Data-analyysi R-ohjelmistolla, kevät 2015

Viikko 2

Ensimmäisen viikon tehtävissä harjoiteltiin matriisien ja vektorien käsittelyä, sekä niiden alkoiden valintaa. Nyt näitä taitoja sovelletaan oikeiden aineistojen käsittelyyn.

Tämän viikon tärkeimpiä teemoja ovat datan alustava tarkastelu kuvien ja tunnuslukujen avulla, simulaatioiden käyttö, tallennetun aineiston käyttöönotto ja tietotyypit.

1 R:n tietotyypit

1.1 Muuttujan tyyppi

R:n tietotyypeistä viime viikolla esiteltiin numeeriset muuttujat ja totuusarvot. Muuttujan tyyppiä voi tarkastella `class()`-funktiolla [1].

Esimerkki 1.

```
> a <- 5
> b <- TRUE
> c <- c(3,76,43,5)
> d <- c("Mies", "Nainen")
> e <- as.factor(d)
>
> class(a)
[1] "numeric"
> class(b)
[1] "logical"
> class(c)
[1] "numeric"
> class(d)
[1] "character"
> class(e)
[1] "factor"
>
```

Muuttujien tyyppiä voi testata `is-` ja muuttaa, `as`-alkuisilla funktioilla. Päätteeksi funktioon laitetaan pisteen jälkeen halutun tietotyypin nimi.

Esimerkki 2.

```
> a <- c("4", "47", "-7")
> a
[1] "4"  "47" "-7"
>
> is.numeric(a)
[1] FALSE
```

[1] Itse asiassa `class()` kertoo R:n objektin luokan, mutta yksinkertaisilla objekteilla, kuten vektoreilla luokka on sama kuin niiden tietotyyppi, ellei muuta ole asetettu.

```
> is.character(a)
[1] TRUE
>
> a <- as.numeric(a)
> a
[1]  4 47 -7
>
> is.numeric(a)
[1] TRUE
> is.character(a)
[1] FALSE
```

1.2 Merkkijonot

Numeeristen muuttujien ja totuusarvojen lisäksi R:ssä on käytössä oma tietotyyppi merkkijonoille. Merkkijonot kirjoitetaan joko yksin- tai kaksinkertaisten lainausmerkkien sisään, ja niitä voidaan sijoittaa muuttujaan aivan kuten numeroita ja totuusarvojakin. Yksittäisille merkeille ei ole omaa tietotyyppiä, vaan ne tallennettaan merkkijonoina, joiden pituus on yksi. Merkkijonoja voi myös sijoittaa peräkkäin vektoriin.

Esimerkki 3.

```
> "R"
[1] "R"
> merkkijono <-'data-analyysi'
> merkkijono
[1] "data-analyysi"
> ohjaajat <- c('Ville', 'Henkka', 'Oskari')
> ohjaajat
[1] "Ville"  "Henkka" "Oskari"
> class(ohjaajat)
[1] "character"
>
```

1.3 Faktorit

Faktori on R:n tietotyyppi, joka on tarkoitettu luokitteluasteikollisten muuttujien tallentamiseen ja käsittelyyn. Jos meillä on esimerkiksi muuttuja johon on tallennettu vastaajan sukupuoli, voimme muuttaa sen faktoriksi funktiolla `as.factor()`.

Esimerkki 4.

```
> sukupuoli <- c("N","N","M","N","M","N")
> sukupuoli
[1] "N" "N" "M" "N" "M" "N"
> str(sukupuoli)
 chr [1:6] "N" "N" "M" "N" "M" "N"
>
> sukupuoli <- as.factor(sukupuoli)
> sukupuoli
```

```
[1] N N M N M N
Levels: M N
> str(sukupuoli)
 Factor w/ 2 levels "M","N": 2 2 1 2 1 2
```

Tarkasteltaessa faktoriksi muutettua muuttujaa `str()`-funktiolla huomataan että sen arvot on koodattu uudelleen luvuiksi 1 ja 2; alkuperäiset arvot M ja N ovat faktorin tasoja. Tasoja on yhtä monta kuin alkuperäisessä muuttuja saa erilaisia arvoja. Faktoreista on hyötyä tehtäessä analyysejä osa-aineistoittain, esimerkiksi juuri sukupuolen mukaan jaoteltuna.

