# Põhilised andmetöötlusvõtted

Andmete sisselugemisest ei tarvitse olla küllalt, et analüüsima hakata. Andmed võivad olla ebasobival kujul. Näiteks võivad olla isendite andmed ühes, aga keskkonna-andmed teises andmetabelis. Või siis võivad sama isendi korduvad mõõtmised olla ühe rea erinevatel veergudel, aga tarvis oleks kõik mõõtmised saada ühte tulpa.

R on vabavara, kuhu saavad kõik soovijad funktsioone juurde kirjutada (ja need siis oma pakettides teistelegi kättesaadavaks teha). Seetõttu on ka andmete formaadi muutmise jaoks rohkesti erinevaid funktsioonide komplekte: vastavate autorite „visioon“ mugavatest andme-teisendustest on kajastatud R-i „baaspaketis“ ja lisapakettides reshape2, plyr, dplyr. Käesolevas materjalis on autori harjumuse tõttu kasutatud pigem baaspakettide funktsioone, kuigi kõiki järgnevaid asju oleks kindlasti võimalik (ilmselt ka kellegi arust mugavamalt ja kiiremini) ära teha ka eelmainitud pakettide vahendeid kasutades.

## Uue tunnuse lisamine andmestikku

Lähtume eelmises praktikumis kirjeldatud andmestikust testdata.txt. Meeldetuletuseks: kopeerige järgmised read (sealhulgas viimane, tühi rida!) Notepad-i ja salvestage see testdata.txt nime all.

nimi,sugu,kaal

Mark,mees,85

Mati,mees,100

Kati,naine,60

Mari,naine,75

Lugege see andmestik R-i sisse (juhis on leitav eelmise praktikumi materjalist) nime all testdata.

Lisame andmestikku nende inimeste pikkused (cm).

testdata$pikkus <- c(174, 180, 169, 165)

****Harjutus****

Arvuta välja ja salvesta kehamassiindeks selle andmetabeli tunnusesse nimega bmi.

## Kahe andmestiku ühendamine

Oletame, et mingis teises andmestikus on meil mõõdetud inimeste jäsemeid. Eesmärgiks on uurida jäsemete pikkuse ja kaalu vahelist seost. Tekitage järgnev andmestik (näiteks kopeerige Notepad-i, salvestage, lugege R-i sisse) nime all jäsemepikkused.

nimi,jäse,jäsemepikkus

Mark,käsi,73

Mari,käsi,70

Mark,jalg,100

Mari,jalg,100

Kati,jalg,95

Karl,jalg,101

Paneme tähele, et Matilt ei ole meil ühegi jäseme pikkust mõõdetud. Küll aga oleme suutnud mõõta Karli jala, kelle kaalu ja pikkust meil ei ole.

Kahe andmestiku ühendamiseks on vaja, et neis oleks mingid võtmetunnused, mille väärtuste kaudu saaks andmetabelite vastavaid ridu siduda. Tunnus nimi kõlbab selleks väga hästi: see on olemas mõlemas andmestikus ja selle läbi saab konkreetselt öelda, et missugused read tabelis testdata vastavad missugustele ridadele tabelis jäsemepikkused.

Andmestike ühendamise töö teeb meie eest ära funktsioon merge. Talle tuleb öelda, et missugused andmestikud on vaja ühendada. Lisaks on talle igaks juhuks hea üle täpsustada, et missuguse tunnuse järgi käib andmestike ühendamine. Tulemus on jällegi mõistlik ära salvestada.

yhendatud <- merge(testdata, jäsemepikkused, by="nimi")

Uurime nüüd korra tulemust.

> yhendatud

nimi sugu kaal pikkus jäse jäsemepikkus

1 Kati naine 60 169 jalg 95

2 Mari naine 75 165 käsi 70

3 Mari naine 75 165 jalg 100

4 Mark mees 85 174 käsi 73

5 Mark mees 85 174 jalg 100

Pange tähele, et vaikimisi jäetakse alles ainult need inimesed, kes olid olemas mõlemas ühendatavas tabelis. Kui me soovime näha kõiki vasakpoolses tabelis leidunud inimesi (näiteks et teada saada, kellel jäid kõik jäsemed mõõtmata), tuleks teha nii:

yhendatud2 <- merge(testdata, jäsemepikkused, by="nimi", **all.x=T**)

Tulemus on siis selline:

> yhendatud2

nimi sugu kaal pikkus jäse jäsemepikkus

1 Kati naine 60 169 jalg 95

2 Mari naine 75 165 käsi 70

3 Mari naine 75 165 jalg 100

4 Mark mees 85 174 käsi 73

5 Mark mees 85 174 jalg 100

**6 Mati mees 100 180 <NA> NA**

NA tähistab R-is puuduvaid väärtusi (Not Available).

