

Vývoj spustitelných aplikací v R pomocí balíčku shiny

17VSADR – Skriptování a analýza dat v jazyce R

Lubomír Štěpánek^{1, 2}



¹Oddělení biomedicínské statistiky
Ústav biofyziky a informatiky
1. lékařská fakulta
Univerzita Karlova v Praze



²Katedra biomedicínské informatiky
Fakulta biomedicínského inženýrství
České vysoké učení technické v Praze

(2019) Lubomír Štěpánek, CC BY-NC-ND 3.0 (CZ)



Dílo lze dále svobodně šířit, ovšem s uvedením původního autora a s uvedením původní licence. Dílo není možné šířit komerčně ani s ním jakkoliv jinak nakládat pro účely komerčního zisku. Dílo nesmí být jakkoliv upravováno. Autor neručí za správnost informací uvedených kdekoli v předložené práci, přesto vynaložil nezanedbatelné úsilí, aby byla uvedená fakta správná a aktuální, a práci sepsal podle svého nejlepšího vědomí a svých „nejlepších“ znalostí problematiky.

Obsah

- 1 Úvod
- 2 Architektura
- 3 Ovládací prvky & rendering
- 4 Reaktivita
- 5 Dynamizace
- 6 Deployment
- 7 Další možnosti
- 8 Zdroje

Motivace

- možnost jednoduše vytvořit vlastní spustitelnou či webovou aplikaci (jen) pomocí jazyka R a balíčku shiny, a to i bez znalostí webařiny
- zároveň možnost aplikaci front-endově libovolně vylepšit při znalosti
 - HTML (HyperText Markup Language)
 - CSS (Cascading Styl E Sheets)
 - javascriptu
 - i dalšího

Motivace

- aplikace může s výhodou využít prakticky libovolnou funkcionalitu R, ale v back-endu i
 - T_EX-u
 - SQL
 - C++
 - a mnohých dalších
- unikátní motivací je pak zpřístupnění speciálních výpočetních (třeba statistických) možností v klikací podobě uživatelům bez znalostí R

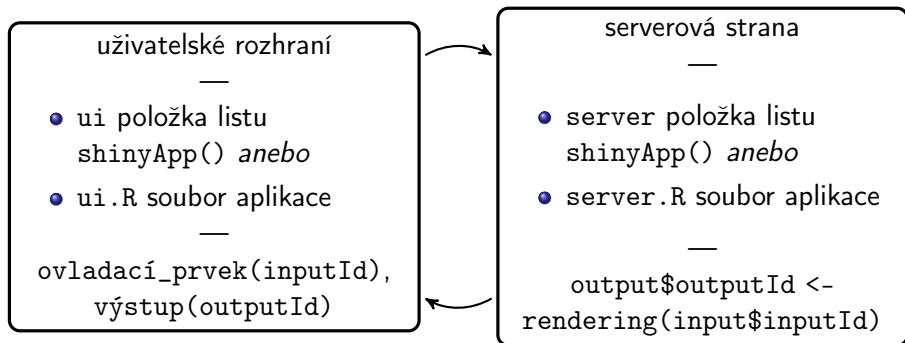
Motivace

- smyslem je
 - napsat kód v R, který by fungoval i v konzoli, resp. RStudios,
 - doplnit prvky uživatelského rozhraní,
 - a samostatně spouštět či vystavit online
- základním frameworkem je R-kový balíček `shiny`

Instalace a inicializace balíčku shiny

```
1 | install.packages(  
2 |   "shiny",  
3 |   dependencies = TRUE,  
4 |   repos = "http://cran.us.r-project.org"  
5 | )  
6 |  
7 | library("shiny")
```

Základní architektura shiny aplikace



Komponenty typické aplikace

- obligátní

- soubor *.R obsahující list `shinyApp()` se složkami `ui` a `R`, tedy

```
1  shinyApp(  
2      ui,  
3      server,  
4      ...  
5  )
```

anebo

- dvojice souborů `ui.R` a `server.R`
- další (ale nepovinné)
 - `global.R`
 - `index.html`
 - libovolné soubory s příponou `.R`
 - složka `www`
 - obrázky, CSS třídy, javascriptové funkce atd.
 - cokoliv dalšího, co R, Shiny a web zná a „snese“

