

## Sissejuhatus

Arvutusteenika levik ja kiire areng tõi kaasa ka statistikatarkvara plahvatusliku leviku. Ilmusid mitmed esimesed statistikapaketid – suuresti halvastiintegreeritud statistikaprotseduuride kogumikud – nagu SAS, SPSS. Peagi tekkisid juurde ka arvutivõõramatele inimestele mõeldud menüüvalikuid pakkuvaid pakette, nagu Statgraphics või Statistica eelkäijad (PsychoStat jt). Arvutused ja joonised saadi kuidagi tehtud, aga ühtegi tõeliselt head tööriista ikka nagu polnud.

Siis aga tulid kokku mõned andekad inimesed (Richard A. Becker, John M. Chambers jt.) ning unistasid statistikule meelepärasest keskkonnast. Oma unistuse panid nad kirja ja avaldasid S-keele kirjelduse nime all 1984. aastal.

Teoreetilisest kirjeldusest haarasid kinni erinevad meeskonnad ja üritasid selle põhjal luua (esialgu teatavate raskustega) töötavat arvutiprogrammi. Täna maailmas on laialt levinud kaks S-keele definitsiooni põhjal loodud programmi: suure raha eest müüdiv S-Plus ja vabavaraline R (esimene versioon aastal 2000). Hetkel (oktoober 2009) on viimane R'i versioon 2.10. Mainitud programmid on kujunenud de facto standardiks teadustööga tegelevate statistikute seas.

R on programmeeritav (tehtud tööst jääb järgi jälg ehk lihtne on tehtud dokumenteerida ja automatiseerida); kohandatav (igauks võib lisada endale meelepäraseid funktsioone/andmetüüpe, võimalik kujundada just antud ülesande jaoks parim graafik jne). R'i ühe puudusena võib mainida teatavaid raskuseid töös väga suurte andmestikega. Tõeliselt suurte andmestike korral võib osutuda mõistlikuks kombineerida tööd mõne teise andmebaasi (Oracle, MySQL, ...) või töövahendiga (Perl, Python...) ning lugeda andmeid R'i jupikaupa (näiteks vaid need tunnused, mida hetkel analüüsiks vaja on).

### Kuidas saada endale R?

R on vaba tarkvara. Uusima versiooni programmist võib maha laadida R-i koduleheküljelt <http://www.r-project.org/> (*Download-CRAN – <vali server> – Windows – base*). Samuti võib R-i koduleheküljelt leida ka ingliskeelse raamatu algajaile (*Help – Manuals – An Introduction to R*) ja spetsiaalsete ülesannete jaoks loodud statistikamooduleid, mis vaikimisi koos R-iga ei installeeru (hetkel saadaval rohkem kui 500 lisamoodulit. Bioconductor'i nime alla on koondatud hunnik lisamooduleid DNA-mikrokiipidelt pärit andmete (ja teiste geenandmete) analüüsimiseks; ruumiandmete analüüsimiseks on mõeldud omad lisamoodulid (gstat; geoRglm jt) jne. Programmi R on võimalik kasutada Windowsi, Linuxi, MacOS X ja paljude teiste operatsioonisüsteemide peal. Standardinstallatsioon võtab umbes 60 Mb ruumi, koos täiendavate lisamoodulitega märgatavalt rohkem

## Kirjandust

Põhjalikumat ülevaadet kui käesolev sissejuhatav material pakkuda suudab võib leida järgmistest allikatest:

- [www.r-project.org](http://www.r-project.org) alt on võimalik leida algajaile mõeldud raamatut (*Help – Manuals – An Introduction to R*). Soovitav on iseseisvalt läbi proovida peatükis *Sample session* toodud käsud. R'i koduleheküljelt on võimalik kätte saada ka mitmeid teisi asjalikke raamatuid, vaata menüüd *Documentation-Contributed*.
- Peter Dalgaard (2002). *Intrductory Statistics with R*. Springer-Verlag. Heas stiilis raamat ühelt R-i loojatest.
- <http://www.ms.ut.ee/mart/R/Rgraafika.html> – näited graafikute joonistamisest R-is.

## Alustuseks

### R kui kalkulaator

R oskab arvutada. Seda väidet saab kontrollida, sisestades näiteks järgmine käsk (viipa „>“ pole tarvis sisestada, rea lõpus vajutage ENTER-klahvile):

```
> (2+3)*6  
[1] 30
```

Tehteid saab teha (ja paljusid funktsioone kasutada) ka tervete arvujadadega ehk vektoritega korraga. Järgmise käsuga palume ruutjuured arvudest ühest kümneni:

```
> sqrt(1:10)  
[1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490 2.645751 2.828427  
[9] 3.000000 3.162278
```

Näiteid käskudest ja funktsioonidest, mida R tunneb:

```
> 2**8 või 2^8           – astendamine, 28;  
> sin(0.5*pi)           – näide trigonomeetrilise funktsiooni kasutamisest;  
> log(exp(10))          – funktsioon log leiab naturaallogaritmi arvust;
```

### Muutujad

Ka kalkulaatorit kasutades tekib peatselt vajadus meeles hoida arvutuste tulemusi. R'is, nagu paljudes teistes programmeerimiskeelteski, võib kasutada sümboleid või sõnu (objektide nimesid) väärtuste hoidmiseks. Näiteks saab omistada  $x$ -le väärtuse 3 järgmise käsu abil:

```
> x=3
```

Selle käsu saamisel teostab R omistamise. Kui kõik läheb hästi, ei ilmu sealjuures ekraanile midagi. Edaspidi saab juba oma töös kasutada  $x$ -i samamoodi kui numbrit 3:

```
> x
[1] 3
> x+5
[1] 8
```

Meelde jätta võib muudki, kui vaid ühte arvu –  $x$  võib olla näiteks terve arvude vektor (kümne hiire kaalud); andmestik (saja katsehiire nimed, kaalud, pikkused, geneetiliste markerite väärtused,...); regressioonanalüüsi tulemused koos jääkide, hinnatud parameetrite väärtuste ja muu juurdekuuluvaga; samuti võib ta olla funktsioon, mis arvutab Gini kordajat ja palju muudki.

