TARTU ÜLIKOOL

MATEMAATIKA-INFORMAATIKATEADUSKOND

MATEMAATILISE STATISTIKA INSTITUUT

Ragnar Vent

ÕPIPROGRAMM „SUURTE ARVUDE SEADUS“

Bakalaureusetöö

Juhendaja: prof. Kalev Pärna

TARTU 2013

Sisukord

Sissejuhatus

Käesoleva bakalaureusetöö eesmärgiks on koostada õpiprogramm, mis võimaldaks kasutajal uurida, kuidas avaldub suurte arvude seadus erinevate jaotuste korral. Programmi õpetlikkuse eesmärgil on kaasatud lisaks enamtuntud jaotustele ka mõned suurte arvude seadusega vastuolus olevad jaotused. Näiteks selgub, mis juhtub aritmeetiliste keskmiste jadadega raske sabaga jaotuse korral või kui jaotusel keskväärtus puudub üldse. Samuti saab näha, millist mõju omab suurte arvude seaduse avaldumisel suur üksikväärtus jaotuse sabas.

Töö lõpptulemuseks on interaktiivne Java rakendus, mis tutvustab kasutajale suurte arvude seaduse teoreetilist poolt ning annab võimaluse testida selle seaduse kehtimist praktikas. Õppeprogramm on mõeldud täies mahus iseseisvalt läbimiseks ning sisaldab teadmiste paremaks kinnistumiseks lühikest kontrolltesti. Antud töö juurde kuulub ka CD, mis sisaldab järgnevaid komponente:

- lähtekoodid,
- käivitatav programm,
- töö kirjalik osa.

Programm on koostatud kasutades Java Development Kit 1.7.0 arenduskeskkonda ning Java teeki JFreeChart. Jaotust kirjeldavad graafikud on loodud statistikapaketis R ning programmis esinevad matemaatilised valemid on koostatud kasutades tarkvara LaTeX.

1 Suurte arvude seadus

Tõenäosusteooria üheks keskseks osaks on piirteoreemid, mis käsitlevad juhuslike sündmuste ja juhuslike suuruste teoreetiliste ning katseliste karakteristikute vahelisi seoseid suure katsete arvu puhul [1]. Klassikalised piirteoreemid võib jaotada kahte rühma:

1. Suurte arvude seadused (SAS).
2. Tsentraalsed piirteoreemid.

Esimese rühma teoreemid väidavad, et piisavalt suure arvu juhuslike suuruste korral koondub nende aritmeetiline keskmine mingiks konstandiks. Koonduvust vaadeldakse eelkõige kui koonduvust tõenäosuse järgi (nõrgad suurte arvude seadused) ja koonduvust peaaegu kindlasti (tugevad suurte arvude seadused). Teise rühma teoreemid ütlevad omakorda, et suure arvu sõltumatute juhuslike suuruste summa läheneb piiramatult normaaljaotusele. Antud töös käsitletakse esmajoones suurte arvude seadust ning selle rakendamist interaktiivses õppeprogrammis. [1] [2]

Käesolevas peatükis on põgusalt kirjeldatud suurte arvude seaduse ajalugu [2], esitatud mõningad vajalikud definitsioonid [3] ning toodud näiteid erinevatest SAS-dest [3] [4].

1.1 Ajalugu

Esimene suurte arvude seadus oli edukalt tõestatud Šveitsi matemaatiku Jakob Bernoulli poolt aastal 1713. Tänapäeval tuntakse seda Bernoulli teoreemina. Ta näitas, et sündmuse esinemise suhteline sagedus koondub tõenäosuse järgi katseseeria lõpmatul pikenemisel selle sama sündmuse tõenäosuseks. Tegemist oli üsnagi tülika tõestusega, arvestades, et sel hetkel Tšebõšovi võrratust veel ei tuntud.

Erinevalt Bernoulli poolt küllaltki kitsastel eeldustel tõestatud suurte arvude seadusest andis palju üldisema versiooni SAS-st prantsuse tõenäosusteoreetik Simon-Denis Poisson aastal 1837. Alates Poissoni poolt sõnastatud teoreemist, hakati antud seadust nimetama suurte arvude seaduseks.

Esimene teadlane, kes seadis eesmärgiks luua üldine piirteoreemide teooria, oli Pafnuti Lvovitš Tšebõšov. Tema jõupingutused viisid pealtnäha lihtsa, kuid väga olulise võrratuse tõestuseni, mis seob juhusliku suuruse hälvete tõenäosusi nende dispersioonidega. Kaasajal tuntakse seda Tšebõšovi võrratusena. Osutus, et selle võrratuse baasil on suurte arvude seaduse tõestamine üldisematel tingimustel palju lihtsam kui varem.

Tarvilikud ja piisavad tingimused suurte arvude seaduse jaoks leidis vene tõenäosusteoreetik Andrei Nikolajevitš Kolmogorov aastal 1926. Aastal 1930 tõestas Kolmogorov ka esimese tugeva suurte arvude seaduse.

1.2 Nõrk suurte arvude seadus

Enne nõrga suurte arvude seaduse täpsemat sõnastamist on tarvis defineerida koondumine tõenäosuse järgi.

Definitsioon 1. Öeldakse, et juhuslike suuruste jada Y_n koondub **tõenäosuse järgi** juhuslikuks suuruseks Y , kui iga $\varepsilon > 0$ korral

$$P\{|Y_n - Y| < \varepsilon\} \to 1, \quad n \to \infty.$$

Koondumist tõenäosuse järgi tähistatakse $Y_n \xrightarrow{\mathrm{P}} Y$.

