Obrada rezultata mjerenja pomoću programskog jezika R

Luketić, Marin

Master's thesis / Diplomski rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:724553

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-07**

Repository / Repozitorij:

Repository of the Faculty of Science - University of Zagreb

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

Marin Luketić

Obrada rezultata mjerenja pomoću programskog jezika R

Diplomski rad

Zagreb, 2016.

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

SMJER: PROFESOR FIZIKE

Marin Luketić

Diplomski rad

Obrada rezultata mjerenja pomoću programskog jezika R

Voditelj diplomskog rada: doc.dr.sc. Davor Horvatić

Ocjena diplomskog rada: _____

Povjerenstvo: 1. _____

2. _____

3. _____

Datum polaganja: _____

Zagreb, 2016.

Zahvaljujem se mentoru, doc.dr.sc. Davoru Horvatiću na stručnoj pomoći, razumijevanju i savjetima tijekom izrade ovog rada.

Najviše se zahvaljujem svojoj obitelji na pomoći i podršci tijekom školovanja.

Sažetak

U diplomskom radu se predstavlja programski jezik R i obrada rezultata mjerenja u navedenom programskom jeziku s naglaskom na regresijsku analizu, odnosno jednostavnu linearnu regresiju te nelinearnu regresiju. Napravljen je pregled statističke obrade podataka, gdje se opisuju pojmovi mjerenja, preciznosti i pogreške kao i osnovni statistički pojmovi koji su važni za kvalitetnu obradu podataka. Obrađuje se regresijski model gdje se opisuju jednostavna linearna regresija i metoda najmanjih kvadrata koja je važna za procijenu regresijskih parametara. Prikazana je i primjena nelinearne regresije. Obrađeni su rezultati mjerenja u programskom jeziku R uz primjere dviju standardnih vježbi na fizikalnom praktikumu: Matematičko njihalo i Ogib svjetlosti.

Processing of measurement results using the programming language R

Abstract

This thesis we will present the programming language R and the analyisis of the measurement results in the specified programming language with a focus on regression analysis, namely simple linear regression and nonlinear regression. Review of statistical data analysis will be presented with empasis on measurement process, precision and error as well as basic statistical concepts that are important for the data processing. Regression model is described with the simple linear regression using least squares method to fit parameters. Application of non-linear regression is also discussed. The results are obtained in the R language for two standard student exercises in physical practicum: Mathematical pendulum and Diffraction of light.

Sadržaj

1 Uvod **1**

2 Programski paket R **2**
2.1 Uvod u R . . . 2
2.2 Povijest . . . 2
2.3 Instalacija programa . . . 3
2.4 RStudio . . . 3
2.5 Pomoć u R-u . . . 5
2.6 Korisničko sučelje Rstudia . . . 6
2.6.1 Konzola . . . 6
2.6.2 R skripta . . . 8
2.7 Vektori . . . 8
2.8 Aritmetički nizovi . . . 9
2.9 Funkcije i argumenti . . . 10
2.9.1 Pisanje vlastitih funkcija . . . 13
2.10 Liste . . . 14
2.11 Polja i matrice (Data frames) . . . 15
2.12 Izrada spremnika podataka . . . 16
2.13 Petlje . . . 18
2.14 Optimizacija u R-u . . . 19
2.15 Grafika u R-u . . . 19

3 Statistička obrada podataka **24**
3.1 Mjerenja, preciznost i pogreške . . . 24
3.1.1 Mjerenja . . . 24
3.1.2 Točnost, preciznost . . . 25
3.1.3 Pogreška . . . 25
3.2 Obrada rezultata mjerenja . . . 26
3.2.1 Osnovni statistički pojmovi, srednja vrijednost i standardna devijacija . . . 26
3.2.2 Distribucije . . . 27
3.3 Ovisna i neovisna mjerenja . . . 30
3.3.1 Neovisna mjerenja . . . 30
3.3.2 Ovisna mjerenja . . . 31
3.4 Prikaz, zaokruživanje mjerenja i finalnih rezultata . . . 31
3.5 Linearna regresija i nelinearna regresija . . . 32
3.5.1 Regresija . . . 32
3.5.2 Linearna Regresija . . . 32
3.5.3 Nelinearna regresija . . . 33

4 Obrada rezultata mjerenja **34**
4.1 Matematičko njihalo 34
4.1.1 Uvod u vježbu 34
4.1.2 Mjerenja 35
4.1.3 Zaključak 45
4.2 Ogib svjetlosti 46
4.2.1 Uvod u vježbu 46
4.2.2 Mjerenja 47
4.2.3 Zaključak 54

5 Zaključak **55**

1 Uvod

Ideja i tema ovog diplomskog rada motivirana je željom da se prikažu mogućnosti programskog jezika R odnosno programskog jezika za statističku obradu podataka kao i njegova primjena te inovativnost u znanstvenom i poslovnom svijetu. Besplatni programski jezik R dugo je prisutan u svijetu, no manje poznat ili, bolje rečeno, manje popularan u Hrvatskoj.

U ovom radu prikazan je način rada, mogućnosti i primjena programskog jezika na konkretnim primjerima. Rad može biti zanimljiv kao jedan oblik skripte za ostale studente fizike, zato što će uz prikaz sintakse programskog jezika biti prikazana i konkretna primjena programskog jezika na primjerima fizikalnog praktikuma.

Tijekom studija student se susreće s mnoštvom praktikuma gdje se rezultati mjerenja obrađuju na različite načine - pomoću $Matlaba$, $Mathematice$, čak i pomoću papira i olovke. Obzirom na razvoj tehnlogije, korisno je i zanimljivo je taj način obrade podataka unaprijediti i pojednostaviti korištenjem novih tehnoloških momenata. Programska podrška R-a je slobodnog dohvata te predstavlja odličnu alternativu za komercijalne programske pakete kao što je $Mathematica$. Uz upoznavanje sintakse jezika, statističke definicije i konkretne primjere ovaj će diplomski rad biti svojevrsni uvod u obradu rezultata fizikalnih mjerenja pomoću programa R i njegovih programskih paketa.

Diplomski rad opisuje nastanak, instalaciju, i mogućnosti programskog jezika R kroz fizikalne primjere. Također, govori o definiciji statističke obrade podataka opisujući pojmove mjerenja, preciznosti i pogreške. Opisani su i osnovni statistički pojmovi nužni za kvalitetnu obradu podataka i razumijevanje. Jednostavna linearna regresija, metoda najmanjih kvadrata, logaritamska nelinearna regresija obrađeni su konkretnim primjerima.

2 Programski paket R

2.1 Uvod u R

U uvodu ćemo se upoznati s programskim jezikom R, dobiti uvid u osnovne karakteristike programskog jezika, opisati sve mogućnosti primjene te saznati nešto više o paketima, funkcijama i objektima R-a čije poznavanje je nužno za razumijevanje, primjenu statističkih metoda i obradu podataka fizikalnih mjerenja.

Što je R? R predstavlja integrirano programsko okruženje za upravljanje podacima, numeričku analizu i grafički prikaz podataka.

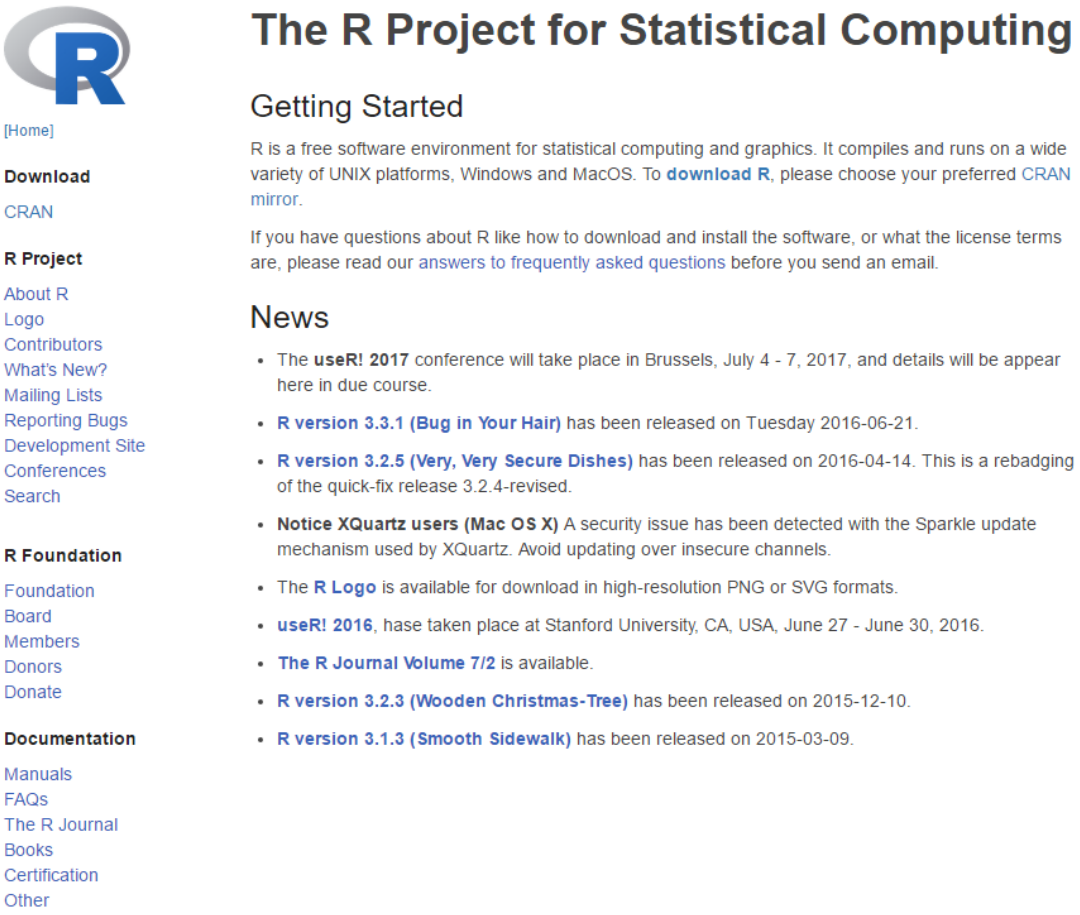
Često ga se naziva statističkim programom, međutim radi se o dobro planiranom i konzistentnom sustavu, možemo reći o univerzalnom programu koji je primjenjiv u bilo kojoj tehničkoj, prirodnoj ili društenoj sferi koja zahtjeva neki oblik fizikalne, matematičke ili statističke analize. R je prije svega jednostavan, dobro razvijen, učinkovit i besplatan programski jezik koji uključuje uvjetne tvrdnje, petlje, rekurzivne funkcije definirane od strane korisnika te postupke za učitavanje i spremanje podataka.

R posjeduje veliku, koherentnu, integriranu zbirku programskih alata za analizu podataka, odlične grafičke mogućnosti za analizu podataka, kao i mnogo operatora za računanje s poljima podataka, a posebno matricama.

Spomenuto je ranije da je riječ o besplatnom programskom jeziku, tj. programska podrška R-a je slobodnog dohvata (engl. Open-source software). To je veliku prednost u odnosu na ostale programe koji su mahom komercijalni, a samim time manje otvoreni i nedostupni velikom broju korisnika. Ovakva vrsta programa omogućava svakom istraživaču da sudjeluje u njegovom razvoju. Na taj način pokrivena je široka paleta matematičkih i statističkih modela koji postaju dostupni krajnjim korisnicima. Trenutno su dostupne programske inačice za najzastupljenije programske sustave kao što su *Windows*, *Linux*, *MacOS*.

2.2 Povijest

R svoje je korijene vuče iz programskog jezika S koji je razvijen u Bell laboratorijima od strane Ricka Beckera, Johna Chambersa i Allana Wilksa. Mijesto nastanka programskog jezika R je Oakland, na Novom Zelandu. Naime, dva profesora statistike sa Sveučilišta u Oaklandu, Ross Ihaka i Robert Gentleman, odlučili su sredinom devedesetih godina "napisati" implementaciju S programskog jezika i izdali je pod GNU licencom (General Public License). Početna slova njihovih imena poslužila su



Slika 2.1: Prikaz početne stranice http://cran.r-roject-org/ (15.9.2016.)

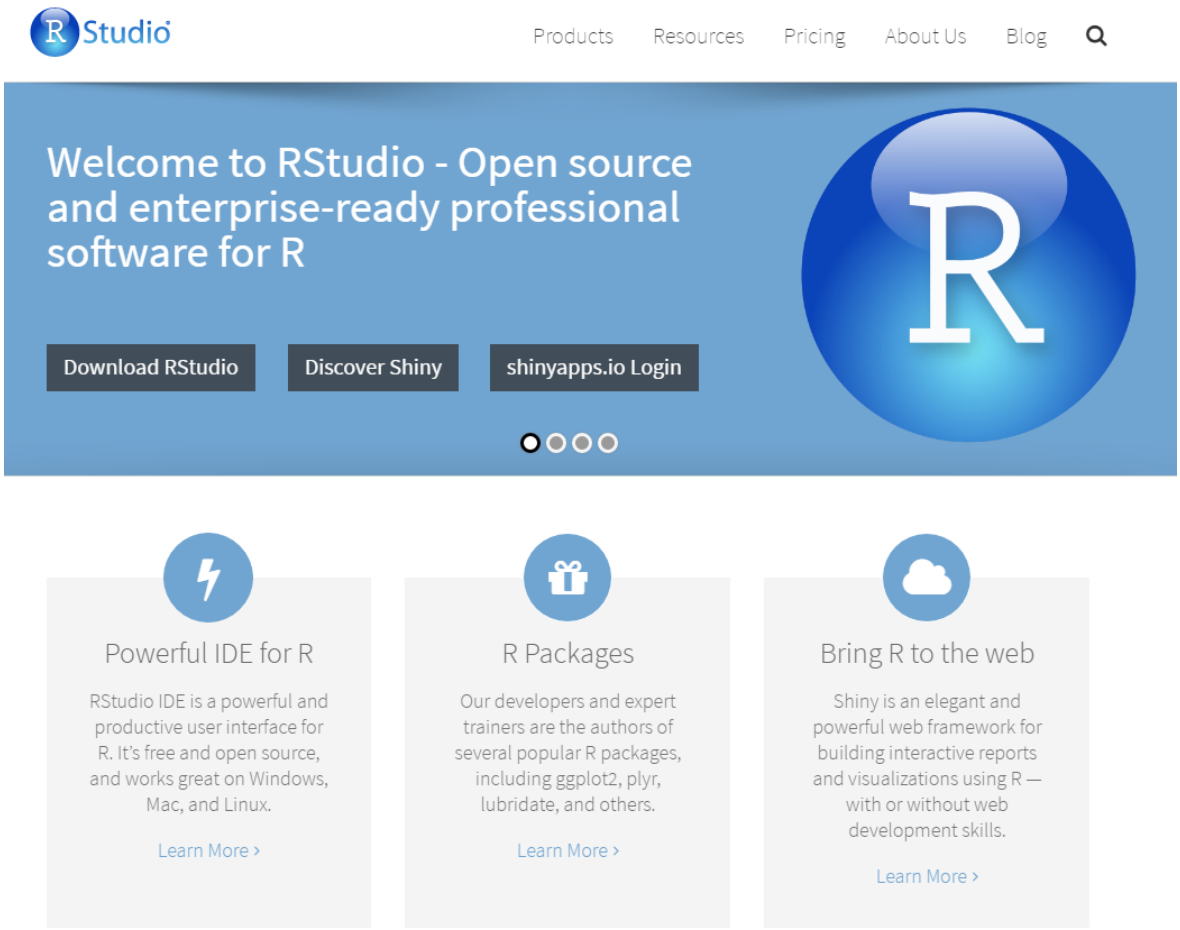
za nastanak imena novog jezika (Ross, Robert), a sam projekt smišljen je 1992. godine, dok je prva verzija ugledala svjetlo dana 1994. godine.

2.3 Instalacija programa

Instalacija programa je relativno jednostavna, potreban je pristup internetu gdje jednostavnim upisom web adrese ***http://cran.r-roject-org/*** (15.9.2016.) u pregledniku pritupamo web stranici na kojoj je moguće odabrati različite verzije programa ovisno o programskom sustavu koji se koristi. Nakon odabira verzije, program će se skinuti na računalo gdje instalaciju pokrećemo klikom na skinutu datoteku.

2.4 RStudio

$RStudio$ je integrirano razvojno okruženje (engl. IDE) za R. On je kao i R također besplatan, a programska podrška je slobodnog dohvata (engl. Open-source software) kao i kod R-a. Osmislio ga je Joseph J. Allaire, tvorac programskog jezika $ColdFusion$.

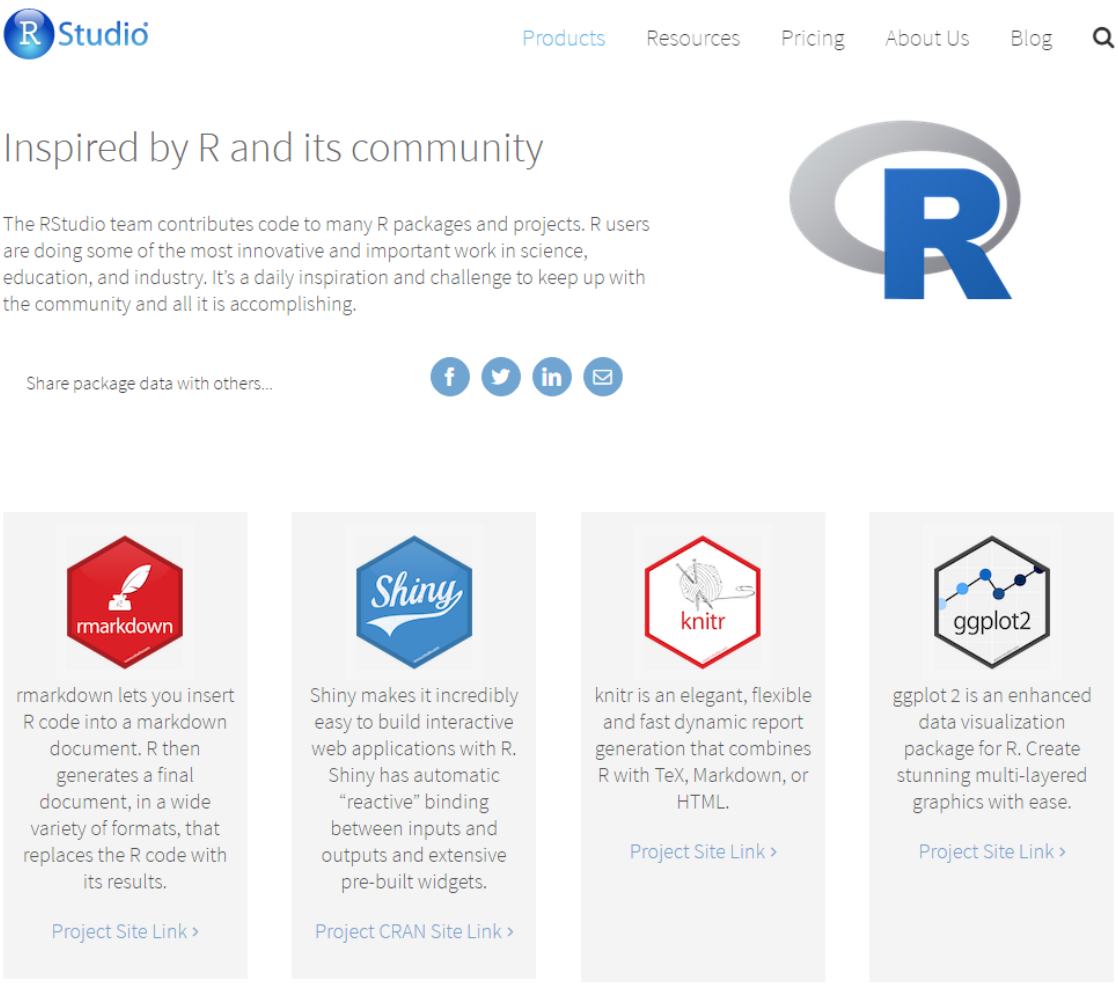


Slika 2.2: Prikaz početne stranice https://www.rstudio.com/home/ (15.9.2016)

Dostupan je u dva izdanja: prvo, *RStudio* za radnu površinu, pri čemu se program pokreće lokalno, kao klasična desktop aplikacija; drugo, *RStudio* Server koji omogućuje pristup *RStudiu* pomoću web-preglednika. Trenutno su dostupne programske inačice za najzastupljenije programske sustave: *Windows*, *Linux*, *MacOS*.

