDIU**

**Shiny package in RStudio**

**Ana Drenik**

**ana.drenik@student.fmf.uni-lj.si**

**Povzetek**

Paket Shiny uporabljamo za dokaj enostaven izris dinamičnih funkcij na spletu, brez znanja o programiranju v Javi, CSS-ju ali HTML-ju, saj lahko to naredimo direktno iz RStudia.

RStudio je okolje programa R (IDE – integrated development environment) in je bolj prijazen do uporabnika kot R. Namenjen je predvsem statistični obdelavi podatkov, saj ima vgrajenih veliko možnosti (ukazov) za izrisovanje raznih grafov, diagramov... Uporablja ga lahko vsak, ki ima dostop do interneta, saj je prosto dostopen na: <http://www.rstudio.com/>.

**Ključne besede**

R, RStudio, Shiny, dinamična spletna stran

**Abstract**

Shiny package is used for drawing dynamical functions in a shape of web pages without knowing how to write code in Java, CSS or HTML. We can do this simply by knowing a little bit of R, RStudio and Shiny.

RStudio is an environment of program R (IDE – integrated development environment) and it is much more user friendly than R. It is mostly meant for statistical data processing because it has a lot of built-in options for drawing different graphs. It can be used by any person who has the access to internet because you can get it for free on <http://www.rstudio.com/>.

**Key words**

R, RStudio, Shiny, dynamical web page

**Uvod**

Ustvarjali bi spletne strani pa ne znamo pisati kode v Javi ali HTML-ju. Radi pa bi ustvarili spletno stran dokaj hitro. Temu je namenjen paket Shiny v RStudiu. Če pa želimo uporabiti Shiny za željeno spletno stran, se potrebujemo naučiti le nekaj osnov programskega jezika R. Le-to lahko osvojimo hitro, tudi če ne poznamo nobenega drugega programskega jezika.

Za začetek potrebujemo tri stvari – program R, njegovo okolje oziroma IDLE RStudio in paket Shiny. Na računalnik si namestimo R in RStudio po postopku, ki je prikazan na videoposnetku <https://www.youtube.com/watch?v=eD07NznguA4>. Omenjeni posnetek je namenjen le uporabnikom operacijskega sistema Windows 7. Vendar lahko brez težav najdemo navodila za namestitev tudi za druge operacijske sisteme. Ko prenesemo prosto dostopno različico programa R, so poleg samega programa vključeni tudi razni paketi, ki jih je za uporabo potrebno posebej namestiti. Kako se to stori, je opisano v nadaljevanju.

Nadalje je predstavljen tudi prikaz RStudia ter uporaba paketa Shiny, ne pa tudi natančna navodila za pisanje kode s Shiny-jem, le nekaj osnovnih zadev. Se pravi prispevek je nekakšen pregled uporabe Shiny paketa, več pa se dobi na <http://shiny.rstudio.com/> in marsikaterih drugih spletnih straneh.

