**Lisamaterjal „Katseandmete analüüsi“ kursuse juurde: jutuks olevate analüüside realiseerimine R keskkonnas**

**kirjutas Ants Kaasik 2021-2022**

**Eessõna.** Käesolevas tekstis on seletatud, kuidas suuremat osa „Katseandmete analüüsi“ (KA) kursuses käsitletud analüüsidest R-s realiseerida. Alapeatükkide numeratsioon on kooskõlas KA kursuse teemade numeratsiooniga. Lisaks R skriptidele leiab huviline lugeja tekstist ka kommentaare, mis laiendavad ja süvendavad KA kursuses käsitletud küsimuste mõistmist ja vastavalt on ka keelekasutus veidi nõudlikum. Selle kõige tõttu on käesolev tekst mõeldud eelkõige neile, kel KA kursus juba läbitud ja kes hakkavad päris andmeid päris ise analüüsima. Lugejalt eeldatakse ka põhiteadmiste olemasolu R keskkonnast, seega – mujalt saadud eelteadmiste puudumisel – on käesolevat teksti mõttekas uurida peale KA kursuse kodutööde sooritamist ja vastavate kodutööspetsiifiliste juhendmaterjalidega tutvumist. Kodutööde tegemiseks piisab küll viimastest, aga kindlasti ei ole keelatud käesolevaga tutvuda juba kodutöid tehes!

1. Kirjeldav statistika 4

 1.1. Histogramm 4

 1.2. Kirjeldavad statistikud: n, min, max, keskmine, mediaan, mood 5

 1.3. Hajuvusstatistikud: dispersioon, SD, CV, SE, kvantiilid 8

 1.4. Karpdiagramm 9

2. Kahe keskmise võrdlemine 13

 2.1. T-test 13

 2.2. Mann-Whitney test ja Wilcoxoni test 15

3. Kahe või enama keskmise võrdlemine 18

 3.1. Ühefaktoriline ANOVA 18

 3.2. Normaaljaotuse test 21

 3.3. Kruskal-Wallise test 24

4. Seos kahe pideva tunnuse vahel 25

 4.1. Korrelatsioonikordaja 26

 4.2. Lineaarne regressioon 28

 4.3. Mudeli jäägid ja ennustused 30

 4.4. Regressioon läbi nullpunkti 34

5. Kahe sõltumatu tunnusega mudel 36

 5.1. Kahefaktoriline ANOVA 36

 5.2. Kovariaadi kaasamisega mudel 41

 5.3. Marginaalkeskmiste hinnang (*LS-means* või *EMMs*) 43

6. Kolme sõltumatu tunnusega mudel 44

 6.1. Kolmefaktoriline ANOVA 44

 6.2. Keskmiste mitmene võrdlus 49

7. Sõltumatute tunnuste testimise järjekord. Alternatiivsed mudelid I 51

 7.1. Sõltumatute tunnuste testimise järjekord 54

 7.2. Sõltumatu muutuja mittelineaarne teisendus 54

 7.3. Mittelineaarne mudel 55

 7.4. II tüüpi regressioon 57

8. Alternatiivsed mudelid II 60

 8.1.Hierarhiline mudel 60

 8.2. Juhusliku faktoriga mudel 61

 8.3. Kordusmõõtmistega mudel 64

 8.4. Muutuva sõltuvusega mudel 65

9. Kui sõltuv tunnus ei ole pidev tunnus 67

 9.1. Testid sagedustabeli korral 68

 9.2. Logistiline regressioon 71

 9.3 Poissoni regressioon 74

 9.4 Ülehajuvus 76

 9.5 Liigsed nullid sõltuva tunnuse väärtustes 79

10. Andmetevaheline ajaline ja ruumiline sõltuvus 82

 10.1 Ajalise autokorrelatsiooni tuvastamine 82

 10.2 Ajalise autokorrelatsiooni arvestamine mudelis 82

 10.3 Ruumilise autokorrelatsiooni tuvastamine 83

 10.4 Ruumilise autokorrelatsiooni arvestamine mudelis 88

11. Mitmemõõtmelised meetodid 94

 11.1 Peakomponentanalüüs 94

 11.2 Diskriminantanalüüs 96

12. Liigilise koosseisu analüüsid 98

 12.1 Liigirikkuse hindamine 98

 12.2 Koosluste võrdlemine 100

 12.3 Koosluste visualiseerimine 101

13. Veel kasulikku 104

 13.1 Informatsioonikriteerium AIC 104

 13.2 Bayesi statistika

**Kirjeldav statistika**

* 1. **Histogramm**

x <- c(1,2,3,4,10)

hist(x) # histogramm
abline(v=mean(x), lwd=2) # v nagu vertical, lwd nagu line width
abline(v=median(x), col="red", lwd=2) # col nagu colour

Loome alguses andmevektori, joonistame selle põhjal histogrammi ja lisame graafikule ka vastava aritmeetilise keskmise (vaikimisi on joone värv must) ning mediaani (punase joonega). Mõlemad jooned on parema vaadeldavuse huvides tehtud vaikeseadest paksemalt.



Ris on graafikud väga laias ulatuses vormindatavad ja tänu sellele, et see vormindamine käib käskude abil (ja mitte interaktiivselt hiire abil) on graafikute (ja analüüside) reprodutseerimine (nt olukorras, kus lisandub andmeid või soovime graafikut niisama täiendada) väga hõlbus. Erinevaid seadeid on aga väga palju. Hea eestikeelne ülevaade Ri abil loodavatest (lihtsamatest) graafikutest on Märt Mölsi poolt loodud aadressile <http://www-1.ms.ut.ee/mart/R/Rgraafika.html>.

Meie soovime järgnevalt kahte vaatlustevektorit võrrelda. Kui me joonistame kaks eraldi histogrammi siis ei saa me neid vaikimisi mugavalt võrrelda. Lisaks on probleem, et me sooviksime horisontaaltelje skaala fikseerida (R valib skaala vaikimisi antud graafiku andmete põhjal).

x1 <- c(1,2,3,4,10)

x2 <- c(2,3,6,10,12,16)

op <- par(mfrow=c(2,1))

Valmistame Ri oma soovide põhjal ette. Funktsioon par reguleerib Ris väga palju graafika parameetreid. Meie valik mfrow=c(2,1) määrab, et graafikuaken jagatakse alaosadeks nõnda, et “ridu on kaks ja veerge üks”. Paigutus on seega kaks graafikut üksteise alla. Töötaks ka lihtsalt käsk par(mfrow=c(2,1)), aga meie antud käsk salvestab eelnevalt kehtinud graafikaparameetrite väärtused muutujana op. Selle muutuja abil saame hiljem eelmised seadistused uuesti jõustada.

hist(x1, xlim=c(0,20)) # xlim nagu x-telje limits

abline(v=mean(x1), lwd=2)

abline(v=median(x1), col="red", lwd=2)

hist(x2, xlim=c(0,20))

abline(v=mean(x2), lwd=2)

abline(v=median(x2), col="red", lwd=2)