Složka listu ui, resp. samostatný soubor ui.R

- user interface (uživatelské rozhraní)
- R-kový kód určující, které prvky a jak budou uživateli zobrazeny a případně je bude moci měnit (vstupy), eventuálně jak na ně bude aplikace reagovat a co bude vracet (výstupy)
- formálně je o funkci `shinyUI()`

```
1 || ?shinyUI
```

lze ji ale i vynechat

Typické ui.R

```

1      shinyUI(fluidPage(
2          titlePanel("..."),      # název aplikace
3          sidebarLayout(
4
5              # ovládací prvky aplikace (vstupy; levý panel)
6              sidebarPanel(
7                  ## první ovládací prvek,
8                  ## druhý ovládací prvek,
9                  ## ...
10             ),
11
12             # výstupy; pravý panel
13             mainPanel(
14                 ## první prvek výstupu,
15                 ## druhý prvek výstupu,
16                 ## ...
17             )
18         )
19     ))
20

```



Složka listu server, resp. samostatný soubor server.R

- skript s prakticky ryze R-kovým kódem, který obsahuje a definuje všechny aplikací používané funkce a procedury
- v podstatě jde o skript, který by měl jít po drobných úpravách spustit sám o sobě v RStudio či R-kové konzoli
- eventualitou (a lepší) je mít procedury a funkce v separátních .R souborech, které bude server.R volat pomocí příkazu `source()` („modulárně“)
- formálně je o funkci `shinyServer()`

```
1 || ?shinyServer
```

lze ji ale i vynechat

Typický server.R

```

1  library(shiny)
2
3  shinyServer(function(input, output) {
4
5      # kód první funkce/procedury využívající vstupy "z ui.R"
6      # a generující výstupy "pro ui.R"
7
8      # kód druhé funkce/procedury využívající vstupy "z ui.R"
9      # a generující výstupy "pro ui.R"
10
11     # ...
12
13 })

```

global.R

- nepovinná část aplikace
- může obsahovat
 - „globální“ nastavení,
 - konstanty,
 - error handling,
 - apod.

index.html

- nepovinná část aplikace
- obsahuje plnohodnotné HTML a CSS, které definuje vzhled uživatelského rozhraní
- lze zastoupit pomocí `ui` nebo `ui.R`

Podsložka www

- nepovinná část aplikace
- může obsahovat obrázky, CSS styly, javascriptí funkce

„Hello world“ vývojáře v shiny

```

1  ui <- fluidPage(                                     # uživatelské rozhraní
2    titlePanel("Ahoj světě!"),
3    sidebarLayout(
4      sidebarPanel(
5        textInput(
6          inputId = "my_text",
7          label = "Sem vložte svůj text",
8          placeholder = "Ahoj světě!"
9        )
10   ),
11   mainPanel(
12     textOutput(outputId = "my_text")
13   )
14 )
15 )
16 server <- function(input, output){ # serverová strana
17   output$my_text <- renderText({input$my_text})
18 }
19 shinyApp(ui, server)                                # spuštění aplikace

```

„Hello world“ vývojáře v shiny

► GitHub

github.com/LStepanek/17VSADR_..._v_..._R/.../hello_world

Gramatika aplikace na straně ui či ui.R

- stavebnicový systém
- kód pro úroveň vstupů (obvykle levý panel)
- `typ_ovládacího_prvku(...)`

```

1  typ_ovládacího_prvku(
2      inputId = "id_vstupu",
3      argumenty,
4      ...
5  )

```

- `typVýstupu(...)`

```

1  typVýstupu(
2      outputId = "id_vystupu"
3  )

```

Gramatika aplikace na straně server či server.R

- taktéž stavebnicový systém
- vhodnéRenderováníVýstupu({...})

```

1  output$id_vystupu <- vhodnéRenderováníVýstupu({
2
3      # R-ková procedura či funkce beroucí jako vstupy
4      input$id_vstupu
5
6      # a vracející vhodný výstup
7
8  })

```

Slovník aplikace

- dvojice správných typů výstupu `server.R` pro vstupy `ui.R`

typVýstupu	vhodnéRenderováníVýstupu
<code>verbatimTextOutput(...)</code>	<code>renderPrint({...})</code>
<code>textOutput(...)</code>	<code>renderText({...})</code>
<code>tableOutput(...)</code>	<code>renderTable({...})</code>
<code>plotOutput(...)</code>	<code>renderPlot({...})</code>
<code>uiOutput(...)</code>	<code>renderUI({...})</code>
...	...