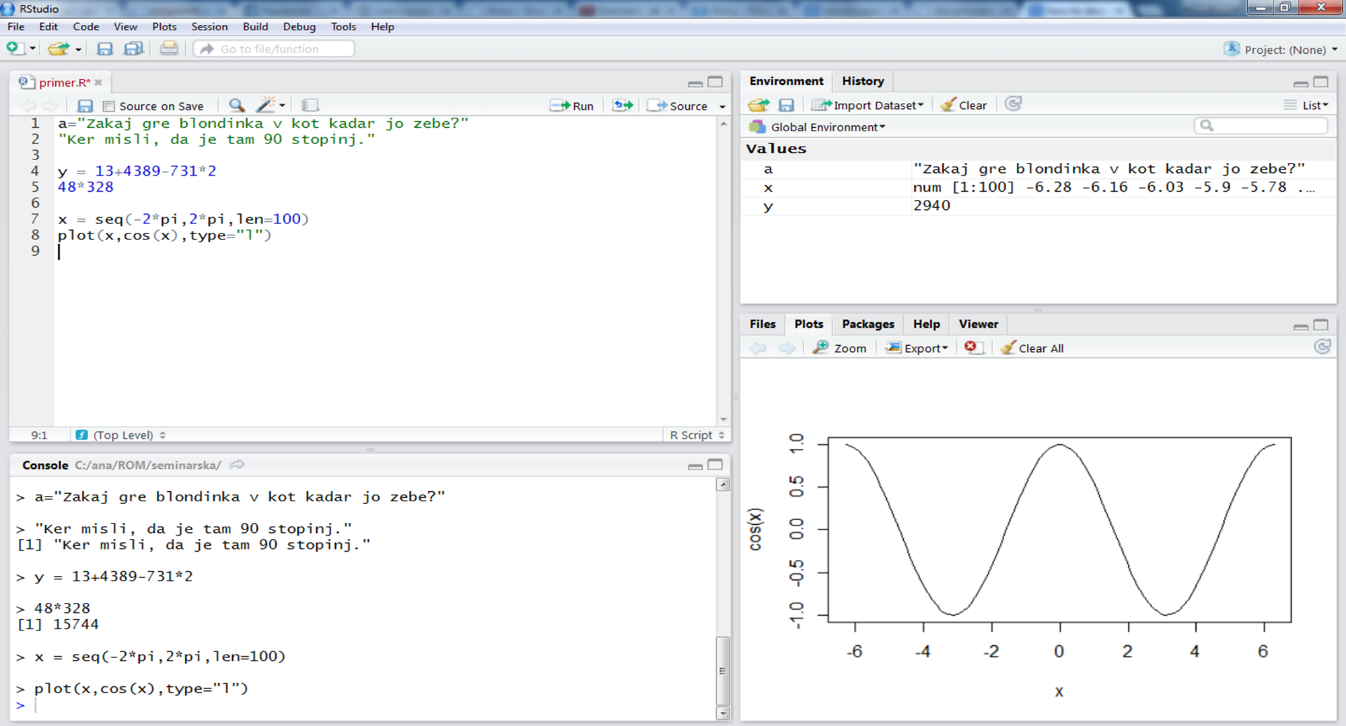
**1 RStudio**

Ko zaženemo program se nam odpre okno, ki je sestavljeno iz 4 komponent (slika 1).

* Levo zgoraj: **ukazno okno,** kjer pišemo program in ga tudi shranimo, če želimo. Da zadevo zaženemo je potrebno pritisniti gumb Run ali krajše Ctrl+Enter.
* Levo spodaj: **konzola,** v katero pišemo najbolj preproste ukaze. V njej lahko preizkušamo razne ukaze – kako delujejo, kaj je sploh potrebno vpisati… Vse kar vpišemo v konzolo pa se ne shranjuje, se pravi, ko bomo naslednjič odprli našo datoteko bo konzola prazna.

Do sedaj opisani okni bi lahko primerjali z različnimi programi kot so Python, R, Matlab, saj ju vsi ti programi potrebujejo za normalno delovanje (vhodna in izhodna enota). Medtem ko okni, ki ju bomo opisali v nadaljevanju nista nujni, vseeno pa ustvarita uporabniku prijaznejši program.

* Desno zgoraj: **okno z okoljem in zgodovino**. V okolju si program shranjuje in izpisuje trenutne globalne spremenljivke (v našem primeru ima trenutno shranjene a, x in y). V zgodovini pa si zapomni pretekle napisane vrstice oziroma ukaze.
* Desno spodaj: okno, v katerem imamo na voljo 5 možnosti: **datoteke, grafi, paketi, pomoč in pregledovalnik.** Tukaj lahko odpiramo datoteke, pregledujemo grafe (izriše le tiste, ki jih vpišemo v konzolo), namestimo razne pakete in uporabljamo pomoč za ukaze. Na sliki 1 imamo v tem oknu izrisano kosinusno funkcijo**.**



