Математички факултет

Универзитет у Београду

Статистички софтвер 4

Семинарски рад

август 2019.
Београд

Аутор

Марија Костић 286/14

Ментор

Мирјана Вељовић

Садржај

Први задатак

Одабрати базу података и у форми мини истраживања илустровати рад са пакетима и функцијама наученим у току овог курса. Од саме базе и вас зависи шта ћете испитивати и на шта ће се истраживање фокусирати. Важно је да то што радите им неког смисла, тј. да можете да доносите закључке и интерпретирате резултате.Потребно је:

(а) Илустровати рад са основним функцијама из пакета $dplyr$.

(б) Променљиве од интереса приказати различитим графицима из пакета $ggplot2$. Избор графика је препуштен вама и зависи од базе,али је пожељно направити неколико графика, средити их,приказати неке од занимљивих и специфичних графика,итд.

(в) Илустровати основне функције за сређивање базе података.

(г) Одабрати још једну базу која је у вези са том(ако не можете да пронађете одговарајућу, направите је сами!) па приказати основне функције за рад са релацијама између њих.

Пакет $tidyverse$ садржи серију пакета($tidyr, dplyr, ggplot2$)који ће нам бити потребни за анализу података.

```
#install.packages("tidyverse")
library(tidyverse)

## -- Attaching packages ---------------------------------------------------------
---------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.0     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------
--------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Радићемо са базом података која садржи информације о полицијском заустављању црнаца и белаца у држави Мисисипи од јануара 2013.године до средине јула 2016.године.

Пре него што кренемо са истраживањем, учитаћемо базу са којом ћемо радити.

```
baza <- read.csv("C:/Users/Korisnik/Desktop/SS4 seminarski/baza.csv")
head(baza,5) # prikazacemo prvih 5 elemenata baze

##              id state  stop_date       county_name county_fips
## 1 MS-2013-00001    MS 2013-01-01      Jones County       28067
## 2 MS-2013-00002    MS 2013-01-01 Lauderdale County       28075
## 3 MS-2013-00003    MS 2013-01-01       Pike County       28113
## 4 MS-2013-00004    MS 2013-01-01    Hancock County       28045
```

```
## 5 MS-2013-00005    MS 2013-01-01     Holmes County       28051
##            police_department driver_gender driver_birthdate driver_race
## 1 Mississippi Highway Patrol          male       1950-06-14       Black
## 2 Mississippi Highway Patrol          male       1967-04-06       Black
## 3 Mississippi Highway Patrol          male       1974-04-15       Black
## 4 Mississippi Highway Patrol          male       1981-03-23       White
## 5 Mississippi Highway Patrol          male       1992-08-03       White
##                                                 violation_raw officer_id
## 1                    Seat belt not used properly as required       J042
## 2                                          Careless driving       B026
## 3 Speeding - Regulated or posted speed limit and actual speed       M009
## 4 Speeding - Regulated or posted speed limit and actual speed       K035
## 5 Speeding - Regulated or posted speed limit and actual speed       D028
##   driver_age        violation
## 1         63        Seat belt
## 2         46 Careless driving
## 3         39         Speeding
## 4         32         Speeding
## 5         20         Speeding
```

База садржи следеће елементе :

- $\boldsymbol{id}$ – идентификациони број заустављеног возача
- $\boldsymbol{state}$ – држава
- $\boldsymbol{stop\_date}$ – датум заустављања
- $\boldsymbol{county\_name}$ – име округа државе
- $\boldsymbol{county\_fips}$ – државна стандардна публикација за обраду информација
- $\boldsymbol{police\_department}$ – полицијска управа
- $\boldsymbol{driver\_gender}$ – пол возача
- $\boldsymbol{driver\_birthdate}$ – датум рођења возача
- $\boldsymbol{driver\_race}$ – раса возача
- $\boldsymbol{violation\_raw}$ – опис преступ
- $\boldsymbol{officer\_id}$ – идентификациони број полицијског службеника
- $\boldsymbol{driver\_age}$ – године возача
- $\boldsymbol{violation}$ – преступ

```
class(baza) #vidimo da je baza oblika data.frame pa je moramo konvertovati u tibble
```

```
## [1] "data.frame"
```

```
trafficstops<-as_tibble(baza)
head(trafficstops,5) #ispis prvih 5 elemenata tibbla
```

```
## # A tibble: 5 x 13
##   id    state stop_date county_name county_fips police_departme~
##   <fct> <fct> <fct>     <fct>             <int> <fct>
## 1 MS-2~ MS    2013-01-~ Jones Coun~       28067 Mississippi Hig~
## 2 MS-2~ MS    2013-01-~ Lauderdale~       28075 Mississippi Hig~
## 3 MS-2~ MS    2013-01-~ Pike County       28113 Mississippi Hig~
## 4 MS-2~ MS    2013-01-~ Hancock Co~       28045 Mississippi Hig~
## 5 MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
## # ... with 7 more variables: driver_gender <fct>, driver_birthdate <fct>,
## #   driver_race <fct>, violation_raw <fct>, officer_id <fct>,
## #   driver_age <int>, violation <fct>
```

```
attach(trafficstops)
```

Коришћењем функције $filter()$ можемо изабрати одређене опсервације на основу неких услова.

```
filter(trafficstops, county_name == "Leake County") # prikazujemo zaustavljanja u ok
rugu "Leake County"
```

```
## # A tibble: 2,997 x 13
##    id    state stop_date county_name county_fips police_departme~
##    <fct> <fct> <fct>     <fct>             <int> <fct>
##  1 MS-2~ MS    2013-01-~ Leake Coun~       28079 Mississippi Hig~
##  2 MS-2~ MS    2013-01-~ Leake Coun~       28079 Mississippi Hig~
##  3 MS-2~ MS    2013-01-~ Leake Coun~       28079 Mississippi Hig~
##  4 MS-2~ MS    2013-01-~ Leake Coun~       28079 Mississippi Hig~
##  5 MS-2~ MS    2013-01-~ Leake Coun~       28079 Mississippi Hig~
##  6 MS-2~ MS    2013-01-~ Leake Coun~       28079 Mississippi Hig~
##  7 MS-2~ MS    2013-01-~ Leake Coun~       28079 Mississippi Hig~
##  8 MS-2~ MS    2013-01-~ Leake Coun~       28079 Mississippi Hig~
##  9 MS-2~ MS    2013-01-~ Leake Coun~       28079 Mississippi Hig~
## 10 MS-2~ MS    2013-01-~ Leake Coun~       28079 Mississippi Hig~
## # ... with 2,987 more rows, and 7 more variables: driver_gender <fct>,
## #   driver_birthdate <fct>, driver_race <fct>, violation_raw <fct>,
## #   officer_id <fct>, driver_age <int>, violation <fct>
```

```
filter(trafficstops,violation == "Speeding") # prikazujemo zaustavljanja zbog prebrz
e voznje
```

```
## # A tibble: 128,277 x 13
##    id    state stop_date county_name county_fips police_departme~
##    <fct> <fct> <fct>     <fct>             <int> <fct>
##  1 MS-2~ MS    2013-01-~ Pike County       28113 Mississippi Hig~
##  2 MS-2~ MS    2013-01-~ Hancock Co~       28045 Mississippi Hig~
##  3 MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
##  4 MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
##  5 MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
##  6 MS-2~ MS    2013-01-~ Grenada Co~       28043 Mississippi Hig~
##  7 MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
##  8 MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
##  9 MS-2~ MS    2013-01-~ Scott Coun~       28123 Mississippi Hig~
## 10 MS-2~ MS    2013-01-~ Wayne Coun~       28153 Mississippi Hig~
## # ... with 128,267 more rows, and 7 more variables: driver_gender <fct>,
## #   driver_birthdate <fct>, driver_race <fct>, violation_raw <fct>,
## #   officer_id <fct>, driver_age <int>, violation <fct>
```

```
filter(trafficstops,violation == "Speeding",driver_gender =="female") # prikazuje ze
nske vozace zaustavljene zgog prebrze voznje
```

```
## # A tibble: 58,959 x 13
##    id    state stop_date county_name county_fips police_departme~
##    <fct> <fct> <fct>     <fct>             <int> <fct>
##  1 MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
##  2 MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
##  3 MS-2~ MS    2013-01-~ Grenada Co~       28043 Mississippi Hig~
##  4 MS-2~ MS    2013-01-~ Wayne Coun~       28153 Mississippi Hig~
##  5 MS-2~ MS    2013-01-~ Coahoma Co~       28027 Mississippi Hig~
##  6 MS-2~ MS    2013-01-~ Quitman Co~       28119 Mississippi Hig~
##  7 MS-2~ MS    2013-01-~ Montgomery~       28097 Mississippi Hig~
##  8 MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
##  9 MS-2~ MS    2013-01-~ Lauderdale~       28075 Mississippi Hig~
## 10 MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
## # ... with 58,949 more rows, and 7 more variables: driver_gender <fct>,
```

```
## #   driver_birthdate <fct>, driver_race <fct>, violation_raw <fct>,
## #   officer_id <fct>, driver_age <int>, violation <fct>
```

Коришћењем функције $select()$ изабраћемо променљиве помоћу њиховог имена.

```
select(trafficstops, police_department, officer_id) # prikazujemo samo trazene kolon
e

## # A tibble: 211,211 x 2
##    police_department          officer_id
##    <fct>                      <fct>
##  1 Mississippi Highway Patrol J042
##  2 Mississippi Highway Patrol B026
##  3 Mississippi Highway Patrol M009
##  4 Mississippi Highway Patrol K035
##  5 Mississippi Highway Patrol D028
##  6 Mississippi Highway Patrol K023
##  7 Mississippi Highway Patrol K032
##  8 Mississippi Highway Patrol D021
##  9 Mississippi Highway Patrol R021
## 10 Mississippi Highway Patrol R021
## # ... with 211,201 more rows

select(trafficstops, starts_with("driver")) # prikazujemo kolone ciji nazivi pocinju
sa driver

## # A tibble: 211,211 x 4
##    driver_gender driver_birthdate driver_race driver_age
##    <fct>         <fct>            <fct>            <int>
##  1 male          1950-06-14       Black               63
##  2 male          1967-04-06       Black               46
##  3 male          1974-04-15       Black               39
##  4 male          1981-03-23       White               32
##  5 male          1992-08-03       White               20
##  6 female        1960-05-02       White               53
##  7 female        1953-03-16       White               60
##  8 female        1993-06-14       White               20
##  9 male          1947-12-11       White               65
## 10 male          1984-07-14       White               28
## # ... with 211,201 more rows

select(trafficstops, violation,everything()) # prkazujemo prvo prestup pa sve ostalo

## # A tibble: 211,211 x 13
##    violation id    state stop_date county_name county_fips police_departme~
##    <fct>     <fct> <fct> <fct>     <fct>             <int> <fct>
##  1 Seat belt MS-2~ MS    2013-01-~ Jones Coun~       28067 Mississippi Hig~
##  2 Careless~ MS-2~ MS    2013-01-~ Lauderdale~       28075 Mississippi Hig~
##  3 Speeding  MS-2~ MS    2013-01-~ Pike County       28113 Mississippi Hig~
##  4 Speeding  MS-2~ MS    2013-01-~ Hancock Co~       28045 Mississippi Hig~
##  5 Speeding  MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
##  6 Speeding  MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
##  7 Speeding  MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
##  8 Speeding  MS-2~ MS    2013-01-~ Grenada Co~       28043 Mississippi Hig~
##  9 Speeding  MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
## 10 Speeding  MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
## # ... with 211,201 more rows, and 6 more variables: driver_gender <fct>,
## #   driver_birthdate <fct>, driver_race <fct>, violation_raw <fct>,
## #   officer_id <fct>, driver_age <int>
```

Коришћењем функције $arrange()$ можемо променити редослед редова.

```
arrange(trafficstops, driver_gender, driver_race) #sortiramo prvo po polu, pa zatim
svaku od dve kategorije sortiramo po rasi
```

```
## # A tibble: 211,211 x 13
##    id    state stop_date county_name county_fips police_departme~
##    <fct> <fct> <fct>     <fct>              <int> <fct>
##  1 MS-2~ MS    2013-01-~ Scott Coun~       28123 Mississippi Hig~
##  2 MS-2~ MS    2013-01-~ Coahoma Co~       28027 Mississippi Hig~
##  3 MS-2~ MS    2013-01-~ Montgomery~       28097 Mississippi Hig~
##  4 MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
##  5 MS-2~ MS    2013-01-~ Lauderdale~       28075 Mississippi Hig~
##  6 MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
##  7 MS-2~ MS    2013-01-~ Scott Coun~       28123 Mississippi Hig~
##  8 MS-2~ MS    2013-01-~ Panola Cou~       28107 Mississippi Hig~
##  9 MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
## 10 MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
## # ... with 211,201 more rows, and 7 more variables: driver_gender <fct>,
## #   driver_birthdate <fct>, driver_race <fct>, violation_raw <fct>,
## #   officer_id <fct>, driver_age <int>, violation <fct>
```

```
arrange(trafficstops, driver_gender, desc(driver_age)) #sortiramo prvo po polu, pa z
atim po godinama opadajuce
```

```
## # A tibble: 211,211 x 13
##    id    state stop_date county_name county_fips police_departme~
##    <fct> <fct> <fct>     <fct>              <int> <fct>
##  1 MS-2~ MS    2014-07-~ Sunflower ~       28133 Mississippi Hig~
##  2 MS-2~ MS    2015-05-~ Tishomingo~       28141 Mississippi Hig~
##  3 MS-2~ MS    2015-07-~ Jefferson ~       28065 Mississippi Hig~
##  4 MS-2~ MS    2015-09-~ Lauderdale~       28075 Mississippi Hig~
##  5 MS-2~ MS    2016-03-~ Bolivar Co~       28011 Mississippi Hig~
##  6 MS-2~ MS    2013-09-~ Hancock Co~       28045 Mississippi Hig~
##  7 MS-2~ MS    2014-06-~ Chickasaw ~       28017 Mississippi Hig~
##  8 MS-2~ MS    2014-06-~ Tate County       28137 Mississippi Hig~
##  9 MS-2~ MS    2015-08-~ Attala Cou~       28007 Mississippi Hig~
## 10 MS-2~ MS    2015-09-~ Adams Coun~       28001 Mississippi Hig~
## # ... with 211,201 more rows, and 7 more variables: driver_gender <fct>,
## #   driver_birthdate <fct>, driver_race <fct>, violation_raw <fct>,
## #   officer_id <fct>, driver_age <int>, violation <fct>
```

Коришћењем функције $mutate()$ можемо креирати нове променљиве од постојећих.

