Uniformna distribucija i simulacije

Flajšman, Filip

Sveučilište u Rijeci - Fakultet za matematiku

Sveučilišni preddiplomski studij Matematika

Filip Flajšman

Uniformna distribucija i simulacije

Završni rad

Rijeka, rujan, 2023.

Sveučilište u Rijeci - Fakultet za matematiku

Sveučilišni preddiplomski studij Matematika

Filip Flajšman

Uniformna distribucija i simulacije

Mentor: izv. prof. dr. sc. Danijel Krizmanić

Završni rad

Rijeka, rujan, 2023.

Sadržaj

Sažetak

Glavni pojmovi koji će se obraditi u ovom radu su: uniformna distribucija, slučajna varijabla i simulacija.
Objasnit će se na koji način računalo stvara nasumične brojeve (random-number generator), što su to slučajni, a što pseudo-slučajni brojevi, te će se navesti načini generiranja nasumičnih brojeva pomoću tog generatora.
Zatim će se generiranje slučajnih brojeva povezati sa simulacijama iz intervala, iz prvih M prirodnih brojeva, iz diskretne distribucije, nasumičnog podskupa cijelih brojeva, povezat će se također sa slučajnim permutacijama i sa vjerojatnosti.

Ključne riječi: generator slučajnih brojeva, uniformna distribucija, slučajna varijabla, Bernoullijeva slučajna varijabla, neprekidna slučajna varijabla, eksponencijalna slučajna varijabla, simulacija.

Uvod

Uniformna distribucija i simulacije spadaju u područje matematike teorija vjerojatnosti. Uniformna distribucija je iznimno važan pojam u vjerojatnosti jer se iza nje kriju razne situacije iz stvarnog života na koje ljudi žele odgovor: "Koliko često se pojavljuje...?", "Koliko je vjerojatno da...?",... Kod takvih pitanja pomažu računala jer se pomoću njih izrađuju simulacije gdje se ista situacija provodi nekoliko stotina, tisuća, milijuna,... puta i dobije približna vrijednost, ili više njih, što se može (najviše) očekivati kao ishod.

Poglavlje 1

Uniformna distribucija

Uniformna distribucija (ili razdioba) je jedna od distribucija koju mogu imati neprekidne slučajne varijable, ali kako bi definirali uniformnu distribuciju, potrebno je definirati još neke pojmove pomoću kojih se dolazi do ovog.

Definicija 1.1. *Familija $\mathcal{F}$ podskupova od Ω je **σ-algebra skupova** na Ω ako je:*
(F1) $\emptyset \in \mathcal{F}$
(F2) $A \in \mathcal{F} \Rightarrow A^C \in \mathcal{F}$
(F3) $A_i \in \mathcal{F}(i \in \mathbb{N}) \Rightarrow \bigcup\limits_{i=1}^{\infty} A_i \in \mathcal{F}$

Definicija 1.2. *Neka je Ω neprazan skup i $\mathcal{F}$ σ-algebra na Ω. Funkcija $P : \mathcal{F} \to \mathbb{R}$ je **vjerojatnost** ako vrijedi:*
(P1) $P(A) \geq 0, \forall A \in \mathcal{F}$
(P2) $P(\Omega) = 1$
(P3) $A_i \in \mathcal{F}(i \in \mathbb{N})$ *i* $A_i \cap A_j = \emptyset$ *za* $i \neq j \Rightarrow P(\bigcup\limits_{i=1}^{\infty} A_i) = \sum\limits_{i=1}^{\infty} P(A_i)$

Definicija 1.3. *Uređena trojka $(\Omega, \mathcal{F}, P)$, gdje je $\mathcal{F}$ σ-algebra na nepraznom skupu Ω i P vjerojatnost na $\mathcal{F}$, zove se **vjerojatnosni prostor**. Elementi od $\mathcal{F}$ nazivaju se **događaji**, a za $A \in \mathcal{F}$ broj $P(A)$ zove se **vjerojatnost događaja** A.*

Definicija 1.4. *Stavimo $\mathcal{S} = \{\langle a, b\rangle : a, b \in \mathbb{R}, a < b\} \subseteq \mathcal{P}(\mathbb{R})$. $\mathcal{S}$ je familija (omeđenih) otvorenih intervala u $\mathbb{R}$. Najmanju σ-algebru na $\mathbb{R}$ koja sadrži familiju $\mathcal{S}$ označavamo s $\mathcal{B}$ i nazivamo **Borelova σ-algebra**, a njezine elemente **Borelovi skupovi**.*

Definicija 1.5. *Neka je $(\Omega, \mathcal{F}, P)$ vjerojatnosni prostor. Funkcija $X : \Omega \to \mathbb{R}$ je **slučajna varijabla** na Ω ako je $X^{-1}(B) \in \mathcal{F}, \forall B \in \mathcal{B}$.*

Definicija 1.6. ***Funkcija distribucije*** *slučajne varijable X je funkcija $F_X : \mathbb{R} \to [0,1]$ definirana sa*

$$F_X(x) = P(X \leq x), \quad x \in \mathbb{R}.$$

Definicija 1.7. *Funkcija $g : \mathbb{R} \to \mathbb{R}$ je* ***Borelova funkcija*** *ako je $g^{-1}(B) \in \mathcal{B}, \forall B \in \mathcal{B}$.*

Definicija 1.8. *Neka je X slučajna varijabla na vjerojatnosnom prostoru $(\Omega, \mathcal{F}, P)$ i neka je F njena funkcija distribucije. Kažemo da je X* ***neprekidna slučajna varijabla*** *ako postoji nenegativna Borelova funkcija $f : \mathbb{R} \to \mathbb{R}$ takva da vrijedi*

$$F(x) = \int_{-\infty}^{x} f(t)dt.$$

Funkciju f zovemo ***funkcija gustoće*** *od X i često je označujemo s f_X.*

Definicija 1.9. *Neprekidna slučajna varijabla ima* ***uniformnu distribuciju*** *(ili* ***razdiobu****) na segmentu $[a,b]$ ako je njena funkcija gustoće f dana sa*

$$f(x) = \begin{cases} \frac{1}{b-a}, & x \in [a,b] \\ 0, & \text{inače} \end{cases}$$

Uniformna distribucija na segmentu $[a,b]$ označava se sa $U(a,b)$.

