

Rakendustarkvara: R. Sügis 2017, 3. praktikum*

1 Toimingud andmestikuga

1.1 Andmestike täiendamine ja ühendamine (mestimine)

Oleme tutvunud, kuidas andmestikust saab eraldada alamhulki (nt ridu või veerge, mis vastavad teatud kriteeriumitele). Ühekaupa saab uusi veerge `data.frame`-tüüpi objektile lisada nii dollarimärgi kui kantsulgude abil, kui anname ette uue veeru nime.

Vaatame siin näiteandmestikuna andmestikku `mk` (*maakonnad.txt*), mis sisaldab infot USA 5 osariigi mõnede maakondade kohta (425 maakonda). Loeme andmed sisse ja lisame tunnuse, mis näitab, kas maakonna sündimuskordaja väärtus ületab 2.1 piiri:

```
mk <- read.table("http://kodu.ut.ee/~annes/Rkursus/maakonnad.txt", sep = " ", header = TRUE)
mk[, "taastootev"] <- mk$births > 2.1 # kas sündimuskordaja on suur?
mk$taastootev <- mk$births > 2.1
```

Mitut uut veergu või rida saab andmestikule lisada `cbind(.)` ja `rbind(.)` käskudega. Neid kasutades tuleb aga olla ettevaatlik: lisatavas reas peab olema sama palju elemente kui on andmestikus veerge ja lisatavas veerus peab olema sama palju elemente, kui on andmestikus ridu. Tunnuste nimed lisatavates ridades peavad vastama andmestiku omale, samuti peab lisatavates veergudes objektide järjestus olema sama kui esialgses andmestikus.

```
# teeme kaks uut tunnust
suurus <- cut(mk$pop_estimate, c(0, 1000, 10000, 1000000, Inf),
             labels = c("mikro", "väike", "keskmine", "suur"), include.lowest = T)
sooylekaal <- ifelse(mk$fem > 50, "F", "M")
# lisame need vektorid andmestiku lõppu veergudeks:
mk <- cbind(mk, suurus, sooylekaal)
# või
lisatabel <- data.frame(suurus, sooylekaal)
mk <- cbind(mk, lisatabel)

# ridade lisamine, praegu tekitame sellega dubleeritud vaatlusi!
valik <- rep(seq(1, nrow(mk), by = 60), c(1:3, 1:3, 1, 1))
mktopelt <- rbind(mk, mk[valik, ])
```

Sageli on analüüsiks vajaminevad andmed mitmes erinevas andmetabelis, mis võivad olla objektide arvu poolest ka erinevad. Näiteks jooksvõistluste andmete puhul võivad ühes tabelis kirjas isikuandmed kohta (nimi, sugu, vanus), teises tabelis aga võistlustulemuste andmed. Kui sooviks analüüsida tulemuste jaotust sugude vahel või sõltuvalt vanusest, oleks mugav need tabelid mestida käsuga `merge(.)`. Andmete ühendamiseks peaks mõlemas tabelis olema nn võtmetunnus, mille abil read vastavusse pannakse.

```
isikud <- data.frame(nimi = c("Peeter", "Mari", "Tiina", "Laine"),
                    sugu = c("M", "N", "N", "N"), vanus = c(30, 22, 25, 20))
tulemused <- data.frame(nimi = c("Mari", "Peeter", "Tiina", "Peeter", "Toomas"),
                       tulemus = c(30.1, 22.5, 18.4, 25.3, 20.4),
                       voistlus = c("jooks1", "jooks1", "jooks2", "jooks2", "jooks2"))
```

*Praktikumijuhendid põhinevad aine MTMS.01.092 Rakendustarkvara: R (2 EAP) materjalidel, mille autorid on Mait Raag ning Raivo Kolde

```
isikud
```

```
##      nimi sugu vanus
## 1 Peeter   M    30
## 2  Mari    N    22
## 3 Tiina    N    25
## 4 Laine    N    20
```

```
tulemused
```

```
##      nimi tulemus voistlus
## 1  Mari    30.1   jooks1
## 2 Peeter   22.5   jooks1
## 3 Tiina    18.4   jooks2
## 4 Peeter   25.3   jooks2
## 5 Toomas   20.4   jooks2
```

```
(kokku <- merge(isikud, tulemused, by = "nimi", all = TRUE))
```

```
##      nimi sugu vanus tulemus voistlus
## 1 Laine    N    20      NA      <NA>
## 2  Mari    N    22    30.1   jooks1
## 3 Peeter   M    30    22.5   jooks1
## 4 Peeter   M    30    25.3   jooks2
## 5 Tiina    N    25    18.4   jooks2
## 6 Toomas <NA>   NA    20.4   jooks2
```

