

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni preddiplomski studij matematike i računarstva

Ana Koturić i Lucija Bardić

R Shiny web aplikacija za praćenje podataka o pandemiji i predviđanje

Završni praktični projekt - dokumentacija

Osijek, 2021.

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni preddiplomski studij matematike i računarstva

Ana Koturić i Lucija Bardić

R Shiny web aplikacija za praćenje podataka o pandemiji i predviđanje

Završni praktični projekt - dokumentacija

Mentor: doc. dr. sc. Danijel Grahovac

Osijek, 2021.

Sadržaj

1	Uvod	1
1.1	Opis problema	1
1.2	Tehnologije	1
1.3	Organizacija koda	1
2	Podaci	2
2.1	Korišteni podaci	2
2.2	Prilagođavanje podataka	2
3	Vizualizacija podataka	3
3.1	Epidemijske krivulje	3
3.2	Histogrami	5
3.3	Karte	8
4	Web stranica	9
4.1	User interface - UI	9
4.2	Server	10
5	Model	11
5.1	Podaci za model	11
5.2	Kreiranje modela	12
5.3	Vizualizacija modela	12
6	Literatura	15

1 Uvod

1.1 Opis problema

Ovaj projekt sastoji se od dva dijela; u prvom dijelu cilj je bio izraditi web stranicu koja bi svakodnevno automatski ažurirala podatke vezane uz pandemiju koronavirusom (npr. broj zaraženih, oporavljenih, umrlih, cijepljenih) korištenjem podataka dostupnih na internetu. U drugom dijelu bavimo se izradom modela koji služi za što točnije kratkoročno predviđanje kretanja pandemije (po broju zaraženih osoba).

Na sljedećem linku nalazi se cijeli kod ovog projekta: <https://github.com/akoturic/COVID-19-RShiny-App>

1.2 Tehnologije

Projekt je izrađen u programskom jeziku R koji se koristi za statističke izračune i prikaz grafikona. Unutar R-a korišteni su sljedeći paketi i biblioteke:

- Shiny - razvoj interaktivnih web aplikacija
- ggplot2 - jednostavan i detaljan način za prikaz grafova
- plotly - izrada interaktivnih grafova
- jsonlite - JSON parser
- covid19.analytics - sadrži podatke vezane uz koronavirus

Korišteno je još nekoliko paketa koji nisu toliko zastupljeni i bit će opisani u kasnijim poglavljima. Također, osim R-a korišteni su HTML i CSS kako bi poboljšali izgled stranice.

1.3 Organizacija koda

Ovaj projekt podijeljen je u 8 datoteka:

- app.R - izgled aplikacije
- opcenito.R - sadrži HTML kod s općenitim informacijama o virusu
- data.R - dohvaćanje podataka koje koristimo
- hrvatska.R - grafovi i važni brojevi vezani uz podatke u Hrvatskoj
- svijet.R - grafovi i važni brojevi vezani uz podatke u svijetu
- podaci.R - sadrži potrebne grafove za karticu *Pregled podataka*
- model.R - predviđanje kretanja pandemije korištenjem NNAR modela
- style.css

Radi lakšeg snalaženja gotovo svaka kartica iz navigacijske trake nalazi se u zasebnoj datoteci.

2 Podaci

2.1 Korišteni podaci

U svrhu izrade projekta korišteni su otvoreni, strojno čitljivi, podaci koje ustupa Hrvatski zavod za javno zdravstvo te Ministarstvo zdravstva Republike Hrvatske. Podaci su zapisani u JSON formatu i nalaze se na sljedećim linkovima:

- <https://www.koronavirus.hr/json/?action=podaci>
- https://www.koronavirus.hr/json/?action=po_osobama

Također, osim podataka dostupnih putem HZJZ-a, korišteni su podaci iz paketa covid19.analytics koji se većinom dobivaju od strane [Johns Hopkins University Center for Systems Science and Engineering](#) te podaci od [Our World in Data](#). Treba naglasiti da postoji određeno odstupanje u podacima HZJZ u odnosu na druge izvore.

2.2 Prilagođavanje podataka

Kako bi rad s podacima bio lakši, JSON podatke pretvaramo u CSV oblik pomoću sljedeće funkcije:

```
9 json_to_csv <- function(url){
10
11   my.JSON <- fromJSON(url, flatten = TRUE)
12   data_frame <- as.data.frame(my.JSON)
13   return(data_frame)
14 }
```

Slika 1: *json_to_csv* funkcija

Budući da nam nije odgovaralo kako su dani podaci izgledali, korištenjem nekoliko jednostavnih, ugrađenih funkcija smo ih prilagodili.

```
podaci_hrv <- json_to_csv(url = "https://www.koronavirus.hr/json/?action=podaci")
podaci_hrv[,11] <- round_date(as.Date(podaci_hrv[,11], origin='01-01-1997'), "day")
podaci_hrv <- podaci_hrv %>% mutate_if(is.character, as.numeric)
```

Slika 2: Prilagođavanje podataka

Nakon primjene kreirane funkcije i primjenom gotovih funkcija dobivamo sljedeće tablice s podacima:

SlucaejiSvijet	SlucaejiHrvatska	UmrliSvijet	UmrliHrvatska	IzlijeeniSvijet	IzlijeeniHrvatska	CijepljenjiUtrošenihDoza	CijepljeniJednomDozom	CijepljeniDvijeDoze	CijepljeniProtetka24	Datum	noviSlucaeji	noviSvijet
211619700	369392	4429425	6298	189367262	356430	3208101	1694827	1576504	7463	2021-08-21	505	745674
210874026	368887	4417858	6295	188810002	356083	3200638	1691778	1571296	9541	2021-08-20	468	665532
210211494	368419	4407211	6294	188306369	357770	3191530	1688016	1564842	9272	2021-08-19	486	764556
209448936	367933	4396137	6291	187727095	357518	3182592	1684554	1558636	6330	2021-08-18	524	710113
208736825	367409	4384159	6288	187123664	357318	3176538	1682143	1554491	6720	2021-08-17	341	733289
208003536	367068	4375020	6285	186475601	357065	3170096	1679731	1549881	422	2021-08-16	46	419296
207384240	367022	4368519	6283	186071552	356836	3169677	1679544	1549488	3443	2021-08-15	296	539780
207044460	366724	4359705	6282	185649218	356660	3166258	1678479	1546645	8244	2021-08-14	367	781427
206263033	366357	4348337	6280	185111814	356437	3157783	1674678	1540880	9252	2021-08-13	308	698189
205564044	366049	4337809	6278	184558988	356219	3148900	1671367	1534745	8414	2021-08-12	333	750214
204814630	365716	4327361	6275	183940837	356043	3140674	1667896	1528492	9340	2021-08-11	381	647703
204168927	365335	4316541	6275	183358160	355883	3131892	1665334	1522586	7008	2021-08-10	246	673138
203493789	365089	4308402	6273	182805998	355714	3125230	1663029	1517761	1465	2021-08-09	44	477044
203016745	365045	4300374	6272	182379931	355542	3123548	1662342	1516479	4605	2021-08-08	213	552963
202483782	364832	4290957	6271	181954860	355360	3118842	1660504	1513106	8905	2021-08-07	233	688711
201775071	364599	4281747	6270	181508665	355190	3110259	1657443	1506924	666	2021-08-06	108	711365
201063706	364491	4271347	6270	181053540	355000	3109586	1656936	1506706	10705	2021-08-05	247	708694
200355012	364244	4260647	6267	180601355	354830	3099209	1653601	1498932	8150	2021-08-04	271	691585
199663427	363973	4250335	6267	180141325	354695	3087467	1650326	1493859	11107	2021-08-03	186	635729

Slika 3: Podaci o zaraženima, umrlima, izliječenima i cijepljenima u Hrvatskoj i svijetu

spol	dob	Datum	Zupanija
ž	2004	2021-08-18	Karlovačka
M	1999	2021-08-18	Karlovačka
ž	2003	2021-08-18	Karlovačka
ž	1999	2021-08-18	Karlovačka
M	1950	2021-08-18	Karlovačka
ž	1996	2021-08-18	Karlovačka
ž	1995	2021-08-18	Karlovačka
ž	1991	2021-08-18	Karlovačka
ž	1991	2021-08-18	Karlovačka
M	2007	2021-08-18	Karlovačka
M	1972	2021-08-18	Koprivničko-križevačka
M	2003	2021-08-18	Koprivničko-križevačka
ž	1991	2021-08-18	Koprivničko-križevačka
M	1993	2021-08-18	Koprivničko-križevačka
M	1964	2021-08-18	Koprivničko-križevačka
M	1994	2021-08-18	Koprivničko-križevačka
M	1986	2021-08-18	Koprivničko-križevačka
M	1983	2021-08-18	Koprivničko-križevačka
ž	1954	2021-08-18	Koprivničko-križevačka

Slika 4: Podaci o zaraženima po spolu, godištu i županiji

3 Vizualizacija podataka

Sljedeći grafovi bit će prikazani za Hrvatsku, ali na web stranici napravljeni su i grafovi za svijet na identičan način pa zbog toga neće biti prikazani u ovoj dokumentaciji.

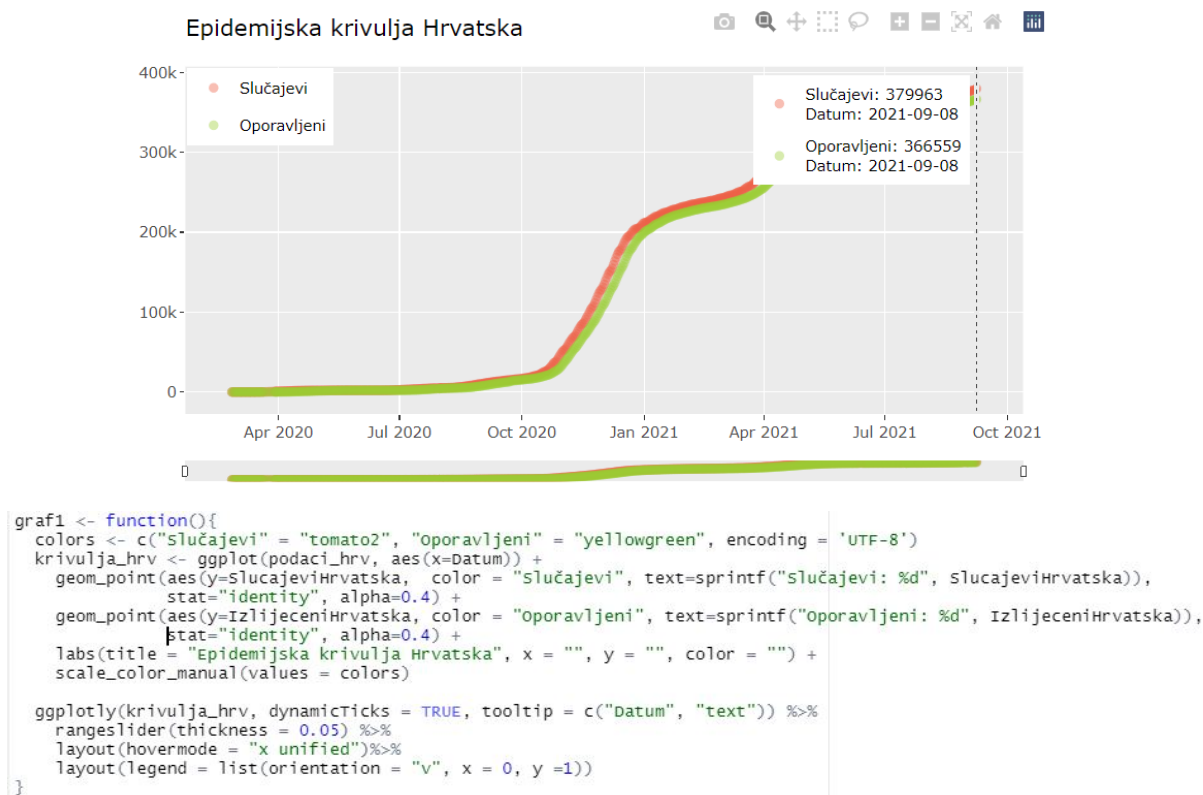
Kako bi podaci bili što pregledniji, dodali smo *pop up* prozorčice, pomoću kojih se jasno mogu vidjeti informacije pridružene vrijednostima na x-osi. Također, dodali smo *rangeslider* kako bi se mogao izabrati interval u kojem želimo promatrati određene podatke.