Instalacija programa je kao i kod samog R-a relativno jednostavna, potreban je samo pristup internetu gdje se jednostavnim upisom web adrese ***https://www.rstudio.com*** (15.9.2016.) u preglednik pristupa web stranici gdje je moguće odabrati različite verzije programa ovisno o programskom sustavu koji se koristi. Nakon odabira verzije, program će se skinuti na računalo nakon čega instalaciju pokrećemo klikom na skinutu datoteku.

S obirom da *Rstudio* dolazi s integriranim osnovnim paketom funkcija, a potrebe korisnika Rstudia su različite, postoji mnogo različitih paketa s posebno prilagođenim funkcijama i mogućnostima. Neki od češće korištenih su *Rmarkdown*, *Shiny*, *Knitr* i *ggplot2*. *Rmarkdown* služi za umetanje R koda u neki od opće prihvaćenih formata dokumenata kao što su PDF, *Micorsoft Word* i ostali. *Shiny* služi za izradu



Slika 2.3: Prikaz paketa dostupnih u Rstudiu (15.9.2016.)

interaktivnih web aplikacija i čini je nevjerojatno jednostavnom. *Knitr* je elegantan, fleksibilan i brz generator izvještaja koji kombinira *R* s *TeX*, *HTML* i ostalim formatima. *ggplot2* je paket za poboljšanu vizualizaciju podataka analiziranih u R-u, služi za laku i jednostavnu izradu višeslojne grafike. Ovo su samo neki od paketa, kompletan popis i više informacija o paketima može se dobiti na web stranici ***https://www.rstudio.com/products/rpackages/*** (15.9.2016.).

2.5 Pomoć u R-u

Ukoliko imamo nedoumica i želimo dobiti nešto više informacija o određenoj funkciji ili paketu, najbolja polazna točka za to je CRAN web-mjesto gdje postoji detaljan opis svih postojećih funkcija, paketa i naredbi koje možemo susresti u R-u.

Možemo ga pokrenuti jednostavnim upisom `help.start()`, ili ako želimo biti precizniji `help("ime funkcije")`. Ukoliko nam je točan naziv funkcije nepoznat, možemo iskoristiti `help.search(dio naziva)` koji pretražuje pomoć i prikazuje gdje

se sve nalazi traženi pojam. Želimo li pak primjer za određenu funkciju/naredbu možemo iskoristiti `example(dio naredbe)`. Dio pomoći i informacija je u $R-$u jako dobro pokriven i možemo reći kako je CRAN ujedno i najbolje mjesto za početničko upoznavanje sa R-om.

```
help.start()

help("while")

example("plot")
```

2.6 Korisničko sučelje Rstudia

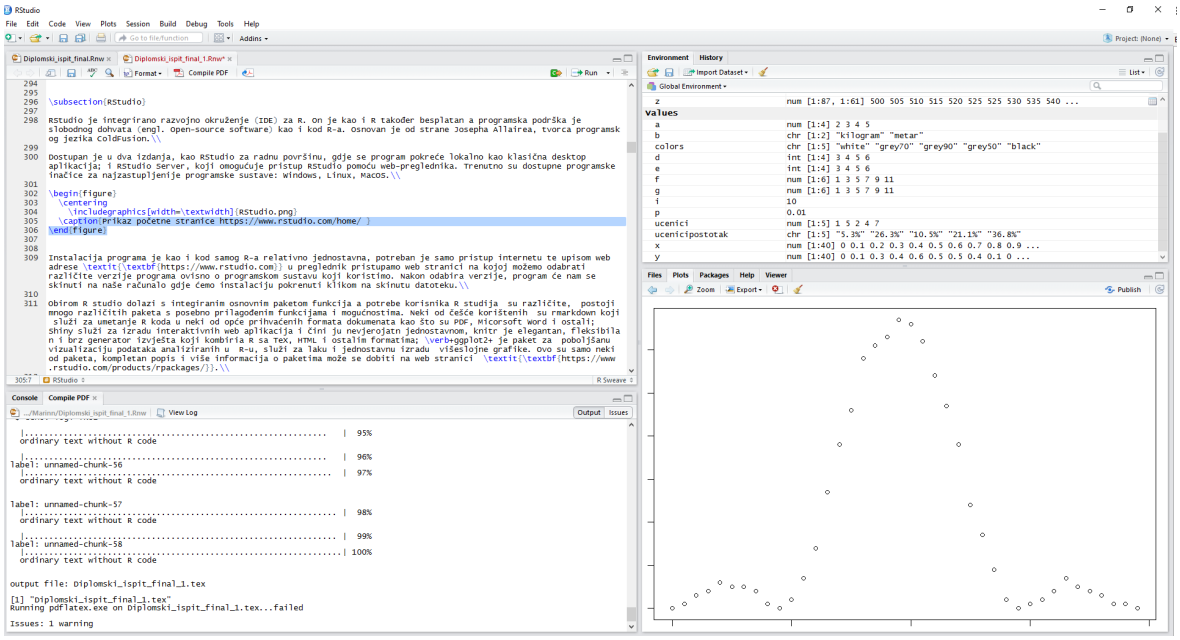
Obradu podataka i većinu aktivnosti odrađivati ćemo u $Rstudiu$ pa je korisno dobro se upoznati s njegovim sučeljem. Korisničko sučelje R-a je relativno jednostavno (slika br. 2.4) te kao i većina drugih programa sadrži traku izbornika(engl. Menu bar) s opcijama $File$, $Edit$, $Code$, $View$, $Plots$, $Sessions$, $Build$, $Debug$, $Tools$ i $Help$.

Uz navedeno sučelje, $RStudio$ sadrži i tri osnovna prozora. U prozor konzole (engl. Console) moguć je unos pojednine naredbe (koristi se jednostavnije operacije) pri čemu se u istom prozoru ispisuju rezultati numeričke analize. U prozoru s radnom površinom (engl. Workspace) vidljiva je lista aktivnih objekata te povijest korištenih funkcija i objekata. U prozoru dokumenata prikazani su svi dokumenti, mape kao i grafički prikazi te dostupni paketi na radnoj površini. Usto, ondje se može pronaći i dio za pomoć s detaljnom dokumentacijom gdje se uvijek može dodatno istražiti nejasni dio programa. Uz spomenuta tri osnovna, $Rstudio$ sadržava i četvrti prozor, skriptu (engl. Script). Ona ima numerirane redove te služi za pisanje složenijih kodova i programa. Drugim riječima, u tom prozoru obavlja se sav posao koji se kasnije može pokrenuti i provjeriti u konzoli.

2.6.1 Konzola

Već je ranije spomenuto da se sve jednostavne naredbe i operacije uglavnom obavljaju u konzoli. Kad očekuje naredbe za unos podataka, R program nam daje znak >(prompt). Riječ je o defaultnom znaku koji se jednostavno može zamijeniti s nekim drugim.

Započnimo s elementarnim primjerom - potrebno je izračunati jednostavan matematički izraz $5 + (45 + 4) - ((12 + 10) - (174 + 18))$. Rješenje će se dobiti unosom u konzolu.



Slika 2.4: Prikaz sučelja u R studiu (15.9.2016.)

```
5+(45+4)-((12+10)-(174+18))

## [1] 224
```

Iz primjera se vidi kako je jednostavno izračunati matematičke operacije zbrajanja, oduzimanja i grupiranja brojeva. Isto vrijedi i za ostale elementarne operacije (množenje, dijeljenje, potenciranje, trigonometrijske funkcije i ostalo).

Računanje elementarnih operacija:

```
5*5

## [1] 25

50/5

## [1] 10

5**5

## [1] 3125

exp(1)

## [1] 2.718282

sin(2*pi) #kutevi su u radijanima

## [1] -2.449213e-16

cos(0)

## [1] 1
```

2.6.2 R skripta

Želimo li izračunati matematički složenije operacije koristeći funkcije i argumente, praktičnije je rad prebaciti u prozor R skripte radi bolje preglednosti koju omogućuju numerirani redovi (biti će jako korisno prilikom pisanja složenijih kodova). Kod zapisan u skripti njezinim će pokretanjem (opcija Run) dati rezultat u konzoli.

2.7 Vektori

S obzirom da vektori predstavljaju osnovnu veličinu u fizici, dobro je krenuti od njih. Želimo li generirati određeni vektor, koristimo `c` (engl. concatenate), a za spajanje određenih dijelova vektora, zadat ćemo vektor `a`.

```
a <-c(2,3,4,5)
a

## [1] 2 3 4 5
```

Vektoru `a` smo pridružili vrijednosti (2,3,4,5) i njega nazivamo numeričkim vektorom.

```
b <-c("kilogram","metar")
b

## [1] "kilogram" "metar"
```

Vektoru `b` smo pridružili vrijednosti (”kilogram, ”metar”) i njega nazivamo vektor stringova.

```
c <-c(T,F,T,T,F)
c

## [1]  TRUE FALSE  TRUE  TRUE FALSE
```

Vektoru `c` smo pridružili vrijednosti (T,F,T,T,F) i njega nazivamo logičkim vektorom.

Logičke operacije često se koriste i za indeksiranje vektora logičkim vektorom, kod pisanja različitih uvjeta koristimo sljedeće logičke vrijednosti:

- == jednako (engl. *equal ton*)
- != nije jednako (engl. *not equal to*)
- > manje (engl. *smaller*)
- < veće (engl. *larger*)

Postoje i logičke operatori $\wedge$ (and) koji označavamo s `&`, $\vee$ (or) koji označavamo s `|`, i $\neq$ (not) koji ozncavamo s `!`.

S vektorima možemo obavljati i različite operacije kao što su zbrajanje, oduzimanje, množenje, potenciranje i slično:

- **x+y** zbrajanje po koordinatama vektora iste duljine
- **x-y** oduzimanje po koordinatama vektora iste duljine
- **x*y** množenje po koordinatama vektora iste duljine
- **x*c** množenje vektor x skalarom c
- **x**n** potenciranje po koordinatama vektora x

Mogu se koristiti i različite funkcije kao što su:

- **length(x)** koristi se za izračun duljine vektora x
- **sum(x)** koristi se za izračun sume elemenata vektora x
- **prod(x)** koristi se za izračun produkta elemenata vektora x
- **sort(x)** koristi se za sortiranje elemenata vektora x
- **split(x,y)** koristi se za razdvajanje elementa vektora x prema vrijednostima

2.8 Aritmetički nizovi

Želimo li generirati određeni aritmetički niz, koristimo funkciju `seq` (od eng. sequence) za generiranje aritmetičkih nizova, a umjesto te naredbe može se upisati i određeni raspon brojeva.

```
d <-seq(3,6)
d

## [1] 3 4 5 6

e <-3:6
e

## [1] 3 4 5 6
```

Naredbom `seq` kreirali smo aritmetički niz cijelih brojeva: 3 4 5 6. Prikazana su dva ekivalentna načina.

```
f <-seq(from=1,to=12,by=2)
f

## [1]  1  3  5  7  9 11

g <-seq (1,12,2)
g

## [1]  1  3  5  7  9 11
```

Naredbom `seq(from=1,to=12,by=2)` kreirali smo aritmetički niz brojeva 1 3 5 7 9 11, odnosno niz od 1 do 12, s korakom 2.

```
seq(1,10,length=7)

## [1]  1.0  2.5  4.0  5.5  7.0  8.5 10.0
```

Naredbom `seq(1,10,length=7)` kreirali smo aritmetički niz realnih brojeva 1.0 2.5 4.0 5.5 7.0 8.5 10.0, odnosno niz od 7 članova u rasponu od 1 do 10.

Ako želimo ponavljanje istih vrijednosti, koristimo funkciju `rep` (eng. replicate) te kao argumente upisujemo raspon brojeva i broj ponavljanja.

```
rep(1:4,2)

## [1] 1 2 3 4 1 2 3 4
```

Naredbom `rep` (eng. *replicate*) kreiran je aritmetički niz brojeva 1 2 3 4 ponovljen 2 puta.

```
rep(1:4,each=3)

##  [1] 1 1 1 2 2 2 3 3 3 4 4 4
```

Argumentom $each = 3$ kreiran je aritmetički niz brojeva 1 2 3 4 u kojem se svaki broj ponavlja tri puta.

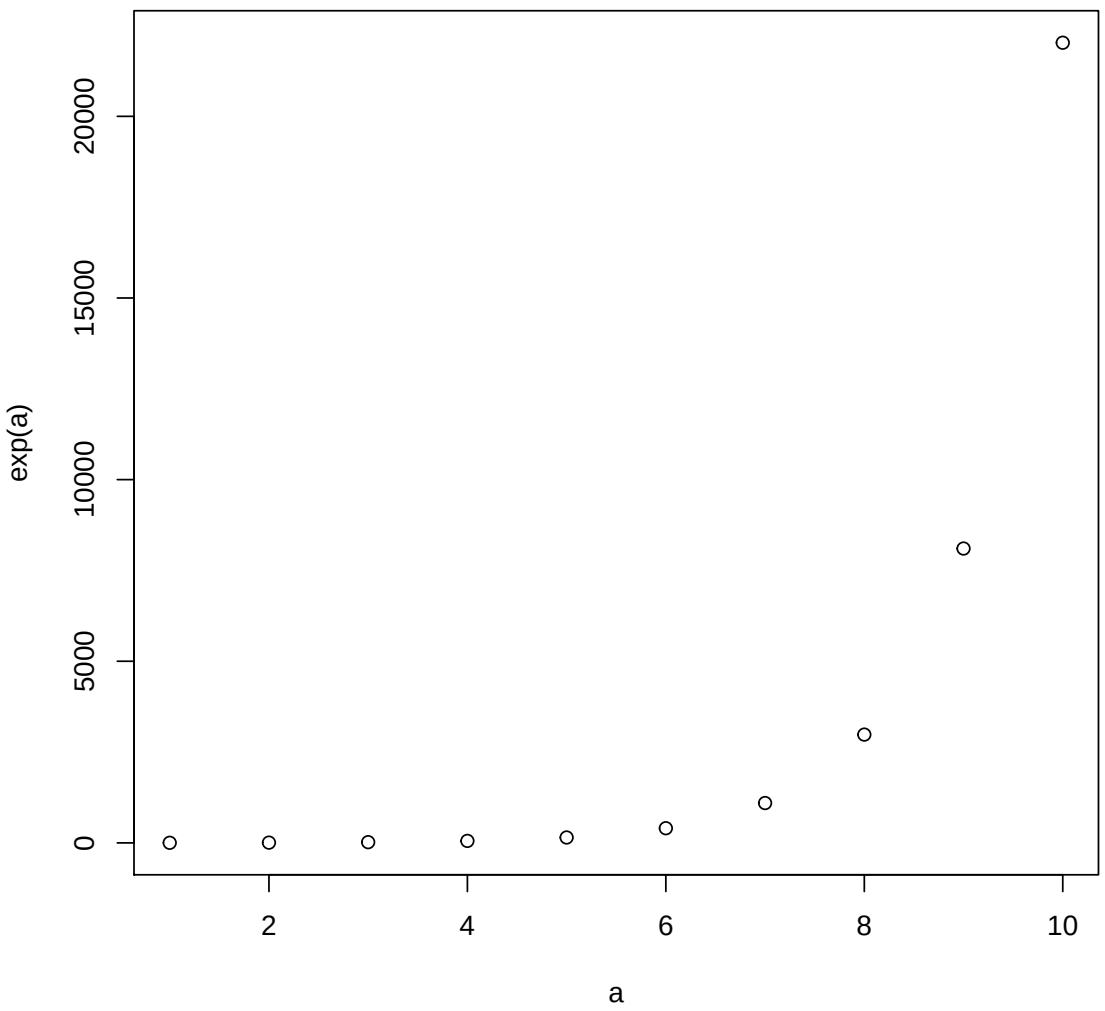
2.9 Funkcije i argumenti

Kod pozivanja funkcija, argumenti su obično navedeni pozicijom ili imenom.

```
a<-(1:10)
```

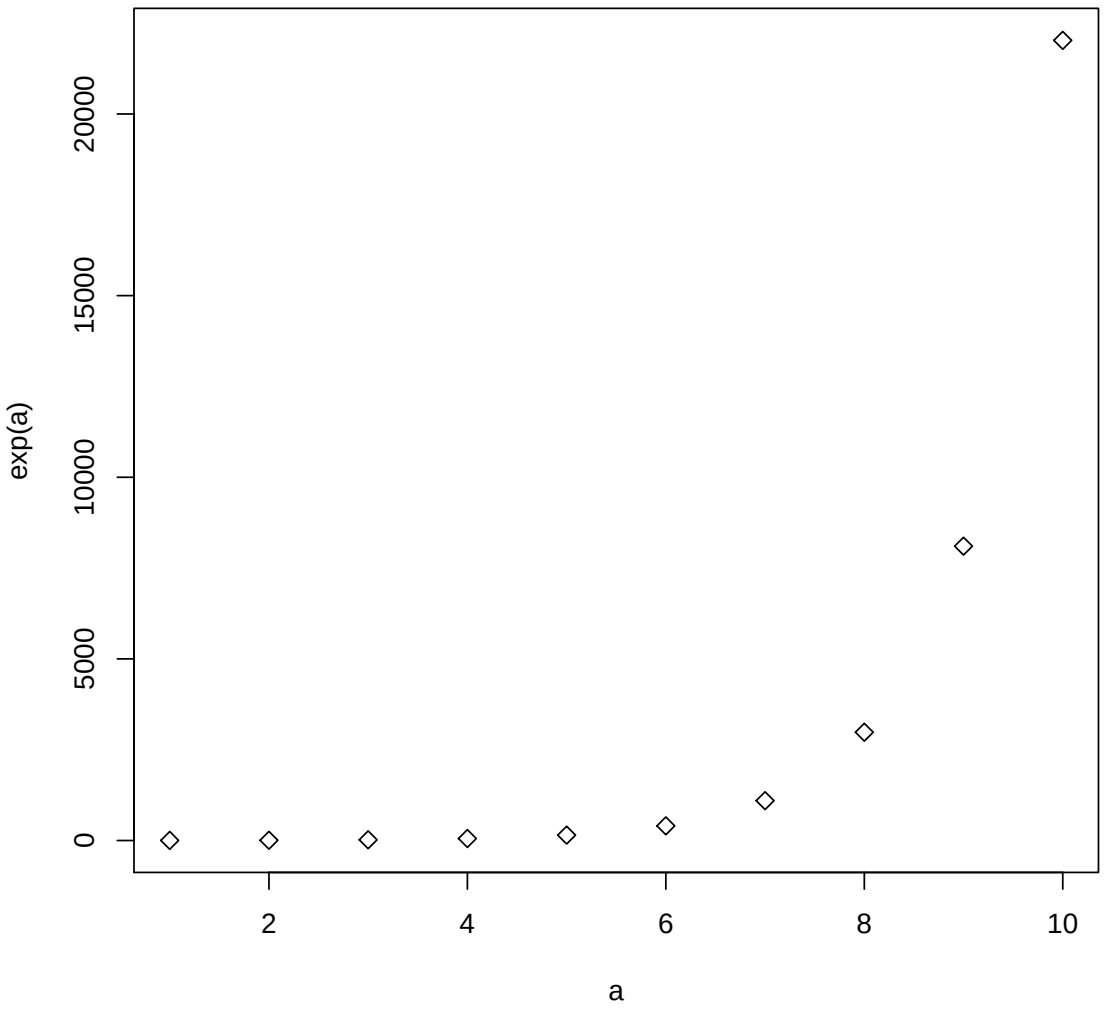
Vektor a se sastoji od prvih 10 prirodnih brojeva te ga možemo uvrsiti u eksponencijalnu funkciju i grafički prikazati pomoću funcije `plot()` čiji su argumenti zadani pozicijom u funkciji.

```
plot(a,exp(a))
```



