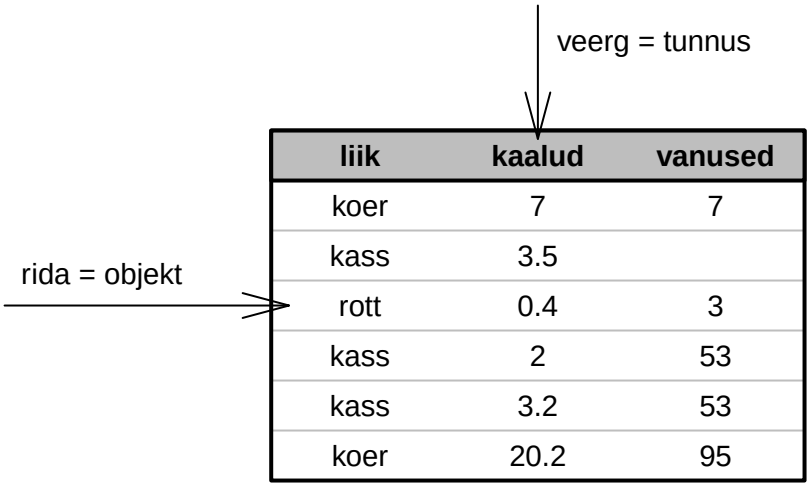
Rakendustarkvara: R. Sügis 2017, 2. praktikum*

1 Andmestik, andmete import

Tavaliselt koosnevad andmestikud ridadest ja veergudest (tulpadest), kus iga rida vastab mingile mõõtmisobjektile ja iga veerg vastab mingile mõõdetud omadusele (tunnusele).



liik	kaalud	vanused
koer	7	7
kass	3.5	
rott	0.4	3
kass	2	53
kass	3.2	53
koer	20.2	95

Erinevad statistikaprogrammid kasutavad andmestike säilitamiseks eri failiformaate. Et andmestikke ühest programmist teise saada, on üheks võimalikuks lahenduseks andmetabel salvestada vahepeal tekstiformaadis failiks, näiteks `.txt` või `.csv` tüüpi failiks ja importida tekstifail.

MS Exceli faile ning statistikatarkvarade spetsiifilisi andmeformaate (nt Stata `.dta`, SPSS-i, `.sav`) ei saa R-i importida baaspaketi käskude abil. Selliste failide impordiks on vaja kasutada mõnda R-i lisapaketti.

Tänases praktikumis vaatame nii tekstifailis andmete kui ka mõne teise statistikaprogrammi andmefailide importimist R-i.

1.1 Andmete sisselugemine ja faili kirjutamine (tekstifail)

Kõige olulisem käsk tekstikujul andmestike sisselugemiseks on `read.table()`. Sellel on mitmeid **argumente**, mille abil saab täpsustada erinevaid asjaolusid, mis antud andmestikku puutuvad:

`file`	-	faili asukoht ja nimi koos faililaiendiga (peab olema jutumärkides)
`header`	-	kas faili esimeses reas on veerunimed? (TRUE = jah, FALSE = ei)
`sep`	-	millise sümboliga on veerud eraldatud? (nt `"\t"` - tabulaatori sümboliga)
`dec`	-	kuidas on märgitud kümnendmurru eraldaja? (nt `"."` või `","`)
`quote`	-	millega on tekstilised väärtused ümbritsetud? (nt `"\""` tähendab, et kahekordsete jutumärkidega)
`dec`	-	kuidas on märgitud kümnendmurru eraldaja? (nt `"."` või `","`)
`na.strings`	-	kuidas on puuduvad väärtused tähistatud?
`fileEncoding`	-	mis tähekodeeringut kasutatakse (Windowsis sageli `"latin9"`, Macis/Linuxis sageli `"utf8"`)

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

Faili asukoht võib olla kõvakettal või võrgus. Kui käsu `setwd` abil on määratud, milline on käimasoleva töösessiooni **töökataloog**, ja andmefail on selles kataloogis, siis piisab ainult faili nime andmisest (täispikka asukohta ei pea andma). Windowsis on kombeks kaustastruktuuri tähistamiseks kasutada kurakaldkriipsu (tagurpidi kaldkriipsu) `\`, aga R-is on sellisel kaldkriipsul eriline tähendus – sellega märgitakse, et järgneb erisümbol (nt `\t` on tabulaatori sümbol). Seepärast tuleb kataloogitee märkimisel kasutada kahekordseid kurakaldkriipse, näiteks:

```
setwd("C:\\Users\\mina\\Rkursus\\")
```