Analoogia põhjal võite mõelda, mida võiks teha merge(testdata, jäsemepikkused, by="nimi", **all.y=T**).

## Kokkuvõtted tunnuse tasemete kaupa

Võime soovida näha näiteks keskmist kaalu sugude kaupa, või siis keskmist jala ja käe pikkust sugude kaupa. Selle jaoks sobib näiteks funktsioon tapply. See funktsioon tahab saada kolme argumenti:

1. arvuline tunnus, mille kohta kokkuvõtteid soovitakse (see peaks olema arvude vektor)
2. faktorid (üks või rohkem), mille tasemete (kombinatsioonide) kaupa soovitakse kokkuvõtteid saada (see peaks olema vektor või siis vektorite list, vaata näidet allpool)
3. funktsioon, mis annab meile soovitud kokkuvõtte

Näiteks keskmised kaalud sugude kaupa saame nii:

tapply(testdata$kaal, testdata$sugu, mean)

Keskmised jäsemete pikkused sugude kaupa saame aga nii:

tapply(yhendatud$jäsemepikkus, **list**(yhendatud$sugu, yhendatud$jäse), mean)

## for-tsükkel

R näeb välja nagu programmeerimiskeel. Nagu igale programmeerimiskeelele kohane, on ka R-is võimalik teha tsükleid. Jätame konkreetsed kasutus-näited hilisemaks ja toome siin ära ainult paar süntaksinäidet:

for (i in **1:nrow(yhendatud)**) {

print(i)

}

Vaadake, mida teeb eelneva programmikese esimesel real paksus kirjas toodud käsk eraldi võttes:

1:nrow(yhendatud)

Ilmselt ei ole üllatav, et funktsioon nrow(yhendatud) annab andmetabeli yhendatud ridade arvu. Üllatav võib olla see, et kooloni abil saab teha järjestikuste täisarvude vektoreid (alates : kuni).

Veel üks näide mitmekesisusest:

for (kala in c("särg", "räim", "mättias")) {print(kala)}

## Uute funktsioonide kirjutamine

Soovime arvutada välja näiteks pikkuste variatsioonikordajat (*coefficient of variation*, standardhälbe ja keskmise suhe) sugude kaupa. Jõle mugav oleks kasutada sedasama tapply funktsiooni. Aga ei tule kohe meelde variatsioonikordaja funktsiooni nimi R-is… aga pole hullu, sest et me saame selle funktsiooni ise kirjutada. Skeem oleks selline:

*funktsiooni nimi* = function (*funktsiooni argumendid*) {

*mida see funktsioon teeb*

return(*mida see funktsioon tagastab*)

}

Näiteks variatsioonikordajat arvutav funktsioon võiks olla selline:

cv <- function(x) {

return( sd(x) / mean(x) )

}

Nüüd saame rakendada oma uut funktsiooni teise funktsiooni sees:

tapply(testdata$pikkus, testdata$sugu, cv)

## Andmetabeli teisendamine pikka ja laia formaati

Eelkõige näiteks korduvate mõõtmiste andmestike korral saab rääkida andmestiku pikast ja laiast kujust. Laiaks formaadiks nimetatakse sellist andmetabelit, kus iga mõõdetud looma kohta on üks rida. Näiteks käesoleva kursuse kodutöö korduvate mõõtmiste andmestik laias formaadis näeb välja selline:

individual treatment time1 time2 time3 time4

1 snail 1 5 4 3

2 snail 1 5 4 2

3 snail 2 4 4 2

4 nosnail 3 1 2 3

5 nosnail 3 1 1 3

6 nosnail 4 2 2 5

Lai formaat näeb välja kompaktsem ja inimesele paremini loetav / täidetav, aga arvutitele meeldib sageli statistilisi analüüse teha just pikas formaadis andmetega. Pikaks formaadiks nimetatakse sellist andmetabelit, kus iga mõõtmine on eraldi rida. Näiteks seesama korduvate mõõtmiste andmestik oma esialgsel kujul on pikas formaadis.

Andmete teisendamiseks pika ja laia kuju vahel on rohkesti funktsioone, üks neist on reshape.