3.1 Epidemijske krivulje

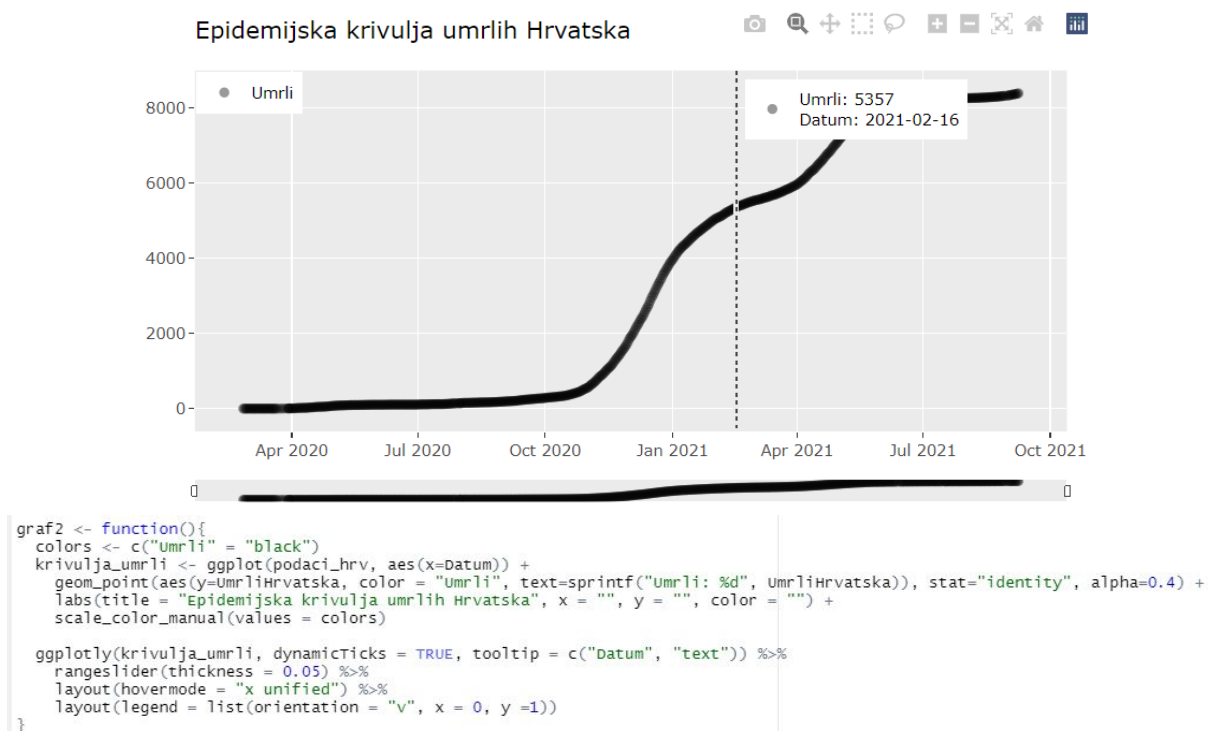
Za prikaz i uređivanje grafova korišteno je nekoliko gotovih funkcija:

- *ggplot()* koja inicijalizira ggplot objekt.
- *geom_point()* koja se koristi za prikaz odnosa između dvije neprekidne varijable (npr. broj slučajeva i datum)

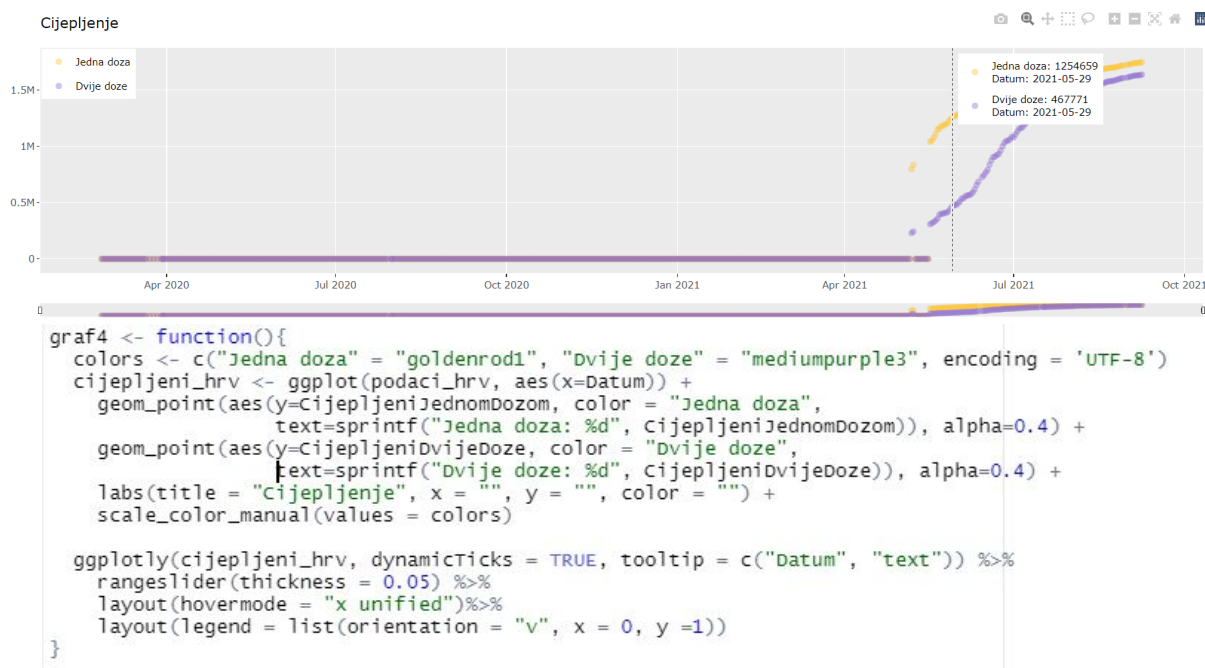
- `ggplotly()` pomoću koje graf postaje interaktivan
- `layout()` pomaže pri uređivanju izgleda kod npr. `pop_up` prozorčića i legende