```
#filtriramo tibble tako da nema NA vrednosti
trafficstops<-filter(trafficstops,!is.na(county_fips),!is.na(officer_id),!is.na(driv
er_age))
trafficstops<-mutate(trafficstops,adult=(ifelse(driver_age < 18, 1, 0)))
#dodajemo novu kolonu koja prikazuje da li je vozac punoletan
trafficstops
```

```
## # A tibble: 211,096 x 14
##    id    state stop_date county_name county_fips police_departme~
##    <fct> <fct> <fct>     <fct>              <int> <fct>
##  1 MS-2~ MS    2013-01-~ Jones Coun~       28067 Mississippi Hig~
##  2 MS-2~ MS    2013-01-~ Lauderdale~       28075 Mississippi Hig~
##  3 MS-2~ MS    2013-01-~ Pike County       28113 Mississippi Hig~
##  4 MS-2~ MS    2013-01-~ Hancock Co~       28045 Mississippi Hig~
##  5 MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
##  6 MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
##  7 MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
##  8 MS-2~ MS    2013-01-~ Grenada Co~       28043 Mississippi Hig~
##  9 MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
```

```
## 10 MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
## # ... with 211,086 more rows, and 8 more variables: driver_gender <fct>,
## #   driver_birthdate <fct>, driver_race <fct>, violation_raw <fct>,
## #   officer_id <fct>, driver_age <int>, violation <fct>, adult <dbl>
```

Илустрација оператора *cevi*:

Издвојићемо из базе возаче страрије од 85 година, и задржаћемо само колоне $violation\_raw, driver\_gender, driver\_race$.

```
senior_drivers <- trafficstops %>%
  filter(driver_age > 85) %>%
  select(violation_raw, driver_gender, driver_race)

senior_drivers # izdvajamo manju bazu podataka
```

```
## # A tibble: 3 x 3
##   violation_raw                                    driver_gender driver_race
##   <fct>                                            <fct>         <fct>
## 1 Seat belt not used properly as required          male          White
## 2 Speeding - Regulated or posted speed limit and~ male          White
## 3 Seat belt not used properly as required          male          Black
```

Сада ћемо нашој бази додати нову колону која ће прихазиватиу годину рођења возача.

```
trafficstops %>% mutate(birth_year = substring(driver_birthdate, 1, 4))
```

```
## # A tibble: 211,096 x 15
##    id    state stop_date county_name county_fips police_departme~
##    <fct> <fct> <fct>     <fct>             <int> <fct>
##  1 MS-2~ MS    2013-01-~ Jones Coun~       28067 Mississippi Hig~
##  2 MS-2~ MS    2013-01-~ Lauderdale~       28075 Mississippi Hig~
##  3 MS-2~ MS    2013-01-~ Pike County       28113 Mississippi Hig~
##  4 MS-2~ MS    2013-01-~ Hancock Co~       28045 Mississippi Hig~
##  5 MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
##  6 MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
##  7 MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
##  8 MS-2~ MS    2013-01-~ Grenada Co~       28043 Mississippi Hig~
##  9 MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
## 10 MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
## # ... with 211,086 more rows, and 9 more variables: driver_gender <fct>,
## #   driver_birthdate <fct>, driver_race <fct>, violation_raw <fct>,
## #   officer_id <fct>, driver_age <int>, violation <fct>, adult <dbl>,
## #   birth_year <chr>
```

Учитаћемо *lubridate* пакет, како би наш низ претворили у стварни формат датума. Користићемо $year()$ функцију за издвајање године.

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

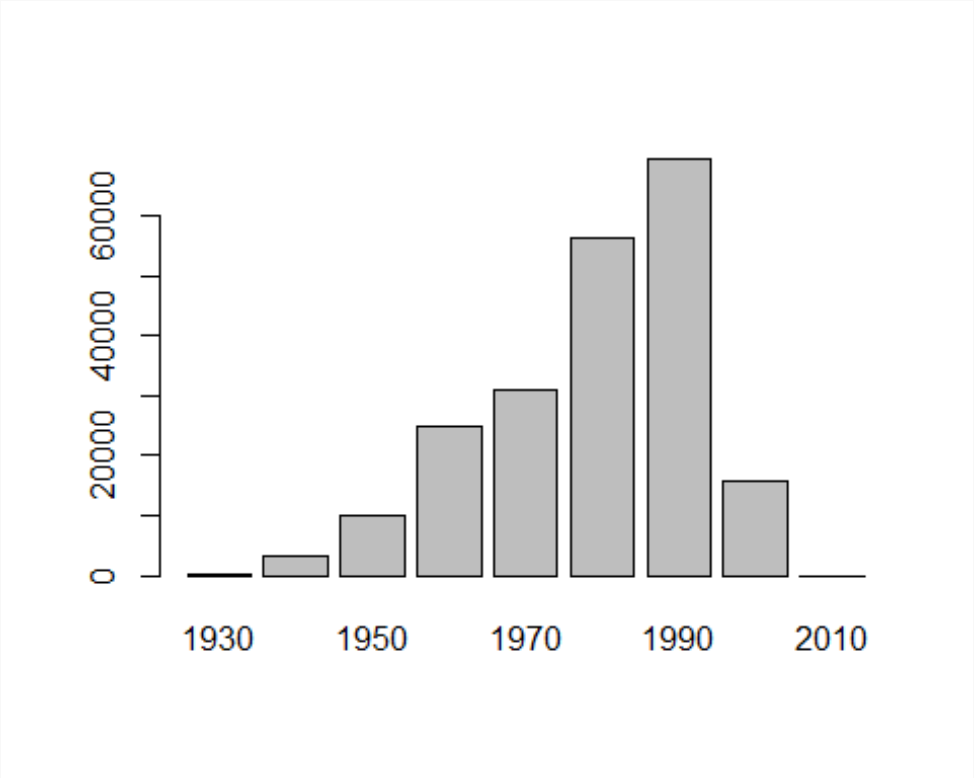
```
trafficstops %>%
  mutate(birth_date = ymd(driver_birthdate),
         birth_year = year(driver_birthdate),
```

```
        birth_cohort = round(birth_year/10)*10) %>%
  head()
```

```
## # A tibble: 6 x 17
##   id    state stop_date county_name county_fips police_departme~
##   <fct> <fct> <fct>     <fct>             <int> <fct>
## 1 MS-2~ MS    2013-01-~ Jones Coun~       28067 Mississippi Hig~
## 2 MS-2~ MS    2013-01-~ Lauderdale~       28075 Mississippi Hig~
## 3 MS-2~ MS    2013-01-~ Pike County       28113 Mississippi Hig~
## 4 MS-2~ MS    2013-01-~ Hancock Co~       28045 Mississippi Hig~
## 5 MS-2~ MS    2013-01-~ Holmes Cou~       28051 Mississippi Hig~
## 6 MS-2~ MS    2013-01-~ Jackson Co~       28059 Mississippi Hig~
## # ... with 11 more variables: driver_gender <fct>, driver_birthdate <fct>,
## #   driver_race <fct>, violation_raw <fct>, officer_id <fct>,
## #   driver_age <int>, violation <fct>, adult <dbl>, birth_date <date>,
## #   birth_year <dbl>, birth_cohort <dbl>
```

Такође, можемо и нацртати хистограм кохорте о рођењу возача :

```
trafficstops %>%
  mutate(birth_date = ymd(driver_birthdate),
         birth_year = year(driver_birthdate),
         birth_cohort = round(birth_year/10)*10,
         birth_cohort = factor(birth_cohort)) %>%
  select(birth_cohort) %>%
  plot()
```



Комбиновањем функција $groupe\_by()$ и $summarise()$ податке можемо поделити у групе, применти неке анализе на сваку групу, а затим комбиновати резултате.

Дакле, за преглед просечне старости возача црне и беле расе:

```
trafficstops %>%
  group_by(driver_race) %>%
  summarize(mean_age = mean(driver_age, na.rm=TRUE))
```

```
## Warning: Factor `driver_race` contains implicit NA, consider using
## `forcats::fct_explicit_na`

## # A tibble: 3 x 2
##   driver_race mean_age
##   <fct>          <dbl>
## 1 Black           34.2
## 2 White           36.2
## 3 <NA>            34.5
```

Преглед просечне старости возача за различите округе:

```
trafficstops %>%
  group_by(county_name) %>%
  summarize(mean_age = mean(driver_age, na.rm=TRUE))

## # A tibble: 82 x 2
##    county_name      mean_age
##    <fct>               <dbl>
##  1 Adams County         37.6
##  2 Alcorn County        33.8
##  3 Amite County         39.6
##  4 Attala County        37.5
##  5 Benton County        33.5
##  6 Bolivar County       35.0
##  7 Calhoun County       34.1
##  8 Carroll County       35.0
##  9 Chickasaw County     34.0
## 10 Choctaw County       36.4
## # ... with 72 more rows
```

Такође, можемо груписати на основу више захтева истовремено:

```
trafficstops %>%
  group_by(county_name, driver_race) %>%
  summarize(mean_age = mean(driver_age, na.rm=TRUE))
```

```
## Warning: Factor `driver_race` contains implicit NA, consider using
## `forcats::fct_explicit_na`

## # A tibble: 177 x 3
## # Groups:   county_name [82]
##    county_name   driver_race mean_age
##    <fct>         <fct>          <dbl>
##  1 Adams County  Black           36.2
##  2 Adams County  White           40.0
##  3 Alcorn County Black           34.6
##  4 Alcorn County White           33.6
##  5 Amite County  Black           37.5
##  6 Amite County  White           42.1
##  7 Amite County  <NA>            24
##  8 Attala County Black           36.4
##  9 Attala County White           38.6
## 10 Benton County Black           34.7
## # ... with 167 more rows
```

Ако желимо да уклонимо NA вредности:

```
trafficstops %>%
  filter(!is.na(driver_race)) %>%
```

```
  group_by(county_name, driver_race) %>%
  summarize(mean_age = mean(driver_age, na.rm=TRUE))
```

```
## # A tibble: 163 x 3
## # Groups:   county_name [82]
##    county_name   driver_race mean_age
##    <fct>         <fct>          <dbl>
##  1 Adams County  Black           36.2
##  2 Adams County  White           40.0
##  3 Alcorn County Black           34.6
##  4 Alcorn County White           33.6
##  5 Amite County  Black           37.5
##  6 Amite County  White           42.1
##  7 Attala County Black           36.4
##  8 Attala County White           38.6
##  9 Benton County Black           34.7
## 10 Benton County White           32.0
## # ... with 153 more rows
```

Када бисмо желели да видимо колико саобраћајних заустављања је сваки службеник забележио користили бисмо функцију $tally()$:

```
trafficstops %>%
  group_by(officer_id) %>%
  tally()
```

```
## # A tibble: 896 x 2
##    officer_id     n
##    <fct>      <int>
##  1 A003           1
##  2 A004           5
##  3 A005           4
##  4 A006           3
##  5 A007         128
##  6 A008           9
##  7 A009          83
##  8 A011           5
##  9 A012           1
## 10 A013           1
## # ... with 886 more rows
```

Сада ћемо средити нашу базу података коришћењем пакета $tidyr$ који је део класе $tidyverse$.

```
library(tidyr)
```

Прво, помоћу $dplyr$, креирајмо оквир података са средњом старошћу сваког возача према полу и округу:

```
trafficstops_ma <- trafficstops %>%
  filter(!is.na(driver_gender)) %>%
  group_by(county_name, driver_gender) %>%
  summarize(mean_age = mean(driver_age, na.rm = TRUE))

head(trafficstops_ma)
```

```
## # A tibble: 6 x 3
## # Groups:   county_name [3]
##   county_name   driver_gender mean_age
##   <fct>         <fct>            <dbl>
```

```
## 1 Adams County  female            36.7
## 2 Adams County  male              38.4
## 3 Alcorn County female            33.3
## 4 Alcorn County male              34.1
## 5 Amite County  female            38.3
## 6 Amite County  male              40.3
```

Видимо да су опсервације разбацане на више редова, па ћемо користити ширење како би сваку опсервацију проширили на два реда(мушки и женски пол). То радимо коришћењем функције $spread$ из $tidyr$ пакета. Потребна су нам два параметра: колона која садржи име променљивих(key колона) и колона која садржи вредности из више променљивих($value$ колона).

```
trafficstops_ma_wide <- trafficstops_ma %>%
  spread(key=driver_gender,value =  mean_age)

head(trafficstops_ma_wide)
```

```
## # A tibble: 6 x 3
## # Groups:   county_name [82]
##   county_name    female  male
##   <fct>           <dbl> <dbl>
## 1 Adams County     36.7  38.4
## 2 Alcorn County    33.3  34.1
## 3 Amite County     38.3  40.3
## 4 Attala County    36.7  38.1
## 5 Benton County    32.1  34.4
## 6 Bolivar County   33.2  36.3
```

Сада можемо да упоредимо просечну старост мушких возача у односу на женске возаче. Користећи разлику у годинама, промаћићемо округе са највећим и најмањим бројем тј. негативан број ће нам представљати да су женски возачи у просеку старији од муских, а позитиван број ће значити да су мушки возачи у просеку старији од женских.