- kompletní seznam prvků použitelných v `ui` a `server` je zde

https://shiny.rstudio.com/reference/shiny/latest/

- renderovací proceduru k očekávanému výstupu lze najít v nápovědě

1 || ?`verbatimTextOutput`

Textový vstup

- kód na straně ui či ui.R

```
1      textInput(  
2          inputId,           # ID vstupu  
3          label,             # statický popis  
4          value = "",        # iniciační hodnota  
5          width = NULL,      # šířka pruku  
6          placeholder = NULL # vepsaný příklad vstupu  
7      )
```

- kód na straně server či server.R

```
1 || input$inputId # takto je kódován jako proměnná
```

- příklad použití v aplikaci

► [GitHub](#)

github.com/LStepanek/17VSADR ... v jazyce R/.../r kalkulator

Intermezzo – r_kalkulator

- Pokud v kódu aplikace `r_kalkulator` změním příkaz `renderText()` na `renderPrint()`, který další příkaz musím změnit, aby aplikace vracela výsledek?

Intermezzo – r_kalkulator

- Pokud v kódu aplikace `r_kalkulator` změním příkaz `renderText()` na `renderPrint()`, který další příkaz musím změnit, aby aplikace vracela výsledek?
- Spočítejte pomocí kalkulatoru hodnotu výrazu $x^T y$, je-li $x = (1, 3, 5, 2, 6)^T$ a $y = (14, 31, 75, 34, 21)^T$.

Intermezzo – r_kalkulator

- Pokud v kódu aplikace `r_kalkulator` změním příkaz `renderText()` na `renderPrint()`, který další příkaz musím změnit, aby aplikace vracela výsledek?
- Spočítejte pomocí kalkulátoru hodnotu výrazu $x^T y$, je-li $x = (1, 3, 5, 2, 6)^T$ a $y = (14, 31, 75, 34, 21)^T$.
- Najděte součet všech lichých přirozených čísel menších než 100.

Intermezzo – r_kalkulator

- Pokud v kódu aplikace `r_kalkulator` změním příkaz `renderText()` na `renderPrint()`, který další příkaz musím změnit, aby aplikace vracela výsledek?
- Spočítejte pomocí kalkulátoru hodnotu výrazu $x^T y$, je-li $x = (1, 3, 5, 2, 6)^T$ a $y = (14, 31, 75, 34, 21)^T$.
- Najděte součet všech lichých přirozených čísel menších než 100.
- Je číslo 4717 prvočíslem?

Numerický vstup

- kód na straně ui či ui.R

```
1      numericInput(
2          inputId,           # ID vstupu
3          label,             # statický popis
4          min,               # minimální hodnota
5          max,               # maximální hodnota
6          value,             # iniciální hodnota
7          step = NULL,      # krok slideru
8          ...                # další argumenty
9      )
```

- kód na straně server či server.R

```
1 || input$inputId # takto je kódován jako proměnná
```

- příklad použití v aplikaci

► [GitHub](#)

github.com/LStepanek/17VSADR ... v jazyce R/.../histogram 2

Roletka

- uživatel může zvolit právě jednu z jejích možností (podobně jako u radiobuttonu)
- kód na straně ui či ui.R

```
1 || selectInput(  
2 ||     inputId,           # ID vstupu  
3 ||     label,             # statický popis  
4 ||     choices,           # vektor možností s popisky  
5 ||     selected,          # iniciálně vybraná hodnota  
6 ||     ...                # další argumenty  
7 || )
```

- kód na straně server či server.R

```
1 || input$inputId          # takto je kódován jako proměnná
```

- příklad použití v aplikaci

► GitHub

github.com/LStepanek/17VSADR_..._v_jazyce_R/.../histogram_3

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

Radiobutton

- uživatel může zvolit právě jednu z jeho možností
- kód na straně `ui` či `ui.R`

```
1      radioButtons (
2          inputId ,           # ID vstupu
3          label ,             # statický popis
4          choices ,           # vektor možností s popisky
5          selected ,          # iniciálně vybraná hodnota
6          ...                 # další argumenty
7      )
```