Slika 5: Prikaz ukupnog broja zaraženih i oporavljenih u Hrvatskoj i pripadni kod



Slika 6: Prikaz ukupnog broja umrlih u Hrvatskoj i pripadni kod

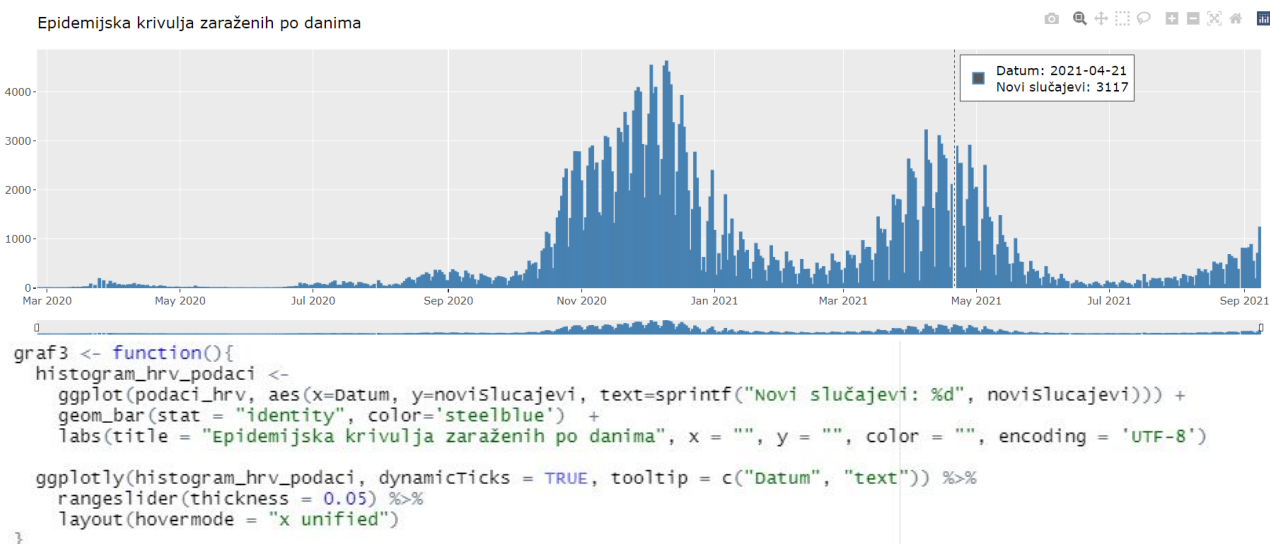


Slika 7: Prikaz ukupnog broja cijepljenih jednom ili dvije doze u Hrvatskoj i pripadni kod

