

Sissejuhatus andmete visualiseerimisse R keskkonnas

Krista Fischer

26.09.2018

R Markdown

Selle loengu slaidid on koostatud R Markdown'i abil. VT.
<http://rmarkdown.rstudio.com>.

RMarkdown on automaatselt kaasatud RStudio installatsiooni.
Matemaatilise teksti jaoks on soovitatav seda kasutada kombinatsioonis LaTeX-iga, mis peaks olema siis arvutisse ka eraldi installeeritud (nt MikTex süsteem).

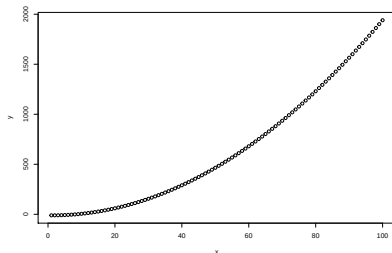
Teemad

- ▶ R graafika ilma andmeteta – matemaatiliste funktsioonide graafikud, vektorgraafika
- ▶ Sagedustabel ja selle kirjeldamine
- ▶ Sagedustabeli kirjeldamine tarkvara R abil
- ▶ Pideva tunnuse kirjeldamine: – Kirjeldavad statistikud – Graafiline kujutamine
- ▶ Mitme tunnuse (seose) graafiline kujutamine

Matemaatilise funktsiooni graafika

Joonistame funktsiooni $y = 0.2x^2 - 0.5x - 10$ graafiku

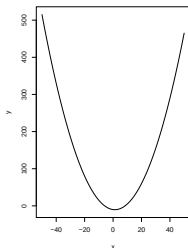
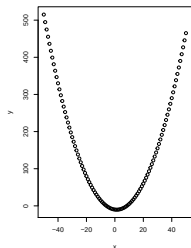
```
x = 1:100  
x = seq(1,100,1) # samaväärne eelmisega  
y = 0.2*x^2 - 0.5*x - 10  
plot(x,y)
```



Kuidas näha parabooli?

Vahetame x skaalat ja lisaks teeme pildi ilusamaks

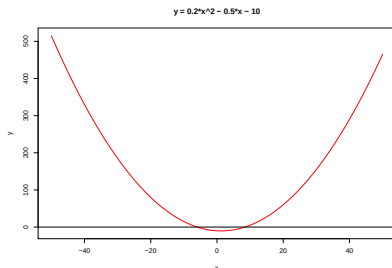
```
x = seq(-50,50)
y = 0.2*x^2 - 0.5*x - 10
par(mfrow=c(1,2)) # 2 pilti kõrvuti
plot(x,y)
plot(x,y,type="l")
```



Veel lisandusi

- ▶ teeme parabooli punaseks
- ▶ lisame graafikule $y=0$ joone - NB! R-s on palju funktsioone, mille abil saab olemasolevat graafikut täiendada
- ▶ kirjutame funktsiooni joonise pealkirjaks

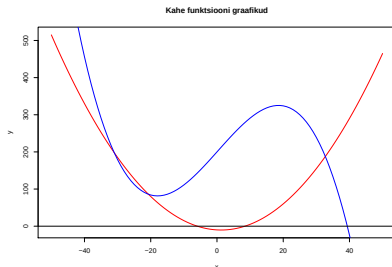
```
plot(x,y,type="l",col="red") # vt colors() - kõik värvinid  
abline(h=0) # abline(v=0) lisaks vertikaalse joone  
title("y = 0.2*x^2 - 0.5*x - 10")
```



Edasijõudnutele

- ▶ joonistame veel teise funktsiooni graafiku, aga ei tee uut joonist, vaid lisame eelmisele
- ▶ kasutades `plotmath()` võimalusi, kirjutame funktsioonid graafikule

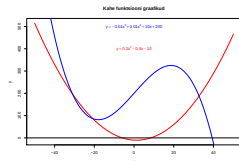
```
plot(x,y,type="l",col="red")  
y1 = -0.01*x^3+0.01*x^2 + 10*x +200  
lines(x,y1,col="blue")  
title("Kahe funktsiooni graafikud")  
abline(h=0)
```