Seadus, mille kohaselt juhuslike suuruste jada koondub tõenäosuse järgi mingiks konstandiks, nimetatakse nõrgaks suurte arvude seaduseks (NSAS). Järgnevalt on välja toodud klassikaline näide NSAS-st, mille korral on tehtud eeldus, et juhuslikud suurused on sõltumatud ja samast jaotusest.

Teoreem 1. Olgu $X_1, X_2, \dots$ sõltumatute sama jaotusega juhuslike suuruste jada, mille korral eksisteerivad keskväärtused $EX_i = a$ ning dispersioonid $DX_i = \sigma^2 < \infty$. Siis aritmeetiliste keskmiste jada koondub tõenäosuse järgi keskväärtuseks a,

$$\frac{S_n}{n} \xrightarrow{\mathrm{P}} a,$$

kus

$$S_n \coloneqq X_1 + \cdots + X_n, \ \ n = 1,2, \dots.$$

Mõnevõrra üldisem versioon nõrgast suurte arvude seadusest oleks aga allpool välja toodud teoreem.

Teoreem 2. Olgu $X_1, X_2, \ldots$ lõpliku dispersiooniga sõltumatud juhuslikud suurused.

Kui

$$\sum_{n=1}^{\infty} \frac{D(X_n)}{n^2} < \infty,$$

siis

$$\frac{S_n - E(S_n)}{n} \xrightarrow{\text{P}} 0.$$

Paneme tähele, et nõrga suurte arvude seaduse korral ei ole tagatud see, et aritmeetilised keskmised $\frac{S_n}{n}$ erinevad küllalt suure n korral konstandist a vähem kui ε võrra. Seetõttu NSAS-i kehtivusel ei saa eeldada iga üksiktrajektoori $\left\{\frac{S_n}{n}\right\}$ koondumist. Küll aga on see võimalik tugeva suurte arvude seaduse puhul.

1.3 Tugev suurte arvude seadus

Tugeva suurte arvude seaduse korrektseks sõnastamiseks oleks esmalt vaja täpsustada, mida tähendab koondumine peaaegu kindlasti.

Definitsioon 2. Öeldakse, et juhuslike suuruste jada $Y_1, Y_2, \ldots$ **koondub peaaegu kindlasti** juhuslikuks suuruseks Y , kui

$$P\left\{\lim_{n\to\infty} Y_n = Y\right\} = 1.$$

Koondumist peaaegu kindlasti tähistatakse $Y_n \to Y\ p.k.$ Koondumine peaaegu kindlasti on tugevam koondumine kui koondumine tõenäosuse järgi.

Teoreemi, mis väidab, et juhuslike suuruste jada koondub peaaegu kindlasti mingiks konstandiks, nimetatakse tugevaks suurte arvude seaduseks (TSAS). Siinkohal esitame ühe klassikalise versiooni tugevast suurte arvude seadusest.

Teoreem 3. Olgu $X_1, X_2, \ldots$ sõltumatute sama jaotusega juhuslike suuruste jada. Siis aritmeetiliste keskmiste jada $\frac{S_n}{n}$ koondub peaaegu kindlasti arvuks a,

$$\frac{S_n}{n} \to a \quad p.k. \ ,$$

parajasti siis, kui leidub lõplik keskväärtus $EX_i = a$.

Üldisema versioonina TSAS-st võib välja tuua Kolmogorovi I teoreemi.

Teoreem 4. (Kolmogorovi I teoreem) Olgu $X_1, X_2, \ldots$ lõpliku dispersiooniga sõltumatud juhuslikud suurused.

Kui

$$\sum_{n=1}^{\infty} \frac{D(X_n)}{n^2} < \infty,$$

siis

$$\frac{S_n - E(S_n)}{n} \to 0 \quad p.k.$$

Õpiprogrammi koostamisel pidasime eeskätt silmas Teoreemi 3. Me illustreerime väidet, et kui genereerida juhuslikke suurusi lõpliku keskväärtusega jaotusest, siis iga artimeetiliste keskmiste trajektoor koondub keskväärtuseks. Järelikult töö tulemusena koostatud õppeprogrammis avaldub tugev suurte arvude seadus praktikas.

2 Programmi suunitlus ning koostamine

Antud peatükis keskendutakse õpiprogrammi „Suurte arvude seadus“ loomise protsessile. Sealjuures antakse üldine ülevaade programmi põhimõttest ning kirjeldatakse, kuidas programmis tekitatakse juhuslikke arve erinevatest jaotustest. Juhuslike suuruste genereerimiseks kasutatavad meetodid põhinevad allikal [5].

2.1 Programmi idee

Õpiprogrammi põhiideeks on illustreerida suurte arvude seadust. Selle simuleerimiseks tuleks Monte-Carlo meetodite abil genereerida juhuslikke suurusi erinevatest statistilistest jaotustest. Programm võimaldab arve tekitada kaheksast erinevast jaotusest, genereerimisel kasutatud meetodid on välja toodud osas 2.2. Pärast iga juhusliku suuruse genereerimist lisatakse see valimisse, leitakse valimi aritmeetiline keskmine ja kantakse see graafikule. Jaotuse keskväärtus, juhul kui see eksisteerib, on samuti graafikul kirjeldatud.

Lisaks visuaalsele poolele, tuleks esitada eelmainitud tulemused ka numbriliselt. Selleks on programmis kasutajale väljastatud erinevad näitajad: viimasena leitud valimi aritmeetiline keskmine, teoreetilise jaotuse keskväärtus ning seni tekitatud suuruste koguarv (valimi maht).

