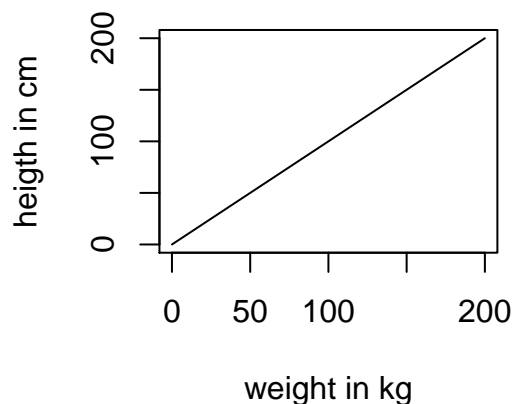


Bayesiaanlik statistika

Lineaarsed mudelid

Oletame, et me mõõtsime N inimese pikkuse cm-s ja kaalu kg-s ning meid huvitab, kuidas inimeste pikkus sõltub nende kaalust. Lihtsaim mudel pikkuse sõltuvusest kaalust on $\text{pikkus} = \text{kaal}$ (formaliseeritult: $y = x$) ja see mudel ennustab, et kui Johni kaal = 80 kg, siis John on 80 cm pikkune. Siin on pikkus muutuja, mille väärtust ennustatakse ja kaal muutuja, mille väärtuste põhjal ennustatakse pikkusi. Selle mudeli saame graafiliselt kujutada nii:

```
x <- 0:200 #y = kaal
y <- x # x = pikkus
plot(y~x,
     type="l",
     xlab="weight in kg",
     ylab="height in cm")
```



Mudeli keeles tähistame me seda, mida me ennustame (antud juhul pikkus) Y -ga ja seda, mille väärtuse põhjal me ennustame (antud juhul kaal) X -ga. Seega sirge mudeli matemaatiline formalism on $Y = X$.

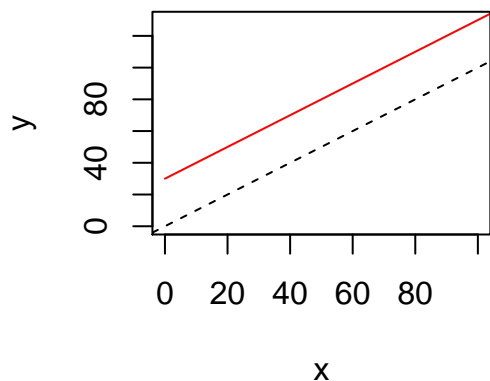
See on äärmiselt jäik mudel: sirge, mille asukoht on rangelt fikseeritud. Sirge lõikab y telge alati 0-s (mudeli keeles: sirge intercept ehk lõikepunkt Y teljel = 0) ja tema tõusunurk saab olla ainult 45 kraadi (mudeli keeles: mudeli slope ehk tõus = 1). Selle mudeli jäikus tuleneb sellest, et temas ei ole parameetreid, mille väärtusi me saaksime vabalt muuta ehk tuunida.

Mis juhtub, kui me lisame mudelisse konstandi, mille liidame x -i väärtustele?

$$y = a + x$$

See konstant on mudeli parameeter, mille väärtuse võime vabalt valida. Järgnevalt anname talle väärtuse 30 (ilma konkreetse põhjuseta).

```
x <- 0:200
a <- 30
y <- a + x
plot(y~x,
     xlim=c(0, 100),
     ylim=c(0, 130),
     col="red",
     type="l")
abline(c(0,1), lty=2)
```

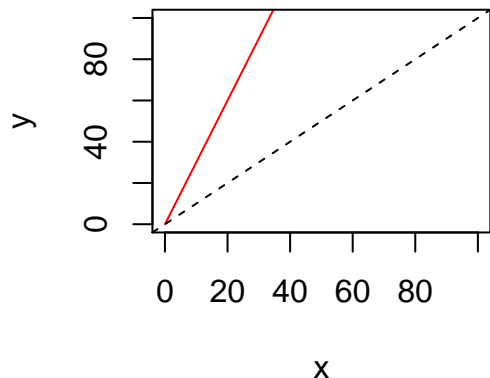


Meie konstant a määrab y väärtuse, kui $x = 0$, ehk sirge lõikepunkti y teljel. Teisisõnu, a = mudeli intercept
 Mis juhtub, kui me mitte ei liida, vaid korrutame x -i konstandiga?

$$y = bx$$

Jällegi, me anname mudeli parameetritele b suvalise väärtuse, 3.

```
x <- 0:200 #y = kaal
b <- 3
y <- b*x # x = pikkus
plot(y~x,
      xlim=c(0, 100),
      ylim=c(0, 100),
      col="red",
      type="l")
abline(c(0,1), lty=2)
```



Nüüd muutub sirge tõusunurk, ehk kui palju me ootame y -t muutumas, kui x muutub näiteks ühe ühiku võrra. Kui $b = 3$, siis x -i tõustes ühe ühiku võrra suureneb y kolme ühiku võrra. Proovi järgi, mis juhtub, kui $b = -3$.

Selleks, et sirget kahes dimensioonis vabalt liigutada, piisab kui me kombineerime eelnevad näited ühte:

$$y = a + bx$$

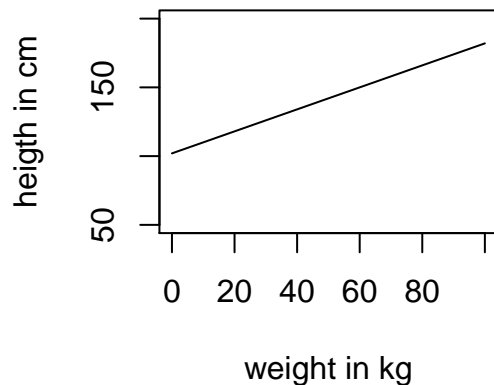
Selleks lisame mudelisse kaks parameetrit, intercept (a) ja tõus (b). Kui $a = 0$ ja $b = 1$, saame me eelpool kirjeldatud mudeli $y = x$. Kui $a = 102$, siis sirge lõikab y telge väärtusel 102. Kui $b = 0.8$, siis x -i tõustes 1 ühiku võrra tõuseb y -i väärtus 0.8 ühiku võrra. Kui $a = 100$ ja $b = 0$, siis saame sirge, mis on paraleelne x -teljega ja lõikab y -telge väärtusel 100. Seega, Teades a ja b väärtusi ning omistades x -le suvalise meid huvitava väärtuse, saab ennustada y -i keskmist väärtust sellel x -i väärtusel. Näiteks, olgu andmete vastu fititud mudel:

$$\text{pikkus(cm)} = 102 + 0.8 * \text{kaal(kg)} \text{ ehk}$$

$$y = 102 + 0.8x.$$

Omistades nüüd kaalule väärtuse 80 kg, tuleb mudeli poolt ennustatud keskmine pikkus $102 + 0.8 * 80 = 166$ cm. Iga kg lisakaalu ennustab mudeli kohaselt 0.8 cm võrra suuremat pikkust.

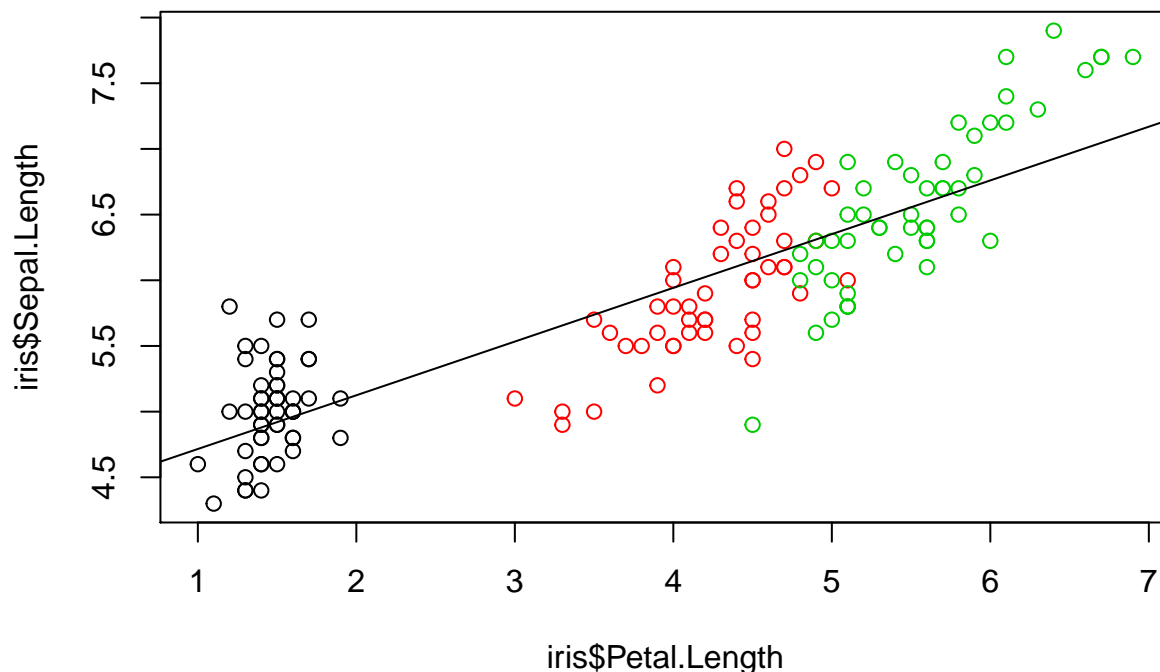
```
a <- 102
b <- 0.8
x <- 0:100
y <- a + b * x
plot(y~x,
     type="l",
     xlab="weight in kg",
     ylab="height in cm",
     ylim=c(50, 200))
```



See mudel ennustab, et 0 kaalu juures on pikku 102 cm, mis on rumal, aga mudelite puhul tavaline, olukord. Me tuunime mudelit andmete peal, mis ei sisalda 0-kaalu. Meie valimiandmed ei peegelda täpselt inimpopulatsiooni. Sirge mudel ei peegelda täpselt pikkuse-kaalu suhteid vahemikus, kus meil on reaalseid kaaluandmeid; ja ta teeb seda veelgi vähem seal, kus meil mõõdetud kaalusid ei ole. Seega pole mõtet imestada, miks mudeli intercept meie üle irvitab.

Kahe parameetriga sirge mudel ongi see, mida me fitime kahedimensiooniliste andmetega. Näiteks nii:

```
#read in the internal dataset iris
data(iris)
# fit a linear model and name the model object as m1
m1 <- lm(Sepal.Length ~ Petal.Length, data=iris)
# make a scatter plot, colored by the var called "Species"
plot(iris$Sepal.Length~iris$Petal.Length, col=iris$Species)
# draw the fitted regression line from m1
abline(m1)
```



Mudeli fitimine tähendab siin lihtsalt, et sirge on 2D ruumi asetatud nii, et see oleks võimalikult lähedal kõikidele punktidele.

oletame, et meil on n andmepunkti ja et me fitime neile sirge. Nüüd plotime fititud sirge koos punktidega ja tõmbame igast punktist mudelsirgeni joone, mis on paraleelne y -teljega. Seejärel mõõdame nende n joone pikkused. Olgu need pikkused a, b, \dots, i . `lm()` funktsioon fitib sirge niimoodi, et summa $a^2 + b^2 + \dots + i^2$ oleks minimaalne. Seda kutsutakse vähimruutude meetodiks.

Fititud koefitsientide väärtused saame nii

```
coef(m1)
```

```
## (Intercept) Petal.Length
## 4.3066034 0.4089223
```

Siin $a = (\text{Intercept})$ ja $b = \text{Petal.Length}$ ehk 0.41.

Neli mõistet

Mudel $y = a + bx$ on x ja y muutujad, ning a ja b on parameetrid. Muutujate väärtused fikseeritakse andmete poolt, parameetrid fititakse andmete põhjal. Fititud mudel ennustab igale x -i väärtusele vastava kõige tõenäolisema y väärtuse (y keskvaartuse sellel x -i väärtusel).

Y - mida me ennustame (dependent variable, predicted variable)

X - mille põhjal me ennustame (independent variable, predictor)

muutuja (variable) - iga asi, mida me valimis mõõdame (X ja Y on kaks muutujat). Muutujal on sama palju fikseeritud väärtusi kui meil on selle muutuja kohta mõõtmisandmeid.

parameeter (parameter) - mudeli koefitsient, millele võib omistada suvalisi väärtusi. Parameetreid tuunides fitime mudeli võimalikult hästi sobituma andmetega.

Mudel on matemaatilise formalism, mis püüab kirjeldada füüsikalist protsessi. Statistilise mudeli struktuuris on komponent, mis kirjeldab ideaalseid ennustusi (nn protsessi mudel) ja eraldi weakomponent (ehk veamudel), mis kirjeldab looduse varieeruvust nende ideaalsete ennustuste ümber. Mudeli koostisosad on (i) muutuja,

mille väärtusi ennustatakse, (ii), muutuja(d), mille väärtuste põhjal ennustatakse, (iii) parameetrid, mille väärtused fititakse ii põhjal ja (iv) konstandid.