Nüüd näeme, et on tekkinud uus häda – kuna histogrammide klassipiirid ei lange kokku siis on võrdlemine ikkagi mõneti raskendatud. Samas tuleb arvestada ka, et kui kahe valimi maht on erinev siis ei pruugigi puhas sageduste võrdlemine olla parim teguviis. Histogrammi joonistamisel määrab argument freq vaikeväärtusega TRUE, et vertikaalteljel kuvatakse sagedused. Alternatiiviks on valik freq=FALSE, mis määrab, et kuvatakse empiirilised tõenäosused (ehk suhtelised sagedused). Kui klassid on võrdse pikkusega, siis graafiku kuju sellest ei muutu, ent kahe ebavõrdse suurusega valimi võrdlemisel on suhteliste sageduste kuvamine tihti parem valik. Klassipiire reguleerib argument breaks, mis vaikimisi tekitab valimimahtu ja haaret arvestavad mõistlikud klassipiirid. Soovi korral saame need ka ise määrata, andes vektorina ette kõik piiripunktid.

hist(x1, freq=F, breaks=c(0,4,8,12,16,20), ylim=c(0,0.2)) # ylim nagu y-telje limits

abline(v=mean(x1), lwd=2)

abline(v=median(x1), col="red", lwd=2)

hist(x2, freq=F, breaks=c(0,4,8,12,16,20), ylim=c(0,0.2))

abline(v=mean(x2), lwd=2)

abline(v=median(x2), col="red", lwd=2)

par(op)



Horisontaaltelje ulatust ei olnud nüüd enam käsitsi määrata vaja, sest piiripunktid olid juba eelnevalt määratud ja R seab horisontaaltelje ulatuse vaikimisi nende järgi. Viimane käsk määrab, et eelnevalt salvestatud graafikaparameetrid jõustatakse ehk siis järgmisel tekitataval graafikul ei ole enam vaikimisi kahte alamosa.

Lõpuks väärib veel märkimist, et piiripunktile langev valimi väärtus määratakse vaikimisi vasakpoolsesse klassi. Seda reguleerib argument right vaikeväärtusega TRUE (sõna tähendus on antud juhul, et tegu on paremalt-suletud poollõikudega).

* 1. **Kirjeldavad statistikud: n, min, max, keskmine, mediaan, mood**

Valimeid võime kirjeldada või kõrvutada ka ilma visuaalita.

x1 <- c(1,2,3,4,10)

x2 <- c(2,3,6,10,12,16)

summary(x1)

## Min. 1st Qu. Median Mean 3rd Qu. Max.

## 1 2 3 4 4 10

summary(x2)

## Min. 1st Qu. Median Mean 3rd Qu. Max.

## 2.000 3.750 8.000 8.167 11.500 16.000

Funktsioonid min, max, range, quantile, median, mean on muidugi (vastavalt miinimumi, maksimumi, samaaegselt mõlema eelmise, suvalise kvantiili, mediaani või aritmeetilise keskmise leidmiseks) kõik ka eraldi rakendatavad. Kui andmeid on rohkem siis võib lisaks vaja minna ka funktsioone length, sum ja table, mis leiavad vastavalt andmevektori elementide arvu (ehk valimimahu), elementide summa ja elementide sageduse (ehk mitu korda mingit väärtust valimis esineb). Viimane võimaldab määrata ka moodväärtuse. Kui aga erinevaid väärtusi on palju, siis on mugavam kui sagedustabel on sageduste järgi kahanevalt sorteeritud

x <- c(x1,x2) # ühendame kaks valimit

sort(table(x), decreasing=T) # T nagu TRUE

## x

## 2 3 10 1 4 6 12 16

## 2 2 2 1 1 1 1 1

Kuidas saab funktsioon sort aru, et sorteerida tuleb just sageduste ja mitte väärtuste endi järgi? Põhjus on selles, et tegelikult on funktsiooni table poolt tagastatav objekt põhimõtteliselt sageduste vektor – väärtused ise on otsekui nende sageduste nimed (funktsiooni summary väljund on ülesehituse mõttes analoogiline, lihtsalt seal on need nimed tekstilised).

**1.3 Hajuvusstatistikud: dispersioon, SD, CV, SE, kvantiilid**

Kvantiilide leidmist vaatasime juba eelmises alapunktis. Lisaks annavad jaotuse hajuvuse kohta infot veel dispersioon, mida saab leida funktsiooniga var ja selle ruutjuur, mille saab leida ka otse funktsiooniga sd. Mõlema korral kasutab R murru nimetajas vaikimisi suurust „valimimaht miinus üks“. Seega, kui soovime nt valimikeskmisest hälbimise ruutude summat, siis saame selle leida kui

sum((x-mean(x))\*\*2)

## 246.1818

aga ka kui

var(x)\*(length(x)-1)

## 246.1818

kusjuures esimene avaldis on Ris sedavõrd lihtsasti kirja pandav tänu kahele asjaolule – kui me teeme liitmis- või lahutamistehte reaalarvu ja vektori vahel, siis R tekitab arvust sobiva pikkusega vektori ning kui me rakendame vektorile või maatriksile mõnd funktsiooni (antud juhul astendamine) siis üldjuhul see funktsioon rakendub igale elemendile eraldi.

Variatsioonikoefitsendi (*CV*) või keskmise standardvea (*SE*) leidmisel peame lihtsalt vajalikud komponendid kokku panema

sd(x)/mean(x)

## 0.7909911

sd(x)/sqrt(length(x))

## 1.496

Usaldusvahemik üldkogumi keskmisele põhineb standardveal ja seega saame selle lihtsasti leida (vaadeldava valimi jaotus on küll ebasümmeetriline ja usaldusvahemik ei pruugi olla seetõttu usaldusväärne), kuid lisaks tuleb appi võtta ka t-jaotuse kvantiile leida võimaldav funktsioon qt

mean(x)-qt(0.975, length(x)-1)\*sd(x)/sqrt(length(x)) # alumine

## 2.939431

mean(x)+qt(0.975, length(x)-1)\*sd(x)/sqrt(length(x)) # ülemine

## 9.606023

Nagu ikka, kui me soovime mingi Ri funktsiooni kohta täpsemalt teada saada, siis peame andma käsu, kus funktsiooni nimi järgneb küsimärgile. Antud juhul siis näiteks

?qt

avab t-jaotuse kvantiile leida võimaldava funktsiooni abifaili. Asümmeetriakordajat ja järsakust arvutavad funktsioonid leiab nt Ri paketist moments.

**1.4 Karpdiagramm**

Karpdiagrammi (*boxplot*) joonistabki funktsioon boxplot. Vaikimisi kuvatakse mediaani ja kvartiilid Q1 ja Q3 (karp) ning vurrud, mis ulatuvad suurima ja vähima valimi vaatluseni, kuid karbi servast mitte kaugemale kui pooleteistkordne kvartiilide vahe. Kui mõni vaatlus jääb karbi servast veel kaugemale, siis kujutatakse see graafikult punkti abil ja vurr ulatub viimase vaatluseni, mis veel lubatud piiridesse jääb.

boxplot(x) 

Seda pooleteistkordset ulatust määrab funktsiooni parameeter range, parameetri notch=T määrangu korral kuvatakse mediaani ligikaudsed usalduspiirid.