```
trafficstops_ma_wide %>%
  mutate(agediff = male - female) %>%
  ungroup() %>%
  filter(agediff %in% range(agediff))
```

```
## # A tibble: 2 x 4
##   county_name      female  male agediff
##   <fct>             <dbl> <dbl>   <dbl>
## 1 Neshoba County     35.1  31.1   -3.94
## 2 Yalobusha County   33.4  39.4    5.99
```

Променљиве од интереса ћемо приказати различитим графицима из пакета $ggplot2$.

```
library(ggplot2)
MS_county_stops <- trafficstops_ma_wide
```

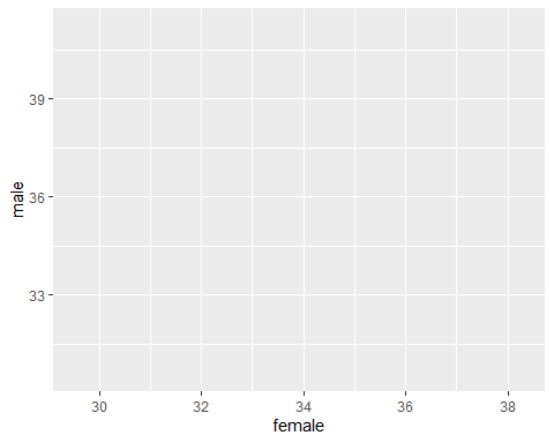
Градимо график корак по корак:

- Цртамо празан $ggplot$ наводећи само базу коју користимо

```
ggplot(data = MS_county_stops)
```

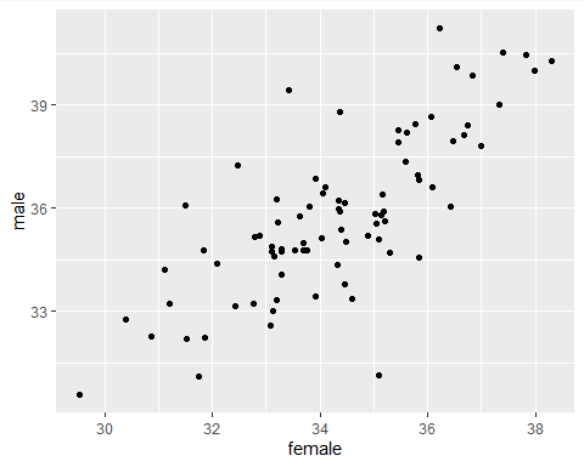
- Прецизирамо шта се налази на x,а шта на y −оси

```
ggplot(data = MS_county_stops, aes(x = female, y = male))
```



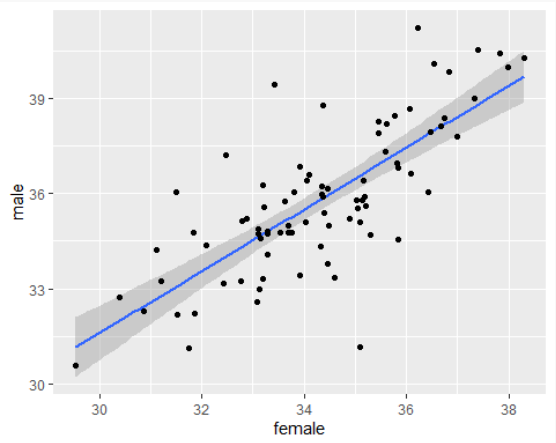
- Додајемо $scatterplot$ на празан $ggplot$ додавањем тачака користећи $geom$ слој који се назива $geom\_point$

```
ggplot(data = MS_county_stops, aes(x = female, y = male)) +
  geom_point()
```



Како би олакшали рад доделићемо име графику и додаћемо праву која најбоље одговара подацима коришћењем линеарне регресије.

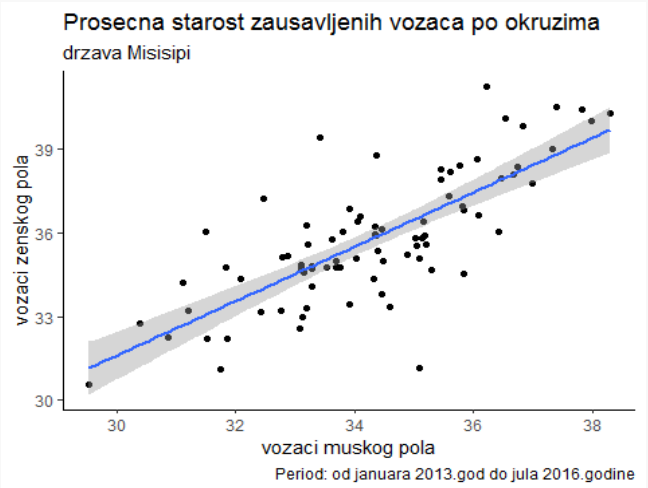
```
MS_plot <- ggplot(data = MS_county_stops, aes(x = female, y = male)) + geom_point()
+ geom_smooth(method="lm")
MS_plot
```



Као што видимо $ggplot$ график се гради корак по корак додавањем нових елемената (слојева) помоћу знака $+$.

Хајде сада да додамо наслов графику и називе осама.

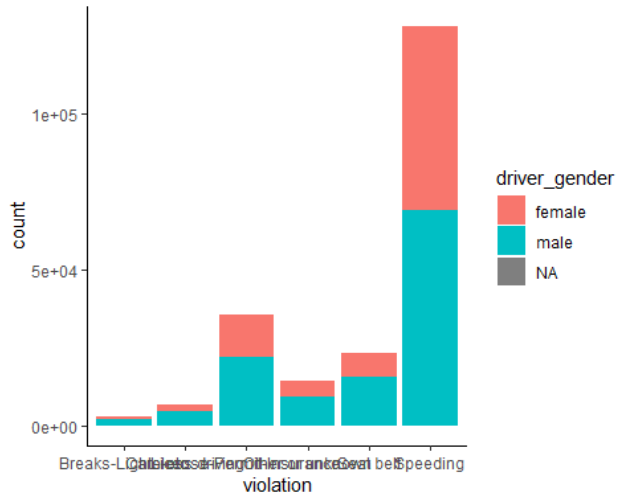
```
MS_plot <- MS_plot+labs(title = "Prosecna starost zausavljenih vozaca po okruzima",
                        subtitle = "drzava Misisipi",
                        y="vozaci zenskog pola",x="vozaci muskog pola",
                        caption="Period: od januara 2013.god do jula 2016.godine")
theme_set(theme_classic())
MS_plot
```



Вратићемо се сада на нашу почетну базу података о заустављању возача. Ако желимо да видимо колико има преступа сваке врсте, можемо нацртати барплот.

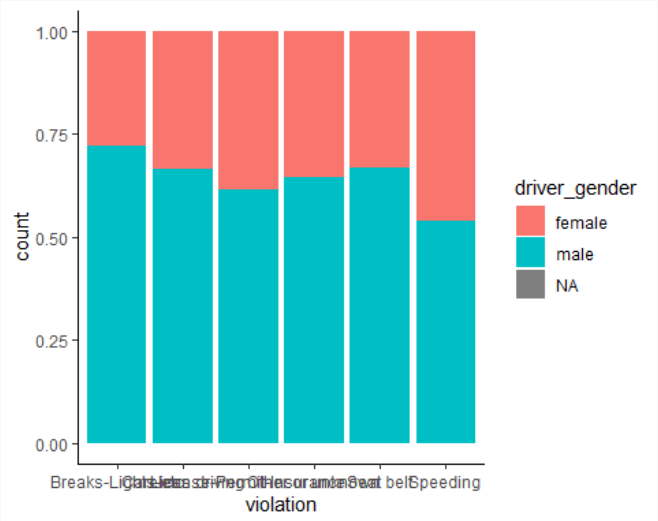
Уместо да све буде исте боје, ми смо бојили различитим бојама различит пол.

```
ggplot(trafficstops, aes(violation)) +
  geom_bar(aes(fill = driver_gender))
```



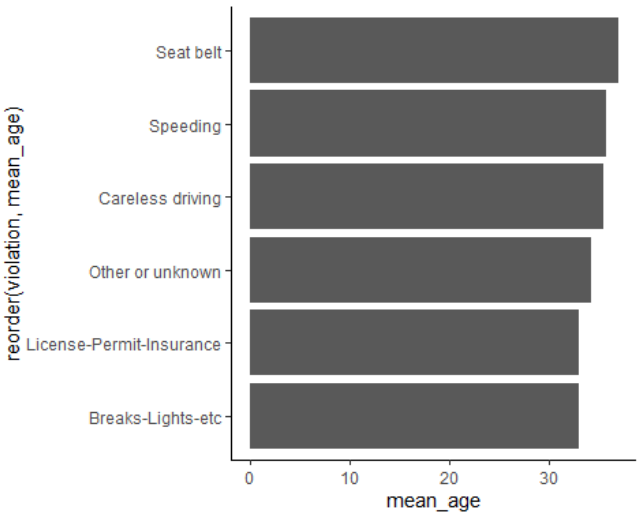
Ако желимо да видимо пропорције унутар сваке категорије, можемо поставити параметар позиције.

```
ggplot(trafficstops, aes(violation)) +
  geom_bar(aes(fill = driver_gender), position = "fill")
```



Како бисмо креирали *bar chart* прво морамо да креирамо резиме статистике, а затим да је убацимо у *ggplot* да бисмо приказали вредности које смо израчунали.Користићемо функцију за окретање координате.

```
trafficstops %>%
  group_by(violation) %>%
  summarize(mean_age = mean(driver_age, na.rm = TRUE)) %>%
  ggplot(aes(x = reorder(violation, mean_age), y = mean_age)) +
  geom_col() +
  coord_flip()
```



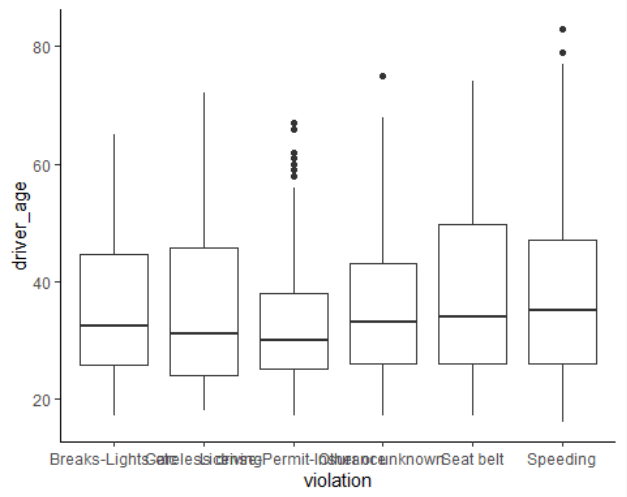
$Box - plot$ је одличан алат за проучавање расподеле.

Издвојићемо и радити само за округ $Yazoo$.

```
Yazoo_stops <- filter(trafficstops, county_name == "Yazoo County")
```

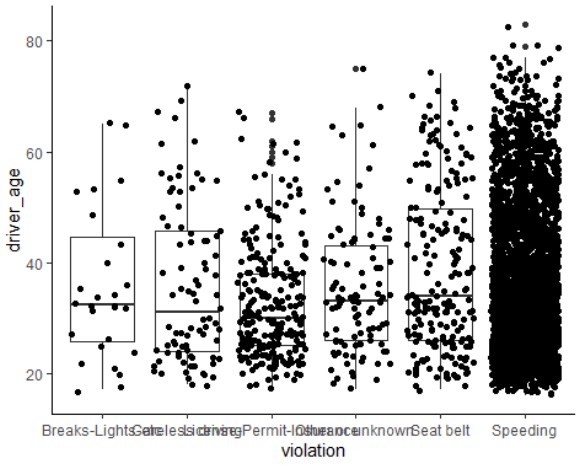
Можемо да прикажемо расподелу узраста возача у оквиру сваког саобрачајног преступа:

```
ggplot(data = Yazoo_stops, aes(x = violation, y = driver_age)) +
  geom_boxplot()
```



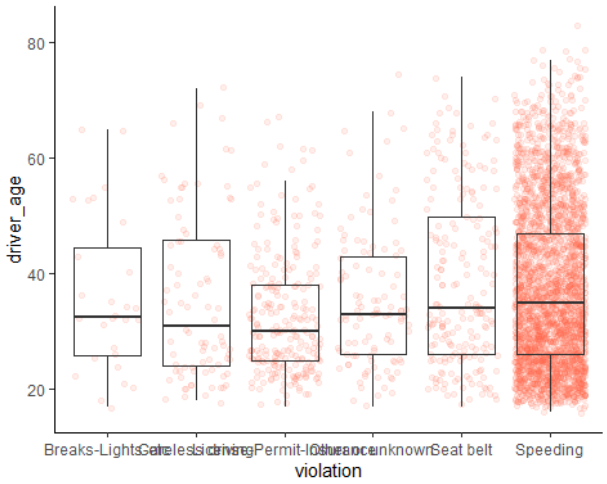
Додавањем тачака, имамо бољу представу о броју мерења и њиховој расподели.

```
ggplot(data = Yazoo_stops, aes(x = violation, y = driver_age)) +
  geom_boxplot() +
  geom_jitter()
```



Ово изгледа прилично неуредно, па ћемо променити транспаретност,као и боју.

