Sissejuhatus R’I, osa 4: II kodutööks vajalik

# Sagedustabel

**Hii-ruut test**

# sagedustabeli loomine  
freqtab <- matrix(c(59, 155, 103, 28), ncol = 2)   
# hii-ruut test   
chisq.test(freqtab)

Sagedustabel tuleb sisestada maatriksina. Lahtrite sagedused (sama näide mis loengus) saab sisestada ühe joruna niiviisi:

freqtab <- matrix(c(59, 155, 103, 28), # maatriksi elemendid ehk sagedused   
 ncol = 2) # tulpade arv   
freqtab

## [,1] [,2]  
## [1,] 59 103  
## [2,] 155 28

Pane tähele, et vaikimisi loetakse arvud sisse tulpade (mitte ridade) kaupa ja allpool on näha, et nii tulpadele kui ridadele saab ka nimed panna.

colnames(freqtab) <- c("females", "males")  
rownames(freqtab) <- c("black", "white")  
freqtab

## females males  
## black 59 103  
## white 155 28

Äärejaotused (ridade ja tulpade summad) saab nii:

colSums(freqtab)

## females males   
## 214 131

rowSums(freqtab)

## black white   
## 162 183

Sagedustabelist saab arvutada igatpidi protsente

prop.table(freqtab) # lahtrite sagedused %-na vaatluste koguarvust

## females males  
## black 0.1710145 0.29855072  
## white 0.4492754 0.08115942

# nt 17% on musti emaseid  
prop.table(freqtab, 1) # % ridade kaupa

## females males  
## black 0.3641975 0.6358025  
## white 0.8469945 0.1530055

# nt 36% mustadest on emased   
prop.table(freqtab, 2) # %’d tulpade kaupa

## females males  
## black 0.2757009 0.7862595  
## white 0.7242991 0.2137405

# e.g. 27% emastest on mustad

**Hii-ruut-test**

Kui funktsiooni chisq.test rakendada maatriksile, siis ta testib assotsiatsiooni asjaomaste muutujate vahel

chisq.test(freqtab, correct = F)

##   
## Pearson's Chi-squared test  
##   
## data: freqtab  
## X-squared = 85.041, df = 1, p-value < 2.2e-16

Suvand correct = F pärsib nn. pidevuse paranduse (*continuity correction*) ehk teeb hii-ruut-testi oma algkujul (nii saame täpselt samad tulemused nagu konspektis) aga üldjuhul pole seda vaja pärssida.

Oodatavad sagedused saab järgmisel viisil:

test.result <- chisq.test(freqtab, correct = F)  
test.result$expected

## females males  
## black 100.487 61.51304  
## white 113.513 69.48696

Pane tähele, et testi tulemused saab salvestada eraldi objektiks, siis ei kao nad ära.

## 

# Korrelatsioonanalüüs

cor.test(weird.people$height, weird.people$weight, method = "pearson")

Alustuseks teeme andmestiku:.

weird.people <- data.frame(  
 weight = c(76.0,77.5,77.0,83.0,69.7,76.0,79.5,73.0,71.4,69.1),  
 height = c(189.4,177.3,177.2,196.3,181.9,188.4,189.0,179.5,166.7,166.9),  
 age = c(11,12,13,14,15,16,17,18,19,20),  
 bmi = c(21.2,24.7,24.5,21.5,21.1,21.4,22.2,22.7,25.7,24.8)  
)

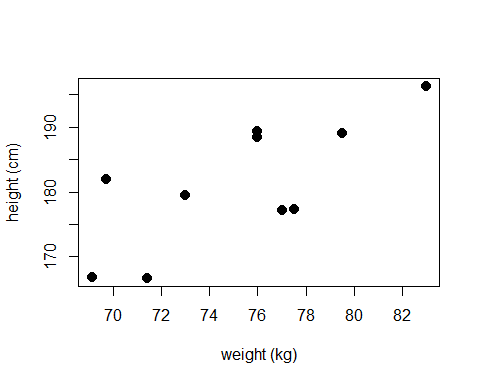
Korrelatsioonikoefitsiendid saab arvutada käsuga cor:.

cor(weird.people$weight, weird.people$height)

## [1] 0.7460451

Asjaomase pildi saab sedamoodi:

plot(weird.people$weight, weird.people$height, # põhiosa  
 xlab = "weight (kg)", ylab = "height (cm)", # telgede tekstid  
 pch = 20, cex = 2) # punktide väljanägemine



Vaikimisi arvutatakse Pearsoni korrelatsioonikordajad. Testi tulemused saab käsuga cor.test:

cor.test(weird.people$weight, weird.people$height)

##   
## Pearson's product-moment correlation  
##   
## data: weird.people$weight and weird.people$height  
## t = 3.1689, df = 8, p-value = 0.01322  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.219546 0.936003  
## sample estimates:  
## cor   
## 0.7460451

Nagu R puhul sageli juhtub, esituvad ka korrelatsioonikordaja 95% usalduspiirid. Näeme, et kaalu ja pikkuse vahel korrelatsioon on (r=0.75, p=0.013).