Seda, milliseid nimesid oled objektide tähistamiseks kasutanud, saab vaadata käsu `ls()` abil:

```
> ls()
[1] "x"
```

Kasutu objekti võib kustutada käsuga `rm()`:

```
> rm(x)
```

R lubab omistamisel kasutada ka tema enda käskude nimesid. Juhul, kui R'i mõni käskudest ümber defineeriteakse, võib tulemuseks olla palju segadust. Sestap tuleks võimaluse korral vältida R'i käskude („`c`“, „`t`“, „`lm`“, „`ls`“ jne) kasutamist muutujanimedena. Muutujate nimedeks sobivad hästi näiteks eestikeelsed salvestatud objekti kirjeldavad sõnad (RebasteArv, mudel2 jne).

**Tähtis! R teeb vahet suurte ja väikeste tähtede vahel.** Seega on „`x`“ ja „`X`“ kaks erinevat objekti. Samuti annab käsk `SQRT(2)` veateate, sest ruutjuure leidmise funktsioon on `sqrt` (väikesed tähed!).

### Abiinfo (help)

Juhul, kui tead küll funktsiooni nime, mida kasutada soovid, kuid sooviks siiski täiendavat informatsiooni tema kohta, tipi „`?`“ ning sind huvitav funktsioon sinna järgi:

```
> ? median
```

R näitab seepeale antud käsu süntaksit ja näiteid sisaldavat ekraani.

## Programmi kirjutamine ja tehtud töö dokumenteerimine

Kuigi lihtsamaid käske ja arvutusi võib sisestada R-is otse käsurealt, on enamasti targem kirjutada käsud/kommentaariid eelnevalt valmis ja alles seejärel kirja pandud käsud käivitada. See võimaldab ühelt poolt vigade olemasolul vigu kergesti parandada ja uuesti katsetada, teiselt poolt võimaldab tehtavat tööd paremini dokumenteerida. Võib kasutada näiteks R-i enda editorit (*File -> New Script*) või ka mõnda välist tekstiredaktorit (näiteks notepad'i). Juhul, kui kasutate R'i enda editorit, saab soovitud käske kergesti käivitada – vali vaid programmilõik mida käivitada soovid ja vajuta CTRL+R.

Kõiki töö käigus R-le antud käske saab salvestada valides menüüst *File -> Save History* (aken „R Console“ peab olema aktiivne). Peale salvestamist tekib *.Rhistory* – lõpuga tekstifail, mida saab tekstieditoriga (näiteks notepad) avada ja redigeerida. Antud hetkeni R-le antud käsud koos vastustega saab tekstifaili salvestada aktiveerides akna „R Console“ ja valides menüüst *File->Save to File*.

Kõiki töö käigus tehtud muutujaid, andmestikke jms saab korraka salvestada valides menüüst *File -> Save Workspace*. Kogu tekitatud muutujate, andmestike ja muu töökeskkonna saab siis hiljem taastada valiku *Load Workspace* abil.

Vaikimisi salvestatakse ja hakatakse faile otsima töökataloogist. Mõistlik on teha iga projekti (näiteks Monte-Carlo meetodite kursuse) jaoks oma alamkataloog ja tööd alustades seada töökataloog viitama mainitud kataloogile. Seda saab teha kas valides menüüst käsu *File -> Change dir* või kasutades käsku *setwd*.

## Ülesanded

1.  $\frac{1}{1+0,00022} - (1 - 0,00022) = ?$
2.  $1,79451 + 1,79451^{1,79451} + \sin(1,79451) = ?$
3. Uuri välja, mida teeb funktsioon *apropos* – alustuseks proovi näiteks *apropos("cos")*.

## Andmed ja nende vaatamine

Andmestikke, muutujaid ja muud (näiteks hinnatud mudeleid jms) saab salvestada ja taas lugeda käskude *save* ja *load* abil. Näiteks, kui meil oleksid R'is olemas juba andmestikud *andmestik1* ja *andmestik2*, saaksime need salvestada faili *minu* järgmise käsu abil: `save(andmestik1, andmestik2, file="c:/minu")`. Failis „minu“ olevaid objekte (andmestike, muutujaid jms) saaks aga näiteks R'i sisse lugeda aga käsuga: `load("c:/minu")`.

NB! Pange tähele – R-is tuleb failiteed anda nn UNIX'i stiilis, st tagurpidi kaldkriipse (/) kasutades.

Alustuseks loeme aga sisse ühe pärisandmestiku – Tartu Ülikooli meditsiiniteaduskonna tudengite küsitluse andmed (Antud küsitluse ankeeti võite näha aadressil <http://www.ms.ut.ee/mart/MC2007/ankeet.pdf>):

```
> load(url("http://www.ms.ut.ee/mart/MC2007/kokku.Rdata"))
```

Andmed on nüüd R-is, aga kuidas neid näha?

Võimalused:

1. tippige R-i andmestiku nimi (vajadusel vaadake *ls()*-käsu abil, mis nimega andmestikud/objektid tekkisid R-i peale load-käsku!):

```
> kokku
```

Andmestik jookseb silme eest mööda, aga eriti targemaks vist ei saanud?

2. Andmeid saab vaadata ja muuta ka *edit*-käsu abil. Vaatamiseks:

```
> edit(kokku)
```

Muutmiseks (muudetud andmestik salvestatakse nime *kokku2* all, esialgne andmestik *kokku* jääb ka alles):

```
> kokku2 = edit(kokku)
```

3. Andmestikus sisalduvate tunnuste nimesid saab vaadata ka käsu *names()* abil:

```
> names(kokku)
 [1] "aasta"      "vanus"      "sugu"       "perekonnaseis"
 [5] "toop.ohk"  "puhkep.ohk" "sport"     "suitsetamine"
 [9] "olu"       "vein"      "viin"      "tervis"
[13] "viirus"    "kiirabi"   "haiglaravi" "toolt.puudu"
[17] "pikkus"    "kaal"      "SVR"       "DVR"
```