Kasutajale võivad huvi pakkuda ka juba valimis sisalduvad arvud. Näiteks olukorras, kus valimi keskmises on toimunud silmmärgatav muutus, võiks kasutajat huvitada, milline konkreetne suurus niivõrd suurt mõju avaldas. Õpiprogrammis on kõik seni genereeritud arvud välja toodud ning kasutaja töö lihtsuse eesmärgil on kasutatud ka järjekorra numbreid.

2.2 Programmis kasutatud jaotused ning nendest arvude genereerimine

2.2.1 Normaaljaotus

Üks võimalus, kuidas simuleerida normaaljaotusega juhuslikku suurust, on kasutada Javas olemasolevat standardse normaaljaotuse generaatorit, mis põhineb Marsaglia polaarkoordinaatide meetodil [5] [6]. Edasi saame teisendada standardse normaaljaotusega juhuslikku arvu nii, et tekiks keskväärtusega μ ning standardhälbega σ juhuslik suurus $Y \sim N(\mu, \sigma)$, kasutades järgnevat seost:

$$Y = \mu + \sigma Y_0,$$

kus Y_0 on standardse normaaljaotusega juhuslik suurus $Y_0 \sim N(0,1)$.

2.2.2 Ühtlane jaotus

Ühtlasest jaotusest juhusliku arvu simuleerimisel võime samuti kasutada Javas juba eksisteerivat generaatorit. See generaator tugineb lineaarsele kongruentsele meetodile ning ta annab arvu ühtlasest jaotusest $U(0,1)$ [5] [6]. Selleks, et saada juhuslik suurus ühtlasest jaotusest $X \sim U(a, b)$, tuleb kasutada seost

$$X = a + (b - a)X_0,$$

kus X_0 on juhuslik suurus ühtlasest jaotusest $U(0,1)$.

2.2.3 Binoomjaotus

Juhusliku suuruse $Y \sim B(n, p)$ simuleerimisel saab kasutada teadaolevaid binoomjaotuse omadusi. Binoomjaotusega arv saadakse n sõltumatu üksikkatse tulemusena, kus igal katsel on edu tõenäosus p. Täpsemalt,

$$Y = \sum_{i=1}^{n} Z_i \sim B(n, p),$$

kus Z_i on sõltumatud Bernoulli jaotusega $Be(p)$ juhuslikud suurused [5]. Nende teadmiste põhjal koostatud algoritm näeb välja järgnev:

1. Genereerida n juhuslikku suurust Bernoulli jaotusest.
    1.1. Selleks genereerida juhuslikud arvud $X_i \sim U(0,1)$, kus $i = 1 \dots n$.
    1.2. Kui $X_i < p$, siis $Z_i = 1$, vastasel juhul $Z_i = 0$.
2. Leida summa $Y = \sum_{i=1}^{n} Z_i$.

2.2.4 Poissoni jaotus

Juhusliku suuruse $Y \sim Po(\lambda)$ modelleerimisel tuleb leida vähim k väärtus, mille korral kehtib

$$\prod_{i=1}^{k} X_i < e^{-\lambda},$$

kus $X_i \sim U(0,1)$. Tingimuse täidetuse korral $Y = k - 1$. [5]

2.2.5 Eksponentjaotus

Eksponentjaotusest juhuslike arvude genereerimiseks võib kasutada jaotusfunktsiooni pööramise meetodit. Eksponentjaotuse jaotusfunktsioon avaldub kujul

$$F(x) = \begin{cases} 1 - e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0. \end{cases}$$

Jaotusfunktsiooni pööramisel saame tulemuseks seose

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

kus X on juhuslik arv ühtlasest jaotusest $U(0,1)$ ning Y eksponentjaotusest $Exp(\lambda)$. Kuid, kuna suurus $1 - X$ on samuti jaotusega $U(0,1)$, siis võime eelnevat võrratust lihtsustada ning viia alternatiivsele kujule:

$$Y = -\frac{1}{\lambda} \ln(X).$$

2.2.6 Suure üksikväärtusega jaotus

Lisaks on programmi kaasatud jaotus, kus esineb teatud suur üksikväärtus, mille tõenäosus on väga väike. Selline jaotus on esitatud tabelis 1.

Tabel 1. *Suure üksikväärtusega jaotus*

Y_i	1	5	10	50	100	500	1000	2000	4000	100000
p_i	0.4	0.25	0.18	0.1	0.06	0.008	0.001	0.0008	0.00019	0.00001

Antud jaotuse keskväärtus on $EX = 22.81$ ja dispersioon $DX = 109594.4$. Juhuslikke arve Y_i saab antud jaotusest tekitada järgneva eeskirja järgi:

1. Jagada vahemik $(0,1)$ kümneks osaks, mille pikkused võrduvad tabelis 1 toodud tõenäosustega.
2. Genereerida arv $X \sim U(0,1)$ ning vaadata, millisesse vahemikku see arv sattus.
3. Tagastada sellele vahemikule vastav juhusliku suuruse väärtus (üks arvudest 1,5,...,100000).

2.2.7 Cauchy jaotus

Sarnaselt eksponentjaotusest arvude genereerimisele on ka Cauchy jaotuse korral võimalik rakendada jaotusfunktsiooni pööramise meetodit. Jaotusfunktsioon avaldub siin kujul

$$F(x) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{x - x_0}{\gamma}\right), -\infty < x < \infty,$$

kus $\gamma > 0$ on skaalaparameeter ning $-\infty < x_0 < \infty$ on asukohaparameeter [7]. Rakendades jaotusfunktsiooni pööramise meetodit, saame

$$Y = \gamma\left(\tan\left(\pi\left(X - \frac{1}{2}\right)\right)\right) + x_0,$$

kus $Y \sim C(x_0, \gamma)$ ning $X \sim U(0,1)$.