Argument `pch` označava simbol kojim se crtaju točke na grafu.

```
plot(a,exp(a),pch=5)
```

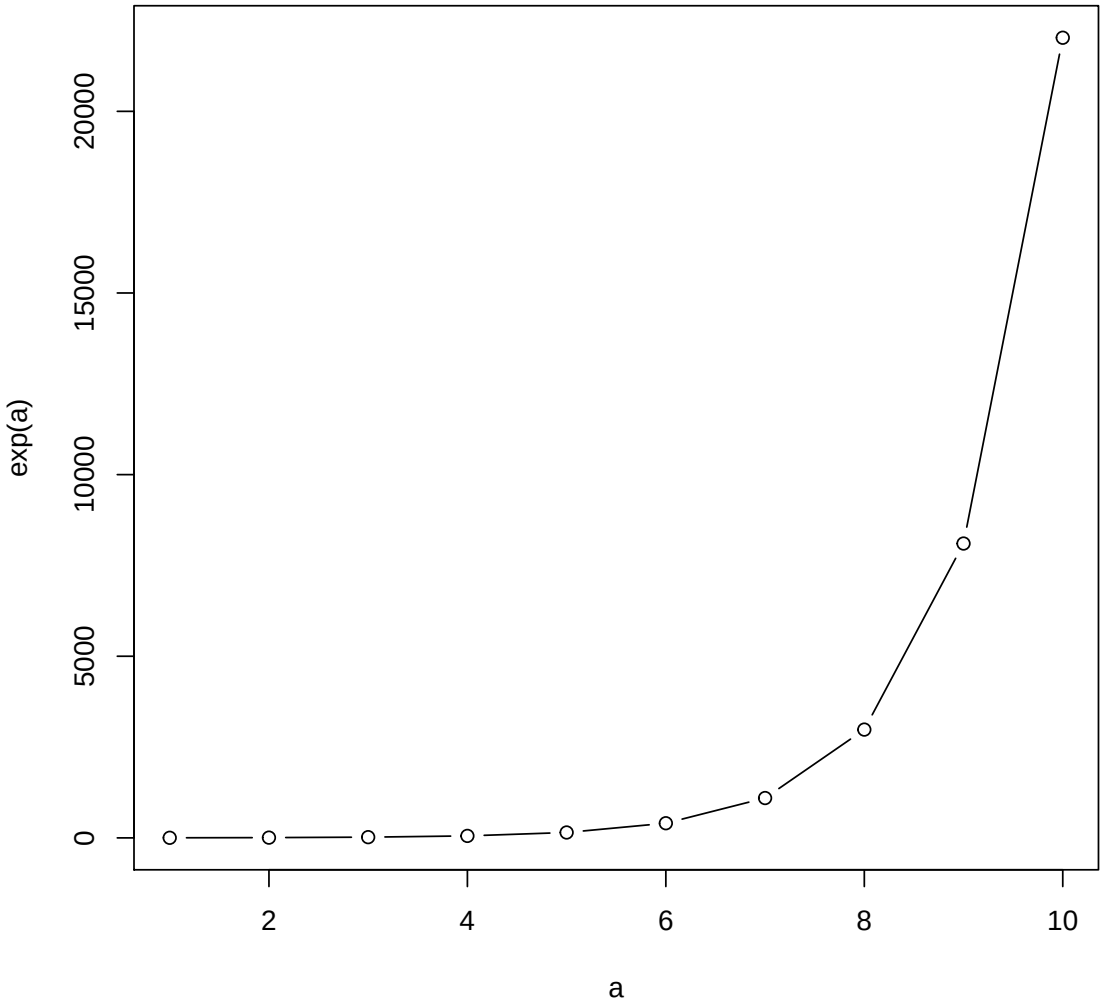


Većina argumenata funkcije ima pretpostavljene vrijednosti i njih možemo vidjeti naredbom `args()`:

```
args(plot.default)

## function (x, y = NULL, type = "p", xlim = NULL, ylim = NULL,
##     log = "", main = NULL, sub = NULL, xlab = NULL, ylab = NULL,
##     ann = par("ann"), axes = TRUE, frame.plot = axes, panel.first = NULL,
##     panel.last = NULL, asp = NA, ...)
## NULL
```

Ukoliko ne specificiramo tip argumenta, funkcija će sama uzeti prvi argument prema listi prioriteta. U primjeru funkcija plot se ponaša kao da smo stavili `type="p"`. Upisom `type="b"` dobivamo:

```
plot(a,exp(a),type="b")
```



graf `exp()` funcije gdje su nam vrijednosti prikazane povezanim točkama. Unosom `help(plot)` možmo vidjeti tip argumenata za funkciju `plot()`.

2.9.1 Pisanje vlastitih funkcija

R nam dozvoljava kreiranje objekata tipa funkcija, odnosno možemo pisati vlastite funkcije. One su funkcije ekvivalentne ugrađenim R-ovim funkcijama, pohranjene su u okruženju našeg koda, a mogu se upotrebljavati u drugim izrazima. Pisanjem vlastitih funkcija programski jezik R dobiva sve više na snazi i spretnosti, pa je učenje pisanja korisnih funkcija dobar način kako pojednostaviti rad i povećati produktivnost.

Većina funkcija R-ovog sustava (pr. `mean( )`, `var( )` i dr.) nemaju većih razlika od funkcija koje pišu pojedini korisnici. Upravo u tome leži velika snaga i fleksibil-

nost R-a.

Funkcija se definira pridruživanjem oblika:

```
ImeFunkcije <-function(arg_1, arg_2, ...) {expression}
```

`ImeFunkcije` je naziv same funkcije, `arg_1`, `arg_2` su argumenti funkcije, a `expression` je neki R izraz koji koristi argumente kako bi se izračunala neka vrijednost. Vrijednost izraza je vrijednost koju daje funkcija. Jednostavan primjer nam je suma kvadrata.

```
SumaKvadrata<-function(x,y){
  x^2 + y^2
}
SumaKvadrata(5,5)

## [1] 50
```

2.10 Liste

Liste su oblik strukture podataka koje omogućavaju kombiniranje objekata različitih tipova. Na primjer, lista se može sastojati od numeričkog vektora, logičke vrijednosti, matrice, složenog vektora, znakovnog polja, funkcije i slično. Komponente liste su uvijek numerirane, a konstruiramo ih naredbom `list()`.

Evo primjera liste koja se sastoji od numeričkog vektora, logičkog vektora i faktora:

```
Marin=list(x=c(111,12,222,21),y=c(FALSE,TRUE,TRUE,TRUE),
           z=factor(c(0,1,1,0,1)))
Marin

## $x
## [1] 111  12 222  21
##
## $y
## [1] FALSE  TRUE  TRUE  TRUE
##
## $z
## [1] 0 1 1 0 1
## Levels: 0 1
```

Ako je `Marin` neka lista, tada funkcija `length(Marin)` daje broj komponenata koje lista sadrži. Ukoliko su komponentama pridružena imena, isto ispisujemo na sljedeći način:

```
length(Marin)

## [1] 3

Marin$x

## [1] 111  12 222  21
```

2.11 Polja i matrice (Data frames)

Polje možemo definirati kao skup podataka s višestrukim indeksima. Matematički, najbolji primjer nam je polje realnih, cijelih brojeva ili skup logičkih vrijednosti. U fizici, polje je preslikavanje koje pridružuje neku fizikalnu veličinu svakoj točki prostora. Polje može biti skalarno, vektorsko ili tenzorsko, ovisno o tipu veličine koja se pridružuje. Tako je, na primjer, gravitacijsko polje vektorsko polje, zato što svakoj točki prostora pridružuje vektor u čijem smjeru djeluje gravitacija.

Pomoću R-a je jednostavno kreiranje i rukovanje s poljima, a poseban slučaj su matrice (dvodimenzionalno polje s posebnim svojstvima). Riječi dimenzija ne trebamo davati nikakvo duboko značenje jer ona ovdje samo opisuje broj indeksa skupa podataka.

U matematici i fizici matrica je dobro definirana struktura, možemo reći da su matrice 2 dimenzionalna polja (array) realnih ili kompleksnih brojeva sa posebno definiranim matematičkim operacijama.

Matrice možemo još nazvati i tablicama koje se sastoje od apstraktnih objekata koje zbrajamo i množimo. Koristimo ih za opisivanje linearnih jednadžbi, za praćenje koeficijenata linearnih transformacija i za čuvanje podataka koji ovise o više parametra.

Nas zanima struktura podataka u kojoj su vektori i/ili faktori iste duljine složeni u matricu kao stupci.

```
rbind (1:5,1:5)

##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    3    4    5
## [2,]    1    2    3    4    5
```

Vezali smo vektor s vrijednostima od jedan do pet za pojedini red u matricu s dva retka i pet stupaca.

```
cbind (1:5,1:5)

##      [,1] [,2]
## [1,]    1    1
## [2,]    2    2
## [3,]    3    3
## [4,]    4    4
## [5,]    5    5
```

Vezali smo vektor s vrijednostima od jedan do pet za pojedni stupac u matricu s pet redaka i dva stupca.

```
t(cbind(1:5,1:5))

##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    3    4    5
## [2,]    1    2    3    4    5
```

Da bi dobili prvu matricu dovoljno je transponirati.

```
matrix (1 ,2 ,3)

##      [,1] [,2] [,3]
## [1,]    1    1    1
## [2,]    1    1    1
```

Kreirali smo jediničnu matricu s dva retka i tri stupca

```
matrix(1:6,nrow = 2)

##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

Kreirali smo matricu sa dva retka i tri stupca iz originalnog raspona od jedan do šest.

2.12 Izrada spremnika podataka

Spremnikom podataka nazivamo listu komponenata koja ima klasu `data.frame`. Dako bi se lista pretvorila u spremnik podataka sve komponente te liste moraju

biti vektori (numerički, znakovni ili logički), faktori, brojčane matrice, liste ili drugi spremnici podataka. Također, matrice, liste i spremnici podataka daju onoliko varijabla novome spremniku koliko imaju stupaca, elemenata odnosno varijabli. Numerički vektori, logički vektori i faktori uključuju se kakvi jesu. Znakovni vektori spajaju se u faktore čiji su nivoi jedinstvene vrijednosti koje se pojavljuju u vektoru. Treba napomenuti da sve vektorske strukture koje se pojavljuju kao varijable spremnika moraju imati istu duljinu, a sve matrice moraju imati istu veličinu retka.

Ako su zadovoljene restrikcije postavljene za stupce (elemente) spremnika, tada se funkcija `data.frame` može upotrijebiti za konstruiranje spremnika podataka:

```
ZakoniFizike<-data.frame("Maxwell=Da","Newton=da","Kazneni=ne","Keppler=da")
```

Naredba `data.frame` koristi se i za konstrukciju matrica:

```
SIjednice<-c("Sekunda","Kelvin","Kilogram","Metar", "Amper")
Oznaka<-c("s","K","kg","m","A")
Vrijednost<-c(60, 10, 100, 1, 1)
d=data.frame(SIjednice,Oznaka,Vrijednost)
d

##   SIjednice Oznaka Vrijednost
## 1   Sekunda      s         60
## 2    Kelvin      K         10
## 3  Kilogram     kg        100
## 4     Metar      m          1
## 5     Amper      A          1
```

Također, funkcija `data frame` ima i dodatan redak s imenima objekata (ako su zadana) i dodatan stupac s imenima redaka koja možemo mijenjati. Fukcija `names()`

```
names(d)

## [1] "SIjednice"  "Oznaka"     "Vrijednost"
```

ispisuje imena objekata, dok funkcija `row.names()`

```
row.names(d)

## [1] "1" "2" "3" "4" "5"
```

ispisuje imena redaka. Postoje i funkcije koje ispisuju dio `data.framea`. Obično taj dio zadovoljava zadani kriterij. Na primjer, funkcija `subset()`

```
subset(d,Vrijednost>15)

##   SIjednice Oznaka Vrijednost
## 1   Sekunda      s         60
## 3  Kilogram     kg        100
```

uz argument `Vrijednost>15` ispisuje sve vrijednosti veće od 15.

Najjednostavniji način konstruiranja novog spremnika podataka je upotrebom funkcije `read.table( )` za čitanje jednog cijelog spremnika iz vanjske datoteke. Uzmimo za primjer mjerenja.

```
OgibMjerenja<-read.table("OgibMjerenja.txt",sep=",",header=TRUE)
kable(head(OgibMjerenja))
```

Pomak	Napon
0.0	0.0
0.1	0.1
0.2	0.3
0.3	0.4
0.4	0.6
0.5	0.5

Često se koristi još i `read.csv()`, a tu je i jednostavnija funkcija za unos, `scan( )`.

2.13 Petlje

Petlje koristimo prilikom repetativnih operacija na velikoj količini podataka. R omogućava standardne petlje kao što su `if()`, `for()`, `while()`

```
#If-petlja
p<-0.01
if(p<0.05) print("Marin piše diplomski!")

## [1] "Marin piše diplomski!"

#For-petlja:
for(i in 1:5) print(i)

## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

```r
#While-petlja:
i<-2
while(i<5) {i <- i+1; print(i);}

## [1] 3
## [1] 4
## [1] 5
```

2.14 Optimizacija u R-u

U R-u se može napraviti i optimizacija. Jedna od osnovnih funkcija jest jednodimenzionalna minimizacija funkcije u zadanom intervalu. Ona se zasniva na kombinaciji metode zlatnog veza i metode parabole, a može se izvršiti naredbom `optimize()`. Višedimenzionalna minimizacija izvršava se naredbom `optim ()` dok na primjer, funkcija optimx () objedinjuje desetak različitih optimizacijskih algoritama. Pomoću nje može se istovremeno isprobati više metoda i usporediti rezultate. Također, postoji još i velik broj funkcija u preko dvadeset paketa, kao i procedure za sve vrste matematičkog programiranja (linearno, nelinearno, cjelobrojno...).

2.15 Grafika u R-u

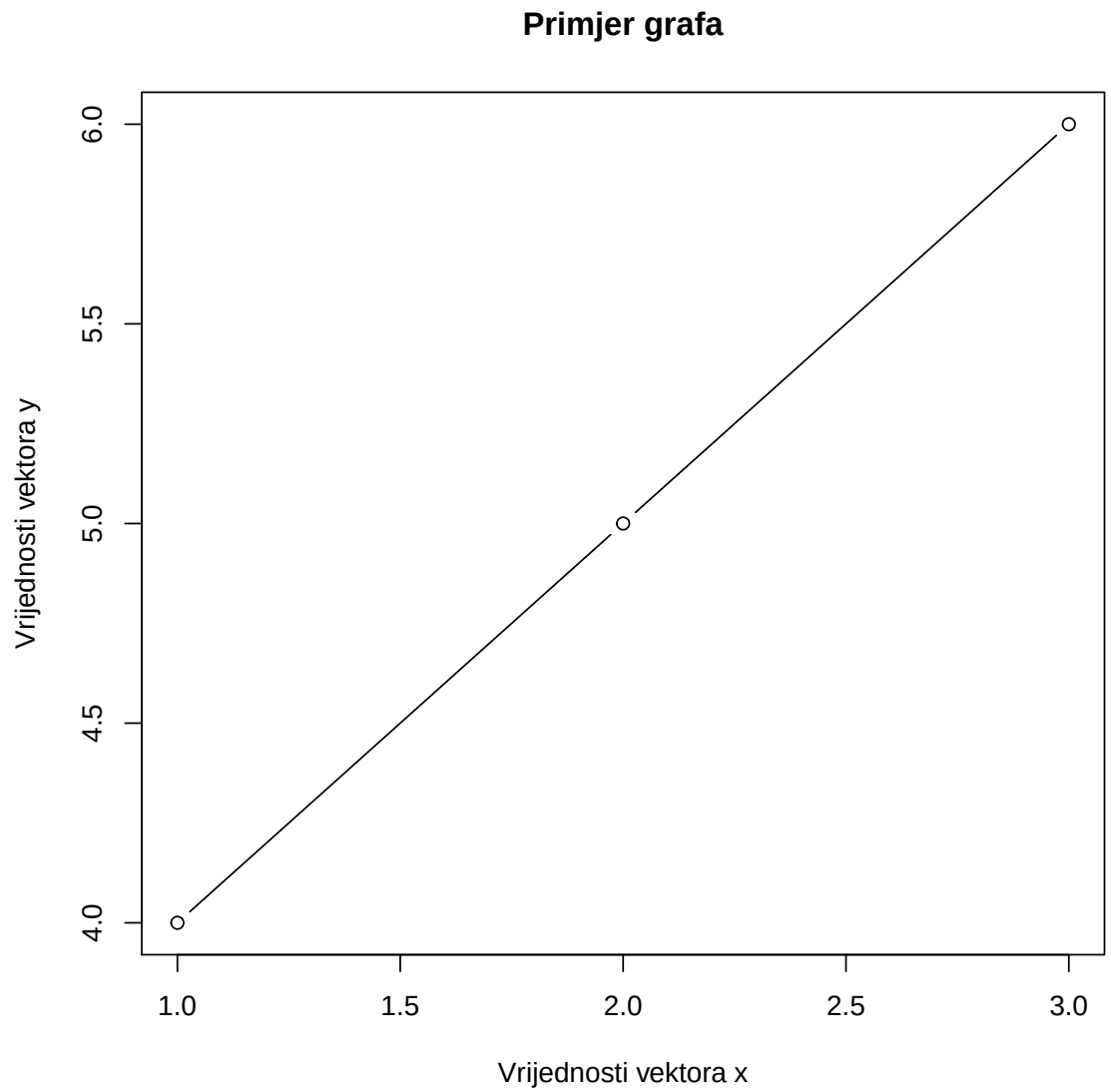
Grafički je dio dobro pokriven u R-u. Sve se grafika temelji na dva grafička sustava pod nazivom `graphics` (osnovni paket) i `grid` (prošireni paket). Osnovni grafički sustav sadržan je u paketu `graphics` i on predstavlja standardni paket. Iz tog osnovnog paketa, izvedeni su i brojni drugi paketi, a sve funkcije iz grafičkih sustava dijele se u tri osnovne grupe:

- funkcije za grafike visoke razine kreiraju novi grafikon na grafičkom podsustavu, moguće s osima, oznakama, naslovima i tako dalje;
- funkcije za grafike niske razine dodaju više informacija na postojeći grafikon kao što su dodatne točke, linije i oznake;
- funkcije za interaktivne grafike omogućuju interaktivno dodavnje informacija na postojeći grafikon ili čitanje informacija izravno sa grafa.

Dobivene slike mogu se spremati u uobičajene formate: PDF, PNG, $JPEG$ i ostale.