```
ggplot(data = Yazoo_stops, aes(x = violation, y = driver_age)) +
  geom_jitter(alpha = 0.1, color = "tomato") +
  geom_boxplot(alpha = 0)
```



Приказаћемо податке временских серјија.

Да бисмо ствари учинили мало лакшим, прво ћемо конвертовати $stop\_date$ колону коју планирамо да користимо у формат датума помоћу $wday$ функције за издвајање и додавање нове колоне са радним даном за сваки од тих датума.За боље разумевање обележићемо радне дане.

```
trafficstops <- trafficstops %>%
  mutate(stop_date = ymd(stop_date),
         wk_day = wday(stop_date, label = TRUE))
```

Рачунамо број прекршаја по радном дану.

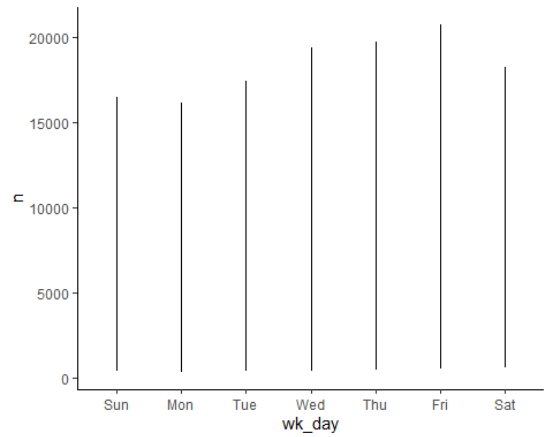
```
trafficstops %>%
  count(wk_day, violation)
```

```
## # A tibble: 42 x 3
##    wk_day violation                   n
##    <ord>  <fct>                   <int>
##  1 Sun    Breaks-Lights-etc         363
##  2 Sun    Careless driving          804
##  3 Sun    License-Permit-Insurance 4176
```

```
##  4 Sun    Other or unknown          1586
##  5 Sun    Seat belt                 2312
##  6 Sun    Speeding                 16518
##  7 Mon    Breaks-Lights-etc          337
##  8 Mon    Careless driving           828
##  9 Mon    License-Permit-Insurance  4086
## 10 Mon    Other or unknown          1571
## # ... with 32 more rows
```

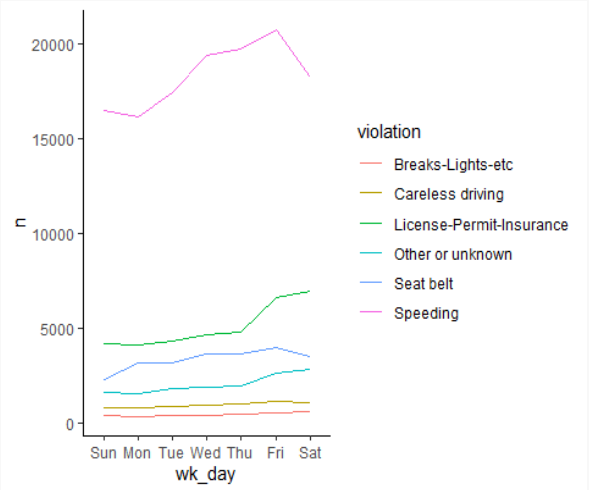
Прказаћемо ово сада као график временске серије.

```
trafficstops %>%
  count(wk_day, violation) %>%
  ggplot(aes(wk_day, n)) +
  geom_line()
```



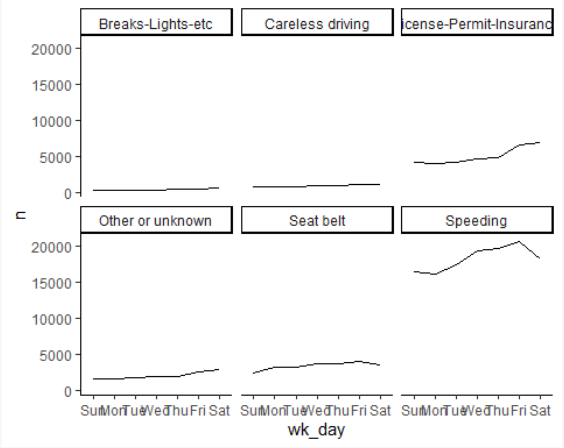
Нисмо баш добили оно што смо очекивали. Оно што нам график приказује је распон свих вредности за сваку годину у вертикалној линији. Морамо задати да се повуће посебна линија за сваки прекршај. Како бисмо боље разликовали прекршаје додаћемо боје.

```
trafficstops %>%
  count(wk_day, violation) %>%
  ggplot(aes(wk_day, n, group = violation, color = violation)) +
  geom_line()
```



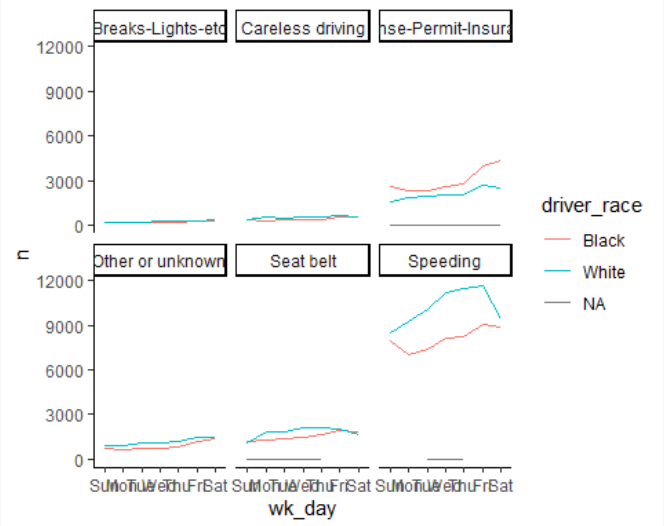
Уместо да користимо бојење за одвајање различитих преступа,за сваки преступ можемо направити посебан график временске серије.

```
trafficstops %>%
  count(wk_day, violation) %>%
  ggplot(aes(wk_day, n, group = violation)) +
  geom_line() +
  facet_wrap(~ violation)
```



Сада бисмо желели да поделимо линију на свакој парцели према раси возача.

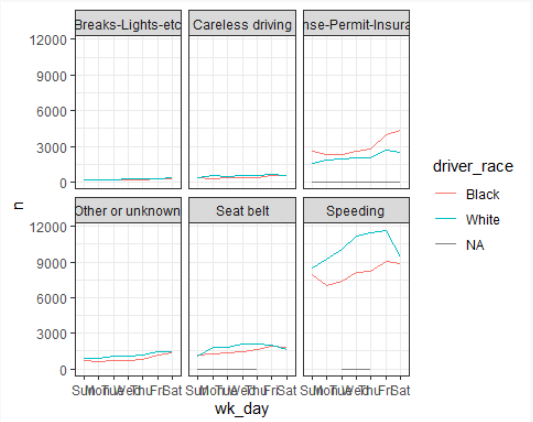
```
trafficstops %>%
  count(wk_day, violation, driver_race) %>%
  ggplot(aes(wk_day, n, color = driver_race, group = driver_race)) +
  geom_line() +
  facet_wrap(~ violation)

## Warning: Factor `driver_race` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```



Додаћемо тему да променимо позадину слике у белу.

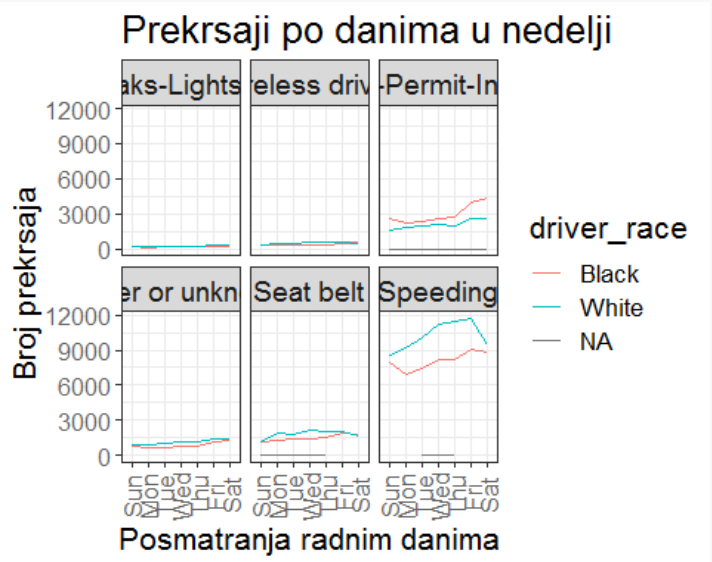
```
stops_facet_plot <- trafficstops %>%
  count(wk_day, violation, driver_race) %>%
  ggplot(aes(wk_day, n, color = driver_race, group = driver_race)) +
  geom_line() +
  facet_wrap(~ violation)

## Warning: Factor `driver_race` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```

```
stops_facet_plot +
  theme_bw()
```



Променићемо још имена осама и додаћемо наслов.

```
stops_facet_plot +
  labs(title = 'Prekrsaji po danima u nedelji',
       x = 'Posmatranja radnim danima',
       y = 'Broj prekrsaja') +
  theme_bw() +
  theme(axis.text.x = element_text(colour="grey40", size=12, angle=90, hjust=.5,
vjust=.5),
        axis.text.y = element_text(colour="grey40", size=12),
        strip.text = element_text(size=14),
        text = element_text(size=16))
```



Учитаћемо нову базу, која садржи укупан број становника по округу, при чему су дати и подаци за сваку расу појединачно. Подаци о становништву су процењене вредности петогодишње Америчке анкете 2011-2015:

```
MS_bw_pop <- read.csv("C:/Users/Korisnik/Desktop/SS4 seminarski/baza2.csv")
head(MS_bw_pop)
```

```
##              County  FIPS black_pop white_pop bw_pop
## 1      Jones County 28067     19711     47154  66865
## 2 Lauderdale County 28075     33893     43482  77375
## 3       Pike County 28113     21028     18282  39310
## 4    Hancock County 28045      4172     39686  43858
## 5     Holmes County 28051     15498      3105  18603
## 6    Jackson County 28059     30704    101686 132390
```

Прво ћемо да избројимо сва саобраћајна заустављања по окрузима.

```
trafficstops %>%
  group_by(county_name) %>%
  summarise(n_stops = n())

## # A tibble: 82 x 2
##    county_name      n_stops
##    <fct>              <int>
##  1 Adams County         942
##  2 Alcorn County       3344
##  3 Amite County        2920
##  4 Attala County       4202
##  5 Benton County        214
##  6 Bolivar County      4524
##  7 Calhoun County      1658
##  8 Carroll County      1787
##  9 Chickasaw County    3869
## 10 Choctaw County       613
## # ... with 72 more rows
```

Сада ћемо то пренети у нашу следећу операцију где обједињујемо две табеле, користићемо $left\_join$.

```
trafficstops %>%
  group_by(county_name) %>%
  summarise(n_stops = n()) %>%
  left_join(MS_bw_pop, by = c("county_name" = "County")) %>%
  head()

## # A tibble: 6 x 6
##   county_name    n_stops  FIPS black_pop white_pop bw_pop
##   <fct>            <int> <int>     <int>     <int>  <int>
## 1 Adams County       942 28001     17757     12856  30613
## 2 Alcorn County     3344 28003      4281     31563  35844
## 3 Amite County      2920 28005      5416      7395  12811
## 4 Attala County     4202 28007      8194     10649  18843
## 5 Benton County      214 28009      3078      5166   8244
## 6 Bolivar County    4524 28011     21648     11197  32845
```

Спајамо сада базе по називу округа, и задржаћемо само преклапања.

```
trafficstops %>%
  inner_join(MS_bw_pop, by = c("county_name" = "County"))

## # A tibble: 211,096 x 19
##    id    state stop_date  county_name county_fips police_departme~
##    <fct> <fct> <date>     <fct>             <int> <fct>
##  1 MS-2~ MS    2013-01-01 Jones Coun~       28067 Mississippi Hig~
##  2 MS-2~ MS    2013-01-01 Lauderdale~       28075 Mississippi Hig~
##  3 MS-2~ MS    2013-01-01 Pike County       28113 Mississippi Hig~
##  4 MS-2~ MS    2013-01-01 Hancock Co~       28045 Mississippi Hig~
```

```
##  5 MS-2~ MS    2013-01-01 Holmes Cou~       28051 Mississippi Hig~
##  6 MS-2~ MS    2013-01-01 Jackson Co~       28059 Mississippi Hig~
##  7 MS-2~ MS    2013-01-01 Jackson Co~       28059 Mississippi Hig~
##  8 MS-2~ MS    2013-01-01 Grenada Co~       28043 Mississippi Hig~
##  9 MS-2~ MS    2013-01-01 Holmes Cou~       28051 Mississippi Hig~
## 10 MS-2~ MS    2013-01-01 Holmes Cou~       28051 Mississippi Hig~
## # ... with 211,086 more rows, and 13 more variables: driver_gender <fct>,
## #   driver_birthdate <fct>, driver_race <fct>, violation_raw <fct>,
## #   officer_id <fct>, driver_age <int>, violation <fct>, adult <dbl>,
## #   wk_day <ord>, FIPS <int>, black_pop <int>, white_pop <int>,
## #   bw_pop <int>
```

Бројимо саобраћајна заустављања по расама и окрузима, затим сваку обсервацију проширимо на два реда на основу расе.