Olukorras, kus meil on mitu valimit, peavad need esmalt olema korrektselt andmestikuna vormindatud. Peatüki alguses toodud andmevektorid paigutame esmalt andmemaatriksisse.

andmed <- data.frame(pikkus=c(x1,x2))

Siin (ja ka mujal) tuleb keeruka käsu mõistmiseks alustada sulgude seest – esmalt ühendasime kaks valimit ning seejärel lõime ühendatud valimi abil andmemaatriksi, kus sellele ühendatud valimite veerule panime nimeks „pikkus“ ning andmemaatriksile endale „andmed“. Muidugi ei ole ühe veeruga andmemaatriksil üldiselt suuremat mõtet ja meiegi soovime lisada kahte valimit eristava tunnuse (ehk nn grupitunnuse)

andmed$grupp <- c(rep("a",length(x1)),rep("b",length(x2)))

kus funktsioon rep kordab väärtust määratud arv kordi. Võime veenduda, et andmemaatriksil on nüüd korrektne kuju

andmed

## pikkus grupp

## 1 1 a

## 2 2 a

## 3 3 a

## 4 4 a

## 5 10 a

## 6 2 b

## 7 3 b

## 8 6 b

## 9 10 b

## 10 12 b

## 11 16 b

Sageli on meil andmed juba eelnevalt tabelarvutusprogrammis (nagu Calc, Excel, Sheets) sisestatud ja siis on meil vaja vaid andmestik korrektselt Ri sisse lugeda (nt funktsiooni read.table abil kui andmefail on mõnes *plain-text* tüüpi (nt .csv) vormingus).

Andmemaatriksi korral oskavad paljud Ri funktsioonid kasutada nn valemisüntaksit, mis kokkuvõttes lihtsustab tööd tunduvalt. Ka boxplot on selline funktsioon. Valemisüntaksis on kesksel kohal lainekese sümbol (*tilde*). Sellest vasakule paigutatakse huvipakkuv tunnus (või tunnused) ja paremale grupeeriv tunnus (või tunnused). Graafikute korral võime alternatiivselt ette kujutada, et vasakule poole paigutatakse tunnus, mille väärtused tuleb kujutada vertikaalteljel, ning paremale poole paigutatakse tunnus, mille väärtused tuleb kujutada horisontaalteljel.

boxplot(pikkus~grupp, data=andmed)



Valemisüntaksit võimaldavate funktsioonide korral on üheks funktsiooni argumendiks alati ka data, millele tuleb väärtuseks anda andmemaatriksi nimi, kust valemisse kirja pandavad tunnusenimed (ja muidugi ka kasutatavad andmed) pärinevad.

1. **Kahe keskmise võrdlemine**

Meie ülesandeks on võrrelda meile huvipakkuva tunnuse üldkeskmisi kahes üldkogumis, kusjuures mõlemast üldkogumist on võetud juhuslik valim. Esmalt eeldame, et tunnus (või teisendatud (nt lograritmitud) tunnus) on mõlemas üldkogumis normaaljaotusega.

* 1. **T-test**

Esmalt sisestame andmed, kusjuures kasutame nüüd funktsiooni rep täiendavat võimalust – argument each määrab, mitu korda järjest iga andmevektori väärtust korratakse.

arvukus <- c(1,2,2,3,3,4,4,5)

grupp <- rep(c("a","b"), each=4)

andmed <- data.frame(arvukus, grupp)

Muidugi oleksime eelneva kolme käsu tulemi saanud anda ka vaid ühe käsureana (siis poleks eraldi objekte „arvukus“ ja „grupp“, mida me tegelikult ei vaja, üldse tekitatud), ent koodi loetavuse huvides on siin kasutatud kolm eraldi koodirida.

Nii nagu mitmeid graafikuid on mugav joonistada valemisüntaksi abil nii on valemisüntaks peaaegu alati aluseks ka statistiliste testide funktsioonidel.

t.test(arvukus~grupp, data=andmed)

## Welch Two Sample t-test

##

## data: arvukus by grupp

## t = -3.4641, df = 6, p-value = 0.0134

## alternative hypothesis: true difference in means is not equal to 0

## 95 percent confidence interval:

## -3.4127252 -0.5872748 # alumine ja ülemine usalduspiir

## sample estimates:

## mean in group a mean in group b

## 2 4

Väljundina saame nii t-statistiku väärtuse, vabadusastmete kui ka p-väärtuse. Lisaks veel ka usalduspiirid gruppidele vastavate üldkogumite keskmiste vahele ja valimikeskmised gruppides.

Funktsioonil t.test on mitu olulist argumenti, mida peame väärtustama, kui meie poolt soovitud test erineb tavapäraseimast variandist. Valime var.equal=T kui oleme kindlad, et jaotuste dispersioonid gruppides on võrdsed (vaikimisi eeldatakse, et see ei pruugi nii olla). Valime paired=T kui meie kaks gruppi on hoopis samade objektide kordusmõõtmised (nt enne ja pärast mingit manipulatsiooni). Valik alternative="greater" või alternative="less" määrab, et soovime testida ühepoolset hüpoteesi (vastavalt, et esimese üldkogumi keskmine on suurem kui teise üldkogumi keskmine või esimese üldkogumi keskmine on väiksem kui teise üldkogumi keskmine). Vaikimisi (ehk kui me ise ei määra alternative väärtust) testitakse kahepoolset hüpoteesi ehk seda, kas kaks üldkeskmist üldse erinevad teineteisest.

Funktsiooni t.test saame mugavalt kasutada ka üldkogumi keskmistele usaldusvahemike leidmiseks. Nii näiteks saame käsuga

t.test(arvukus~1, data=andmed, subset=(grupp=="b"))

## One Sample t-test

##

## data: arvukus

## t = 9.798, df = 3, p-value = 0.00226 # grupi b keskmine nullist erinev

## alternative hypothesis: true mean is not equal to 0

## 95 percent confidence interval:

## 2.700772 5.299228 # usaldusvahemik grupi b üldkeskmisele

## sample estimates:

## mean of x

## 4

Siin vajab selgitamist kaks aspekti. Kui lainekese taga on vaid number üks (mitte grupeeriva tunnuse nimi) siis see, tähendab, et kõik meie andmed on samast grupist (ehk grupeerimist ei kasutata). Argument subset on tüüpiliselt statistilist analüüsi võimaldavatel funktsioonidel olemas ning selle abil saame kergelt kasutada oma andmemaatriksi mingit alamosa ilma uut alamandmestikku eraldi objektina loomata. Antud näites valisime oma analüüsi kaasamiseks vaid need andmeread, kus tunnuse grupp väärtus oli “b”. Oluline on tähele panna, et kaks võrdusmärki võrdsuse kontrollimisel ei ole mitte kirjaviga, vaid just nii see Ris käibki.