# Lineaarsed mudelid (regression, ANOVA….)

**Regressioonanalüüs kahe muutujaga, I ja III tüüpi analüüsid**

cor(weird.people$weight, weird.people$height)  
plot(weird.people$weight, weird.people$height, pch = 20,  
 xlab = "Weight", ylab = "Height", main = "Height vs weight")  
cor(weird.people$weight, weird.people$age)  
plot(weird.people$weight, weird.people$age)  
m1 <- lm(weight ~ height + age, data = weird.people)  
m2 <- lm(weight ~ age + height, data = weird.people)  
anova(m1) # I tüüpi ANOVA  
anova(m2)  
drop1(m1, test = "F") # III tüüpi ANOVA  
drop1(m2, test = "F")

**Kahefaktoriline ANOVA ja jääkide kuvamine**

m1 <- lm(weight ~ color + sex, data = anova.data)  
m2 <- lm(weight ~ sex + color, data = anova.data)  
anova(m1) # I tüüpi ANOVA  
anova(m2)  
resids <- residuals(m1)  
hist(resids[anova.data$sex == "male"],  
 xlab = "", main = "Histogram of residuals\n(male)")  
hist(resids[anova.data$sex == "female"],   
 xlab = "", main = "Histogram of residuals\n(female)")

**Mittelineaarsuse test**

m <- lm(weight ~ age + I(age\*\*2), data = weird.people)  
anova(m) # I tüüpi ANOVA: ruutliige peab olema viimasena   
plot(weight ~ age, data = weird.people, pch=20)  
x0 <- seq(0,20,length=100)  
y0 <- predict(m, newdata = data.frame(age=x0))  
lines(x0,y0, col="blue")  
coef(m) # weight = 41.7 + 5.3\*age - 0.20\*age\*\*2

**Koosmõjude visualiseerimine**

interaction.plot(anova.data$color, anova.data$sex, anova.data$weight,  
 ylab = "color", xlab = "mean weight")

ANOVA, ANCOVA, lineaarne regressioon ja mitmene regressioon on kõik üldise lineaarse mudeli varjundid ja R-s saab neid kõiki teha funktsiooniga lm (**l**inear **m**odel).

## Mitmene regressioon

Teeme hakatuseks lihtsa kolme sõltumatu muutujaga regressiooni ja salvestame tulemused objektiks m1

m1 <- lm(weight ~ height + age + bmi, data = weird.people)

Funktsiooni lm esimene argument on mudel ise (tema valem), st sõltuv muutuja vasakul pool ~ märki ja sõltumatud muutujad paremal pool ~märki, viimased pussmärkidega eraldatud. Kuna kõik need muutujad pole niisama, vaid kuuluvad kuskile *data frame*sse, siis tuleb funktsiooni viimase argumendina see *data frame* ära nimetada).

Käsk summary(m1) kuvab regressioonanalüüsi põhitulemused.

##   
## Call:  
## lm(formula = weight ~ height + age + bmi, data = weird.people)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.39684 -0.13500 0.06283 0.14680 0.28409   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -154.39308 6.17331 -25.010 2.69e-07 \*\*\*  
## height 0.85028 0.02073 41.017 1.40e-08 \*\*\*  
## age 0.01285 0.03617 0.355 0.734   
## bmi 3.27643 0.10354 31.643 6.62e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2694 on 6 degrees of freedom  
## Multiple R-squared: 0.9975, Adjusted R-squared: 0.9963   
## F-statistic: 806.1 on 3 and 6 DF, p-value: 3.314e-08

Call:tuletab meelde, mis analüüsi parasjagu tehti. Residuals: annab teavet jääkide jaotuse kohta (miinimum, 25% kvantiil, mediaan, ülemine kvartiil, maksimum). Kolmas osa nimega Coefficients: on kõige tähtsam ja näitab meile kätte kõik mudeli parameetrid ja testib iga sõltumatu muutujaga seotud tõusude erinevust nullist. Ülalesitatud tulemuste põhjal saame kirjutada järgmise regressioonvõrrandi.

weight = -154.39 + 0.85 \* height + 0.01 \* age + 3.28 \* bmi

*Coefficients* tabelist näeme ka seda, et muutujate height ja BMI seos kaaluga on statistiliselt oluline (p<0.0001 mõlema puhul), aga age ei mõjuta kaalu oluliselt (p=0.7). Muideks siis 1.40e-08 tähendab 1.40 korda 10 astmel -8 ehk siis

1.40e-08 = 0.0000000140

Viimane osa sisaldab mudeli kui terviku sobivuse näitajaid nagu R-ruudu väärtus ja F-testi tulemus kogu mudeli kohta (kas sõltumatutel muutujatel kamba peale üldse mingit mõju on).