- kód na straně server či server.R

```
1 || input$inputId      # takto je kódován jako proměnná
```

- příklad použití v aplikaci

► [GitHub](#)

github.com/LStepanek/17VSADR ... v jazyce R/.../histogram 5

Multiple checkbox

- uživatel může zvolit jednu až všechny jeho možnosti
- kód na straně `ui` či `ui.R`

```
1 checkboxInput (
2     inputId ,           # ID vstupu
3     label ,             # statický popis
4     choices ,           # vektor možností s popisky
5     selected ,          # iniciálně vybraná hodnota
6     ...                 # další argumenty
7 )
```

- kód na straně server či server.R

```
1 || input$inputId           # takto je kódován jako proměnná
```

- příklad použití v aplikaci

► [GitHub](#)

github.com/LStepanek/17VSADR ... v jazyce R/.../mtcars app

Vstup typu datum

- uživatel volí v embedovaném okně jednu hodnotu typu `as.Date`
- kód na straně `ui` či `ui.R`

```
1 | dateInput(  
2 |   inputId,           # ID vstupu  
3 |   label,             # statický popis  
4 |   value,             # iniciální datum  
5 |   min, max           # minimální, maximální hodnota  
6 |   format,            # defaultně "yyyy-mm-dd"  
7 |   language,         # jazyk ("cs" pro češtinu)  
8 |   ...,              # další argumenty  
9 | )
```

- na straně `server` či `server.R` jako `input$inputId`
- příklad použití v aplikaci

► [GitHub](#)

github.com/LStepanek/17VSADR_..._v_jazyce_R/.../calendar_app

Intermezzo – vstup typu datum

- Použijte aplikaci `calendar_app` k nalezení, kolika započatých dní se dožil Albert Einstein, jestliže žil mezi 14. březnem 1879 a 18. dubnem 1955.

HINT: Zamyslete se, zda se nebude hodit do aplikace doplnit ještě druhý kalendářový vstup.

Externě nahrávání vlastních dat

- umožní uživateli nahrát flat-file (i jiná vlastní data)
- kód na straně ui či ui.R

```
1 fileInput(  
2   inputId,           # ID vstupu  
3   label,             # statický popis  
4   multiple = FALSE,  # nahrání více souborů najednou  
5   accept,            # vektor podporovaných přípon  
6   ...                # další argumenty  
7 )
```

- na straně server či server.R jako `input$inputId`
- příklad použití v aplikaci

► GitHub

github.com/LStepanek/17VSADR_..._v_jazyce_R/.../file_upload

Chování shiny aplikace při změně vstupů

- shiny aplikace se může chovat rozdílně v závislosti na okamžité změně vstupních hodnot
- defaultně se při jakékoliv změně vstupních hodnot ihned přepočítávají a aktualizují výstupy
 - to je někdy nevýhodné (a uživatelsky nepříjemné)
- reaktivitu aplikace lze však zrelaxovat ovládacími prvky či „meziukládáním“ počítaných hodnot do reaktivních objektů
 - „meziukládání“ je obecně šikovné
- našimi přáteli jsou
 - `submitButton()`, `isolate()`
 - `reactive({})`, `reactiveValues()`

submitButton()

- nejjednodušším řešením, jak aplikaci zabránit v „nechtěném“ přepočítání hodnot, je použití `submitButton()`
- k prvnímu výpočtu resp. přepočtu dojde jen a tehdy, je-li stisknuto tlačítko `submitButton()`
- jako příklad porovnejme aplikace `r_kalkulator` s ne- a zakomentovaným řádkem 28, tedy

```
1 || submitButton(text = "Spočítej!")
```

resp.

```
1 || #submitButton(text = "Spočítej!")
```

► GitHub

github.com/LStepanek/17VSADR_..._v_jazyce_R/.../r_kalkulator

reactive({})

- slouží k „meziuložení“ objektu v rámci shiny aplikace tak, že lze tento objekt opakovaně volat v libovolných částech server či server.R
- hodnoty objektu jsou po „meziuložení“ neměnné
- příkladem bud' aplikace `histogram_6`