Vahel võib tekkida vajadus testida erinevust nullist erinevast arvust. Näiteks võime ette kujutada, et meile pakub huvi, kas lageraie tulemusena väheneb meid huvitava linnuliigi kooruvate linnupoegade arv terves metsas (normeerituna metsa pindala suhtes) enam kui 10 võrra. Olgu meil kasutada kordusmõõtmised eri metsades läbi aastate hooajal enne ja pärast lageraiet.

enne=c(39.3,46.8,21.1,48.3,59.7,73.1,31.0,52.9,74.3)

pärast=c(19.3,19.7,17.4,38.2,35.2,67.3,11.2,33.0,34.1)

andmed=data.frame(poegi=c(enne,pärast), grupp=rep(c("enne","pärast"), each=length(enne)))

Muidugi peavad kordusmõõtmiste korral olema vaatlused samas järjekorras ehk siis nt sama metsa kohta käivad väärtused 39.3 (enne) ja 19.3 (pärast).

Testi tehes valime altrernative="greater", sest arvestame, et lageraiete mõju on antud kontekstis kindlasti negatiivne ja mu=10, sest me soovime näidata, et lageraiete tulemusel tekkinud erinevus on keskmiselt enam kui 10 ühiku suurune.

t.test(poegi~grupp, data=andmed, paired=T, altrernative="greater", mu=10)

## Paired t-test

##

## data: poegi by grupp

## t = 2.3764, df = 8, p-value = 0.0448

## alternative hypothesis: true difference in means is not equal to 10

## 95 percent confidence interval:

## 10.26682 27.75540

## sample estimates:

## mean of the differences

## 19.01111

Nagu näha on keskmine erinevus (enne ja pärast väärtuste vahel) lausa 19 ühikut, ent erinevuste suur varieeruvus eri metsades võimaldab sisuka hüpoteesi (keskmine erinevus üldkogumis enam kui 10 ühikut) üsna napilt tõestatuks lugeda (p=0.045).

* 1. **Mann-Whitney test ja Wilcoxoni test**

Kui tunnuse jaotust ei õnnestu normaaljaotuseks teisendada siis me üldiselt t-testi kasutada ei tohiks. Ka järjestustunnuse korral ei ole t-testi kasutamine õigustatud. Küll aga võime kasutada Mann-Whitney mitteparameetrilist testi.

Tekitame andmestiku. Kasutame sealjuures juhuslike arvude tekitamise võimalust: R oskab nimelt küsimise peale meile genereerida (pseudo)juhuslikke arve õige mitmest jaotusest.

faktor <- rep(c("A","B"), each=20)

y <- c(rexp(20, 1), 2+rexp(20, 1)) # 2x 20 arvu eksponentjaotusest

andmed2 <- data.frame(faktor, y)

boxplot(y~faktor, data=andmed2)

Kuivõrd andmed on genereeritud juhuslikult siis ei saa me koodi uuesti jooksutamisel täpselt identset tulemust (ei graafikul ega siis ka testimise tulemusena).



Päris üldistel eeldustel on Mann-Whitney sisukas hüpotees lihtsalt, et jaotused erinevad. Kui oleme valmis tegema väikese lisaeelduse (gruppide jaotused on sama varieeruvuse ja kujuga) siis võime lugeda, et Mann-Whitney test kontrollib mediaanide võrdsust üldkogumis.

wilcox.test(y~faktor, data=andmed2)

## Wilcoxon rank sum exact test

## data: y by faktor

## W = 71, p-value = 0.0002919

## alternative hypothesis: true location shift is not equal to 0

Ris kannab Mann-Whitney testi tegev funktsioon nime wilcox.test, kuna sõltuvate valimite korral (kordusmõõtmised) ongi kasutatavaks testiks just Wilcoxoni test. Analoogiliselt t-testiga tuleb sõltuvate valimite korral valida paired=T.

1. **Kahe või enama keskmise võrdlemine**

Kui meil on vaid kaks üldkogumit siis saame kasutada t-testi, ent kui üldkogumeid on rohkem kui kaks siis ei saa me ainult t-testiga läbi. Nt 6 üldkogumi korral peaksime küsimusele “kas kõikides üldkogumites on tunnuse keskväärtus sama” vastuse leidmiseks tegema 15 eraldi paariviisilist testi. See ei ole üldiselt hea mõte (sest võib kergesti viia mõne valepositiivse tulemuseni).

**3.1. Ühefaktoriline ANOVA**

Tekitame esmalt andmed

faktor <- rep(c("A","B","C"), each=5)

y <- c(1,3,4,5,7,4,6,6,7,8,7,8,8,9,9)

andmed <- data.frame(faktor, y)

boxplot(y~faktor, data=andmed)



Kõiki lineaarseid mudeleid (ANOVA, ANCOVA, lineaarne regressioon, üldine lineaarne mudel) saab R-is andmetele sobitada funktsiooniga lm. Kuivõrd sageli on valmis mudelile vaja hiljem rakendada veel erinevaid funktsioone, siis on otstarbekas mudel kohe alguses objektina salvestada.

m <- lm(y~faktor, data=andmed)

Tüüp I ANOVA tabeli saab kätte funktsiooniga anova

anova(m)

## Analysis of Variance Table
##

## Response: y

## Df Sum Sq Mean Sq F value Pr(>F)

## faktor 2 44.133 22.0667 8.3797 0.005277 \*\*

## Residuals 12 31.600 2.6333

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Antud tabelist leiame nii mudeli vabadusastmed (2), jääkhajuvuse vabadusastmed (12), mudeli ruutude summa (44.1), jääkhajuvuse ruutude summa (31.6) kui ka F-statistiku väärtuse (8.38) ja p-väärtuse (0.005).

Funktsiooniga summary saame näha veel muudki huvipakkuvat.

summary(m)

## Call:

## lm(formula = y ~ faktor, data = andmed)

##

## Residuals:

## Min 1Q Median 3Q Max

## -3.0 -0.6 -0.2 0.8 3.0

##

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) 4.0000 0.7257 5.512 0.000134 \*\*\*

## faktorB 2.2000 1.0263 2.144 0.053250 .

## faktorC 4.2000 1.0263 4.092 0.001493 \*\*

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

##

## Residual standard error: 1.623 on 12 degrees of freedom

## Multiple R-squared: 0.5827, Adjusted R-squared: 0.5132

## F-statistic: 8.38 on 2 and 12 DF, p-value: 0.005277

Siit tabelist leiame determinatsioonikordaja (R-ruut 0.583 ehk 58.3% ja selle parandatud version 0.513 ehk 51.3%), mudeli jääkide jaotuse ülevaate, mudeli jääkide standardviga (residual standard error 1.62 iseloomustab mudeli poolt hinnatud vaatluste hajuvust grupikeskmiste ümber), kogu mudeli olulisuse F-testi (p-väärtus 0.005), mis antud juhul langeb kokku eelmisest tabelist saadud testiga (sest kogu mudel koosnebki vaid ühest faktorist).