## Mittelineaarsuse test – ruutliige mudelis

Mudeli valemisse saab sisse kirjutada matemaatilisi funktsioone kasutades funktsiooni I. Nii näiteks uurimaks seda, kas seos vanuse ja kaalu vahel on mittelineaarne (kõverjooneline) saab mudelisse kaasata vanuse ruudu age\*\*2 funktsiooni I() argumendina ja siis teha I tüüpi testi.

m2 <- lm(weight ~ age + I(age\*\*2), data = weird.people)  
anova(m2) # Type I test

## Analysis of Variance Table  
##   
## Response: weight  
## Df Sum Sq Mean Sq F value Pr(>F)  
## age 1 50.427 50.427 3.3605 0.1094  
## I(age^2) 1 20.406 20.406 1.3599 0.2818  
## Residuals 7 105.043 15.006

Meil pole alust väitmaks, et see seos on mittelineaarne (p=0.2818).

Kui see ruutu tõstetud liige ei ole esitatud I( ) abil, siis ei saa R meist aru või teeb seda väga valesti. Muidugi on alati võimalik eraldi defineerida uus muutuja age-ruut ja kaasata see sellisena mudelisse.

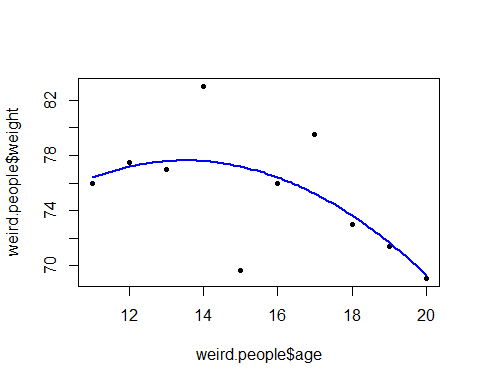
weird.people$age.squared <- weird.people$age \*\* 2  
m2a <- lm(weight ~ age + age.squared, data = weird.people)  
anova(m2a)

## Analysis of Variance Table  
##   
## Response: weight  
## Df Sum Sq Mean Sq F value Pr(>F)  
## age 1 50.427 50.427 3.3605 0.1094  
## age.squared 1 20.406 20.406 1.3599 0.2818  
## Residuals 7 105.043 15.006

Kõverjoone saamine pildile on paraku keerulisem. Selleks tuleb arvutada mudeli poolt ennustatud väärtused paljude tihedalt asetsevate x-väärtuste tarbeks ja ühendada nad sellistena joonisel. Vat nii:

# Kõigepealt laseme joonisele panna päris andmepunktid   
plot(weird.people$age, weird.people$weight, pch = 20)  
# Järgmiseks genereerime hulga x-väärtusi kõverjoone joonistamiseks

x0 <- seq( # jada  
 min(weird.people$age), # vähimast age väärtusest  
 max(weird.people$age), # suurima age väärtuseni  
 length = 100) # sada tükki  
# leia mudeli ennustatud väärtused  
y0 <- predict( # ennusta  
 m2, # mudelist m2  
 newdata = data.frame( # kasutades uusi andmeid (tulpade nimed peavad klappima!)  
 age = x0) # algsete age väärtuste asemel kasutame jada  
 )   
# Lõpuks kanname saadud joone pildile   
lines(x0, y0,   
 lwd = 2, col = "blue") # määrab joone värvi



Mudeli parameetrid saab niimoodi:

coef(m2)

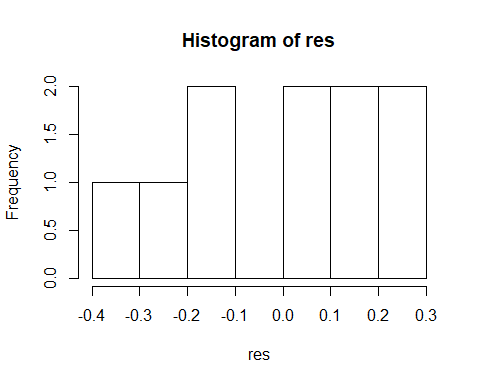
## (Intercept) age I(age^2)   
## 41.7290909 5.3125000 -0.1965909

… ja nende põhjal siis saame joone võrrandi kirja panna: .