2.2.8 Pareto jaotus

Raske sabaga jaotuse simuleerimiseks kaasati programmi Pareto jaotus, mille jaotusfunktsioon avaldub kujul

$$F(x) = 1 - \frac{1}{x^{\alpha}}\ , 1 < x < \infty,$$

kus $\alpha > 0$ on Pareto jaotuse parameeter [7]. Selle põhjal saame modelleerida Pareto jaotust jaotusfunktsiooni pööramise teel:

$$Y = \frac{1}{(1 - X)^{\frac{1}{\alpha}}},$$

kus $Y \sim Pa(\alpha)$ ning $X \sim U(0,1)$. Kasutades teadmist, et nii $(1 - X) \sim U(0,1)$, kui ka $X \sim U(0,1)$, võime eelneva seose asemel kasutada järgmist seost:

$$Y = \frac{1}{X^{\frac{1}{\alpha}}}.$$

3 Õpiprogrammi kasutusjuhend

3.1 Programmi installeerimine ja käivitamine

3.1.1 Vajaliku tarkvara installeerimine

Käesoleva õpiprogrammi kasutamiseks peab kasutajal arvutisse olema installeeritud Java Runtime Environment. Juhul kui vastav tarkvara puudub, saab viimase versiooni alla laadida järgnevalt aadressilt: http://www.java.com/en/download/index.jsp .

3.1.2 Programmi käivitamine

Tööga on kaasas CD, mis sisaldab kompileeritud programmi kujul *sas_2013.jar*. Selle käivitamiseks on kasutajal mitmeid võimalusi. Kõige lihtsam viis oleks kaks korda programmifailile peale vajutada. Alternatiivselt võib kasutaja operatsioonisüsteemi käsuliinis liikuda kausta, kus asub programmi jooksutamiseks vajalik fail ning sisestada käsurida

java –jar sas_2013.jar .

Tulemusena peaks avanema aken nagu on näha joonisel 1.

Joonis 1. *Programmi avaleht*

3.2 Programmi kasutamine

3.2.1 Avalehelt navigeerimine

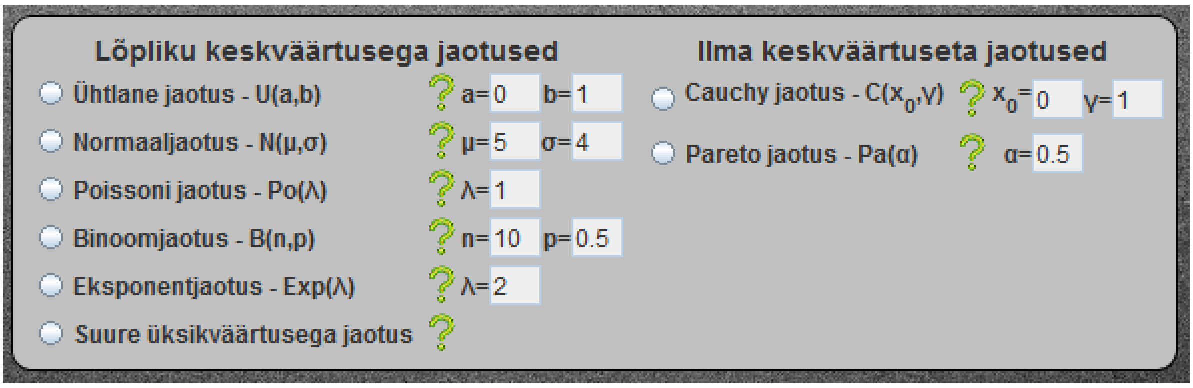
Programmi käivitamisel tekkinud avalehel on kasutajal võimalik valida nelja erineva valiku vahel:

1. Autorid - viib kasutaja lehele, kus on nimetatud programmi koostajad.
2. Teooria - suunab kasutaja lehele, kus on käsitletud suurte arvude seaduse teoreetilist külge. Teooria osa lõpus asub kolmest küsimusest koosnev lühike test, kontrollimaks, kas kasutaja on teema lõplikult omandanud.
3. Juhend - suunab kasutaja lehele, mis annab programmi kasutamiseks vajalikud instruktsioonid.
4. Edasi - viib kasutaja programmi põhilehele, kus toimub programmi peamine töö.

Avalehelt teistele lehtedele navigeerides on alati võimalik tagasi avalehele minna, kasutades selleks nuppu „Avalehele".

3.2.2 Katsetingimuste määramine ja genereerimise alustamine

Esimese sammuna tuleks valida sobiv jaotus, millest juhuslikke arve genereerida. Kokku on võimalik valida kaheksa erineva jaotuse vahel, mis on grupeeritud kahte rühma: lõpliku keskväärtusega jaotused ja ilma keskväärtuseta jaotused. Kasutatud jaotusi ja nende grupeerimist on näha joonisel 2.



Joonis 2. *Jaotuste paneel*

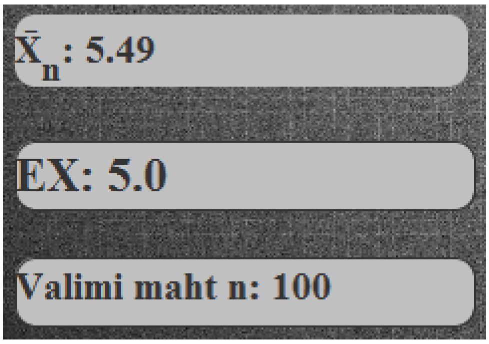
Jaotuse vahetamine toob kaasa seniste tulemuste nullimise.

