

Domača naloga 3

Alen Kahteran

09. 10. 2020

Pridobivanje podatkov

Preko knjižnice `curl` pridobimo podatke ki jih nato spremenimo v `tibble`, ki se obnaša praktično enako kot `data.frame`, le da je meni osebno lepši prikaz.

```
library(curl)
df_full <- read.csv(
  curl("https://raw.githubusercontent.com/sledilnik/data/master/csv/stats.csv"))
library(tidyverse)
library(lubridate)
df_full <- tibble(df_full)
```

Obdelava podatkov

Za ustrezen prikaz podatkov potrebujemo naslednje stolpce:

- `cases.confirmed`
- `tests.performed`
- `date`

stolpec `date` je v obliki `character`, zato ga ustrezno pretvorimo v `date`.

Dodamo stolpec z zapisom dneva v tednu (Po, T, Sr, C, P, S, n)

```
# select only required columns
df_simple <- select(df_full, date, tests.performed, cases.confirmed)
# convert dates from character to datetime and then to date (due to ggplot)
df_simple$date <- parse_date_time(df_simple$date, c("Ymd"))
df_simple$date <- as.Date(df_simple$date)

# add a column day_in_week which converts dates into numbers from 1-7
# 1 is Sunday, 2 is monday, ...
df_simple <- mutate(df_simple, day_in_week = wday(date))
# recode changes numerical values to what we want to have in the plot.
df_simple$day_in_week <- recode(df_simple$day_in_week,
  `1` = "N",
  `2` = "Po",
  `3` = "T",
  `4` = "Sr",
  `5` = "C",
  `6` = "P",
  `7` = "S")

# adding a column for correct coloring in ggplot
df_simple <- mutate(df_simple, day_color = if_else(day_in_week == "N", 1, 0, NA_real_))
```