Mudeli fittimine

Mudelid sisaldavad (1) matemaatilisi struktuure, mis määravad mudeli tüübi ning (2) parameetreid, mida saab andmete põhjal tuunida, niiviisi täpsustades mudeli kuju.

Seda tuunimist nimetatakse mudeli fittimiseks. Mudelit fittides on eesmärk saavutada antud tüüpi mudeli maksimaalne sobivus andmetega. Näiteks võrrand $y = a + bx$ määrab mudeli, kus $y = x$ on on see struktuur, mis tagab, et mudeli tüüp on sirge, ning a ja b on parameetrid, mis määravad sirge asendi. Seevastu struktuur $y = x + x^2$ tagab, et mudeli $y = a + b1x + b2x^2$ tüüp on parabool, ning parameetrite a , $b1$ ja $b2$ väärtused määravad selle parabooli täpse kuju. Ja nii edasi.

lineraarse mudeli parima sobivuse andmetega saab tagada kahel erineval viisil: (i) vähimruutude meetod mõõdab y telje suunaliselt iga andmepunkti kauguse mudeli ennustusest, võtab selle kauguse ruutu, summeerib kauguste ruudud ning leiab sirge asendi, mille korral see summa on minimaalne; (ii) Bayesi teoreem annab väheinformatiivse priori korral praktiliselt sama fiti.

Hea mudel on

- (1) võimalikult lihtsa struktuuriga, mille põhjal on veel võimalik teha järeldusi protsessi kohta, mis genereeris mudeli fittimiseks kasutatud andmeid;
- (2) sobitub piisavalt hästi andmetega (eriti uute andmetega, mida ei kasutatud selle mudeli fittimiseks), et olla relevantne andmeid genereeriva protsessi kirjeldus;
- (3) genereerib usutavaid simuleeritud andmeid.

Sageli fititakse samade andmetega mitu erinevat tüüpi mudelit ja püütakse otsustada, milline neist vastab kõige paremini eeltoodud tingimustele. Näiteks, kui sirge suudab kaalu järgi pikkust ennustada paremini kui parabool, siis on sirge mudel paremas kooskõlas teadusliku hüpoteesiga, mis annaks mehhanismi protsessile, mille käigus kilode lisandumine viiks laias kaaluvahemikus inimeste pikkuse kasvule ilma, et pikkuse kasvu tempo kaalu tõustes langeks.

See, et teie andmed sobivad hästi mingi mudeliga, ei tähenda automaatselt, et see fakt oleks teaduslikult huvitav. Mudeli parameetrid on mõtekaud mudeli matemaatilise kirjelduse kontekstis, aga mitte tingimata suure maailma põhjusliku seletamise kontekstis. Siiski, kui mudeli matemaatiline struktuur loodi andmeid genereeriva loodusliku protsessi olemust silmas pidades, võib mudeli koefitsientide uurimisest selguda olulisi tõsiasju suure maailma kohta.

Mudeli fittimine: X ja Y saavad oma väärtused otse andmetest; parameetrid võivad omandada ükskõik millise väärtuse.

Fititud mudelist ennustamine: X -le saab omistada ükskõik millise väärtuse; parameetrite väärtused on fikseeritud; Y väärtus arvutatakse mudelist.

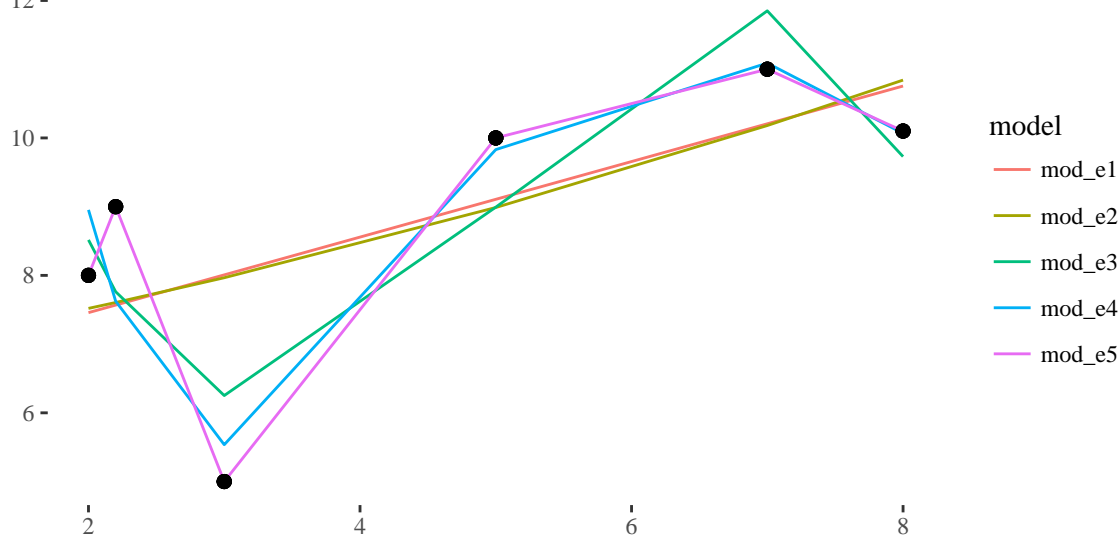
Üle- ja alafittimine

Osad mudelite tüübid on vähem paindlikud kui teised (parameetreid tuunides on neil vähem liikumisruumi). Kuigi sellised mudelid sobituvad halvemini andmetega, võivad need ikkagi paremini kui mõni paindlikum mudel välja tuua andmete peidetud olemuse. Mudeldamine eeldab, et me usume, et meie andmetes leidub nii müra (mida mudel võiks ignoreerida), kui signaal (mida mudel püüab tabada). Kuna mudeli jaoks näeb müra samamoodi välja, kui signaal, on iga mudel kompromiss üle- ja alafittimise vahel. Me lihtsalt loodame, et meie mudel on piisavalt jäik, et mitte liiga palju müra modelleerida ja samas piisavalt paindlik, et piisaval määral signaali tabada.

Üks kõige jäigemaid mudeleid on sirge, mis tähendab, et sirge mudel on suure tõenäosusega alafittitud. Keera sirget kuipalju tahad, ikka ei sobitu ta enamiku andmekogudega. Ja need vähesed andmekogud, mis

sirge mudeliga sobivad, on genereeritud teatud tüüpi lineaarsete protsesside poolt. Sirge on seega üks kõige paremini tõlgendatavaid mudeleid. Teises äärmuses on polünoomised mudelid, mis on väga paindlikud, mida on väga raske tõlgendada ja mille puhul esineb suur mudeli ülefittimise oht. Ülefittitud mudel järgib nii täpselt valimiandmeid, et sobitub hästi valimis leiduva juhusliku müraga ning seetõttu sobitub halvasti järgmise valimiga samast populatsioonist (igal valimil on oma juhuslik müra). Üldiselt, mida rohkem on mudelis tuunitavaid parameetreid, seda paindlikum on mudel, seda kergem on seda valimiandmetega sobitada ja seda raskem on seda tõlgendada. Veelgi enam, alati on võimalik konstrueerida mudel, mis sobitub täiuslikult kõikide andmepunktidega (selle mudeli parameetrite arv = N). Selline mudel on täpselt sama informatiivne kui andmed, mille põhjal see fititi — ja täiesti kasutu.

12 -



Joonis:

Kasvava paindlikusega polünoomised mudelid. *mod_e1* on sirge võrrand $y = a + b_1x$ (2 parameetrit: a ja b_1), *mod_e2* on lihtsaim võimalik polünoom: $y = a + b_1x + b_2x^2$ (3 parameetrit), ..., *mod_e5*: $y = a + b_1x + b_2x^2 + b_3x^3 + b_4x^4 + b_5x^5$ (6 parameetrit). *mod_e5* vastab täpselt andmepunktidele ($N = 6$).

Vähimruutude meetodil fititud mudeleid saame võrrelda AIC-i näitaja järgi. AIC - Akaike Informatsiooni Kriteerium - vaatab mudeli sobivust andmetega ja mudeli parameetrite arvu. Väikseim AIC tähistab parimat fiti väikseima parameetrite arvu juures (kompromissi) ja väikseima AIC-ga mudel on eelistatuim mudel. Aga seda ainult võrreldud mudelite hulgas. AIC-i absoluutväärtus ei loe - see on suhteline näitaja.

```
AIC(mod_e1, mod_e2, mod_e3, mod_e4, mod_e5)
```

```
##          df      AIC
## mod_e1   3 27.77993
## mod_e2   4 29.76669
## mod_e3   5 26.21330
## mod_e4   6 25.11245
## mod_e5   7      -Inf
```

AIC näitab, et parim mudel on *mod_e4*. Aga kas see on ka kõige kasulikum mudel? Mis siis, kui 3-s andmepunkt on andmesisestaja näpuviga?

Ülefittimise vältimiseks kasutavad Bayesi mudelid informatiivseid prioreid, mis välistavad ekstreemsed parameetriväärtused.

Vt <http://eleventh.org/blog/2017/08/22/there-is-always-prior-information/>

kaks lineaarse mudeli laiendust.

mitme sõltumatu prediktoriga mudel

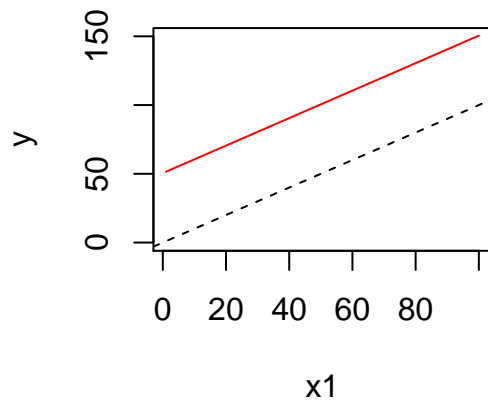
Esiteks vaatame mudelit, kus on mitu prediktorit x_1, x_2, \dots, x_n , mis on additiivse mõjuga. See tähendab, et me liidame nende mõjud, mis omakorda tähendab, et me usume, et $x_1 \dots x_n$ mõjud y -i väärtusele on üksteisest sõltumatud. Mudel on siis kujul

$$y = a + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

Kõigepealt ehitame lihtsa variandi sellest mudelist: $y = x_1 + x_2$ ($a = 0, b_1 = b_2 = 1$), ja plotime selle mudeli y sõltuvusena x_1 -st juhul, kus x_2 on fikseeritud oma keskmisele väärtusele

```
x1 <- 1:100
x2 <- 1:100

y <- x1 + mean(x2)
plot(y~x1,
     type="l",
     ylim=c(0, 150), col="red")
abline(c(0,1), lty=2)
```

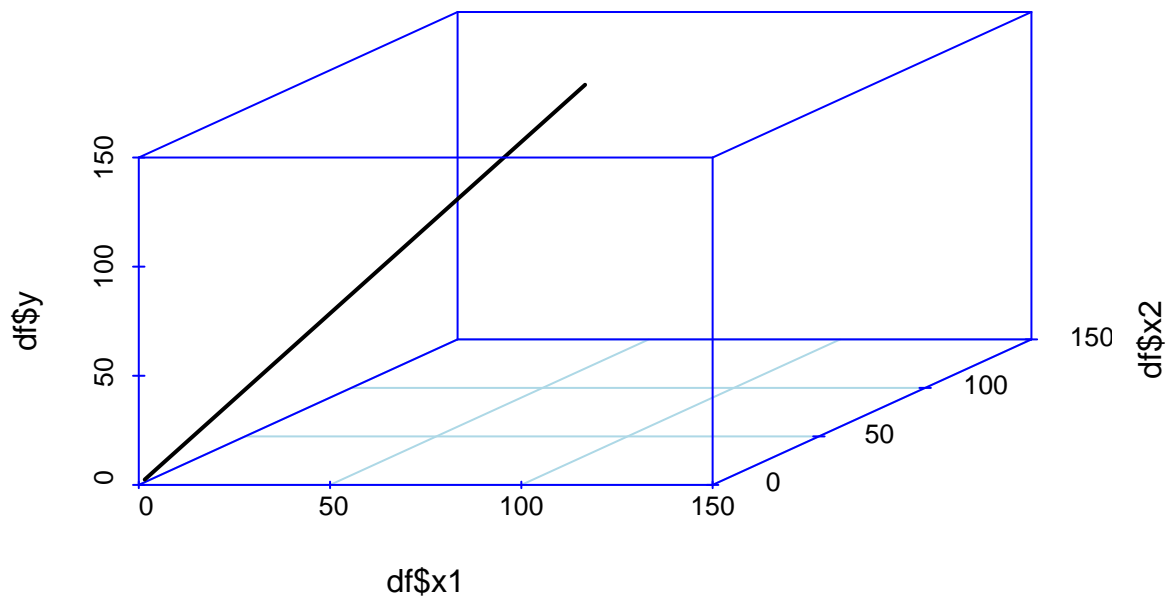


Nagu näha, tulemuseks on paraleelne sirge mudeliga $y = x_1$, mille *intercept* = $\text{mean}(x_2)$. Muidugi võime me plottida selle mudeli suvalise x_2 väärtuse korral või lasta muutuma hoopis x_2 ja fikseerida x_1 väärtuse.