2 Taulukko

Vektoreihin ja matriiseihin voi tallentaa vain yhden tietotyypin alkoita kerrallaan. Yritettäessä tallentaa tallentaa useaa eri tietotyyppiä olevia alkioita samaan vektoriin tai matriisiin, R muuttaa koko vektorin tai matriisin "yleisempää" muotoa olevaan tietotyyppiin. Esimerkiksi liitettäessä merkkijonoja ja numeroita samaan vektoriin R muuttaa numerot merkkijonoiksi.

Esimerkki 5.

```
> a <- c(ohjaajat, 5)
> a
[1] "Ville"  "Henkka" "Oskari" "5"
> class(a)
[1] "character"
```

2.1 Taulukon luominen

Tilastollisia aineistoja käsitellessä haluamme kuitenkin monesti sekä numeerisia että merkkijonomuotoisia muuttujia samaan tietorakenteeseen. Yleensä käsiteltävät aineistot ovat taulukoita, joissa rivit edustavat havaintoyksiköitä ja, seuraavassa esimerkissämme kurssin ohjaajat, ja sarakkeet tarkasteltavia muuttujia, esimerkissämme ohjaajien pyöräilemät tunnit vuoden aikana, ja ohjaajien kengännumero.

Data frame eli taulukko on R:n tietotyyppi, joka on tarkoitettu juuri tällaisten tilastollisten aineistojen tallentamiseen. Esimerkkiaineistostamme luodaan taulukko funktiolla `data.frame`. Argumentti `stringAsFactors = FALSE` määrittää, että funktio pitää aineiston merkkijonot merkkijonoina, eikä muuta niitä tyypiltään faktoreiksi.

Esimerkki 6.

```
> nimi <- c('Ville', 'Henkka','Oskari')
> pyorailytunnit <- c(3, 130, 200)
> kengannumero <- c (42,47,42)
>
> ohjaajat <- data.frame(nimi, pyorailytunnit, kengannumero,
stringsAsFactors=FALSE)
>
> ohjaajat
    nimi pyorailytunnit kengannumero
```

```
1  Ville              3           42
2 Henkka            130           47
3 Oskari            200           42
>
> class(ohjaajat)
[1] "data.frame"
>
```

Funktiolla `str()` voidaan tutkia tarkemmin taulukon sarakkeiden tietotyyppejä. Funktio myös tulostaa rivien ensimmäiset arvot.

Esimerkki 7.

```
> str(ohjaajat)
'data.frame': 3 obs. of  3 variables:
 $ nimi          : chr  "Ville" "Henkka" "Oskari"
 $ pyorailytunnit: num  3 130 200
 $ kengannumero  : num  42 47 42
>
```

2.2 Aineiston lataaminen tiedostosta

Luettaessa tiedostoa R:n työhakemisto tulee ensin asettaa siihen hakemistoon, jossa tiedosto sijaitsee. Pieniä tiedostoja kannattaa säilyttää samassa hakemistossa jonne R-koodi, jossa ne luetaan, tallennetaan, niin ne on helppo löytää. R:n nykyisen työhakemiston näkee funktiolla `getwd()` ja uuden työhakemiston pystyy asettamaan funktiolla `setwd()`. Huomaa kenoviivojen suunta, ne ovat toiseen suuntaan kuin Windows-järjestelmissä, joten hakemiston nimeä kopioitaessa ne pitää kääntää. Myös toisen "vääränsuuntaisen"kenoviivan lisääminen ensimmäisen perään toimii ja voi olla helpompaa näppäillä.