Teine võimalus on kasutada tavalist kaldkriipsu, nagu MacOS-is ja Linuxites:

```
setwd("C:/Users/mina/Rkursus/")
```

Andmete sisselugemisel tuleks anda andmetabelile nimi (salvestada see mingi objektina), vastasel juhul andmestik trükitakse lihtsalt ekraanile ja sellega enam midagi muud teha ei saa:

```
näide1 <- read.table("http://kodu.ut.ee/~annes/Rkursus/esimene.txt", sep = "\t", dec = ",")
```

Töökeskkonnas olevaid andmetabeleid saab faili kirjutada käsuga `write.table(.)`, andes käsule ette salvestatava andmestiku nime ning loodava faili nime jutumärkides (vajaduse korral koos kataloogiteega):

```
write.table(näide1, "failinimi.txt", sep = "\t")
```

1.1.1 Ülesanded

1. Vaata `read.table` käsu argumentide täielikku loetelu abifailist.
2. Aadressil http://www.ut.ee/~annes/Rkursus/ leiad failid *tabel1.csv*, *tabel2*, *tabel3.txt*, *tabel4.tab*. Tutvu nendega (Notepadi vmt kasutades) ja proovi need seejärel R-i korrektselt sisse lugeda. Veendu, et R-is olevates tabelites on sama palju veerge kui originaalandmestikes.
3. USA riiklikud institutsioonid võimaldavad päris sageli andmekogudele vaba ligipääsu. Selles aines kasutame näidisandmestikena USA Rahvaloendusbüroo[1] poolt kogutud andmeid, millele saab ligi IPUMS-USA[2] liidese kaudu[3]. Aadressil http://www.ut.ee/~annes/Rkursus/ on fail *mass.txt*, milles on Massachusettsi osariigis ühe valikuuringuga kogutud andmed. Loe see R-i sisse, andmestiku objektile pane nimeks `andmed`. Tutvu ka tunnuste kirjeldusega: http://www.ut.ee/~annes/Rkursus/descr.txt. Andmestikuobjektist esmase ülevaate saamiseks kasuta käsku `str(andmed)`.

```
##   HIV Age Gender   Süstijadef
## 1   1  NA      1      süstija
## 2   0  38      1 mittesüstija
```

```
##   HIV Age Gender   Süstijadef
## 1   1  NA      1      süstija
## 2   0  38      1 mittesüstija
```

```
##   HIV Age Gender   Süstijadef
## 1   1  NA      1      süstija
## 2   0  38      1 mittesüstija
```

```
##   HIV Age Gender  S.ustijadef
## 1   1  NA      1     s"ustija
## 2   0  38      1 mittesüstija
```

[1] http://www.census.gov/data.html

[2] Steven Ruggles, J. Trent Alexander, Katie Genadek, Ronald Goeken, Matthew B. Schroeder, and Matthew Sobek. Integrated Public Use Microdata Series: Version 5.0 [Machine-readable database]. Minneapolis, MN: Minnesota Population Center [producer and distributor], 2010.

[3] https://usa.ipums.org/usa-action/variables/group

1.2 Lisapakettide kasutamine

Üks R-i populaarseks muutumise põhjuseid on rikkalik lisapakkettide olemasolu. Tõenäoliselt leidub iga praktilise statistika-alase (ja ka mõne muu valdkonna) probleemi jaoks omaette pakett (*package*). Näiteks paketid `foreign`, `readstata13`, `haven`, `readxl` sisaldavad funktsioone, mis aitavad teistes kui tekstiformaatides andmestikke hõlpsamini R-i sisse lugeda. Andmete impordiks sobilikke lisapakette on lisaks nimetatutele veel. Siinkohal vaatame aga kahe paketi `haven` ja `readxl` võimalusi.