1.1.1 Ülesanded

1. Kuidas peaks `merge` käsu kirja panema, kui `isikud` ja `tulemused` andmestikes on võtmetunnusel erinev nimi?
2. Proovi, mis muutub `merge` käsu tulemuses, kui kasutada `all = TRUE` argumenti asemel `all.x = TRUE` või `all.y = TRUE`.

1.2 Unikaalsed ja mitmekordsed elemendid. Hulgatehted.

Mõnikord on üks objekt andmestikus mitu korda, ent me soovime seda ainult ühel korral analüüsi kaasata. Etteantud vektorist saab unikaalsed elemendid kätte käsuga `unique(.)`. Käsuga `duplicated(.)` saab teada, kas ette antud vektoris on element esimest või juba mitmendat korda (tulemuseks on tõeväärtusvektor, kus `TRUE` tähendab seda, et antud väärtus on juba mitmendat korda).

```
kokku$nimi
```

```
## [1] Laine  Mari   Peeter Peeter Tiina  Toomas
## Levels: Laine Mari Peeter Tiina Toomas
```

```
unique(kokku$nimi)
```

```
## [1] Laine  Mari   Peeter Tiina  Toomas
## Levels: Laine Mari Peeter Tiina Toomas
```

```
duplicated(kokku$nimi)
```

```
## [1] FALSE FALSE FALSE  TRUE FALSE FALSE
```

R-is on realiseeritud ka elementaarsed hulgaoperaatorid: ühisosa, ühend ja vahe. Käsud `union(x, y)` tagastab ette antud kahe vektori `x` ja `y` elementidest koostatud uue vektori, sealjuures mõlema vektori kõik elemendid

on esindatud. Käsk `intersect(x, y)` tagastab vektori elementidest, sealjuures on esindatud ainult need elemendid mis on nii vektoris `x` kui ka `y`. Käsk `setdiff(x, y)` tagastab vektori, kus on ainult need `x` elemendid, mida vektoris `y` ei ole. Kõik hulgatehete käsud tagastavad sellised vektorid, kus igat elementi on ainult üks kord

```
x <- c(1:5, 1:5)
y <- 3:7
union(x, y)
intersect(x, y)
setdiff(x, y)
```

1.2.1 Ülesanded

1. Kontrolli kas andmestikus `mktopelt` on dubleeritud andmeid. Mitu objekti on korduvad? Kas mõni maakond on rohkem kui 2 korda korratud? Kui jah, siis millised?

1.3 Sorteerimine

Väga sageli soovime, et andmestiku read oleks mingi tunnuse (nt inimese vanuse) alusel sorteeritud. Sorteerimiseks on R-is kaks olulist käsku:

- `order(.)` tagastab etteantud vektori elementide järjekorranumbrid sellises järjekorras, et saaksime järjestatud vektori, mh argumentiga `decreasing` saab määrata, kas see on kahanev või kasvav;
- `sort(.)` tagastab etteantud vektori elemendid kasvavas (või kahanevas) järjekorras.

```
x <- c(8, 1, NA, 7, 7)
order(x)
```

```
## [1] 2 4 5 1 3
```

```
sort(x, na.last = TRUE)
```

```
## [1] 1 7 7 8 NA
```

```
x[order(x)]
```

```
## [1] 1 7 7 8 NA
```

```
#andmestiku ridade sorteerimine kahe vektori järgi:
```

```
kokku[order(kokku$vanus, kokku$tulemus, decreasing = TRUE), ]
```

```
##      nimi sugu vanus tulemus voistlus
## 4 Peeter   M    30    25.3   jooks2
## 3 Peeter   M    30    22.5   jooks1
## 5 Tiina    N    25    18.4   jooks2
## 2 Mari     N    22    30.1   jooks1
## 1 Laine    N    20     NA    <NA>
## 6 Toomas  <NA>   NA    20.4   jooks2
```