Esimerkki 8.

```
> getwd()
[1] "C:/Users/Ville/Documents/R/win-library/3.0/muste"
>
> setwd("C:/Users/Ville/Desktop")
> getwd()
[1] "C:/Users/Ville/Desktop"
>
> setwd("C:\\Users\\Ville\\Desktop\\DA_R\\Vuosi_2015")
> getwd()
[1] "C:/Users/Ville/Desktop/DA_R/Vuosi_2015"
```

Aineistot, kuten esimerkkiaineistonamme käyttämä pyöräilybarometri 2014 -aineisto, tulevat usein Excel-taulukkona. Tällöin kannattaa avata tiedosto Excelissä tai vastaavassa taulukkolaskentaohjelmassa ja tallentaa tiedostosta uusi versio CSV-muodossa. CSV on lyhenne sanoista comma separated values; CSV-tiedostossa taulukko on tallennettu selväkielisenä ja taulukon sarakkeet on erotettu pilkulla tai muulla vastaavalla merkillä (tässä tapauksessa puolipisteellä). Harjoituksissa käytettävä aineisto on jo valmiiksi tallennettu CSV-muotoon.

CSV-tiedostoja luetaan taulukkoon funktiolla read.csv. Argumenteiksi annetaan tässä tapauksessa tiedoston nimi (tiedoston on sijaittava työhakemistossa),

tiedostossa käytettävä sarakkeiden erotin (puolipiste) ja stringsAsFactors, joka määrittää muutetaanko aineiston merkkijono-sarakkeet faktoreiksi (oletusarvo on TRUE). Tarkastetaan että aineiston lataus onnistui tulostamalla viisi ensimmäistä riviä kymmenestä ensimmäisestä sarakkeesta. Taulukon tuhatta ensimmäistä riviä voi tarkastella Excel-tyyppisessä taulukossa klikkaamalla taulukon nimeä R-studiossa tai konsolissa `View()`-funktiolla. Sarakkeiden nimien tulkinnat löytyvät aineiston mukana tulevasta koodikirjasta.

Esimerkki 9.

```
> pb <- read.csv(file='PB_2014.csv', sep=';', stringsAsFactors=FALSE)
>
> pb[1:5,1:10]
  RespondentID Aidnr.1  ADate.1 Aalue AA AB.1 AC pnro Aq1 Aq2
1            1    1005 20140822     2  2   38 78  920   1   1
2            2    1010 20140822     2  2   24 59  730   1   5
3            3    1016 20140822     2  2   63 12  210   1   5
4            4    1032 20140822     1  2   61  8  170   1   5
5            5    1034 20140822     2  2   60 24  330   2   4
```

2.3 Alkioihin, sarakkeisiin ja riveihin viittaminen

Taulukkoa voi ajatella matriisina [2], jonka sarakkeet on nimetty, ja jonka sarakkeet voivat olla keskenään eri tietotyyppejä. Taulukon alkiohin, riveihin ja sarakkeisiin voi viitata indekseillä samalla tavalla kuin matriisien alkioihin, riveihin ja sarakkeisiin.

Esimerkki 10.

```
> ohjaajat[3,1]
[1] "Oskari"
>
> ohjaajat[2,]
    nimi pyorailytunnit kengannumero
2 Henkka            130           47
>
> ohjaajat[,3]
[1] 42 47 42
```

Yleensä taulukon sarakkeisiin kannattaa kuitenkin viitata niiden nimellä. Se toisaalta tekee koodista helpommin luettavaa, ja jos taulukkoon tulee uusia sarakkeita tai sarakkeiden järjestys vaihtuu, nimillä viitattaessa koodia ei tarvitse muuttaa. Taulukon sarakkeita voi valita `$`-operaattorilla. Taulukon sarakkeet ovat vektoreita, ja niitä voidaan käsitellä kaikilla ensimmäisellä viikolla opituilla vektorioperaatioilla.