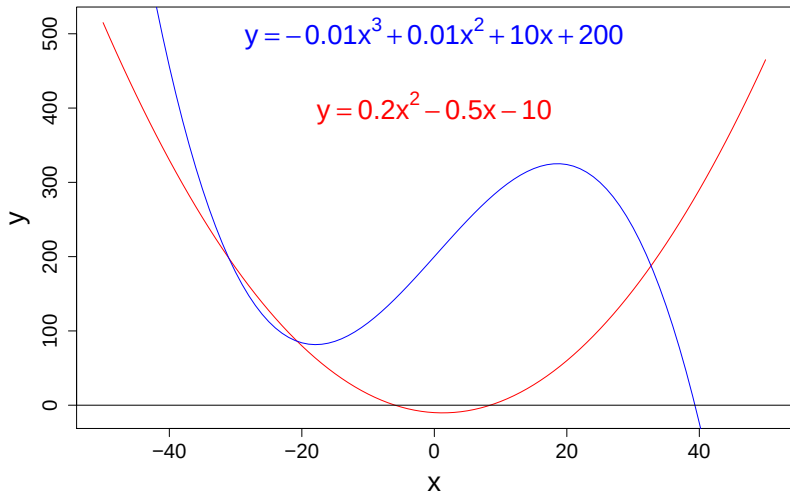
Veel edasijõudnumatele

Kasutades `plotmath()` võimalusi, kirjutame funktsioonid graafikule

```
plot(x,y,type="l",col="red")  
y1 = -0.01*x^3+0.01*x^2 + 10*x +200  
lines(x,y1,col="blue")  
title("Kahe funktsiooni graafikud")  
abline(h=0)  
text(0,400,expression(y == 0.2*x^2 - 0.5*x - 10),  
      col="red")  
text(0,500,expression(y == -0.01*x^3+0.01*x^2 + 10*x +200),  
      col="blue")
```



Kahe funktsiooni graafikud

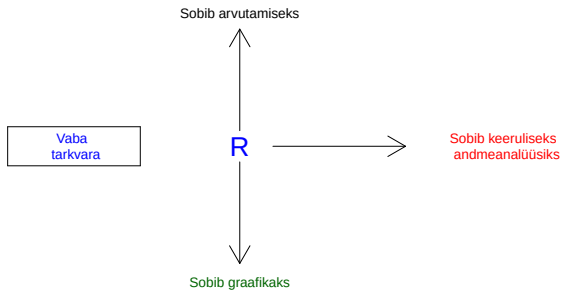


Koordinaatsüsteemi joonistamine

Tekitame tühja pildi ja lisame sinna elemente

```
plot(NULL,xlim=c(0,100),ylim=c(0,100),axes=F,  
      xlab="",ylab="")  
text(50,50,"R",cex=2,col="blue")  
arrows(50,54,50,80)  
text(50,84,"Sobib arvutamiseks")  
arrows(50,46,50,20)  
text(50,15,"Sobib graafikaks",col="darkgreen")  
arrows(55,50,75,50)  
text(90,50,"Sobib keeruliseks \n andmeanalüüsiks",  
      col="red")  
text(25,50,"Vaba \n tarkvara",col="blue")  
segments(15,55,35,55)  
segments(35,55,35,45)  
segments(35,45,15,45)  
segments(15,45,15,55)
```

Tulemus



Lihtne sagedustabel R-s

Loeme sisse andmestiku (saab ka URL-ist!) ja teeme tabeli maiustuste tarbimise kohta

“<https://courses.ms.ut.ee/2018/sissejuhmat/fall/uploads/Main/valim.csv>”

```
dat = read.csv(file=
"https://courses.ms.ut.ee/2018/sissejuhmat/fall/uploads/Main/valim.csv"
table(dat$maiust)
```

```
##
```

```
##  1  2  3  4
```

```
##  4 43 39 14
```

```
mtabel = table(dat$maiust)
```

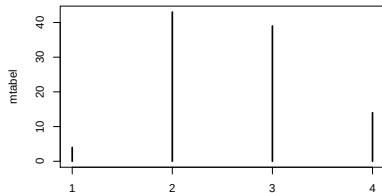
```
mtabel
```

```
##
```

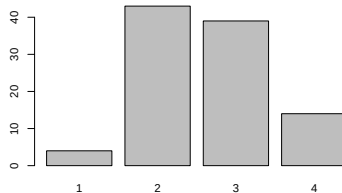
```
##  1  2  3  4
```

Sagedustabeli visualiseerimine

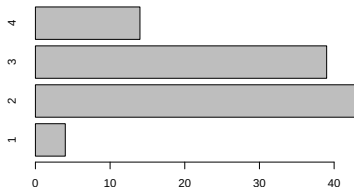
plot(mtabel)