1.1 Generator nasumičnih (slučajnih) brojeva

Generirati (pravi) slučajan broj znači odabrati broj, među ponuđenim brojevima, na način da se ne može predvidjeti koji broj će biti izvučen. Taj postupak se koristi kada želimo što manje utjecati na namještanje određenih situacija: podjela nogometnih reprezentacija u skupine na prvenstvu, izvlačenje loto brojeva, odlučivanje o pobjedniku nagradne igre,... Spomenuli smo pravi slučajan broj, kasnije u radu ćemo objasniti na koji način računalo generira slučajne brojeve, odnosno *pseudo-slučajne brojeve.* Na sljedećem primjeru ćemo pobliže objasniti na koji način se može generirati pravi slučajni broj.

Primjer 1.1.1. *Neka se u kutiji nalazi* 150 *ping-pong loptica, svaka označena jednim brojem i neka su svi brojevi različiti. Promiješamo loptice u kutiji i bez gledanja uzimamo jednu lopticu. Na taj način je* ***generiran nasumičan broj*** *(od* 150 *ponuđenih). Zapišemo na papir izvučeni broj i vratimo lopticu u kutiju. Ukoliko ponovo promiješamo* *kutiju i ponovno izvučemo lopticu, opet ćemo dobiti nasumičan broj, zapišemo izvučeni broj na papir, vratimo lopticu u kutiju. Ponavljamo postupak koliko dugo želimo. Ovim načinom smo generirali niz nasumično odabranih brojeva. Niz dobiven na ovakav način je* ***niz nasumično generiranih brojeva****.*

Svojstvo koje ima niz nasumično generiranih brojeva je da svaki generirani broj u nizu slučajnih brojeva mora biti neovisan o ostalim izvučenim brojevima. Također, jedno svojstvo je da, kada bismo uzeli veliki podniz (mnogo uzastopnih članova) niza slučajno odabranih brojeva iz neke distribucije, generirali bismo niz koji izgleda kao da je odabran iz iste distribucije. Uzeti niz brojeva iz neke distribucije znači generirati slučajne brojeve koje obuhvaća ta distribucija (distribucija realnih, cijelih, ... brojeva, proizvoljnog skupa brojeva) jedan za drugim, gdje se neki brojevi pojavljuju češće, neki rijeđe, ovisno o tome kako izgleda distribucija, tj. kolika je vjerojatnost da se određeni broj pojavi. Na primjer, uzmimo skup ocjena (prirodni brojevi od 1 do 5). Pretpostavimo da se prilikom ocjenjivanja testova u nekom razredu ocjena 3 pojavljuje najviše puta, zatim 2 i 4, zatim 1 i 5. Prilikom odabira dovoljno dugog niza, svaki niz slučajno odabranih ocjena iz te distribucije će imati najviše ocjena 3, zatim 2 i 4, zatim 1 i 5. Nakon generiranog niza ocjena, uzmimo podniz (mnogo uzastopnih članova) odabranog niza ocjena. Tada će i taj podniz izgledati kao da je niz slučajnih brojeva odabran iz te distribucije.

Skup iz kojeg se generiraju brojevi može biti proizvoljno malen ili proizvoljno velik pa uzmimo interval $\langle 0, 1\rangle$.

Generator slučajnih brojeva generira niz brojeva iz intervala $\langle 0, 1\rangle$. Kada govorimo o slučajno odabranom broju iz $\langle 0, 1\rangle$, zapravo govorimo o tome da je vjerojatnost da se generirani broj nalazi u nekom podintervalu intervala $\langle 0, 1\rangle$ jednaka duljini tog podintervala. Svaka dva podintervala jednake duljine imaju jednaku vjerojatnost da se generirani broj nalazi u svakom od njih. S obzirom na svojstva koja treba zadovoljavati generator nasumičnih brojeva, vjerojatnosnu distribuciju slučajnog broja između 0 i 1 možemo poistovjetiti sa uniformnom distribucijom na $\langle 0, 1\rangle$, odnosno $U(0, 1)$. To je neprekidna distribucija, što znači da je ispravno govoriti da se slučajno odabrani broj nalazi u danom intervalu. Nema smisla pitati se za vjerojatnost pojedine vrijednosti. Vjerojatnost pridružena nekom intervalu se smanjuje kako se smanjuje duljina intervala i postaje 0 ukoliko je duljina intervala 0.

Generatorom slučajnih brojeva može se modelirati bacanje simetričnog novčića (*pismo-*

glava): ukoliko se generirani broj nalazi unutar intervala $\langle 0, \frac{1}{2}\rangle$, tada pišemo da je ishod *pismo*, a ako se generiran broj nalazi unutar $\langle \frac{1}{2}, 1\rangle$, tada pišemo da je ishod *glava* [1]. Slično se može modelirati i bacanje simetrične kocke, izvlačenje određene karte iz špila, ...
Ukoliko bismo htjeli vidjeti što se događa prilikom bacanja simetričnog novčića u nekoliko stotina, tisuća, milijuna, ... puta, u stvarnom životu bi to predugo trajalo i u takvim situacijama nam može pomoći ugrađena funkcija *random()* (u nekom obliku) u računalu. Proces funkcije *random()* je iterativan i određen je prikladno odabranom funkcijom f. Niz započinje proizvoljnim brojem z_0, a brojevi $z_1, z_2, \dots$ su generirani na način

$$z_1 = f(z_0), z_2 = f(z_1), \dots, z_n = f(z_{n-1}), \dots.$$

Funkciju f nazivamo **generator slučajnih brojeva** i treba biti odabrana tako da niz $\{z_i\}$ bude nerazlučiv od stvarnog niza slučajnih brojeva. Vrijednost dobivena funkcijom f treba biti "otporna" na jako veliki broj statističkih testova za slučajnost. Dodatna prednost je da se niz brojeva generiran generatorom slučajnih brojeva može ponoviti ukoliko se uzme isti početni broj z_0. Ovo može biti korisno prilikom testiranja funkcije na statističkim testovima jer će na svakom testu biti isti niz. S obzirom na to da računalo ne stvara zapravo slučajne brojeve, možemo reći i da su ti generirani brojevi *pseudo-slučajni brojevi.*
Najnoviji generatori slučajnih brojeva mogu proizvesti niz duljine (prije no što se počnu ponavljati brojevi) od oko 2^{1492} (Christopher Columbus generator) ili npr. Mersenne twister generator sa duljinom niza (prije početka ponavljanja) oko $2^{19937} - 1$.