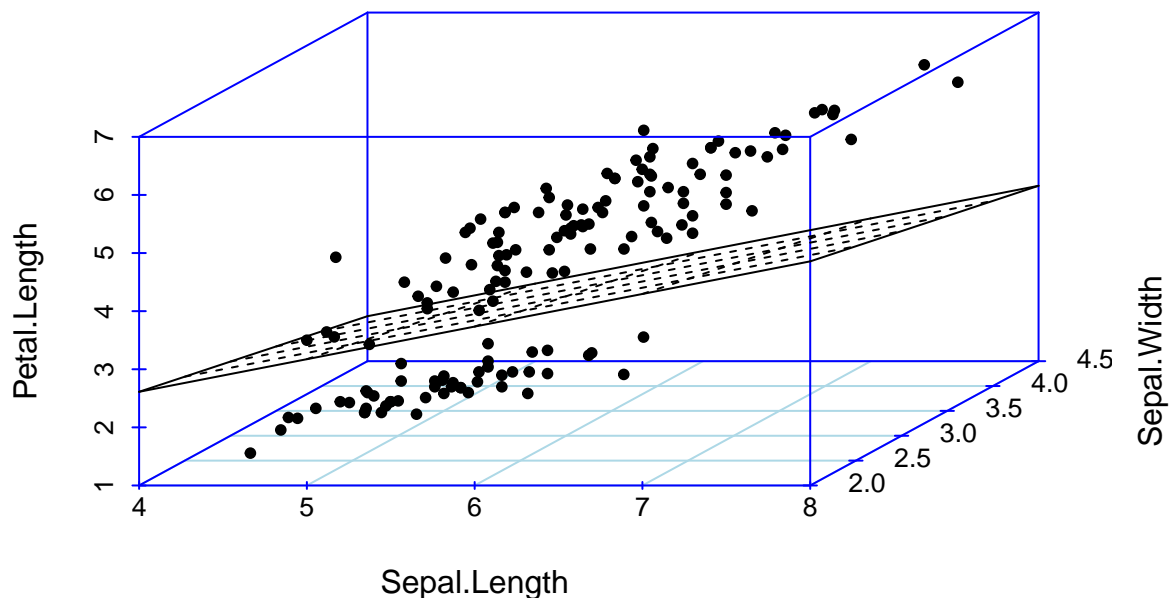
mitme prediktoriga mudeli iga prediktori tõus (beta koefitsient) ütleb, mitme ühiku võrra ennustab mudel y muutumist juhul kui see prediktor muutub ühe ühiku võrra ja kõik teised prediktorid ei muutu üldse. Seega pole teiste (kollapseeritud) prediktorite absoluutväärtus ennustusel oluline.

Me võime siiski joonistada 3D pildi olukorrast, kus nii x_1 kui x_2 omandavad rea väärtusi. Selles mudelis erineb sirge $y = x_1 + x_2$ tõus sirge $y = x_1$ tõusust, aga tegemist on ikkagi sirgega – mis sest, et 3D ruumis.

```
#library(rgl) #plot3d()
library(scatterplot3d)
x1 <- 1:100
x2 <- 1:100
y <- x1 + x2
df <- data.frame(y=y, x1=x1, x2=x2)
scatterplot3d(df$x1, df$x2, df$y,
              col.axis="blue",
              col.grid="lightblue",
              type="l", lwd=2,
              zlim=c(0, 150), xlim=c(0, 150), ylim=c(0, 150))
```



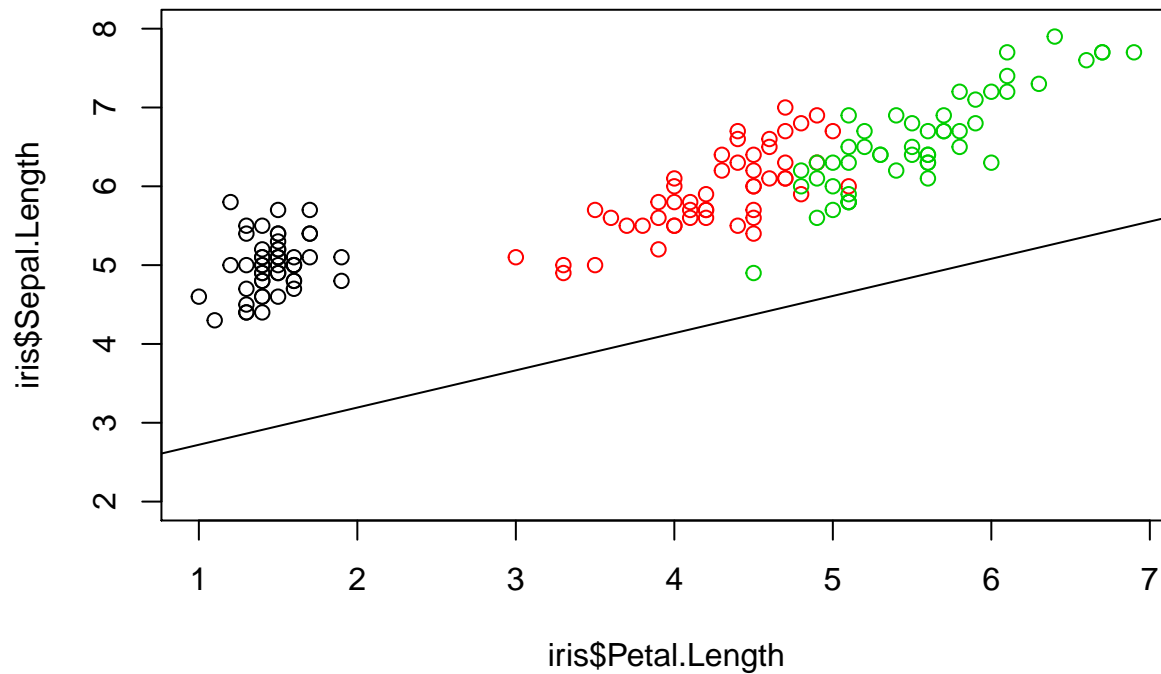
Kui meie andmed on mõõdetud 3D-s ja me tahame ennustada ühe muutuja väärtust kahe teise muutuja väärtuste põhjal (meil on 2 prediktorit), siis tuleb meie 3 parameetriga lineaarne regressioonimudel tasapinna kujul. Kui meil on 4 prediktoriga mudel, siis me liigume 4-mõõtmelisse ruumi, jne. 3D ruumi on veel võimalik mõistlikult plottida.



Seda mudelit saab kaeda 2D ruumis, juhul kui kollapseerida kolmas mõõde ühele väärtusele (pole oluline millisele).

```
data(iris)
m1 <- lm(Sepal.Length~Petal.Length + Sepal.Width, data=iris)
plot(iris$Sepal.Length~iris$Petal.Length, ylim=c(2,8), col=iris$Species)
abline(m1)
```

```
## Warning in abline(m1): only using the first two of 3 regression
## coefficients
```

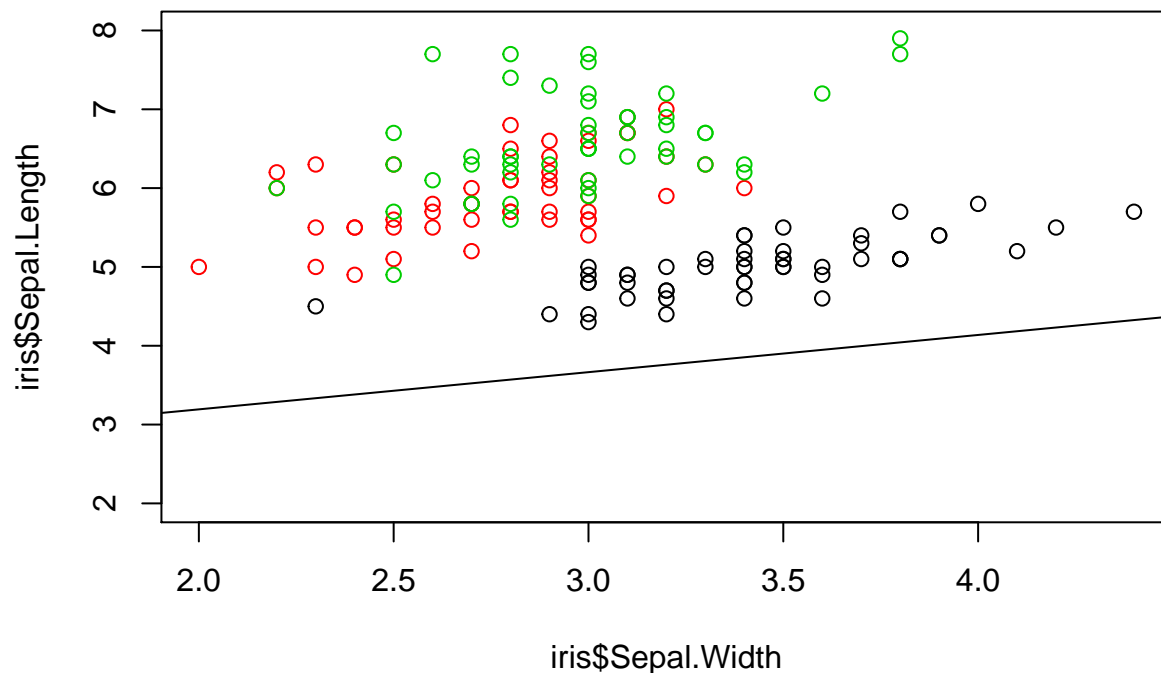



Sellise mudeli plottimise tõlgendus on, et me modelleerime muutuja nimega Sepal.Length väärtuse sõltuvuse muutuja Petal.Length väärtusest, juhul kui muutuja Petal.Length on konstant.

Nüüd kollapseerime muutuja Petal.Length ja plotime Sepal.Length sõltuvuse Sepal.Width-ist, suvalisel Petal.Length väärtusel.

```
plot(iris$Sepal.Length~iris$Sepal.Width, ylim=c(2,8), col=iris$Species)
abline(m1)
```

```
## Warning in abline(m1): only using the first two of 3 regression
## coefficients
```



Nõnda võrdleme kahe mudeli koefitsiente

```
m <- lm(data=iris, Sepal.Length~Petal.Length)
m1 <- lm(data=iris, Sepal.Length~Petal.Length + Sepal.Width)
coef(m); coef(m1)
```

```
## (Intercept) Petal.Length
## 4.3066034 0.4089223

## (Intercept) Petal.Length Sepal.Width
## 2.2491402 0.4719200 0.5955247
```

Nagu näha, mudeli m b1 koefitsient erineb oluliselt mudeli m1 vastavast koefitsiendist. Kumb mudel on siis parem? AIC-i järgi on m1 kõvasti parem, kui m.

```
AIC(m, m1)
```

```
## df AIC
## m 3 160.0404
## m1 4 101.0255
```

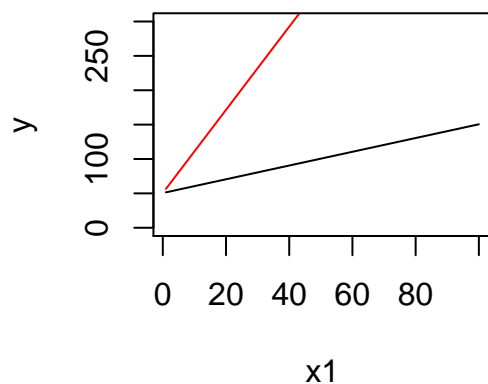
Interaktsioonimudel - ühe prediktori mõju sõltub teise prediktori väärtusest

$$y = a + b_1x_1 + b_2x_2 + b_3x_1x_2$$

Siin on mudeli $Y = a + b_1X_1 + b_2X_2 + b_3X_1X_2$ koefitsientide tõlgendamine raskem. Sellise mudeli järgi erineb sirge tõus b_1 erinevatel b_2 väärtustel, ja erinevuse määr sõltub b_3 -st sõltuvalt interaktsiooni tugevusest. Samamoodi ja sümmeetriliselt erineb ka b_2 väärtus erinevatel b_1 väärtustel.