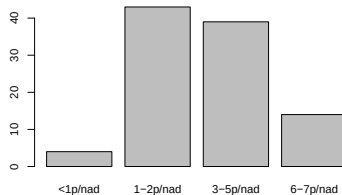
barplot(mtabel)



barplot(mtabel,horiz=T)



**barplot(mtabel,
names=c("<1p/nad","1-2p/nad","3-5p/nad","6-7p/nad"))**

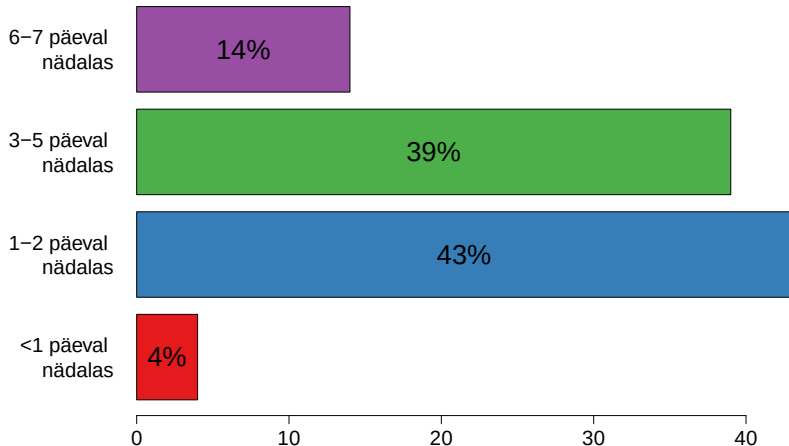


Sagedustabeli visualiseerimine – edasijõudnutele

```
par(mai=c(1,1.5,0.8,0.4))  
# jätame vasakule rohkem ruumi  
library("RColorBrewer")  
  #kui see on installeeritud  
display.brewer.all() #paletid  
cl = brewer.pal(4,"Set1")  
  # valime paleti ja 4 värvi sealt  
nimed = c("<1 päeval \n nädalas","1-2 päeval \n nädalas",  
          "3-5 päeval \n nädalas","6-7 päeval \n nädalas")  
b = barplot(mtabel,names=nimed,hORIZ=T,lAS=1,  
            col=cl, xlim=c(0,50))  
# las=1 pöörab y-telje annotatsiooni teistpidi  
text(mtabel/2,b,paste(round(100*prop.table(mtabel)), "%", sep=""),  
title("Maiustuste tarbimine geenivaramu kohordis"))
```

Sagedustabeli visualiseerimine – edasijõudnutele

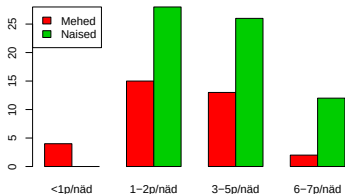
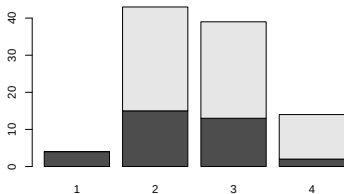
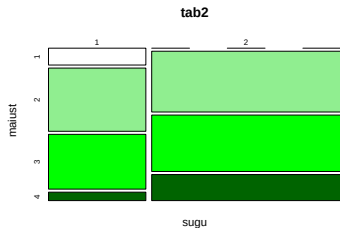
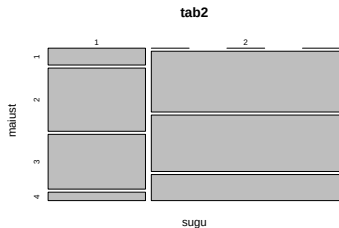
Maiustuste tarbimine geenivaramu kohordis



Kahemõõtmelise sagedustabeli visualiseerimine

```
par(mfrow=c(2,2))
tab2 = with(dat, table(sugu,maiust))
plot(tab2)
plot(tab2,col=c("white","lightgreen","green","darkgreen"))
barplot(tab2)
nim=c("<1p/näd","1-2p/näd","3-5p/näd","6-7p/näd")
barplot(tab2,names=nim,beside=T,col=c(2,3))
# beside=T paneb 2 grupi tulbad kõrvuti
legend("topleft",c("Mehed","Naised"),fill=c(2,3))
```