Lisapaketid ei ole tavaliselt R-iga kaasas, vaid need tuleb installeerida. Kui paigaldatav pakett vajab omakorda mingeid muid pakette, siis installeeritakse ka need. Kui kasutatakse R-i installeerimisel paika pandud vaikeseadistusi, siis töösessiooni korral esimest korda mingit paketti installeerides küsitakse, millisest serverist soovib kasutaja neid alla laadida. Järgmistel kordadel sama töösessiooni jooksul pakette paigaldades seda enam ei küsita.

Installeerime lisapaketid `readxl` ja `haven`

```
install.packages("readxl")
install.packages("haven")
# NB! Arvutiklassi arvuti korral võib olla vajalik salvestuskoha ettemääramine
# install.packages("readxl", lib = "C:/kataloog-installimiseks")
```

Kui pakett on installitud, siis järgmisel töösessioonil seda enam uuesti installeerima ei pea. Arvutisse paigaldatud paketid tuleb iga kord uut R-i sessiooni alustades kasutamiseks sisse laadida käsuga `library` (või `require`), mille argumendiks on laaditava paketi nimi:

```
library(readxl)
# Kui installimisel oli käsitsi määratud salvestuskoht:
# library(readxl, lib.loc = "C:/kataloog-installimiseks")
```

1.3 Andmete import programmide MS Excel, SAS, Stata, SPSS failidest

1.3.1 MS Excel failid

Lisapaketi `readxl` käsu `read_excel(.)` abil saab importida MS Exceli failide (nii *.xlsx* kui *xls*) töölehti. Salvesta aadressilt http://www.ut.ee/~annes/Rkursus/ oma töökataloogi fail *tudengite-arv.xlsx*, mis sisaldab infot TÜ tudengite arvude kohta aastate lõikes. Ava esmalt fail MS Exceli abil ning vaata millised töölehed failis on. Seejärel vaata ka R abil millised on töölehtede nimed ja impordi teine tööleht, millel on andmed avatud ülikooli õpilaste arvude kohta.

```
list.files() # vaata, mis nimega failid on töökaustas
excel_sheets("tudengite-arv.xlsx") # töölehtede nimed MS Exceli failis
AY <- read_excel("tudengite-arv.xlsx", sheet = "avatud ylikool") # importimine
# AY <- read_excel("tudengite-arv.xlsx", sheet = 2)  # sama
str(AY)  # vaata tulemust
```

MS Exceli failis ei pruugi andmed alata alati esimesest reast, näiteks töölehel *tabel* on esimeses reas tabeli nimi ning teine rida on tühi, andmetabel algab kolmandalt realt. Selle, et esimesed kaks rida tuleks andmete impordil vahele jätta, saab käsule `read_excel(.)` ette anda argumendi `skip` abil:

```
tabel <- read_excel("tudengite-arv.xlsx", sheet = "tabel", skip = 2)
str(tabel)
```

1.3.2 SAS, Stata ja SPSS failid

Programmide SAS, Stata ja SPSS failide importimiseks saab kasutada paketi `haven` vastavaid funktsioone `read_sas(.)`, `read_stata(.)` ja `read_spss(.)`.

```
library(haven)
andmestik_SAS <- read_sas("effort.sas7bdat")
andmestik_Stata <-  read_stata("effort.dta")
andmestik_SPSS <- read_spss("effort.sav")

str(andmestik_SAS)
str(andmestik_Stata)
str(andmestik_SPSS)
```

R-i andmestiku saab `haven` paketi abil ka eksportida vaadeldud programmide andmefailiks käsuga `write_<...>(.)`, kus `<...>` tuleks siis asendada sobiliku programminimega.

1.3.3 Ülesanded

1. Impordi failist *tudengite-arv.xlsx* viimane tööleht *kokku*, kus on tudengite arvud aastate lõikes (liidetud statsionaar ja avatud ülikool). Vaata esmalt ka abilehte kasutatava funktsiooni kohta: `?read_excel`

2 Lihtsamad toimingud andmestikuga

2.1 Andmestiku kirjeldamine.

Käsuga `read.table(.)` sisse loetud andmestik on erilist tüüpi, **`data.frame`**-tüüpi objekt. Andmestikust saab kiire ülevaate järgmiste käskudega:

`nrow(andmed)`	-	mitu rida on andmestikus
`ncol(andmed)`	-	mitu veergu on andmestikus
`dim(andmed)`	-	mitu rida ja mitu veergu on andmestikus
`str(andmed)`	-	andmestiku struktuur: mis tüüpi iga tunnus on ja millised on mõned esimesed väärtused
`summary(andmed)`	-	lühike kirjeldav statistika iga tunnuse kohta
`names(andmed)`	-	veergude nimed
`head(andmed)`	-	andmestiku mõned esimesed read

Käsu `summary(.)` puhul on näha, et mõne tunnuse puhul arvutatakse keskmine, miinimum, maksimum jne, aga teise tunnuse puhul esitatakse sagedused. See, millist kirjeldusviisi kasutatakse, tuleneb tunnuse tüübist. Kui käsuga `str(.)` vaadata, mis on tunnuste tüübid selles andmestikus, on näha kahte tüüpi tunnuseid: `int` ja `Factor`. On näha, et `summary(.)` käsk arvutab `int`-tüüpi tunnustele keskmisi jne, `Factor`-tüüpi tunnustele aga sagedusi. Faktortüüpi tunnus sarnaneb sõnele (*character*), kuid üheks erinevuseks on näiteks fikseeritud väärtuste hulk. Faktortunnust vaatame selles praktikumis ka täpsemalt.

2.2 Veergude ja ridade eraldamine andmestikust: indeksi ja nime järgi

Kui tahame andmestikust ainult üht veergu uurida, siis kõige mugavam on kasutada dollari-sümbolit:

```
vanused <- andmed$AGEP
median(andmed$AGEP)
median(vanused)
```

Üldiselt on `data.frame` kahemõõtmeline tabel, mis tähendab, et iga elemendi asukoht selles tabelis on ära määratud rea ja veeru numbriga. Rea- ja veerunumbrite abil andmestikust infot eraldades tuleb kasutada kantsulgusid:

```
andmed[3, 2]  # kolmas rida, teine veerg
andmed[ , 2]  # kogu teine veerg
andmed[3,  ]  # kogu kolmas rida
```

Korraga on võimalik eraldada ka mitut rida või veergu, kasutades selleks käsku `c(.)`:

```
andmed[, c(2, 4)]  # teine ja neljas veerg
valik <- c(2, 4)  # tekitame objekti, milles on kirjas huvipakkuvate veergude numbrid
andmed[, valik]  # kasutame seda objekti andmestikust veergude eraldamiseks
andmed[c(5, 3,  9), ]  # viies, kolmas ja üheksas rida
```

Tihti on veeruindeksite asemel mugavam kasutada veergude nimesid, mida samuti kantsulgudes kasutada (peavad olema jutumärkides):

```
andmed[, c("AGEP", "WAGP")]  # eraldame veerud "AGEP" ja "WAGP"
```

Kui andmestikus on ka ridadel nimed, siis saab neid sama moodi kasutada ridade eraldamisel. Selle läbiproovimiseks lisame oma andmestikule praegu ise reanimed:

```
rownames(andmed) <- paste("rida", rownames(andmed), sep = "-") # paneme ise nimed kujul 'rida-jrk'
head(andmed[, 5:9])  # vaatame milline on tulemus
andmed[c("rida-23", "rida-62"), 5:9]  # eraldame read 23 ja 62
```

2.2.1 Uue tunnuse lisamine

Uue tunnuse lisamiseks andmestikku tuleb valida tunnuse nimi, mida andmestikus veel ei esine, ja omistada uuele andmeveerule valitud väärtused. Lisame oma andmestikku näiteks keskmise kuusissetuleku veeru, mis on arvutatud olemasoleva 12 kuu sissetuleku tunnuse `WAGP` abil. Uus tunnus lisatakse viimaseks veeruks andmestikus:

```
andmed$kuusissetulek <- andmed$WAGP/12
# või
andmed[, "kuusissetulek"] <- andmed$WAGP/12
str(andmed)
```

