

Biomeetria 2. praktikum

Harjumine R'iga jätkub. Uuritava tunnuse jaotuse kirjeldamine.

Loeme sisse andmestiku kalad:

```
load(url("http://www.ms.ut.ee/mart/biomeetria2012/kalamees.RData"))
```

Andmestiku lühikirjeldus:

Soomes Tampere lähedal asuvast Laenelmavesi järvest püüti 159 kala. Püütud kalad on pärit 7 liigist. Mõõdetud tunnuste kirjeldused:

Species on kodeeritud tunnus kalaliikidest:

- 1 latikas
- 2 siig
- 3 särg
- 4 linask
- 5 tint
- 6 haug
- 7 ahven

Weight on kala kaal grammides

Length1 on kala pikkus ninast saba alguseni sentimeetrites.

Length2 on kala pikkus ninast saba keskosani sentimeetrites

Length3 on kala pikkus ninast saba tipuni sentimeetrites.

Height on maksimaalne kõrgus, mis antud protsendina *length3*-st.

Width on maksimaalne paksus, mis on samuti

Sex on kala sugu: 0-emane; 1-isane

Esmalt paar ülesannet eelmises praktikumis õpitu meenutamiseks.

Ülesanne 1.

Vaata esimese 10 kala andmeid. Seda saab teha käsuga:

Ülesanne 2.

Andmestikust saab ühte tunnust välja noppida käsuga kujul *andmestik\$tunnus*, näiteks kalade andmestikust kala kaalu (*Weight*) saame kätte käsuga *kalad\$Weight*. Ühe ja sama andmestikuga töötades muutub andmestiku nime igale poole ettekirjutamine tülilikaks. Milline käsk tuleb anda, et saaksime andmestiku *kalad* tunnuste poole pöörduda ilma, et peaksime igale poole andmestiku nime kirjutama? Anna see käsk arvutile!

Kirjuta kasutatud käsk ka siia:

Nominaalse tunnuse jaotuse iseloomustamine

Kalade andmestikus on üks nominaalne tunnus – kala liik (*Species*). Leiame esmalt sagedustabeli:

```
table(Species)
```

Millist liiki kala esines kõige rohkem meie valimis? Sellele küsimusele vastamine osutub mõnevõrra ebamugavaks, sest tunnus *Species* on kodeeritud. Edaspidi soovime, et järgnevatel analüüsidel ja graafikutel näeksime uuritava tunnuse sõnalisi väärtuseid („ahven“, „lest“ jne), mitte tunnuse kodeeritud väärtuseid. Selleks loome uue faktortunnuse ja ütleme, milline väärtus mida tähendab.

```
liik=factor(Species, labels=c("latikas", "siig", "särg",  
                             "linask", "tint", "haug", "ahven"))
```

```
table(liik)
```

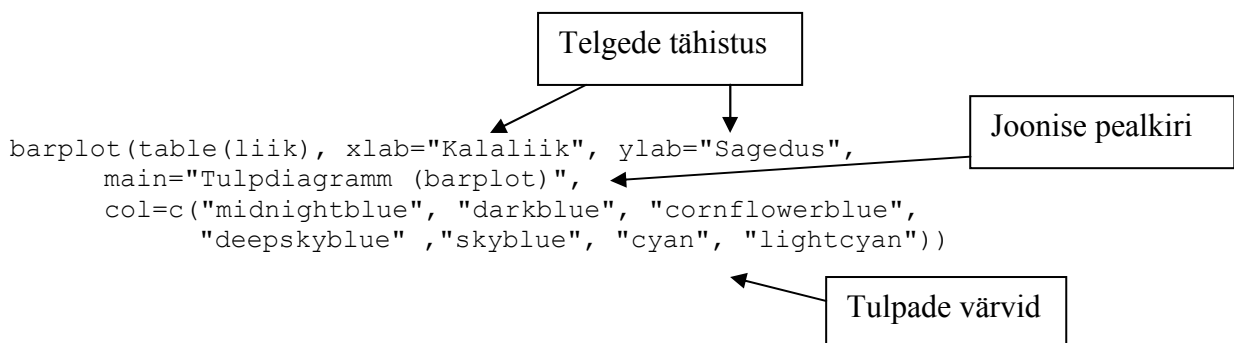
Märkus: antud käsk ei lisa andmestikku kalad uut tunnust. Loodud objekt liik on täiesti eraldiseisev, omaette objekt. Kui sooviksime loodud tunnust liik lisada andmestikku kalad, siis saame seda teha näiteks käsuga:

```
kalad=data.frame(kalad, liik)
```