Kahemõõtmelise sagedustabeli visualiseerimine



Pideva tunnuse kirjeldav statistika – näide

Vt ka eraldi slide pideva tunnuse kirjeldava statistika kohta

```
mean(dat$kaal)
```

```
## [1] 72.22
```

```
# keskmised kaalud sugude kaupa (1-mees, 2-naine)  
with(dat, mean(kaal[sugu==1])) #"with" kasutata $ asemel
```

```
## [1] 81.47059
```

```
with(dat, mean(kaal[sugu==2]))
```

```
## [1] 67.45455
```

Pideva tunnuse kirjeldav statistika – näide

```
# veel üks moodus statistikute saamiseks grupiti
```

```
by(dat$kaal, dat$sugu, mean)
```

```
## dat$sugu: 1
```

```
## [1] 81.47059
```

```
## -----
```

```
## dat$sugu: 2
```

```
## [1] 67.45455
```

```
by(dat$kaal, dat$sugu, sd)
```

```
## dat$sugu: 1
```

```
## [1] 14.1405
```

```
## -----
```

```
## dat$sugu: 2
```

```
## [1] 15.62659
```

Pideva tunnuse kirjeldav statistika – näide 2

```
median(dat$kaal[dat$sugu==2])
```

```
## [1] 63.5
```

```
quantile(dat$kaal[dat$sugu==2])
```

```
##      0%      25%      50%      75%     100%  
## 45.00  56.25  63.50  75.00 116.00
```

```
# teistsugused kvaltiilid
```

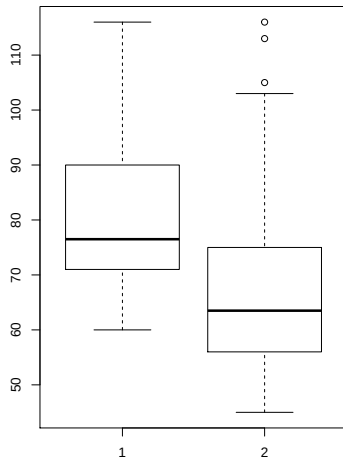
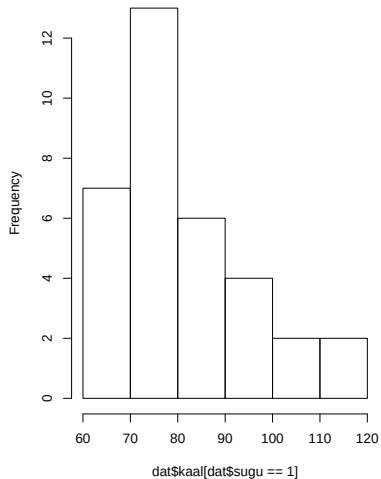
```
quantile(dat$kaal[dat$sugu==2],c(0,0.1,0.5,0.9,1))
```

```
##      0%     10%     50%     90%    100%  
## 45.0  49.5  63.5  86.5 116.0
```

```
hist(dat$kaal[dat$sugu==1]) # vt naiste kohta ka  
boxplot(dat$kaal~dat$sugu)
```

Pideva tunnuse kirjeldav statistika – näide 2

Histogram of dat\$kaal[dat\$sugu == 1]



Kokkuvõte R graafikast

- ▶ Kõige lihtsam ja universaalsem graafiline funktsioon on `plot()`. Et R on objekt-orienteeritud ja `plot()` on geneeriline funktsioon, siis sõltub selle väljund argumendina antud objekti(de) tüübist
- ▶ Töö R-i graafiku loomisel: alusta lihtsast joonisest nt `plot(x)`, seejärel lisa parameetreid, mis muudavad graafiku kujunduse sobivamaks ja ilusamaks
- ▶ R-is on palju funktsioone, mille abil saab lisada elemente juba valmis graafikule
- ▶ Kui soovitate luua publikatsioonikõlbulikku graafikat, siis on hea kirjutada R kood skriptiaknasse – tõenäoliselt tekib rahuldav joonis alles pärast paljukordset muudatuste tegemist ja skripti uuesti jooksumist