**Slika 1: Izgled RStudia**

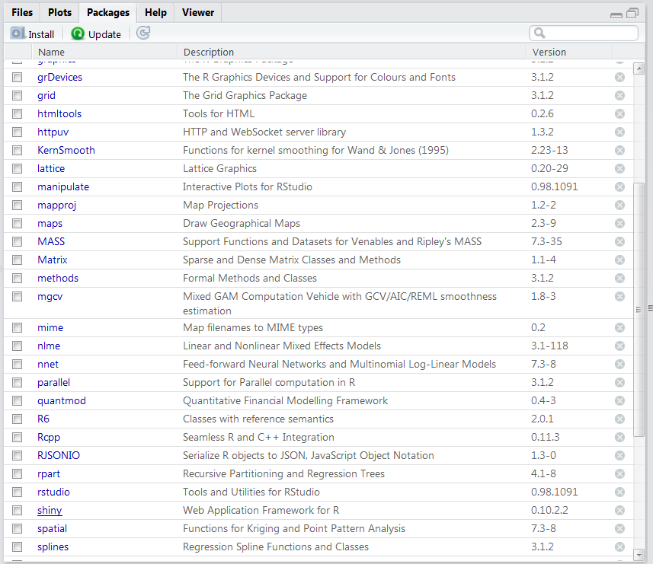
**2 Paket Shiny**

**2.1 Kaj je?**

Shiny je paket v RStudiu, ki nam omogoča enostavno ustvarjanje dinamičnih aplikacij na spletnih straneh. To pomeni, da so aplikacije na nek način žive, se pravi spletna stran se spremeni oziroma posodobi takoj, ko uporabnik nekaj spremeni oziroma uporabi tako imenovani »widget«. To si bomo pogledali v poglavju 2.3 Videz aplikacije.

**2.2 Uporaba**

Program RStudio vsebuje veliko paketov oziroma knjižnic (slika 2), ki jih lahko uporabljamo le, če jih predhodno namestimo. To je zelo preprosto, v konzolo vpišemo install.packages(»ime paketa«). V našem primeru namesto ime paketa napišemo Shiny.



**Slika 2: Nekaj paketov, njihovi opisi in verzije**

Shiny ima tudi zelo stroga pravila glede samih datotek, njihovih imen, lokacij… Tako potrebuje za zagon spletne aplikacije kar dve datoteki. Prva datoteka se ukvarja z izgledom strani in jo je potrebno poimenovati ui.R, druga pa je glavni program, ki računa, riše... Slednja mora imeti ime server.R. Ti dve datoteki imata seveda tudi natančno določena pravila za sestavo kode, ki se jih je dokaj enostavno naučiti na <http://shiny.rstudio.com/tutorial/>

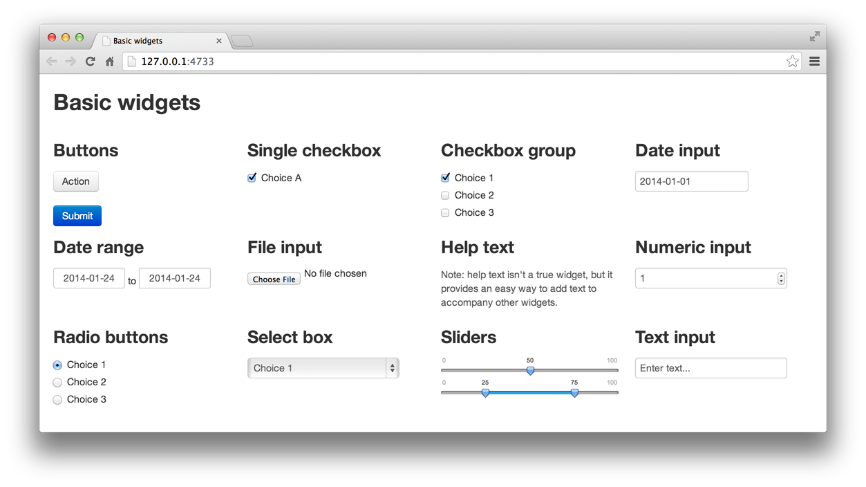
Potrebno je še omeniti, da kamor shranimo datoteko server.R, program privzame, da se tam nahaja delovni imenik oziroma delovni prostor, kar pomeni, da je potrebno vse datoteke, ki bi jih želeli uporabljati, shranjevati v tisti imenik. Če želimo na spletno stran dodati tudi slike, je potrebno ustvariti imenik z imenom www, v kateri se nahajajo željene slike.

