

MCM, praktikum 1

Kuna enamusel tudengitest on põhiteadmised R keskkonna kasutamisest olemas, siis praktikumides lihtsalt tutvustatakse põgusalt vajalikke käsk/R keele vahendeid. Neil tudengeid, kel R varasema kasutamise kogemus kas puudub või on väga vähene, on tungivalt soovitatav võimalikult kiiresti enda algteadmisi täiendada, kasutades selleks näiteks kursuse Moodle kodulehel viidatud materjale. Samuti võib alati küsida abi ja selgitusi nii kaastudengite kui praktikumijuhendaja käest.

Funktsioonid, vektorid, maatriksid

Funktsiooni definitsiooni kuju R-s on järgmine

```
f_nimi <- function(arg1,arg2,...,argn=vaikeväärtus){
  käsk1
  käsk2
  jne
  return(tulemus)
}
```

Näited:

```
ruut <- function(x){
  return(x^2)
}
lineaarne <- function(x,a=7,b=0){
  tulemus <- a*x+b
  return(tulemus)
}
```

Harjutus 1. Defineerige funktsioon `lineaarne` ning kõigepealt katsuge ära arvata tulemus ja siis kontrollige seda järgnevate programmilõikude korral

```
lineaarne(3)
lineaarne(1:5)
lineaarne(1,2)

x <- 3;a <- 2;b <- 1
lineaarne(a,b,x)
lineaarne(a=3,b=2,x=0)
lineaarne(a=a,b=b,x=x)
```

R-is on programmikoodi töökiiruse huvides hea kasutada tsüklite asemel vektoroperatsioone. Selleks on vaja teada mõningaid võimalusi vektorite tekitamiseks ja kasutamiseks.

```
x <- 1:10 # Teeb vektori, mis sisaldab väärtuseid 1, 2, 3, ..., 10
y <- rep(NA,10) # Teeb 10-elementi pikkuse vektori, mille elementideks on NA (puuduv väärtus)
z <- c(3,1,2,7,1,5,1,4,2,10) #Tekitab vektori toodud väärtustest
print(x[x>5]) # Trükitab välja need vektori x elemendid, mis on suuremad kui 5
y[x>5] <- 20 # y-tunnuse väärtused nende indeksite korral, mille puhul keltib x>5, muudetakse 20-ks
print(y)
y[x<=5] <- x[x<=5] # y-tunnuse väärtused nende indeksite korral, mille puhul kehtib x<=5,
# muudetakse võrdseks vastavate x väärtustega
print(y)
y[2] <- 102 # y-tunnuse (vektori y) teine element muudetakse ära 102-ks.
y2 <- y %% 3 # Leiab vektori, mis vastab y komponentide kolmega jagamise jääkidele.
M <- matrix(z,ncol=2) # moodustab z väärtustest maatriksi, täites selle veergude kaupa
```

```
M <- matrix(z,ncol=2,byrow = T) # moodustab z väärtustest matriksi, täites selle ridade kaupa
M[2,] # matriksi teine rida
M[,1]+M[,2] # matriksi M esimese ja teise veeruvektori summa
sum(z<4) # leab vektori z arvust 4 väiksemate elementide arvu
```

Harjutus 2 Defineerige funktsioon `loendaja`, mis etteantud kaheveerulise matriksi `M` ning kahe muutuja funktsiooni `f` korral väljastab `M` ridadele vastavate tasandi punktide (x, y) arvu, mille korral kehtib võrratus $f(x, y) < 0$. Kontrolliks: funktsiooni $f(x, y) = 2 \cdot x - y$ korral peaks `R` näiteandmestiku `cars` puhul käsu `loendaja(cars, f)` tulemuseks tulema 33. Tehke funktsiooni `loendaja` abil kindlaks, kui palju andmestikule `cars` vastavaid xy -tasandi punkte jäävad keskpunktiga $(15, 20)$ ja raadiusega 5 ringi sisse.