4. Võime ka välja trükkida paar esimest (funktsioon *head*) või viimast (funktsioon *tail*) kirjet:

```
> tail(kokku)
      aasta vanus sugu perekonnaseis toop.ohk puhkep.ohk sport
656  2001    27    1           2        NA         NA     2
657  2001    20    1           1        NA         NA     2
658  2001    19    1           1        NA         NA     2
659  2001    19    1           1        NA         NA     2
660  2001    19    1           1        NA         NA     2
661  2001    20    1           1        NA         NA     2
      suitsetamine olu vein viin tervis viirus kiirabi haiglaravi
656           1    1    3    2    3    1    NA    NA
657           1    1    2    1    1    3    NA    NA
658           1    1    2    3    2    4    NA    NA
659           1    1    2    2    2    4    NA    NA
660           1    2    3    2    3    1    NA    NA
661           1    3    3    2    3    1    NA    NA
      toolt.puudu pikkus kaal SVR DVR
656           NA 173.0 73.0 120 83
657           NA 166.5 57.5 114 76
658           NA 167.0 52.0 100 88
659           NA 182.5 74.0 127 88
660           NA 170.0 63.0 111 76
661           NA 154.0 51.0 108 63
```

5. Samuti võime andmestikust kiirülevaate saada *summary*-käsu abil, proovi näiteks käsku *summary(kokku)*.

## Andmete väljanappimine andmestikust

Ühe tunnuse väärtuseid saab kätte kasutades süntaksit `<andmestiku nimi>${<tunnuse nimi>}`; näiteks tudengite kaalude vaatamiseks kirjuta:

```
> kokku$kaal
```

ja tudengite pikkuste histogrammi saamiseks kasuta:

```
> hist(kokku$pikkus)
```

Samuti saab andmeid välja nõuda kasutades indekseid. Proovi, mida teevad järgmised käsud:

```
kokku[1, ]
kokku[1, 2]
kokku[1:5, ]
kokku[, 2]
kokku[kokku$pikkus>195, ]
kokku[kokku$sugu==1 & kokku$pikkus>180, ]
```

Kui töötame pidevalt ühe ja sama andmestikuga, on iga tunnuse ette andmestiku nime kirjutamine tülikas. Selle vältimiseks saame R-le öelda, millise andmestikuga me parajasti töötame – seda saab teha *attach()* käsku kasutades:

```
> attach(kokku)
```

Peale *attach*-käsku võime tunnuste poole pöörduda ka ilma andmestiku nime täpsustamata:

```
> table(sugu)
```

```
sugu
  1   2
512 149
```

Kui oleme töö oma andmestikuga lõpetanud (näiteks soovime hakata töötama mõne teise andmestikuga) tuleks varemkasutatud andmestik „lahti ühendada“ käsu *detach()* abil. Peale *detach()*-käsu andmist saab andmestikus olevate tunnuste poole pöörduda vaid „täisnime“ – *andmestikunimi\$tunnusenimi* – kasutades.

```
> detach(kokku)
```

```
> table(sugu)
```

```
Error in table(sugu) : object "sugu" not found
```

```
> table(kokku$sugu)
```

```
  1   2
512 149
```

Antud materjalis edaspidi toodud näidete läbimiseks *attach*-ige andmestik *kokku* tagasi.

## Puuduvad väärtused

Puuduva väärtuse tähis R-is on NA (Not Available). R käsitleb kohati puuduvaid väärtuseid peaaegu segavalt korrektselt:

```
> mean(pikkus)
```

```
[1] NA
```

Sest osade tudengite pikkus pole teada, ja seega pole võimalik leida ka tudengite keskmist pikkust – keskmine pikkus on puuduv väärtus. Osad funktsioonid võimaldavad leida otsitava statistiku väärtuse kasutades vaid olemasolevaid väärtuseid, kui lisame lisaparametri *na.rm=T*:

```
> mean(pikkus, na.rm=T)
```

```
[1] 171.1167
```

Samuti võime luua uue andmestiku, kust on eemaldatud need tudengid, kelle pikkust või kaalu pole teada:

```
> tudengid=kokku[!is.na(pikkus)&!is.na(kaal),]
```

ning edasi võime analüüsida juba olemasolevaid vaatluseid sisaldavat andmestikku:

```
> detach(kokku); attach(tudengid)
> mean(pikkus)
[1] 171.1052
```

## Põhistatistikud

Põhistatistikute leidmine on imelihtne. Järgnevalt leiame kõigi tudengite keskmise pikkuse, 2001.a küsitletud tudengite keskmise pikkuse ja keskmise pikkuse küsitlusaasta järgi:

```
> mean(pikkus)
[1] 171.1052
> mean(pikkus[aasta==2001])
[1] 170.8861
> by(pikkus, aasta, mean)
INDICES: 2001
[1] 170.8861
```

```
-----
INDICES: 2002
[1] 170.981
-----
```

```
INDICES: 2003
[1] 171.4079
-----
```

```
INDICES: 2004
[1] 171.1837
```

Pideva või diskreetse tunnuse jaotust iseloomustavaid statistikuid saab leida näiteks järgmiste funktsioonide abil:

<code>length(pikkus)</code>	– vaatluste arv (vektori pikkus)
<code>min(pikkus)</code>	– miinimum
<code>max(pikkus)</code>	– maksimum
<code>range(pikkus)</code>	– miinimum ja maksimum
<code>mean(pikkus)</code>	– keskmine
<code>median(pikkus)</code>	– mediaan
<code>sd(pikkus)</code>	– standardhälve
<code>var(pikkus)</code>	– dispersioon (kasutatav ka kovariatsiooni leidmiseks)
<code>quantile(pikkus, 0.13)</code>	– 0.13-kvantiil
<code>summary(pikkus)</code>	– lühiiseloostus (miinimum ja maksimum, alumine ja ülemine kvantiil, mediaan)
<code>cor(pikkus, kaal)</code>	– pikkuse ja kaalu vaheline korrelatsioonikordaja



## Ülesanded

### Ülesanne 1

Tunnus *olu* näitab, mitu pudelit õlut tudeng nädala jooksul ära joob:

- 1 - mitte kunagi
- 2 - vähem kui pudel nädalas (1 pudel = 0,33 l)
- 3 - 1-4 pudelit nädalas
- 4 - 5-12 pudelit nädalas
- 5 - 13-või rohkem pudelit nädalas

1. Leia sagedustabel. Kas tudengid joovad palju?
2. Nuputa välja, mida arvutab järgmine käsk:  
`prop.table(table(olu)) * 100`
3. Leia õlut mittetarbivate tudengite keskmine pikkus; alla pudeli nädalas joovate tudengite keskmine pikkus jne. Mida märkad? Kas õlut joovad tudengid on pikemad või lühemad kui õlut mittejoovad tudengid? Kuidas seletad/põhjendad nähtud tulemust?