Sagedustabelit kasutades võime joonistada ka mõne graafiku – näiteks tulpdiagrammi või kakukese:

```
pie(table(liik))  
barplot(table(liik))
```

Muidugi tuleb vahel graafik veidi esinduslikumale kujule viia:



Kõrvalepõige: värvide etteandmiseks on R'is paar erinevat võimalust:

1. Värvide nime kasutades. Võimalikke värvinimesi vaata `colors()` –käsuga või vaata näiteks <http://www.ms.ut.ee/mart/R/varvid2.html>
2. Võid kasutada värvipaletti:

```
barplot(table(liik), col=rainbow(7))  
barplot(table(liik), col=terrain.colors(7))  
barplot(table(liik), col=topo.colors(7))  
barplot(table(liik), col=heat.colors(7))
```
3. Saab kasutada ka RGB-„numbreid“ (red, green, blue + läbipaistvus):

```
barplot(table(liik), col="#FF0000FF")
```

Ülesanne 3.

Enamasti on tulpdiaagramm parem vahend nominaalse tunnuse jaotuse kirjeldamiseks (tulpdiagrammilt on kergem algset informatsiooni välja lugeda). Joonista tulpdiaagramm tunnusele liik, kusjuures tee nii, et y-teljel poleks mitte sagedused vaid iga liigi protsentuaalne osakaal. Vajadusel vaata „Jaotuse kirjeldamine“-nimelist abimaterjali. Ilmtingimata lase R-l ka graafiku y-teljele kirjutada, et tegemist on protsendiga. Soovitud graafiku saad järgmise käsu abil:

.....

Ülesanne 4.

Tee tunnuse *Sex* – kalade sugu – jaotust iseloomustav graafik või tabel. Kas mõlemast soost kalu on ligikaudu samapalju? Kui ei, siis miks (paku välja paar erinevat võimalikku põhjendust).

Vaikimisi kipub R sagedustabelis (ja tema põhjal joonistatud graafikutes) tunnuse puuduvaid väärtuseid „maha vaikima“. Soovi korral on aga puuduvad väärtused võimalik ka nähtavaks teha:

```
table(Sex, exclude=NULL)
sugu=factor(Sex, exclude=NULL, labels=c("emane", "isane", "PUUDU"))
table(sugu)
barplot(table(sugu), col=c("pink", "blue", "gray"))
```

Pideva tunnuse jaotus.

Heidame esmalt pilgu peale kalade pikkuste (tunnus *Length3*) jaotusele – kasutame selleks histogrammi:

```
hist(Length3)
```

või, korrektsemalt vormistatud tulemuse jaoks:

```
hist(Length3, ylab="Sagedus", xlab="Pikkus (cm)", main="Kalade pikkused",
     col="gold")
```

Saadud graafikul peaks olema näha mitu „tippu“ (kalade pikkuse jaotus on multimodaalne). Miks?

Vaata ka ahvenate pikkuste jaotust. Kas ahvenate pikkuste jaotus on unimodaalne (ühe tipuga)? Proovi põhjendada nähtud tulemust.

Vahel soovime võrrelda uuritava tunnuse väärtuseid erinevates gruppides. Näiteks iseloomustada, kuidas erineb ahvenate ja haugide pikkus. Proovime seda teha graafiku abil.

Katse 1 (EBASOOVITAV!!!)

```
par(mfrow=c(2,1))  
hist(Length3[liik=="ahven"], main="Ahven")  
hist(Length3[liik=="haug"], main="Haug")
```

Jagab graafikaakna nii, et tekiks 2 rida graafikuid, igas reas 1 graafik

Miks on tekkinud joonis eksitav?