Jednostavna i najčešće korištena grafička funkcija je funkcija `plot()`. Radi se o generičkoj funkciji, što znači da može primati različite objekte kao argumente, a vrsta grafa koji se kreira ovisi o vrsti ili klasi prvoga argumenta.

```r
x<-c(1,2,3)
y<-c(4,5,6)
plot(x,y,type="b", main="Primjer grafa",
     xlab="Vrijednosti vektora x",
     ylab="Vrijednosti vektora y")
```

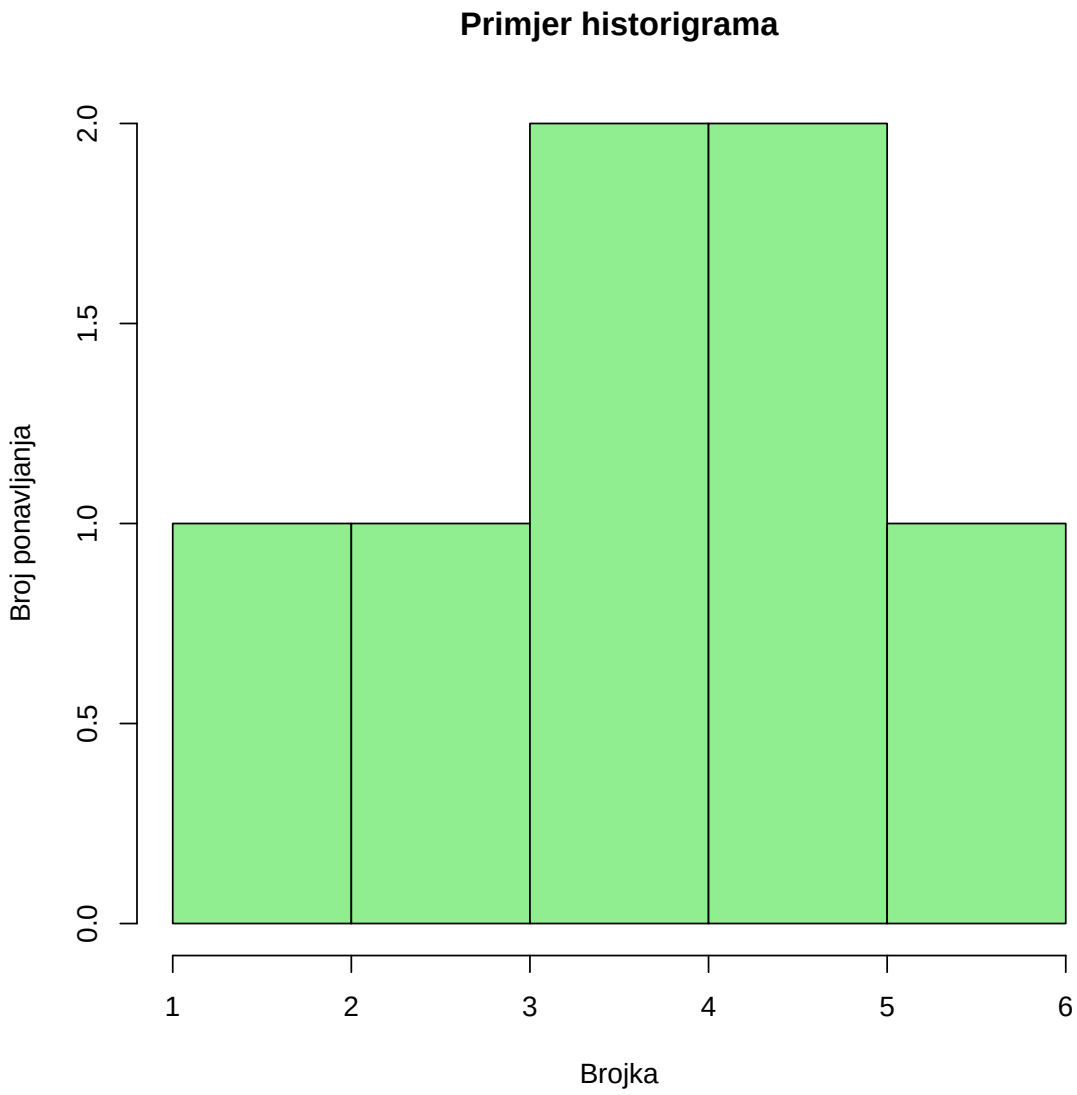


X i y su vektori, a funkcija `plot(x,y)` kao rezultat daje točkasti graf `(scatterplot)` sa prikazom zavisnoti `y`-a o `x`-u.

Ako varijabla `x` predstavlja vremenski niz, bit će prikazan graf vremenskih nizova. Ako je `x` numerički vektor, bit će prikazan graf vrijednosti vektora u zavisnosti od indeksa u vektoru. Ako je `x` vektor kompleksnih brojeva, bit će prikazan graf imaginarnog dijela komleksnog broja u odnosu na realni dio.

Osim funkcije `plot()`, R sadrži još dosta zanimljivih funkcija kao što je `hist(x)` koja služi za konstrukciju histograma:

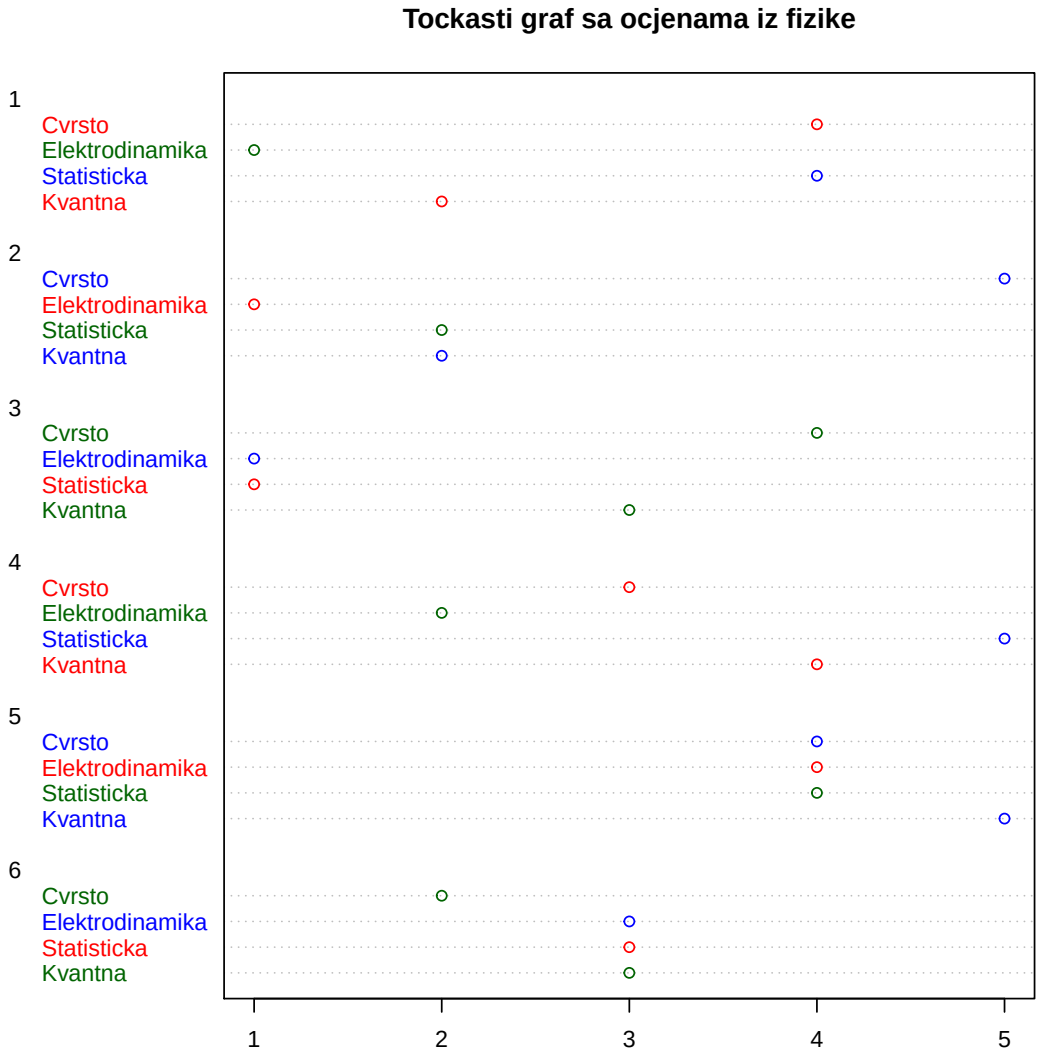
```r
y<-c(4,5,5,6,3,1,4)
hist(y,freq = TRUE,col="light green",
     main="Primjer historigrama",
     xlab ="Brojka",ylab ="Broj ponavljanja")
```



Prikazan je broj ponavaljanja vrijednosti vektora `Y`.

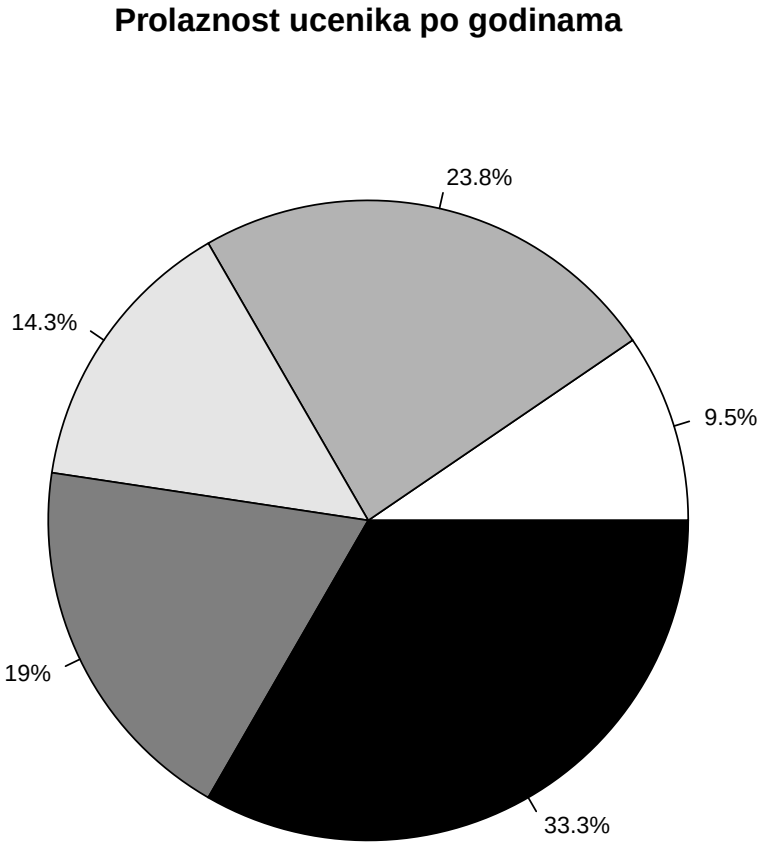
Funkcija `dotchart()` služi za konstrukciju točkastog grafa vrijednosti sadržanih u određenoj datoteci. Ovim grafom moguć je lagan vizualni prikaz svih unesenih podataka s vrijednostima koje leže u točno određenim rasponima.

```r
#setwd("C:/Users/Marin Luketic/Desktop/Marinn")
ocjenefizika<-read.csv("primjer2.csv",sep=";",header=TRUE)
dotchart(t(ocjenefizika), color=c("red","blue","darkgreen"),
   main="Točkasti graf sa ocjenama iz fizike", cex=0.8)
```



Također, zanimljiva je funkcija `pie()` koja služi za precizan prikaz udjela određenih numeričkih vrijednosti u ukupnoj vrijednosti.

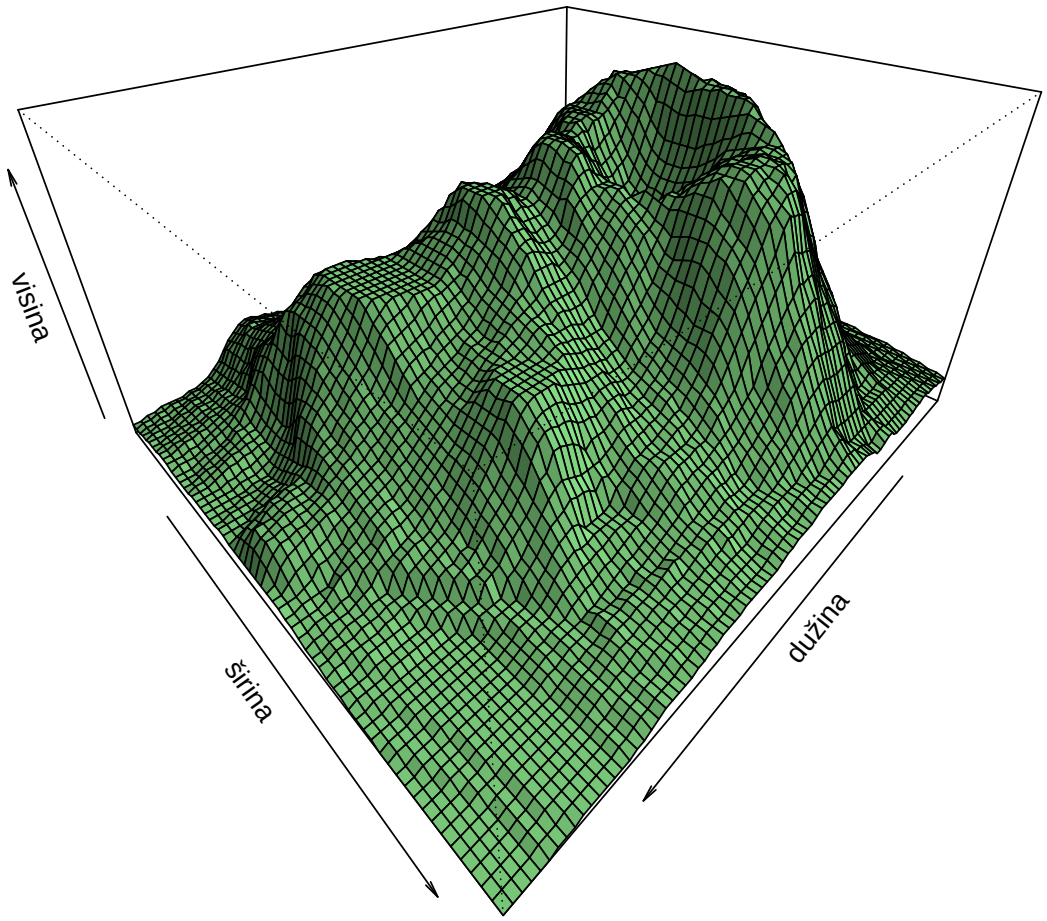
```
ucenici<-c(2, 5, 3, 4, 7)
colors<-c("white","grey70","grey90","grey50","black")
ucenicipostotak<-round(ucenici/sum(ucenici)*100, 1)
ucenicipostotak<-paste(ucenicipostotak, "%",sep="")
pie(ucenici, main="Prolaznost učenika po godinama", col=colors,
    labels=ucenicipostotak, cex=0.8)
legend(1.5, 0.5, c("Prva","Druga","Treća","Četvrta","Peta"), cex=0.8,
   fill=colors)
```



R također nudi i napredne mogućnosti vezane u 3D prikaz. Uz naredbu `persp()` crtamo 3D plohu:

```
z<-5*volcano
x<-11*(1:nrow(z))
y<-14*(1:ncol(z))
par(mar=rep(1,4))
persp(x,y,z,theta=135,phi=30,r=0.2,col="light green",
      scale =FALSE,ltheta=-120,lphi=30,shade=0.65,
      main="Primjer 3D grafa", xlab="dužina", ylab="širina",
      zlab="visina")
```

Primjer 3D grafa



3 Statistička obrada podataka

3.1 Mjerenja, preciznost i pogreške

3.1.1 Mjerenja

Mjerenja su prvi i osnovi korak svake fizikalne analize podataka i nužno ih je odraditi kvalitetno i precizno. Cilj svakog fizikalnog mjerenja je saznati brojčanu vrijednost koja predstavlja određenu, promatranu fizikalnu veličnu. Nepisano pravilo kaže da svako prvo mjerenje nekog eksperimenta rezultira neuspjehom, odnosno odstupa od očekivanih rezultata. Ponavljajući mjerenja, sve više se približavamo očekivanim vrijednostima, odnosno zakonitostima. Uzmemo li u obzir da naša osjetila, kao ni mjerni instrumenti kojima mjerimo nisu savršeni, realno je za očekivati kako ni jedno od naših mjerenja nije potpuno točno, odnosno podložno je pogrešci.

Svako fizikalno mjerenje podložno je mnogobrojnim vanjskim utjecajima. Uzimajući u obzir fizička ograničenja kao što je oštrina ljudskog oka, ali i ograničenja i nesavršenost mjernih instrumenata, jasno je kako se pojedinačni rezultati mjerenja neće potpuno podudarati. Upravo zbog toga se svakoj fizikalnoj veličini pridaje pogreška. Zbog navedenih ograničenja, mjerenja određene fizikalne veličine rade uzastopno, odnosno više se puta mjeri ista fizikalna veličina, a svaki se finalni rezultat mjerenja iskazuje s mjerom točnosti, tj. s pogreškom.

Pogrešku je definirao Webster kao razliku između izračunate vrijednosti ili promatrane vrijednosti u odnosu na pravu vrijednost.

Ako uzmemo da je A_n vrijednost pojedinog mjerenja, A_s prava vrijednost mjerene fizikalne veličine, tada rezultat pojedinog mjerenja A_n odstupa od prave vrijednosti A_s za određeni delta. Takvo odstupanje nazivamo pogreškom ΔA$_n$ i prikazujemo

$$\Delta A_n = A_s - A_n \tag{3.1}$$

3.1.2 Točnost, preciznost

Točnost, preciznost i pouzdanost smo spomenuli kao jako bitne elemente svakog mjerenja. Razlika između točnosti i preciznosti zna biti poprilično poprilično zbunjujuća. Točnost definiramo kao odstupanje rezultata mjerenja od prave vrijednosti mjerene fizikalne veličine. U nekim je situacijama teško odrediti pravu vrijednost, odnosno točnost pojedinog mjerenja, ali statističkim metodama možemo odrediti interval u kojem se prava vrijednost najvjerojatnije nalazi. Kada govorimo o preciznosti, razlikujemo preciznost uređaja i preciznost mjerenja.

Govorimo li o preciznosti instrumenta, tj. mjernog uređaja, ona je najčešće navedena na samom instrumentu. Međutim, nužno je procijeniti preciznost koja je određena podjelom skale na instrumentu. Na primjer, ako je najmanji podjeljak skale metra 1 mm, preciznost metra je 0,5 mm. Preciznost mjerenja govori o prosječnom rasipanju rezultata. Ako se ponavljanjem mjerenja uvijek dobije ista vrijednosti mjerene veličine, za preciznost mjerenja uzimamo preciznost instrumenta.

3.1.3 Pogreška

Ranije je bilo riječi o pogreškama. Tri su vrste pogrešaka: sistematske, grube i slučajne.

- Sistematske pogreške nije jednostavno uočiti, uzrokovane su greškom u sustavu mjerenja, npr. lošim baždarenjem uređaja. One se ponavljaju i javljaju u istom smjeru i iznosu. Dobar primjer su pomaknuti nulti položaj instrumenata ili pogrešno baždarene skale instrumenata. Često ih možemo ukloniti ili smanjiti

tijekom samih mjerenja jednostavnom provjerom ili zamjenom instrumenta.