### Ülesanne 2

Leia tudengite keskmine kaal ja kaalude mediaan. Kumb tuleb suurem? Miks? Selgituse otsimisel võid vaadata ka tudengite kaalude histogrammi. Selle saad sisestades käsu: `hist(kaal)`

### Ülesanne 3

Tunnus *haiglaravi* näitab, kas tudeng on viimase kahe aasta jooksul vajanud haiglaravi. Nendel tudengitel, kes haiglasse pole sattunud, on *haiglaravi*=0. Kes aga on kahe aasta jooksul haiglas viibinud, neil on *haiglaravi*=1. Leia tunnuse *haiglaravi* keskmine. Mida tavakeeles öeldult leitud keskmine näitab?

### Ülesanne 4

Millise käsu abil saaks kätte kolme esimese tudengi andmed?

## Andmete sisestamine klaviatuurilt/programmist

Vahel soovime mõningaid väärtuseid ka käigupealt sisestada. Lihtsaim viis vaatluste (või arvujada) sisestamiseks on kasutada funktsiooni *c*. Loome vektori *h* mis sisaldab viie puu kõrguseid:

```
> h = c(20, 12, 14, 16, 33)
> h
[1] 20 12 14 16 33
```

Andmed, mida vektoris hoitakse, ei pea olema arvulised. Me võime tekitada ka vektori, mis sisaldab uuritud puude liigilist kuuluvust näitavaid andmeid:

```
> liik = c("Kuusk", "Kask", "Kask", "Kask", "Kuusk")
> liik
[1] "Kuusk" "Kask" "Kask" "Kask" "Kuusk"
```

Üks võimalus andmematriksit luua on teha seda kasutades olemasolevaid vektoreid ehk üksiktunnuste väärtuseid. Üksiktunnused saab andmestikuks kokku panna *data.frame* käsu abil:

```
> minuandmed=data.frame(liik, h)
> minuandmed
  liik h
1 Kuusk 20
2 Kask 12
3 Kask 14
4 Kask 16
5 Kuusk 33
```

## Väärtuste automaatne genereerimine

Vahel tekib vajadus kiiresti genereerida tunnuse väärtused mingi kindla skeemi kohaselt. Näiteks järjestikustest arvudest koosnevat vektorit saab tekitada järgmise käsu abil:

```
> 3:10
[1] 3 4 5 6 7 8 9 10
```

Näiteid kasutamisest:

```
h[1:3]
for(i in 1:10) {print(„R on kole keeruline“)}
```

Käsuga `seq` saab tekitada järjestikuseid arve mingi etteantud sammuga:

```
> seq(6,10,0.5)
[1] 6.0 6.5 7.0 7.5 8.0 8.5 9.0 9.5 10.0
```

või saab tema abil tekitada etteantud pikkusega vektori:

```
> seq(0,1,length=4)
[1] 0.0000000 0.3333333 0.6666667 1.0000000
```

Käsuga `rep` saab esimest argumenti korrata soovitud arv kordi:

```
> rep(3,10)
[1] 3 3 3 3 3 3 3 3 3 3
> rep(c(1,7,9),2)
[1] 1 7 9 1 7 9
> rep(c("Tallinn","Tartu"),c(3,2))
[1] "Tallinn" "Tallinn" "Tallinn" "Tartu" "Tartu"
```

## Ülesanded

1. Pika, 10 aastat kestnud uuringu käigus püüti lõksudega metsast kinni 123 halli karvaga hiirt, 156 täpilise karvaga hiirt ja 23 mummulise karvaga hiirt. Tekita vektor, mis sisaldaks kinnipüütud hiirte karvavärvi (iga hiire värv eraldi kirjas).

## Graafikast

Andmete visualiseerimiseks ja statistikas vajaminevate jooniste tegemiseks pole arvatavasti olemas R'ist paremat vahendit. Jooniste kvaliteet vastab ka rangemaile nõuetele ja sisu poolest olulist on äärmiselt lihtne esile tõsta. Lühülevaate R'i graafika võimalustest võib saada käsuga `demo(graphics)`, lisamaterialina võid piiluda ka järgmiseid weebilehti:

<http://www.ms.ut.ee/mart/R/Rgraafika.html>

[http://www.ms.ut.ee/mart/biomeetria2009/Jaotuse\\_kirjeldamine.pdf](http://www.ms.ut.ee/mart/biomeetria2009/Jaotuse_kirjeldamine.pdf)

[http://www.ms.ut.ee/mart/biomeetria2009/Seosed\\_tunnuste\\_vahel.pdf](http://www.ms.ut.ee/mart/biomeetria2009/Seosed_tunnuste_vahel.pdf)

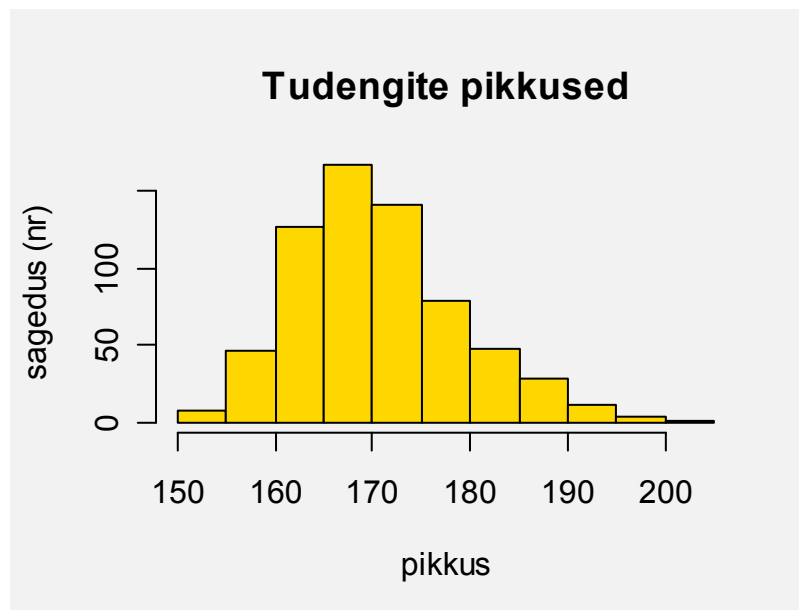
Esmalt mõned sagedamini kasutamist leidvad käsud.