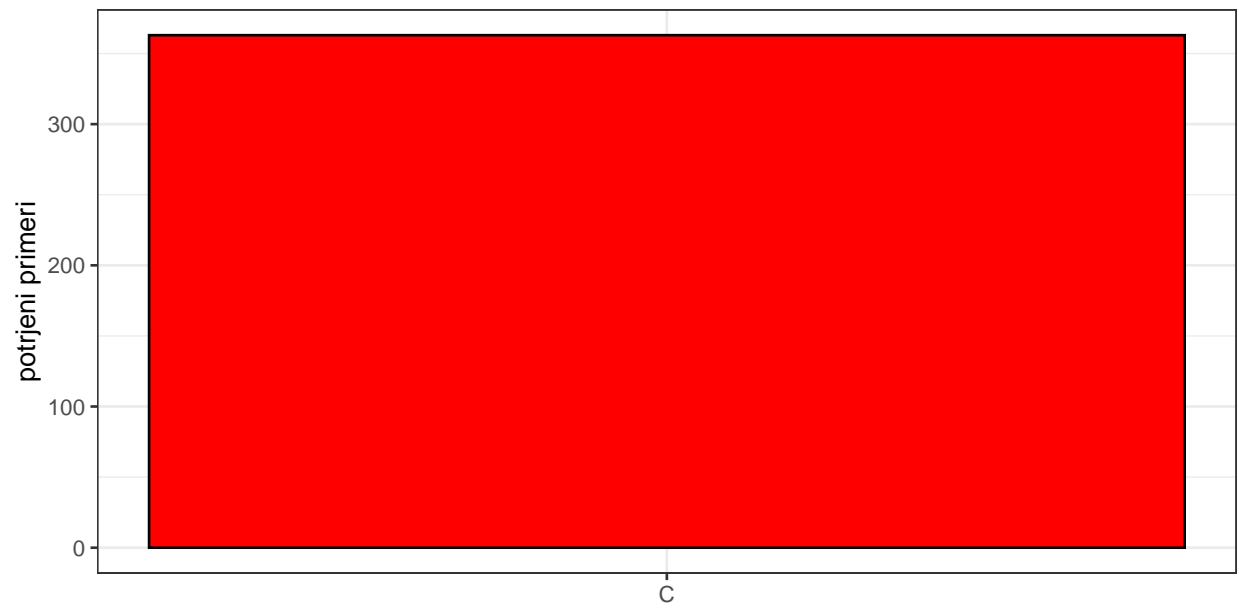
V navodilih domače naloge ni jasno napisano ali je željeno da se to počne znotraj funkcije, ali ne. Osebnostno rad večino obdelav podatkov počnem ločeno od drugih operacij (risanje grafov, izpisovanje, exportanje, itd.). Zato sem dovolj obdelal podatke že prej, in bo funkcija `narisi(n)` le ustrezno izrisala sliko.

```
# a function for plotting n last days with data in cases.confirmed column
narisi <- function(n) {

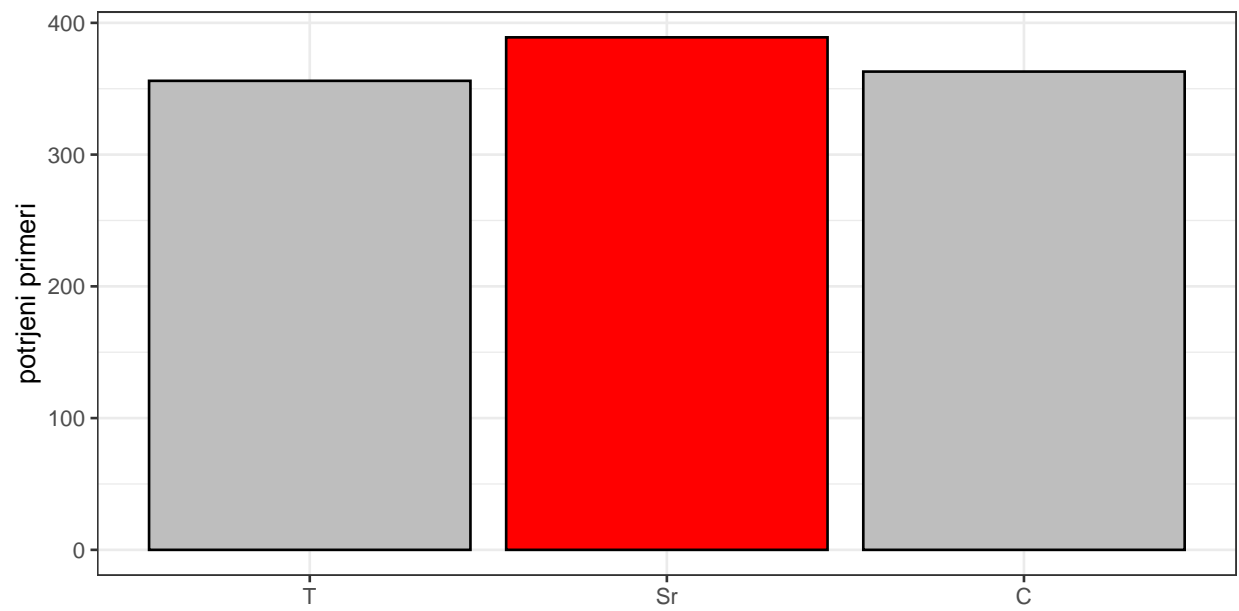
  # keeping rows which don't have NA values in cases.confirmed
  df_simple_n <- filter(df_simple, !is.na(cases.confirmed))
  # select n last rows (looking at dates)
  df_simple_n <- df_simple_n[df_simple_n$date > (max(df_simple_n$date) - ddays(n)), ]
  # changing day_color into 2 (due to correct coloring) where cases.confirmed was
  # maximum value.
  df_simple_n <- mutate(df_simple_n,
                        day_color = if_else(cases.confirmed == max(cases.confirmed),
                                             2,