- Grube pogreške, kao što im ime kaže, mogu nastati nekim grubim, naglim poremećajem u okolini, u mjernom uređaju ili ljudskim propustom. Pogreške nastale naglim poremećajem u okolini ili uređaju relativno jednostavno uočavamo jer jako odstupaju od ostalih rezltata mjerenja. Potrebno ih je izuzeti iz daljnje analize. Netočno očitavanje mjerne skale ili pogrešno upisivanje iznosa mjerne velične neki su od najboljih primjera.
- Slučajne pogreške mogu se smanjivati, ali se ne daju potpuno izbjeći. Problem smanjenja slučajnih pogrešaka jedan je od najvažnijih elemenata provedbe uspješnih mjerenja. Uzrok je različit, načešće je riječ o nestalnosti okoline i mjernog uređaja, a unutar jedne serije mjerenja razlikuju se po smjeru i iznosu. Kvalitetnijom izolacijom od okoline i savršenijim uređajem možemo ih smanjivati, no bez obzira na to radimo li manje ili više savršenim uređajem i njegovim okružjem, nećemo ih uspjeti potpuno izbjeći.

3.2 Obrada rezultata mjerenja

3.2.1 Osnovni statistički pojmovi, srednja vrijednost i standardna devijacija

Računom srednjih vrijednosti dolazimo do informacija o vrijednostima oko kojih se grupiraju vrijednosti mjerenja, odnosno elementi statističkog niza. Dijelimo ih na one ovisne o položaju kao što je medijan i potpune srednje vrijednosti kao što je aritmetička sredina.

Medijan je vrijednost koja statistički niz dijeli na dva jednaka dijela, odnosno u određenom nizu podataka 50 % elemenata ima vrijednost manju ili jednaku medijanu i 50 % elemenata ima vrijednost veću ili jednaku medijanu.

Aritmetička sredina je omjer zbroja svih vrijednosti mjerenja, odnosno jednog niza i broja elemenata tog niza. Matematički je definiramo:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{3.2}$$

Varijanca je prosječno kvadratno odstupanje vrijednosti od aritmetičke sredine. Poput srednje vrijednosti spada u potpune vrijednosti, zato što obuhvaća sve elemente mjerenja, odnosno nekog numeričkog statističkog niza. Govori o mjeri odstupanja, odnosno raspršenosti elemenata skupa od aritmetičke sredine. Matematički je definiramo:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{3.3}$$

Standardna devijacija je pozitivan korijen iz varijance, definiramo je kao prosječno odstupanje od aritmetičke sredine. Pomoću standardne devijacije možemo uspoređivati raspršenost oko aritmetičke sredine nizova koji su grupirani po jednakom obilježju. Matematički je definiramo:

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3.4}$$

3.2.2 Distribucije

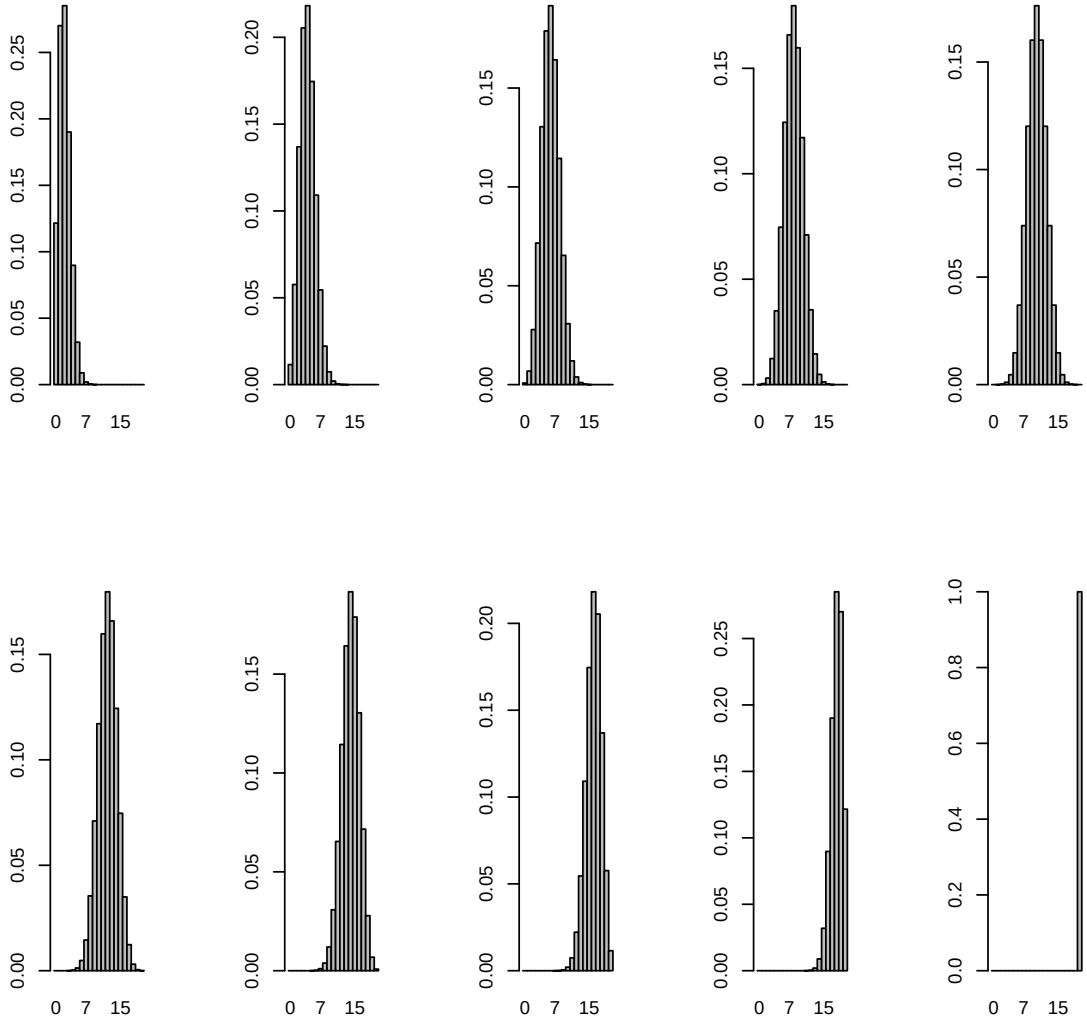
Statistička analiza rezultata mjerenja obično počinje provjerom oblika distribucije. U statistici označava raspodjelu rezultata, odnosno frekvenciju kojom se u nekom skupu rezultata, poredanih po veličini pojavljuju pojedini rezultati. Oblik distribucije ukazuje na model uz koji pristaju dobiveni rezultati, što uvelike pomaže u interpretaciji rezultata i utječe na način statističke obrade rezultata mjerenja.

Ovisno o vrsti varijable razlikujemo distribucije za diskretnu varijablu primjer koje je Binomna distribucija te distribucije za kontinuiranu varijablu primjer koje je Gaussova distribucija.

Binomna distribucija

Binomna distribucija pokazuje vjerojatnost događanja međusobno isključivih događaja. Za primjer uzmimo bacanje novčića: pismo i glava međusobno su isključivi događaji i vjerojatnost pojavljivanja je jednaka.

```
par(mfrow=c(2, 5))
for(p in seq(0.1, 1, len=10))
{
    x <- dbinom(0:20, size=20, p=p)
    barplot(x, names.arg=0:20, space=0)
}
```

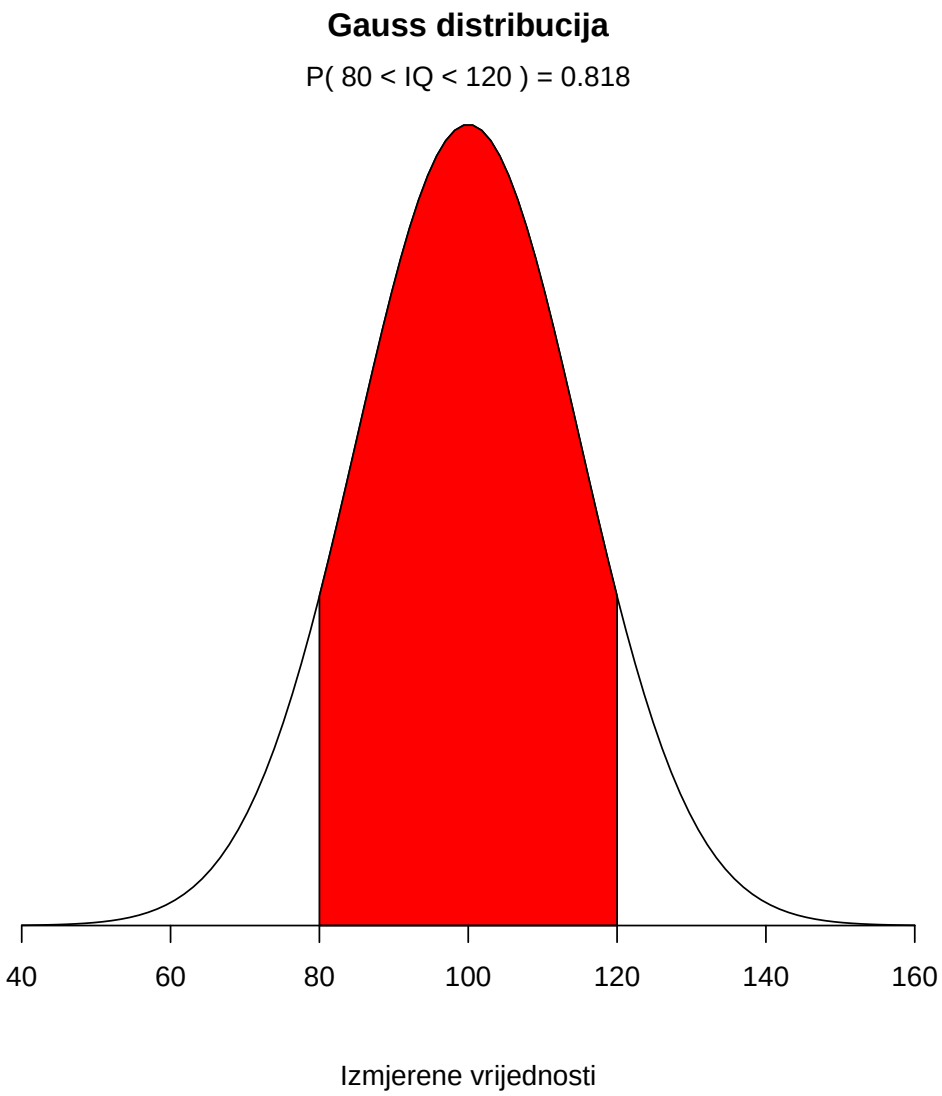


Gaussova distribucija

Gaussova distribucija je najvažnija distribucija u statistici, nazivamo ju još i normalnom distibucijom. Radi se o potpuno simetričnoj distribuciji zvonolikog oblika, koja se asimptomatski približava osi apscisa. Uzet je primjer iz psihologije koji se odnosi na razdiobu kvocijenta inteligencije. Izračunajmo koliki postotak ljudi ima kvocijent inteligencije između 80 i 120.

```
mean=100; sd=15
lb=80; ub=120

x <- seq(-4,4,length=100)*sd + mean
hx <- dnorm(x,mean,sd)

plot(x, hx, type="n", xlab="Izmjerene vrijednosti", ylab="",
```

```
  main="Gauss distribucija", axes=FALSE)

i <- x >= lb & x <= ub
lines(x, hx)
polygon(c(lb,x[i],ub), c(0,hx[i],0), col="red")

area <- pnorm(ub, mean, sd) - pnorm(lb, mean, sd)
result <- paste("P(",lb,"< IQ <",ub,") =",
   signif(area, digits=3))
mtext(result,3)
axis(1, at=seq(40, 160, 20), pos=0)
```



3.3 Ovisna i neovisna mjerenja

Razlikujemo ovisna i neovisna mjerenja. Glavna karakteristika neovisnih mjerenja jest da ono sadrži samo slučajne pogreške i nije ovisno o nekoj drugoj veličini. Drugim riječima, jednoj vrijednosti jedne pojave odgovara više različitih vrijednosti druge pojave. Glavna karakteristika ovisnih mjerenja jest da je tražena veličina funkcija neposredno izmjerenih vrijednosti, odnosno svakoj vrijednosti jedne pojave odgovara točno određena vrijednost druge pojave.

3.3.1 Neovisna mjerenja

Promatramo li određenu fizikalnu veličinu i izvodimo li niz mjerenja, dobit ćemo rezultate od x_1 do x_n. Rezultati nisu međusobno jednaki, već se razlikuju zbog prisustva slučajnih pogrešaka koja se ne mogu izbjeći, već samo smanjiti. Kako izvodimo mjerenja, uzorak mjerenja se ponavlja. Želimo li odrediti najvjerojatniju vrijednost neke mjerene veličine i pogrešku mjerenja iste, ponovit ćemo stare i definirati neke nove pojmove.

Srednju vrijednost računamo kao artimetičku sredinu izmjerenih vrijednosti, n je broj mjerenja dok $\bar{x}$ nazivamo očekivanom vrijednosti, odnosno najvjerojatnijom pravom vrijednosti mjerene fizikalne veličine.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.5}$$

Srednju kvadratnu pogrešku pojedinog mjerenja nazivamo još i standardnom devijacijom, a govori o nepreciznosti mjerenja, odnosno o kvadratnom odstupanju izmjerenih vrijednosti od srednje vrijednosti.

$$\Sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3.6}$$

Srednja kvadratna pogreška aritmetičke sredine M_n govori o nepouzdanosti mjerenja, matematički je definiramo:

$$M_n = \sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3.7}$$

Relativna nepouzdanost R_n je omjer srednje kvadratne pogreške aritmetičke sredine i srednje vrijednosti promatrane veličine. Matematički je definiramo:

$$R_n = \frac{M_n}{\bar{x}} \cdot 100\% \tag{3.8}$$

Maksimalna apsolutna pogreška ΔX jest najveće odstupanje pojedinačnog mjerenja od srednje vrijednosti. Matematički je definiramo:

$$\Delta X = |\bar{X} - x_i|_{max} \tag{3.9}$$

3.3.2 Ovisna mjerenja

Za ovisna mjerenja smo kazali kako je tražena veličina funkcija neposredno izmjerenih vrijednosti. To u pravilu znači da je neka tražena veličina G funkcija više neposredno izmjerenih veličina $G = f(y_1, ...y_n)$, od kojih svaka ima neku određenu pogrešku Δy. Želimo li odrediti najvjerojatniju vrijednost mjerene veličine i pogrešku mjerenja, definiramo sljedeće:

- najvjerojatniju vrijednost fizikalne veličine, odnosno srednju vrijednost te veličine:

$$\bar{G} = f(\bar{y}_1, ...\bar{y}_n) \tag{3.10}$$

- maksimalnu apsolutnu pogrešku:

$$\Delta G = \sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} \Delta \bar{y}_i \right) \tag{3.11}$$

- srednju kvadratnu pogrešku:

$$\Delta M_G = \sqrt{\sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} \Delta \bar{y}_i \right)} \tag{3.12}$$

3.4 Prikaz, zaokruživanje mjerenja i finalnih rezultata

Preciznost mjerenja definirana je načinom na koji su izmjereni rezultati pikazani. Postoji nekoliko pravila za ispravno zapisivanje mjerenja, a ona su:

- krajnje lijeva znamenka različita od nule je najznačajnija znamenka;
- ako ne postoji decimalna točka, najdesnija znamenka različita od nule je najneznačajnija znamenka;
- ako postoji decimalna točka, krajnje desna znamenka je najneznačajnija znamenka, čak ako je jednaka nuli;
- sve znamenke između najznačajnije znamenke i najneznačajnije znamenke se tretiraju kao značajne znamenke.

Pravilno zaokruživanje je važno jer može bitno utjecati na završni rezultat. Rezultat iskazujemo na sljedeći način:

$$x = (\bar{x} \pm M_n)\,\mathrm{mjerna\,jedinica} \tag{3.13}$$

Pogreška se zaokružuje u skladu s prije navednim pravilima za zapis mjerenja, odnosno zaokružuje se na prvu znamenku različitu od nule. Rezultat mjerenja zaokružuje se na prvu nepouzdanu znamenku, odnosno na onu koja se nalazi na istom decimalnom mjestu kao i zaokružena pogreška.

3.5 Linearna regresija i nelinearna regresija

3.5.1 Regresija

Regresija se bavi opisivanjem ovisnosti jedne varijable o jednoj ili više drugih varijabli, a model kojim se izražava statistička povezanost između zavisne varijable i odabranog skupa nezavisnih varijabli je regresijski model. Regresijska analiza se koristi za donošenje zaključaka o nizu slučajnih varijabli $Y_1, \ldots, Y_n$ koje ovise o nezavisnoj varijabli x.

Zaključci se donose na temelju niza sparenih mjerenja (x_1, y_1),...,(x_n, y_n), gdje su $x_1, \ldots, x_n$ vrijednosti nezavisne varijable x, a $y_1, \ldots, y_n$ odgovarajuće vrijednosti slučajnih varijabli $y_1, \ldots, y_n$.

3.5.2 Linearna Regresija

Linearna regresija govori o statističkoj linearnoj vezi između zavisne varijable y i nezavisne varijable x, koju možemo opisati izrazom:

$$y = ax + b \tag{3.14}$$

u kojem je a nagib pravca, a b odsječak na osi y.

Metoda najmanjih kvadrata

Problem procjene nepoznatih parametara a i b možemo identificirati s procjenom nepoznatog regresijskog pravca pa takvu procjenu $\hat{a}$ i $\hat{b}$ nazivamo metoda najmanjih kvadrata. Metoda najmanjih kvadrata (engl. Least Square method; LS-metoda) sastoji se od procjene nepoznatih parametara koji minimiziraju sumu kvadrata odstupanja empirijskih vrijednosti zavisne varijable i y od procijenjenih ili regresijskih vrijednosti. Dakle, temelji se na uvjetu da zbroj kvadrata vertikalnih odstupanja točaka u dijagramu raspršenja od traženog pravca regresije bude minimalan. Osnovni problem koji rješnava ova metoda je kako iz dobivenih eksperimentalnih podataka dobiti funkcionalnu ovisnost. Parametar a zovemo regresijskim koeficijentom, a pokazuje koliko se u prosjeku mijenja zavisna varijabla, ako se nezavisna varijabla promijeni za jedan. Parametar b je konstanta i pokazuje vrijednost zavisne varijable kad je nezavisna jedna nuli. Matematički ih definiramo:

$$a = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2} \tag{3.15}$$

$$b = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{n\sum x_i^2 - (\sum x_i)^2} \tag{3.16}$$

Njihove pogreške su:

$$M_a = \sqrt{\frac{1}{n-2}\left[\frac{n\sum y_i^2 - (\sum y_i)^2}{n\sum x_i^2 - (\sum x_i)^2} - a^2\right]} \tag{3.17}$$

$$M_b = M_a\sqrt{\frac{1}{n}\sum x_i^2} \tag{3.18}$$

3.5.3 Nelinearna regresija

Nelinearna regresija pokazuje statističku nelinearnu vezu između zavisne varijable y i nezavisne varijable x. Dakle, funkcionalan odnos između zavisne varijable y i nezavisne varijable x ne može biti aproksimiran linearnom regresijom. U većini slučajeva, neleinearnu regresiju, odnosno nelinerne zavisnosti varijabli nastoje se linearizirati i provesti linearna regresija. Takav linearizirani nelinearni model nazivamo modelom višestruke linearne regresije. No, ukoliko model nije moguće linearizirati, nazivamo ga pravim linearnim modelom.

Primjer nelinearne regresije je eksponencijalni regresijski model. Eksponencijalni regresijski model je nelinearan u parametrima i ima oblik:

$$y = ab^x \tag{3.19}$$

Pomoću logaritamske transformacije model postaje linearan u parametrima (iako nelinearan u varijablama), a nezavisna varijabla x ostaje u nepromijenjenom obliku. Vrijednosti parametara se nakon linearizacije računaju kao i kod linearne regresije.