`barplot` - tulpdiaagrammi tegemine;  
`plot` - käsk, mida võib proovida enam-vähem kõige visualiseerimiseks, olgu see siis andmevektor, regressioonanalüüsi mudel või klasteranalüüsi tulemused;  
`pie` - „kakuke“;  
`boxplot` - nn karp-vurrud diagramm;  
`hist` - histogramm.

Näiteid nende käskude kasutamisest:

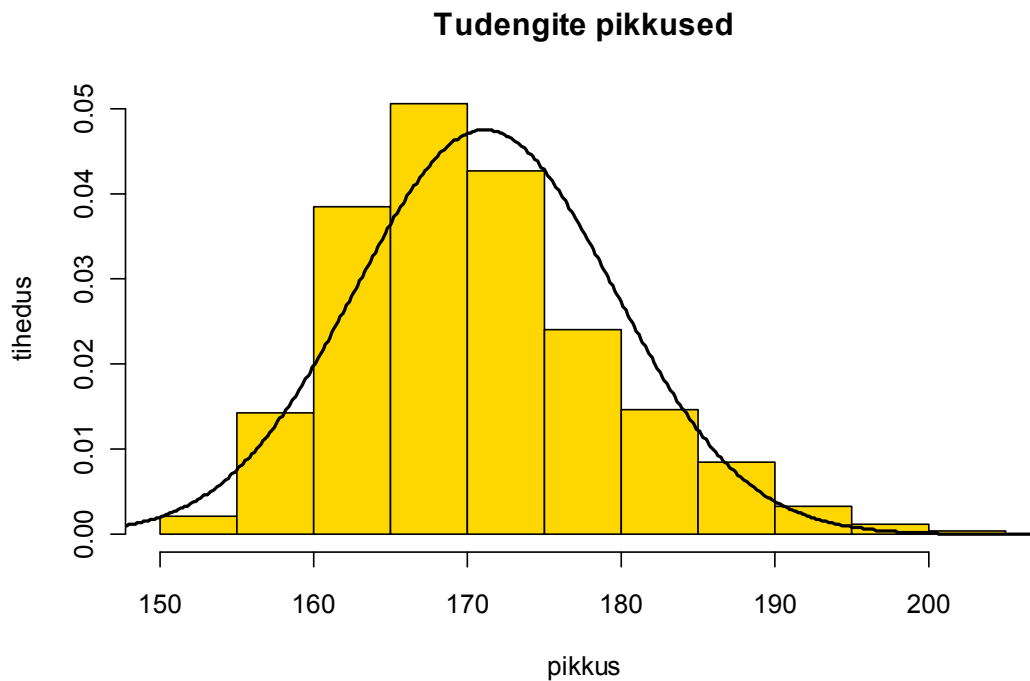
```
barplot(table(sugu))
pie(table(olu))
pie(c(7,5,10,10,20), labels=c(„Reform“, „Isamaa“, „Kesk“,
„Res Publica“, „üksik.“))
boxplot(pikkus~sugu, names=c("Naised", "Mehed"))
hist(pikkus)
plot(pikkus)
plot(pikkus, kaal)
```

```
hist(pikkus, main="Tudengite pikkused", ylab="sagedus", col="gold")
```



Tudengite pikkuste histogramm koos andmetega kõige paremini sobiva normaaljaotusega (normaaljaotuse tihedusfunktsiooni graafikuga):

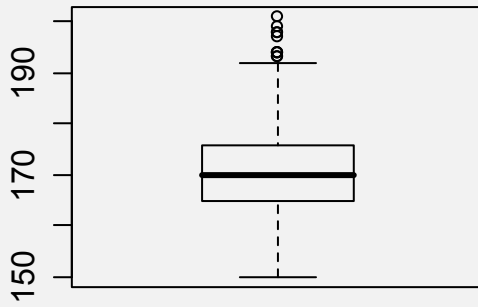
```
hist(pikkus, main="Tudengite pikkused", ylab="tihedus",  
      col="gold", freq=FALSE)  
x=seq(140, 210, length=500)  
lines(x, dnorm(x, mean=mean(pikkus,na.rm=T), sd=sd(pikkus,na.rm=T) ), lwd=2)
```



## Karpdiagramm pikkusele

### Lihtne variant

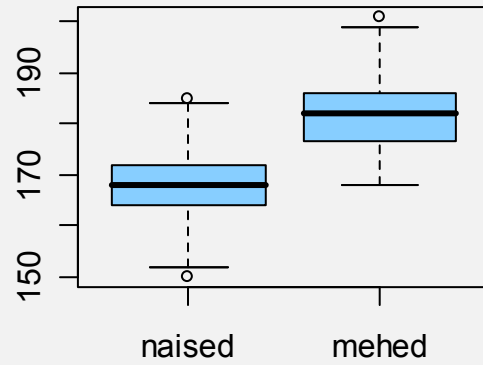
```
boxplot(pikkus)
```



### Ilustatud variant

```
boxplot(pikkus~sugu, col="skyblue1",  
names=c("naised","mehed"), main="Karpdiagramm  
pikkusel
```

## Karpdiagramm pikkusele



## Jooniste kohandamine

Kui arvuti joonistab teile esimesel katsel just sellise joonise, nagu te soovite, siis pole teil arvatavasti soovi lugejatele midagi sisulist ja uudset öelda. Kõigil ülejäänud juhtudel läheb tarvis graafikute sättimist sobivamaks, sellisteks kust sisuline informatsioon esile tõuseks. Selleks on kasulikud mõned käsud, mis (enam-vähem) töötavad kõigi ülalloetletud graafikafunktsioonidega.