2.3 Veergude ja ridade eraldamine andmestikust: tõeväärtusvektori abil

Tõeväärtusvektoreid on väga mugav kasutada nn filtritena. Nimelt on `data.frame` puhul võimalik ridu (ja ka veerge!) eraldada mitte ainult indeksi või -nime järgi, vaid ka tõeväärtusvektori abil. Kui eraldada ridu, siis vastav tõeväärtusvektor peab olema sama pikk kui on andmestikus ridu ning väärtused selles vektoris näitavad, kas vastavat rida kasutada (`TRUE`) või mitte(`FALSE`). Tõeväärtusvektorite kombineerimisel saab andmestikust väga spetsiifilisi alamhulki eraldada.

```
# eraldame kõik read, kus SEX == "Male" ning salvestame selle uueks objektiks
mehed <- andmed[andmed$SEX == "Male", ]

# moodustame kaks filtritunnust ja kombineerime need alagrupi valikuks
filter_kod <- andmed$CIT == "Not a citizen of the U.S."  # mittekodanikud
filter_vanus <- andmed$AGEP >= 80                        # vähemalt 80 aastased
alamandmestik <- andmed[filter_kod & filter_vanus, ]   # Ära unusta: [read, veerud]
```

Üks näide ka tõeväärtusvektori abil veergude eraldamise kohta: valime kõik need veerud, mille päis algab sõnega *MAR* (tunnused, mis seotud abieluga)

```
onMAR <- substr(names(andmed), 1, 3) == "MAR"
abielu <- andmed[, onMAR]
str(abielu)
```

2.3.1 Ülesanded

1. Selekteeri andmestikust iga 5. rida ja salvesta see alamandmestik uue nimega `valik5`. Mitu vaatlust on selles andmestikus?
2. Moodusta alamandmestik, kuhu kuuluvad uuritavad, kelle kohta pole teada, kas nad on viimase aasta jooksul elukohta vahetanud.

2.4 Lihtsam kirjeldav statistika

Allpool on loetletud mõned käsud, mille abil mõnd konkreetset tunnust (veergu andmestikust) kirjeldada.

- `min(tunnus)`, `max(tunnus)`, `median(tunnus)`, `mean(tunnus)`, `sd(tunnus)` – arvulise tunnuse karakteristikud
- `quantile(tunnus, kvantiil)` – saab leida kvantiile ehk protsentiile arvulisele tunnusele
- `length(tunnus)` – mitu elementi on antud veerus
- `table(tunnus)` – saab koostada sagedustabelit (kasulik `Factor`-tüüpi tunnuse kirjeldamisel)
  `table(tunnus1, tunnus2)` – koostab kahemõõtmelise sagedustabeli
- `t(tabel)` – vahetab tabeli read ja veerud (transponeerib)
- `ftable(tunnus1 + tunnus2 ~ tunnus3 + tunnus4, data = andmed)` – teeb mitmemõõtmelise sagedustabeli, mis on inimesele lihtsasti loetav

Kui on soov korraga mitme arvulise tunnuse keskmisi arvutada, sobib selleks käsk `colMeans(andmetabel)`, sarnane käsk on `rowMeans(.)`. Ridade või veergude summasid saab leida käskudega `rowSums(.)` ja `colSums(.)`. Seda saab näiteks kasutada sagedustabeli põhjal protsentide arvutamiseks:

```
sagedustabel <- table(andmed$SEX, andmed$LANX)
sagedustabel / rowSums(sagedustabel)  #proovi, mis juhtub, kui kasutada /colSums(.)
```

```
##
##          No, speaks only English Yes, speaks another language
##   Female               0.8031949                    0.1968051
##   Male                 0.8147019                    0.1852981
```

Sagedustabeli põhjal protsentide arvutamiseks (jaotustabeli arvutamiseks) on eelnevalt näidatud konstruktsioonist mugavam kasutada käsku `prop.table(.)`:

```
prop.table(sagedustabel)  # ühisjaotus
prop.table(sagedustabel, margin = 1)  # iga rida kokku 1 (ehk 100%)
prop.table(sagedustabel, margin = 2)  # iga veerg kokku 1 (ehk 100%)
```

2.4.1 Ülesanded

Kõik ülesanded on Massachusettsi andmestiku kohta.

1. Mitu protsenti vastajatest on mehed?
2. Milline on palk, millest väiksemat palka saab 80% inimestest?
3. Kas lahutatute osakaal on suurem meeste või naiste hulgas?
4. Mitu üle 74 aasta vanust doktorikraadiga naist on andmestikus?
5. Mitmel inimesel on bakalaureuse-, magistri- või doktorikraad?