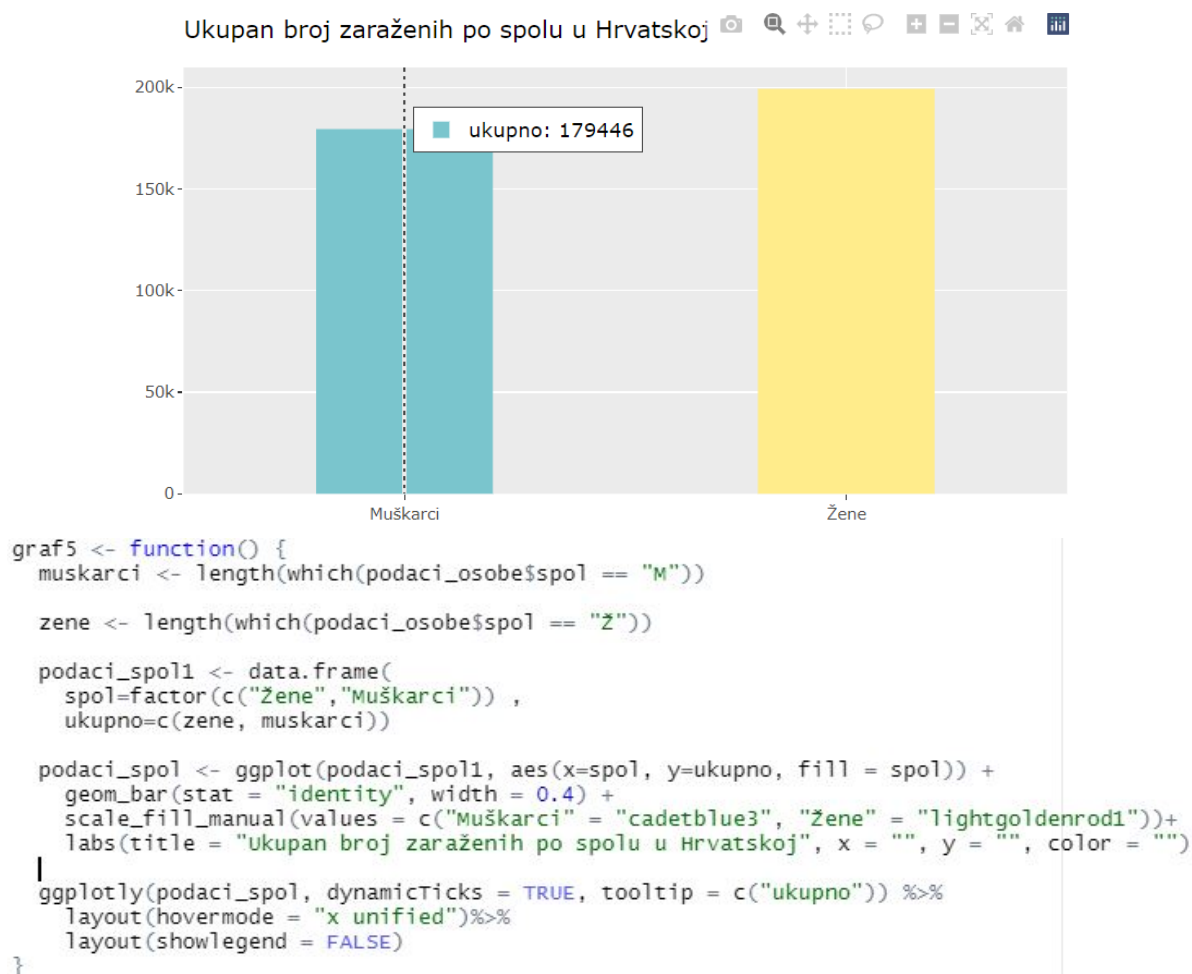
3.2 Histogrami

Za prikaz i uređivanje histograma korištena je sljedeća gotova funkcija (uz one koje su već spomenute kod krivulja):

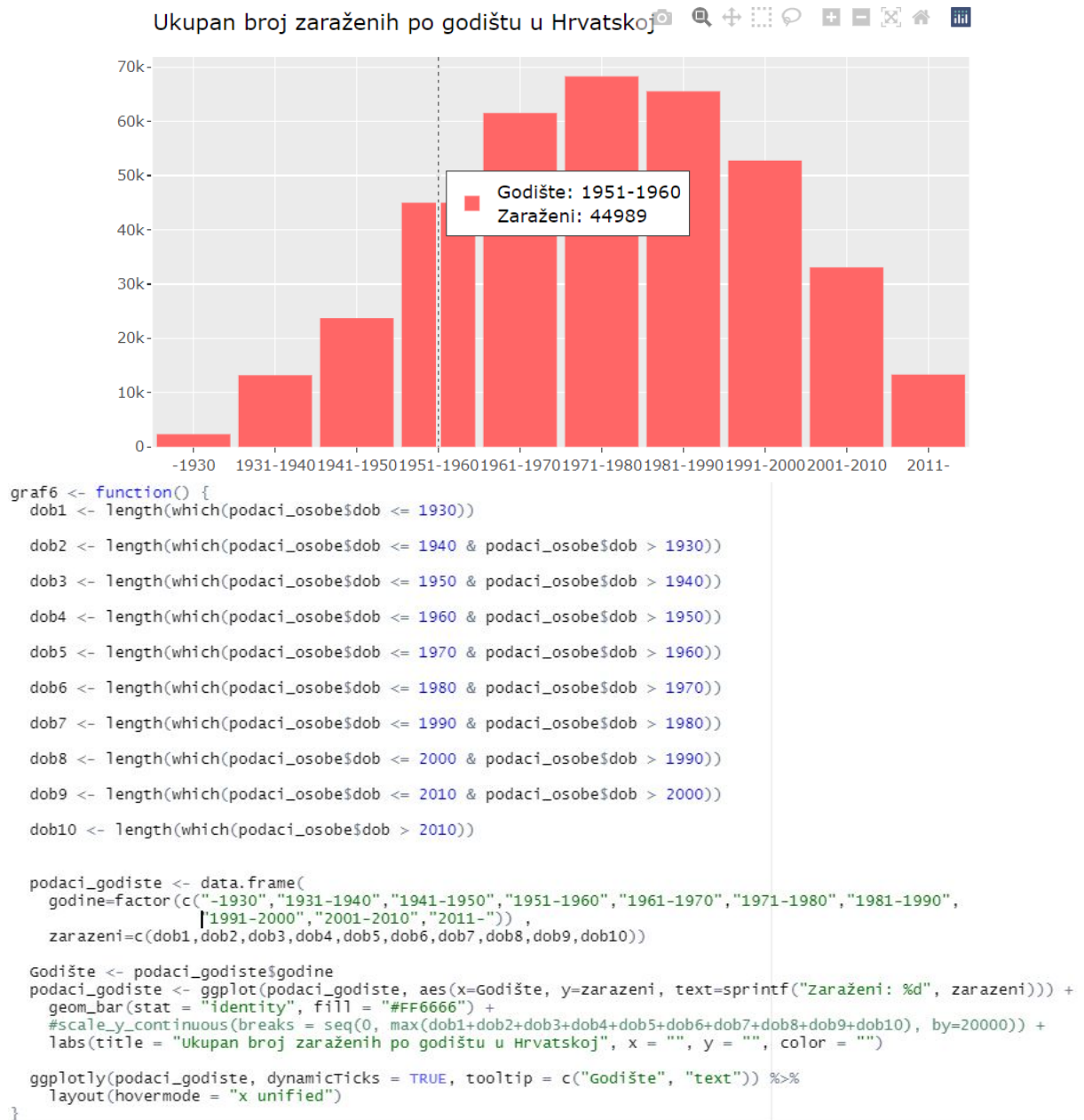
- `geom_bar()` kod koje visina predstavlja broj zaraženih u odnosu na neku varijablu