1.3.1 Ülesanded

1. Loe sisse kaks andmestikku, esmalt arstivisiitide andmestik:
`visiidid <- read.table("http://kodu.ut.ee/~annes/Rkursus/visiidid.txt", sep = "\t", header = TRUE)`. Selles andmestikus on neli veergu: `ik` – isikukood, `visiidi_kp` – arsti külastamise kuupäev, `vererohk` – süstoolne vererõhk antud arstivisiidil (mmHg), `crv` – C-reaktiivse valgu kontsentratsioon (nn näpuveri) (mg/L). Visiitide andmestik kirjeldab 10 aasta jooksul

arsti külastamisel tehtud vererõhu ja CRV mõõtmisi. Teiseks loe sisse andmestik, kus on inimeste isikukoodid ja isikukoodist tuletatud info(sugu, sünnikuupäev, vanus): `inimesed <- read.table("http://kodu.ut.ee/~annes/Rkursus/isikud.txt", sep = "\t", header = TRUE)`. Tutvu andmestikega (dimensioon, tunnused).

2. Järjesta visiitide andmestik kasvavalt isikukoodi ja arstivisiidi kuupäeva järgi.
3. Ühenda isikukoodide ning visiitide andmestik. Mitu inimest on käinud 10 a jooksul arsti juures? Kas arsti mitte külastanud isikute osakaal (protsentuaalselt) on suurem meeste või naiste hulgas?

1.4 Pikk ja lai andmetabel. Pakett reshape2

Teise praktikumi alguses näiteks toodud traditsioonilist andmetabeli kuju ehk *objekt* $\times$ *tunnus*-maatriksit nimetatakse vahel ka nn **laias formaadis andmestikuks**, tabeli iga rida vastab ühele objektile, infoliasus on viidud miinimumini. Näide laias formaadis andmestikust:

```
##   nimi kaal pikkus sugu pulss0m pulss10m pulss30m
## 1 Mari   68   170    2     70      130      150
## 2 Jaan   65   180    1     80      120      120
## 3 Jüri  100   190    1     80      190       NA
```

Pikas formaadis andmestiku puhul proovitakse hoida kõiki sama omadust kirjeldavaid andmeid ühes veerus; üks objekt võib kajastuda mitmel real. Sellisel kujul andmestikku on mõnikord mugav arvuti abil analüüsida (nt segamudelite hindamine, joonised ggplot2 paketiga):

```
##   nimi kaal pikkus sugu variable value
## 1 Mari   68   170    2 pulss0m     70
## 2 Jaan   65   180    1 pulss0m     80
## 3 Jüri  100   190    1 pulss0m     80
## 4 Mari   68   170    2 pulss10m    130
## 5 Jaan   65   180    1 pulss10m    120
## 6 Jüri  100   190    1 pulss10m    190
## 7 Mari   68   170    2 pulss30m    150
## 8 Jaan   65   180    1 pulss30m    120
## 9 Jüri  100   190    1 pulss30m     NA
```

Eriti äärmuslik on pika formaadi puhul hoida kõiki arvulisi tunnuseid ühes veerus:

```
##   nimi variable value
## 1 Mari      kaal    68
## 2 Jaan      kaal    65
## 3 Jüri      kaal   100
## 4 Mari     pikkus   170
## 5 Jaan     pikkus   180
## 6 Jüri     pikkus   190
## 7 Mari      sugu     2
## 8 Jaan      sugu     1
## 9 Jüri      sugu     1
## 10 Mari    pulss0m    70
## 11 Jaan    pulss0m    80
## 12 Jüri    pulss0m    80
## 13 Mari    pulss10m   130
## 14 Jaan    pulss10m   120
## 15 Jüri    pulss10m   190
## 16 Mari    pulss30m   150
## 17 Jaan    pulss30m   120
## 18 Jüri    pulss30m    NA
```

Kuigi R-i baaspaketis on kaasas käsk `reshape()`, siis on sellega üsna tüütu teisendada andmestikku ühest formaadist teise. Seetõttu tutvume paketi **reshape2**, milles olulisimad käsud `melt()` ja `dcast()` aitavad vastavalt teisendada andmestikku laiast formaadist pikka ja vastupidi, ning teha veel täiendavaid toiminguid/arvutusi.