### Pikk -> lai

Et saada andmed laia formaati, tuleks eelnevalt loodud jäsemepikkuste andmestikuga talitada nii:

jäsemed.lai <- reshape(

jäsemepikkused, # mis andmestikku teisendada

direction = "wide", # mis suunas teisendada

idvar = c("nimi"), # mis tunnus(ed) iseloomustavad indiviidi

timevar = "jäse" # mis tunnuse väärtused saavad tulbanimedeks

)

Harjutus

Vala eelmainitud korduvate mõõtmiste andmestik repedata18a.xls laia formaati.

### Lai -> pikk

Andmete „sulatamiseks“ pikka formaati kasutame funktsiooni melt lisapaketis nimega reshape2. Pakett tuleks siis kasutusele võtta käsuga

library(reshape2)

Eelnevalt loodud andmestikuga testdata tuleks siis talitada nii:

testdata.pikk <- melt(

testdata, # mis andmestikku teisendada

id.vars = c("nimi", "sugu"), # mis tunnus(ed) iseloomustavad indiviidi

variable.name = "tunnus", # endiste veerunimede veeru nimeks

value.name="väärtus" # väärtuste veeru nimeks

)

## Esimene suurem harjutus: liblikaloendus

Ühe suve jooksul loeti eelnevalt kindlaks määratud aladel kolmel korral liblikaid. Kõik need loendus-alad on kirjas tabelis alad.csv, kus on lisaks ala koodidele kirjas ka ala koordinaadid (antud juhul mitte pikkus- ja laiuskraadid, vaid sisuliselt meetrite arv kuskilt kokkuleppelisest lähtepunktist). Saadud liblikaloenduste tulemused (poole tunni jooksul alal kohatud liblikate arv) on failis loendused.csv. Et uurida liblikaliikide eelistusi keskkonnatingimuste suhtes, ongi alade tabelisse juba juurde otsitud mulla niiskus. Lisaks mulla niiskusele võivad liblikaid mõjutada igasugused asjad, miks mitte näiteks ümbritsevate metsade tüüp. Õnneks on Eestis olemas selline tore asi nagu metsaregister, kus on kõige muu kasuliku info seas olemas ka okaspuude osakaal iga väikese metsatükikese kohta. Neid metsatükikesi nimetatakse eraldisteks ja nende andmed on tabelis eraldised.csv. Targad onud arvavad, et eelkõige ala keskpunktist kuni 500 meetri kaugusel asuv mets võiks liblikate eluga kuidagi seotud olla.

Need kolm eraldiseisvat tabelit aga ei kõlba analüüsiks. Andmed tuleb saada sellisele kujule, kus tabeli iga rea peal on üks loendus-ala ja liblikaliigid on veergudes. Sealjuures iga liigi kohta peaks kirjas olema maksimaalne arvukus, mida antud alal suve jooksul kohati (kõiki alasid külastati ju kolm korda, et ära näha nii hiljem kui ka varem lendavad liigid).

Üks võimalik ettepanek tööde järjekorra suhtes:

1. Loo alade tabelisse uus tunnus, kus on iga ala jaoks kirjas nende eraldiste keskmine okaspuude osakaal, mis jäävad antud alast 500 meetri kaugusele (meenuta matemaatikatundi ja Pythagorase valemit). Kasuta for-tsüklit üle alade (for (i in 1:nrow(alad)) {…}).
2. Kasuta funktsiooni tapply, et leida iga liblikaliigi maksimaalne arvukus igal alal.
3. Veendu, et loendusandmete tabel oleks lõpuks laias formaadis (et alad oleksid ridades ja liblikaliigid veergudes).
4. Ühenda loendusandmete tabel ja alade tabel.

Kui andmestik on sobival kujul käes, võid proovida uurida erinevate liblikaliikide arvukuste omavahelisi seoseid, seoseid keskkonnatingimustega või lausa midagi mitmemõõtmelist.

Erilised huvilised võivad proovida ära arvata, kuidas (missugusest teile käesoleva kursuse raames teada olevast statistilisest jaotusest, millele vastavat analüüsimudelit te ka praktikumis nägite) ma genereerisin liblikate arvukused (käsitledes kõiki vaatlusi tabelis loendused.csv omavahel sõltumatutena, ärge arvutage mingeid suve maksimume vms) ja mis on selle mudeli parameetrid (nagu nad õigest analüüsimudelist peamõjude hinnangutena välja tulevad, arvestades et ainult eelmainitud kolmes tabelis leiduvad tunnused võivad seda liblikate arvukust mõjutada).