Slika 8: Prikaz novih slučajeva po danima u Hrvatskoj i pripadni kod



Slika 9: Ukupan broj zaraženih po spolu u Hrvatskoj i pripadni kod



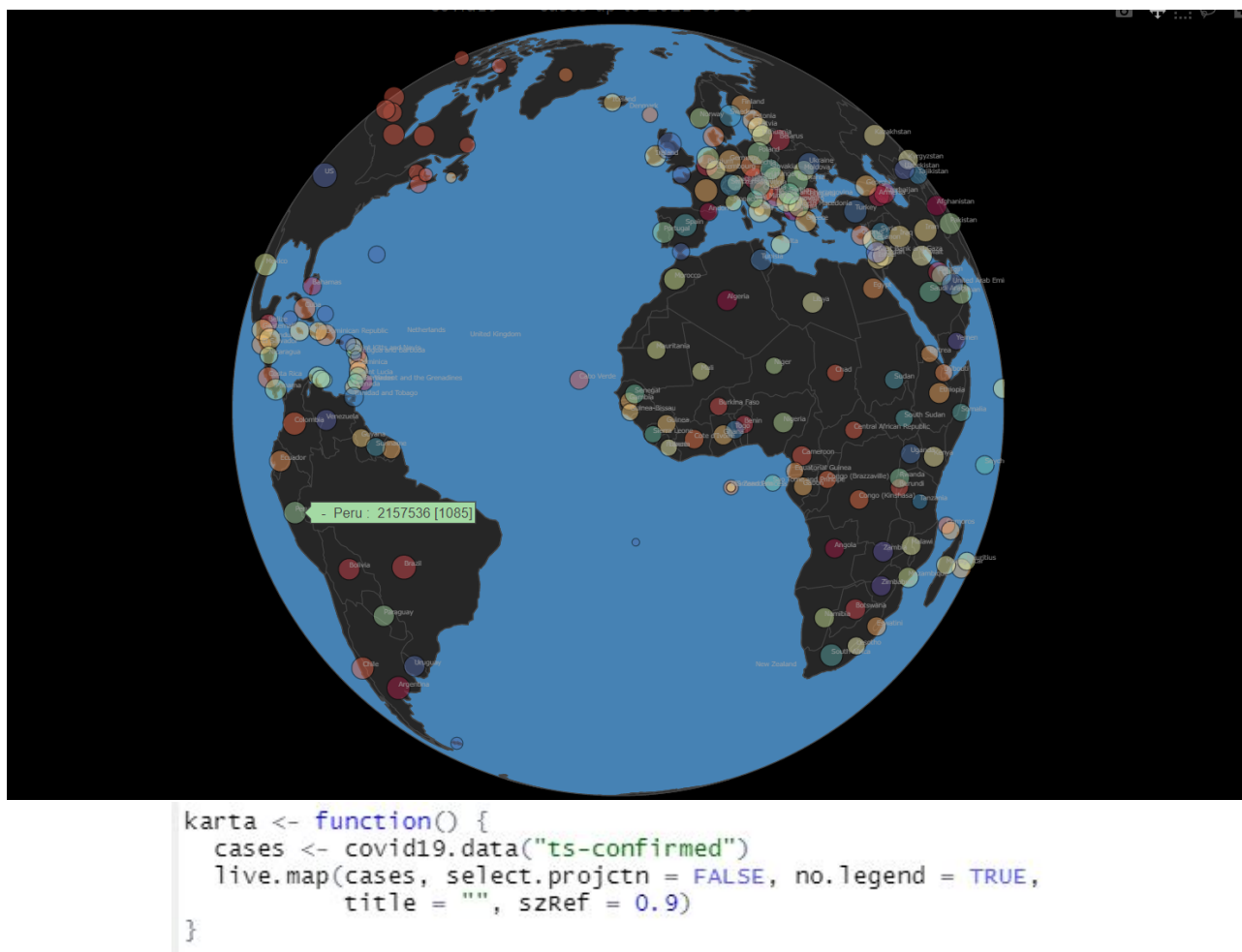
Slika 10: Ukupan broj zaraženih po godištu u Hrvatskoj i pripadni kod



Slika 11: Ukupan broj zaraženih po županijama u Hrvatskoj i pripadni kod

3.3 Karte

Kao vizualno ljepši i zanimljiviji prikaz podataka koristimo interaktivne karte. Paket *co-vid19.analytics*, kao dio R-a, sadrži sve potrebne podatke i funkcije za jednostavno kreiranje karte svijeta na kojoj su prikazani ukupni i novi slučajevi u pojedinoj državi.



Slika 12: Ukupno zaraženih i novozaraženi po državama

4 Web stranica

Paket Shiny nam omogućava jednostavno kreiranje interaktivnih web stranica. Shiny web stranica sastoji se od dva dijela, User interface i servera, koji su detaljnije objašnjeni u sljedećim potpoglavljima. Shiny nudi besplatan deployment aplikacije na Shinyapps.io. Link na web stranicu: <https://covid-19.shinyapps.io/covid-19/>

4.1 User interface - UI

UI kreiramo korištenjem funkcije *bootstrapPage()* koja omogućava da se stranica automatski prilagodi veličini zaslona kojeg korisnik koristi. Sve elemente koji će se nalaziti na stranici ubacujemo unutar ove funkcije.

Kako bismo dobili estetski lijepu navigacijsku traku, koristimo paket *shinythemes* i temu "untitled" te pomoću funkcije *navbarPage* istu kreiramo. Svaka od kartica na navigacijskoj traci kreirana je s funkcijom *tabPanel*. Kao i u Bootstrapu, i ovdje se koristi *grid system* s 12 kontejnera za slaganje elemenata.