Tabeli põhisisuks on mudeli parameetrite hinnangud. Mudeli vabaliikmeks (baastasemeks) võetakse vaikimisi faktori kõige esimene tase (tähestikulises järjekorras). Parameeter “(Intercept)” vastab seega antud juhul grupi A üldkeskmise hinnangule (4.0), teises reas leiduv parameeter “faktorB” hindab grupi B üldkeskmise erinevust grupi A üldkeskmisest (2.2) ning kolmanda rea parameeter “faktorC” hindab grupi C üldkeskmise erinevust grupi A üldkeskmisest (4.2). Tabelis sisalduvad olulisuse testid kontrollivad vastava parameetri nulliga võrdumist. Seega on esimese rea testi küsimuseks “Kas grupi A üldkeskmine on 0?” (ei ole, p<0.001), teise rea testi küsimuseks “Kas grupi A ja B üldkeskmised on võrdsed?” (ei saa väita, et ei ole, p=0.053) ja kolmada rea testi küsimuseks “Kas grupi A ja C üldkeskmised on võrdsed?” (ei ole, p=0.001).

Teistsuguse mudeli parametriseeringu saame kui sobitame mudeli kujul

m2 <- lm(y~faktor-1, data=andmed)

Täiendus valemis tähendab, et me ei soovi mudelis vabaliiget kasutada. Praktikas tähendab see, et esimene parameeter vastab jätkuvalt esimese taseme keskmisele, ent teised parameetrid on siis samuti tasemete keskmise, mitte erinevuse baastasemest, hinnangud.

summary(m2)

## Call:

## lm(formula = y ~ faktor - 1, data = andmed)

##

## Residuals:

## Min 1Q Median 3Q Max

## -3.0 -0.6 -0.2 0.8 3.0

##

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## faktorA 4.0000 0.7257 5.512 0.000134 \*\*\*

## faktorB 6.2000 0.7257 8.543 1.91e-06 \*\*\*

## faktorC 8.2000 0.7257 11.299 9.43e-08 \*\*\*

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

##

## Residual standard error: 1.623 on 12 degrees of freedom

## Multiple R-squared: 0.9506, Adjusted R-squared: 0.9383

## F-statistic: 77.01 on 3 and 12 DF, p-value: 4.158e-08

Nagu näeme on teise ja kolmanda parameetri hinnangud nüüd hoopis teistsugused (vastavalt 6.2 ja 8.2) ning mõistagi ei ole ka p-väärtused enam samad (sest nüüd on nt teise rea testi küsimuseks hoopis “Kas grupi B üldkeskmine on 0?”.

Märkame, et ka determinatsioonikordaja ja kogu mudeli olulisuse testi väärtused on muutunud. Miks nii? Mudel on ju põhimõtteliselt sama, mis enne. Mudel ongi tegelikult sama (seda teame teooriast ja näeme ka nt jääkide jaotusest), ent R arvestab, et kui vabaliiget mudelis ei ole, siis järelikult on meie nö nullmudeliks selline mudel, kus puudub ka vabaliige ehk siis mudel, mille keskmine on lihtsalt 0. Kuna üldiselt võib R-ruutu tõlgendada kui “mitu protsenti jääkide dispersioon väheneb, kui jääke mõõdetakse rühmade keskmiste suhtes võrreldes sellega, kui neid arvutatakse kogu valimi keskmise suhtes” siis antud juhul on R-ruutu tõlgenduseks “mitu protsenti jääkide dispersioon väheneb, kui jääke mõõdetakse rühmade keskmiste suhtes võrreldes sellega, kui neid arvutatakse lihtsalt nulli suhtes”. Ka F-test, mille aluseks muidu oleks küsimus “kas mõni keskmine on teistest erinev”, võtab nüüd aluseks küsimuse “kas mõni keskmine on nullist erinev”.

**3.2 Normaaljaotuse test**

ANOVA ehk dispersioonanalüüsi eelduste täidetuse kontrolliks on palju võimalusi, sest nii normaaljaotusele vastavust kui ka dispersioonide võrdsust võimaldavad kontrollida mitmed erinevad statistilised testid. Alustada tuleks aga alati joonistest. Antud juhul oleme nt karpdiagrammi juba joonistanud. Keerulisemate analüüside korral tegeleme eelduste täidetuse kontrollimisel üldiselt vaid mudeli jääkidega. Need saame kätte funktsiooni resid abil.

hist(resid(m))

Kui kõigis gruppides oleks uuritav tunnus võrdse dispersiooniga ja normaaljaotusega siis oleks kõikide jääkide ühine jaotus samuti sellesama dispersiooniga nullkeskmisega normaaljaotus.

Normaaljaotuse eeldust võimaldab Ris kontrollida nt Shapiro-Wilksi testi tegev funktsioon shapiro.test. Eelduste kontrollimisel vastab sisukas hüpotees (ehk väike p-väärtus) eelduste rikutusele. Paraku ei ole selle funktsiooniga võimalik kasutada valemisüntaksit. Kui me soovime teha testi igas grupis eraldi siis üheks võimaluseks on alamandmestiku vahetu selekteerimine.

shapiro.test(andmed$y[andmed$faktor=="A"])

## Shapiro-Wilk normality test

## data: andmed$y[andmed$faktor == "A"]

## W = 0.99929, p-value = 0.9998

Näeme, et ootuspäraselt ei näita test sedavõrd väikese valimimahu korral eelduste rikutust. Kui me soovime läbi viia testi iga grupi jaoks siis on Ris tunduvalt lihtsam kasutada abifunktsiooni. Funktsioon by ei võimalda kahjuks valemisüntaksi kasutamist, seetõttu peame tunnused koos andmestikuga ette andma vahetult. Funktsiooni esimene argument on uuritav tunnus, teine argument on grupeeriv tunnus ning kolmas argument on funktsioon, mida igas grupis uuritavale tunnusele rakendada soovime.

by(andmed$y,andmed$faktor,shapiro.test)

## andmed$faktor: A

## Shapiro-Wilk normality test

## data: dd[x, ]

## W = 0.99929, p-value = 0.9998

## ------------------------------------------------------------

## andmed$faktor: B

## Shapiro-Wilk normality test

## data: dd[x, ]

## W = 0.95563, p-value = 0.7773

## ------------------------------------------------------------

## andmed$faktor: C

## Shapiro-Wilk normality test

## data: dd[x, ]

## W = 0.88104, p-value = 0.314

Ootuspäraselt ei tuvasta test üheski grupis erinevust normaaljaotusest.

Dispersioonide võrdsuse testimisel on heaks robustseks testiks Fligner-Killeen’i test (robustsus tähendab siin seda, et test töötab korrektselt ka siis, kui tunnuse jaotuse gruppides erineb normaaljaotusest).

fligner.test(y~faktor, data=andmed)

## Fligner-Killeen test of homogeneity of variances

##

## data: y by faktor

## Fligner-Killeen:med chi-squared = 2.3077, df = 2, p-value = 0.3154

Ka dispersioonide võrdsuse eeldus näib kehtivat.