Interaktsiooni modelleerime korrutades läbi x_1 ja x_2 ja andes sellele korrutisele kaalu b_3 (antud juhul 0.1). Jällegi plotime kõigepealt olukorra, kus x_1 omab rida väärtusi, aga x_2 on fikseeritud oma keskvärtusele.

```
x1 <- 1:100
x2 <- 1:100
b3 <- 0.1
y <- x1 + mean(x2) + b3*x1*mean(x2)
y1 <- x1 + mean(x2)
plot(y~x1,
     type="l",
     ylim=c(0, 300), col="red")
lines(y1~x1)
```

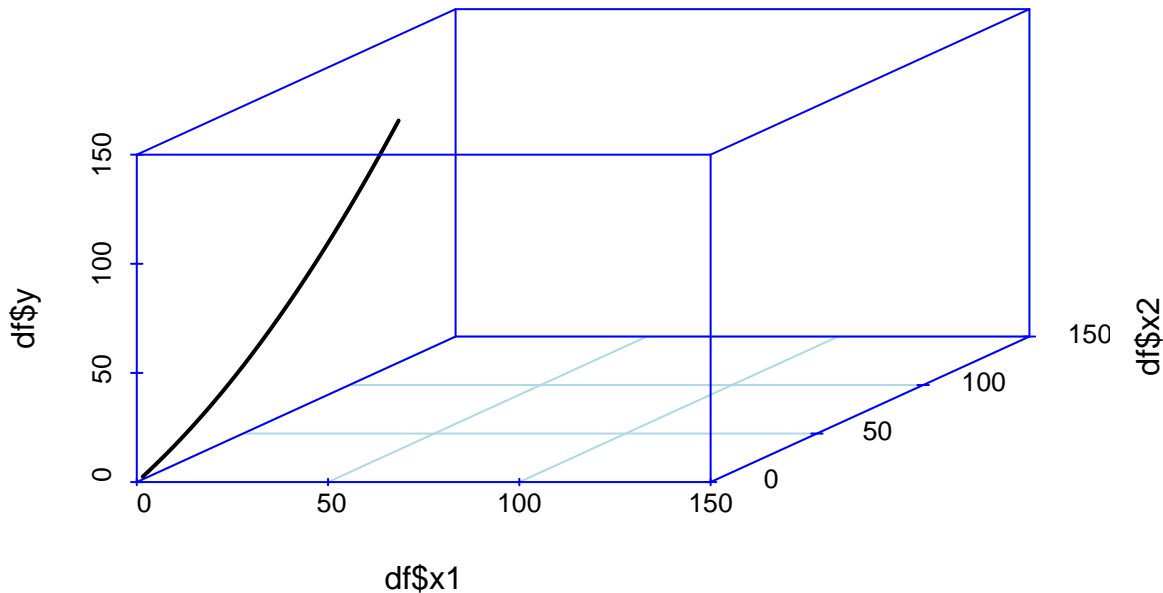


Nagu näha, korrutamistehe viib selleni, et interaktsioonimudeli tõus erineb ilma interaktsioonita mudeli tõusust. Kui aga interaktsioonimudel plottida välja 3D-s üle paljude x_1 ja x_2 väärtuste, saame me regressioonikurvi (mitte sirge), kus b_3 annab kurvatuuri.

```

x1 <- 1:100
x2 <- 1:100
b3 <- 0.03
y <- x1 + x2 + b3*x1*x2
df <- data.frame(y=y, x1=x1, x2=x2)
scatterplot3d(df$x1, df$x2, df$y, col.axis="blue",
              col.grid="lightblue", type="l", lwd=2,
              zlim=c(0, 150), xlim=c(0, 150), ylim=c(0, 150))

```



Interaktsioonimudeli 2D avaldus on kurvatuuriga tasapind, kusjuures kurvatuuri määrab interaktsioonikoefitsient (mudelis $y = a + b_1x_1 + b_2x_2 + b_3x_1x_2$ on see b_3).

Interaktsiooniga mudel on veidike eelistatud võrreldes m1-ga, aga see eelis on väga väike. Seega, lihtsuse huvides võime, aga ei pruugi, eelistada m1-e.

```

m <- lm(data=iris, Sepal.Length~Petal.Length)
m1 <- lm(data=iris, Sepal.Length~Petal.Length + Sepal.Width)
m2 <- lm(data=iris, Sepal.Length~Petal.Length + Sepal.Width + Petal.Length * Sepal.Width)
AIC(m, m1, m2)

```

```

##      df      AIC
## m    3 160.04042
## m1   4 101.02550
## m2   5  99.77334

```

Veamudel

Eelpool kirjeldatud mudelid on deterministlikud — nad ei sisalda hinnangut andmete varieeruvusele ennustuse ümber. Neid kutsutakse ka **protsessi mudeliteks** sest nad modelleerivad protsessi täpselt. Ehk, kui mudel ennustab, et 80 kg inimene on 166 cm pikkune, siis protsessi mudel ei ütle, kui suurt kaalust sõltumatut pikkuste varieeruvust võime oodata 80 kg-ste inimeste hulgas. Selle hinnangu andmiseks tuleb mudelile lisada veel üks komponent, **veamudel** ehk veakomponent, mis sageli tuuakse sisse normaaljaotuse kujul. Veakomponent modelleerib üksikute inimeste pikkuste varieeruvust (mitte keskmise pikkuse varieeruvust) igal mõeldaval ja mittemõeldaval kaalul. Tänu sellele ei ole mudeli ennustused enam deterministlikud, vaid tõenäosuslikud.

Bioloogid, erinevalt füüsikutest, usuvad, et valimisisene andmete varieeruvus on tingitud enam bioloogilisest varieeruvusest, kui mõõtmisveast. Aga loomulikult sisaldub selles ka mõõtmisviga. Lihtsuse huvides räägime edaspidi veamudelist, selle asemel, et öelda “bioloogilise varieeruvuse ja veamudel”.

Kuidas weakomponent lineaarsesse mudelisse sisse tuua?

ilma weakomponendita mudel:

$$y = a + bx$$

Weakomponent tähendab, et y -i väärtus varieerub ümber mudeli poolt ennustatud keskvaartuse ja näiteks seda varieeruvust normaaljaotusega modelleerides saame

$$y \sim \text{dnorm}(\mu, \sigma)$$

kus μ on mudeli poolt ennustatud keskvaartus ja σ on mudeli poolt ennustatud standardhälve ehk varieeruvus andmepunktide tasemel. Tilde \sim tähistab seose tõenäosuslikkust. Tegelikult on ka veamudel keskvaartuse ehk μ ennustus deterministlik ja σ töötab originaalsel andmetasemel, mitte keskvaartuste tasemel. See võimaldab protsessi mudeli veamudelisse sisse kirjutada lihtsalt μ ümber defineerides.

Sirge mudelisse varieeruvuse sisse toomiseks defineerime μ ümber nõnda:

$$\mu = a + bx,$$

mis tähendab, et

$$y \sim \text{dnorm}(a + bx, \sigma)$$

See ongi sirge mudel koos weakomponendiga. Seega on sellel lineaarsel regressioonimudelil kolm parameetrit: intercept a , tõus b ja “veaparaameeter” σ . Sellist mudelit on mõistlik fitida Bayesi teoreemi abil. Bayesi meetodiga fititud mudel, mida kutsutakse posterrioriks, näitab, millised kombinatsioonid nendest kolmest parameetrist usutavalt koos esinevad, ja millised mitte. Seega on fititud 3 parameetriga bayesi mudel 3-dimensionaalne tõenäosusjaotus (3D posterrior). Muidugi saame ka ükshaaval välja plottida kolm 1D posteriori, millest igaüks iseloomustab üht parameetrit ning on kollapseeritud üle kahe ülejäänud parameetri. 4. peatükis õpime selliste mudelitega töötama.

Kõik statistilised mudelid on tõenäosusmudelid ning sisaldavad weakomponenti.

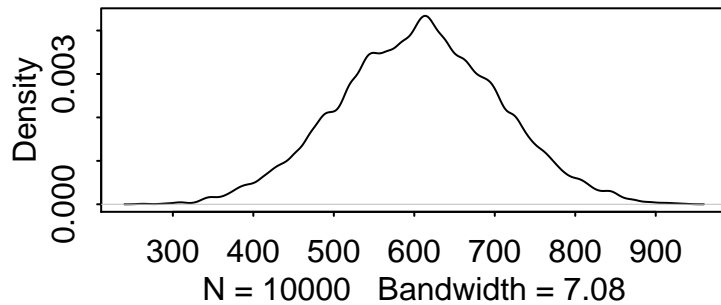
Muide, kõik veamudelid, millega me edaspidi töötame, modelleerivad igale x -i väärtusele (kaalule) sama suure y -i suunalise varieeruvuse (pikkuste sd). Suurem osa statistikast kasutab eeldusi, mida keegi päriselt tõe pähe ei võta, aga millega on arvutuslikus mõttes lihtsam elada.

Enimkasutatud veamudel on normaaljaotus.

Alustuseks simuleerime lihtsate vahenditega looduslikku protsessi, mille tulemusel tekib normaaljaotus.

Oletame, et bakteri kasvukiirust mõjutavad 12 geeni, mille mõjud võivad olla väga erineva tugevusega, kuid mille mõjude suurused ei sõltu üksteisest. Seega nende 12 geeni mõjud kasvukiirusele liituvad. Järgnevas koodis võtame 12 juhuslikku arvu 1 ja 100 vahel (kasutades `runif()` funktsiooni). Need 12 arvu näitavad 12 erineva geeni individuaalsete mõjude suurusi bakteritüve kasvukiirusele. Meil on seega kuni 100-kordsed erinevused erinevate geenide mõjude suuruste vahel. Seejärel liidame need 12 arvu. Nüüd võtame uue 12-se valimi ja kordame eelnevat. Me teeme seda 10 000 korda järjest ja plotime saadud 10 000 arvu (10 000 liitmistehte tulemust) tihedusfunktsioonina.

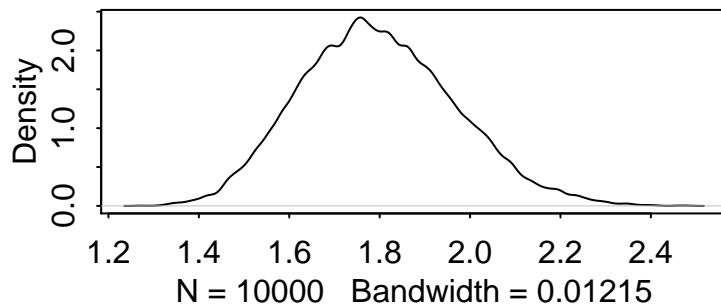
```
kasv <- replicate( 10000 , sum( runif( 12, 1, 100 ) ) )
dens( kasv )
```



Selles näites võrdub iga andmepunkt 10 000st ühe bakteritüve kasvukiiruse mõõtmisega. Seega, antud eelduste korral on bakteritüvede kasvukiirused normaaljaotusega.

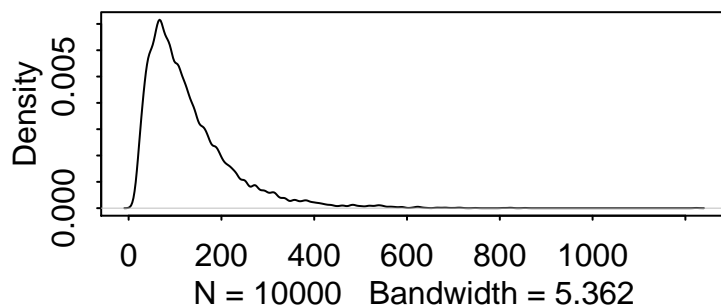
Nüüd vaatame, mis juhtub, kui 12 geeni mõjud ei ole üksteisest sõltumatud. Kui 12 geeni on omavahel vastasmõjudes, siis nende geenide mõjud korrutuvad, mitte ei liitu. (Korrutamine pole ainus viis, kuidas vastasmõjusid modelleerida, küll aga kõige levinum.) Kõigepealt vaatleme juhtu, kus 12 geeni on kõik väikeste mõjudega ning seega mitte ühegi geeni mõju ei domineeri teiste üle. Seekord genereerime 12 juhuslikku arvu 1 ja 1.1 vahel. Siin tähendab arv 1.1 kasvu tõusu 10% võrra. Seejärel korrutame need 12 arvu, misjärel kordame eelnevat 10 000 korda.

```
kasv <- replicate( 10000 , prod( runif( 12, 1, 1.1 ) ) )
dens( kasv )
```



Tulemuseks on jällegi normaaljaotus. Selles näites olid üksikud interakteeruvad geenid ükshaaval väikeste mõjudega ja ühegi geeni mõju ei domineerinud teiste üle. Mis juhtub, kui mõnel geenil on kuni 2 korda suurem mõju kui teisel?

```
kasv <- replicate( 10000 , prod( runif(12,1,2) ) )
dens( kasv )
```

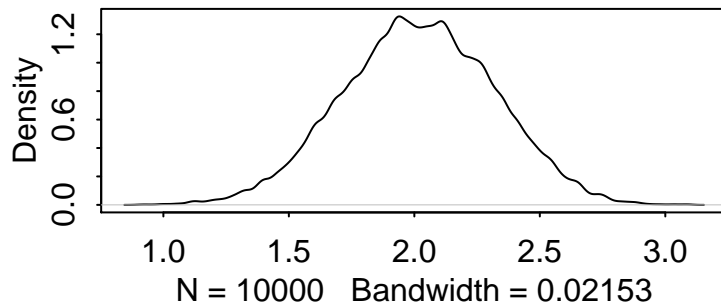


Nüüd on tulemuseks log-normaaljaotus. Mis teie arvate, kas teie poolt uuritavat tunnust mõjutavad faktorid, mis omavahel ei interakteeru või kui interakteeruvad, on kõik ühtlaselt väikeste efektidega? Või on tegu vastasmõjudes olevate faktoritega, millest osad on palju suuremate mõjudega, kui teised? Ühel juhul eelistate te normaaljaotust, teisel juhul peate õppima töötama ka lognormaaljaotusega.