1.2 Generiranje realizacija slučajnih varijabli

Kako smo u prethodnom potpoglavlju naveli, generator slučajnih brojeva daje broj između 0 i 1, čime oponaša razdiobu $U(0, 1)$. U daljnjem tekstu ćemo vidjeti neke vrste slučajnih varijabli koje se izvode iz uniformne razdiobe.

[1] Vjerojatnost da je slučajno generirani broj točno $\frac{1}{2}$ je 0 pa tu vrijednost možemo zanemariti, a može se i staviti da za slučajno generirani broj iz skupa $[\frac{1}{2}, 1\rangle$ pišemo ishod *glava* (umjesto intervala $\langle \frac{1}{2}, 1\rangle$)

1.2.1 Bernoullijeva slučajna varijabla

Neka U ima uniformnu distribuciju $U(0,1)$. Definiramo **Bernoullijevu slučajnu varijablu** s parametrom p, $p \in \langle 0,1 \rangle$, u oznaci Ber(p), sa

$$X = \begin{cases} 1, & U < p \\ 0, & U \geq p \end{cases}$$

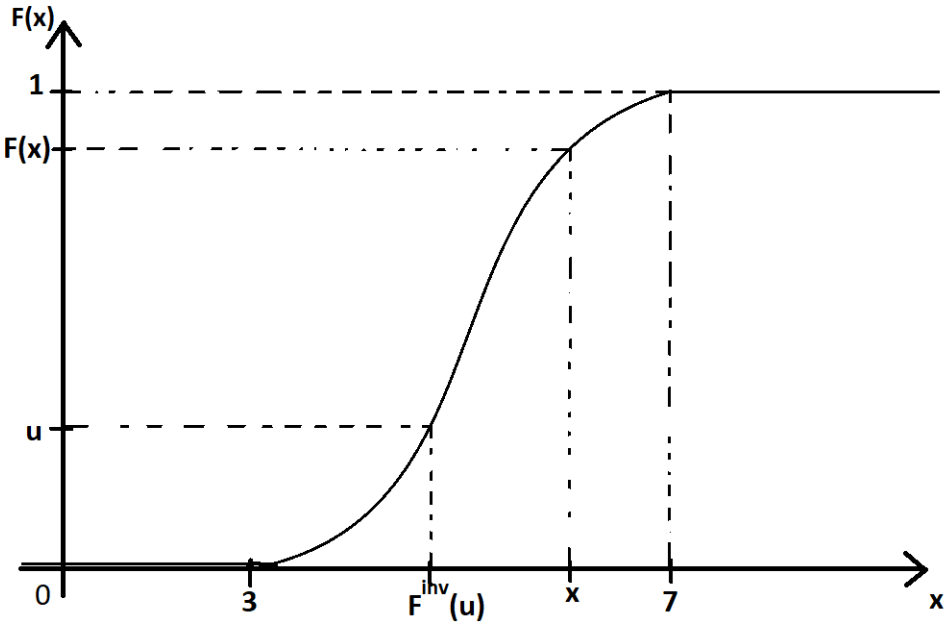
te tada vrijedi

$$P(X = 1) = P(U < p) = p,$$
$$P(X = 0) = P(U \geq p) = 1 - p.$$

Analogno se može napraviti za n vrijednosti $v_1, v_2, \ldots, v_n$ i n parametara $p_1, p_2, \ldots, p_n$.

1.2.2 Neprekidna slučajna varijabla

Neka je zadana funkcija distribucije F neprekidne slučajne varijable. Cilj nam je konstruirati slučajnu varijablu s istom distribucijom. To ćemo pokazati u slučaju kada F strogo raste od 0 prema 1. Tada F ima inverznu funkciju F^{inv}. Na slici 1.1 je prikazan primjer: F je strogo rastuća na segmentu $[3,7]$; postoji inverz $F^{inv} : [0,1] \to [3,7]$.



Slika 1.1: Graf funkcije distribucije neprekidne slučajne varijable

Primijetimo da se u odnosi prema $F^{inv}(u)$ kao što se $F(x)$ odnosi prema x. Vidimo da je $u \leq F(x)$ ekvivalentno sa $F^{inv}(u) \leq x$. Ako umjesto realnog broja u promatramo slučajnu varijablu U s razdiobom $U(0,1)$, dobivamo sljedeće:

$$\{U \leq F(x)\} = \{F^{inv}(U) \leq x\}. \tag{1.1}$$

Kako za slučajnu varijablu U vrijedi $P(U \leq b) = b$ za bilo koji $0 \leq b \leq 1$, supstitucijom $b = F(x)$ dobivamo

$$P(U \leq F(x)) = F(x). \tag{1.2}$$

Slijedi, iz (1.1)

$$P(F^{inv}(U) \leq x) = F(x),$$

tj. F je funkcija distribucije od $F^{inv}(U)$. Odredimo sada F^{inv}. Iz slike 1.1 vidimo

$$F(x) = u \Rightarrow x = F^{inv}(u).$$

Dakle, riješimo li jednadžbu $F(x) = u$ za x, dobivamo izraz za $F^{inv}(u)$.

1.2.3 Eksponencijalna slučajna varijabla

Eksponencijalna razdioba s parametrom $\lambda \in \langle 0, \infty \rangle$, u oznaci $X \sim Exp(\lambda)$, je definirana sa funkcijom gustoće

$$f_X(x) = \begin{cases} \lambda e^{-\lambda x}, & x > 0, \\ 0, & x \leq 0. \end{cases}$$

Funkcija distribucije od $Exp(\lambda)$ je oblika

$$F_X(x) = \int_{-\infty}^{x} f_X(t)dt = \begin{cases} 0, & x < 0, \\ \int_0^x \lambda e^{-\lambda t} dt = 1 - e^{-\lambda x}, & x \geq 0. \end{cases}$$

Ponavljamo isti postupak kao i za općenitu neprekidnu slučajnu varijablu, ali gledamo restrikciju funkcije distribucije na $\langle 0, \infty \rangle$ jer na tom intervalu funkcija distribucije strogo raste i dana je sa

$$F_X(x) = 1 - e^{-\lambda x}.$$

Da bismo našli F_X^{inv}, rješavamo jednadžbu $F_X(x) = u$:

$$\begin{aligned} F_X(x) = u &\Leftrightarrow 1 - e^{-\lambda x} = u \\ &\Leftrightarrow e^{-\lambda x} = 1 - u \\ &\Leftrightarrow -\lambda x = ln(1-u) \\ &\Leftrightarrow x = -\frac{1}{\lambda} ln(1-u). \end{aligned}$$