**3.3. Kruskal-Wallise test**

Juhul kui tunnused jaotused gruppides jäävad selgelt ebasümmeetriliseks ka pärast meiepoolset võimalikku teisendust siis peame appi võtma mitteparameetrilise testi. Kruskal-Wallise test ongi Mann-Whitney testi laiendus juhuks kui gruppe on enam kui kaks. Analoogiliselt Mann-Whitney testiga peame mõistliku sisuka hüpoteesi jaoks tegema lisaeelduse, päris üldisel juhul on sisukaks hüpoteesiks lihtsalt, et „vähemalt ühe grupi jaotus erineb teistest“. Kui me oleme aga valmis eeldama, et gruppide jaotused on sama varieeruvuse ja kujuga, siis kontrollib test sisuliselt mediaanide võrdsust.

Muuhulgas tähendab eelnev, et olukorras, kus (pärast ühtki teisendust) tunnuse jaotus gruppides ei ole normaaljaotus ja mõnes grupis on dispersioon selgelt suurem/väiksem kui teistes, pole meil võimalik keskmiste võrdsust üldkogumis usaldusväärselt testida. Õnneks on sellised juhud väga harvad.

kruskal.test(y~faktor, data=andmed)

 Kruskal-Wallis rank sum test

data: y by faktor

Kruskal-Wallis chi-squared = 8.8794, df = 2, p-value = 0.0118

Antud väljundi põhjal näeme mitteparameetrilise testi tüüpilist omadust – testi võimsus on parameetrilise testiga võrreldes väiksem (ehk p-väärtus suurem). See omab muidugi vaid tähendust olukorras, kus mõlemad testid on kasutatavad (ehk parameetrilise testi eeldused on täidetud).

1. **Seos kahe pideva tunnuse vahel**

Ka ainult pidevate tunnuste kastamisel peaks meie esimeseks sammuks olema jooniste vaatlemine. Histogrammide joonistamist oleme juba vaadanud. Kahe tunnuse ühisjaotust valimis kujutame üldjuhul hajuvusdiagrammi (*scatterplot*) abil.

Ris on hulganisti näidisandmestikke. Nendest saab ülevaate käsuga data(). Meie valime kasutuseks hilistoomingate (*Prunus serotina*) andmestiku nimega „trees“. Andmestiku detailkirjeldust saame vaadata abikäsuga ?trees. Sealt selgub muuhulgas, et tunnus „Girth“ on tegelikult puu diameeter rinna kõrgusel. Andmestikust lühiülevaate saamiseks on väärt abimehed funktsioonid head ja str. Esimene esitab andmestiku algusread koos tunnuste nimedega, teine on kasutatav mistahes Ri objektidega ja annab andmemaatriksi korral üsna sarnase väljundi.

head(trees)

## Girth Height Volume

## 1 8.3 70 10.3

## 2 8.6 65 10.3

## 3 8.8 63 10.2

## 4 10.5 72 16.4

## 5 10.7 81 18.8

## 6 10.8 83 19.7

str(trees)

## 'data.frame': 31 obs. of 3 variables:

## $ Girth : num 8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2 ...

## $ Height: num 70 65 63 72 81 83 66 75 80 75 ...

## $ Volume: num 10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 ...

Nagu näha näitab str ära ka valimimahu ja tunnuste tüübid (antud juhul kõik numbrilised). Käsu plot abil saame luua hajuvusdiagramme (kuigi R valib antud käsu korral graafikutüübi tegelikult vastavalt tunnuste tüübile, nt kui sõltuv tunnus on numbriline ja sõltumatu kategooriline (Ris tüüpi „factor“ või „character“) siis joonistab R karpidagrammi).

plot(Girth~Height, data=trees)



**4.1 Korrelatsioonikordaja**

Funktsioon cor võimaldab leida korrelatsioonikordaja väärtuse. See funktsioon ei tunne aga valemisüntaksit. Vaikimisi on leitava korrelatsioonikordaja tüüp Pearsoni korrelatsioon.

cor(trees$Girth, trees$Height)

## [1] 0.5192801

Spearmani korrelatsioonikordaja saame leida lisavalikuga

cor(trees$Girth, trees$Height, method="spearman")

## [1] 0.4408387

Üldiselt on mugavam on kasutada funktsiooni cor.test, sest see tunneb ka valemisüntaksit ning leiab lisaks veel ka korrelatsioonikordaja usaldusvahemiku ja teeb statistilise olulisuse testi.

cor.test(~Height+Girth, data=trees)

## Pearson's product-moment correlation

## data: Height and Girth

## t = 3.2722, df = 29, p-value = 0.002758

## alternative hypothesis: true correlation is not equal to 0

## 95 percent confidence interval:

## 0.2021327 0.7378538

## sample estimates:

## cor

## 0.5192801

Valemi kirjutamisel tuleb antud funktsiooni korral mõlemad tunnusenimed märkida paremale poole lainekest. Selle põhimõtte aluseks on korrelatsiooni sümmeetrilisus – tunnuste järjekord ei ole oluline.

Analoogiliselt saab valikuga method=“spearman“ valida Spearmani korrelatsioonikordaja (tegelikult võimaldab R nn osalist valimist, mis tähendab, et tekstilisi argumendi väärtusi ei pea tervikuna välja kirjutama – piisab, et meie poolt kirjutatu võimaldaks soovitava väärtuse üheselt määrata)

cor.test(~Height+Girth, data=trees, method="s")

## Spearman's rank correlation rho

## data: Height and Girth

## S = 2773.4, p-value = 0.01306

## alternative hypothesis: true rho is not equal to 0

## sample estimates:

## rho

## 0.4408387

## Warning message:

## In cor.test.default(x = c(70, 65, 63, 72, 81, 83, 66, 75, 80, 75, :

## Cannot compute exact p-value with ties

Näeme, et R kurdab kokkulangevate väärtuste üle. Ril on kaks veateatetüüpi. Hoiatus (*warning*) on üldjuhul ohutu ja juhib vaid meie tähelepanu mingile asjaolule (antud juhul sellele, et kui tunnused oleksid pidevad siis pmst ei tohiks kokkulangevaid väärtusi andmestikus olla). Viga (*error*) on tõsine probleem, mis ei võimalda vastaval funktsioonil tööd lõpetada ja seetõttu ei väljasta funktsioon oodatud tulemust.

Ka funktsioonil cor on siiski oma väärtus. Kui anname sisendiks numbriliste tunnustega maatriksi (kus enam kui 2 tunnust), siis saame väljundiks terve korrelatsioonimaatriksi (maatriks, kus kõikvõimalikud paariviisilised korrelatsioonid)

cor(trees)

## Girth Height Volume

## Girth 1.0000000 0.5192801 0.9671194

## Height 0.5192801 1.0000000 0.5982497

## Volume 0.9671194 0.5982497 1.0000000

Nii näeme antud maatriksist, et tüve ruumala ja diameetri vaheline korrelatsioonikordaja on 0.97 (mis on muidugi ootuspärane, sest ruumala arvutamisel kasutatakse muuhulgas ka diameetrit).