Katse 2 (PAREM!!)

```
par(mfrow=c(2,1))  
hist(Length3[liik=="ahven"], main="Ahven", xlim=c(0,70))  
hist(Length3[liik=="haug"], main="Haug", xlim=c(0,70))
```

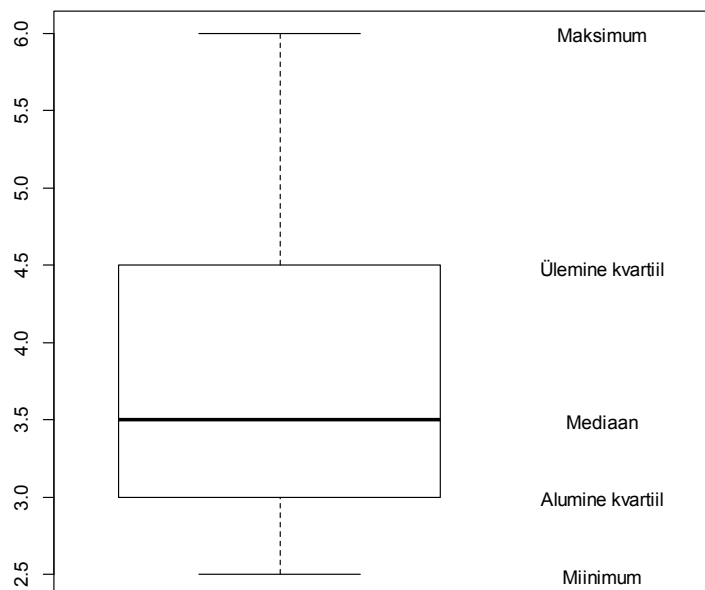
Määrab x-telje alguse ja lõpu

NB! Graafikaakna kaheks-jaotatusest lahtisaamiseks pane kas graafikaaken lihtsalt kinni (lihtsam võimalus) või anna käsk

```
par(mfrow=c(1,1))
```

Enam kui paari grupi puhul muutub uuritava tunnuse jaotuse võrdlemine sellisel viisil raskeks. Parem võimalus on kasutada nn karp-vurrud diagrammi.

```
boxplot(Length3~liik)
```



Histogramm on ilus küll, aga vahel soovime ka pideva tunnuse puhul esitada andmeid sagedustabeli abil. Selleks peame uuritava tunnuse enne „lõikama“ paraja pikkusega juppideks (kasutades cut-käsku):

```
pikkusklass=cut(Length3, breaks=10)
table(pikkusklass)
```

Tulemus on üsna kole, kas pole? Lahenduseks oleks, kui annaksime ise ette piirid, mille järgi pidev tunnus klassideks jaotatakse. R võimaldab klassipiirideks ette anda (suvalist) vektorit, näiteks võime kasutada breaks-käsu taga sellist vektorit:

```
seq(0, 70, 10)
```

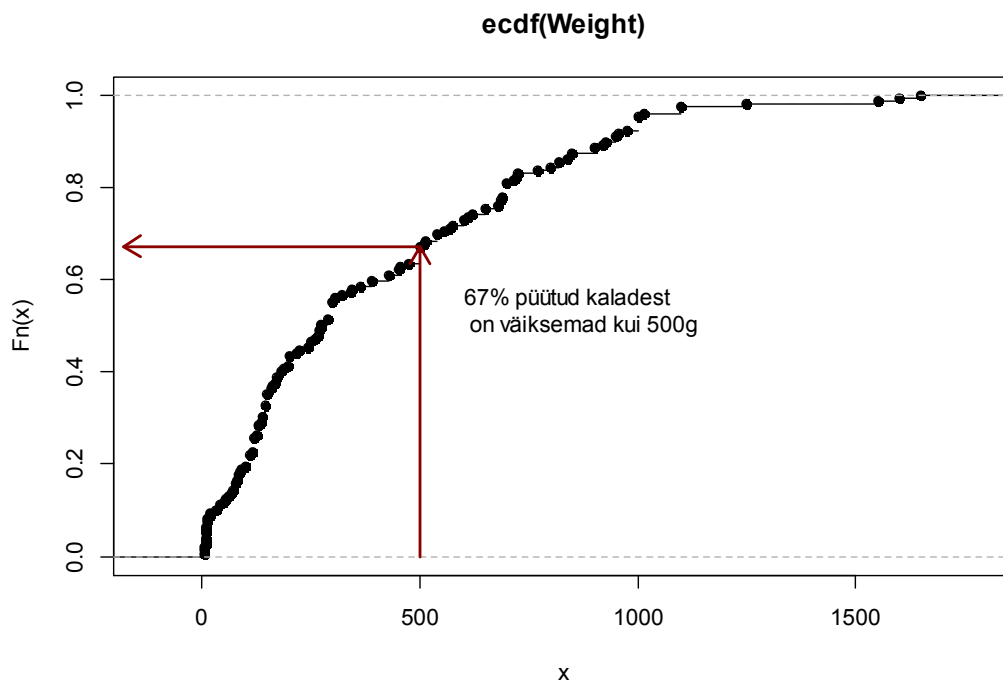
Tulemus on palju parem, proovi ise järgi!

```
pikkuskl=cut(Length3, breaks=seq(0, 70, 10))
table(pikkuskl)
```