Kui me vaatame samu andmeid logaritmilises skaalas, avastame, et need andmed on normaaljaotusega. See

ongi andmete logaritmime mõte.

```
kasv <- replicate( 10000 , log10(prod( runif(12,1,2) ) ) )
dens( kasv )
```



Normaaljaotuse mudel väikestel valimitel

Oletame, et meil on kolm andmepunkti ning me usume, et need andmed on juhuslikult tõmmatud normaaljaotusest või sellele lähedasest jaotusest. Normaaljaotuse mudelit kasutades me sisuliselt deklareerime, et me usume, et kui me oleksime olnud vähem laisad ja 3 mõõtmise asemel sooritanuks 3000, siis need mõõtmised sobituksid piisavalt hästi meie 3 väärtuse peal fititud normaaljaotusega. Seega, me usume, et omades 3 andmepunkti me teame juba umbkaudu, millised tulemused me oleksime saanud korjates näiteks 3 miljonit andmepunkti. Oma mudelist võime simuleerida ükskõik kui palju andmepunkte.

Aga pidage meeles, et selle mudeli fittimiseks kasutame me ainult neid andmeid, mis meil päriselt on — ja kui meil on ainult 3 andmepunkti, on tõenäoline, et fititud mudel ei kajasta hästi tegelikkust.

Halvad andmed ei anna kunagi head tulemust.

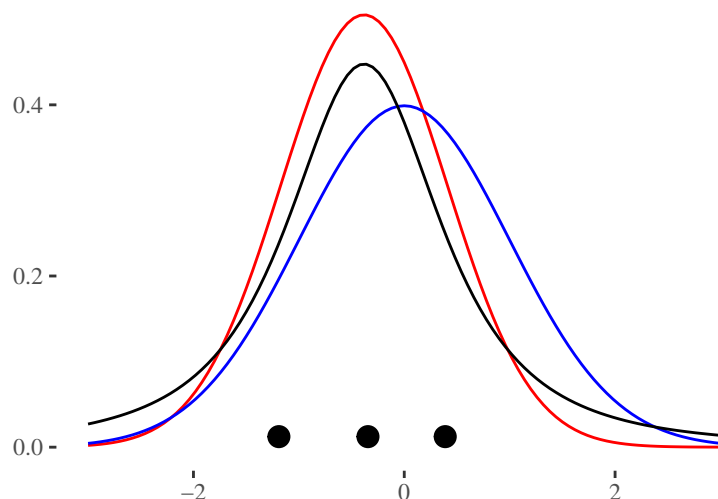
Eelnev ei kehti Bayesi mudelite kohta, mis toovad priorite kaudu sisse lisainfot, mis ei kajastu valimiandmetes ja võib analüüsi päästa.

Kuidas panna skeptik uskuma, et statistilised meetodid töötavad halvasti väikestel valimitel? Siin aitab simulatsioon, kus me tõmbame 3-se valimi etteantud populatsioonist ning üritame selle valimi põhjal ennustada selleasama populatsiooni struktuuri. Kuna tegemist on simulatsiooniga, teame täpselt, et populatsioon, kust me tõmbame oma kolmese valimi, on normaaljaotusega, et tema keskväärtsus = 0 ja et tema sd = 1. Me fitime oma valimi andmetega 2 erinevat mudelit: normaaljaotuse ja Studenti t jaotuse.

```
library(brms) #t jaotuse modelleerimiseks

#pakett ggfortify pakub alt lahendust jaotuste joonistamiseks
set.seed(19)
df <- tibble(a=rnorm(3))

df %>% ggplot(aes(x=a)) +
  geom_dotplot() +
  stat_function(fun=dnorm, args=list(mean=mean(df$a),sd=sd(df$a)), colour="red") +
  stat_function(fun=dnorm, args=list(mean=0, sd=1), colour="blue") +
  stat_function(fun=dstudent_t, args=list(df=2, mu=mean(df$a), sigma=sd(df$a)))+
  xlim(-3, 3) + ylim(0, 0.55) + labs(x=NULL, y=NULL) + theme_tufte()
```



Joonis: juhuvalim normaaljaotusest, mille keskmine=0 ja $sd=1$ ($n=3$; andmepunktid on näidatud mustade munadena). Sinine joon - populatsioon, millest tõmmati valim; punane joon - normaaljaotuse mudel, mis on fititud valimi andmetel; must joon - Studenti t jaotuse mudel, mis on fititud samade andmetega.

Siin saame hinnata mudelite fitte jumala positsioonilt, võrreldes fititud mudelite jaotusi “tõese” sinise jaotusega. Mõlemad mudelid on süstemaatiliselt nihutatud väiksemate väärtuste poole ja alahindavad varieeruvust. t jaotuse mudel on oodatult paksemate sabadega ja ennustab 0-st kaugele palju rohkem väärtusi kui normaaljaotuse mudel. Kuna me teame, et populatsioon on normaaljaotusega, pole väga üllatav, et t jaotus modelleerib seda halvemini kui normaaljaotus.

Igal juhul, mõni teine juhuvalim annaks meile hoopis teistsugused mudelid, mis rohkem või vähem erinevad algsest populatsioonist.

Mis juhtub kui me kasutame oma normaaljaotuse mudelit uute andmete simuleerimiseks? Kui lähedased on need simuleeritud andmed populatsiooni andmetega ja kui lähedased valimi andmetega, millega me normaaljaotuse mudeli fittisime?

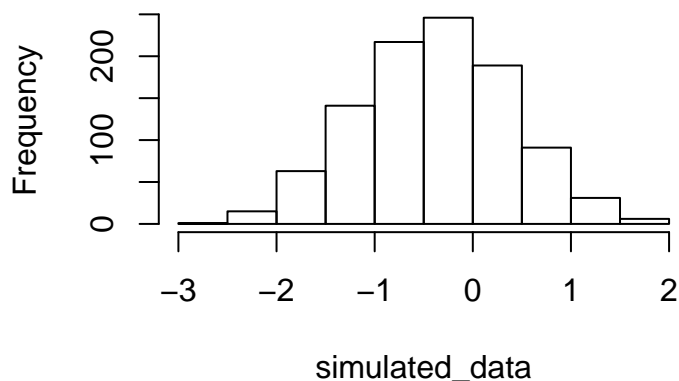
```
set.seed(19) #muudab simulatsiooni korratavaks
#tõmbame 3 juhuslikku arvu normaaljaotusest, mille keskväärtus = 0 ja sd = 1.
df <- tibble(sample_data=rnorm(3))
#fitime normaaljaotuse mudeli valimi keskmise ja sd-ga
mean(df$sample_data); sd(df$sample_data)

## [1] -0.3817353
## [1] 0.7896821

#simuleerime 1000 uut andmepunkti fititud mudelist
simulated_data <- rnorm(1000, mean(df$sample_data), sd(df$sample_data))
#arvutame simuleeritud andmete keskmise ja sd ning joonistame neist histogrammi
mean(simulated_data); sd(simulated_data); hist(simulated_data)

## [1] -0.3848133
## [1] 0.7749198
```

Histogram of simulated_data



Nagu näha, on uute (simuleeritud) andmete keskväärts ja SD väga sarnased algsete andmete omale, mida kasutasime mudeli fittimisel. Kahjuks ei ole need aga kaugeltki nii sarnased algsele jaotusele, mille kuju me püüame oma andmete ja mudeli pealt ennustada. Seega on meie mudel üle-fittitud, mis tähendab, et ta kajastab liigselt neid valimi aspekte, mis ei peegelda algse populatsiooni omadusi. Loomulikult ei vasta ükski mudel päriselt tegelikkusele. Küsimus on pigem selles, kas mõni meie mudelitest on piisavalt hea, et olla kasulik. Vastus sellele sõltub, milleks plaanime oma mudelit kasutada.

```
mean(simulated_data > 0); mean(simulated_data > 1)
```

```
## [1] 0.317
```

```
## [1] 0.037
```

Kui populatsiooniväärtustest on 50% suuremad kui 0, siis mudeli järgi vaevalt 32%. Kui populatsiooniväärtustest on 16% suuremad kui 1, siis mudeli järgi vaevalt 4%. See illustreerib hästi mudeli kvaliteeti.

```
library(brms)
sim_t <- rstudent_t(1000, 2, mean(df$sample_data), sd(df$sample_data))
mean(sim_t > 0); mean(sim_t > 1)
```

```
## [1] 0.338
```

```
## [1] 0.11
```

Samad ennustused t jaotusest on isegi paremad! Aga kumb on ikkagi parem mudel populatsioonile?

normaaljaotuse ja lognormaaljaotuse erilisus

Normaaljaotus ja lognormaaljaotus on erilised sest

- (1) keskne piirteoreem ütleb, et olgu teie valim ükskõik millise jaotusega, paljudest valimitest arvutatud **aritmeetilised keskmised** on alati enam-vähem normaaljaotusega (kui $n > 30$). Selle matemaatilise formalismi tuletus füüsikalisse maailma on nn “elementaarsete vigade hüpotees”, mille kohaselt paljude väikeste üksteisest sõltumatute juhuslike efektide (vigade) summa annab tulemuseks normaaljaotuse. Paraku annavad enamus bioloogilisi mõõtmisi eranditult mitte-negatiivseid tulemusi. Sageli on selliste mõõtmiste tulemuste jaotused ebasümmeetrilised (v.a. siis, kui $cv = sd/mean$ on väike) ja siis on meil sageli tegu lognormaaljaotusega, mis tekib log-normaalsete muutujate korrutamisel (mitte liitmisest, nagu normaaljaotuse puhul). Keskne piirteoreem 2: suvalise jaotusega muutujate **geomeetrilised keskmised** on lognormaaljaotusega. Elementaarsete vigade hüpotees 2: Kui juhuslik varieeruvus tekib paljude juhuslike efektide korrutamisel, on tulemuseks lognormaaljaotus. Lognormaaljaotuse elementide (arvude) logaritmimeisel saame normaaljaotuse.

- (2) Mõlemad jaotused (normaal ja lognormaal) on maksimaalse entroopiaga jaotused. Entroopiat vaadeldakse siin informatsiooni/müra kaudu — maksimaalse entroopiaga süsteem sisaldab maksimaalselt müra ja minimaalselt informatsiooni (Shannoni informatsiooniteooria). See tähendab, et väljaspool oma parameetrite tuunitud väärtusi on need normaal- ja lognormaaljaotused minimaalselt informatiivsed. Näiteks normaaljaotusel on kaks parameetrit, mu ja sigma (ehk keskmine ja standardhälve). Seega, andes normaaljaotusele ette keskväärtuse ja standardhälbe fikseerime üheselt jaotuse ehk mudeli kuju ja samas lisame sinna minimaalselt muud (sooviamtut) informatsiooni. Teised maksimaalse entroopiaga jaotused on eksponentsiaalne jaotus, binoomjaotus ja poissoni jaotus. Maksimaalse entroopiaga jaotused sobivad hästi Bayesi prioriteks sest me suudame paremini kontrollida, millist informatsiooni me neisse surume.

Küsimused, mida statistika küsib

Statistika abil saab vastuseid järgmistele küsimustele:

- 1) kuidas näevad välja teie andmed ehk milline on just teie andmete jaotus, keskväärtus, varieeruvus ja koos-varieeruvus? Näiteks, mõõdetud pikkuste ja kaalude koos-varieeruvust saab mõõta korrelatsioonikordaja abil.
- 2) mida me peaksime teie valimi andmete põhjal uskuma populatsiooni parameetri tegeliku väärtuse kohta? Näiteks, kui meie andmete põhjal arvatud keskmine pikkus on 178 cm, siis kui palju on meil põhjust arvata, et tegelik populatsiooni keskmine pikkus > 185 cm?
- 3) mida ütleb statistilise mudeli struktuur teadusliku hüpoteesi kohta? Näiteks, kui meie poolt mõõdetud pikkuste ja kaalude koos-varieeruvust saab hästi kirjeldada kindlat tüüpi lineaarse regressioonimudeliga, siis on meil ehk tõendusmaterjali, et pikkus ja kaal on omavahel sellisel viisil seotud ja eelistatud peaks olema teaduslik teooria, mis just sellise seose tekkimisele bioloogilise mehhanismi annab.
- 4) mida ennustab mudel tuleviku kohta? Näiteks, meie lineaarne pikkuse-kaalu mudel suudab ennustada tulevikus kogutavaid pikkuse andmeid. Aga kui hästi?

statistika ülesanne on lähtuvalt piiratud hulgast andmetest ja mudelitest kvantifitseerida parimal võimalikul viisil kõhedust, mida peaksime tundma vastates eeltoodud küsimustele.