```
# Brojimo saobracajna zaustavljanja po drzavama i rasama
trafficstops_2 <- trafficstops %>%
  filter(!is.na(driver_race)) %>%
  group_by(county_name,driver_race) %>%
  summarise(n_stops = n())

# Svaku opservaciju cemo prosiriti na dva reda-black i white rasa
trafficstops_race <- trafficstops_2 %>%
  spread(key=driver_race,value =  n_stops)

head(trafficstops_race)
```

```
## # A tibble: 6 x 3
## # Groups:   county_name [82]
##   county_name    Black White
##   <fct>          <int> <int>
## 1 Adams County     583   359
## 2 Alcorn County    468  2876
## 3 Amite County    1589  1330
## 4 Attala County   2096  2106
## 5 Benton County    121    93
## 6 Bolivar County  3162  1362
```

```
# Rezultat je broj zausavljenih crnaca, odnosno belaca po okruzima
```

Сада ћемо спојити новодобијену базу и базу која садржи број становника по окрузима.

```
trafficstops_race %>%
  group_by(county_name) %>%
  full_join(MS_bw_pop, by = c("county_name" = "County")) %>%
  head()
```

```
## # A tibble: 6 x 7
## # Groups:   county_name [82]
##   county_name    Black White  FIPS black_pop white_pop bw_pop
##   <fct>          <int> <int> <int>     <int>     <int>  <int>
## 1 Adams County     583   359 28001     17757     12856  30613
## 2 Alcorn County    468  2876 28003      4281     31563  35844
## 3 Amite County    1589  1330 28005      5416      7395  12811
## 4 Attala County   2096  2106 28007      8194     10649  18843
## 5 Benton County    121    93 28009      3078      5166   8244
## 6 Bolivar County  3162  1362 28011     21648     11197  32845
```

Други задатак

Одабрати текст по избору и на њему илустровати основне функције за рад са стринговима. Требало би да подаци буду интерпретабилни и знимљиви колико је то могуће (потрудите се да формирате неке мало сложеније регуларне изразе).

На почетку ћемо учитати пакет за рад са стринговима.

```
library(stringr)
```

Учитавамо и текст на коме ћемо илустровати основне функције за рад са стринговима.Користићемо једностуке знакове наводника, зато што у тексту има цитата за које користимо двоструке наводнике.

```
# Ucitavamo tekst kao string
string <- 'Paradoksi kretanja
Ahil i kornjaca
"U utrci, najbrzi trkac nikada ne moze prestici najsporijeg, zato sto gonitelj prvo
mora doci do tacke odakle je gonjeni posao, pa prema tome najsporiji uvijek ima pred
nost."-Aristotelova Fizika VI:9, 239b15
Zamislite da Ahil trci protiv kornjace. Ahil trci 10 puta brze od kornjace, ali poci
nje od tacke A, 100 metara iza kornjace koja je u tacki K1 (kornjaci , koja je spori
ja, data je prednost). Da bi prestigao kornjacu, Ahil mora prvo doci do tacke K1. Me
dutim, kada je Ahil stigao do tacke K1, kornjaca je presla 10 metara i dosla do tack
e K2. Ponovo Ahil trci do K2. Ali, kao i prije, kada je presao 10 metara kornjaca je
metar ispred njega, kod tacke K3, i tako dalje (kornjaca ce uvijek imati prednost na
d Ahilom, bez obzira na to koliko mala ona bila). Prema tome Ahil nikada ne moze pre
stici kornjacu.
A---------------------------K1----------------K2---K3
Paradoks dihotomije
"Kretanje je nemoguce jer ono sto je u pokretu mora prvo preci pola puta prije nego
sto stigne do cilja".-Aristotelova Fizika VI:9, 239b10
Zamislite stvar koja treba ici od tacke A do tacke B. Da bi dosla do tacke B stvar p
rvo mora doci do srednje tacke B1 koja je izmedu tacaka A i B. Ali, prije nego sto s
e ovo dogodi stvar mora doci do tacke B2 koja je izmedu tacaka A i B1. Slicno, prije
nego sto moze i to uraditi, mora prvo doci do tacke B3 koja je izmedu A i B2, i tako
dalje. Prema tome kretanje nikada ne moze poceti.
A-----B3-----B2-----------B1------------------------B

Paradoks strijele
Zenon je dokazivao da je strijela u letu nepokretna.
"Ako je sve nepomicno sto zauzima prostor, i ako sve sto je u pokretu zauzima takav
prostor u nekom vremenu, onda je leteca strijela nepokretna."-Aristotelova Fizika VI
:9, 239b5
Zamislite da strijela leti neprestano naprijed, tokom jednog vremenskog intervala. U
zmite svaki momenat u tom vremenskom intervalu. Nemoguce je da se strijela mice u ta
kvom momentu, jer trenutak ima trajanje 0, i strijela ne moze biti na dva mjesta u i
sto vrijeme. Prema tome, u svakom trenutku je strijela nepomicna, i tako strijela je
nepomicna tokom citavog intervala.

Predlozena rjesenja za Ahila i kornjacu

Aristotel je istakao da kao sto se udaljenost smanjuje, vrijeme potrebno da se ta ud
```

```
aljenost prede takode se smanjuje.Takav pristup rijesavanju paradoksa bi doveo do de
manta tvrdnje da je potrebno beskonacno mnogo vremena da se prede preko beskonacno m
nogo udaljenosti, iako neki to spore.Prije 212. p. n. e., Arhimed je razvio metod da
izvede konacni odgovor za beskonacno mnogo clanova koji postaju progresivno manji. T
eoreme su razvijene u modernijim oblicima da bi postigle isti rezultat, ali sa tacni
jom metodom za dokazivanje. Ove metode dozvoljavaju konstrukciju rijesenja koje kazu
da (pod normalnim uslovima) ako se udaljenosti stalno smanjuju, vrijeme je konacno.O
va rijesenja su u biti geometrijski nizovi.
Predlozena rjesenja za paradoks dihotomije
Aristotel je istakao da kao sto se udaljenost smanjuje, vrijeme potrebno da se ta ud
aljenost prede takode se smanjuje. Takav pristup rijesavanju paradoksa bi doveo do d
emanta tvrdnje da je potrebno beskonacno mnogo vremena da se prede preko beskonacno
mnogo udaljenosti.
Predlozena rjesenja za paradoks strijele
Paradoks o strijeli postavlja pitanja o prirodi kretanja koja nisu odgovorena na mat
ematicki nacin, kao u slucaju Ahila i kornjace i Dihotomije.Ovaj paradoks se moze ri
jesiti matematicki na slijedeci nacin: u granicnoj vrijednosti, duzina momenta tezi
nuli, trenutacna stopa mijenjanja ili brzine (koja je kolicnik predenog puta u odred
enom vremenu) ne mora teziti nuli. Ova nenultna granicna vrijednost je brzina strije
le u trenutku.Problem sa racunskim rijesenjem je taj da racunska radnja moze opisati
samo kretanje dok se granicna vrijednost priblizava, bazirano na vanjskoj observacij
i da se strijela krece naprijed. Medutim, u Zenonovom paradoksu, koncepti kao brzina
gube svoje znacenje i nepostoji cinilac, koji nije pod djelovanjem paradoksa, koji b
i mogao strijeli omoguciti letenje.Drugo glediste je to da premisa kaze da je u svak
om trenutku, strijela nepomicna. Medutim, ne kretati se- je relativan pojam. Niko ne
moze suditi, posmatrajuci jedan trenutak, da strijela stoji u mjestu. Tacnije, potre
bni su drugi, slicni trenuci koji bi odredili, poredeci se sa drugim trenucima, da j
e strijela u jednom trenutku nepomicna. Prema tome, u poredenju sa drugim trenucima,
strijela bi bila na drugom mjestu nego sto je bila i sto ce biti u vremenu prije i p
oslije. Uzevsi ovo u obzir, strijela se krece.
'

# Stampani prikaz stringa (nije isti kao sam string, jer štampani prikaz prikazuje i
escape karaktere)

writeLines(string)

# Koliko karaktera ima u stringu

str_length(string)

# Ispis stringa, tako da sve bude ispisano malim slovima

str_to_lower(string)

# Iz stringa izdvajamo podstringove oblike "Aristotelova Fizika "

str_view_all(string, "Aristotelova Fizika ")

# Pravimo podstring od 800 do 1000 karaktera

substr(string,800,1000 )

# Menjamo rec paradoks sa problem

str_replace(string, "paradoks", "problem")

# Izdvajanje podudaranja

# Recimo da zelimo da pronađemo sve recenice koje sadrze imena

# Prvo kreiramo vektor koji sadrži imena , a zatim ga pretvaramo u jedan regularni i
zraz
```

```r
names <- c("Ahil","Aristotel","Dihotomije","Zenon")

names_match <- str_c(names, collapse = "|")

names_match

# Sada mozemo da izaberemo recenice koje sadrze imena, a zatim i da je izvucemo

has_names <- str_subset(string, names_match)

matches <- str_extract_all(has_names,names_match)

head(matches)

# Funkcija strsplit vraca listu reci na koje je podelila string

podeljen_string<-strsplit(string," ")[[1]]

# Od liste pravimo vektor reci koji i dalje ima posebne karaktere, tj. nije bas vekt
or reci

reci<-unlist(podeljen_string, recursive = TRUE, use.names = TRUE)

# Izbacujemo sve zareze iz vektora reci, ti zarezi se nalaze spojeni uz prethodnu re
c, i analogno za \n, \. i \?

reci<-str_remove_all(reci, ",")

reci<-str_remove_all(reci, "\\n")

reci<-str_remove_all(reci, "\\.")

reci<-str_remove_all(reci, "\\?")

# Broj reci u tekstu je zapravo duzina vektora reci

broj_reci<-length(reci)

# Zelimo da izdvojimo sve reci koje sadrze samo jedan samoglasnik

# Posmatramo tri slucaja

# rec pocinje sa samoglasnikom i ima odredjen broj suglasnika,tacno jedan samoglasni
k a zatim sve suglasnike

# rec pocinje sa samoglasnikom i sve ostale suglasnike

# rec ima samo suglasnike i na kraju samo samoglasnik

str_view_all(reci, "(^[^aeiou][^aeiou]{0,}+[aeiou]{1}+[^aeiou]{0,}[^aeiou]$)|(^[aeio
u][^aeiou]{0,}[^aeiou]$)|(^[^aeiou][^aeiou]{0,}[aeiou]$)")

# Broj reci koje se zavrsavaju na a

sum(str_detect(reci, "a$"))

# prosecan broj samoglasnika u recima

mean(str_count(reci, "[aeiou]"))

# Ilustrujemo rad sa tibblovima i stringovima

df <- tibble(

  reci = reci,
```

```
  i = seq_along(reci)

)

df %>%

  filter(str_detect(reci, "a$")) #izdvaja sve reci koje se zavrsavaju na a

df %>%

  mutate(

    sa_m = str_count(reci, "m"), #izdvaja koliko karaktera u reci je m, a koliko nij
e d

    nema_d= str_count(reci, "[^d]")

  )
```

Трећи задатак

Нека су X и Y случајне величине са заједничком расподелом:

$X\backslash Y$	0	1
0	0.5	0.1
1	0.3	0.1

Израчунати условне расподеле, а затим, користећи $Gibbs$ –ово узорковање, извадти узорак обима 10000 из заједничке расподеле.

(НАПОМЕНА: Узорачка расподела треба да буде приближна правој расподели датој у табели!)

X и Y су случајне величине са расподелом веоватноћа $P(X = x, Y = y) = p_{X,Y}(x, y)$ датој у табели.

Дакле, из табеле имамо да важи:

$$p_{X,Y}(0,0) = P(X = 0, Y = 0) = 0.5$$

$$p_{X,Y}(0,1) = P(X = 0, Y = 1) = 0.1$$

$$p_{X,Y}(1,0) = P(X = 1, Y = 0) = 0.3$$

$$p_{X,Y}(1,1) = P(X = 1, Y = 1) = 0.1$$

Закони расподеле случајних величина X и Y су:

$$X: \begin{pmatrix} 0 & 1 \\ 0.6 & 0.4 \end{pmatrix}$$

$$Y: \begin{pmatrix} 0 & 1 \\ 0.8 & 0.2 \end{pmatrix}$$

Дакле, случајна величина $X \in Ber(0.4)$ и $Y \in Ber(0.2)$.

Користећи формулу за условну вероватноћу $P(A|B) = \frac{P(A \cap B)}{P(B)}$ можемо наћи условне расподеле за X и Y.

Условне расподеле за X:

$$P(X = 0|Y = 0) = \frac{P(X = 0, Y = 0)}{P(Y = 0)} = \frac{0.5}{0.8} = \frac{5}{8} = 0.625$$

$$P(X = 0|Y = 1) = \frac{P(X = 0, Y = 1)}{P(Y = 1)} = \frac{0.1}{0.2} = \frac{1}{2} = 0.5$$

$$P(X = 1|Y = 0) = \frac{P(X = 1, Y = 0)}{P(Y = 0)} = \frac{0.3}{0.8} = \frac{3}{8} = 0.375$$

$$P(X=1|Y=1)=\frac{P(X=1,Y=1)}{P(Y=1)}=\frac{0.1}{0.2}=\frac{1}{2}=0.5$$

Условне расподеле за Y:

$$P(Y=0|X=0)=\frac{P(X=0,Y=0)}{P(X=0)}=\frac{0.5}{0.6}=\frac{5}{6}$$

$$P(Y=0|X=1)=\frac{P(X=1,Y=0)}{P(X=1)}=\frac{0.3}{0.4}=\frac{3}{4}=0.75$$

$$P(Y=1|X=0)=\frac{P(X=0,Y=1)}{P(X=0)}=\frac{0.1}{0.6}=\frac{1}{6}$$

$$P(Y=1|X=1)=\frac{P(X=1,Y=1)}{P(X=1)}=\frac{0.1}{0.4}=\frac{1}{4}=0.25$$

Прво ћемо имплементирати Бернулијеву расподелу са параметром p.