Slijedi $F_X^{inv}(u) = -\frac{1}{\lambda} ln(1-u)$ i ako uvrstimo slučajnu varijablu U sa distribucijom $U(0,1)$, tada slučajna varijabla X definirana sa

$$X = F_X^{inv}(U) = -\frac{1}{\lambda} ln(1-U)$$

ima eksponencijalnu ($Exp(\lambda)$) distribuciju sa parametrom λ.
Prilikom rješavanja zadataka, može se umjesto $1-U$ koristiti samo U jer obje slučajne varijable imaju distribuciju $U(0,1)$. To se često radi ukoliko se koristi računalo zbog efikasnosti. Sada, umjesto X imamo

$$Y = -\frac{1}{\lambda} ln(U),$$

koja također ima $Exp(\lambda)$ distribuciju.

Poglavlje 2

Simulacije

Za mnoge pojave iz stvarnog života želimo znati kako se ponašaju prilikom ponavljanja istih 10, 100, 1 000 000,... puta. To se posebno odnosi na pojave u kojima slučajnost ima veliku ulogu: izvlačenje loto brojeva, casino i druge igre na sreću, raspored karata kod igrača,... Kod takvih pitanja nam uveliko mogu pomoći računala gdje se može stvoriti matematički model pojave i testirati kako se ponaša prilikom ponavljanja istog postupka po nekoliko desetaka, stotina tisuća ili više puta. Svako ponavljanje zovemo jedan **pokus**, a cijeli postupak zovemo **simulacija** (pokus napravljen samo jednom je također simulacija - nula puta ponovljen).

2.1 Uspoređivanje dvaju pravila za ocjenjivanje

Na olimpijskim igrama je nekoliko sportova u kojima plasman ovisi o ljudima, sucima, koji ocjenjuju sportaše. Jedan od takvih sportova je gimnastika. Opisat ćemo 2 pravila koja se mogu koristiti za ocjenjivanje i želimo spriječiti podmićivanje sudaca što je više moguće, odnosno prigušiti odstupanje od stvarne ocjene najviše moguće. Žiri se sastoji od 7 sudaca i za svaki nastup, svaki sudac daje svoju ocjenu. Tih 7 ocjena se formira u završnu ocjenu. Postoje 2 pravila po kojima suci mogu formirati konačnu ocjenu:

(1) 7 ocjena se sortira po veličini i kao konačna ocjena odabere ona u sredini.

ili

(2) Najviša i najniža ocjena se izbace i konačna ocjena je prosjek preostalih 5.

Model

Pretpostavimo da ocjene sudaca odstupaju od prave, zaslužene ocjene. Neka je sudac i, za zasluženu ocjenu g, dao

$$Y_i = g + Z_i, \quad \text{za } i = 1, \ldots 7, \tag{2.1}$$

gdje su $Z_1, \ldots, Z_7$ slučajne varijable s vrijednostima oko nule. Neka su h_1 i h_2 funkcije koje opisuju navedena pravila:

$$h_1(y_1, \ldots, y_7) = \text{"srednja vrijednost od } y_1, \ldots, y_7\text{"},$$
$$h_2(y_1, \ldots, y_7) = \text{"prosjek srednjih 5 od } y_1, \ldots, y_7\text{"}.$$

Odstupanja od zaslužene ocjene su:

$$M = h_1(Y_1, \ldots, Y_7) - g, \tag{2.2}$$
$$T = h_2(Y_1, \ldots, Y_7) - g. \tag{2.3}$$

Distribucije od M i T ovise o pojedinim ocjenama sudaca, a time i o $Z_1, \ldots, Z_7$, koje modeliramo kao $U(-\frac{1}{2}, \frac{1}{2})$.
Sljedeća faza je analiza: koje odstupanje je bliže nuli? Kako su M i T slučajne varijable, možemo napraviti simulaciju.

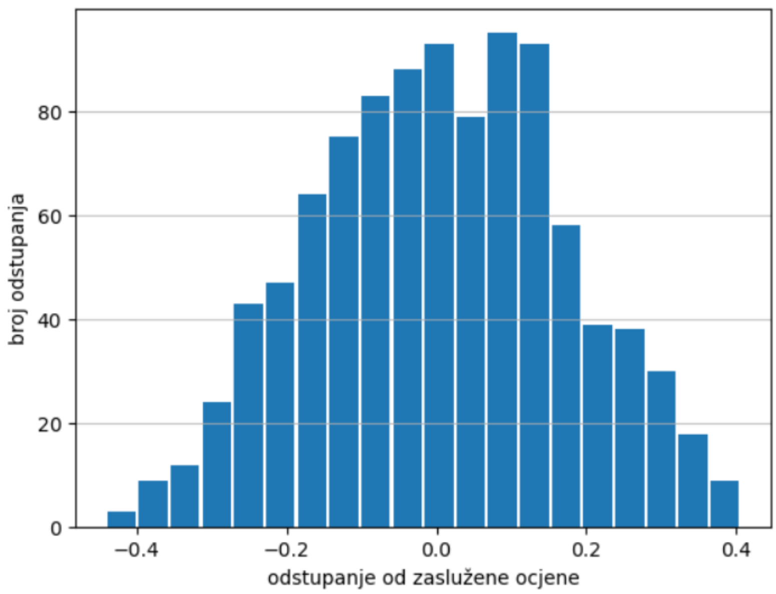
Simulacija

Kako bismo dobili vrijednosti u $U(-\frac{1}{2}, \frac{1}{2})$, trebamo od vrijednosti dobivene generatorom slučajnih brojeva oduzeti $\frac{1}{2}$. Napravimo to 7 puta i uvrstimo u (2.1) kao odstupanja $Z_1, \ldots, Z_7$, i supstituiramo ih u jednadžbi (2.2) da bismo dobili M i T:

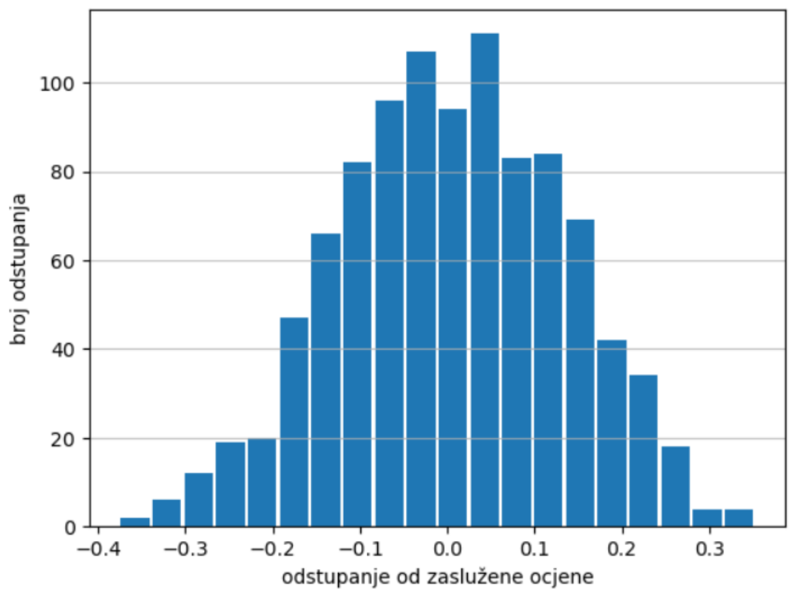
$$M = \text{srednja vrijednost od } Z_1, \ldots, Z_7,$$
$$T = \text{prosjek srednjih 5 od } Z_1, \ldots, Z_7.$$

Sada smo napravili jedan pokus. Ponovimo postupak 1 000 puta i načinimo histograme, jedan za M, drugi za T.

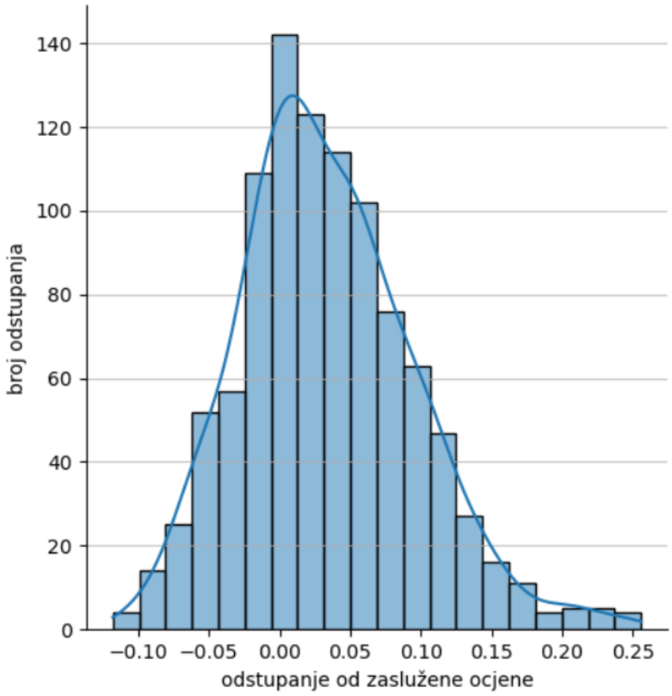
Ova dva histograma ne govore o odnosu između M i T, pa radimo histogram $|M| - |T|$, gdje je $|M|$ apsolutno odsupanje s obzirom na pravilo (1), a $|T|$ apsolutno odsupanje s obzirom na pravilo (2). Ako je razlika $|M| - |T|$ pozitivna, tada je T bliže nuli, a ako je



Slika 2.1: M, 1000 pokusa, preuzeto iz Python koda



Slika 2.2: T, 1000 pokusa, preuzeto iz Python koda



Slika 2.3: |M|-|T|, 1000 pokusa, preuzeto iz Python koda

negativna, M je bliže nuli. Na histogramu 2.3 može se vidjeti da je rezultat većeg broja simulacija desno od nule, što znači da je češće T bliže nuli, nego M. Odnosno, pravilo (2) je bolje u većini slučajeva.

2.2 Simulacije iz vjerojatnosnih distribucija

Ponekad je poželjno da se generirani broj nalazi između a i b realnih brojeva, da bude cijeli broj, ili želimo izvršiti nekakvo miješanje elemenata, ili nam treba samo podskup slučajno odabranih elemenata nekog skupa, ili želimo pomoću simulacije doći do neke konstante (ili barem njene približne vrijednosti).
Upravo o tome ćemo govoriti u ovom potpoglavlju, odnosno o široj uporabi generatora slučajnih brojeva.

2.2.1 Simulacija iz intervala

Primjer 2.2.1. *Najavljeno je da će kiša početi u slučajnom vremenu između* $3:00$ *i* $4:45$ *sati. Želimo napraviti model gdje nam je slučajno odabran broj vrijeme između* $3:00$ *i* $4:45$*?*

Potrebno je generirati broj između brojeva a i b, kada je $a < b$ (napravit ćemo odmah općeniti slučaj, a kasnije se lako uvrsti u formulu $a = 3$ i $b = 4\frac{3}{4}$). To se radi na sljedeći način: prvo se generira broj u između 0 i 1. Tada je traženi generirani broj između a i b

$$a + (b - a)u.$$

2.2.2 Simulacija iz prvih M prirodnih brojeva

Ako želimo generirati neki prirodan broj $n \leq M$, za neki $M > 0$, prvo odabiremo slučajan broj u između 0 i 1. Tada, koristeći notaciju $\lfloor f \rfloor$ za najveći cijeli broj koji je manji od f, broj

$$1 + \lfloor Mu \rfloor$$

je slučajno odabrani prirodan broj iz skupa $1, 2, \ldots, M$.
Ukoliko se želi dobiti neki prirodan broj n koji je jedan od brojeva $a, a + 1, \ldots, b$, uzima se broj

$$a + \lfloor (b - a + 1)u \rfloor.$$

Primjer 2.2.2. *Kolika je vjerojatnost da se u razredu od 23 djece nalaze 2 ili više djece koji imaju rođendane istog datuma?*
Pretpostavimo da godina ima 365 dana i da je vjerojatnost svakog rođendana jednaka.
U svakom pokretanju simulacije (pokusu) generiraju se brojevi $u_1, \ldots, u_{23}$ i dobiveni su rođendani $k_i = 1 + \lfloor 365u_i \rfloor, i \in \{1, \ldots, 23\}$. Pokus se smatra uspješnim ukoliko je $|k_i - k_j| = 0$, za neke $k_i \neq k_j$. Tražena vjerojatnost je omjer uspješnih pokusa i ukupnog broja pokusa u simulaciji. Koristeći 100 000 pokusa, dobili smo da je vjerojatnost da 2 ili više djece od 23 u jednom razredu imaju rođendan isti dan približno jednaka 0.5068 (zaokružena na 4 decimale). Kada bi se analitičkim putem dolazilo do rješenja, dobiva se da je vjerojatnost približno jednaka 0.5073 (zaokružena na 4 decimale).