6. Milline on keskmine aastapalk meestel, milline naistel?
7. Kui suur osa (protsentides) ilma kõrghariduseta inimestest saab suuremat aastapalka kui keskmine palk?
8. Kas Massachusettsi andmete alamandmestikus `valik5`(tekitatud eelmises ülesanneteplokis) on meeste ja naiste keskmised aastapalgad samasugused kui kogu andmestikus?

2.5 Faktortunnus

Faktortunnus pole tegelikult nn elementaartüüp (nagu näiteks `integer`), vaid keerulisem konstruktsioon. Nimelt faktortunnus on sildistatud koodide tunnus. Kui andmestik R-i sisse loetakse, siis vaikimisi pannakse tähelisi väärtuseid sisaldavate tunnuste tüübiks `factor` (seda võib ka `read.table(.)` argumendiga `stringsAsFactors` keelata) ning iga erinev väärtus kodeeritakse mingi täisarvuga, aga lisaks tehakse kodeerimistabel, kus on kirjas iga täisarvu (kodeeringu) tekstiline väärtus (ehk silt).

Et teada saada, mitu erinevat väärtust antud faktortunnusel võib üldse olla, kasutatakse käsku `levels(.)`. Sealjuures ei pruugi kõiki faktori väärtustasemeid antud andmetes üldse esineda:

```
levels(mehed$SEX)
```

```
## [1] "Female" "Male"
```

```
table(mehed$SEX)
```

```
##
## Female   Male
##      0   3125
```

Faktortunnuse tekitamiseks saab kasutada käsku `factor(.)`. Vaikimisi pannakse faktortunnuse tekitamisel faktori tasemed tähestiku järjekorda. Seda saab aga muuta käsu `factor(.)` argumenti `levels` kasutades:

```
table(andmed$MARHT)   # Mitu korda abielus olnud?
andmed$MARHT <- factor(andmed$MARHT, levels = c("One time", "Two times", "Three or more times"))
table(andmed$MARHT)
```

Mõnikord on imporditud andmestikus arvulised tunnused faktorkujul sellepärast, et sisseloetavas failis oli viga ning ühes lahtris oli arvu asemel mingi tekst. Kui nüüd proovida `as.numeric(.)` käsuga see tunnus arvuliseks teisendada, tekib segadus: `Factor`-tüüpi tunnus on juba täisarvude tunnus (kuigi neil on sildid juures) ning seetõttu antakse tulemuseks need täisarvud ehk kodeeringud. Segadust ei teki, kui faktortunnus kõigepealt sõneks teisendada (kodeeringud kaotatakse, jäävad ainult sildid) ning alles seejärel arvudeks.

```
(x <- factor(c("1", "8", "ei vastanud", "12"))) # välimised sulud tingivad väljatrüki
```

```
## [1] 1           8           ei vastanud 12
## Levels: 1 12 8 ei vastanud
```

```
as.numeric(x)
```

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

```
as.numeric(as.character(x))
```

```
## Warning: NAs introduced by coercion
```

```
## [1]  1  8 NA 12
```

Mõnikord soovime arvulist tunnust muuta nn ordinaaltunnuseks (st selliseks, kus on mõned üksikud kategooriad, mis on omavahel järjestatud). Näiteks palkade statistika esitamisel on soov teada infot palgavahemike kaupa. Arvulist tunnust aitab lõikudeks tükeldada käsk `cut(.)`. Selle käsu tulemusel tekib faktortunnus, mille silte saab `cut(.)` käsu argumendiga `labels` ette anda:

```r
palgad <- cut(andmed$WAGP, breaks = c(0, 999, 4999, Inf), include.lowest = T,
              labels = c("0-999", "1000-4999", ">= 5000"))
table(palgad)
```

```
## palgad
##     0-999 1000-4999   >= 5000
##      1903       296      3114
```

2.5.1 Ülesanded

1. Tekita tunnus, kus oleks kirjas, millisesse vanusgruppi inimene kuulub: 0–17, 18–49, 50–64, 65 või vanem.
2. Kas USA kodakonsust mitteomavate naiste ja meeste hulgas on vanus erinevalt jaotunud?