```
rbernuli <- function(p)
{
  # Generisemo prvo slucajan broj iz (0,1)
  U <- runif(1)
  # Zatim vracamo slucajan broj iz Ber(p)
  ifelse(U < p, 1, 0)
}
```

Сада ћемо правити узорак из расподеле X када нам је Y познато. Из условних расподела које смо мало пре рачунали, видимо да X има Бернулијеву расподелу $Ber(0.375)$ ако је $Y=0$, односно $Ber(0.5)$ ако је $Y=1$.

```
uzorakX_poznatoY <- function(y)
{
  if(y==0)
  {
    x <- rbernuli(0.375) # vraca 1 sa verovatnocom 0.375,inace vraca 0
  }
  else
  {
    x <- rbernuli(0.5)
  }
  return(x)
}
```

Сада ћемо правити узорак из расподеле Y када нам је X познато. Из условних расподела које смо мало пре рачунали, видимо да Y има Бернулијеву расподелу $Ber(\frac{1}{6})$ ако је $X=0$, односно $Ber(0.25)$ ако је $X=1$.

```
uzorakY_poznatoX <- function(x)
{
  if(x==0)
  {
    y <- rbernuli(1/6) # vraca 1 sa verovatnocom 0.375,inace vraca 0
  }
  else
  {
    y <- rbernuli(0.25)
```

```
  }
  return(y)
}
```

Ланац иницијализујемо на $(1,1)$ и правимо $Gibbs$ −ов узорак узастопним узорковањем из условних расподела.

Сада ћемо почети од иницијализоване вредности X и Y и понављати следеће кораке:

1. Симулираћемо нову вредност за X из условне вероватноће $P(X|Y=y)$ где је y тренутна вредност случајне величине Y
2. Симулираћемо нову вредност за Y из условне вероватноће $P(Y|X=x)$ где је x тренутна вредност случајне величине X (генерисане у кораку 1)

```
set.seed(100)
niter <- 10000
X <- rep(0,niter)
Y <- rep(0,niter)
X[1]=1
Y[1]=1 # pocinjemo od (1,1)

for(i in 2:niter)
{
  X[i] <- uzorakX_poznatoY(Y[i-1])
  Y[i] <- uzorakY_poznatoX(X[i])
}


Uzorak <- data.frame(X=X,Y=Y)
```

Исписаћемо шта се добија за првих 10 итерација:

```
head(Uzorak,10)
```

```
##    X Y
## 1  1 1
## 2  1 0
## 3  0 1
## 4  1 0
## 5  0 0
## 6  0 0
## 7  0 0
## 8  1 0
## 9  0 0
## 10 1 0
```

Још ћемо проверити колико се добијени резултат слаже са полазном заједничком расподелом:

```
table(data.frame(X=X,Y=Y))/niter
```

```
##    Y
## X        0      1
##   0 0.5000 0.1012
##   1 0.3008 0.0980
```

Видимо да је узорачка расподела приближна расподели веоватноћа $P(X=x,Y=y)=p_{X,Y}(x,y)$ датој у табели.

Четврти задатак

Стандардна претпоставка при моделовању генотипа са двоструким алелима је да се укрштање врши на случајан начин.
Према томе, за популацију где је p вероватноћа алела A, генотипови AA, Aa и aa имају вероватноће $p^2, 2p(1-p)$ и $(1-p)^2$.
Претпоставимо да p има $\mathcal{U}[0,1]$ априорну расподелу.
Претпоставимо да имамо узорак од n јединки: n_{AA} са генотипом AA, n_{Aa} са генотипом Aa и n_{aa} са генотипом aa.
Направити функцију $MCMCsampler(nAA, nAa, naa, iter, start\_valute, prop\_sd)$ која ће коришћењем Metropolis алгоритма вратити апроксимативно узорке из апостериорне расподеле за p. Предлог креирати додавањем шума из $\mathcal{N}(0, prop\_sd)$ расподеле.
Ако је $n_{AA} = 50, n_{Aa} = 21$ и $n_{aa} = 29$, покренути алгоритам за 10000 итерација, са почетном вредношћу 0.5 и ширином расподеле предлога 0.01.
Нацртати хистограм, а на њега доцртати густину праве апостериорне расподеле.
Покренути алгоритам за другу почетну вредност, и мањи број итерација, нпр. 0.1 и 1000, редом. Нацртати график ланца(временске серије) и на основу њега проценити колико би почетних вредности требало одбацити као $burn - in$.

Дата нам је расподела вероватноћа генотипова :

$$G: \begin{pmatrix} AA & Aa & aa \\ p^2 & 2p(1-p) & (1-p)^2 \end{pmatrix}$$

Претпоставили смо да p има $\mathcal{U}[0,1]$ априорну расподелу, па је густина $q_{apriorna}(p) = 1$.

```
prior <- function(p)
{
  if((p<0)||(p>1))
  {
    return(0)
  }
  else
  {
    return(1)
  }
}
```

Како имамо узорак од n јединки: n_{AA} са генотипом AA, n_{Aa} са генотипом Aa и n_{aa} са генотипом aa функција веродостојности је $L(p) = p^{2 \cdot nAA} \cdot \big(2p(1-p)\big)^{nAa} \cdot (1-p)^{2 \cdot naa}$

```
likelihood <- function(p,nAA,nAa,naa)
{
  L <- p^(2*nAA)*(2*p*(1-p))^nAa*(1-p)^(2*naa)
  return(L)
}
```

Сада правимо функцију $MCMCsampler(nAA, nAa, naa, iter, start\_valute, prop\_sd)$ која ће коришћењем $Metropolis$ алгоритма вратити апроксимативно узорке из апостериорне расподеле за p.

```
MCMCsampler <- function(nAA,nAa,naa,iter,start_valute,prop_sd)
{
  p <- rep(0,iter)
  p[1] <- start_valute # prva vrednost vektora ce biti unapred zadata vrednost start
_valute,a ostale dobijamo prolazeci kroz petlju

  for(i in 2:iter)
  {
    trenutno_p <- p[i-1] # trenutna vrednost p
    novo_p <- trenutno_p + rnorm(1,0,prop_sd) # novu vrednost za p dobijamo dodavanj
em suma iz N(0,prop_sd) raspodele
    q <-prior(novo_p)*likelihood(novo_p,nAA,nAa,naa)/(prior(trenutno_p)*likelihood(t
renutno_p,nAA,nAa,naa)) # aposteriorna raspodela

    if (runif(1)<q)
    {
      p[i] <- novo_p
    }
    else
    {
      p[i] <-trenutno_p
    }
  }
  return(p)
}
```

Покренућемо функцију за $n_{AA} = 50, n_{Aa} = 21$ и $n_{aa} = 29$ са 10000 итерација, почетном вредношћу 0.5 и ширином расподеле предлога 0.01.

```
X <- MCMCsampler(50,21,29,10000,0.5,0.01)
head(X,10)
```

```
##  [1] 0.5000000 0.5026283 0.5044836 0.5051837 0.5051837 0.5068216 0.5140161
##  [8] 0.5219719 0.5321921 0.5322378
```

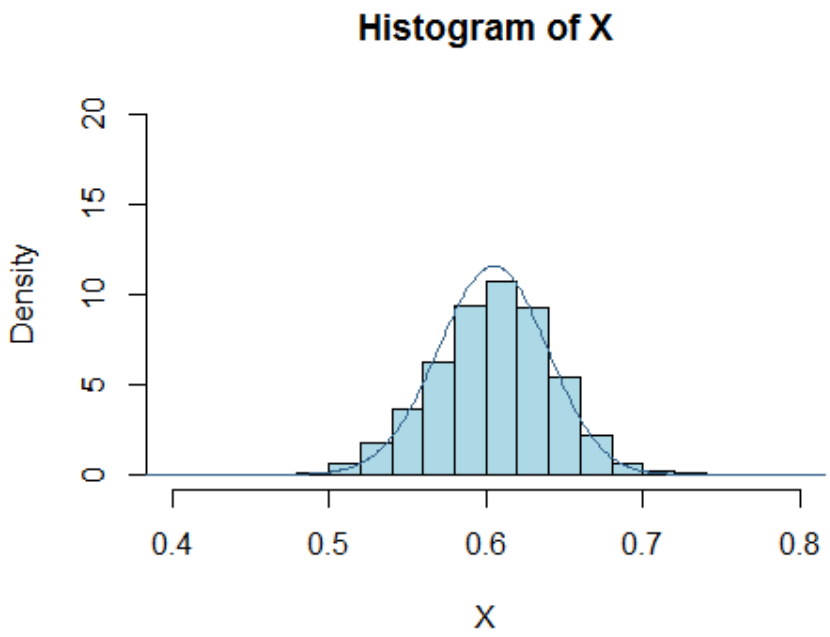
Израчунаћемо теоријску апостериорну расподелу за p.

$$q_{aposteriorna}(p) = \frac{L(p) \cdot q_{apriorna}(p)}{\int L(p) \cdot q_{apriorna}(p)dp} = \frac{p^{2 \cdot nAA} \cdot \left(2p(1-p)\right)^{nAa} \cdot (1-p)^{2 \cdot naa}}{\int p^{2 \cdot nAA} \cdot \left(2p(1-p)\right)^{nAa} \cdot (1-p)^{2 \cdot naa} dp}$$
$$= \frac{p^{2 \cdot nAA + nAa} \cdot (1-p)^{nAa + 2 \cdot naa}}{\beta(2 \cdot nAA + nAa + 1, nAa + 2 \cdot naa + 1)}$$

Апостериорна расподела за p је бета расподела $\beta(2 \cdot nAA + nAa + 1, nAa + 2 \cdot naa + 1)$.

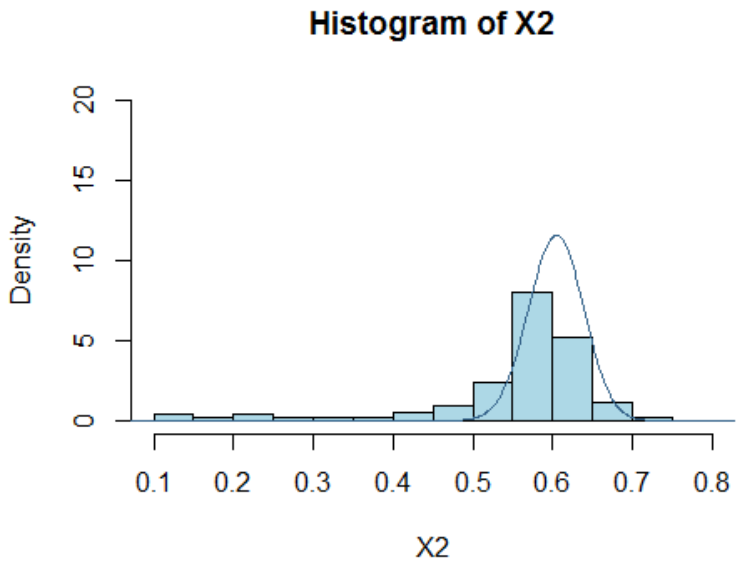
Када је $n_{AA} = 50, n_{Aa} = 21$ и $n_{aa} = 29$ апостериорна расподела је $\beta(122,80)$.

```
Y <- seq (0,1,length=1000) # segment [0,1] delimo na 1000 jednakih delova
hist(X, probability = T,xlim = c(0.4,0.8),col="lightblue",ylim=c(0,20))
lines(Y,dbeta(Y,122,80),col="steelblue4")
```



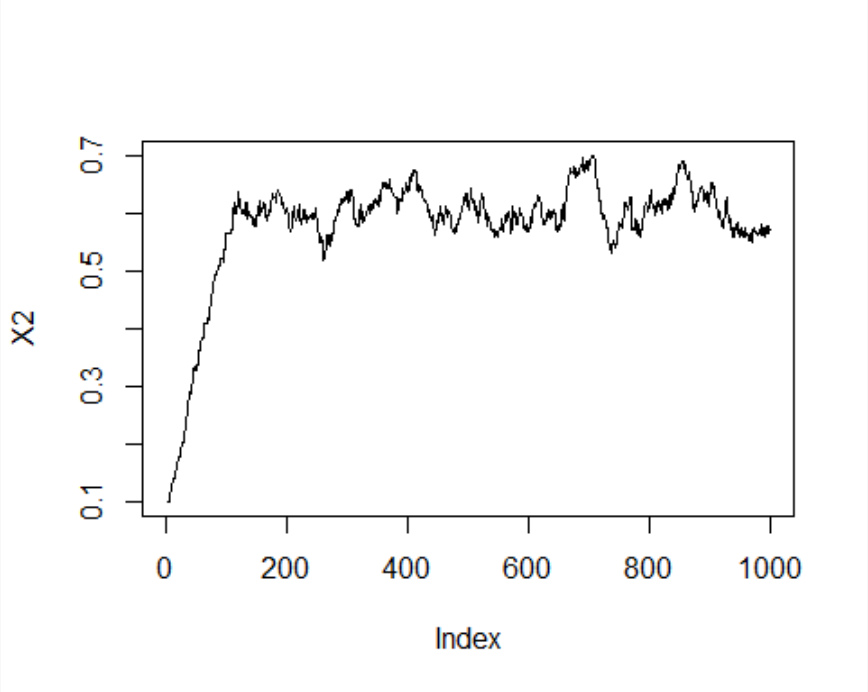
Покренућемо алгоритам за другу почетну вредност 0.1 и мањи број итерација 1000.