Vahel võime eelistada uuritava tunnuse jaotuse kirjeldamiseks kasutada (kumulatiivse) jaotusfunktsiooni graafikut. Joonistamegi kalade kaalude empiirilise (kumulatiivse) jaotusfunktsiooni graafiku (*ecdf* - *empirical cumulative distribution function*):

```
plot(ecdf(Weight))
```

Kuidas lugeda jaotusfunktsiooni graafikut?



Teistmoodi vormistatud pildi saamiseks:

```
plot(ecdf(Weight), do.points=F, verticals=T,
      xlab="Kaal (g)", main="Kala kaalu jaotusfunktsioon")
```

Märkus: Kui soovime kaalule leitud jaotusfunktsiooni täpset väärtust kohas 500 (kui suur osa püütavatest kaladest ei kaalu rohkem kui 500grammi), siis kasuta käsku: `ecdf(Weight)(500)`.

Ülesanne: vaata leitud graafikut ja ütle, millise tõenäosusega kaalub kala, mille me Laenelmavesi järvest välja püüame, üle 1kg?

Vahel soovime võrrelda uuritava tunnuse histogrammi mõne tuntud jaotusega (näiteks normaaljaotusega). Enne selle ülesande juurde asumist proovime aru saada, mille poolest erinevad järgmised kaks graafikut:

Graafik 1:

```
hist(Length3)
```

Graafik 2:

```
hist(Length3, freq=FALSE)
```

Kas leiad erinevuse üles?

Jah, muutunud on graafiku y-telg. Kui esimesel joonisel oli y-teljel sagedus (loeti kokku, mitu vaatlust sattus ühte või teise väärtuste vahemikku), siis teise graafiku interpretatsioon on veidi teistsugune. Teisel graafikul on y-telje ühikud valitud selliselt, et histogrammi tulba pindala näitaks vastavasse vahemikku sattuvate väärtuste osakaalu. Vaatame näiteks tulpa vahemikus 25..30. Selle tulba kõrgus on 0,03, tulba pindala on $5 \cdot 0,03 = 0,15$. Seega paikneb $0,15 \cdot 100\% = 15\%$ kõigist vaatlustest vahemikus 25..30.

Miks on tarvilik selline y-telje väärtuste muutmine? Sagedused sõltuvad valimi suurusest. Võtame suurema valimi ja tulbad muutuvad 1. tüüpi joonisel ka kõrgemaks. Seega muutub raskeks erineva suurustega valimite pealt leitud graafikute võrdlemine (vähemalt kantuna samale joonisele). Eriti raskeks muutuks aga histogrammi võrdlemine teoreetilise jaotusega (mis peaaegu alati iseloomustab lõpmatult suurt valimit) – kaks erinevat joonist oleksid lihtsalt kordades erineva skaalaga.

Olles saanud histogrammi teoreetilise tihedusfunktsiooniga sobivasse skaalasse, peame leidma sobiva tihedusfunktsiooni väärtused grafikule lisamiseks. Normaaljaotuse tihedusfunktsiooni väärtuseid saab R'is leida käsuga `dnorm`.