Graafikul nähtavat osa saab piiritleda käskudega *xlim* ja *ylim* (mõne mitteloetletud funktsiooni korral saab kasutada ka käsku *zlim*). Näiteks proovi, mille poolest erinevad järgmise kahe käsu tuemused:

```
plot(pikkus, kaal)
plot(pikkus, kaal, ylim=c(0,130))
```

Telgede tähistusi saab muuta käskudega *xlab* ja *ylab*; pealkirja saab lisada käsuga *main*:

```
barplot(table(olu), names=c("ei joo", "alla 1",
    "1-4", "5-12", "13+"), xlab="pudelit nädalas",
    ylab="Sagedus", main="Tudengite õlletarbimine",
    col=rainbow(8))
```

Kasutatavaid värve saab sageli muuta käsuga *col*:

```
> barplot(table(sugu), col=c("orange", "skyblue"))
```

Andes käsu *colors()* näidatakse kõiki R'ile nimepidi tuttavaid värve, värve saab valida ka mõnest valmiskomplektist. Näiteks funktsioon *heat.colors(6)* tekitab 6 värvi soe-tulikuum skaalal:

```
> barplot(1:10, col=heat.colors(10))
```

Täpsemalt informatsiooni värvide kohta saab käsuga *?colors*.

Joonistades kauneid graafikuid soovitakse mõnikord suurendada telgedel kasutatavaid tekste, muuta tausta vms värve, suurendada või vähendada telgede jaoks jäetud ruumi jms. Selliste spetsiifiliste lisaparameetrite kohta, mida sageli võib lisada kõigile joonistamisega tegelevatele käskudele, saab lisainformatsiooni käsuga *?par*.

Teisi olulisi graafikaga seotud käske:

<i>lines</i>	- lisa olemasolevale pildile jooni
<i>points</i>	- lisa olemasolevale pildile punkte
<i>text</i>	- lisa pildile teksti
<i>legend</i>	- lisa pildile legend
<i>arrows</i>	- joonista kuhugi osutav nooleke
<i>axis</i>	- graafikule ebatüüpilise telje lisamiseks

## Ülesanded

1. Mida teeb järgmine käsk?

```
boxplot(pikkus~olu, col="tan")
```

Kuidas toodud käsku tuleks täiendada, et telgedele tekiks mõistlikud pealkirjad?

2. Tudengite süstoolne vererõhk on kirjas tunnuses SVR ja diastoolne vererõhk on kirjas tunnuses DVR. Joonista mõlema tunnuse jaoks histogrammid. Kas näed midagi üllatavat, ootamatut (seoses tunnusega *DVR*)?

2. Pikkuste ja kaalude seost iseloomustava joonise saame tellida käsuga

```
plot(pikkus, kaal)
```

Lisa sellele joonisele punane nooleke, mis viitaks ühele lühikesele ja paksule tudengile.

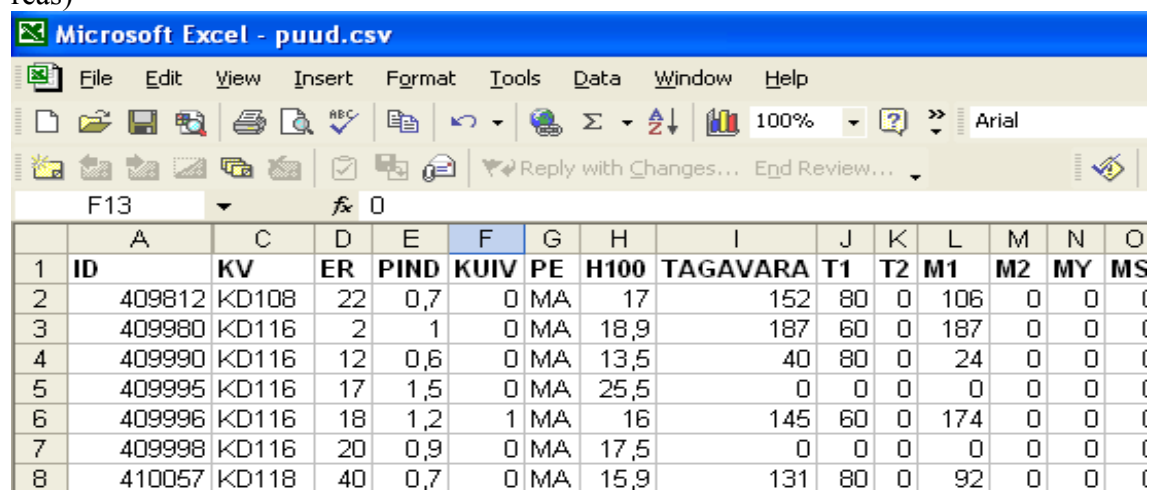
Vihje: noole lisamiseks saab kasutada `arrows`-käsku.



## Lisa 1. Andmete importimisest

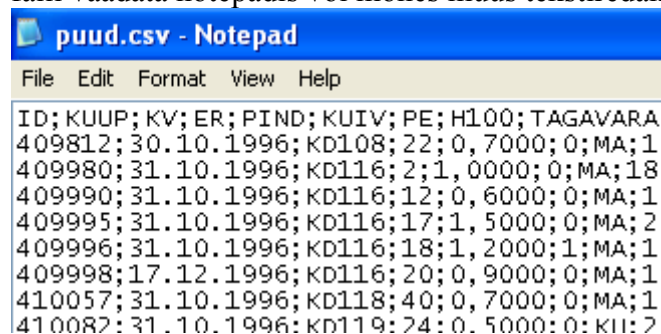
Suuremate andmestike sisestamiseks pole R just kõige sobivam vahend. Enamasti sisestatakse andmeid kas spetsiaalselt andmete sisestamiseks mõeldud tarkvara abil või kasutades mõnda tabelarvutusprogrammi (näiteks Excelit). Andmestiku importimisel tabelarvutusprogrammist (Excelist) on soovitatav andmefail esmalt salvestada CSV-formaadis (Comma Separated Values), näiteks faili "C:\puud.csv" (File -> Save As -> muuda Save As type aknas failitüüp „CSV (Comma Delimited) (\*.csv)“-ks).