```
X2 <- MCMCsampler(50,21,29,1000,0.1,0.01)
head(X2,10)

##  [1] 0.1000000 0.1000000 0.1000000 0.1000000 0.1073581 0.1081183 0.1201683
##  [8] 0.1201683 0.1201683 0.1389774

Y2 <- seq (0,1,length=1000)
hist(X2, probability = T,xlim = c(0.1,0.8),col="lightblue",ylim=c(0,20))
lines(Y2,dbeta(Y2,122,80),col="steelblue4")
```

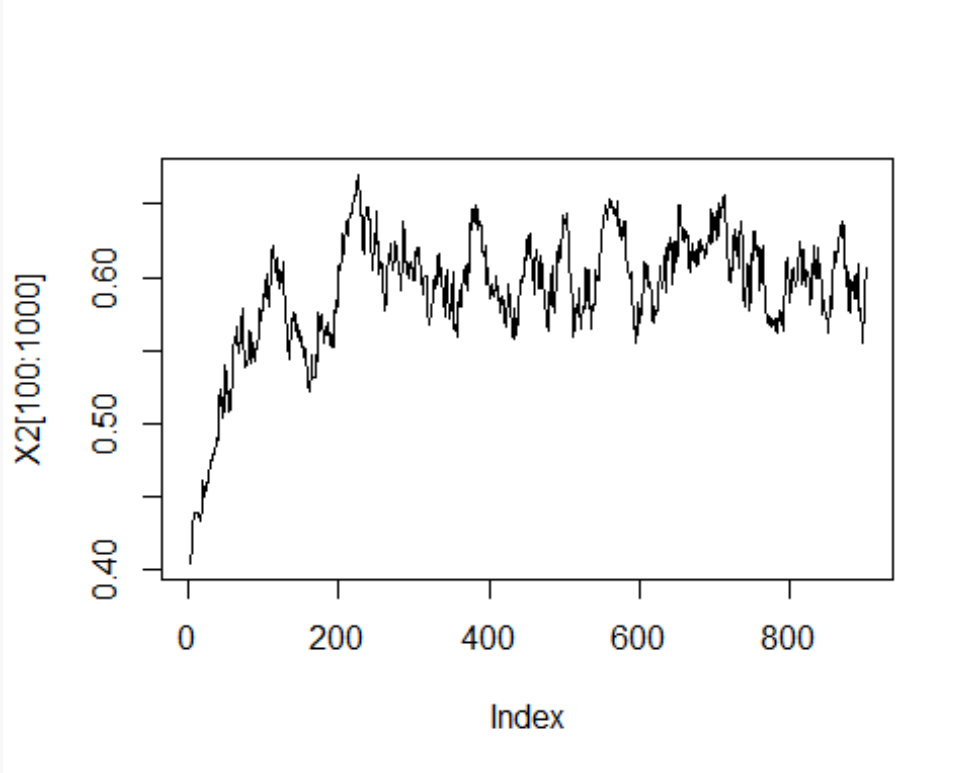


Цртамо график ланца(временске серије):

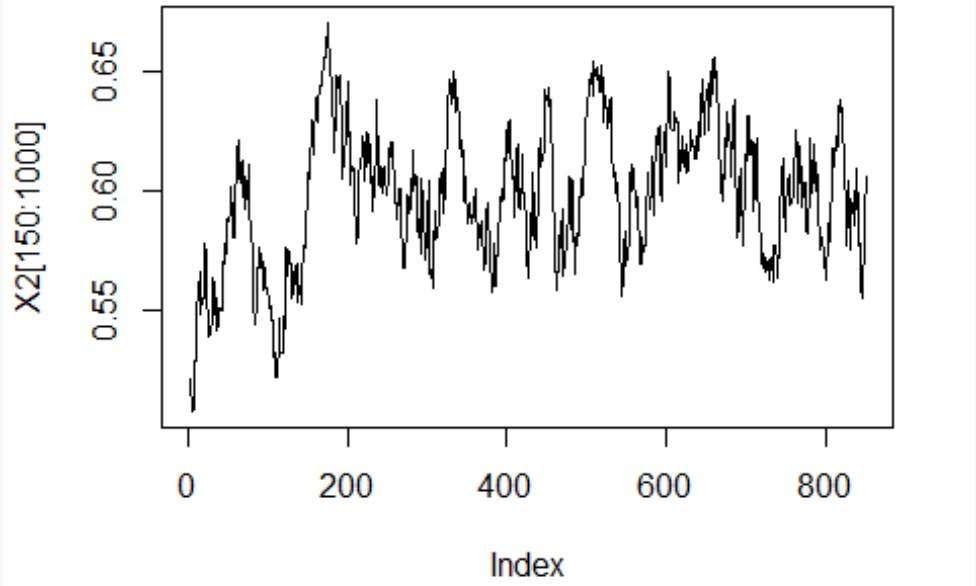
```
Y <- seq (0,1,length=1000) # segment [0,1] delimo na 1000 jednakih delova
plot(X2,type="l")
```




```
plot(X2[100:1000],type="l")
```




```
plot(X2[150:1000],type="l")
```



Са графика можемо видети да би отприлике првих 100-150 почетних вредности требало одбацити као $burn - in$, односно око 100-150 итерација је потребно да би серија постала ергодична (да се креће око константне вредности).

Пети задатак

Претпоставимо да за низ случајних величина $Y_1, Y_2, \dots, Y_n$ важи:

$$Y_i \overset{iid}{\sim} \mathcal{P}(\lambda_1) \quad \text{за } i = 1, \dots, m$$

$$Y_i \overset{iid}{\sim} \mathcal{P}(\lambda_2) \quad \text{за } i = m+1, \dots, n$$

Априорне расподеле за непознате параметре λ_1, λ_2 и m су дате са:

$$\pi(\lambda_1) \sim \gamma(a_1, b_1)$$

$$\pi(\lambda_2) \sim \gamma(a_2, b_2)$$

$$\pi(m) \sim \frac{1}{n}$$

Извести условне апостериорне расподеле за $\pi(\lambda_1 | \boldsymbol{Y}, m), \pi(\lambda_2 | \boldsymbol{Y}, m)$ и $\pi(m | \boldsymbol{Y}, \lambda_1, \lambda_2)$, а затим, кроз 10000 итерација узорковати:

$$\theta_1^{(k)} \sim \pi(\theta_1 | \mathbf{Y}, m^{(k-1)})$$

$$\theta_2^{(k)} \sim \pi(\theta_2 | \mathbf{Y}, m^{(k-1)})$$

па

$$m^{(k)} \sim \pi(m | \mathbf{Y}, \lambda_1^{(k)}, \lambda_2^{(k)})$$

Дати су подаци

$\boldsymbol{Y} = (4,5,4,1,0,4,3,4,0,6,3,3,4,0,2,6,3,3,5,4,5,3,2,4,4,1,5,5,3,4,2,5,2,2,3,4,2,1,3,2,$
$2,1,1,1,1,3,0,0,1,0,1,1,0,0,3,1,0,3,2,2,0,1,1,1,0,1,0,1,0,0,0,2,1,0,0,0,1,1,0,2,3,3,1,$
$1,2,1,1,1,1,2,4,2,0,0,0,1,4,0,0,0,1,0,0,0,0,0,1,0,0,1,0,1,0,1), a_1 = 3, a_2 = 1, b_1 = 0.5, b_2 = 0.5.$

НАПОМЕНА: Као условне апостериорне расподеле за λ_1 и λ_2 добијају се познате расподеле, док се за m добија израз који не указује ни на једну познату расподелу. Међутим, лако можемо добити апостериорну условну расподелу за $m, \pi(m | \boldsymbol{Y}, \lambda_1^{(k)}, \lambda_2^{(k)})$, тако што прођемо добијеним изразом кроз све вредности које може да узме случајна величина m, сумирамо, и нормализујемо (поделимо са добијеном сумом,да добијемо праву расподелу, тј. да у збиру буде 1),а онда узоркујемо $m^{(k)} \sim \pi(m | \boldsymbol{Y}, \lambda_1^{(k)}, \lambda_2^{(k)})$, као најмањи број из скупа допустивих вредности за m, за које функција расподеле (израчунате у претходном кораку) прелази случајно изабрани број из $(0,1)$ (помоћ:за добијање функције расподеле може се користити функција $cumsum$).

Нацртати хистограме појединачних компоненти и оценити вредности одговарајућих параметара.