Praktikumi põhiülesanne

Lineaarse kongruentse generaatori algoritm:

- Valime täisarvu $m > 1$ ja mittenegatiivsed täisarvud $a, b, x_0 < m$
- Leiame $i = 0, 1, \dots$ korral pseudojuhuslikud täisarvud arvutustega $x_{i+1} = ax_i + b \pmod m$
- Leiame pseudojuhuslikud arvud kujul $u_{i+1} = \frac{x_{i+1}}{m}$

Kirjutage funktsioon `LKG`, mis genereerib lineaarset kongruentset generaatorit kasutades pseudojuhuslikke arve. Generaator peaks väljastama soovitud arvu pseudojuhuslikke suuruseid. Näiteks kui pöördume tema poole käsuga `LKG(10)`, siis peaksime saama vastuseks 10st arvust koosneva vektori. Lisaks peab olema võimalik ette anda kasutatav seeme (ehk x_0 väärtus) ning muuta kasutatava generaatori parameetreid a, b, m , mille vaikeväärtusteks võtta `seeme=1, a=511, b=17, m=10000`.

Näiteks käsu `LKG(5)` tulemuseks peaks tulema

```
## [1] 0.0528 0.9825 0.0592 0.2529 0.2336
```

Harjutus 3 Tekitage funktsioniga `LKG` 300 juhuarvu, moodustage neist kaheveeruline matriks (ridade kaupa) ja kujutage saadud punktid tasandil.

Harjutus 4 Tekitage umbes 50 aastat tagasi populaarse `RANDU` generaatoriga (vastab valikutele $a = 65539$, $b = 0$, $m = 2^{31}$) seemnest $x_0 = 11$ lähtuvalt 30000 väärtust, moodustage neist kolmeveeruline matriks (ridade kaupa) ja uurige järjestikku genereeritud pseudojuhuslike arvude kolmikute paiknemist ruumis, kasutades paketi `rgl` käsku `plot3d`. Pöörake graafikaakent ning vaadake, kas punktide paiknemine tundub juhuslik. Pildi saab salvestada käsuga `snapshot3d`.

Seejärel tehke kindlaks, kui palju erinevaid väärtuseid omandab suurus $W = Z - 6Y + 9X$, kus X, Y, Z vastavad tekitatud matriksi esimesele, teisele ja kolmandale tulpale. Tulemuste üle järgi mõtlemisel on abiks teadmine, et kui tekitada kolmemõõtmelises ruumis punkte sõltumatute $U(0, 1)$ juhuslike arvude abil, siis peaks W omandama väärtuseid vahemikus $(-6, 10)$ ning näiteks tõenäosus, et $2 < W < 3$ on suurem kui 0.1.

Kontrolliks: W väärtuste tabel

W															
	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9
	73	255	473	632	878	1053	1173	1150	1071	1010	827	604	436	256	109

Kodutöö 1

Esitamine Moodle kaudu, tähtaeg 11.09.2022. Esitada tuleb ka `.R` või `.Rmd` fail ning lisaks kas pdf või HTML file, kus on näha kood koos tuletuskäikude, tulemuste ja kommentaaridega.

- **Ülesanne 1** (1 punkt) Vaatleme juhuslikku suurust $Y = (X + 2)^2$, kus X on ühtlase jaotusega lõigus $[0, 1]$. Suurte arvude seaduse kohaselt peaks ühtlasele jaotusele vastavate juhuslike arvude $x_1, x_2, \dots$ korral arvude $y_i = (x_i + 2)^2$, $i = 1, 2, \dots$ aritmeetiline keskmine

$$\frac{y_1 + y_2 + \dots + y_n}{n}$$

läheneb n kasvades juhusliku suuruse Y keskvaärtusele. Leidke ligikaudselt juhusliku suuruse Y keskvaärtus, kasutades juhuslike arvude asemel 10000 RANDU generaatoriga (enda valitud seemnest lähtuvalt) tekitatud pseudojuhuslikku arvu. Arvutage ka täpne keskvaärtus EY (selgitades, mida selleks teha tuleb) ning leidke eelnevalt saadud ligikaudse keskvaärtuse viga. NB! täpse keskvaärtuse leidmiseks tuleb leida keskvaärtusele vastava integraali väärtus valemi kujul ning alles seejärel välja arvutada valemile vastav väärtus, numbrilise integreerimise vahendid (nt käsk `integrate`) ei sobi siin kursuses täpse väärtuse leidmiseks!