Statistika ei vasta otse teaduslikele küsimustele ega küsimustele päris maailma kohta. Statistilised vastused jäävad alati kasutatud andmete ja mudelite piiridesse. Sellega seoses peaksime eelistama hästi kogutud rikkalikke andmeid ja paindlikke mudeleid. Siis on lootust, et hüpe mudeli koefitsientidest päris maailma kirjeldamiseks tuleb üle kitsama kuristiku. Bayesil on siin eelis, sest osav statistik suudab koostöös teadlastega priori mudelisse küllalt palju kasulikku infot koguda. Samas, amatöör suudab bayesi abil samavõrra kähki keerata. Mida paindlikum on meetod, seda vähem automaatne on selle mõistlik kasutamine.

Kuidas näevad välja teie andmed?

summaarsed statistikud

Summaarne statistik = üks number.

Milliseid statistikuid arvutada ja milliseid vältida, sõltub statistilisest mudelist

summaarse statistika abil iseloomustame a) tüüpilist valimi liiget (keskmist), b) muutuja sisest varieeruvust, c) erinevate muutujate (pikkus, kaal vms) koos-varieeruvust

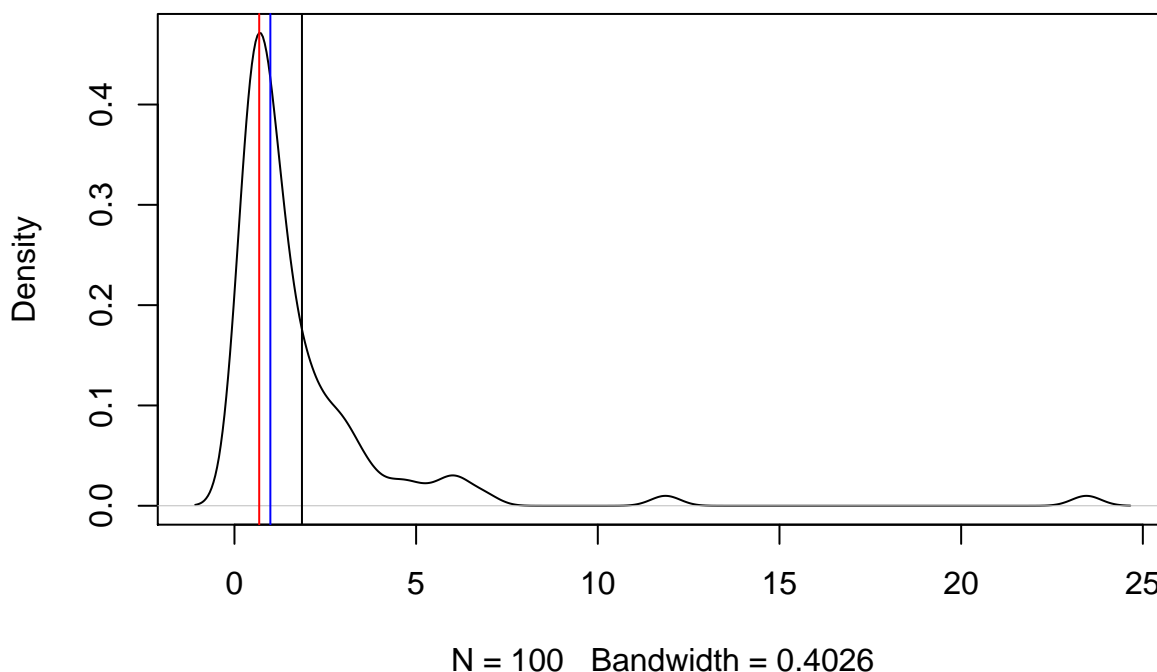
keskväärtused

Keskväärtust saab mõõta paaril tosinal erineval viisil, millest järgnevalt kasutame kolme või nelja. Enne kui te arvutama kukute, mõelge järele, miks te soovite keskväärtust teada. Kas teid huvitab valimi tüüpiline liige? Kuidas te sooviksite seda tüüpilisust defineerida? Kas valimi keskmise liikmena või valimi kõige arvukama liikmena? või veel kuidagi? See, millist keskväärtust kasutada sõltub sageli andmejaotuse kujust. Sümmeetrilisi jaotusi on lihtsam iseloomustada ja mitmetipulised jaotused on selles osas kõige kehvemad.

Mina eelistan selliseid nõuandeid (mis on rangelt soovituslikud):

- (1) Kui valim on normaaljaotusega (histogramm on sümmeetriline), hinda tüüpilist liiget läbi aritmeetilise keskmise (mean).
- (2) Muidu kasuta mediaani (median). Kui valim on liiga väike, et jaotust hinnata ($n < 4$), eelista mediaani. Mediaani saamiseks järjestatakse mõõdetud väärtused suuruse järgi ja võetakse selle rea keskmine liige. Mediaan on vähem tundlik ekstreemsete väärtuste (outlierite) suhtes kui mean.
- (3) Valimi kõige levinumat esindajat iseloomustab mood ehk jaotuse tipp. Seda on aga raskem täpselt määrata ja mitmetipulisel jaotusel on mitu moodi. Töötamisel posterioorsete jaotustega on mood sageli parim lahendus.

density.default(x = andmed, adjust = 1)



Joonis: Simuleeritud lognormaaljaotusega andmed. Punane joon - mood; sinine joon - mediaan; must joon - aritmeetiline keskmine (mean). Milline neist vastab parimini teie intuitsiooniga nende andmete “keskväärtusest”? Miks?

muutuja sisene varieeruvus

Mean-iga käib kokku standardhälve (SD).

SD on sama ühikuga, mis andmed (ja andmete keskväärtus). Statistike hulgas eelistatud formaat on mean (SD), mitte mean (+/- SD). 1 SD katab 68% normaaljaotusest, 2 SD – 96% ja 3 SD – 99%. Normaaljaotus

langeb servades kiiresti, mis tähendab, et tal on peenikesed sabad ja näiteks 5 SD kaugusel keskmisest paikneb vaid üks punkt miljonist.

Näiteks: inimeste IQ on normaaljaotusega, $\text{mean}=100$, $\text{sd}=15$. See tähendab, et kui sinu IQ=115 (ülikooli astujate keskmine IQ), siis on tõenäosus, et juhuslikult kohatud inimene on sinust nutikam, 18% ($((100\% - 68\%)/2 = 18\%)$).

Kui aga “tegelikul” andmejaotusel on “paks saba” või esinevad outlierid, siis normaaljaotust eeldav mudel tagab ülehinnatud SD ja seega ülehinnatud varieeruvuse. Kui andmed saavad olla ainult positiivsed, siis $\text{SD} > \text{mean}/2$ viitab, et andmed ei sobi normaaljaotuse mudeliga (sest mudel ennustab negatiivsete andmete esinemist küllalt suure sagedusega).

Standardhälve on defineeritud ka mõnede teiste jaotuste jaoks peale normaaljaotuse (Poissoni jaotus, binoomjaotus). Funktsioon `sd()` ja selle taga olev valem on loodud normaaljaotuse tarbeks ja neid alternatiivseid standardhälbeid ei arvuta. Igale jaotusele, mida me oskame integreerida, saab ka integraali abil standardhälbe arvutada, mis on täpselt õige katvusega. Seega tasub mees pidada, et tavapärane viis sd arvutamiseks kehtib normaaljaotuse mudeli piirides ja ei kusagil mujal!

Kui andmed ei sobi normaaljaotusesse, võib pakkuda kahte alternatiivset lahendust:

(1) logaritmi andmed.

Kui kõik andmeväärtused on positiivsed ja andmed on lognormaaljaotusega, siis logaritmimine muudab andmed normaalseks. Logaritmitud andmetest tuleks arvutada aritmeetiline keskmine ja SD ning seejärel mõlemad anti-logaritmid (näiteks kui $\log_2(10) = 3.32$, siis antilogaritm sellest on $2^{3.32} = 10$). Sellisel juhul avaldatakse lõpuks geomeetriline keskmine ja multiplikatiivne SD algses lineaarses skaalas (multiplikatiivne $\text{SD} = \text{geom mean} \times \text{SD}$; $\text{geom mean}/\text{SD}$). Geomeetriline keskmine on alati väiksem kui aritmeetiline keskmine. Lisaks on SD intervall nüüd asümmeetriline ja SD on alati > 0 . See protseduur tagab, et 68% lognormaalsetest andmetest jääb 1 SD vahemikku ning 96% andmetest jääb 2 SD vahemikku.

Kui lognormaalsetele andmetele arvutada tavaline sd lineaarses skaalas kasutades `sd()` funktsiooni, mille algoritm on välja töötatud spetsiifiliselt normaalsete andmete jaoks, siis tuleb SD sageli palju laiem kui peaks ja hõlmab ka negatiivseid väärtusi (pea mees, et SD definitsiooni järgi jääb 96% populatsioonist 2 SD vahemikku).

Sageli on aga negatiivsed muutuja väärtused võimatud (näiteks nädalas suitsetatud sigarettide arv). See on näide halvast mudelist! Kui te rakendate tavapäraselt `sd()` funktsiooni teadmata jaotusega andmetele, võite siiski kindel olla, et 2 SD hõlmab mitte vähem kui 75% populatsiooni andmetest.

Kirjutame logaritmimise kaudu avaldatud multiplikatiivse SD arvutamiseks funktsiooni `multiplicative_sd()`:

```
multiplicative_sd <- function(x) {  
  x <- na.omit(x)  
  log_data <- log10(x)  
  log_mean <- mean(log_data)  
  log_sd <- sd(log_data)  
  geom_mean <- 10**log_mean  
  mult_sd <- 10**log_sd  
  lower1 <- geom_mean/mult_sd  
  upper1 <- geom_mean * mult_sd  
  lower2 <- geom_mean/(mult_sd**2)  
  upper2 <- geom_mean * (mult_sd**2)  
  Mean <- mean(x)  
  lower3 <- mean(x) - sd(x)  
  upper3 <- mean(x) + sd(x)  
  lower4 <- mean(x) - sd(x)*2  
  upper4 <- mean(x) + sd(x)*2  
}
```

```

results <- tibble(SD=c("multiplicative_SD",
                      "multiplicative_2_SD",
                      "additive_SD",
                      "additive_2_SD"),
                 MEAN=c(geom_mean,
                       geom_mean,
                       Mean,
                       Mean),
                 lower=c(lower1,
                         lower2,
                         lower3,
                         lower4),
                 upper=c(upper1,
                         upper2,
                         upper3,
                         upper4) )

results
}

```

```

multiplicative_sd(andmed)

```

```

## # A tibble: 4 x 4
##           SD      MEAN    lower    upper
##           <chr>    <dbl>    <dbl>    <dbl>
## 1 multiplicative_SD 1.084891  0.4010893 2.934481
## 2 multiplicative_2_SD 1.084891  0.1482845 7.937367
## 3 additive_SD 1.857924 -0.9636351 4.679482
## 4 additive_2_SD 1.857924 -3.7851938 7.501041

```

Tavalise aritmeetilise keskmise asemel on meil nüüd geomeetriline keskmine. Võrdluseks on antud ka tavaline (aritmeetiline) keskmine ja (aditiivne) SD. Additiivne SD on selle jaotuse kirjeldamiseks selgelt ebaadekvaatne (vt jaotuse pilti ülalpool ja võrdle mitiplikatiivse SD-ga).

Kuidas aga töötab mitiplikatiivne standardhälve normaaljaotusest pärit andmetega? Kui mitiplikatiivse sd rakendamine normaalsete andmete peal viib katastroofini, siis pole sel statistikul suurt kasutusruumi.

```

set.seed(5363)
norm_andmed <- rnorm(3, 100, 20)
multiplicative_sd(norm_andmed)

```

```

## # A tibble: 4 x 4
##           SD      MEAN    lower    upper
##           <chr>    <dbl>    <dbl>    <dbl>
## 1 multiplicative_SD 108.1088  92.80205 125.9403
## 2 multiplicative_2_SD 108.1088  79.66252 146.7128
## 3 additive_SD 108.9603  92.08395 125.8367
## 4 additive_2_SD 108.9603  75.20756 142.7131

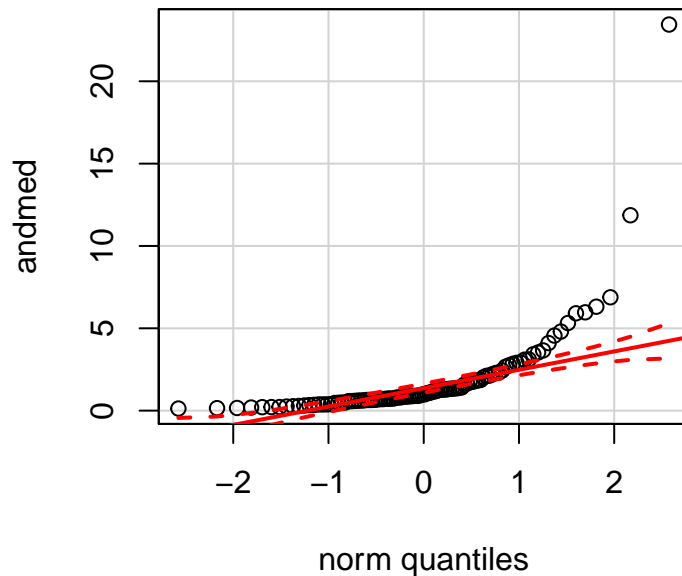
```

Nagu näha, on mitiplikatiivse sd kasutamine normaalsete andmetega pigem ohutu (kui andmed on positiivsed). Arvestades, et additiivne SD on lognormaalsete andmete korral kõike muud kui ohutu ning et lognormaaljaotus on bioloogias üsna tavaline (eriti ensüümreaktsioonide ja kasvuprotsesside juures), on mõistlik alati kasutada `multiplicative_sd()` funktsiooni. Kui mõlema SD väärtused on sarnased, siis võib loota, et andmed on normaalsed ning saab refereede rõõmuks avaldada tavapärase additiivse SD.

kui $n < 10$, siis mõlemad SD-d alahindavad tehnilistel põhjustel tegelikku sd-d. Ettevaatust väikeste valimitega!