## Jäägid

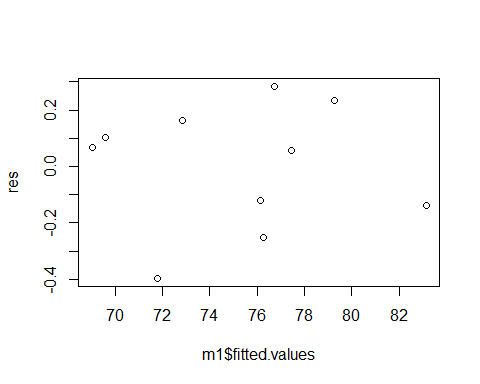
Lineaarsete regressioonmudelite eelduseks on jääkide normaaljaotus. Siinkohal piirdume jääkide visuaalse vaatlemisega:

res <- residuals(m1) # leia jäägid  
hist(res) # jääkide histogramm



Asjalik on ka vaadata pildilt jääkide sõltuvust pidevatest sõltumatute muutujatest veendumaks, et meil ei ole probleeme regressioonanalüüsi eelduste täidetusega:

plot(m1$fitted.values, res)



Aga siin näites on andmeid mõistagi liiga vähe millegi mõistliku järeldamiseks.

## ANOVA

Kategooriliste sõltumatute tunnustega ANOVA saab R-i jaoks kirja panna täpselt samamoodi nagu mitmese regressiooni. Koosmõju tähistatakse tärniga (\*) faktorite vahel. Kirjapilt  sex \* color tähistab mudelit sex + color + (sex ja color koosmõju).

# loeme andmed sisse  
anova.data <- read.table("anova-data.csv", header = T, sep = ";", dec = ",")  
anova.data # vaatame andmeid

## color sex weight  
## 1 red male 10.1  
## 2 red male 9.4  
## 3 red male 10.0  
## 4 red male 10.4  
## 5 red female 13.0  
## 6 red female 13.1  
## 7 red female 13.1  
## 8 red female 12.9  
## 9 green male 11.0  
## 10 green male 11.4  
## 11 green male 10.8  
## 12 green male 11.2  
## 13 green female 14.0  
## 14 green female 14.1  
## 15 green female 14.2  
## 16 green female 14.3  
## 17 blue male 12.0  
## 18 blue male 11.9  
## 19 blue male 12.0  
## 20 blue male 12.0  
## 21 blue female 15.5  
## 22 blue female 15.0  
## 23 blue female 15.0  
## 24 blue female 15.0

# viime analüüsi läbi   
m2 <- lm(weight ~ sex \* color, data = anova.data)

ANOVA tabeli saab kätte funktsiooniga anova. Koosmõju tähistuseks on koolon :.

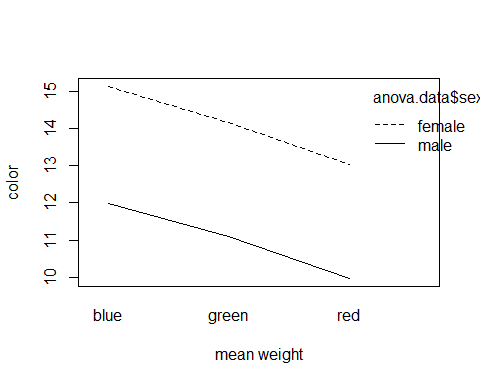
anova(m2)

## Analysis of Variance Table  
##   
## Response: weight  
## Df Sum Sq Mean Sq F value Pr(>F)   
## sex 1 57.042 57.042 1026.75 < 2.2e-16 \*\*\*  
## color 2 16.863 8.432 151.77 5.399e-12 \*\*\*  
## sex:color 2 0.013 0.007 0.12 0.8876   
## Residuals 18 1.000 0.056   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Näeme, et nii sex kui color mõjutavad muutujat weight, kuid color mõju ei erine sooti (p=0.9).

Kahe muutuja koosmõju või selle puudumist saab visualiseerida asjakohase joonisega, mis näitab rühmade keskmisi algandmetes.

interaction.plot(anova.data$color, anova.data$sex, anova.data$weight,  
 ylab = "color", xlab = "mean weight")



## Esimest ja kolmandat tüüpi analüüsid

Vaikimisi esitab funktsioon anova I tüüpi ruutude summasid ja neile tugineva testi tulemusi, III tüüpi testi tulemused saab kätte funktsiooniga drop1.

drop1(m2, test = "F") # III tüüpi test

## Single term deletions  
##   
## Model:  
## weight ~ sex \* color  
## Df Sum of Sq RSS AIC F value Pr(>F)  
## <none> 1.0000 -64.273   
## sex:color 2 0.013333 1.0133 -67.955 0.12 0.8876

Nagu ülal näha, saame testi tulemused vaid koosmõju kohta. Nüüd jätame koosmõju välja ja saame testi tulemused peamõjude kohta.