4 Obrada rezultata mjerenja

4.1 Matematičko njihalo

4.1.1 Uvod u vježbu

Matematičkim njihalom nazivamo sitno tijelo ili materijalnu točku mase m, koja se njiše obješena o donji kraj nerastezljive, lagane nit duljine l. Masu niti zanemarujemo, a učvršćena je na gornjem kraju. U eksperimentu koji izvodimo masa nije koncentrirana u samo jednoj točki, već je raspodijeljena po cijelom volumenu kugle. Uvrsti li se uvjet da je promjer kugle mnogo manji od duljine niti, problem se može svesti na razmatranje matematičkog njihala. Pomakne li se kuglica iz položaja ravnoteže, započinje titranje oko ravnotežnog položaja. Njihalo ćemo promatrati pomoću zakona očuvanja energije. Zakon očuvanja energije kaže:

$$E_{uk} = E_k + E_p = konst. \tag{4.20}$$

Potencijalna energija kuglice raste, dok se kinetička energija smanjuje. Vrijedi i obrnuto: kinetička energija raste, dok se potencijalna energija smanjuje. Tako je energija sustava očuvana. Kinetička E_k i potencijalna energija E_p su definirane:

$$E_k = \frac{mv^2}{2} = \frac{1}{2}ml^2\left(\frac{\partial\phi}{\partial t}\right)^2 \tag{4.21}$$

$$E_p = mgl(1-\cos\phi) \tag{4.22}$$

Ukupnu energiju sustava možemo naći uz uvijet da u položaju maksimalnog otklona vrijedi ϕ=α

$$E_{uk} = E_{p_{max}} = mgl(1-cos(\alpha)) \tag{4.23}$$

Period titranja je definiran kao:

$$T = 2\pi\sqrt{\frac{l}{g}} \tag{4.24}$$

On se razlikuje za malene i velike kutove otklona. Za malene kutove je definiran kao:

$$T = 2\pi\sqrt{\frac{l}{g}}\int_{-\alpha}^{\alpha}\frac{d\phi}{\sqrt{\alpha^2-\phi^2}} = 2\pi\sqrt{\frac{l}{g}}, \tag{4.25}$$

a za velike kuteve otklona kao:

$$T = 2\pi\sqrt{\frac{l}{g}}\left(1+\frac{1}{4}\sin^2\left(\frac{\alpha}{2}\right)+\ldots\right) \tag{4.26}$$

Slika 4.5: Prikaz eksperimentalnog postava vježbe Matematičko njihalo

4.1.2 Mjerenja

U ovoj je vježbi zadatak izmjeriti period titranja za male kutove otklona za desetak različitih duljina niti te prikazati rezultate u T^2 - l dijagramu. Također, potrebno je metodom najmanjih kvadrata odrediti konstantu gravitacije g. Drugi zadatak je izmjeriti period titranja za pet različitih kutova otklona i prikazati grafički ovisnost o $\sin \alpha^2/2$.

Eksperimentalni postav se sastoji od njihala, metalnog stalka sa postoljem, i zaporne ure sa senzorom. Prikazan je na slici br. 4.5 i br. 4.6.

Prije početka mjerenja, nužno je postaviti eksperimentalni postav i pripremiti aparaturu za početak mjerenja. Nakon što smo postavili elektroničku zapornu uru, potrebno je inicijalizirati fotoćeliju pritiskom na tipku `set` na pozadini fotoćelije. Kada kugla izmaknuta iz položaja ravnoteže presiječe svjetlosni snop, detektor šalje signal na elektroničku zapornu uru. Okidač zaporne ure namjestimo tako da s primitkom prvog signala počne mjeriti vrijeme, dok se s primitkom drugog signala zaustavlja. Na taj način mjeri se poluperiod titranja.

Prvi dio mjerenja izvodimo tako da nit s kuglom učvrstimo u objesište i izmjerimo udaljenost od objesišta do kugle. Otklonimo kuglu za neki mali kut (nekoliko stupnjeva) i očitavamo poluperiod jednak broj puta s obiju strana detektora ne prekidajući uspostavljeno njihanje. Isti postupak ponovimo za deset raznih duljina niti.

Slika 4.6: Prikaz matematičkog njihala i senzora

Br. mjerenja	α [°]	l [cm]	$T_1/2[ms]$	$T_2/2[ms]$	$T_3/2[ms]$	$T_4/2[ms]$	$T_5/2[ms]$
1	15	65	817.5	819.6	815.9	819.4	819.1
2	15	60	736.7	735.6	730.5	736.5	735
3	15	55	773.6	775.5	775.5	778.3	771.4
4	15	50	694.1	691.5	693	692.4	693.4
5	15	45	693.6	693.5	693.7	694.1	694.5
6	15	40	629.1	625.9	627.5	629.7	632
7	15	35	578.3	579.8	582.1	579.3	580
8	15	30	563.4	563.5	562.5	563	562.9
9	15	25	493.1	495.9	499.4	500.3	494.4
10	15	20	435	444.5	441.2	448.7	441.3

Tablica br.1 Poluperiod titranja matematičkog njihala kuta otklona $\alpha = 15°$ za različite duljine niti

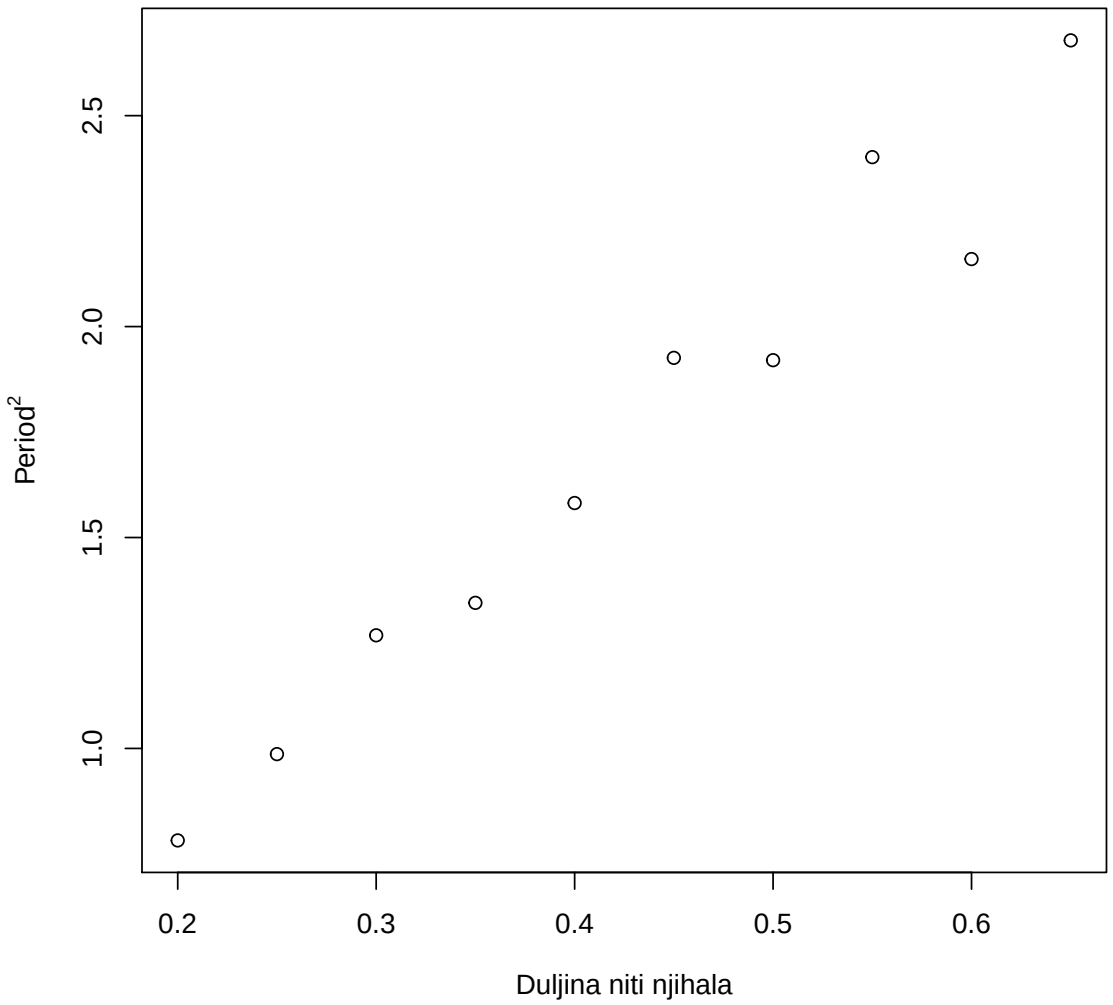
```
#Računamo srednju vrijednost perioda za različite duljine niti
Tsrednje65=mean(c(817.5,819.6,815.9,819.4,819.1)*2)/1000
Tsrednje60=mean(c(736.7,735.6,730.5,736.5,735)*2)/1000
Tsrednje55=mean(c(773.6,775.5,775.5,778.3,771.4)*2)/1000
Tsrednje50=mean(c(694.1,691.5,693,692.4,693.4)*2)/1000
Tsrednje45=mean(c(693.6,693.5,693.7,694.1,694.5)*2)/1000
Tsrednje40=mean(c(629.1,625.9,627.5,629.7,632)*2)/1000
Tsrednje35=mean(c(578.3,579.8,582.1,579.3,580)*2)/1000
Tsrednje30=mean(c(563.4,563.5,562.5,563,562.9)*2)/1000
```

```
Tsrednje25=mean(c(493.1,495.9,499.4,500.3,494.4)*2)/1000
Tsrednje20=mean(c(435,444.5,441.2,448.7,441.3)*2)/1000
Broj_mjerenja<-c(1,2,3,4,5,6,7,8,9,10)
Kut_otklona<-c(15,15,15,15,15,15,15,15,15,15)
Duljina_niti<-c(0.65,0.60,0.55,0.50,0.45,0.40,0.35,0.30,0.25,0.20)
Period<-c(Tsrednje65,Tsrednje60,Tsrednje55,Tsrednje50,Tsrednje45,
          Tsrednje40,Tsrednje35,Tsrednje30,Tsrednje25,Tsrednje20)
Tablica=data.frame(Broj_mjerenja,Kut_otklona,Duljina_niti,Period)
Tablica
```

```
##    Broj_mjerenja Kut_otklona Duljina_niti  Period
## 1              1          15         0.65 1.63660
## 2              2          15         0.60 1.46972
## 3              3          15         0.55 1.54972
## 4              4          15         0.50 1.38576
## 5              5          15         0.45 1.38776
## 6              6          15         0.40 1.25768
## 7              7          15         0.35 1.15980
## 8              8          15         0.30 1.12612
## 9              9          15         0.25 0.99324
## 10            10          15         0.20 0.88428
```

Nakon što smo izračunali srednju vrijednost perioda, formirali smo tablicu mjerenja koristeći naredbu `data.frame()` gdje su prikazani redni broj mjerenja, duljina niti, kut otklona i srednja vrijednost perioda za 10 različitih duljina niti. Navedene podatke možemo grafički prikazati koristići fukncju `plot()`.

```
#Grafički prikazujemo ovisnost varijabli
plot(Duljina_niti,Period^2, xlab=TeX("Duljina niti njihala"),
     ylab=TeX("Period^2"))
```

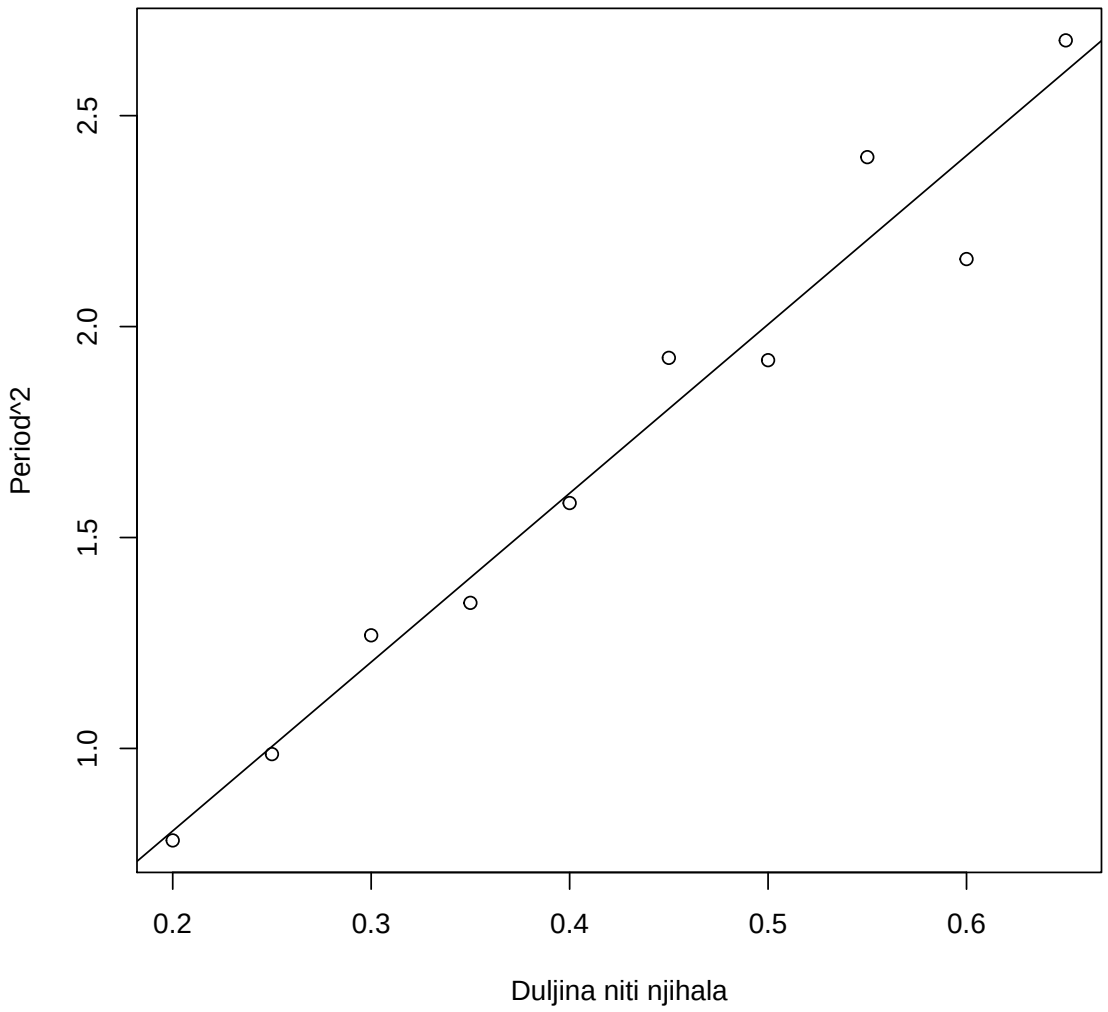


Možemo uočiti kako su izmjerene vrijednosti u $T^2 - l$ dijagramu linearne. Prema tome, može se izračunati vrijednost konstante a koristeći metodu najmanjih kvadrata:

```
#Računamo linearnu regresiju
Linearna<-lm(Period^2~Duljina_niti)
Linearna

##
## Call:
## lm(formula = Period^2 ~ Duljina_niti)
##
## Coefficients:
##  (Intercept)  Duljina_niti
##      0.00393       4.00249

summary(Linearna)
```

```
##
## Call:
## lm(formula = Period^2 ~ Duljina_niti)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.24535 -0.05054 -0.02025  0.07055  0.19633
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.00393    0.12823   0.031    0.976
## Duljina_niti  4.00249    0.28584  14.003 6.56e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1298 on 8 degrees of freedom
## Multiple R-squared:  0.9608,Adjusted R-squared:  0.9559
## F-statistic: 196.1 on 1 and 8 DF,  p-value: 6.561e-07
```

```
#Grafički prikazujemo ovisnost varijabli
#s ucrtanim pravcem linearne regresije
plot(Duljina_niti, Period^2, xlab="Duljina niti njihala",
     ylab="Period^2")
abline(Linearna)
```



Metodom najmanjih kvadrata izračunali smo vrijednost koeficijenta pravca a i on iznosi:

$$a = (4,0 \pm 0,3) \tag{4.27}$$

uz već spomenutu linearnu ovisnost

$$y = ax + b \tag{4.28}$$

iz jednadžbe

$$T = 2\pi\sqrt{\frac{l}{g}} \tag{4.29}$$

možemo izračunati vrijednost g uz nekoliko tranformacija tako da ranije navednu jednažbu kvadriramo pa dobivamo sljedeće

$$T^2 = 4\pi^2\frac{l}{g} \tag{4.30}$$

ili preciznije

$$T^2 = 4\pi^2 \frac{l}{g} + b \tag{4.31}$$

dakle a možemo zapisati kao

$$a = 4\pi^2 \frac{1}{g} \tag{4.32}$$

sada jednadžbu podijelimo s g i pomnožimo s a i dobiti ćemo izraz za g

$$g = 4\pi^2 \frac{1}{a} \tag{4.33}$$

gdje jednostavnim uvrštavanjem možemo doći do vrijednosti g, i ona iznosi

```
#Računamo g
g=4*pi^2*(1/4.0)
g

## [1] 9.869604
```

Pogrešku računamo pomoću izraza

$$\Delta M_g = \sqrt{\sum_{i=1}^{n} \left(\frac{\partial g}{\partial a} M_a \right)^2} \tag{4.34}$$

odnosno:

$$\Delta M_g = \sqrt{\left(\frac{\partial}{\partial a} (4\pi a^{-1}) M_a \right)^2} = \left(\frac{2\pi}{a} \right)^2 M_a \tag{4.35}$$

```
#Računamo grešku od g
aaa<-4.0
Maaa<-0.3
Mbbb<-(((2*pi)/aaa)^2)*Maaa
Mbbb

## [1] 0.7402203
```

sada konstantu g možemo zapisati kao:

$$g = (9,9 \pm 0,7)\,\mathrm{m^2/s} \tag{4.36}$$

U drugom dijelu mjerenja proučavamo ovisnost perioda titranja o kutu otklona. Eksperimentalni postav je jednak kao u prvom dijelu vježbe. Nit s kuglom učvrstimo u objesište, a duljinu niti više ne mijenjamo tijekom mjerenja. Kutovi otklona mjere se pomoću kutomjera učvršćenog na stalak kod objesišta.