2.2.3 Simulacija iz diskretne distribucije

Da bismo govorili išta o diskretnoj distribuciji, prvo je moramo definirati.

Definicija 2.1. *Slučajna varijabla X ima **diskretnu uniformnu razdiobu** s parametrom $n \in \mathbb{N}$ ako joj je distribucija dana sa*

$$X \sim \begin{pmatrix} k_1 & k_2 & \ldots & k_n \\ \frac{1}{n} & \frac{1}{n} & \ldots & \frac{1}{n} \end{pmatrix},$$

tj. $P(X = k_i) = \frac{1}{n}, k_i \in \mathbb{R}$.

Simulacija iz diskretne distribucije je jednostavna za računalo u slučaju kada je $n = 2$. U tom slučaju imamo razdiobu Bernoullijeve slučajne varijable X s parametrom $p \in \langle 0, 1\rangle$

$$X \sim \begin{pmatrix} x_1 & x_2 \\ p & 1-p \end{pmatrix}.$$

Za generirani slučajan broj $u \in \langle 0, 1\rangle$ X poprima vrijednost x_1, ako je $u \leq p$ (s vjerojatnošću p), ili x_2, s vjerojatnošću $1 - p$.

Neka je sada X slučajna varijabla zadana distribucijom $X \sim \begin{pmatrix} x_1 & \cdots & x_M \\ p_1 & \cdots & p_M \end{pmatrix}$ i neka je $M > 2, M \in \mathbb{N}$. Kada je M dovoljno velik, traženje indeksa l takvog da zadovoljava $p_1 + \ldots + p_{l-1} < u \leq p_1 + \ldots + p_l$ je previše vremenski zahtjevno za računalo pa se koristi sljedeća metoda. Cilj je podijeliti ukupnu vjerojatnost 1 vrijednosti $x_1, \ldots, x_M$ na D jednakih dijelova vrijednosti $\frac{1}{D}$, gdje je D dovoljno velik prirodan broj t.d. vrijedi $D > M$ (npr. $D = 2M$). Za svaki od D dijelova $d = 1, 2, \ldots, D$ stavimo vjerojatnost $\frac{1}{D}$. Posljedica toga da je D dovoljno velik je taj da će se samo nekoliko vrijednosti x_j nalaziti unutar svakog dijela. Sada možemo simulirati vjerojatnost na sljedeći način. Prvo se nasumično odabire jedan od D dijelova. Zatim, odredimo u tom dijelu vrijednost x_l za koju je $p_1 + \ldots + p_{l-1} < u \leq p_1 + \ldots + p_l$. Kako se dodijeljuju vrijednosti x_j u dijelove d, najbolje je pokazati na primjeru.

Primjer 2.2.3. *Neka je zadana slučajna varijabla X sa $M = 4$ vrijednosti i njihovim vjerojatnostima $p_j = P(X = x_j)$, za $j = 1, 2, 3, 4$ sa*

$$p_1 = 0.3, p_2 = 0.2, p_3 = 0.35, p_4 = 0.15.$$

Uzmimo $D = 5$ dijelova $d_1, \ldots, d_5$. Svaki dio predstavlja vjerojatnost od 0.2 U d_1 ćemo staviti vrijednost x_1, ali samo vjerojatnost 0.2 (od cijelog $p_1 = 0.3$) jer je d_1 veličine 0.2. x_1 ćemo staviti i u d_2 s vjerojatnošću 0.1, a u d_2 ćemo staviti i x_2 s preostalom mogućom vjerojatnošću za d_2 0.1. U d_3 ćemo staviti x_2 i x_3 s vjerojatnostima 0.1. U d_4 ćemo staviti vrijednost x_3 s vjerojatnošću 0.2, a u d_5 ćemo staviti vrijednosti x_3 i x_4 s vjerojatnostima 0.05 i 0.15 respektivno. Sada imamo 5 dijelova s 5 jednakih vjerojatnosti za svaki dio, odnosno imamo diskretnu slučajnu varijablu pa možemo raditi simulaciju iz nje. Simulacija se provodi sljedećim algoritmom:

(i) Generira se broj u između 0 i 1.

(ii) Na slučajan način se odabire jedan od dijelova $d_1, \ldots, d_5$ *preko indeksa* s *sa* $s = 1 + \lfloor Du \rfloor$.

(iii) U dijelu d_s *tražimo vrijednost* x_l *sa* $\sum_{i=1}^{l-1} p_i < u \leq \sum_{i=1}^{l} p_i$.

Neka je u koraku (i) generiran broj 0.5673... Tada je u koraku (ii) odabran dio d_3. *U* d_3 *se nalaze vrijednosti* x_2 *i* x_3 *s vjerojatnostima* 0.1. *U koraku (iii) je kao rezultat dobivena vrijednost* x_3.

U slučaju kada su vjerojatnosti $p_1, \ldots, p_M$ dane sa najviše konačno mnogo decimala d, tada se p_j može prikazati kao $\frac{k_j}{10^d}$, za neki $0 < k_j \leq 10^d$, za $j = 1, \ldots, M$. Nakon toga se formira lista $A[i], i = 1, \ldots, 10^M$ i prvim k_1 elementima liste $A[i]$ pridružena je vrijednost x_1, idućim k_2 elementima liste $A[i]$ vrijednosti x_2, ..., zadnjim k_M elementima liste $A[i]$ pridružena je vrijednost x_M. Analognim načinom, kao i u primjeru, dobiva se slučajno generirana vrijednost simulacijom iz diskretne distribucije. Jedina razlika je da se u koraku (iii) algoritma iz primjera direktno dolazi do vrijednosti x_j na način da se uzme s-ti element liste, gdje je s prirodan broj dobiven u koraku (ii).