Esimerkki 11.

```
> ohjaajat$nimi
[1] "Ville"  "Henkka" "Oskari"
>
> ohjaajat$kengannumero
```

[2] Oikeasti taulukko on pohjimmiltaan tyypiltään lista eikä matriisi, mutta tästä lisää listojen yhteydessä.

```
[1] 42 47 42
>
> class(ohjaajat$nimi)
[1] "character"
>
> class(ohjaajat$kengannumero)
[1] "numeric"
>
> 5 * ohjaajat$kengannumero
[1] 210 235 210
>
> ohjaajat$kengannumero[ohjaajat$kengannumero < 45]
[1] 42 42
```

2.4 Osa-aineistojen valinta

Aivan kuten matriiseistakin, taulukoista voidaan valita rivejä ehtolauseiden avulla. Jos halutaan esimerkiksi tarkastella kaikkia ohjaajia, joiden nimi on Oskari, tai kaikkia ohjaajia, joiden kengännumero on pienempää kuin 45, sijoitetaan vain haluttu ehto rivin indeksin paikalle (huomaa ehdon jälkeinen pilkku, joka erottaa rivin ja sarakkeen indeksin). Osa-aineistot ovat myös taulukoita, ja ne voidaan tallentaa myöhempää käyttöä varten.

Esimerkki 12.

```
> ohjaajat[ohjaajat$nimi == 'Oskari', ]
    nimi pyorailytunnit kengannumero
3 Oskari            200           42
>
> ohjaajat2 <- ohjaajat[ohjaajat$kengannumero < 45, ]
>
> ohjaajat2
    nimi pyorailytunnit kengannumero
1  Ville              3           42
3 Oskari            200           42
>
> class(ohjaajat2)
[1] "data.frame"
```

Valmiiseen taulukkoon voidaan lisätä sarakkeita kirjoittamalla haluttu sarakkeen nimi $-operaattorilla, ja sijoittamalla arvot siihen vektorina. Sarakkeita voi poistaa sijoittamalla tyhjäarvon `NULL` poistettavaan sarakkeeseen.

Esimerkki 13.

```
> ohjaajat$viikkotunnit <- c(8,8,2)
> ohjaajat
    nimi pyorailytunnit kengannumero viikkotunnit
1  Ville              3           42            8
2 Henkka            130           47            8
3 Oskari            200           42            2
>
> ohjaajat$viikkotunnit <- NULL
```

```
> ohjaajat
    nimi pyorailytunnit kengannumero
1  Ville              3           42
2 Henkka            130           47
3 Oskari            200           42
```

3 Tilastolliset funktiot

3.1 Tunnusluvut

Lasketaan tunnuslukuja esimerkkiaineistostamme. Funktio `length()` palauttaa argumenttina annetun vektorin pituuden, ja `sum()` palauttaa argumentin alkioiden summan, joten näiden avulla saadaan laskettua pyöräilytuntien keskiarvo.

Esimerkki 14.

```
> sum(ohjaajat$pyorailytunnit) / length(ohjaajat$pyorailytunnit)
[1] 111
```

R:ssä on valmiina laaja valikoima tilastollisia funktioita, joten keskiarvo voidaan laskea helpommin käyttäen funktiota `mean()`.

Esimerkki 15.

```
> mean(ohjaajat$pyorailytunnit)
[1] 111
```

R:ssä puuttuvaa arvoa merkitään `NA`:lla. Jos aineistoa luettaessa solu on tyhjä, R sijoittaa sen paikalle `NA`:n. Laskettaessa tunnuslukuja, kuten keskiarvoa vektorista jossa on yksikin puuttuva arvo, R palauttaa puuttuvan arvon. Jos halutaan laskea keskiarvo niistä alkioista joilla on arvo, on `mean()`-funktiolle annettava argumentiksi `na.rm=TRUE`. Sama argumentti toimii myös monen muun funktion kanssa.