m2.1 <- lm(weight ~ sex + color, data = anova.data)  
drop1(m2.1, test = "F")

## Single term deletions  
##   
## Model:  
## weight ~ sex + color  
## Df Sum of Sq RSS AIC F value Pr(>F)   
## <none> 1.013 -67.955   
## sex 1 57.042 58.055 27.200 1125.82 < 2.2e-16 \*\*\*  
## color 2 16.863 17.877 -3.069 166.41 3.425e-13 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Juhusliku faktoriga mudelid

**Kaks juhuslikku faktorit**

m1 <- lmer(para ~ trea + (1|nr) + (1|time))  
anova(m1) # ANOVA fikseeritud muutujaile  
summary(m1) # juhuslike muutujate dispersioonid, koefitsiendid fikseeritud muutujaile

**Korduvmõõtmistega ANOVA**

# naiivne ANOVA, mis ei võta arvesse asjaolu, et isendeid on korduvalt mõõdetud

m1 <- lm(para ~ trea \* time)  
anova(m1)

# korduvmõõtmised: m3 <- lme(para ~ trea\*time, random = ~1|nr, correlation = corAR1(,form = ~time|nr))  
anova(m3)

Siin kasutame kahte võimalust, mis on üldiselt üksteisele väga sarnased: paketi lme4 funktsioon lmer, ja paketi nlme funktsioon lme. Mõlemaid neid asju saab kasutada pea samamoodi nagu tavalist lm funktsiooni ja lihtsates olukordades on lmer ja lme tulemused pea samased. Siiski on rohkem kui ühe juhusliku faktori kasutamine lihtne lme4 paketis, kuid üllatavalt keeruline nlme’s. Teisalt võimaldab nlme kaasata mitmesuguseid autokorrelatsioonistruktuure (loeng 10!), mida lme4-s vaid suure hädaga teha saab.

Kõigepealt tuleb need paketid kasutusele võtta.

#install.packages("nlme") # ühe korra igas arvutis  
library(nlme) # iga kord peale arvuti käivitamist   
#install.packages("lme4") # ühe korra igas arvutis

#install.packages("lmerTest") # ühe korra igas arvutis

library(lme4) # iga kord peale arvuti käivitamist

## Warning: package 'lme4' was built under R version 3.5.3

## Loading required package: Matrix

##   
## Attaching package: 'lme4'

## The following object is masked from 'package:nlme':  
##   
## lmList

Miskipärast ei anna lme4 vaikimisi p-väärtusi, nood tekivad siis, kui lmerTest on aktiveeritud, mida siis tulebki teha

library(lmerTest) # iga kord peale arvuti käivitamist. Peab tegema selles järjekorras: enne lme4 ja siis lmerTest!

## Warning: package 'lmerTest' was built under R version 3.5.3

## ## Attaching package: 'lmerTest'

## The following object is masked from 'package:lme4': ## ## lmer

## The following object is masked from 'package:stats': ## ## step

In addition, we use the package lmerTest to get p-values into the output of lme4 models.

## Fikseeritud ja juhuslikud faktorid

Vaatame algul mänguandmestikku, kus uuritakse kahe eri toidu (feed) peal kasvanud linnupoegade kaalu, linnupojad on pärit mitmest eri pesakonnast (brood).

# teeme andmestiku (ei eelda süvenemist kodutöö tarbeks)   
feeding.data <- data.frame(  
 brood = as.factor(rep(letters[1:5],each=6)),  
 feed = rep(as.factor(rep(letters[1:2],each=3)),5),  
 weight = c(5,5,6,11,12,11,3,4,5,7,8,9,5,7,6,8,10,12,9,9,9,11,12,12,12,11,12,17,16,14)  
)  
# viime analüüsi läbi   
m3 <- lmer(weight ~ feed + (1|brood), data = feeding.data)

Mudelis on nii fikseeritud (feed) kui ka juhuslik (brood) muutuja. Juhusliku faktori tähistus on (1|brood), st muutuja nimi (brood) tähistuse 1| järel ja kogu asi sulgudes. Kui juhuslikke faktoreid on rohkem kui üks, tuleb nad panna igaüks eraldi selliste sulgude sisse weight ~ feed + (1|brood) + (1|randfact2).

summary(m3)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: weight ~ feed + (1 | brood)  
## Data: feeding.data  
##   
## REML criterion at convergence: 105.8  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.88060 -0.75864 0.00965 0.68095 1.70507   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## brood (Intercept) 8.203 2.864   
## Residual 1.244 1.116   
## Number of obs: 30, groups: brood, 5  
##   
## Fixed effects:  
## Estimate Std. Error df t value Pr(>|t|)   
## (Intercept) 7.2000 1.3129 4.1997 5.484 0.00467 \*\*   
## feedb 4.1333 0.4073 23.9999 10.147 3.69e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr)  
## feedb -0.155

Summary käsuga saadav väljund on üsna sarnane sellisele lm-s, oluline lisa on mõistagi Random effects osakond. Juhuslike faktorite kohta esitatakse standardhälbed ja dispersioonid

Fixed effects table on väga sarnane tabelile lm summary väljundis.