2.2.4 Slučajne permutacije

Neka je zadan niz brojeva $1, \ldots, n$. Želimo promiješati brojeve, odnosno uzeti jednu permutaciju niza $1, \ldots, n$. Za to postoji sljedeći algoritam:

(i) Neka je $t := n$ i $a[j] := j$, za $j = 1, \ldots, n$.

(ii) Generira se slučajan broj u između 0 i 1.

(iii) Neka je $k := 1 + \lfloor tu \rfloor$ (slučajan cijeli broj između 1 i t)

Zamijene se vrijednosti $a[k]$ i $a[t]$.

(iv) Neka je $t := t - 1$. Ako je $t > 1$, vraća se na korak (ii), u suprotnom staje i dobivena je permutacija $(a[1], \ldots, a[n])$.

Ideja algoritma je da prvo nasumično odabere jedan od n cijelih brojeva i zamijeni ga sa zadnjim, tada nasumično odabere broj od preostalih $n - 1$ brojeva i zamijeni ga sa predzadnjim,... Taj postupak se nastavlja dok ne dođe do prvog broja i s obzirom na to da ga ne može zamijeniti ni s jednim drugim, tu je kraj algoritma.

Ova metoda se koristi kad god se želi nasumično svakoj osobi pridružiti neki element ili npr. prilikom miješanja špila karata, što ćemo i koristiti u sljedećem primjeru

Primjer 2.2.4. *Neka je špil karata od* 52 *karte nasumično promiješan i okrenut licem prema dolje. Kolika je vjerojatnost da je treća karta* 4*-srce?*

Označimo karte na sljedeći način: as-pik brojem 1*, 2-pik brojem* 2*, ..., kralj-pik brojem* 13*, as-tref brojem* 14*, ..., kralj-tref brojem* 26*, as-karo brojem* 27*, ..., kralj-karo brojem* 39*, as-srce brojem* 40*, ..., 4-srce brojem* 43*, ..., kralj-srce brojem* 52*. Kad se karte okrenu licem prema dolje, želimo da treća karta (indeks* 49*) bude označena brojem* 43*.*
Koristimo prethodni algoritam za miješanje karata i iz dobivene permutacije $(a[1], \ldots, a[52])$ *uzimamo kartu* $a[49]$ *i kažemo da je pokus uspješan ukoliko vrijedi* $a[49] = 43$*. Takvih pokusa radimo* 1 000 000*. Kao traženu vjerojatnost uzimamo omjer uspješnih pokusa i ukupnog broja simulacija.*
Dobiven je broj 0.0195*, a analitičkim načinom se dolazi do broja* $\frac{1}{52} \approx 0.0192$*.*

2.2.5 Simulacija nasumičnog podskupa cijelih brojeva

Potrebno je odabrati jedan podskup od k elemenata nekog skupa od n elemenata ($k < n$). Ovaj postupak možemo i na primjeru objasniti jer je sličan kao i slučajne permutacije.

Primjer 2.2.5. *Modeliramo izvlačenje Lotta* 6/45*, tj. odabiremo* 6 *naumično odabranih brojeva od njih* 45*. Radi se isti algoritam kao i kod slučajnih permutacija, a kako nam je potrebno* 6 *brojeva, napravi se samo* 6 *koraka, a ne* 45*, odnosno popuni se zadnjih* 6 *mjesta s nasumičnim brojevima, što je u konačnici prikladnije za računalo.*

2.2.6 Simulacija i vjerojatnost

Često se u fizici konstante određuju na ekperimentalni način. Odredimo približno vrijednost broja π u sljedećem primjeru.

Primjer 2.2.6. *Neka je u koordinatnom sustavu zadan kvadrat s vrhovima* $(-1, -1)$, $(-1, 1)$, $(1, -1)$ *i* $(1, 1)$*, te neka je u taj kvadrat upisan krug. Površina zadanog kruga je* $\lambda(KRUG) = 1^2\pi = \pi$*, a površina kvadrata je* $\lambda(KVADRAT) = 4$*. Nasumično odabiremo* 10 000 *točaka unutar kvadrata. Neke točke će biti unutar kruga i kvadrata, ostale će biti unutar kvadrata, ali ne i kruga. Iz geometrijske vjerojatnosti, vjerojatnost da će se slučajno odabrana točka nalaziti unutar kruga,* $P_{UnutarKruga}$*, je*

$$\begin{aligned} P_{UnutarKruga} &= \frac{\lambda(KRUG)}{\lambda(KVADRAT)} \\ &= \frac{\pi}{4} \end{aligned}$$

Iz ovoga slijedi

$$\pi = 4 \cdot P_{UnutarKruga}.$$

Sada, napravimo 1 000 *aproksimacija broja* π *(*$k_1, \ldots, k_{1000}$*) prebrojavanjem točaka unutar kruga i računanjem omjera broja točaka unutar kruga i ukupnog broja točaka te množenjem tog omjera sa* 4*. Zatim izračunamo prosjek svih* 1 000 *aproksimacija od* π*. Dobili smo da je aproksimacija broja* π 3.142324*, dok je* π *zaokružen na* 6 *decimala* 3.141592.

Zaključak

Glavni pojam u ovom radu je bio uniformna distirbucija - $U(0,1)$. Upoznali smo se s načinom kako generator slučajnih brojeva generira brojeve, na koji način se uniformna distribucija može uklopiti u Bernoullijevu slučajnu varijablu, te eksponencijalnu slučajnu varijablu, odnosno općenito u neprekidnu slučajnu varijablu. Zatim smo definirali simulacije, vidjeli jedan primjer sa provjerom legitimnosti ocjene gdje se koriste simulacije i nakon toga smo naveli neke situacije gdje se mogu koristiti simulacije na razne načine: simulacija iz intervala, simulacija iz prvih M prirodnih brojeva, simulacija iz diskretne distribucije, slučajne permutacije, simulacija nasumičnog podskupa cijelih brojeva, simulacija i vjerojatnost.