Esimerkki 16.

```
> ika <- c(30, 30, NA, 44)
> mean(ika)
[1] NA
> mean(ika, na.rm=TRUE)
[1] 34.66667
```

R:ssä on funktiot mm. myös keskihajonnalle, mediaanille, minimille ja maksimille.

Esimerkki 17.

```
> sd(ohjaajat$pyorailytunnit)
[1] 99.86491
>
> median(ohjaajat$pyorailytunnit)
[1] 130
>
> min(ohjaajat$pyorailytunnit)
[1] 3
>
```

```
> max(ohjaajat$pyorailytunnit)
[1] 200
```

3.2 Simulointi ja jakaumafunktiot

Tällä kurssilla keskitytään jakaumien osalta lähinnä normaali- ja binomijakaumien käsittelyyn. Näiden käsittelyä varten R:ssä on käteviä funktiota, joihin perehdytään seuraavaksi.

Seuraavissa esimerkeissä tutkitaan jakaumia $\mathrm{N}(0,1)$ ja $\mathrm{Bin}(1/3, 13)$. Vastaavat funktiot löytyy myös muille keskeisille jakaumille.

Esimerkki 18. *Kvantiilifunktiot: Haetaan piste, jonka vasemmalla puolella on* $1/4$ *jakauman todennäköisyysmassasta*

```
# Normaalijakauma
> qnorm(mean=0, sd = 1, p = 1/4, lower=T)
[1] -0.6744898

# Binomijakauma
> qbinom(p = 1/4, size=13, prob=1/3, lower.tail = T)
[1] 3
```

Esimerkki 19. *Tiheysfunktiot: Tiheys- ja pistetodennäköisyysfunktion* $f(x)$ *arvo kohdassa* $x = 4$*.*

```
# Normaalijakauma
> dnorm(x=4, mean=0, sd = 1)
[1] 0.0001338302

# Binomijakauma
> dbinom(x = 4, size = 13, prob = 1/3)
[1] 0.2296147
```

Esimerkki 20. *Kertymäfunktiot: Arvo kohdassa* $q = 4$*.*

```
# Normaalijakauma
> pnorm(q = 4, mean=0, sd = 1)
[1] 0.9999683

# Binomijakauma
> pbinom(q = 4, size = 13, prob = 1/3)
[1] 0.5520387
```

Esimerkki 21. *Jakauman simulointi: Satunnaisotos*

```
# Normaalijakauma
> rnorm(mean = 0, sd=1, n=10)
 [1] -0.8876916 -1.3342456  0.2967970 -0.0250188  0.8236606  1.0947668 -0.3756786
 [8] -0.2220601 -1.2274948 -0.4169028

# Binomijakauma
> rbinom(size = 13, prob=1/3, n=10)
 [1] 2 4 5 6 3 3 5 6 5 5
```

4 Aineiston visuaalinen tarkastelu

Aineistoa voidaan R:ssä visualisoida monilla eri tavoilla, joista käsitellään nyt alkeiden kannalta oleellisimmat:

- `curve()`: Funktion kuvaaja
- `plot()`: Monikäyttöinen piirtofunktio
- `hist()`: Histogrammi
- `boxplot()`: Boxplot (laatikko ja viikset)

Tutkitaan näiden käyttöä seuraavaksi esimerkein. Piirrä kuvat R:llä nähdäksesi miltä ne näyttävät.

Esimerkki 22. *Piirretään funktion x^2 kuvaaja välillä $[0, 1]$ käyttäen `curve()`-funktiota*

```
curve(x^2, from=0, to=1)
```

Esimerkki 23. *Piirretään funktion x^2 kuvaaja välillä $[0, 1]$ käyttäen `plot()`-funktiota*

```
x <- seq(0,1,by=0.01)
plot(x=x, y=x^2, type='l')
```

Huomaa, että tässä valinta `type='l'` käskee piirtämään kuvaajan viivoina. Kokeile myös, mitä tapahtuu ilman tätä.