Fikseeritud faktorite p-väärtused saab ka tavalise funktsiooniga anova:

anova(m3)

## Type III Analysis of Variance Table with Satterthwaite's method  
## Sum Sq Mean Sq NumDF DenDF F value Pr(>F)   
## feed 128.13 128.13 1 24 102.96 3.693e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Sama analüüsi saab lihtsasti teha funktsiooniga lme:

m3a <- lme(weight ~ feed, random = ~ 1|brood, data = feeding.data)  
summary(m3a)

## Linear mixed-effects model fit by REML  
## Data: feeding.data   
## AIC BIC logLik  
## 113.8105 119.1393 -52.90523  
##   
## Random effects:  
## Formula: ~1 | brood  
## (Intercept) Residual  
## StdDev: 2.864211 1.115547  
##   
## Fixed effects: weight ~ feed   
## Value Std.Error DF t-value p-value  
## (Intercept) 7.200000 1.3128990 24 5.484047 0  
## feedb 4.133333 0.4073401 24 10.147132 0  
## Correlation:   
## (Intr)  
## feedb -0.155  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -1.880603535 -0.758642545 0.009657616 0.680954001 1.705082293   
##   
## Number of Observations: 30  
## Number of Groups: 5

## Korduvmõõtmised

Uuriti toidu (kaks eri taime, a ja b) mõju parasiitide ohtrusele lindudel. Igat lindu mõõdeti 4 korda. Mõistlik on eeldada, et ajas lähestikku olevad mõõtmise on sarnasemad (autokorrelatsioon ajas, vt aegrea teemal loengus), mispuhul peaks ka analüüs va autokorrelatsiooni arvestama. Kuna meid huvitab parasiteerituse muutus ajas (ja selle muutuse sõltuvus toidust), siis on analüüsi kaasatud ka aeg (time) pideva muutujana.

Teeme andmestiku (pole oluline kodutöö kontekstis)  
nr <- rep(1:6, each=4) # isendi number  
trea <- factor(rep(c("a","b"),each=12)) # treatment  
time <- rep(1:4,6) # mõõtmise aeg (igaühte mõõdetud 4 korda)   
para <- c(1,2,3,4,2,2,3,4,2,3,4,5,1,4,4,4,2,5,5,4,2,6,5,5) # sõltuv muutuja  
# analüüsi läbiviimine  
m16 <- lme(para ~ trea \* time, # aeg on siin pidev ja fikseeritud random = ~1|nr, # isend (tema number) juhusliku faktorina  
 correlation = corAR1(,form = ~time|nr)) # arvestab autokorrelatsiooni, st ajas lähedal asuvate punktide sarnasust  
anova(m16)

## numDF denDF F-value p-value  
## (Intercept) 1 16 210.15020 <.0001  
## trea 1 4 4.48672 0.1015  
## time 1 16 24.54783 0.0001  
## trea:time 1 16 0.15536 0.6987

Järeldame, et toidu mõju ei muutu ajas (p=0.7). Parasiteerituse indeks ei sõltu üldse toidust (p=0.10). Parasiteeritus siiski muutub ajas (p=0.0001).

# Logistiline regression

m <- glm(y ~ x1, family = binomial)  
drop1(m, test="LR")

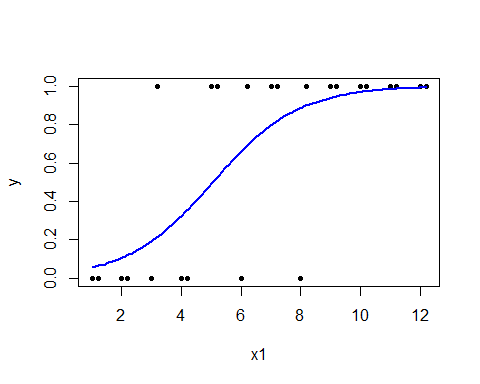
## Single term deletions  
##   
## Model:  
## y ~ x1  
## Df Deviance AIC LRT Pr(>Chi)   
## <none> 17.355 21.355   
## x1 1 31.755 33.755 14.4 0.0001478 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

coef(m)

## (Intercept) x1   
## -3.5242965 0.6985107

Logistilise kõvera võrrand on

plot(x1, y, pch=20)  
p <- function(x1) {  
 exp(-3.5243+0.6985\*x1) / (1 + exp(-3.5243+0.6985\*x1))  
}  
curve(p, add = T, col = "blue", lwd = 2)

 \*\*\*

Logistiline regressioon, Poissoni jaotust eeldav regression ja paljud teised sellised asjad koonduvad mõiste **g**eneralized **l**inear **m**odels alla, sellest siis ka asjakohase funktsiooni nimi R-s: glm.