Kui kasutajale sobiv jaotus on valitud, siis järgmisena tuleks valida jaotuse parameetrid. Kokku on võimalik valida parameetreid seitsmel erineval jaotusel, suure ükskväärtusega jaotuse puhul antud valik puudub. Programmi käivitades määratakse parameetritele ka vaikeväärtused. Erinevaid parameetreid, mida on võimalik sisestada, saab näha joonisel 2. Jaotuse parameetrite muutmine ning seejärel numbrite genereerimise alustamine toob endaga kaasa seniste tulemuste nullimise.

Uusi juhuslikke arve genereeritakse vaikimisi ühekaupa. Selleks, et lasta arve jooksvalt genereerida, tuleb valida genereerimise viisiks „Arvuti genereerib". Lisaks saab tööprotsessi kiirust korrigeerida sõltuvalt vajadusest, muutes ühel sammul genereeritavate arvude kogust. Ühe korraga (ühel sammul) on maksimaalselt võimalik genereerida 1 000 arvu.

Joonis 3. *Genereerimise viisi ja ühel sammul genereeritavate arvude koguse paneel*

Juhul, kui kõik eelnevalt esitatud tingimused on määratud, siis saab alustada programmi tööprotsessi, vajutades noolega tähistatud genereerimise nupule (vt. joonis 3). Tulemusena tekib ekraanile genereeritud arvude aritmeetiliste keskmiste trajektoor. Eraldi on välja toodud viimasena leitud valimi artimeetiline keskmine, valitud jaotuse keskväärtus (sõltuvalt sisestatud parameetritest) ning genereeritud arvude kogus ehk valimi maht (vt. joonis 4).

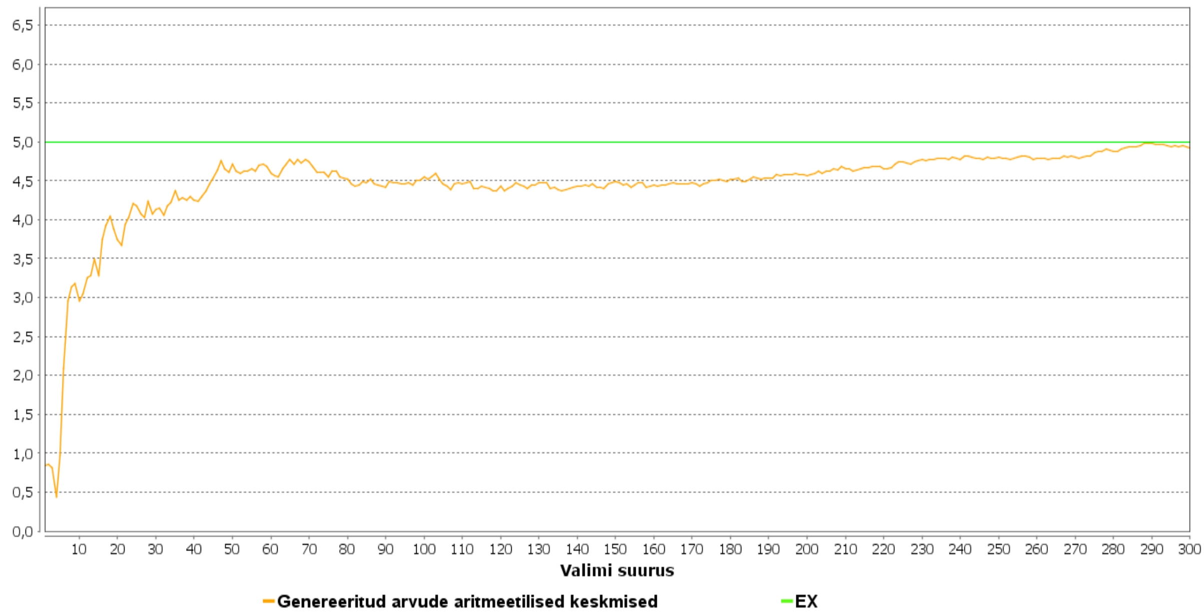


Joonis 4. *Näide Poissoni jaotusest Po(5) genereeritud 100 arvu aritmeetilise keskmise kohta*

Kõiki genereeritud arve saab näha, kui vajutada nupule „Genereeritud arvud“. Avanevas dialoogiaknas tuleb täpsustada genereeritud arvude indeksite vahemikku ning seejärel vajutada nupule „Leia arvud“. Maksimaalne indeksite vahemiku pikkus on 500 000.