Br. mjerenja	α [°]	l [cm]	$T_1[ms]$	$T_2[ms]$	$T_3[ms]$	$T_4[ms]$	$T_5[ms]$
1	15	35	578.3	579.8	582.1	579.3	580
2	20	35	587.8	589.4	594.4	593.3	592.6
3	25	35	598.1	601.5	599.7	600.2	601.1
4	30	35	601.6	600.5	601.5	600.3	600.9
5	35	35	605.3	608.1	609.4	608.1	607.5

Tablica br.2 Period titranja matematičkog njihala duljine niti 35 cm za različite kutove otklona

```
#Računamo srednju vrijednost perioda za različite kuteve otklona
TTsrednje15=mean(c(581.3,582.8,583.1,579.3,580)*2)/1000
TTsrednje20=mean(c(591.8,589.4,594.4,593.3,592.6)*2)/1000
TTsrednje25=mean(c(599.1,601.5,599.7,600.2,601.1)*2)/1000
TTsrednje30=mean(c(601.6,600.5,601.5,600.3,600.9)*2)/1000
TTsrednje35=mean(c(604.3,603.1,604.4,605.1,602.5)*2)/1000
Broj_mjerenja2<-c(1,2,3,4,5)
Kut_otklona2<-c(15,20,25,30,35)
Duljina_niti2<-c(0.35,0.35,0.35,0.35,0.35)
Period2<-c(TTsrednje15,TTsrednje20,TTsrednje25,TTsrednje30,TTsrednje35)
Tablica2=data.frame(Broj_mjerenja2,Kut_otklona2,Duljina_niti2,Period2)
Tablica2

##   Broj_mjerenja2 Kut_otklona2 Duljina_niti2 Period2
## 1              1           15          0.35 1.16260
## 2              2           20          0.35 1.18460
## 3              3           25          0.35 1.20064
## 4              4           30          0.35 1.20192
## 5              5           35          0.35 1.20776
```

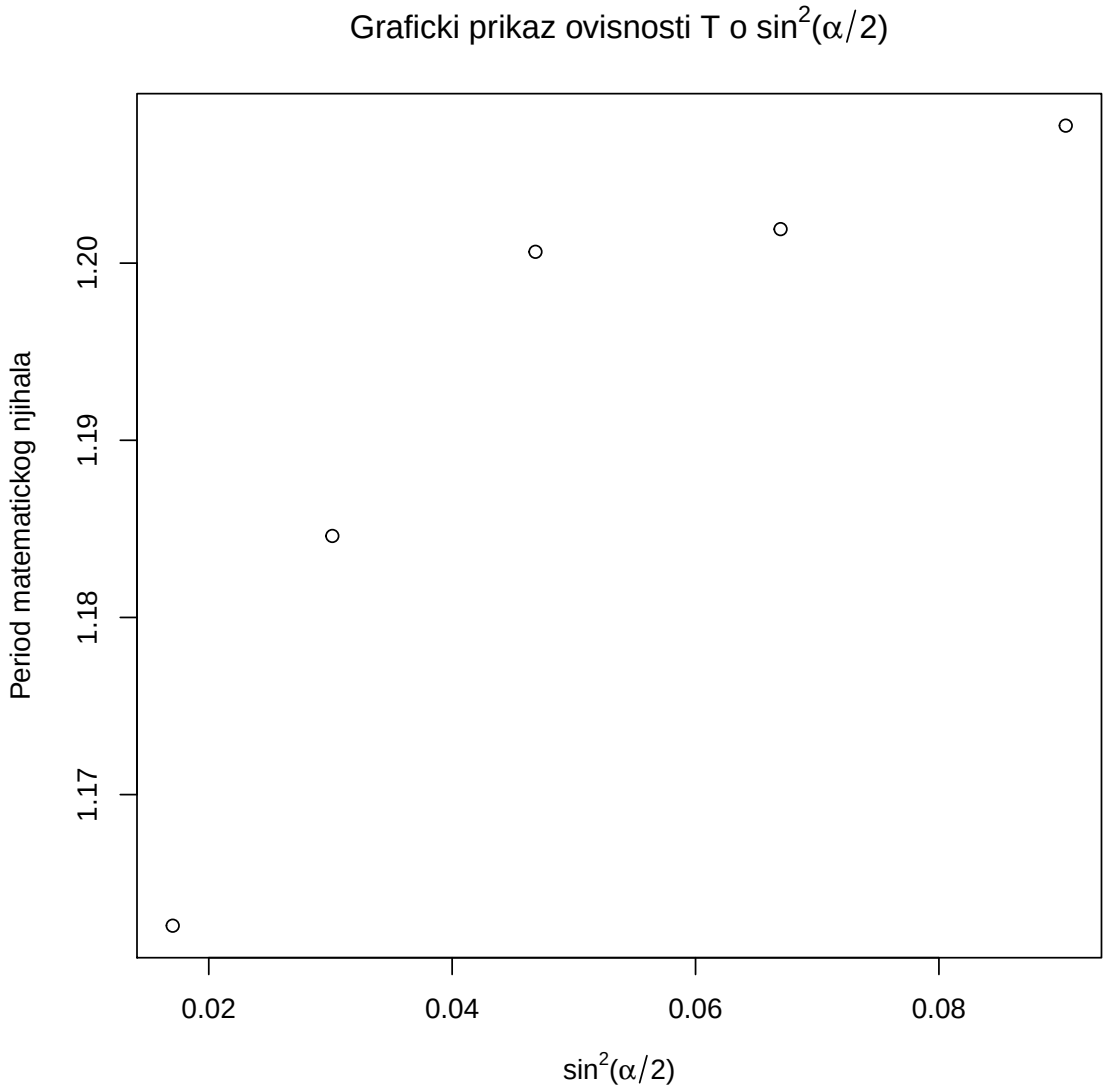
Nakon što smo izračunali srednju vrijednost perioda, formirali smo tablicu mjerenja koristeći naredbu `data.frame()` gdje su nam prikazani redni broj mjerenja, duljina niti, kut otklona i srednja vrijednost perioda za pet različitih kutova otklona. Spomenuli smo ranije kako se stupnjevi u R-u računaju u radijanima, a nama je potrebna funkcija `sin()` te je potrebno pretvoriti stupnjeve u radijane, to ćemo odraditi pomoću vlastite pomoćne fukcije `deg2rad`:

```
#Pretvaramo stupnjeve u radijane
deg2rad<-function(deg)
{(deg*pi) / (180)}
Kut_otklona22<-deg2rad(Kut_otklona2)
Kut_otklona22
```

```
## [1] 0.2617994 0.3490659 0.4363323 0.5235988 0.6108652
```

Pretvorili smo stupnjeve u radije i pridružili vrijednosti vektoru `Kut_otklona22`. Prikažimo podatke grafički, zanima nas ovisnost o $\sin \alpha^2/2$

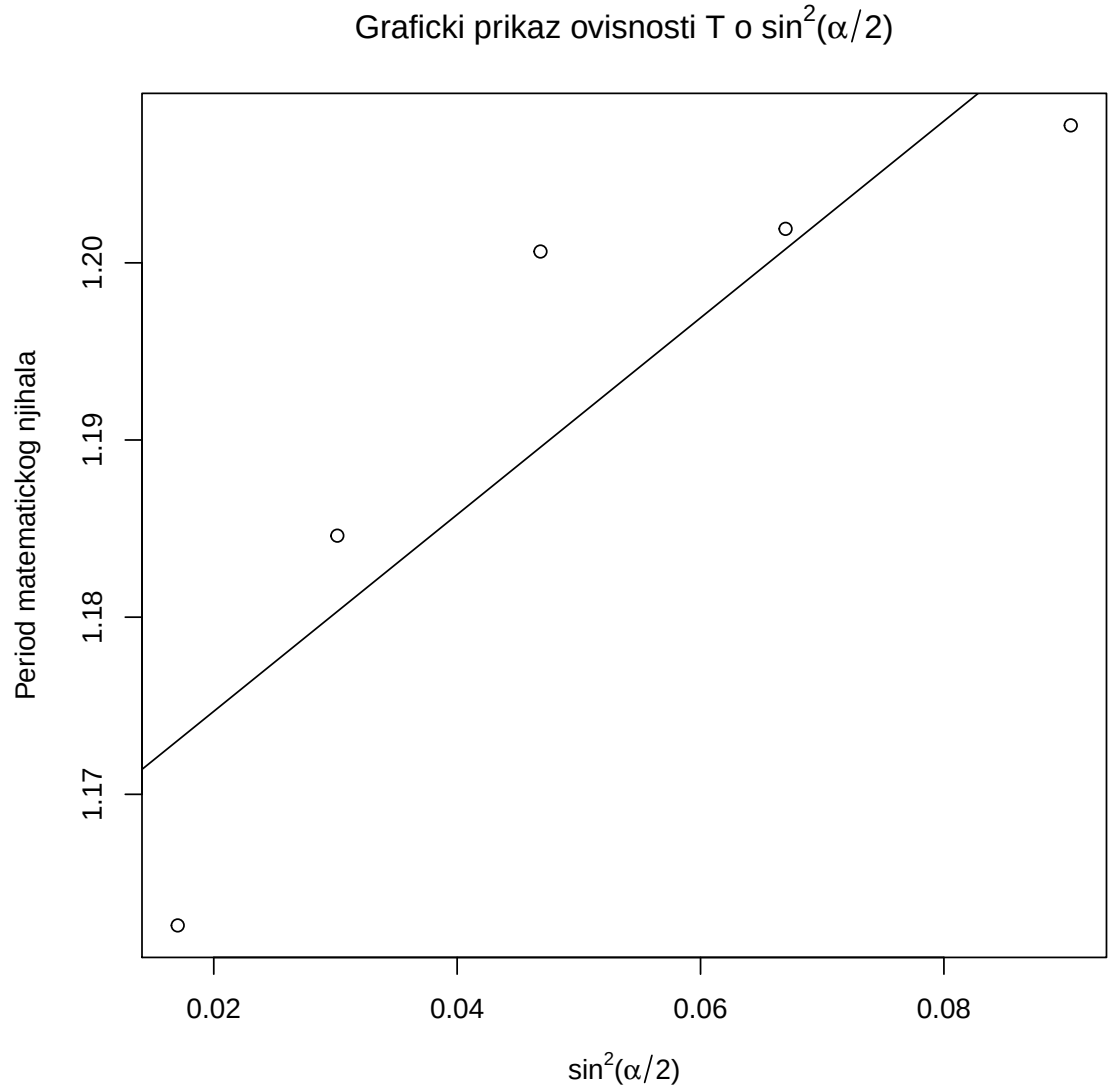
```
#Grafički prikazujemo ovisnost varijabli
Sinus=(sin((Kut_otklona22)/2))^2
plot(Sinus,Period2, ylab="Period matematičkog njihala",
     xlab=TeX('$\\sin^2(\\alpha/2)$'),
     main=TeX('Graficki prikaz ovisnosti T o $\\sin^2(\\alpha/2)$'))
```



Možemo uočiti kako su izmjerene vrijednosti u $T - \sin^2 \alpha/2$ dijagramu linearne. Prema tome, možemo izračunati vrijednost konstante a koristeći metodu najmnjih kvadrata:

```r
#Računamo linearnu regresiju
Linearna2<-lm(Period2~Sinus)
Linearna2

##
## Call:
## lm(formula = Period2 ~ Sinus)
##
## Coefficients:
## (Intercept)        Sinus
##      1.1636       0.5553

summary(Linearna2)

##
## Call:
## lm(formula = Period2 ~ Sinus)
##
## Residuals:
##         1         2         3         4         5
## -0.010439  0.004278  0.011048  0.001144 -0.006031
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.163578   0.009489 122.630  1.2e-06 ***
## Sinus       0.555302   0.167433   3.317   0.0452 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009781 on 3 degrees of freedom
## Multiple R-squared:  0.7857,Adjusted R-squared:  0.7143
## F-statistic:    11 on 1 and 3 DF,  p-value: 0.04517
```

```r
#Grafički prikazujemo ovisnost varijabli a pravcem linearne regresije
plot(Sinus,Period2, ylab="Period matematičkog njihala",
     xlab=TeX('$\\sin^2(\\alpha/2)$'),
     main=TeX('Graficki prikaz ovisnosti T o $\\sin^2(\\alpha/2)$'))
abline(Linearna2)
```



Metodom najmanjih kvadrata izračunali smo vrijednost koeficijenta pravca a i on iznosi:

$$a = (0,6 \pm 0,2) \tag{4.37}$$

4.1.3 Zaključak

Rezultati dobiveni mjerenjem uglavnom potvrđuju naša teorijska predviđanja. U prvom dijelu vježbe zadatak je bio pokazati kako se mijenja period titranja u ovisnosti o duljini njihala te izračunati vrijednost konstante gravitacije g. Pokazali smo kako su promatrane veličine u linearnoj ovisnosti i kako se period mijenja proporcionalno s duljinom njihala, a izračunata vrijednost konstante $g = (9,9 \pm 0,7)\,\mathrm{m}^2/\mathrm{s}$ je usporediva sa stvarnom vrijednosti gravitacijeske konstante g koja za područje grada Zagreba iznosi $g = 9,81\,\mathrm{m}^2/\mathrm{s}$.

Razlika vrijednosti, odnosno pogreška rezultat je nepreciznosti mjernih uređaja

(mjerili smo milisekunde) ali i nesavršenosti mjeritelja i mjernog postupka. U drugom dijelu vježbe zadatak je bio pokazati kako period titranja njihala ne ovisi o kutu otklona α. Grafički smo prikazali ovisnost i uočili kako period raste (i pada) proporcionalno s kutom α što je posljedica nesavršenosti mjerne opreme i mjernog postupka. Programski jezik smo koristili u svim dijelovima obrade podataka i pokazali dio njegovih mogućnosti u obliku jednostavnih operacija kao što su množenje, oduzimanje, grafičko prikazivanje i slično te složenijih kao što su linearna regresija i pisanje vlastitih funkcija.

4.2 Ogib svjetlosti

4.2.1 Uvod u vježbu

Obasjamo li maleni predmet ili usku pukotinu svjetlošću, uočit ćemo svjetlost i iza obasjanog predmeta, odnosno iza prepreke u slučaju pukotine, što je rezultat valnih svojstava svjetlosti. Kažemo da se svjetlost širi i iz zapreke, odnosno da se ogiba. Uzmimo primjer pukotine: snop svjetlosti koji dolazi s jedne strane interferira sa snopom svjetlosti koji dolazi s druge strane, a prema Huygensovom načelu strane pukotine predstavljaju dva koherentna izvora svjetlosti. Ogibna slika nastaje interferencijom beskonačno mnogo koherentnih valova. Intenzitet svjetlosti koja je prikazana na zastoru definirana je:

$$I(x) = I_0 \left(\frac{\sin(\alpha \sin\theta)}{\alpha \sin\theta} \right)^2 \tag{4.38}$$

gdje nam je $\alpha = \pi b/\lambda$, λ valna duljina svjetlosti, b širina pukotine i kut θ je kut upada na zastor i on je određen izrazom:

$$\sin\theta = \frac{x}{\sqrt{x^2 + a^2}} \tag{4.39}$$

gdje nam je x udaljenost između središta i ruba maksimuma, a a udaljenost između pukotine i zastora. Ogibni minimum zadovoljava sljedeći uvjet

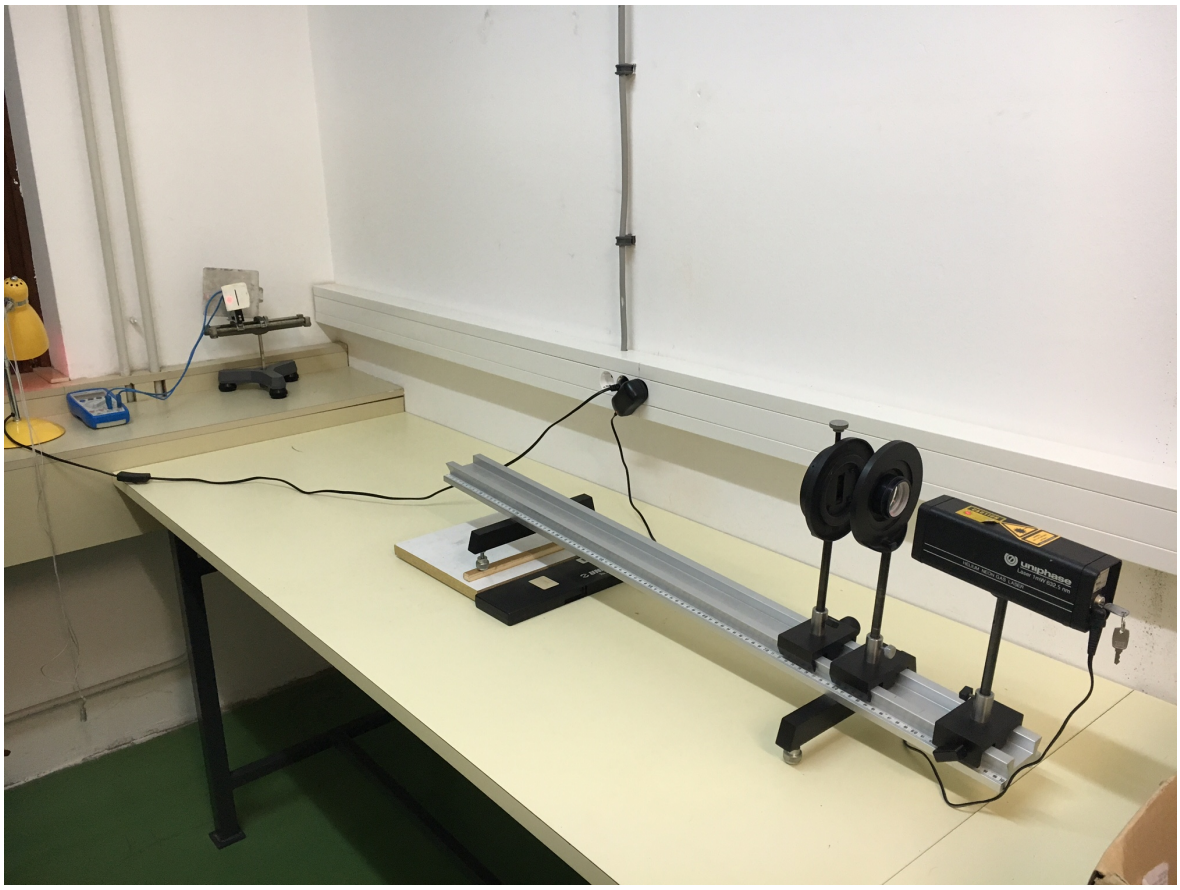
$$\frac{\pi b}{\lambda} \sin\theta^{min} = n\pi \tag{4.40}$$

dok su ogibni maksimumi zadovoljeni uvjetom:

$$\frac{\pi b}{\lambda} \sin\theta^{max} = \frac{2n+1}{2}\pi \tag{4.41}$$

Širinu pukotine možemo odrediti mjereći položaj n-tog ogibnog maksimuma relativno prema centralnom maksimumu :

$$b = \frac{(2n+1)\lambda}{2\sin\theta^{max}} \tag{4.42}$$



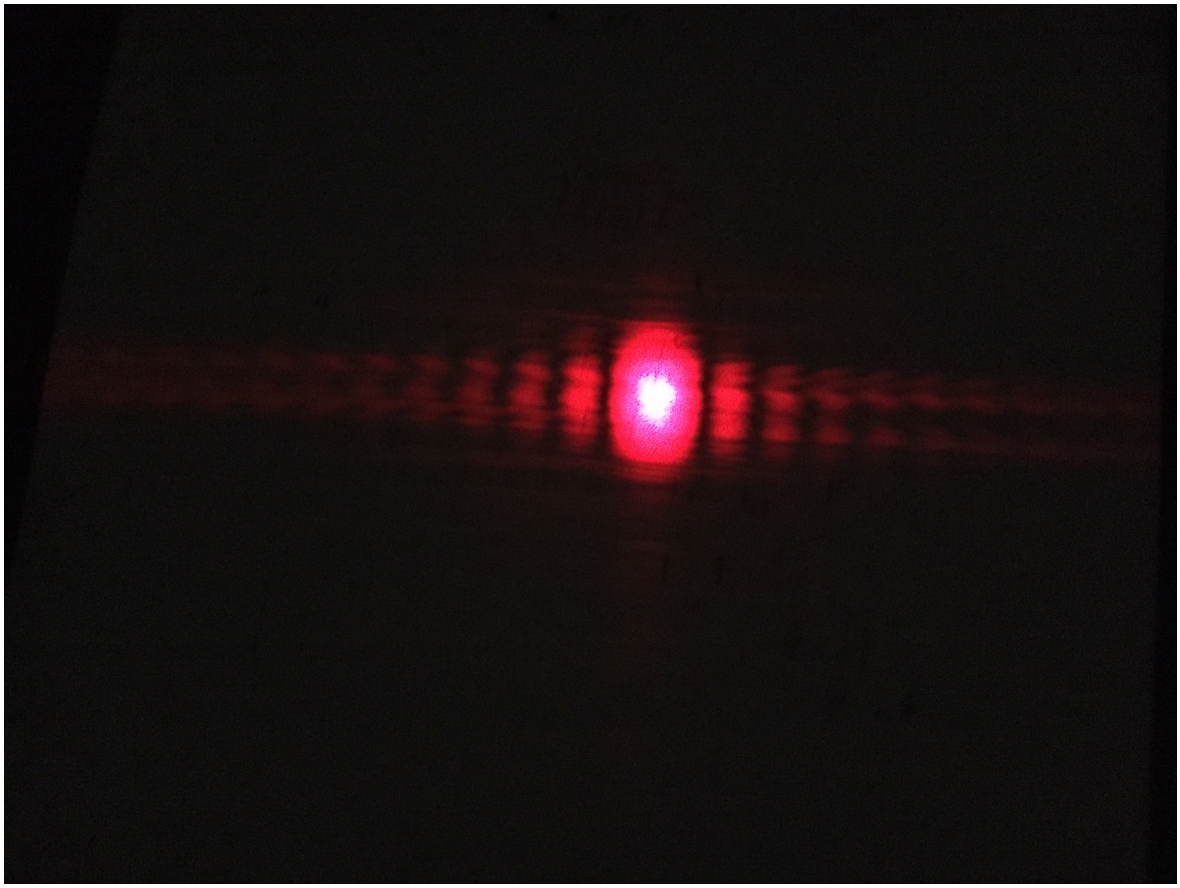
Slika 4.7: Prikaz eksperimentalnog postava vježbe Ogib svjetlosti

4.2.2 Mjerenja

Eksperimentalni postav nam se sastoji od lasera valne duljine λ = 632,8 nm, fotoćelije, digitalnog voltmetra, optičke klupe s milimetaskom skalom, pukotine i leće žarišne daljine $f = 5$ cm.