Funktsiooni glm kasutamine ei erine suurt funktsiooni lm kasutamisest. Siiski kuna glms-s on võimalik teha palju erinevaid analüüse, tuleb konkreetse analüüsi (sõltuva muutuja jaotuse) tüüp määrata suvandiga *family.* Logistilise regressiooni puhul tuleb kirjutada binomial.

x1 <- c(c(1:12),c(1:12)+0.2) # leiutame andmed   
x2 <- factor(rep(c("a","b"), each=12))  
y <- c(0,0,0,0,1,0,1,0,1,1,1,1,0,0,1,0,1,1,1,1,1,1,1,1)  
# mudel   
m5 <- glm(y ~ x1, # mudeli valem   
 family = binomial) # sõltuva muutuja jaotuse määramine   
drop1(m5, test = "LRT") # kolmandat tüüpi analüüs

## Single term deletions  
##   
## Model:  
## y ~ x1  
## Df Deviance AIC LRT Pr(>Chi)   
## <none> 17.355 21.355   
## x1 1 31.755 33.755 14.4 0.0001478 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Ülalolevast tabelist näeme testi tulemusi (p-väärtust).

Parameetrite väärtused saab summary-käsuga.

summary(m5) # Parameter estimates

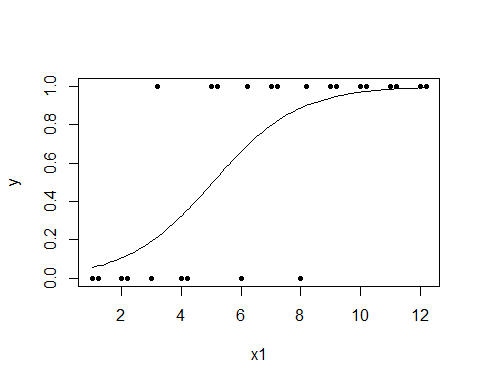
##   
## Call:  
## glm(formula = y ~ x1, family = binomial)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.0896 -0.4826 0.1702 0.5016 1.7507   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.5243 1.5574 -2.263 0.0236 \*  
## x1 0.6985 0.2727 2.561 0.0104 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 31.755 on 23 degrees of freedom  
## Residual deviance: 17.355 on 22 degrees of freedom  
## AIC: 21.355  
##   
## Number of Fisher Scoring iterations: 6

See näeb välja üsna samamoodi nagu tavalise regressiooni puhul, aga parameetrite (k ja b) tõlgendused ei ole nii ilmsed, vaata konspekti sesasjas.

Siinnäites on sõltuva muutuja väärtus (selle tõenäosus, et ) leitav järgmise võrrandiga

Selle kõvera saamiseks pildi peale tuleb teha nii:

plot(x1,y, pch=20)  
# defineerime seose kui R funktsiooni   
p <- function(x1) {  
 exp(-3.5243+0.6985\*x1) / (1 + exp(-3.5243+0.6985\*x1))  
}  
curve(p, add = T)



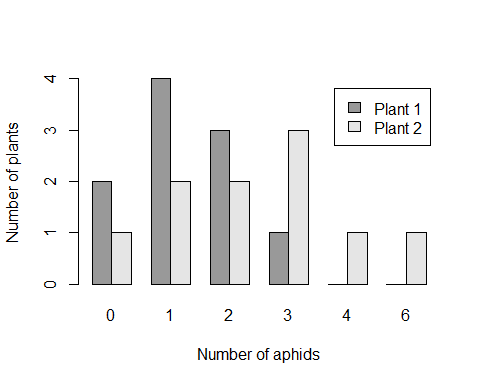
# Poissoni jaotuse eeldusega regressioon

m <- glm(aphids ~ plant, family = poisson)  
drop1(m, test = "LRT")

# loome andmestiku   
plant <- factor(rep(c("Plant 1","Plant 2"), each=10))  
aphids <- c(0,0,1,1,1,1,2,2,2,3,0,1,1,2,2,3,3,3,4,6)

Uurime asja pildi pealt:

barplot(table(plant, aphids), beside = T, # defineerib andmestiku   
 xlab = "Number of aphids", ylab = "Number of plants", # telgede kirjad  
 legend.text = T, col = c("gray60", "gray90")) # kujundus