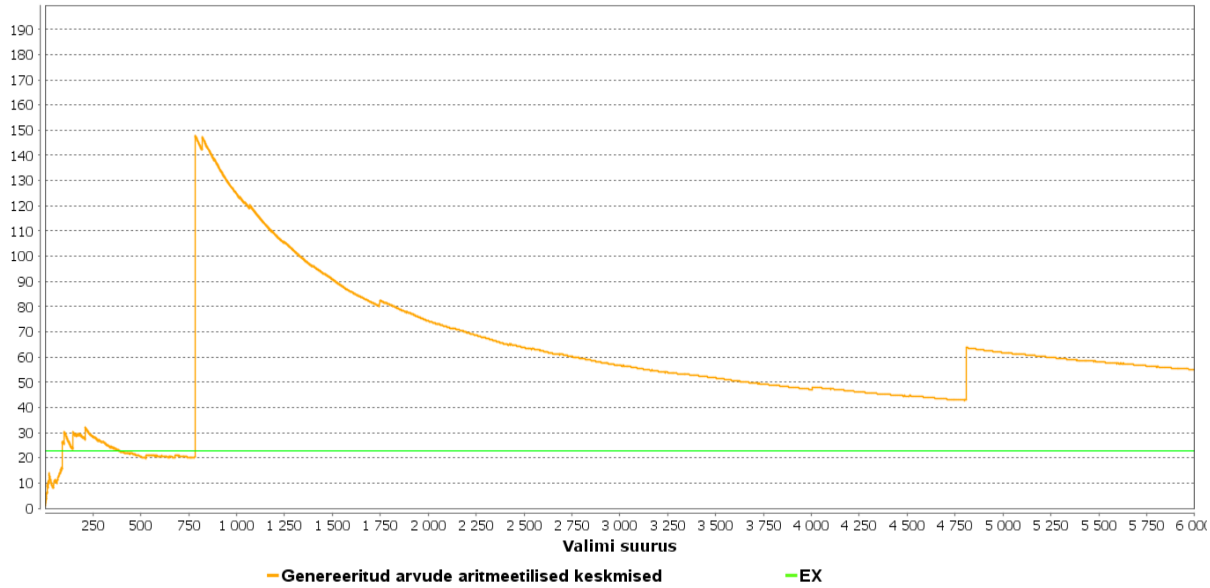
3.2.3 Tulemuse tõlgendus

Programmi töö tulemused varieeruvad sõltuvalt katsetingimustest. Kui genereerida juhuslikke suurusi mõnest lõpliku keskväärtusega jaotusest, siis on alatest mingist hetkest näha valimi keskmise koondumist vastava teoreetilise jaotuse keskväärtuseks. Joonisel 5 on esitatud näide normaaljaotusega $N(5,4)$ juhuslike suuruste aritmeetiliste keskmiste koondumisest. Antud näite korral on jaotuse keskväärtuseks $\mu = 5$, seega valimi suurenedes läheneb valimi keskmine arvule 5, mis on ka joonisel märgatav.



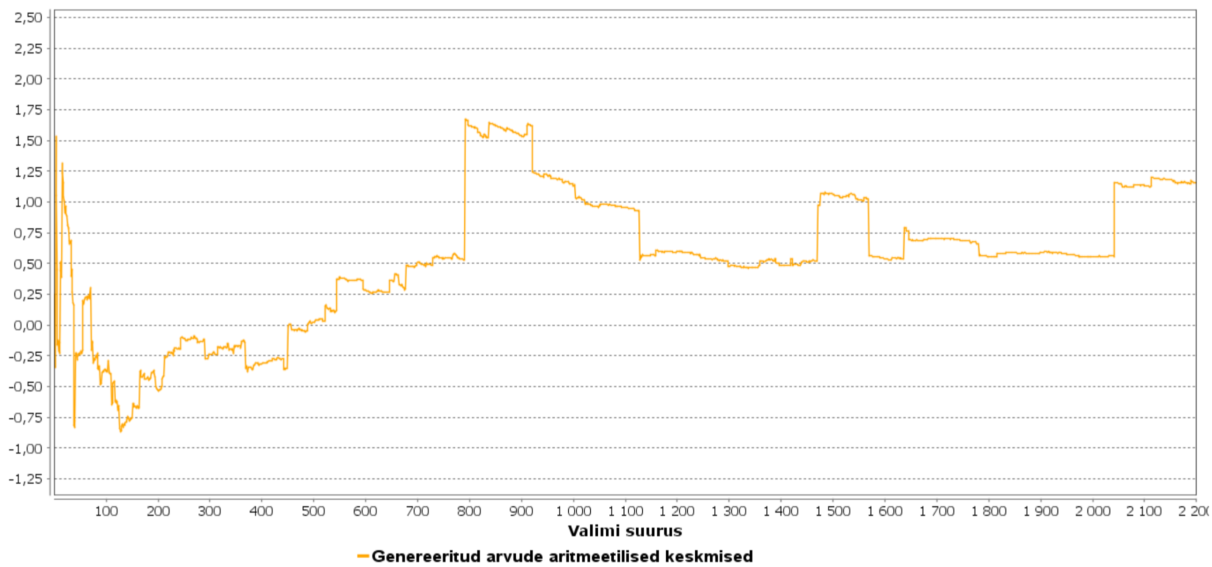
Joonis 5. *Näide aritmeetilise keskmise trajektoorist normaaljaotuse N(5,4) korral*

Lõpliku keskväärtusega jaotustest võib eraldi huvi pakkuda juht, kus on võimalik ühe suure üksikväärtuse esinemine väikese tõenäosusega. Joonisel 6 on esitatud üks aritmeetiliste keskmiste trajektoor sellisest suure üksikväärtusega jaotusest genereeritud arvude puhul. Võib näha trajektoori kahte suuremat hüpet, mis viitab sellele, et suur üksikväärtus esines kaks korda. Graafikust võib järeldada, et isegi siis, kui valimi keskmine on juba näiliselt koondunud, võib suure ükskväärtuse esinemine selle paigast lüüa. Märgata on ka suure üksikväärtuse mõju kahanemist valimi mahu suurenedes.