**2.3 Videz aplikacije**

Za razliko od HTML-ja imajo Shiny aplikacije kar strogo določeno strukturo spletne strani, se pravi niso zelo prilagodljive. Na voljo so 4 različni videzi, torej tudi 4 različne kode programa, vendar le rahlo odstopajo ena od druge, tako da to ni težava za razumevanje same sestave programa. Bolj natančno je to prikazano v poglavju 2.4 Koda programa.

V glavnem oknu ponavadi izrisujemo grafe, legende, slike…

V pomožnem oknu pa se nahajajo pripomočki za spreminjanje spletne strani, besedilo... Na voljo je veliko oblik prej omenjenih »widget-ov« (nekakšnih gumbov), kot so razni drsniki, izbire itd. (slika 3).

****

**Slika 3: Primeri gumbov za spreminjanje spletne strani**

Za primer si poglejmo zelo osnovno aplikacijo (slika 4). Sestavljena je iz naslova in pa dveh oken – glavnega in pomožnega. Spreminjamo lahko lokaciji teh dveh oken, zgoraj/spodaj, pomožno okno levo ali desno od glavnega, ne moremo pa imeti na primer dveh pomožnih oken poleg glavnega. V tem primeru imamo v glavnem oknu neko besedilo in sliko, ki predstavlja ikono RStudia. V pomožnem okencu pa se nahaja neko besedilo in dva naključno izbrana gumba, ki sta v tem primeru nedelujoča. Tudi če bi ju poskusili spremeniti, npr. drsnik nastaviti iz 25 na 50, se aplikacija ne bi spremenila.



**Slika 4: Primer osnovne Shiny aplikacije**

**2.4 Koda programa**

Kot smo omenili že prej, je koda Shiny aplikacij napisana v dveh datotekah – ui.R in server.R. Pa si sedaj poglejmo osnovne stvari pri obeh.

**2.4.1 Server**

Datoteka server.R je sestavljena iz funkcij, ki jih uporabljamo za izračune in risanje grafov, diagramov. Minimalna koda, ki jo ta datoteka potrebuje je shinyServer(function(input, output){}). Vso preostalo kodo pišemo v zavite oklepaje. Za vsako zadevo, ki jo želimo uporabiti v datoteki UI je potrebno napisati output$ime\_funkcije in nato tako imenovane »render« funkcije. Te so zasnovane tako, da če se vrednosti spremenljivk spremenijo, pošljejo nove podatke datoteki UI, ta pa takoj osveži vse kar je odvisno od teh spremenljivk oziroma osveži spletno stran. Tako dobimo dinamično aplikacijo, ki se posodobi takoj, ko uporabnik povzroči spremembo. Za določanje gumbov pa vpišemo ukaz input$ime\_gumba.

Poglejmo si sedaj primer osnovne datoteke server.R, s katero bomo kasneje skupaj z datoteko UI narisali sinusno funkcijo.

1: x = seq(-2\*pi,2\*pi,len=100)

2: shinyServer(function(input,output){

3: output$plot1 <- renderPlot({

4: y=input$a\*(sin(input$b\*x))\*\*input$c

5: plot(x,y,type="l", col="red")

6: abline(h=0)

7: abline(v=0)

8: })

9: output$text1 <- renderText({

10: paste("Izris funkcije", input$a, "(sin(",input$b,"x))^", input$c)

11: })

12: })

Zgoraj vidimo, da se naš glavni ukaz za izvajanje nahaja v 2. vrstici. Nato pa imamo definirano funkcijo z imenom plot1 (vrstica 3), ki uporablja render funkcijo, saj bomo želeli našo sliko spreminjati z vsemi spremenljivkami, pri katerih imamo ukaze input$\_ (v našem primeru so to a, b, c). V vrstici 9 ponovno ustvarimo funkcijo, tokrat z imenom text1, s primerno render funkcijo. S tem bomo spreminjali tekst glede na spremenljivke.