Otsustame esmalt, kus soovime normaaljaotuse tihedusfunktsiooni väärtuseid arvutada. Arvutame näiteks normaaljaotuse tihedusfunktsiooni väärtuseid vahemikus 17,5..80 (500-s erinevas kohas antud vahemikus):

```
x=seq(17.5,80, length=500)
y=dnorm(x, mean=mean(Length3), sd=sd(Length3))
```

Saame punktipaarid – koht x , kus tihedusfunktsiooni väärtus on leitud ja y-normaaljaotuse tihedusfunktsiooni väärtus. Ühendame need punktid joonekestega ja kanname varem joonistatud histogrammile (loodame, et sa pole histogrammiga-graafikuakent varem sulgenud?):

```
lines(x,y, lwd=3)
```

Tulemus ei tule kuigi ilus. Normaaljaotuse kellukesekujuline kõverik ei kata kogu joonist. Miks? Sest 17,5 väiksemate väärtuste jaoks me ju ei arvutanud normaaljaotuse tihedusfunktsiooni väärtuseid! Paranda oma programmi selliselt, et normaaljaotuse tihedusfunktsiooni iseloomustav kõverik ulatuks ikka ilusti üle kogu graafiku!

Lisaks on joonistatud graafik lihtsalt koledavõitu. Proovi näiteks järgmisi käske eeskujuks võttes leida enda maitsele sobiv graafik!

Variant 1:

```
hist(Length3, freq=FALSE, col="skyblue", border="darkblue")
x=seq(0,80, length=1000)
y=dnorm(x, mean=mean(Length3), sd=sd(Length3))
lines(x,y, lwd=2, col="dodgerblue4")
```

Variant 2 (must-valge trüki jaoks):

```
hist(Length3, freq=FALSE, angle=45, density=20, main="Histogramm",
     ylab="tihedus", xlab="Pikkus (ninast sabaotsani, cm)")
lines(x,y, lwd=2)
```

Normaaljaotus ei sobi kaugeltki mitte kõikjale. Eks histogramm ise kirjeldab ka, milline ikkagi võiks olla uuritava tunnuse tegelik jaotus.

Järjestustunnus

Antud andmestikus pole ühtegi järjestustunnust. Kuid alati on võimalik muuta mõni arvuline tunnus järjestustunnuseks (**NB! Selline teisendus kaotab osa olemasolevast informatsioonist ja pole pärisandmete analüüsimisel soovitatav! Kasutame sellist teisendust siin vaid selleks, et oleks võimalik õppida ka järjestustunnusega ümberkäimist!!!**)

Tekitame järjestustunnuse, mis kirjeldab kalamehe hinnangut püütud kala suurusele:

```
kalasuurus=rep(NA, length(Weight))
kalasuurus[Weight<=250]="kassikala"
kalasuurus[Weight>250 & Weight<=500]="paras purakas"
kalasuurus[Weight>500 & Weight<=1000]="hiigelkala"
kalasuurus[Weight>1000]="maailmarekord!"
```

Teeme esimese katse saadud tunnuse jaotust kirjeldada.

```
table(kalasuurus)
barplot(table(kalasuurus))
```

Mis on halvasti?

Proovi kavalamalt (**ettevaatust**: vaata, et kasutatavad hinnangud oleksid tähepealt samad mis kasutusel andmetes):

```
kalasuurus2=factor(kalasuurus, levels=c("kassikala","paras purakas",
    "hiigelkala","maailmarekord!"))
table(kalasuurus2)
barplot(table(kalasuurus2), density=seq(10, 40, 10))
```

Diskreetne tunnus

Üks näide ka diskreetse tunnuse kohta.

```
poegi=c(0,0,0,0,1,1,1,1,1,3,3,3,3,4,4,5)
barplot(table(poegi), col=terrain.colors(6))
```

Mis on tekkinud graafikul viga (sarnane probleem võib esineda ka järjestustunnuse või nominaalse tunnuse korral)?

Proovime tekkinud viga parandada.

```
poegiF=factor(poegi, levels=0:5)
table(poegiF)
barplot(table(poegiF), col=terrain.colors(6))
```

Hoiatus: kasutaja peab ise vastutama, et kõik võimalikud tunnuse väärtused on levels-parameetri taga üles loetletud!