                                             day_color)
                      )

  # changing into factor due to ggplot2 taking that as input
  df_simple_n$day_color <- factor(df_simple_n$day_color)

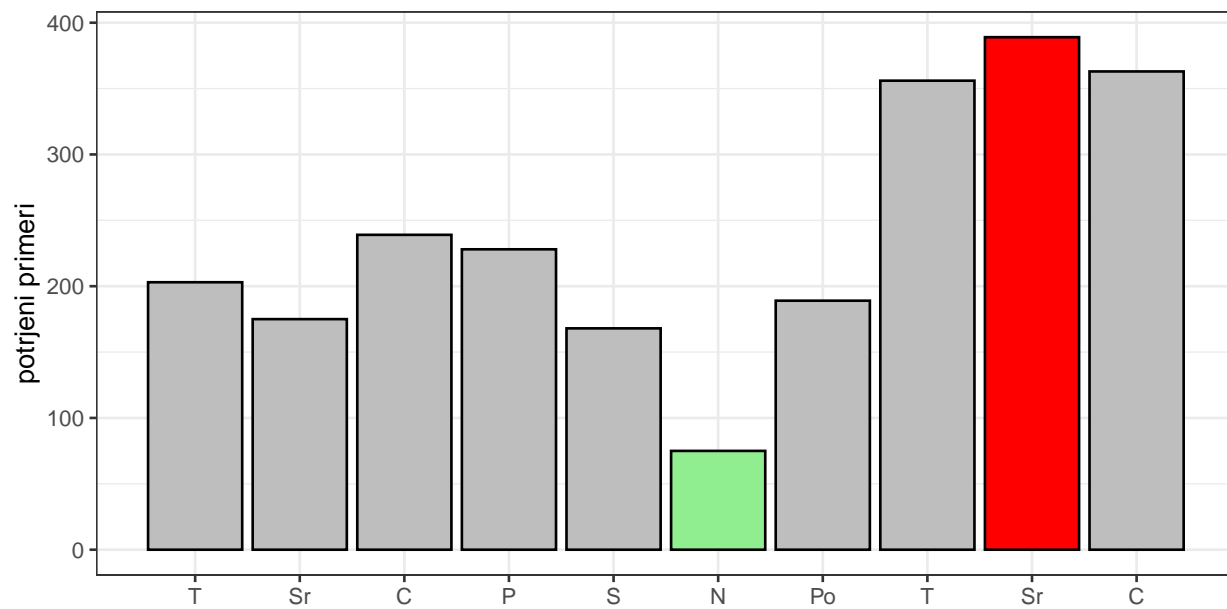
  # correctly selecting data for ggplot
  g <- ggplot(df_simple_n, aes(x=date, y=cases.confirmed, fill=day_color))
  # adding barplot and outline as black color
  g + geom_bar(stat="Identity", color="black") +
  # add y label
  ylab("potrjeni primeri") +
  # define colors for factors (0, 1, 2) and remove legend
  scale_fill_manual(breaks=c(0, 1, 2),
                    values=c("#bebebe", "#90ee90", "#ff0000"),
                    guide=FALSE) +
  # change ticks from date to pre-defined labels
  scale_x_date(breaks=df_simple_n$date,
               labels=df_simple_n$day_in_week,
               minor_breaks="1 day") +
  # changing the theme (my favourite)
  theme_bw() +
  # removing x label as there isn't one in the instructions
  theme(axis.title.x = element_blank())
}
narisi(1)
```