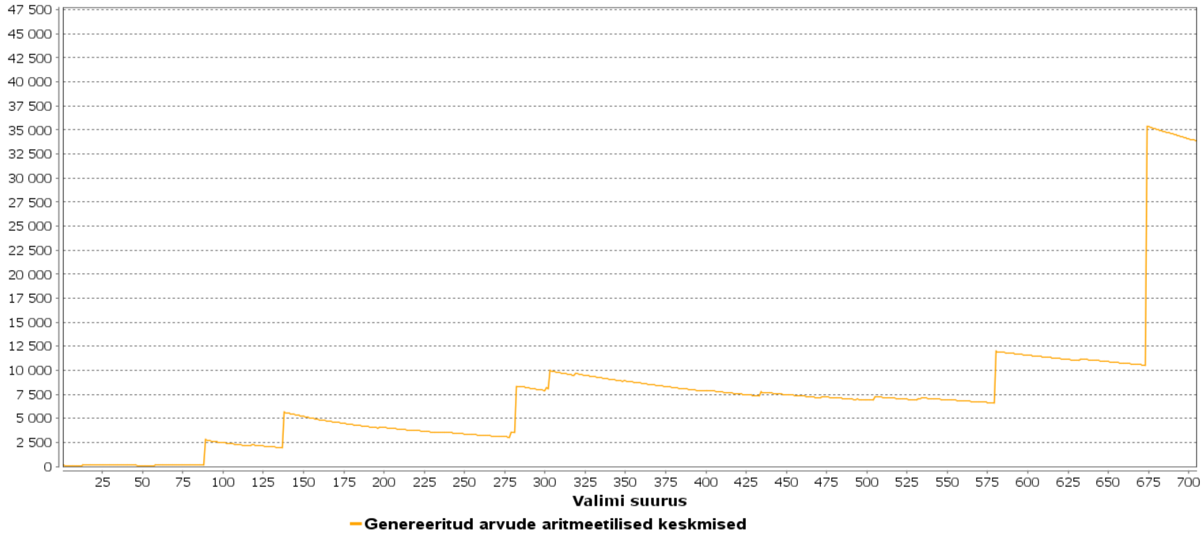
Joonis 6. *Näide aritmeetilise keskmise trajektoorist suure üksikväärtusega jaotuse korral*

Liigume edasi ilma keskväärtuseta jaotuste juurde. Joonisel 7 on esitatud üks versioon standardsest Cauchy jaotusest $C(0,1)$ genereeritud arvude keskmise trajektoori kohta. Võib täheldada, et valimi aritmeetiline keskmine ei lähene ühelegi kindlale arvule ja liikumine toimub kohati väga drastiliselt mõlemas suunas. Kuna Cauchy jaotusel keskväärtus puudub, siis mingit koondumist ei toimu ja antud juhul suurte arvude seaduse toime ei ilmne.



Joonis 7. *Näide aritmeetilise keskmise trajektoorist standardse Cauchy jaotuse korral*

Pareto jaotuse keskväärtus on lõpmatu, kui parameeter α rahuldab tingimust $0 < \alpha \leq 1$. Joonisel 8 on toodud näide Pareto jaotusest genereeritud arvude keskmise trajektoori kohta, kui $\alpha = 0.4$. Esile tõusevad järjestikused hüpped, mis suurendavad valimi keskmist silmmärgatavalt. Siinkohal tuleb märkida, et mida väiksem on parameeter α, seda raskem on Pareto jaotuse saba ning seda intensiivsemad ja suuremad on hüpped.



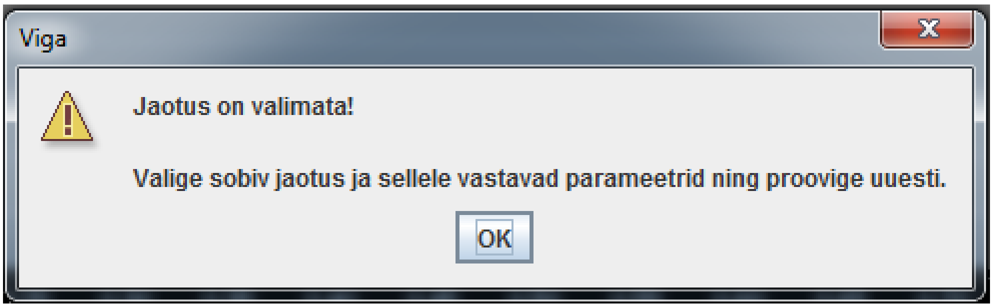
Joonis 8. *Näide aritmeetilise keskmise trajektoorist Pareto jaotuse Pa(0.4) korral*

3.2.4 Informatsiooni leidmine jaotuste kohta

Programmis on võimalik leida üldist informatsiooni kasutatud jaotuste ning neid iseloomustavate karakteristikute kohta. Selle nägemiseks tuleb vajutada vastava jaotuse kõrval asuvale küsimärgile (vt. joonis 2) . Tulemusena avaneb uus dialoogiaken, kus on kasutajale välja toodud keskväärtuse ja dispersiooni leidmise valemid, jaotust iseloomustavad parameetrid ja funktsioonid ning illustreeriv graafik.

3.2.5 Võimalikud veateated

Korrektse ja sujuva töö tagamiseks on programmis defineeritud erinevad veateated, mis ilmuvad andmete puudumisel või nende valel sisestamisel.



Joonis 9. *Valimata jaotuse puhul esinev veateade*

Kokku võib esineda kolme erinevat tüüpi veateadet:

1. Valimata jaotus – kuvatakse siis, kui arvude genereerimise alustamisel puudub valitud jaotus.
2. Vormistusviga – kuvatakse siis, kui parameetri väli sisaldab mittearvulisi sümboleid või valesti vormistatud arve.
3. Parameeter piiridest väljas – kuvatakse, kui jaotuse parameetri sisendiks antud arv ei vasta tema kriteeriumitele. Näiteks binoomjaotuse parameetrit p (tõenäosus) ei ole võimalik valida väljaspool lõiku [0,1]. Erandina võib välja tuua Pareto jaotuse parameetri α, mida on üldjuhul võimalik valida vahemikust $(0,\infty)$, kuid käesolevas programmis on selleks poollõik (0,1], tagamaks Pareto jaotuse korral lõpmatu keskväärtuse.

Kokkuvõte

Bakalaureusetöö tulemusena valmis õpiprogramm, mis tutvustab ja illustreerib suurte arvude seadust. Programm genereerib jooksvalt mõnest jaotusest juhuslikke suurusi ja iga uue arvu genereerimise järel leiab valimi aritmeetilise keskmise ning lisab selle graafikule. Kasutaja saab paralleelselt näha viimasena leitud valimi aritmeetilist keskmist, jaotuse keskväärtust, ning valimi mahtu.