Käytetään seuraavaksi R:n mukana valmiiksi tulevaa esimerkkiaineistoa Iris, johon on kerätty mittaustuloksia kolmesta erilaisesta kurjenmiekkalajista. Lisätietoja kyseisestä aineistosta saat komennolla `?iris`.

Esimerkki 24. *Tarkastellaan kaunokurjenmiekkojen (iris setosa) terälehden pituuksia visuaalisesti histogrammin avulla:*

```
hist(iris[iris$Species=="setosa",]$Petal.Length)
```

Esimerkki 25. *Tarkastellaan vielä kirjokurjenmiekkojen (iris versicolor) terälehden pituuksia boxplotilla:*

```
boxplot(iris[iris$Species=="versicolor",]$Petal.Length)
```

Esimerkki 26. *Tarkastellaan vielä koko aineiston terälehden pituuksia boxplotilla, lajitteluperusteena kasvin laji :*

```
boxplot(iris$Petal.Length ~ iris$Species)
```

Tutkitaan seuraavaksi eri kurjenmiekkalajien terälehtien pituutta suhteessa terälehtien leveyteen. Aloitetaan ensin valitsemalla yksi laji.

Esimerkki 27. *Tutkitaan nyt lajia Iris setosa ja valitaan nyt x-akselille terälehden havaittu pituus ja y-akselille leveys ja piirretään koko komeus `plot()`-funktiolla. Huomaa, että aineiston voi antaa `plot()`-funktiolle parametrilla `data`.*

```
plot(Petal.Width ~ Petal.Length,
    data=iris[iris$Species == "setosa",])
```

Kuvaajiin saa väriä antamalla `plot()`-funktiolle parametrin `col` arvoksi värin joko tekstina (esim "red", "blue"jne) tai jonkin numeron. Myös monia muita vaihtoehtoja löytyy.

Esimerkki 28. *Esitetään seuraavaksi Iris-aineiston kaikkien lajien terälehtien pituus suhteessa niiden leveyteen ja merkitään eri lajeja eri väreillä.*

```
> # Katsotaan lajien järjestys
> levels(iris$Species)
[1] "setosa"     "versicolor" "virginica"

> # Luodaan värivektori tälle järjestykselle
> iris_colors <- c("red","green","blue")

> # Piirretään kuvaaja koko aineistosta väreillä
> plot(Petal.Width~Petal.Length, data=iris,
     col=iris_colors[Species])
```

5 Tehtäviä

1. Aineiston lataaminen ja ikäjakauman tarkastelu.
   a) Lataa Pyöräilybarometri 2014 - aineisto tiedostosta PB_2014.csv taulukkoon `pb` ja varmista latauksen onnistuminen tulostamalla taulukon 10 ensimmäistä riviä ja saraketta.
   b) Laske kyselyn vastaajien keski-ikä, mediaani-ikä, ja pienin ja suurin ikä. Aineiston sarakkeiden nimien selitykset löytyvät koodikirjasta.
   c) Piirrä histogrammi kyselyn vastaajien ikäjakaumasta.
   d) Laske miesten ja naisten keski-ikä erikseen aineistosta.
2. Puuttuvat arvot ja keskiarvo.
   a) Laske kuinka moni vastaaja vastasi kysymykseen Aq5 (Kuinka turvalliseksi koet pyöräilyn Helsingissä 'ei osaa sanoa' (katso arvojen tulkinta koodikirjasta.
   b) Muuta muuttujan Aq5 'ei osaa sanoa'-arvot puuttuviksi arvoiksi sijoittamalla `NA` niihin. Varmista onnistuminen laskemalla a-kohdan tulos uudelleen ja toteamalla että se on nolla.
   c) Laske keskiarvo muuttujasta Aq5 EOS-vastauksien poistamisen jälkeen, eli kuinka turvalliseksi vastaajat kokevat keskimäärin pyöräilyn Helsingissä asteikolla yhdestä neljään, missä yksi vastaa turvallista ja neljä turvatonta.
   d) Laske edellisen kohdan keskiarvot sukupuolittain.
3. Osa-aineistojen valinta.
   a) Kuinka moni kyselyn vastaajista on yli 40-vuotias?
   b) Kuinka moni kysely vastaajista on 23-27 vuotias nainen?
   c) Laske muuttujan Aq5 (josta 'ei osaa sanoa'-vastaukset on poistettu) keskiarvo yli 50-vuotialle naisille.