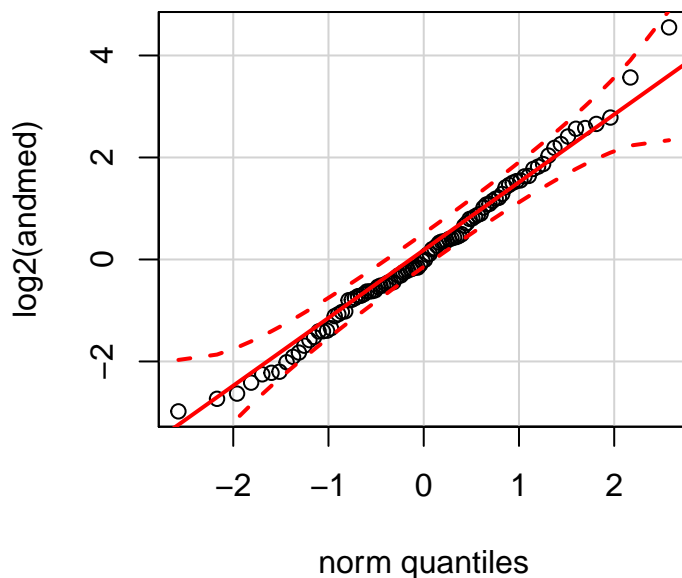
Vahest tekkib teil vajadus empiiriliselt määrata, kas teie andmed on normaaljaotusega. Enne kui seda tegema asute, peaksite mõistma, et see, et teie valim ei ole normaalne, ei tähenda automaatselt, et populatsioon, millest see valim tõmmati, ei oleks normaaljaotusega. Igal juhul, valimiandmete normaalsuse määramiseks on kõige mõistlikum kasutada qq-plotti. QQ-plot (kvantiil-kvantiil plot) võrdleb andmete jaotust ideaalse normaaljaotusega andmepunkti haaval. Kui empiiriline jaotus kattub referentsjaotusega, siis on tulemuseks sirgel paiknevad punktid. Järgneval qq plotil on näha, mis juhtub, kui plottida lognormaalseid andmeid normaaljaotuse vastu:

```
library(car)
qqPlot(andmed)
```



Nüüd joonistame qq-ploti logaritmitud andmetele.

```
qqPlot(log2(andmed))
```



Pole kahtlust, andmed on logaritmitud kujul normaaljaotusega.

qqPlot() võimaldab võrrelda teie andmeid ükskõik millise R-is leiduva jaotusega (?car::qqPlot).

Normaaljaotuse kindlakstegemiseks on loodud ka peotäis sageduslikke teste, mis annavad väljundina p väärtuse. Nende kasutamisest soovitame siiski hoiduda, sest tulemused on sageli ebakindlad, eriti väikestel ja suurtel valimitel. Mõistlikum on vaadata kõikide andmepunktide plotti normaaljaotuse vastu, kui jõllitada ühte numbrit (p), mille väärtus, muuseas, monotooniliselt langeb koos valimi suuruse kasvuga.

(2) iseloomusta andmeid algses skaalas: mediaan (MAD).

MAD — median absolute deviation — on vähem tundlik outlierite suhtes ja ei eelda normaaljaotust. Puuduseks on, et MAD ei oma tõlgendust, mille kohaselt ta hõlmaks kindlat protsenti populatsiooni või valimi andmejaotusest. Seevastu sd puhul võime olla kindlad, et isegi kõige hullema jaotuse korral jäävad vähemalt 75% andmetest 2 SD piiridesse.

```
mad(andmed, constant = 1)
```

```
## [1] 0.5950562
```

Ära kunagi avalda andmeid vormis: mean (MAD) või median (SD). Korrektne vorm on mean (SD) või median (MAD).

muutujate koos-varieeruvus

Andmete koos-varieeruvust mõõdetakse korrelatsiooni abil. Tulemuseks on üks number - korrelatsioonikordaja r, mis varieerub -1 ja 1 vahel.

$r = 0$ – kahte tüüpi mõõtmised (x =pikkus, y =kaal) samadest mõõteobjektidest varieeruvad üksteisest sõltumatult. $r = 1$: kui ühe muutuja väärtus kasvab, kasvab ka teise muutuja väärtus alati täpselt samas proportsioonis. $r = -1$: kui ühe muutuja väärtus kasvab, kahaneb teise muutuja väärtus alati täpselt samas proportsioonis.

Kui r on -1 või 1, saame me x väärtust teades täpselt ennustada y väärtuse (ja vastupidi, teades y väärtust saame täpselt ennustada x väärtuse).

Kuidas tõlgendada aga tulemust $r = 0.9$? Mitte kuidagi. Selle asemel tõlgendame $r^2 = 0.9^2 = 0.81$ – mis tähendab, et x -i varieeruvus suudab seletada 81% y varieeruvusest ja vastupidi, et Y -i varieeruvus suudab seletada 81% X -i varieeruvusest.

Korrelatsiooni saab mõõta mitmel viisil (?cor.test, method=). Kõige levinum on Pearsoni korrelatsioonikoeffitsient, mis eeldab, (i) et me mõõdame pidevaid muutujaid, (ii) et valim on esinduslik populatsiooni suhtes, (iii) et populatsiooniandmed on normaaljaotusega ja (iv) et igal mõõteobjektil on mõõdetud 2 omadust (pikkus ja kaal, näiteks). Tuntuim alternatiiv on mitteparameetriline Spearmani korrelatsioon, mis ei eelda andmete normaaljaotust ega seda, et mõõdetakse pidevaid suurusid (ordinaalsed andmed käivad kah). Kui kõik Pearsoni korrelatsiooni eeldused on täidetud ja te kasutate siiski Spearmani korrelatsiooni, siis on teie arvutus ca 10% vähemefektiivne.

```
#correlation<-cor.test(iris$Sepal.Length,  
# iris$Sepal.Width, na.rm=T, method = "pearson")  
#names(correlation)  
#str(correlation)  
#correlation$conf.int  
cor(iris$Sepal.Length, iris$Sepal.Width, use="complete.obs")
```

```
## [1] -0.1175698
```

```
#complete.obs uses only such observations where neither x or y value is NA
```

Korrelatsioonikordaja väärtus sõltub mitte ainult andmete koos-varieeruvusest vaid ka andmete ulatusest. Suurema ulatusega andmed X ja/või Y teljel annavad keskeltläbi 0-st kaugemal oleva korrelatsioonikordaja. Selle pärast sobib korrelatsioon halvasti näiteks korduskatsete kooskõla mõõtmiseks.

Lisaks, korrelatsioonikordaja mõõdab vaid andmete *lineaarset* koos-varieeruvust: kui andmed koos-varieeruvad mitte-lineaarselt, siis võivad ka väga tugevad koos-varieeruvused jääda märkamatuks.

Moraal seisneb selles, et enne korrelatsioonikordaja arvutamist tasub alati plottida andmed, et veenduda võimaliku seose lineaarsuses. Lineaarsuse puudumine andmete koosvarieeruvuse mustris tähendab, et korrelatsioonikordaja tuleb kindlasti eksitav. Kordamisküsimus: miks on paneelil a) r ligikaudu 0?

Korrelatsioonikordaja mõõdab pelgalt määra, mil üks muutuja muutub siis, kui teine muutuja muutub. Seega ei ole suurt mõtet arvutada korrelatsioonikordajat juhul kui me teame ette seose olemasolust kahe muutuja vahel. Näiteks, kui sama entiteeti mõõdetakse kahel erineval viisil, või kahes korduses, või kui esimene muutuja arvutatakse teise muutuja kaudu.

Kõik summaarsed statistikud kaotavad enamuse teie andmetes leiduvast infost – see kaotus on õigustatud ainult siis, kui teie poolt valitud statistik iseloomustab hästi andmete sügavamat olemust (näiteks tüüpilist mõõtmistulemust või andmete varieeruvust).

Korrelatsioonimaatriksi saab niimoodi:

```
#numeric columns only!  
print(psych::corr.test(iris[-5], use="complete"), short = FALSE)
```

EDA — eksploratoorne andmeanalüüs

Kui ühenumbiline andmete summeerimine täidab eelkõige kokkuvõtliku kommunikatsiooni eesmärgi, siis EDA on suunatud teadlasele endale. EDA eesmärk on andmeid eelkõige graafiliselt vaadata, et saada aimu 1) andmete kvaliteedist ja 2) lasta andmetel kõneleda “sellisena nagu nad on” ja sugereerida uudseid teaduslikke hüpoteese. Neid hüpoteese peaks siis testima formaalse statistilise analüüsi abil (ptk järeldev statistika). Näiteid erinevate graafiliste lahenduste kohta vt graafika peatükist.

EDA: mida rohkem graafikuid, seda rohkem võimalusi uute mõtete tekkeks!

EDA on rohkem kunst kui teadus selles mõttes, et teil on suur vabadus küsida selle abil erinevaid küsimusi oma andmete kohta. Ja seda nii tehnilisest aspektist lähtuvalt (milline on minu andmete kvaliteet?), kui teaduslikke küsimusi küsides (kas muutuja A võiks põhjustada muutusi muutujas B?).

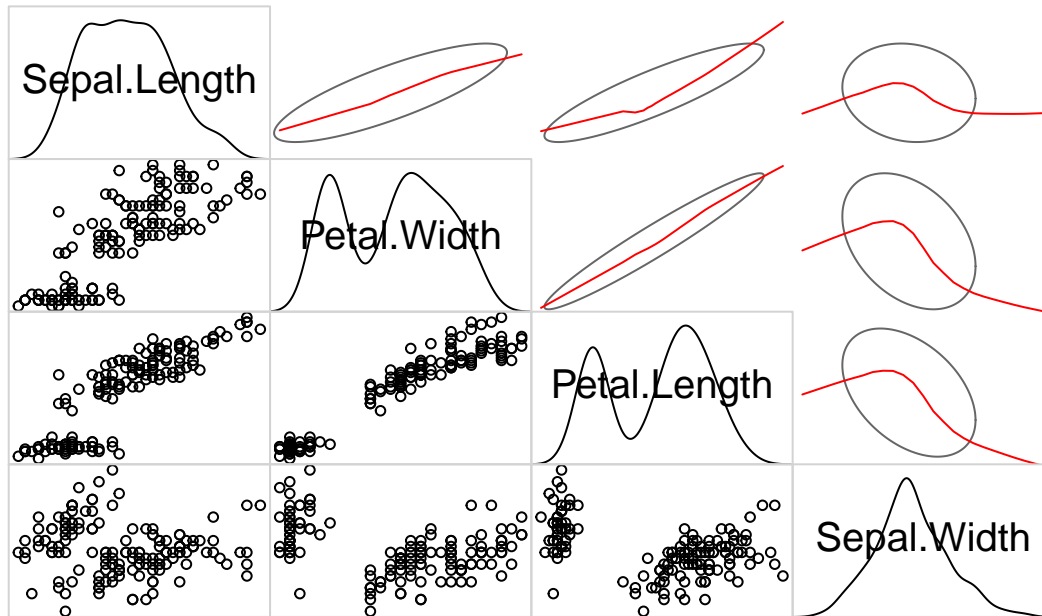
Mõned üldised soovitusid võib siiski anda.

1. alusta analüüsi tasemest, kus andmed on kõige inforikkamad — toorandmete plottimisest punktidenä. Kui andmehulk ei ole väga massiivne, näitab see hästi nii andmete kvaliteeti, kui ka võimalikke sõltuvussuhteid erinevate muutujate vahel.