Sve funkcije koje u nazivu imaju "box" služe za kreiranje kartica s informacijama vidljivim na stranici. Kako bi web stranica imala lijep izgled, za estetsko uređivanje nekih elemenata unutar *UI-a* koristili smo CSS i u tu svrhu smo tim elementima dodijelili *ID* vrijednosti te uz pomoć njih u datoteci *style.css* definirali izgled tih elemenata.

```

ui <- bootstrapPage(
  includeCSS("www/style.css"),
  theme = shinytheme("united"),
  navbarPage(id = "naslovna", title = "COVID-19", collapsible = TRUE, windowTitle = "COVID19",
    tabPanel(title = "Općenito o virusu",
      column(width = 6, align = "center", box_opcenito()),
      column(width = 3, align = "center", box_simptomi()),
      column(width = 3, align = "center", box_prevenција()),
    tabPanel("Covid-19 u Hrvatskoj",
      column(width = 3, align = "center", box_zarazeni()),
      column(width = 3, align = "center", box_oporavljeni()),
      column(width = 3, align = "center", box_umrli()),
      column(width = 3, align = "center", box_cijepljeni()),
      fluidRow(id = "graf_1", column(6, graf1()),
        column(6, graf2() )),
      fluidRow(id = "graf_3", graf3()),
      fluidRow(id = "graf_3", graf4()),
    tabPanel("Covid-19 u svijetu",
      column(width = 3, align = "center", box_zarazeni2()),
      column(width = 3, align = "center", box_oporavljeni2()),
      column(width = 3, align = "center", box_umrli2()),
      column(width = 3, align = "center", box_cijepljeni2()),
      fluidRow(id = "graf_1", column(6, graf11()),
        column(6, graf22() )),
      fluidRow(id = "graf_3", graf33()),
      fluidRow(id = "graf_3", graf44()),
    navbarMenu("Interaktivna karta",
      tabPanel("Ukupno slučajeva", plotlyOutput("karta", height="800px")),
      tabPanel("Ukupno umrlih", plotlyOutput("karta2", height="800px"))),
    tabPanel(title = "Pregled podataka",
      column(6, align = "center", graf5()),
      column(6, align = "center", graf6()),
      column(6, align = "center", graf7()),
      column(6, align = "center", graf1D(100000))),
    tabPanel(title = "Model",
      fluidRow(id = "graf_3", box_model()),
      fluidRow(id = "range_input", column(3, uiOutput("pocetak")),
        column(3, uiOutput("kraj"))),
      fluidRow(id = "graf_3", plotlyOutput("graf_predvidanje")))
  )
)

```

Slika 13: Kreiranje user interfeacea

4.2 Server

Drugi dio web stranice nalazi se u funkciji `server()`. Ova funkcija gradi objekte pod nazivom *output* koji sadrže sav kod koji je potreban da bi se neki R objekti updateali.

Output objekti spremljeni su kao varijable s nazivom `output$outputId`, a `outputId` nam je potreban kako bismo mogli pozvati te objekte u UI-u kroz `output` funkcije kao što su `plotlyOutput(outputId)` i `uiOutput(outputId)`.


```

server <- function(input, output, session) {
  output$karta <- renderPlotly({
    karta()
  })
  output$karta2 <- renderPlotly({
    karta2()
  })

  output$pocetak <- renderUI({
    dateInput('pocetak',
              label = "Početak predikcije:",
              value = as.Date("2021-08-23"),
              min = as.Date("2020-10-01"),
              max = Sys.Date()-1)
  })

  output$kraj <- renderUI({
    dateInput('kraj',
              label = "Kraj predikcije:",
              value = as.Date("2021-09-23"),
              min = input$pocetak + 5,
              )
  })

  output$graf_predvidanje <- renderPlotly({
    model(input$pocetak[1], input$kraj[1])
  })
}

```

Slika 14: Kreiranje server funkcije

5 Model

Cilj modela bio je napraviti model koji će predviđati kretanje zaraze virusom SARS-CoV-2 za određeni vremenski interval i na osnovu dostupnih podataka. Za izradu modela korišten je *forecast* paket, koji sadrži više različitih metoda, funkcija i alata za prikazivanje i analiziranje predviđanja za određeni atribut, odnosno karakteristiku u nekom vremenskom razdoblju.

5.1 Podaci za model

Za ovaj model koristili smo podatke dostupne putem HZJZ-a, ali bilo ih je potrebno modificirati kako bi predviđanje bilo što realnije. S obzirom na to da se lako može uočiti da je broj novozaraženih osoba u dane vikenda i blagdana znatno manji nego ostalim danima, modificirali smo podatke na način da je broj novozaraženih osoba za određeni dan jednak aritmetičkoj sredini broja novo zaraženih osoba u posljednja tri dana i na taj način smo dobili 'zaglađeniju' krivulju broja novozaraženih osoba.

```

model_data <- data.frame(rev(podaci_hrv$noviSlucajevi), rev(podaci_hrv$Datum))
for(x in 3:length(podaci_hrv$noviSlucajevi)){
  model_data$rev.podaci_hrv.noviSlucajevi.[x] <- round(mean(c(rev(podaci_hrv$noviSlucajevi)[x],
                                                                rev(podaci_hrv$noviSlucajevi)[x-1],
                                                                rev(podaci_hrv$noviSlucajevi)[x-2])))
}
names(model_data)[1] <- "noviSlucajevi"
names(model_data)[2] <- "Datum"

```

Slika 15: Modificiranje podataka za model

5.2 Kreiranje modela

Za kreiranje modela koristili smo funkciju `nnetar()` iz već spomenutog paketa `forecast`; to je funkcija koja koristi neuronske mreže za predviđanje određenog vremenskog atributa u zadanom vremenskom intervalu. Nakon što smo kreirali model napravili smo predviđanje uz pomoć funkcije `forecast()`.

```
real_data <- model_data$noviSlucajevi
original_data <- model_data$noviSlucajevi[1:(which(model_data$Datum==pocetak)-1)]
rows <- NROW(original_data) #racuna broj redaka
training_data<-original_data[1:(rows)]
N_forecasting_days <- as.numeric(kraj-pocetak)+1

#NNAR Model
data_series<-ts(training_data)
model_NNAR<-nnetar(data_series, size = 5)

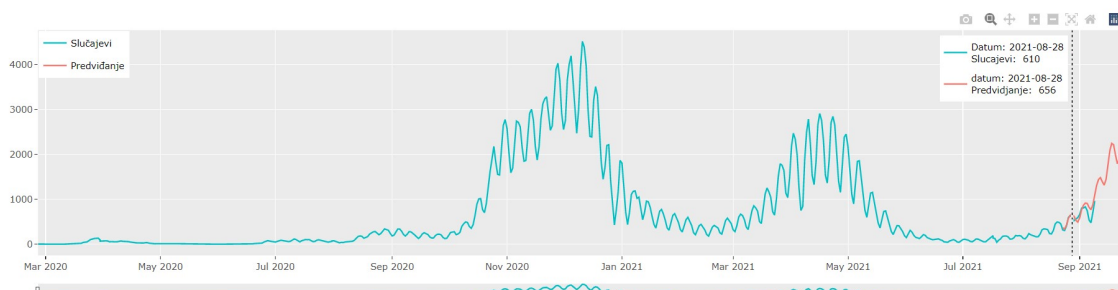
#predvidanje
forecasting_NNAR <- forecast(model_NNAR, h=N_forecasting_days)
```