Poissoni regressioon on üks üldistatud lineaarse mudeli erijuht. Seega saame jälle kasutada glm’i, ainult et me peame ütlema, et seekord tahame kasutada Poissoni jaotuse eeldusega mudelit).

m6 <- glm(aphids ~ plant, family = poisson)  
drop1(m6, test = "LRT")

## Single term deletions  
##   
## Model:  
## aphids ~ plant  
## Df Deviance AIC LRT Pr(>Chi)   
## <none> 20.185 68.144   
## plant 1 24.040 69.999 3.8551 0.04959 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Taime mõju on piiripealselt oluline (tõepära suhte test p=0.04959), seetõttu võime öelda, et taimeliikide vahel on erinevusi lehetäide hulgas osas. Erinevuse suurust näitavad mudeli parameetrid:

summary(m6)

##   
## Call:  
## glm(formula = aphids ~ plant, family = poisson)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2361 -0.5159 -0.2744 0.5685 1.8723   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.2624 0.2774 0.946 0.3442   
## plantPlant 2 0.6539 0.3419 1.912 0.0558 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 24.040 on 19 degrees of freedom  
## Residual deviance: 20.185 on 18 degrees of freedom  
## AIC: 68.144  
##   
## Number of Fisher Scoring iterations: 5

Näeme, et taimel 2 on täisid korda rohkem kui taimel 1.

# Peakomponentanalüüs

pca <- prcomp(cbind(wear,tear,date,size))  
cbind(wear, tear, date, size, pca$x[,1])  
cor.test(eggs, pca$x[,1])

# teeme andmed  
wear <- c(5,6,8,9,10,8,9,11,13,15)  
tear <- c(4,2,6,7,8,9,9,9,8,12)  
date <- c(1,1,5,5,6,7,6,8,7,9)  
size <- c(3,2,3,2,1,2,3,4,3,2)  
eggs <- c(100,90,85,88,67,50,45,40,30,20)  
# viime analüüsi läbi  
pca <- prcomp(cbind(wear,tear,date,size))

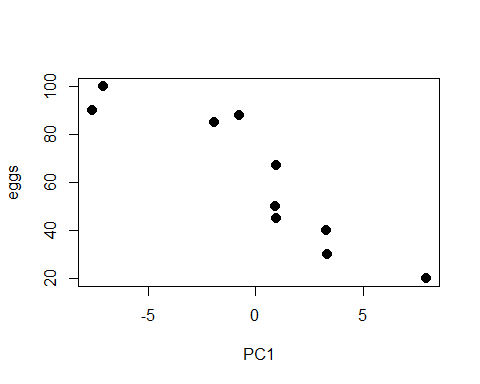
Kõigi *principal component score*’de väärtused kõigi objektide jaoks väljastatakse maatriksina. R arvutab niimitu peakomponenti kui jaksab, aga praktikas on neid harva rohkem vaja kui kahte esimest: (pca$x[,1:2]).

pca$x # Näitab kõiki peakomponente

## PC1 PC2 PC3 PC4  
## [1,] -7.0930552 -0.32834563 0.2410251 0.97090863  
## [2,] -7.6480933 1.58653697 -0.3010269 -0.34489417  
## [3,] -1.9238876 -0.22524609 0.6654823 -0.47439852  
## [4,] -0.7497843 0.07318845 -0.4993887 -0.05547887  
## [5,] 0.9769312 0.07627476 -1.4526854 -0.38664292  
## [6,] 0.9062149 -2.38251488 -0.4892436 -0.53201333  
## [7,] 0.9598520 -1.34966755 0.2520284 0.64308890  
## [8,] 3.2733115 -0.41549006 1.6258143 -0.29921131  
## [9,] 3.3440279 2.04329959 0.6623725 -0.15384090  
## [10,] 7.9544830 0.92196444 -0.7043780 0.63248247

Kui me nüüd tahame analüüsida esimese peakomponendi seost munade arvuga, peame välja võtma esimese tulba pca$x maatriksist:

plot(pca$x[,1], eggs, pch = 20, cex = 2,  
 xlab = "PC1", ylab = "eggs")



cor.test(pca$x[,1], eggs)

##   
## Pearson's product-moment correlation  
##   
## data: pca$x[, 1] and eggs  
## t = -5.9356, df = 8, p-value = 0.0003475  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.9770334 -0.6327942  
## sample estimates:  
## cor   
## -0.9027452

Näeme, et esimene peakomponent on munade arvuga tugevalt korreleerunud (r=-0.90, p=0.0003). Mäletame siinjuures, et peakomponendi suund (positiivsed ja negatiivsed väärtused kumba pidi) on suvaline ja seetõttu ei saa korrelatsiooni märgist otse midagi järeldada (enne tuleb peakomponenti sisuliselt interpreteerida).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* jutu lõpp \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*