Millised korrelatsioonid võiksid andmetes esineda?

```
library(corrgram) #PCA for ordering  
  
corrgram(iris, order=TRUE,  
  lower.panel = panel.pts,  
  upper.panel = panel.ellipse,  
  diag.panel = panel.density,  
  main="Correlogram of Iris dataset")
```

Correlogram of Iris dataset



2. vaata andmeid numbrilise kokkuvõtena.

```
psych::describe(iris)
```

```
##          vars    n mean   sd median trimmed  mad min max range  skew
## Sepal.Length    1 150 5.84 0.83   5.80    5.81 1.04 4.3 7.9   3.6  0.31
## Sepal.Width     2 150 3.06 0.44   3.00    3.04 0.44 2.0 4.4   2.4  0.31
## Petal.Length     3 150 3.76 1.77   4.35    3.76 1.85 1.0 6.9   5.9 -0.27
## Petal.Width      4 150 1.20 0.76   1.30    1.18 1.04 0.1 2.5   2.4 -0.10
## Species*         5 150 2.00 0.82   2.00    2.00 1.48 1.0 3.0   2.0  0.00
##          kurtosis   se
## Sepal.Length   -0.61 0.07
## Sepal.Width     0.14 0.04
## Petal.Length   -1.42 0.14
## Petal.Width    -1.36 0.06
## Species*       -1.52 0.07
```

Siin pööra kindlasti tähelepanu tulpadele min ja max, mis annavad kiire võimalusi outliereid ära tunda. Kontrolli, kas andmete keskmised (mediaan, mean ja trimmed mean) on üksteisele piisavalt lähedal — kui ei ole, siis on andmete jaotus pika õlaga, ja kindlasti mitte normaalne. Kontrolli, kas erinevate muutujate keskvaartused ja hälbed on teaduslikus mõttes usutavas vahemikus. Ära unusta, et ka väga väike standardhälve võib tähendada, et teie valim ei peegelda bioloogilist varieeruvust populatsioonis, mis teile teaduslikku huvi pakub. NB! selles `psych::describe()` väljundis on mad läbi korrutatud konstandiga 1.4826, mis toob selle väärtuse lähemale sd-le. Seega on mad siin sd robustne analoog — kui mad on palju väiksem sd-st, siis on karta, et muutujas on outliereid.

3. kontrolli NA-de esinemist oma andmetes VIM paketi abil või käsitsi (vt esimene ptk). Kontrolli, et NA-d ei oleks tähistatud mingil muul viisil (näiteks 0-i või mõne muu numbriga). Kui vaja, rekodeeri NAd. Mõtle selle peale, millised protsessid looduses võiksid genereerida puuduvaid andmeid. Kui NA-d ei jaotu andmetes juhuslikult, võib olla hea mõte andmeid imputeerida (vt hilisemaid ptk, bayesianlik imputeerimine). Näiteks, kui ravimiuuringust kukuvad eeskätt välja patsiendid, kellel ravim ei tööta, on ilmselt halb mõte nende patsientide andmed lihtsalt uuringust välja vistata (muidugi, kui te ei esinda

kasumit taotleva ettevõtte huve). Kui NA-d jaotuvad juhuslikult, mõtle sellele, kas sa tahad NA-dega read tabelist välja visata, või hoopis osad muutujad, mis sisaldavad liiga palju NA-sid, või mitte midagi välja vistata. NB! NA-dega andmed ei sobi hästi regresiooniks.

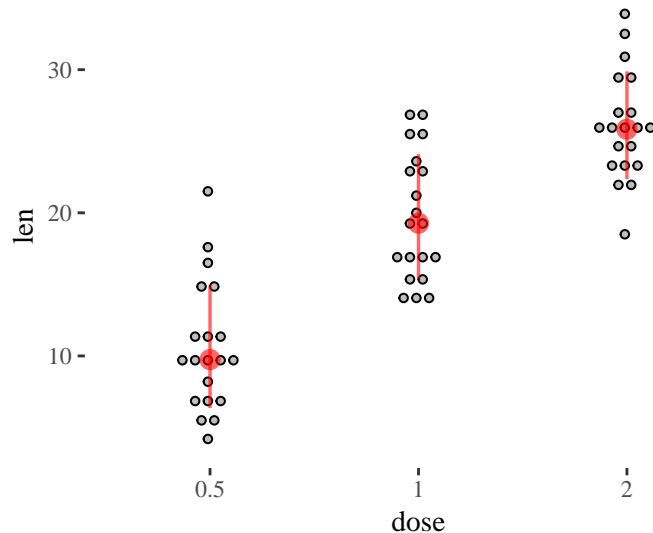
4. Kui andmeid on nii palju, et üksikute andmepunktide vaatlemine paneb pea valutama, siis järgmine informatiivsuse tase on histogramm.
5. kui tahame kõrvuti vaadata paljude erinevate muutujate varieeruvust ja keskväärtusi, siis on head valikud joyplot, violin plot, ja vähem hea valik (sest ta kaotab andmetest rohkem infot) on boxplot. Kui meil on vaid 2-4 jaotust, mida võrrelda, siis saab mängida histogramme facetisse või üksteise otsa pannes (vt ptk graphics).
6. Tulpdiagramm on hea valik siis, kui tahate kõrvuti näidata proportsioonide erinevust. Näiteks, kui meil on 3 liiki kalu, millest igas on erinevas proportsioonis parasiidid, võime joonistada 3 tulpa, millest igas on näidatud ühe kalaliigi parasiitide omavaheline proportsioon.
7. Tulpdiagramm on hädaga pooleks kasutuskõlblik, kui iga muutuja kohta on vaid üks number, mida plottida. Kuigi, siin on meil parem võimalus — cleveland plot. Me ei õpeta tulpade joonistamist olukorras, kus te tahate plottida valimi keskväärtust ja usalduspiire või varieeruvusnäitajat (sd, mad), sest selle jaoks on olemas paremad meetodid. Samas, ehki tulpdiagrammide kasutamine teaduskirjanduses on pikas langustrendis, kasutatakse neid ikkagi liiga palju just eelpoolmainitud viisil.
8. Ära piirdu muutuja tasemel varieeruvuse plottimisega. Teaduslikult on sageli huvitavam mitme muutuja koosvarieerumine. Järgmistes peatükkides modelleerime seda formaalselt regresioonanalüüsis aga alati tasub alustada lihtsatest plottidest. Scatterplot on lihtne viis kovarieeruvuse vaatamiseks.
9. Kui erinevad muutujad on mõõdetud erinevates skaalades (ühikutes), siis võib nende koosvarieeruvust olla kergem võrrelda, kui nad eelnevalt normaliseerida (kõigi muutujate keskväärtus = 0, aga varieeruvus jääb algsesse skaalasse), või standardiseerida (kõik keskväärtused = 0-ga ja sd-d = 1-ga). Standardiseerida tohib ainult normaaljaotusega muutujaid (seega võib olla vajalik muutuja kõigepealt logaritmidada). normaliseerimine: arvuta igale valimi väärtusele: $mean(x) - x$; standardiseerimine: $(mean(x) - x)/sd(x)$.
10. Visualiseeringu valik sõltub valimi suurusest. Väikse valimi korral ($N < 10$) boxploti, histogrammi vms kasutamine on lihtsalt rumal. Ära mängi lolli ja ploti parem punkti kaupa.
 - $N < 20$ - ploti iga andmepunkt eraldi (stripchart(), plot()) ja keskmine või mediaan.
 - $20 > N > 100$: geom_dotplot() histogrammi vaates
 - $N > 100$: geom_histogram(), geom_density() — nende abil saab ka 2 kuni 6 jaotust võrrelda
 - Mitme jaotuse kõrvuti vaatamiseks, kui $N > 15$: geom_boxplot(), or geom_violin(), geom_joy()
11. Nii saab plottida multiplikatiivse sd:

```
# Function to produce summary statistics (geometric mean and multiplicative sd)
multi_sd <- function(x) {
  x <- na.omit(x)
  a <- log10(x)
  b <- mean(a)
  c <- sd(a)
  g_mean <- 10**b
  msd <- 10**c
  ymin <- g_mean/msd
  ymax <- g_mean * msd
  return(c(y = g_mean, ymin = ymin, ymax = ymax))
}

ToothGrowth <- ToothGrowth
ToothGrowth$dose <- as.factor(ToothGrowth$dose)
```

```
ggplot(ToothGrowth, aes(x=dose, y=len)) +
  geom_dotplot(binaxis='y', stackdir='center',
               stackratio=1.5, dotsize=0.6, fill="grey") +
  stat_summary(fun.data=multi_sd, color="red", size=0.6, alpha=0.6) +
  theme_tufte()

## `stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.
```



Kokkuvõte:

- Andmepunktide plottimine säilitab maksimaalselt andmetes olevat infot (nii kasulikku infot kui müra). Aitab leida outliereid (valesti sisestatud andmeid, valesti mõõdetud proove jms). Kui valim on väiksem kui 20, piisab täiesti üksikute andmepunktide plotist koos mediaaniga. Dot-plot ruulib.
- Histogramm – kõigepealt mõõtskaala ja seejärel andmed jagatakse võrdse laiusega binnidesse ja plotitakse binnide kõrgused. Bin, kuhu läks 20 andmepunkti on 2X kõrgem kui bin, kuhu läks 10 andmepunkti. Samas, bini laius/ulatus mõõteskaalal pole teile ette antud – ja sellest võib sõltuda histogrammi kuju. Seega on soovitatav proovida erinevaid bini laiusi ja võrrelda saadud histograme. Histogramm sisaldab vähem infot kui dot plot, aga võimaldab paremini tabada seaduspärasid & andmejaotust & outliereid suurte andmekoguste korral.
- Density plot. Silutud versioon histogrammist, mis kaotab infot aga toob vahest välja signaali müra arvel. Density plotte on hea kõrvuti vaadelda joy ploti abil.
- Box-plot – sisaldab vähem infot kui histogramm, kuid neid on lihtsam kõrvuti võrrelda. Levinuim variant (kuid kahjuks mitte ainus) on Tukey box-plot – mediaan (joon), 50% IQR (box) ja 1,5x IQR (vuntsid), pluss outlierid eraldi punktidenä.
- Violin plot – informatiivsusest box-ploti ja histogrammi vahepeal – sobib paljude jaotuste kõrvuti võrdlemiseks
- Line plot – kasuta ainult siis kui nii X kui Y teljele on kantud pidev väärtus (pikkus, kaal, kontsentratsioon, aeg jms). Ära kasuta, kui teljele kantud punktide vahel ei ole looduses mõtet omavaid pidevaid väärtusi (näiteks X teljel on katse ja kontroll või erinevad valgumutatsioonid, mille aktiivsust on mõõdetud)
- Suhete võrdlemine (pie vs bar)

- h. Cleveland plot on hea countide jaoks. Kasuta Barplotti ainult siis, kui Cleveland plot vm plot mingil põhjusel ei sobi.
- i. Pie chart on proportsioonide vaatamiseks enam-vähem kõlblik ainult siis, kui teil pole vaja võrrelda proportsioone erinevates objektides. Kõik graafikud, kus lugeja peab võrdlema pindalasid, on inimõistusele petlikud — lugeja alahindab süstemaatiliselt erinevuste suurus! Selle pärast on proportsioonide võrdlemiseks palju parem tulpdiagramm, kus võrreldavad tulbad on ühekõrgused, et proportsioonide erinevused iga tulba sees paremini tulpade vahel võrreldavad oleks.

Informatsiooni hulk kahanevalt: iga andmepunkt plotitud —> histogramm —> density plot & violin plot —> box plot —> tulpdiagramm standardhälvetega —> cleveland plot (ilma veapiirideta)

Jäta meelde:

1. Statistika jagatakse kolme ossa: kirjeldav (summary), uuriv (exploratory) ja järeldav (inferential).
2. Kirjeldav statistika kirjeldab teie andmeid summaarsete statistikute abil.
3. uuriv statistika püstitab valimi põhjal uusi teaduslikke hüpoteese, kasutades selleks põhiliselt graafilisi meetodeid
4. Järeldav statistika kasutab formaalseid mudeleid, et kontrollida uuriva statistika abil püstitatud hüpoteese. Järeldav statistika teeb valimi põhjal järeldusi statistilise populatsiooni kohta, millest see valim pärineb.
5. Need järeldused on alati ebakindlad; ka siis kui need esitatakse punkthinnanguna parameetriväärtusele. Nii punkthinnangud kui intervall-hinnangud on lihtsustused: tegelik ebakindluse määr on n-dimensionaalne tõenäosuspilv, kus n on mudeli parameetrite arv.
6. Statistika põhiline ülesanne on kvantifitseerida ebakindlust, mis ümbritseb järeldava statistika abil saadud hinnanguid. Selle ebakindluse numbriline mõõt on tõenäosus, mis jääb 0 ja 1 vahele.
7. tõenäosus omistab numbrilise väärtuse sellele, kui palju me usuksime hüpoteesi x kehtimisse, juhul kui me usuksime, et selle tõenäosuse arvutamiseks kasutatud statistilised mudelid vastavad tegelikkusele.
8. ükski statistiline mudel ei vasta tegelikkusele.