Funktsioon `melt()` teisendab andmed laiast formaadist pikka. Argumendiga `measure.vars` saab sellele ette anda veerunimede või -indeksite vektori, milles olevad tunnused pannakse kõik ühte veergu. Vaikimisi pannakse kõik arvulised väärtused ühte veergu nimega `value` ning teises veerus nimega `variable` on kirjas, mida antud väärtus tähendab (millises veerus see väärtus esialgses andmestikus oli).

```
library(reshape2) # kui arvutis pole, siis installida: install.packages("reshape2")
vr <- data.frame(nimi = c("Mari", "Jaan", "Jüri"), kaal = c(68, 65, 100),
                pikkus = c(170, 180, 190), sugu = c(2, 1, 1), pulss0m = c(70, 80, 80),
                pulss10m = c(130, 120, 190), pulss30m = c(150, 120, NA))
(m <- melt(vr, measure.vars = 5:7)) # välimised sulud tingivad ekraanile trükkimise
```

```
##   nimi kaal pikkus sugu variable value
## 1 Mari   68   170    2  pulss0m    70
## 2 Jaan   65   180    1  pulss0m    80
## 3 Jüri  100   190    1  pulss0m    80
## 4 Mari   68   170    2  pulss10m   130
## 5 Jaan   65   180    1  pulss10m   120
## 6 Jüri  100   190    1  pulss10m   190
## 7 Mari   68   170    2  pulss30m   150
## 8 Jaan   65   180    1  pulss30m   120
## 9 Jüri  100   190    1  pulss30m    NA
```

Funktsioon `dcast()` aitab andmeid pikast formaadist laia teisendada, sealjuures tuleb argumendiga formula kindlasti ette öelda, millised tunnused määravad ära tulemusandmestiku read ning millised määravad ära veerud: `formula = reatunnus1 + reatunnus2 ~ veerutunnus1 + veerutunnus2`. Väga kasulik on argument `fun.aggregate`, mille abil saab määrata, kas ja millist funktsiooni peaks kasutama veerutunnustega antud väärtuste agregatsiooniks. Funktsiooniga `recast()` saab korrigeerida `melt()` ja sejärel `dcast()` käsku.

```
dcast(m, formula = nimi ~ variable)
```

```
##   nimi pulss0m pulss10m pulss30m
## 1 Jaan      80      120      120
## 2 Jüri      80      190       NA
## 3 Mari      70      130      150
```

```
dcast(m, nimi + kaal ~ variable + sugu)
```

```
##   nimi kaal pulss0m_1 pulss0m_2 pulss10m_1 pulss10m_2 pulss30m_1 pulss30m_2
## 1 Jaan   65        80        NA       120        NA       120        NA
## 2 Jüri  100        80        NA       190        NA        NA        NA
## 3 Mari   68        NA       70        NA       130        NA       150
```

```
dcast(m, nimi ~ ., fun.aggregate = mean, na.rm = TRUE, value.var = "value")
```

```
##   nimi
## 1 Jaan 106.6667
## 2 Jüri 135.0000
## 3 Mari 116.6667
```

1.4.1 Ülesanded

1. Kasuta arstivisiitide ja isikute andmestiku ühte alamosa:

```
valik <- read.table("http://kodu.ut.ee/~annes/Rkursus/valik.txt", sep = "\t", header = TRUE).
```

Vaata andmestik üle `head(valik)`. Tegu on valikuga isikutest, tunnuste osas on lisatud visiidi järjekorranumber.

- Vii andmed laiale kujule, kus iga isiku jaoks oleks andmestikus üks rida, kus on kirjas isikukood, sugu, sünnikuupäev, vanus ning vererõhu väärtus esimesel ja teisel visiidil.
- Vii andmed laiale kujule, kus iga isiku jaoks oleks andmestikus üks rida, kus on kirjas isikukood, sugu, sünnikuupäev, vanus, vererõhu väärtus esimesel ja teisel visiidil ning crv väärtus esimesel ja teisel visiidil.

2. Kasuta arstivisiitide andmestikku.

```
visiidid <- read.table("http://kodu.ut.ee/~annes/Rkursus/visiidid.txt", sep = "\t", header = TRUE)
```

Leia iga isiku keskmine vererõhk ja CRV.

3. Võta aluseks visiitide ja isikuandmete ühendatud andmestik, ning vii andmestik laiale kujule nii, et igas reas oleks ühe isiku isikukood, sugu, sünnikuupäev, vanus ning edasi crv väärtused vastavalt esimesel, teisel, kolmandal jne visiidil.