Andmestik Excelis (tunnuste nimed – lühikesed, soovitatavalt ühesõnalised – esimeses reas)



	A	C	D	E	F	G	H	I	J	K	L	M	N	O
1	ID	KV	ER	PIND	KUIV	PE	H100	TAGAVARA	T1	T2	M1	M2	MY	MS
2	409812	KD108	22	0,7	0	MA	17	152	80	0	106	0	0	
3	409980	KD116	2	1	0	MA	18,9	187	60	0	187	0	0	
4	409990	KD116	12	0,6	0	MA	13,5	40	80	0	24	0	0	
5	409995	KD116	17	1,5	0	MA	25,5	0	0	0	0	0	0	
6	409996	KD116	18	1,2	1	MA	16	145	60	0	174	0	0	
7	409998	KD116	20	0,9	0	MA	17,5	0	0	0	0	0	0	
8	410057	KD118	40	0,7	0	MA	15,9	131	80	0	92	0	0	

Andmestik peale csv-faili salvestamist näeb välja järgmine (soovi korral saame csv-faili vaadata notepadis või mõnes muus tekstiredaktoris):



```
puud.csv - Notepad
File Edit Format View Help
ID;KUUP;KV;ER;PIND;KUIV;PE;H100;TAGAVARA
409812;30.10.1996;KD108;22;0,7000;0;MA;1
409980;31.10.1996;KD116;2;1,0000;0;MA;18
409990;31.10.1996;KD116;12;0,6000;0;MA;1
409995;31.10.1996;KD116;17;1,5000;0;MA;2
409996;31.10.1996;KD116;18;1,2000;1;MA;1
409998;17.12.1996;KD116;20;0,9000;0;MA;1
410057;31.10.1996;KD118;40;0,7000;0;MA;1
410087;31.10.1996;KD119;24;0,5000;0;MA;7
```

Salvestatud andmestiku lugemiseks R-i tuleb anda näiteks järgmine käsk:

```
puud=read.csv2("C:/puud.csv", header=T)
```

ja vaatamaks, kas andmete sisselugemine läks valutult:

```
head(puud)
names(puud)
```

Vahel tekivad probleemid – andmeid ei loeta R-i õigel kujul, saame veateateid vm. Üks põhjus – andmefail sisaldas tekstikirjeid, mis sisaldasid keelatud märke (Näiteks sümbolit „;“). Vahel aga on põhjuseks see, et erinevad Excelid võivad (erinevates masinates) teha erinevaid csv-faile. Vahel pannakse andmeväljade vahele semikooloni (;) asemel näiteks koma (,) ja kümnendkohtade eraldaja arvus kasutatakse hoopis

punkti (.) – seda näeme avades salvestatud csv-faili notepadis/tekstiredaktoris. Sellisel juhul tuleb andmete sisselugemiseks R-i anda hoopis käsk

```
puud=read.csv2("C:/puud.csv", header=T, dec=".", sep=",")
```

Pange tähele:

- sisseloetud andmestik tuleb kuhugi salvestada, kui soovime teda hiljem ka kasutada! Siin salvestasime ta andmestikuks nimega “puud”.
- Failinimes on kasutatud tagurpidi kaldkriipse („“)!
- Käsuga *sep=* määratakse sümbol, mis eristab eri tunnuste väärtuseid (Excel võib salvestamisel kasutada eraldajana nii sümbolit “;” kui ka “;”)
- *dec=* parameetri abil saab määratleda sümboli, mis tähistab arvus koma (Näiteks Excel kasutab kümnendkohtade tähistamiseks vahel sümbolit “.” ja vahel sümbolit “;”).
- Parameeter *header=T* ütleb arvutile, et tunnuste nimed on kirjas tekstifaili esimeses reas.

R-i andmestikku saab samuti salvestada tekstifaili kasutades käsku

```
write.table(andmestik, "C:/andmed/uustabel.csv",  
            sep="," , dec=".", header=T, row.names=F)
```

### Teine võimalus

Alternatiivne võimalus oleks installeerida lisamoodul *xlsReadWrite*, milles sisalduv funktsioon *read.xls* võimaldab ka otse Exceli faile lugeda. Lisamooduli lisamiseks (Vanemuise tänava arvutiklassis saab seda teha ainult administraator) valige R-is menüüst Packages->Install Packages -> <vali server> -> *xlsReadWrite*

Seejärel saate Exceli andmestiku sisse lugeda näiteks järgmise käsu abil (ärge käivitage, teie arvutis nimetatud Exceli-faili ju pole!):

```
andmed2=read.xls("C:/too/MinuAndmed.xls")
```

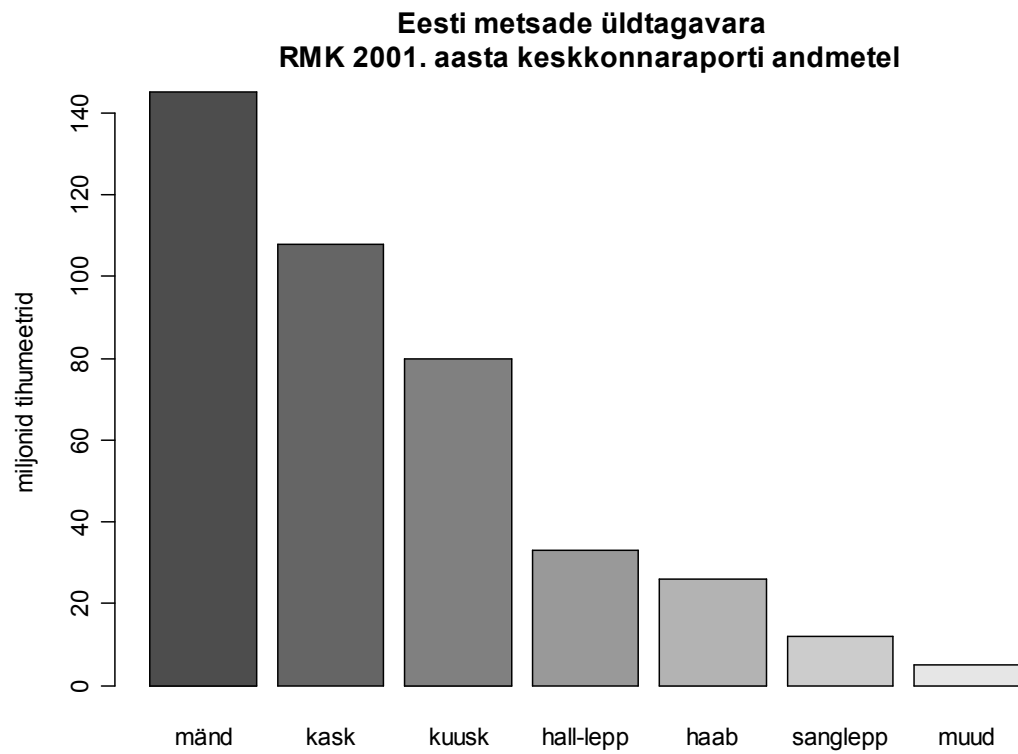
Märkused:

- Kasutades lisamoodulit *foreign* saab R-i (otse) lugeda ka statistikapakettide S-Plus, Stata, Minitab ja SPSS andmefailide.