**2.4.2 Ui**

Datoteka ui.R je sestavljena iz ukazov, ki oblikujejo spletno stran, prav tako pa uporablja funkcije, ki so zapisane v server datoteki. Minimalna koda, ki jo ta datoteka potrebuje je shinyUI(fluidPage()). Vso preostalo kodo pišemo v notranje oklepaje. Kot smo že prej omenili ima datoteka 4 možnosti glede izgleda spletne strani:

* Stran lahko razdelimo na glavno in pomožno oziroma stransko okno (primer je slika 4) – sidebarLayout(),
* Na dva enakovredna dela – splitLayout(),
* Na več enako velikih oken – flowLayout(),
* Mrežasta razporeditev – fluidRow()

Najbolj pogosto uporabljena možnost je prva od naštetih. Tako bomo tudi sedaj nadaljevali naš primer od prej. Takole bi izgledala ui.R datoteka za izris sinusne funkcije.

1: shinyUI(fluidPage(

2: titlePanel("Primer sinusne funkcije"),

3: sidebarLayout(

4: sidebarPanel(

5: "Funkcijo bomo gledali kot",

6: strong("a(sin(bx))^c"),

7: ", kjer lahko spreminjas a, b in c.",

8: p(),

9: p(),

10: sliderInput("a", label="a", min=-3, max=3, value=2),

11: sliderInput("b", label="b", min=-3, max=3, step=0.5, value=-1.5),

12: sliderInput("c", label="c",

13: min=-1,max=3,value=2)

14: ),

15: mainPanel(

16: textOutput("text1"),

17: plotOutput("plot1")

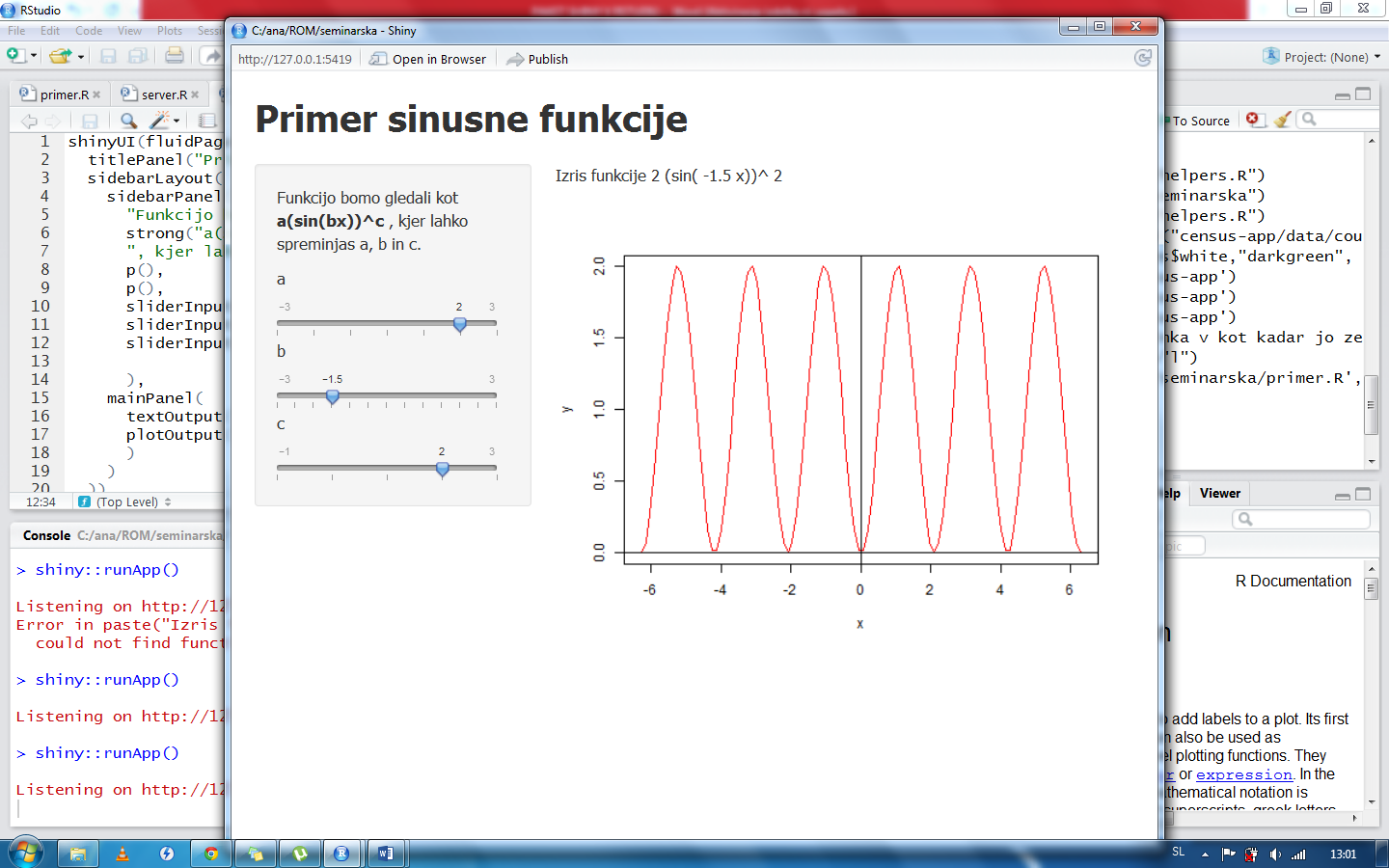
18: )