Loodud rakenduse käivitamisel on kasutajal esmalt võimalus tutvuda suurte arvude seaduse teoreetilise poolega ning vajalike juhistega programmi kasutamiseks. Põhilehele liikudes peaks kasutaja esimese sammuna valima jaotuse, millest juhuslikke suurusi genereerida. Edasi tuleks täpsustada valitud jaotuse parameetrid ning genereeritavate arvude kogus. Viimase sammuna on kasutajal võimalik valida, kas genereerida arve ühekordselt või lasta programmil teha seda jooksvalt. Kui kõik eelnevalt mainitud sammud on läbitud, saab alustada tööprotsessi, vajutades vastavale genereerimisnupule.

Lisaks sisaldab programm iga kasutatud jaotuse kohta lühikirjeldust.

Töö tekstiline osa sisaldab suurte arvude seaduse täpsemat kirjeldust, programmi ülesehitust ning kasutajajuhendit.

Õpiprogramm on koostatud JDK 1.7.0 arenduskeskkonnas.

Learning program “The Law of Large Numbers“

Bachelor Thesis

Ragnar Vent

Summary

The goal of this thesis is to create an interactive program that introduces and demonstrates an essential part of probability theory – the law of large numbers. The idea is to generate random numbers from a statistical distribution and see if the sample mean converges to the expected value of the random variable. The program itself allows the user to generate numbers from eight different distributions. After each number is generated, a new sample mean is calculated and placed onto the graph. The current sample mean and the expected value of the random variable are both displayed for the user.

Upon starting the program, the user is given a brief overview of what the program is about and a chance to look at the instructions if necessary. The first step would be choosing a suitable distribution from which the random numbers will be generated. After that, appropriate distribution parameters have to be chosen. If the parameters are not chosen from the required range or are missing altogether, a warning message will appear. The next step involves choosing the amount of numbers to generate and whether to generate the numbers once or let the program do it continuously. Once all the steps are completed, the user can start generating numbers by clicking the generate button.

In addition, a short overview of every used distribution is available in the program.

The textual part of the bachelor thesis includes the theoretical approach to the law of large numbers, program overview and a user guide for the created program.

Program was created in JDK 1.7.0 development environment.

Viited

1. Jõgi, A., 2000. *Tõenäosusteooria II osa*. Tallinn: TTÜ kirjastus.
2. Tiit, E., 1969. *Tõenäosusteooria II*. Tartu: Tartu Riiklik Ülikool.
3. Pärna, K., 2013. *Tõenäosusteooria algkursus*. Tartu: TÜ Kirjastus.
4. Lember, J., 2005. *Tõenäosusteooria II Loengukonspekt ja ülesanded*.
5. Kollo, T., 2004. *Monte Carlo meetodid*. Tartu: TÜ Kirjastus.
6. Oracle, 2013. *Java™ Platform, Standard Edition 7 API Specification*. http://docs.oracle.com/javase/7/docs/api/ [Viimati vaadatud 05.05.2013]
7. Balakrishnan, N., Nevzorov, V., 2003. *A Primer on Statistical Distributions*. Hoboken: A John Wiley & Sons, Inc., Publication.

Lisad

Lisa A. Programmikood erinevatest jaotustest arvude genereerimiseks

```
import java.util.Random;

public class NumbGen extends Random{

	static final Double[] customDistrP = {0.4,0.65,0.83,0.93,0.99,0.998,0.999,0.9998,0.99999,1.0};
	static final int[] customDistrX = {1,5,10,50,100,500,1000,2000,4000,100000};
	double L,U;
	int k,x;

	//Normaaljaotusest arvu genereerimine
	public double nextNormal(double mean,double std){
		return (mean+std*nextGaussian());
	}

	//Poissoni jaotusest arvu genereerimine
	public double nextPoisson(double lambda){

		L = Math.exp(-lambda);
		U = nextDouble();
		k = 1;

		while(U > L){
			k++;
		   U *= nextDouble();
		}
		return (k-1);
	}

	//Ühtlasest jaotusest arvu genereerimine
	public double nextUniform(double a,double b){
		return (a+(b-a)*nextDouble());
	}

	//Binoomjaotusest arvu genereerimine
	public double nextBinomial(double n,double p){
		x = 0;
		for(int i = 0; i < n; i++) {
			if(nextDouble() < p)
				x++;
		}
		return (x);
	}

	//Eksponentjaotusest arvu genereerimine
	public double nextExponent(double lambda){
		return (Math.log(nextDouble())/(-lambda));
	}

	//Cauchy jaotusest arvu genereerimine
	public double nextCauchy(double x0,double gamma){
		return (gamma*(Math.tan(Math.PI*(nextDouble()-0.5)))+x0);
```

```
	}

	//Pareto jaotusest arvu genereerimine
	public double nextPareto(double alpha){
		return (Math.pow(nextDouble(), -1/alpha));
	}

	//Suure üksikväärtusega jaotusest arvu genereerimine
	public double nextLotoDistr(){

		U = nextDouble();

		for(int i=0; i<customDistrP.length; i++){
			if(U<customDistrP[i]){
				return (customDistrX[i]);
			}
		}
		return (0.0);
	}
}
```

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina Ragnar Vent (sünnikuupäev: 17.06.1991)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
Õpiprogramm "Suurte arvude seadus",

mille juhendaja on Kalev Pärna,

1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus/Tallinnas/Narvas/Pärnus/Viljandis, **06.05.2013**