[► GitHub](#)

```
github.com/LStepanek/17VSADR_..._v_jazyce_R/.../histogram_6
```

reactiveValues()

- slouží k „meziuložení“ objektu v rámci shiny aplikace tak, že lze tento objekt opakovaně volat v libovolných částech server či server.R
- hodnoty objektu lze i po „meziuložení“ opakovaně updatovat se zachováním jeho dosavadních hodnot
- příkladem buď aplikace konvergence

► GitHub

`github.com/LStepanek/17VSADR_..._v_jazyce_R/.../konvergence`

Příklad komplexnější aplikace pracující s reaktivitou

- aplikace Conwayova_hra_života
- publikována online na

http://shiny.statest.cz:3838/Conwayova_hra_zivota/

- a na GitHubu

► GitHub

[github.com/LStepanek/vecerni_skola_R_shiny/.../Conway_Game_of_Life](https://github.com/LStepanek/vecerni_skola_R_shiny/blob/master/Conway_Game_of_Life)

Dynamický rendering aplikací v R

- smyslem je uživateli nabízet jen ty ovládací prvky nebo vlastnosti aplikace, které mají v daný okamžik smysl
- obvykle se zajišťuje kombinací funkcí `observe()` a `observeEvent({})` a logických výroků či událostí

Funkce `observeEvent({})` a `observe()`

- funkce `observeEvent({})` reaguje na stisknutí `actionButton` tlačítka
- příkladem je aplikace

► [GitHub](#)

github.com/LStepanek/17VSADR ... v jazyce R/.../konvergence

- funkce `observe()` vyhodnocuje kombinaci logických výroků tlačítka
- příkladem je aplikace

► [GitHub](#)

github.com/LStepanek/17VSADR ... v jazyce R/.../file upload 2

Desktopové spuštění aplikace v RStudiosu či R konzoli

- příkaz `runApp(launch.browser = TRUE)`
- příkaz `shinyApp(ui, server)`
- zelený trojúhelníček vpravo nahoře („Run app“)

Desktopové spuštění aplikace bez RStudio (a bez R konzole)

- spouštěcí (R-kový) soubor je na

► [GitHub](#)

github.com/LStepanek/vecerni_skola_R_shiny/.../spust_aplikaci

- vhodné dodržet architekturu `ui.R` a `server.R` (nikoliv jen jediný soubor `*.R` s aplikací)
- nutné spárovat příponu `.myRscript` s programem, který bude aplikaci spouštět, tedy `spust_aplikaci.myRscript`
- před prvním spuštěním se po poklepaní na tento soubor otevře dialog pro nastavení výchozího spouštěcího programu; v lokální nabídce přes *Další možnosti* a *Najít jinou aplikaci v tomto počítači* vyberme nástroj `Rscript` ve složce *bin* složky `R`
- obvyklá cesta k nástroji vypadá například takto

C:/Program Files/R/R-3.3.0/bin/Rscript

- pokud postup nezafunguje, je nutná editace registrů

Online vypublikování aplikace

- aplikaci lze vystavit zdarma online na web

<https://www.shinyapps.io/>

- nutná registrace, zdarma může běžet maximálně pět aplikací maximálně 25 hodin měsíčně
- link aplikace je obvykle neatraktivní, obsahuje doménu `shinyapps.io`

Vlastní R-kový server

- možné nainštalovať na vlastný web pomocou návodu

<https://www.rstudio.com/products/rstudio/download-server/>

- je třeba se orientovat v Linuxu
- je vhodné používat běžnou linuxovou distribuci (např. Ubuntu, Fedora, „Debian“)
- nutné mít možnost root SSH přístupu k serveru, což běžný provider neumožní
- anebo je možné mít virtuální server na Amazon Web Services (AWS), více zde

<https://aws.amazon.com/blogs/big-data/running-r-on-aws/>

Příklad aplikace s pokročilejším formátováním a funkcionalitami

- aplikace adventni_kalendar_2018
- na aplikaci si ukážeme další možnosti formátování a vlastností
- publikována online na

http://shiny.statest.cz:3838/adventni_kalendar_2018/

- a na GitHubu

► GitHub

[github.com/LStepanek/vecerni_skola_R_shiny/.../adventni_kalendar_2018](https://github.com/LStepanek/vecerni_skola_R_shiny/blob/master/adventni_kalendar_2018)