d) Laske muuttujan Aq5 (josta 'ei osaa sanoa'-vastaukset on poistettu) keskiarvo 18-25-vuotiaille miehille.

4. Tutustutaan funktioiden kuvaajien piirtämiseen

   a) Piirrä funktion $\log(x)$ kuvaaja välillä $(0, 10)$ käyttäen funktiota `curve()`.

   b) Tutki funktion `abline` toimintaa esimerkiksi komennon `?abline` avulla ja lisää edelliseen kuvaajaan vaakaviiva kohtaan $y = 1$. Vihje: Argumentit `h` ja `v`.

   c) Lisää vielä kuvaan **punainen** pystyviiva kohtaan $x = e$, missä e on neperin luku.

5. Simuloidaan kolikonheittoa reilulla kolikolla, eli kruunan todenäköisyys on $p = 1/2$. Olkoon 1 = kruuna, 0 = klaava.

   a) Simuloi kymmenen heiton sarja.

   b) Simuloi 100 kymmenen heiton sarjaa ja laske kruunien lukumäärä. Vihje: Argumentti `size` ja funktio `sum()`.

   c) Simuloi 10000 kymmenen heiton sarjaa ja laske kruunien lukumäärä. Vihje: Argumentti `size` ja funktio `sum()`.

6. Tutkitaan binomijakaumaa ja pohditaan mitä tapahtuu, kun otoskokoa kasvattaa.

   a) Simuloi otos binomijakaumasta kun $p = 1/4$ ja $n = 500$.

   b) Ota nyt a-kohdan otoksia 10000 kappaletta ja tallenna tulos muuttujaan `result`. Vihje: argumentti `size`.

   c) Piirrä vektorista `result` histogrammi.

7. Tutkitaan kurssin Johdatus tilastolliseen päättelyyn laskuharjoitusten 1 tehtävää 4 d) simulaation avulla.

   a) Simuloi 50 arvoa jakaumasta $\text{Tas}(0, \theta)$, missä $\theta = 8$. Vihje: `runif`.

   b) Simuloi nyt 500000 arvoa ja tallenna tulos vektoriin.

   c) Luo vektorista matriisi, jossa rivien lukumäärä on 50. Tallenna matriisi muuttujaan `A`.

   d) Nyt jokainen matriisin `A` sarake edustaa yhtä 50 havainnon otosta. Sarakekohtaiset maksimiarvot saa laskettua esimerkiksi funktiokutsulla `sapply(1:10000, function(x) max(A[,x]))`. Kuinka monen sarakkeen maksimiarvo on alle 4.8? Entä alle 7?

   e) Piirrä histogrammi d-kohdan maksimivektorista.

8. Tutki esimerkkiä 28 ja tee vastaava vertailu kurjenmiekkojen verholehdille (Sepal.Width ja Sepal.Length). Käytä grafiikassasi värejä lajeittain seuraavasti:

   - Iris Setosa = Keltainen
   - Iris Versicolor = Harmaa
   - Iris Virginica = Pinkki

Etsi myös itseäsi miellyttävä merkki vakioasetuksena olevan pallukan sijaan antamalla argumentiksi `pch` jokin kokonaisluku.

9. Tee edellisen tehtävän tarkastelut myös käyttäen boxplotia. Tällä kertaa ei tarvitse muuttaa värejä tai merkkejä.