Divergentnu leću žarišne daljine $f = 5$ cm postavimo ispred lasera, ispred leće postavljamo pukotinu na udaljenost od 7,5 cm, zastor (fotoćeliju) postavljamo na udaljednost 1,5 do 2,5 metara. Raspodjelu intenziteta uzduž ogibne slike u smjeru okomitom na pukotinu mjerimo pomoću fotoćelije, koja se može pomicati pomoću vijka po optičkoj klupici s milimetarskom skalom. Napon mjerimo pomoću digitalnog voltmetra. Eksperimentalni postav i ogibna slika prikazani su na slikama 4.7 i 4.8.

Zadatak je pronaći raspodjelu intenziteta svjetlosti uzduž ogibne slike nastale na pukotini te nacrtati ovisnost intenziteta o udaljenosti od centralnog maksimuma $I/I_0 - x$. Valna duljina svjetlosti je λ = 632,8 nm. Također, potrebno je izračunati širinu pukotine b pomoću relacije (4.42)



Slika 4.8: Prikaz ogiba svjetlosti

Broj mjerenja	X[mm]	I/I$_0[mV$	Broj mjerenja	X[mm]	Napon[mv
1	0	0	21	2	6.6
2	0.1	0.1	22	2.1	6.2
3	0.2	0.3	23	2.2	5.4
4	0.3	0.4	24	2.3	4.7
5	0.4	0.6	25	2.4	3.8
6	0.5	0.5	26	2.5	2.4
7	0.6	0.5	27	2.6	1.7
8	0.7	0.4	28	2.7	0.9
9	0.8	0.1	29	2.8	0.2
10	0.9	0	30	2.9	0
11	1	0.2	31	3	0.1
12	1.1	0.7	32	3.1	0.2
13	1.2	1.4	33	3.2	0.4
14	1.3	2.7	34	3.3	0.7
15	1.4	3.8	35	3.4	0.5
16	1.5	4.6	36	3.5	0.4
17	1.6	5.8	37	3.6	0.3
18	1.7	6.1	38	3.7	0.1
19	1.8	6.3	39	3.8	0.1
20	1.9	6.7	40	3.9	0

Tablica br.3 Prikaz ovisnosti napona o pomaku promatrane pruge kod ogiba svjetlost.

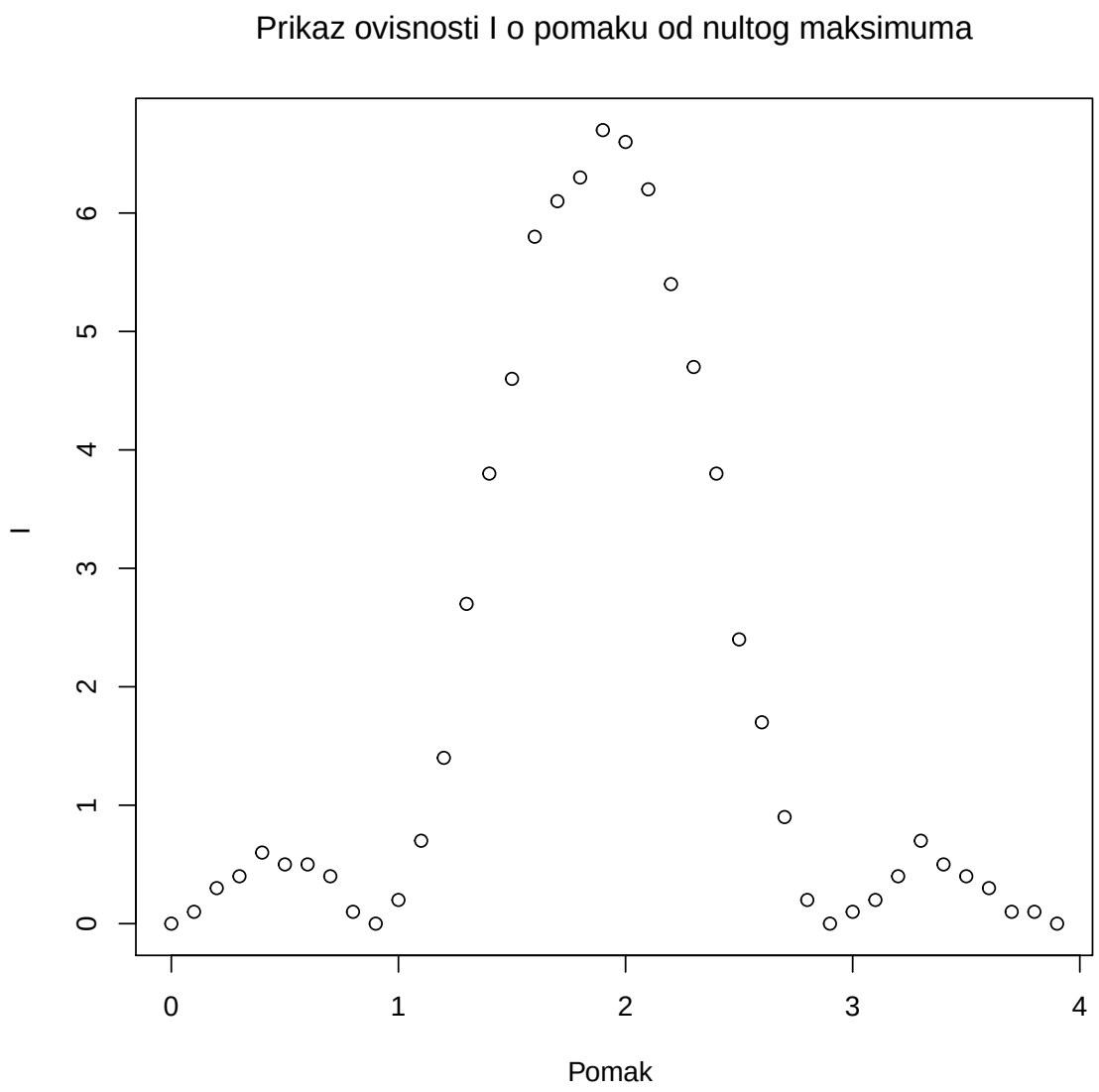
Podatke u R-u možemo učitati i iz vanjske datoteke:

```
#Učitavmo podatke iz vanjske datoteke
ogb <-read.csv("Ogibmjerenja.csv",sep=";",header=TRUE)
ogb$Napon.mv. <- round(ogb$Napon.mv./max(ogb$Napon.mv.),2)
ogb$X.mm.<-seq(-1.9,2,by=0.1)
kable(head(ogb))
```

Broj.mjerenja	X.mm.	Napon.mv.
1	-1.9	0.00
2	-1.8	0.01
3	-1.7	0.04
4	-1.6	0.06
5	-1.5	0.09
6	-1.4	0.07

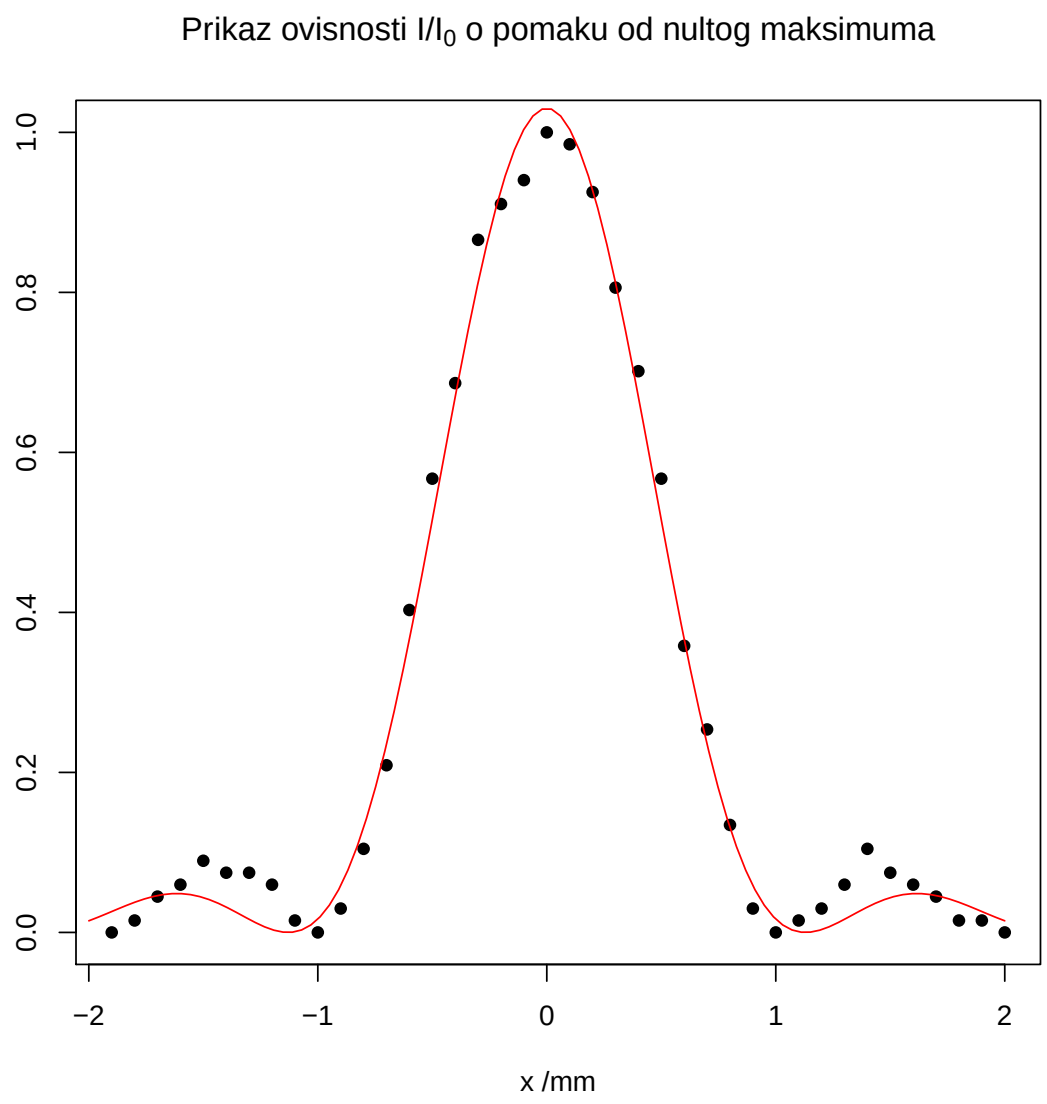
```
x<-c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,
     1,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,
```

```
     2,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9,3,
     3.1,3.2,3.3,3.4,3.5,3.6,3.7,3.8,3.9)
y<-c(0,0.1,0.3,0.4,0.6,0.5,0.5,0.4,0.1,0,0.2,
     0.7,1.4,2.7,3.8,4.6,5.8,6.1,6.3,6.7,6.6,
     6.2,5.4,4.7,3.8,2.4,1.7,0.9,0.2,0,0.1,0.2,
     0.4,0.7,0.5,0.4,0.3,0.1,0.1,0)
plot(x,y, xlab=TeX('Pomak [m]'), ylab=TeX('$I$ [mV]'),
     main=TeX('Prikaz ovisnosti $I$ o pomaku od nultog maksimuma'))
```




```
mjerenje <- read.csv("Ogibmjerenja.csv",sep=";",header=TRUE)
mjerenje$Napon.mv.<- mjerenje$Napon.mv./max(mjerenje$Napon.mv.)
mjerenje$X.mm.<-seq(-1.9,2,by=0.1)

m<-nls(Napon.mv.~b*(sin(a*(X.mm./(sqrt(X.mm.^2+1765^2))))/
                      (a*(X.mm./(sqrt(X.mm.^2+1765^2)))))^2,
```

```
       data=mjerenje,start=list(a=100,b=6.9))
tt <- seq(-2, 2, length = 100)
mjerenje2<-data.frame(X.mm.=tt,Napon.mv.=predict(m,list(X.mm.=tt)))
summary(m)

##
## Formula: Napon.mv. ~ b * (sin(a * (X.mm./(sqrt(X.mm.^2 + 1765^2))))/(a *
##     (X.mm./(sqrt(X.mm.^2 + 1765^2)))))^2
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## a 4908.4663    71.2899   68.85   <2e-16 ***
## b    1.0302     0.0143   72.03   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03346 on 38 degrees of freedom
##
## Number of iterations to convergence: 8
## Achieved convergence tolerance: 1.103e-06

plot(mjerenje$X.mm.,mjerenje$Napon.mv.,col="black",pch=16,
     main=TeX("Prikaz ovisnosti $I/I_0$ o pomaku od nultog maksimuma"),
     xlab=TeX("x /mm"),ylab=TeX(""))
lines(mjerenje2$X.mm.,mjerenje2$Napon.mv.,col="red")
```



Mjerenja prikazuju promjenu intenziteta ogibne svjetlosti prema intenzitetu nultog maksimuma.

Bitno je definirati sljedeće:

- žarišna duljina divergentne leće f =-0.05 m
- razmak između leće i pukotine 0.075 m
- valna dužina svjetlosti $\lambda = 632.810^{-9}$ m
- udaljenost pukotine do fotoćelije, odnosno $a = 1.765$ m
- procjena pogreške od a, $M_a = 0.001$ m
- udaljenost između drugog i prvog ogibnog minimuma $\Delta x_{2,1} = 0.008$ m

Prema tome, a možemo zapisati kao:

$$a = (1,765 \pm 0,001)m \tag{4.43}$$

a širinu pukotine možemo odrediti pomoću realcije:

$$b = \frac{(2n+1)\lambda}{2\sin\theta^{max}} = (n=1), (\alpha = sin\theta^{max}) = \frac{3\lambda}{2\alpha} \tag{4.44}$$

uz uvjet

$$\alpha = \sin\theta = \frac{x_1^{max}}{\sqrt{(X_1^{max})^2 + a^2}} \tag{4.45}$$

b je jednak:

```
#Određujemo širine pukotine b
lamda<-632.8*10^(-9)

xmax1<-0.008
aa<-1.765
alpha<-((xmax1)/sqrt((xmax1^2)+(aa^2)))
be<-((3*lamda)/(2*alpha))
be

## [1] 0.0002094194
```

a pogrešku računamo pomoću relacije:

$$\Delta M_G = \sqrt{\sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} \Delta \bar{y}_i \right)} \tag{4.46}$$

Odnosno:

$$\Delta M_b = \left| \frac{(3\lambda a^2) \cdot M_a}{2x_1^{max}\sqrt{(X_1^{max})^2 + a^2}} \right| \tag{4.47}$$

```
#Određujemo grešku za širinu pukotine b
lamda<-632.8*10^(-9)
xmax1<-0.008
aa<-1.765
Maa<-0.001
Mbe<-((3*lamda*aa^2)*Maa/(2*xmax1+sqrt((xmax1^2)+(aa^2))))
Mbe

## [1] 3.320541e-09
```

a širinu pukotine definiramo kao:

$$b = (20,942 \pm 0,003) \cdot 10^{-5}\,\mathrm{m} \tag{4.48}$$

4.2.3 Zaključak

U prvom dijelu vježbe zadatak je bio pronaći raspodjelu intenziteta svjetlosti uzduž ogibne slike nastale na pukotini te nacrtati ovisnost intenziteta o udaljenosti od centralnog maksimuma. Raspodjela intenziteta svjetlosti je nelinearna, središnja nulta pruga je najbolje osvijetljena a intenzitet osvijetljenosti ostalih pruga naglo pada. Drugim riječima, što je promatrana pruga dalje od središnje nulte pruge (maksimuma), njezin intenzitet je sve manji. U drugom dijelu vježbe zadatak je bio izračunati širinu pukotine. Izračunata vrijednost $b = (20,942 \pm 0,003) \cdot 10^{-5}\,\mathrm{m}$ u skladu je sa stvarnom vrijednosti širine pukotine. Programski jezik R korišten je u svim djelovima obrade podataka, pokazan je dio njegovih mogućnosti u obliku jednostavnih (množenje, oduzimanje, grafičko prikazivanje...) i složenijih operacija (nelinearna regresija).

5 Zaključak

Ovim diplomskim radom prikazane su osnove, način rada, mogućnosti i primjena programskog jezika R na primjerima fizikalnog praktikuma. Vježbe, Matematičko njihalo i Ogib svjetlosti su prikazale lakoću kojom programski jezik odrađuje jednostavne operacije poput množenja i grafičkog prikaza, te složenije poput linearne i nelinearne regresije.

Jednostavnost, dostupnost (programska podrška R-a slobodnog je dohvata) i velika statistička moć glavne su karakteristike programskog jezika R što omogućava široku primjenu u obrazovnim i poslovnim krugovima.

Možemo zaključiti kako programski jezik R i njegove mogućnosti zadovoljavaju sve zahtjeve jednog fizikalnog praktikuma te R može poslužiti kao kvalitetna zamjena za komercijalne programe koji se koriste prilikom obrade podataka.

Literatura

[1] Čižmenšija Mirjana, Sorić Petar: Statističke metode za poslovno upravljanje u programskom jeziku R, 1. izdanje, Zagreb, 2016.

[2] Bevingtton.P.R.: Data reducion and Error Analaysis for Physical sciences, Megraw -Hill Book Company, 1969.

[3] M.Kumbatović, D.Kasum, T.Legović : Uvod u korištenje R-a Verzija 2.0.1 (15.11.2004) https://cran.r-project.org/doc/contrib/Kasum+Legovic-UvodUr.pdf (3.9.2016.)

[4] Pomoć u R-u, baza svih funkcija i primjera https://www.r-project.org/help.html

[5] Simple Linear Regression with R, Statistic Courese http://courses.statistics.com/software/R/R_Ch02.htm, (14.7.2016).

[6] Fizikalni praktikum, Vježba br. 1, Matematičko njihalo, http://www.phy.pmf.unizg.hr/~gorjana/nastava/Praktikum/Fizicki%20praktikum%201/vjezbe%20i%20upute/Vjezba1.pdf, 1.5.2016.

[7] Fizikalni praktikum, Vježba br. 6, Ogib svjetlosti, http://www.phy.pmf.unizg.hr/~gorjana/nastava/Praktikum/Fizicki%20praktikum%202/vj6_ogib.pdf, 1.5.2016.