Počítadlo návštěv

- založeno na permanentním souboru `counter.RData` obsahující dosavadní počet návštěv (jako integer)
- kód v `server.R` pak vypadá

```

1  output$counter <- renderText({
2      if(!file.exists("counter.Rdata")){
3          counter <- 0
4      }else{
5          load(file = "counter.Rdata")
6      }
7
8      counter <- counter + 1
9
10     save(counter, file = "counter.Rdata")
11     paste("Hits:", counter, sep = "")
12
13 })

```

Javascript v shiny aplikaci

- javascript zajišťuje uživatelsky komfortní prostředí aplikace, je překládán přímo prohlížečem
- lze vložit vlastní javascriptí funkcionalitu pomocí kódu do ui.R

```
1 | tags$head(  
2 |   tags$script(  
3 |     type = "text/javascript",  
4 |     src = "*.js"  
5 |   )  
6 | )
```

- eventuálně lze využít více „R“ přístup a zavolat shinyjs balíček pomocí

```
1 | library("shinyjs")
```

a inicializovat ho

```
1 | shinyjs::useShinyjs()
```

Kaskádové styly (CSS) v shiny aplikaci

- kaskádové styly (CSS) umožňují pokročilejší formátování aplikace, než na které stačí HTML
- lze je vložit pomocí kódu do ui.R

```

1 | tags$head(
2 |   tags$link(rel = "stylesheet",
3 |             type = "text/css",
4 |             href = "style.css")
5 | )

```

Busy indikátor

- založen na třídě `style.css`, javascriptové funkci `busy.js` (obě by měly být ve složce `www`) a GIFu `free_busy_indicator.gif`
- kód v `ui.R` pak vypadá

```
1 tags$head(  
2   tags$link(rel = "stylesheet",  
3             type = "text/css",  
4             href = "style.css"),  
5   tags$script(type = "text/javascript",  
6               src = "busy.js")  
7  
8 ),  
9 div(class = "busy",  
10    p("Application is busy..."),  
11    img(src = "free_busy_indicator.gif", height =  
12        50, width = 50)  
13 )
```

data.table

- jQuery plug-in pro hezčí a funkční formátování výstupních tabulek v shiny aplikaci
- na straně ui.R je namísto `tableOutput()` třeba použít `dataTableOutput()`
- na straně server.R je namísto `renderTable({})` třeba použít `renderDataTable({})`

► GitHub

`github.com/LStepanek/17VSADR_..._v_jazyce_R/.../mtcars_app_2`

Další možnosti

- renderování $\text{T}_{\text{E}}\text{X}$ -ové matematické notace v shiny (pomocí MathJaX)
- formátování (header, footer)
- pop-up vyskakovací hlášky, tooltips
- Google Analytics

Kde získat templáty aditivních funkcionalit?

- sledovat přední vývojáře R-kových aplikací (jmenovitě Dean Attali)
- sledovat jejich githubí účty a streamy z useR! konferencí

Příklady existujících aplikací

- aplikace `statisticke_nastroje`

http://shiny.statest.cz:3838/statisticke_nastroje/

- aplikace `Conwayova_hra_zivota`

http://shiny.statest.cz:3838/Conwayova_hra_zivota/

- aplikace `ShinyItemAnalysis` (Patrícia Martinková et al.)

www.shinyitemanalysis.org

Literatura



Chris Beeley. *Web Application Development with R using Shiny*.
City: Packt Publishing, 2013. ISBN: 1783284471.



Hadley Wickham. *Advanced R*. Boca Raton, FL: CRC Press, 2015.
ISBN: 978-1466586963.



Karel Zvára. *Základy statistiky v prostředí R*. Praha, Česká
republika: Karolinum, 2013. ISBN: 978-80-246-2245-3.

Děkuji za pozornost!

lubomir.stepanek@lf1.cuni.cz

lubomir.stepanek@fbmi.cvut.cz

► GitHub

github.com/LStepanek/17VSADR_Skriptovani_a_analyza_dat_v_jazyce_R