```r
# Dati su podaci:
Y <- c(4,5,4,1,0,4,3,4,0,6,3,3,4,0,2,6,3,3,5,4,5,3,2,4,4,1,5,5,3,4,2,5,2,2,3,4,2,
       1,3,2,2,1,1,1,1,3,0,0,1,0,1,1,0,0,3,1,0,3,2,2,0,1,1,1,0,1,0,1,0,0,0,2,1,0,
       0,0,1,1,0,2,3,3,1,1,2,1,1,1,1,2,4,2,0,0,0,1,4,0,0,0,1,0,0,0,0,0,1,0,0,1,0,1)
a1 <- 3
a2 <- 1
b1 <- 0.5
b2 <- 0.5
```

Случајне величине $Y_i, i = 1, \dots, m$ су независне и једнако расподељене случајне величине са $\mathcal{P}(\lambda_1)$ расподелом.

Дакле,

$$P(\boldsymbol{Y}|\lambda_1) = \frac{{\lambda_1}^y e^{-\lambda_1}}{y!}$$

$$L(\boldsymbol{Y}|\lambda_1) = \prod_{i=1}^{m} P(\boldsymbol{Y}|\lambda_1) = \frac{{\lambda_1}^{\sum_{i=1}^m y_i} e^{-m\lambda_1}}{\prod_{i=1}^m y_i!}$$

Априорна расподела за непознати параметар λ_1 је $\gamma(a_1, b_1)$, па је

$$\pi(\lambda_1) = \frac{b_1^{a_1} \lambda_1^{a_1-1} e^{-b_1\lambda_1}}{\Gamma(a_1)}$$

Сада можемо извести условну апостериорну расподелу $\pi(\lambda_1|\boldsymbol{Y}, m)$:

$$\pi(\lambda_1|\boldsymbol{Y}, m) = \frac{L(\boldsymbol{Y}|\lambda_1)\pi(\lambda_1)}{\int L(\boldsymbol{Y}|\lambda_1)\pi(\lambda_1)d\lambda_1} = \frac{\frac{{\lambda_1}^{\sum_{i=1}^m y_i} e^{-m\lambda_1}}{\prod_{i=1}^m y_i!} \cdot \frac{b_1^{a_1}\lambda_1^{a_1-1}e^{-b_1\lambda_1}}{\Gamma(a_1)}}{\int \frac{{\lambda_1}^{\sum_{i=1}^m y_i} e^{-m\lambda_1}}{\prod_{i=1}^m y_i!} \cdot \frac{b_1^{a_1}\lambda_1^{a_1-1}e^{-b_1\lambda_1}}{\Gamma(a_1)} d\lambda_1}$$

$$= K_1 \cdot {\lambda_1}^{a_1+\sum_{i=1}^m y_i - 1} e^{-\lambda_1(m+b_1)}$$

где је K_1 константа из $\mathbb{R}$.

Дакле, $\pi(\lambda_1|\boldsymbol{Y}, m) \sim \gamma(a_1 + \sum_{i=1}^m y_i, m + b_1)$.

Случајне величине $Y_i, i = m+1, \dots, n$ су независне и једнако расподељене случајне величине са $\mathcal{P}(\lambda_2)$ расподелом.

Дакле,

$$P(\boldsymbol{Y}|\lambda_2) = \frac{{\lambda_2}^y e^{-\lambda_2}}{y!}$$

$$L(\boldsymbol{Y}|\lambda_2) = \prod_{i=m+1}^{n} P(\boldsymbol{Y}|\lambda_2) = \frac{{\lambda_2}^{\sum_{i=m+1}^n y_i} e^{-(n-m)\lambda_2}}{\prod_{i=m+1}^n y_i!}$$

Априорна расподела за непознати параметар λ_2 је $\gamma(a_2, b_2)$, па је

$$\pi(\lambda_2) = \frac{b_2^{a_2}\lambda_2^{a_2-1}e^{-b_2\lambda_2}}{\Gamma(a_2)}$$

Сада можемо извести условну апостериорну расподелу $\pi(\lambda_2|\boldsymbol{Y}, m)$:

$$\pi(\lambda_2|\boldsymbol{Y}, m) = \frac{L(\boldsymbol{Y}|\lambda_2)\pi(\lambda_2)}{\int L(\boldsymbol{Y}|\lambda_2)\pi(\lambda_2)d\lambda_2} = \frac{\frac{\lambda_2^{\sum_{i=m+1}^{n} y_i}e^{-(n-m)\lambda_2}}{\prod_{i=m+1}^{n} y_i!} \cdot \frac{b_2^{a_2}\lambda_2^{a_2-1}e^{-b_2\lambda_2}}{\Gamma(a_2)}}{\int \frac{\lambda_2^{\sum_{i=m+1}^{n} y_i}e^{-(n-m)\lambda_2}}{\prod_{i=m+1}^{n} y_i!} \cdot \frac{b_2^{a_2}\lambda_2^{a_2-1}e^{-b_2\lambda_2}}{\Gamma(a_2)} d\lambda_2}$$
$$= K_2 \cdot {\lambda_2}^{a_2+\sum_{i=m+1}^{n} y_i - 1} e^{-\lambda_2(n-m+b_2)}$$

где је K_2 константа из $\mathbb{R}$.

Дакле, $\pi(\lambda_2|\boldsymbol{Y}, m) \sim \gamma(a_2 + \sum_{i=m+1}^{n} y_i, n - m + b_2)$.

Сада рачунамо условну апостериорну расподелу $\pi(m|\boldsymbol{Y}, \lambda_1, \lambda_2)$:

$$\pi(m|\boldsymbol{Y}, \lambda_1, \lambda_2) = \frac{\pi(\lambda_1|\boldsymbol{Y}, m) \cdot \pi(\lambda_2|\boldsymbol{Y}, m) \cdot \pi(m)}{\sum_1^n \pi(\lambda_1|\boldsymbol{Y}, m) \cdot \pi(\lambda_2|\boldsymbol{Y}, m) \cdot \pi(m)}$$
$$= \frac{K_1 \cdot {\lambda_1}^{a_1+\sum_{i=1}^{m} y_i - 1} e^{-\lambda_1(m+b_1)} \cdot K_2 \cdot {\lambda_2}^{a_2+\sum_{i=m+1}^{n} y_i - 1} e^{-\lambda_2(n-m+b_2)} \cdot \frac{1}{n}}{\sum_1^n K_1 \cdot {\lambda_1}^{a_1+\sum_{i=1}^{m} y_i - 1} e^{-\lambda_1(m+b_1)} \cdot K_2 \cdot {\lambda_2}^{a_2+\sum_{i=m+1}^{n} y_i - 1} e^{-\lambda_2(n-m+b_2)} \cdot \frac{1}{n}}$$
$$= K_3 \cdot \frac{\lambda_1^{\sum_{i=1}^{m} y_i} e^{-\lambda_1 m} \cdot \lambda_2^{\sum_{i=m+1}^{n} y_i} e^{\lambda_2 m}}{\sum_1^n \lambda_1^{\sum_{i=1}^{m} y_i} e^{-\lambda_1 m} \cdot \lambda_2^{\sum_{i=m+1}^{n} y_i} e^{\lambda_2 m}} = K_3 \cdot \frac{\lambda_1^{\sum_{i=1}^{m} y_i} e^{-\lambda_1 m} \cdot \lambda_2^{\sum_{i=1}^{n} y_i - \sum_{i=1}^{m} y_i} e^{\lambda_2 m}}{\sum_1^n \lambda_1^{\sum_{i=1}^{m} y_i} e^{-\lambda_1 m} \cdot \lambda_2^{\sum_{i=m+1}^{n} y_i} e^{\lambda_2 m}}$$
$$= K \cdot e^{(\lambda_2 - \lambda_1)m} \cdot \left(\frac{\lambda_1}{\lambda_2}\right)^{\sum_{i=1}^{m} y_i}$$

где је K константа из $\mathbb{R}$.

Дакле, за m смо добили израз који не указује ни на једну познату расподелу.

Примењујемо $MCMC$ узорковање на следећи начин:

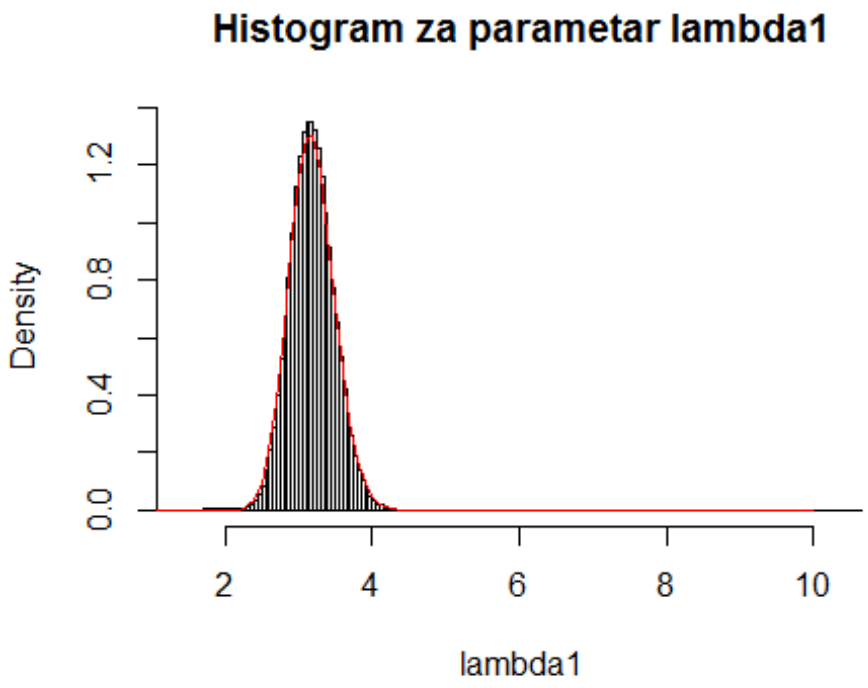
- КОРАК 1:
$$\lambda_1^{(k)} \sim \pi(\lambda_1|\boldsymbol{Y}, m^{(k-1)})$$
$$\lambda_2^{(k)} \sim \pi(\lambda_2|\boldsymbol{Y}, m^{(k-1)})$$

- КОРАК 2:
$$m^{(k)} \sim \pi(m|\boldsymbol{Y}, \lambda_1^{(k)}, \lambda_2^{(k)})$$

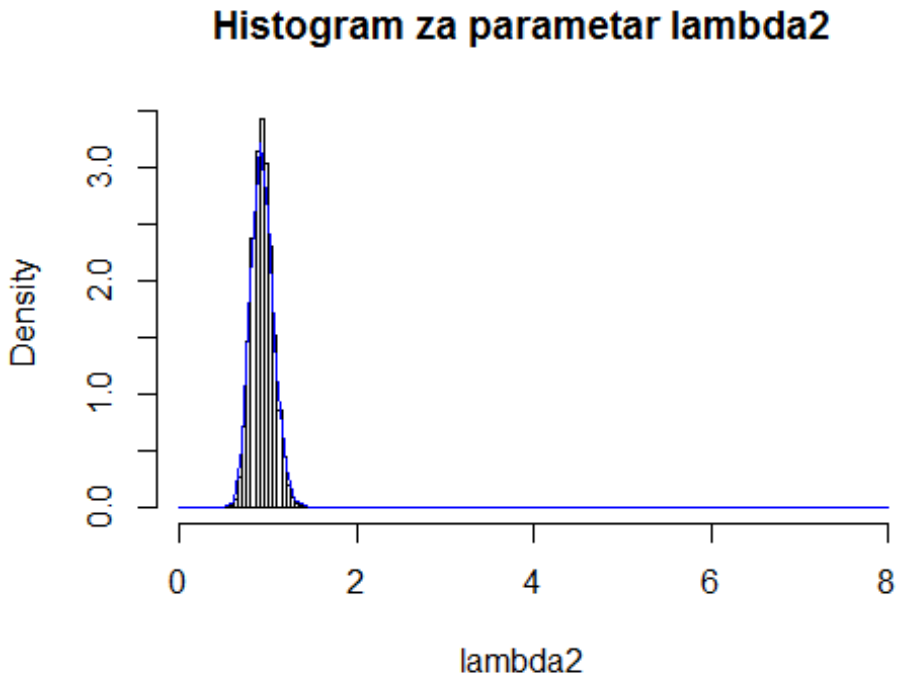
Понављамо претходна два корака док се не постигне стационарна расподела.

```
MC <- 1000 # broj iteracija
N <- 200 # duzina lanca

n <- length(Y) # obim uzorka
m <- n # nepokretna tacka
p <- rep(0,3*MC*N) # pravimo niz u kome cemo cuvati rezultate
dim(p)<-c(3,MC,N)
```

```
for (j in (1:MC))
{
  m <- as.integer(n*runif(1))+1
  for (i in (1:N))
  {
    lambda1 <- rgamma(1,a1+sum(Y[1:m]),m+b1)
    lambda2 <- rgamma(1,a2+sum(Y)-sum(Y[1:m]),n-m+b2)
    # za dobijanje funkcije raspodele koristimo funkciju cumsum
    pm <- exp((lambda2-lambda1)*(1:n))*(lambda1/lambda2)^cumsum(Y)
    # sumiramo i normalizujemo da dobijemo pravu raspodelu tj. da u zbiru bude 1
    pm <- pm/sum(pm)
    m <- min((1:n)[runif(1)<cumsum(pm)])
    # m uzorkujemo kao najmanji broj iz skupa dopustivih vrednosti za m, za koji
dobijena funkcija raspodele,
    # prelazi slucajno izabrani broj iz (0,1)
    # cuvamo rezultate
    p[1,j,i]<-m
    p[2,j,i]<-lambda1
    p[3,j,i]<-lambda2
  }
}


parametar_m <-p[1,,]
parametar_lambda1 <-p[2,,]
parametar_lambda2 <-p[3,,]


# Ocenjujemo parametre aposteriornih raspodjela za lambda1 i lambda 2 metodo
m momenata funkcijom gamaMM
gamaMM <- function(x)
{
  n <- length(x) # duzina x
  mean_x <- mean(x) # x srednja vrednost
  alpha <- n*(mean_x^2)/sum((x-mean_x)^2) # ocena metodom momenata za alfa
  beta <- 1/(sum((x-mean_x)^2)/n/mean_x) # ocena metodom momenata za beta
  estimate <- data.frame(alpha,beta) # spojimo dve ocene u data frame
  return(estimate)
}
parametar_lambda1_MM<-gamaMM(parametar_lambda1) #parametri za aposteriornu r
aspodelu za lambda1
parametar_lambda2_MM<-gamaMM(parametar_lambda2) #parametri za aposteriornu r
aspodelu za lambda2


x<-seq(0,1,length=1000) # delimo interval zbog fukcije raspodjele
# pravimo histogram za uzorak iz aposteriorne raspodjele za lambda1 i na nje
mu crtamo funkciju raspodele
# sa ocenjenim parametrima

hist(parametar_lambda1, main = "Histogram za parametar lambda1",xlab = "lamb
da1",probability = T,breaks = N)
curve(dgamma(x, parametar_lambda1_MM$alpha, parametar_lambda1_MM$beta),xlim
= c(0,10),add=TRUE, col='red')
```

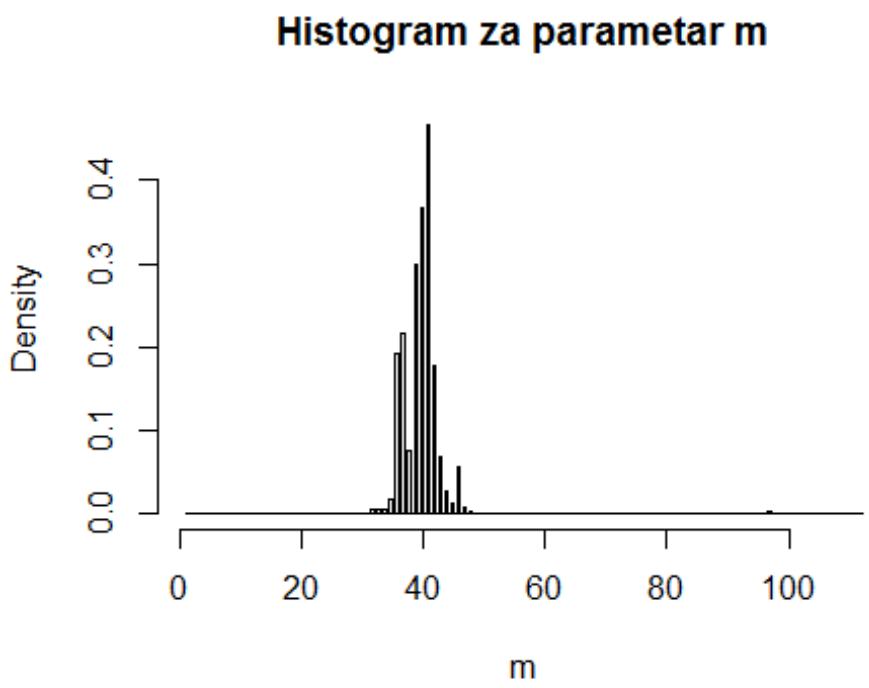



```
# isto to radimo i sa aposteriornom raspodelom za lambda2
hist(parametar_lambda2,main = "Histogram za parametar lambda2",xlab = "lambd
a2",probability = T,breaks = N)
curve(dgamma(x, parametar_lambda2_MM$alpha, parametar_lambda2_MM$beta),xlim
= c(0,10),add=TRUE, col='blue')
```




```
# u oba slucaja se poklapaju funkcija raspodele i histogrami dobijenih uzora
ka
```

```
hist(parametar_m, main = "Histogram za parametar m",xlab = "m",probability =
T,breaks = N)
```




```
# histogram za uzorak iz aposteriorne raspodjele za m, koji ne lici ni na je
dnu poznatu raspodelu
```

Бонус задатак

Из стринга ("06435.213", "aswww", "2112*121", "011", "232424321", "1232", "23423aaa21321", "0.1", "3424", "123*131", "232 232", "2.3", "4543.45") издвојити исправно написане бројеве (прослеђивањем одговарајућег регуларног израза функцији $str\_view()$).

```
#install.packages("htmlwidgets")
library(htmlwidgets)

string<-c("06435.213", "aswww", "2112*121", "011", "232424321", "1232", "23423aaa213
21", "0.1", "3424","123*131", "232 232", "2.3", "4543.45")

str_view(string, "((.*[^0123456789\\.].*)|(^0+[^\\.].*))",match = FALSE)
```

```
232424321
1232
0.1
3424
2.3
4543.45
```

Фукцијом str_view издвајамо елементе који одговарају прослеђеном регуларном изразу функцији. У нашем случају, ставили смо да издвоји све стрингове који садрже неки карактер осим бројева и тачке или стингове који почињу са два броја од којих је први нула. На овај начин издвајамо све стрингове који нису бројеви, а како нама треба да издвојимо исправно написане бројеве, задали смо услов $match = FALSE$, што уствари издваја све стрингове који не испуњавају задати услов, тј. у нашем случају све бројеве.