Prilozi

Kod: Uspoređivanje dvaju pravila za ocjenjivanje

```
import random as rnd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

def avg(x):
    return sum(x)/len(x)

def h2(y):             #h2 -> uzimanje prosjeka od srednjih pet vrijednosti
    y.sort()
    del y[0]
    del y[-1]
    prosjek=avg(y)
    return prosjek

def h1(y):             #h1 -> uzimanje srednje vrijednosti
    y.sort()
    return y[int(len(y)/2)]

def odst(z):
    return z-0.5

n=1000
```

```
br_ocjena=7

odstupanja=[]
for j in range(n):
    odstupanja.append([])
    for i in range(br_ocjena):
        odstupanja[j].append(odst(rnd.random()))

M=np.zeros(n)
T=np.zeros(n)

for i in range(n):
    M[i]=h1(odstupanja[i])

for i in range(n):
    T[i]=h2(odstupanja[i])




plt.hist(M,bins=20,rwidth=0.9)
plt.xlabel('odstupanje od zaslužene ocjene')
plt.ylabel('broj odstupanja')
plt.grid(axis='y', alpha=0.75)
plt.show()
plt.hist(T,bins=20,rwidth=0.9)
plt.xlabel('odstupanje od zaslužene ocjene')
plt.ylabel('broj odstupanja')
plt.grid(axis='y', alpha=0.75)
plt.show()
sns.displot(abs(M)-abs(T), kde=True, bins=20)
plt.xlabel('odstupanje od zaslužene ocjene')
```

```
plt.ylabel('broj odstupanja')
plt.grid(axis='y', alpha=0.75)
plt.show()
```

Kod: Simulacija iz cijelih brojeva - rođendani

```
import random as rnd
import math

def rodendani_od_n_ljudi(n):
    R=[]
    for i in range(n):
        r=rnd.random()
        R.append(1+math.floor(365*r))
    return R

n = 100000
br_ljudi=23
istina = 0
for a in range(n):
    R = rodendani_od_n_ljudi(br_ljudi)

    isti_dan = False
    for b in range(len(R)):
        for c in range(len(R)):
            if (b!=c) and (abs(R[b]-R[c]) == 0):
               #za primjer "jedan dan razlike" je linija:
             if (b!=c) and (abs(R[b]-R[c]) == 1 or abs(R[b]-R[c]) == 364):
                isti_dan = True
                break
    if isti_dan == True:
        istina += 1
print("Aproksimacija vjerojatnosti da dvije osobe od",br_ljudi,
```

```
"imaju rođendan isti dan je",istina/n,".")
```

Kod: Slučajne permutacije - 3. karta u špilu 4-srce

```
#Označimo karte na sljedeći način: as-pik brojem 1, 2-pik brojem 2, ...,
#kralj-pik brojem 13,as-tref brojem 14, ..., kralj-tref brojem 26,
#as-karo brojem 27, ..., kralj-karo brojem 39,
#as-srce brojem 40, ..., 4-srce brojem 43, ..., kralj-srce brojem 52

#Kad se karte okrenu licem prema dolje, želimo da treća karta (indeks 52-3=49)
#bude označena brojem 43

import numpy as np
import random as rnd
import math

def sluc_perm(n):
    t=n-1
    a=np.arange(n)+1
    while(t>1):
        u=rnd.random()
        k=math.floor(t*u)
        zamjena=a[t]
        a[t]=a[k]
        a[k]=zamjena
        t=t-1
    return a

uspjeh=0
trazena_karta=43
br_pokusa=1000000
for i in range(br_pokusa):
```

```
    x=sluc_perm(52)
    if x[49]==43:
        uspjeh+=1

print("Vjerojatnost:",uspjeh/br_pokusa)
```

Kod: Simulacije nasumičnog podskupa cijelih brojeva - Lotto 6/45

```
def loto_r_n(r,n):
    t=n-1
    a=np.arange(n)+1
    izvuceni=np.arange(r)
    for i in range(6):
        u=rnd.random()
        k=math.floor(t*u)
        zamjena=a[t]
        a[t]=a[k]
        a[k]=zamjena
        t-=1
        izvuceni[i]=a[n-1-i]



    return izvuceni
print(np.sort(loto_r_n(6,45)))
```

Kod: Simulacije i vjerojatnost - Aproksimacija od π

```
import random as rnd
import numpy as np
import pylab as pl
```

```
def modul(x,y):
    return np.sqrt(x**2+y**2)

n = 10000
avg = 1000
avg_list = []

for j in range(avg):
    X = np.empty(n)
    Y = np.empty(n)
    pogodak = 0
    pogodak_ind_X = []
    pogodak_ind_Y = []
    ne_pogodak_ind_X = []
    ne_pogodak_ind_Y = []
    for i in range(n):
        X[i] = rnd.random()*2-1
        Y[i] = rnd.random()*2-1
        if modul(X[i],Y[i]) < 1:
            pogodak += 1
            pogodak_ind_X.append(X[i])
            pogodak_ind_Y.append(Y[i])
        else:
            ne_pogodak_ind_X.append(X[i])
            ne_pogodak_ind_Y.append(Y[i])
    #P_pogodak = pogodak / n
    #print(j+1, 4 * P_pogodak)   #aproksimacija broja pi
    avg_list.append(pogodak/n)



size_plot = 200
```

```
x = np.linspace(-1,1,size_plot)
x_1 = np.linspace(1,1,size_plot)
x_neg1 = np.linspace(-1,-1,size_plot)
pl.figure(figsize = (10,10))
pl.plot(np.linspace(-1,-1,200),np.linspace(-1,1,200),'.','r')
pl.plot(np.linspace(1,1,200),np.linspace(-1,1,200),'.','r')
pl.plot(np.linspace(-1,1,200),np.linspace(-1,-1,200),'.','r')
pl.plot(np.linspace(-1,1,200),np.linspace(1,1,200),'.','r')
pl.plot(x,np.sqrt(1-x**2),'r')
pl.plot(x,-np.sqrt(1-x**2),'r')
pl.plot(pogodak_ind_X,pogodak_ind_Y,'.')
pl.plot(ne_pogodak_ind_X, ne_pogodak_ind_Y,'.')
pl.show()


print("Aproksimacija broja pi na ovaj način je",4 * sum(avg_list)/avg)
```

Literatura

[1] Dekking, F. M. - Kraaikamp, C. - Lopuhaä, H. P. - Meester, L. E.: *A Modern Introduction to Probability and Statistics, Understanding Why and How*, Springer, London, 2005.

[2] Krizmanić, Danijel: *Uvod u teoriju vjerojatnosti*, skripta za kolegij Uvod u vjerojatnost i matematičku statistiku

[3] Tijms, Henk: *Understanding probability*, Third edition, Cambridge, Amsterdam, 2012.