**4.2 Lineaarne regressioon**

Lineaarne regressioon on Ri jaoks analoogiline lineaarne mudel nagu dispersioonanalüüs ja seetõttu on ka selle mudeliga seotud käsud identsed.

m3 <- lm(Girth~Height, data=trees)

anova(m3)

## Analysis of Variance Table

## Response: Girth

## Df Sum Sq Mean Sq F value Pr(>F)

## Height 1 79.665 79.665 10.707 0.002758 \*\*

## Residuals 29 215.772 7.440

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

summary(m3)

## Call:

## lm(formula = Girth ~ Height, data = trees)

## Residuals:

## Min 1Q Median 3Q Max

## -4.2386 -1.9205 -0.0714 2.7450 4.5384

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) -6.18839 5.96020 -1.038 0.30772

## Height 0.25575 0.07816 3.272 0.00276 \*\*

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

## Residual standard error: 2.728 on 29 degrees of freedom

## Multiple R-squared: 0.2697, Adjusted R-squared: 0.2445

## F-statistic: 10.71 on 1 and 29 DF, p-value: 0.002758

Viimasest väljundist leiame mudeli parameetrite hinnangud (vabaliige=-6.2, sirge tõus 0.26), hinnagute standardvead (vabaliikme SE -5.96, sirge tõusu SE 0.08) ja nullist erinemise testid (p=0.31 ja p=0.003 vastavalt vabaliikme ja tõusu jaoks), mudeli jääkide standardhälbe hinnangu (2.7) ja mudeli determinatsioonikordaja (0.27), mis on lineaarse regressiooni korral võrdne Pearsoni korrelatsioonikordaja ruuduga. Lisaks on väljundis ka jääkide jaotuse ülevaade. Kogu mudeli olulisuse testi p-väärtus langeb kokku tõusu olulisuse testi p-väärtusega, sest taaskord on eelduseks, et vabaliige on mudelis juba vaikimisi.

Edaspidi kui mudelites on mitu tunnust on veelgi enam kasulik järgnev loogika. Iga meid huvitava tunnuse (või meid huvitavate tunnuste) olulisuse test on sisuliselt kahe mudeli (nt mudel A ja mudel B) võrdlus: A ja B on muidu täpselt samasugused aga mudelis A ei sisaldu meid huvitav tunnus (või tunnused) ning mudelis B sisaldub. Kui mudel B on mudelist A oluliselt parem siis järelikult on lisatud tunnus (või tunnused) olulised. Mudelit A võime nimetada baasmudeliks ja mudelit B täiendatud mudeliks.

Viimases näites oli mudeliks A ehk baasmudeliks vaid vabaliiget sisaldav mudel ning mudeliks B ehk täiendatud mudeliks vabaliiget ja kõrgust sisaldav mudel.

Ris saame jooniseid jooksvalt täiendada (nägime seda juba histogrammi joonistamisel). See tähendab, et kui me pole graafikuakent vahepeal sulgenud siis saame mõnede käskudega graafikule detaile lisada (kui oleme vahepeal akna sulgenud või mõne muu graafiku joonistanud siis tuleb muidugi esmalt anda uuesti ka põhikäsk (meie näites siis plot(Girth~Height, data=trees)). Üks selline täienduskäsk on abline, mis, saades argumendiks lineaarse regressiooni mudeli, lisab joonisele regressioonisirge.

abline(m3)



Kui soovime siis saame sedasama käsku kasutada ka nii, et anname ise väärtustena ette argumendid a ja b, mis on vastavalt vabaliige ja tõus. Muidugi saame muuta ka joone värvust ja paksust (nt abline(a=0, b=0.18, lwd=2, col="red")).

**4.3 Mudeli jäägid ja ennustused**

Mudeli jäägid saame jätkuvalt leida funktsiooni resid abil ning ennustused annab funktsioon predict. Vaikimisi on ennustusteks lihtsalt regressioonisirge väärtused antud argumentide korral (ja jääkideks siis vastavalt tegelike väärtuste ja ennustatud väärtuste vahed). Andmestikus esimese puu kõrgus on

trees$Height[1]

## [1] 70

ning regressioonivõrrandi põhjal on sellise kasvuga puude keskmine diameeter

predict(m3)[1]

## 1

## 11.7139

Kuivõrd antud puu tegelik diameeter on

trees$Girth[1]

## [1] 8.3

siis järelikult on vastava andmepunkti korral mudeli jäägiks

resid(m3)[1]

## 1

## -3.413904

Vaatleme nüüd meie mudeli jääkide jaotuse histogrammi

hist(resid(m3))



Kui funktsioonile plot anda argumendiks lineaarne mudel siis joonistatakse selle mudeli kohta erinevaid mudeli eeldusi kontrollida aitavaid graafikuid (teine argument määrab, milliseid neist nägha soovime). Selliseid graafikuid nimetatakse ka mudeli diagnostikagraafikuteks. Neist esimene visualiseerib samal graafikul mudeli ennustused ja jäägid (analoogiline meie üle-eelmise graafikuga).

plot(m3,1)



Horsiontaalteljel on mudeli ennustused, vertikaalteljel mudeli jäägid. Lisaväärtuseks on siin absoluutväärtuselt suuremaid jääke omavate valimi elementide väljatoomine (järjekorranumber andmestikus) ja graafikule sobitatud libiseva keskmise joon, mis peaks aitama otsustada, kas jääkide keskväärtus on ikka läbivalt null (ehk kas tunnustevaheline seos on ikka lineaarne).

Erinevaid diagnostikagraafikuid (ja igaühel omakorda sätteid) on palju. Nende kohta saab täpsemalt vaadata käsuga ?plot.lm

Funktsiooninimi on seesugune sellepärast, et Ris on tegelikult palju sama nimega funktsioone, ent kui me funktsiooni välja kutsume siis otsustakse argumendi põhjal, millist neist kasutada tuleb (seetõttu valimegi abi küsimusel, et meile pakub huvi lineaarsete mudelite (Ris objekti klass „lm“) kohta graafikuid joonistav funktsioon).

Homoskedastilisuse kontrolliks sobib hästi antud funktsiooni väljastatav kolmas graafik

plot(m3,3)



Graafikul on horisontaalteljel taas sobitatud väärtused. Vertikaalteljel on standardiseeritud jääkide absoluutväärtuste ruutjuured. Jääke standardiseeritakse, kuna tavajääkide korral on lineaarses regressioonis servmiste jääkide dispersioon suurem ning lisaks saavutatakse standardiseerimisega ka see, et jääkide dispersioon on võrdne ühega. Absoluutväärtustest on täiendavalt ruutjuur võetud absoluutjääkide jaotuse asümmeetria vähendamise eesmärgil. Libiseva keskmise joon peaks homoskedastilisuse eelduse täidetuse korral olema ligikaudu horisontaalne sirge.