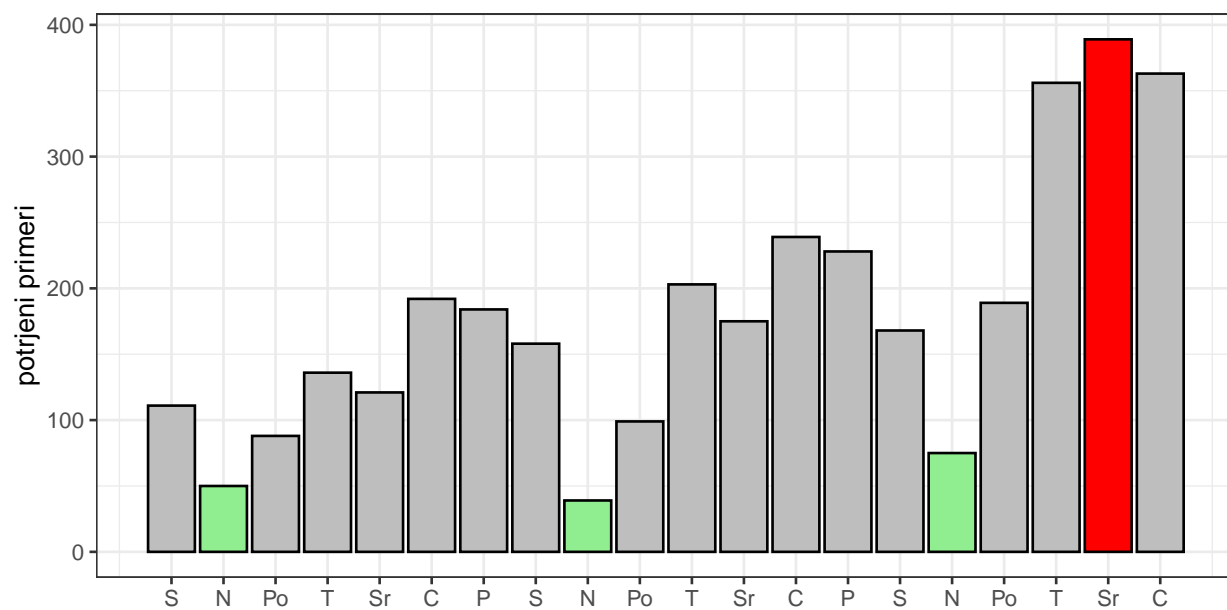
```
narisi(3)
```



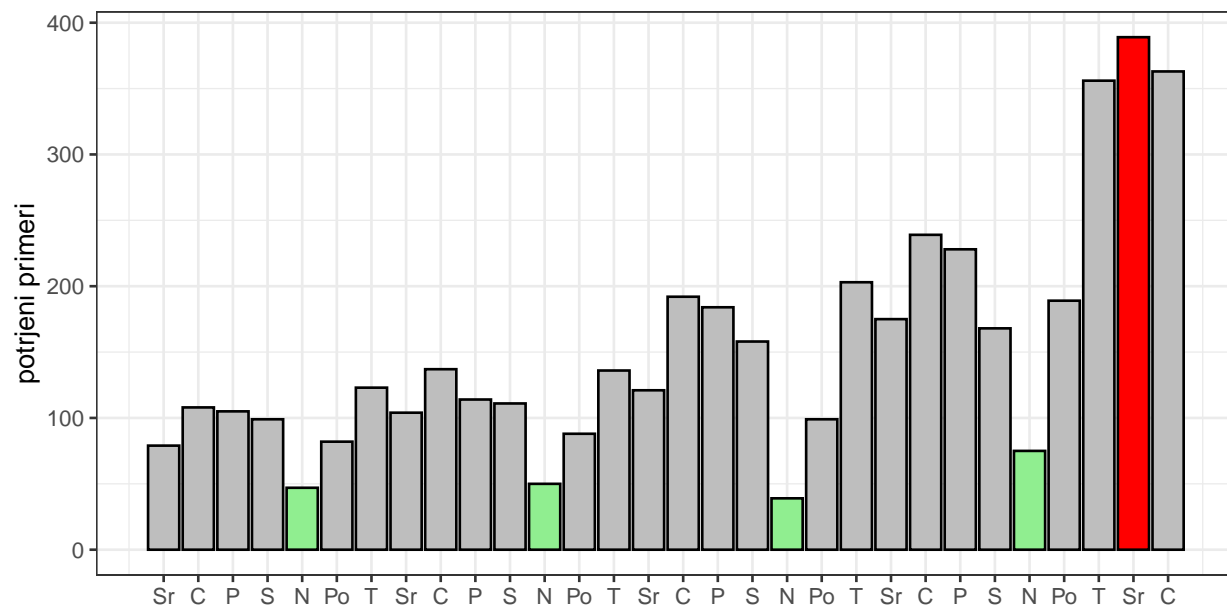
```
narisi(10)
```



`narisi(20)`



`narisi(30)`



narisi(40)