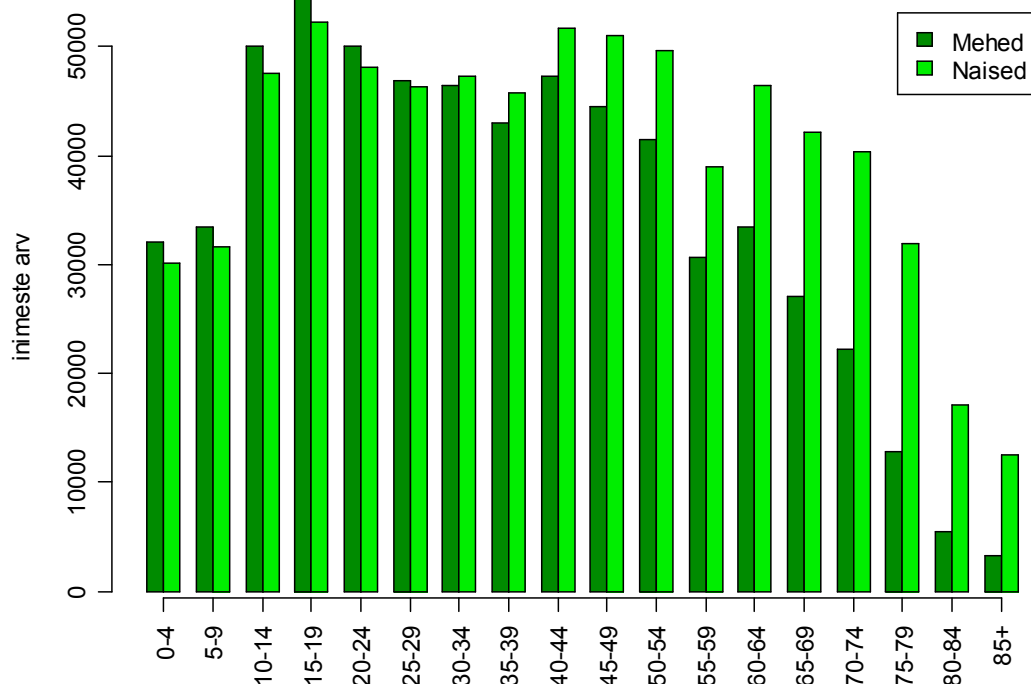
Täiendavat infot vaata R-i koduleheküljelt (*Manuals*→*R Data Import/Export*)

Juhuslikke näiteid R'i graafikutest:

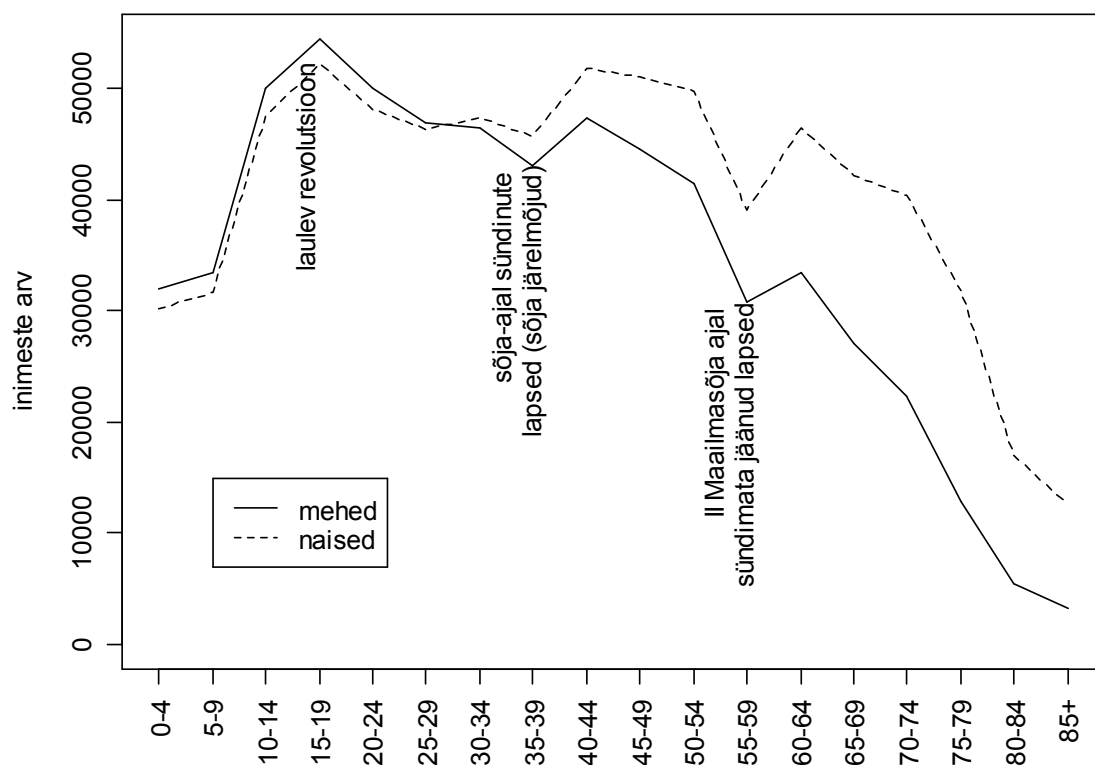
```
> barplot(c(145,108,80,33,26,12,5),
names.arg=c("mänd","kask","kuusk","hall-lepp", "haab", "sanglepp",
"muud"), main="Eesti metsade üldtagavara \n RMK 2001. aasta
keskkonnaraporti andmetel", ylab="miljonid tihumeetrid",
col=gray(3:10/10))
```



Eesti rahvastiku jaotus vanuse ja soo järgi  
1. jaanuaril 2003



Eesti rahvastiku jaotus vanuse ja soo järgi  
1. jaanuaril 2003



Expected DS cases with 95%-tolerance interval and DS live birth and PD cases