Slika 16: Kreiranje modela i predikcija

5.3 Vizualizacija modela

Unutar paketa `forecast` postoje funkcije za automatsku vizualizaciju predikcije, no kako smo mi htjeli da naš prikaz bude interaktivan, vizualizaciju smo napravili sami.

Da bismo vizualizirali predviđanje, morali smo prvo izvući 'predviđene' podatke o broju novozaraženih osoba iz varijable `forecasting_NNAR` koja sadrži i neke dodatne podatke o predviđanju. Nakon što smo pristupili podacima o predviđanju prikazali smo ih uz pomoć paketa `ggplot2` i `plotly`, te smo uz predviđene podatke, prikazali i stvarne podatke.



```
#vizualizacija
Predvidjanje <- round(forecasting_NNAR$mean)
datum <- seq(pocetak, kraj, by="days")
Datum <- model_data$Datum
Slucajevi <- model_data$noviSlucajevi
colors <- c("Slučajevi" = "tomato2", "Predviđanje" = "yellowgreen", encoding = 'UTF-8')
krivulja_hrv <- ggplot() +
  geom_line(aes(x=Datum, y = Slucajevi, color = "Slučajevi")) +
  geom_line(aes(x=datum, y = Predvidjanje, color = "Predviđanje")) +
  labs(title = "", x = "", y = "", color = "")

graf <- ggplotly(krivulja_hrv, dynamicTicks = TRUE, tooltip = c("Datum", "datum", "Slucajevi", "Predvidjanje")) %>%
  rangeslider(thickness = 0.05) %>%
  layout(hovermode = "x unified") %>%
  layout(legend = list(orientation = "v", x = 0, y = 1))
```

Slika 17: Vizualizacija predviđanja i pripadni kod

Uz pomoć funkcije `dateInput()` kreirali smo sučelje za unos intervala za predikciju, te smo postavili neke uvjete:

- prvi dan u intervalu može najviše biti trenutni dan

- zadnji dan u intervalu mora biti veći od prvog dana u intervalu
- interval mora biti najmanje razdoblje od 6 dana

```
output$pocetak <- renderUI({
  dateInput('pocetak',
    label = "Početak predikcije:",
    value = as.Date("2021-08-23"),
    min = as.Date("2020-10-01"),
    max = Sys.Date() - 1)
})

output$kraj <- renderUI({
  dateInput('kraj',
    label = "Kraj predikcije:",
    value = as.Date("2021-09-23"),
    min = input$pocetak + 5,
    max = Sys.Date() - 1)
})
```

Slika 18: Sučelje za unos intervala za predikciju i pripadni kod

Sučelje za unos intervala omogućilo nam je da predviđanje radimo za interval po želji, odnosno interval unesen u sučelje za interval. Zbog toga smo kreirali funkciju *model()* koja se sastoji od kodova koje smo gore spomenuli.

```
model <- function(pocetak, kraj){
  real_data <- model_data$noviSlucajevi
  original_data <- model_data$noviSlucajevi[1:(which(model_data$Datum==pocetak)-1)]
  rows <- NROW(original_data) #racuna broj redaka
  training_data <- original_data[1:(rows)]
  N_forecasting_days <- as.numeric(kraj-pocetak)+1

  #NNAR Model
  data_series <- ts(training_data)
  model_NNAR <- nnetar(data_series, size = 5)

  #predviđanje
  forecasting_NNAR <- forecast(model_NNAR, h=N_forecasting_days)

  #vizualizacija
  Predvidjanje <- round(forecasting_NNAR$mean)
  datum <- seq(pocetak, kraj, by="days")
  Datum <- model_data$Datum
  Slucajevi <- model_data$noviSlucajevi
  colors <- c("Slucajevi" = "tomato2", "Predvidjanje" = "yellowgreen", encoding = 'UTF-8')
  krivulja_hrv <- ggplot() +
    geom_line(aes(x=Datum, y = Slucajevi, color = "Slucajevi")) +
    geom_line(aes(x=datum, y = Predvidjanje, color = "Predvidjanje")) +
    labs(title = "", x = "", y = "", color = "")

  graf <- ggplotly(krivulja_hrv, dynamicTicks = TRUE, tooltip = c("Datum", "datum", "Slucajevi", "Predvidjanje")) %>%
    rangeslider(thickness = 0.05) %>%
    layout(hovermode = "x unified") %>%
    layout(legend = list(orientation = "v", x = 0, y = 1))

  return(graf)
}
```

Slika 19: Kod funkcije *model()*

Funkcija *model()* prima vrijednosti *pocetak* i *kraj*, koje označavaju početak i kraj intervala i te vrijednosti su joj proslijeđene iz sučelja za unos intervala.

```
output$graf_predvidanje <- renderPlotly({  
  model(input$pocetak[1], input$kraj[1])  
})
```

Slika 20: Prosljeđivanje intervala funkciji *model()*

6 Literatura

Literatura

- [1] <https://shiny.rstudio.com/>
- [2] <https://www.r-graph-gallery.com/index.html>
- [3] <https://www.koronavirus.hr/>
- [4] <https://cran.r-project.org/web/packages/covid19.analytics/covid19.analytics.pdf>
- [5] <https://rpubs.com/abotalebmostafa/744347>
- [6] <https://www.rdocumentation.org>
- [7] <https://github.com/owid/covid-19-data/tree/master/public/data/vaccinations>