19: )

20: ))

Naš glavni ukaz za izvajanje datoteke se nahaja v 1. vrstici kode. Nato napišemo naslov strani in uporabimo ukaz za razdelitev na glavno - mainPanel() in pomožno okno – sidebarPanel(). V vsako posebej napišemo kaj se tam nahaja in uporabimo imena funkcij, gumbov, ki smo jih definirali v datoteki server.R.

Sedaj pa poglejmo kaj je nastalo z našima dvema datotekama (slika 5).



**Slika 5: Primer aplikacije, napisane z omenjenima dvema datotekama**

Ustvarili smo spletno stran, kjer so uporabniku na voljo trije drsniki, s katerimi lahko spreminja sliko naše sinusne funkcije.

**Zaključek**

Program R in Shiny paket sta zelo uporabna za statistično obdelavo podatkov, tudi če je teh podatkov večja količina. Zato se strinjam z izjavo nekega uporabnika: »I see great opportunities to built applets for teaching statistics. I am using java applets now, but in R everything is much easier and faster.«. Program je primeren za učenje in bi bilo super, če bi se to izvajalo po raznih šolah, predvsem faksih, saj prej niti ni toliko poudarka na statistiki.

Če primerjam program s programi, ki so meni znani (Matlab, Geogebra, Python, Scratch, Mathematica…), se mi zdi da je eden od enostavnejših, najbolj pa je primerljiv z Matlabom. Imata zelo podobno strukturo, oba temeljita na vektorjih in matrikah, tako da mi je znanje Matlaba zelo pripomoglo pri osvajanju osnov R-a.

Udeležila sem se tudi predavanja o Shiny-ju na Biotehnični fakulteti, kjer je bil seveda večji poudarek na uporabi primernejši za biologe. Opazila sem, da ima program R razne pakete, ki olajšajo zadevo tudi biologu, ne samo statistikom, predvsem pri genetiki, kjer so genski zapisi zelo dolgi in je obdelava podatkov z njim hitra in dokaj enostavna.

Na splošno sem navdušena nad samim programom R, kot tudi nad Shiny-jem. Sicer ima Shiny nekaj pomanjkljivosti, kar je pričakovano, saj je novejši paket, vendar ga uspešno posodabljajo.

**Viri**

1. Luštrik, Roman: Predstavitev Shiny-ja (predavanje), Biotehnična fakulteta, 16.12.2014.
2. Torres-Reyna, Oscar (2013), Introduction to RStudio, <http://dss.princeton.edu/training/RStudio101.pdf> (ogled 14.4.2015).
3. Shiny Cheat Sheet, (2014), <http://www.rstudio.com/resources/cheatsheets/> , 6. odstavek, (ogled 15.4.2014)
4. <http://www.rstudio.com/> (ogled 14.4.2015)
5. <http://en.wikipedia.org/wiki/R_(programming_language)> (ogled 14.4.2015)
6. <http://blog.rstudio.org/2012/11/08/introducing-shiny/> (ogled 15.4.2015)