Vahel on meil oma regressioonimudelit vaja ka päris ennustamiseks. Näiteks me teame puude kõrgusi (aga ei tea nende puude diameetreid) ning soovime eelnevalt leitud regressioonisirge abil neid prognoosida. Ris vajame selleks täiendavat andmemaatriksit, kus need kõrgused kirjas ning tunnusenimi peab täpselt vastama mudelis kasutatule.

uued <- data.frame(Height=c(90,95,100))

predict(m3, uued)

## 1 2 3

## 16.82885 18.10758 19.38632

**4.4 Regressioon läbi nullpunkti**

Regressioon läbi nullpunkti on üldjuhul kasutusel olukorras, kus nii sõltumatu kui sõltuva tunnuse väärtused on mittenegatiivsed. Ka meie näiteandmestik on selliste tunnustega.

Kuivõrd nullpunkti läbiv regressioon on lihtsalt selline regressioonimudel, kus puudub vabaliige, siis oskame sellist mudelit koheselt tellida.

m4=lm(Girth~Height-1, data=trees)

summary(m4)

## Call:

## lm(formula = Girth ~ Height - 1, data = trees)

##

## Residuals:

## Min 1Q Median 3Q Max

## -3.9410 -2.1030 -0.5415 2.7216 5.3862

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## Height 0.174871 0.006433 27.18 <2e-16 \*\*\*

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

## Residual standard error: 2.731 on 30 degrees of freedom

## Multiple R-squared: 0.961, Adjusted R-squared: 0.9597

## F-statistic: 739 on 1 and 30 DF, p-value: < 2.2e-16

Siin on taas asjakohane meenutada, et vabaliikmeta mudeli korral on baasmudeliks mudel, mis ei sisalda ühtegi liiget (ehk siis mudel, mis prognoosib sõltumata sõltumatu tunnuse väärtusest sõltuva tunnuse väärtuseks nulli). Ilmselgelt on meie täiendatud mudel sellisest nullmudelist mäekõrguselt parem ja see väljendub ka täiendatud mudeli väga väikeses p-väärtuses. Samas on ka selge, et nulli prognoosiv mudel ei ole mittenegatiivsete andmete korral sugugi mõistlik (ehk siis teisiti öeldes: baasmudel on antud olukorras sõltumata andmete täpsetest väärtustest ebamõistlik).

Mõistlik oleks vabaliikmeta mudeli võrdlemine vaid vabaliiget sisaldava mudeliga, ent seda pole statistilise testi abil teha võimalik (küll aga saab võrrelda informatsioonikriteeriumite alusel).

Kui me aga oleme otsustanud mittenegatiivsetele andmetele nullpunkti läbiva regressioonisirge sobitada ja me soovime vabaliikmeta regressioonimudeli tõusu jaoks saada mõistlikku statistilist testi, siis on otstarbekas nii sõltuv kui sõltumatu tunnus esmalt keskmistada. Seda saame Ris teha funktsiooni scale abil (vaikimisi see funktsioon lausa standardiseerib ehk siis tsentreerib (nullkeskmine) ja normeerib (dispersioon võrdne ühega) tunnuse), ent antud olukorras see statistilist olulisust ei mõjuta.

m5=lm(scale(Girth)~scale(Height)-1, data=trees)

summary(m5)

## Call:

## lm(formula = scale(Girth) ~ scale(Height) - 1, data = trees)

## Residuals:

## Min 1Q Median 3Q Max

## -1.35068 -0.61199 -0.02274 0.87472 1.44621

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## scale(Height) 0.5193 0.1560 3.328 0.00232 \*\*

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

## Residual standard error: 0.8546 on 30 degrees of freedom

## Multiple R-squared: 0.2697, Adjusted R-squared: 0.2453

## F-statistic: 11.08 on 1 and 30 DF, p-value: 0.002323

Nüüd on tõusu statistiline olulisus samas suurusjärgus vabaliikmega mudeli korral saaduga. See on nii seetõttu, et keskmistatud sõltuva tunnuse korral on ühtlaselt nulli prognoosiv baasmudel täiesti mõeldav ja seetõttu ka mudelite omavaheline võrdlus mõistlik.

Seega oleks sellises ebatavalises olukorras tulemuste presenteerimisel mõistlik ka lugejale/kuulajale situatsiooni mõne lausega selgitada ning esitada küll tõusu hinnang (koos SEga) mudelist m4, ent tõusu olulisuse testi jaoks võtta andmed mudeli m5 väljundi alumiselt realt (antud juhul siis F=11.1, df1=1, df2=30, p=0.002).

1. **Kahe sõltumatu tunnusega mudel**

Tavaliselt on vaja luua mudel, kus on koos mitut sõltumatut tunnust, kuna sõltuv tunnus on (eeldatavalt) nende kõigi poolt mõjutatud. Sageli võib seejuures ühe sõltumatu tunnuse mõju sõltuda teise sõltumatu tunnuse väärtusest ehk siis sõltumatute tunnuste vahel on interaktsioon.

* 1. **Kahefaktoriline ANOVA**

Vaatleme merisigade hammaste kasvu andmestikku ToothGrowth, kus vaadeldud hammaste kasvu eest vastutavate rakkude (odontoblastide) pikkust erineva doosi C-vitamiini manustamise järgselt. Vitamiini on katses seejuures manustatud kahel eri viisil: apelsinimahlaga (OJ) või askorbiinhappega (VC).

head(ToothGrowth)

## len supp dose

## 1 4.2 VC 0.5

## 2 11.5 VC 0.5

## 3 7.3 VC 0.5

## 4 5.8 VC 0.5

## 5 6.4 VC 0.5

## 6 10.0 VC 0.5

Valemisüntaksit tundev funktsioon xtabs võimaldab meil mugavalt sagedustabeleid koostada – siis näeme kohe, kuidas vaatlused jaotuvad.

xtabs(~dose+supp, data=ToothGrowth)

## supp

## dose OJ VC

## 0.5 10 10

## 1 10 10

## 2 10 10

Näeme, et valim on tasakaaluline (igat faktorite kombinatsiooni on mõõdetud kümnel korral) ja doosil on kolm taset.

m1 <- lm(len~factor(dose)\*supp, data=ToothGrowth)

Kasutasime funktsiooni factor, kuivõrd tunnus „dose“ oli andmestikus numbriline (nüüd muutsime selle mudelis kategooriliseks). Ehkki doos on oma olemuselt pidev tunnus ei ole meil ilmselt põhjust oodata, et selle mõju peaks olema lineaarne (s.o. et ilmselt ei pea kaks korda suurema doosiga kaasnema kaks korda suurem mõju). Korrutusmärk valemis näitab, et me soovime kaasata nii antud tunnuste peamõjud kui ka koosmõju (kui kasutanuksime korrutismärgi asemel plussmärki siis oleks mudelisse kaasatud vaid peamõjud).

anova(m1)

## Analysis of Variance Table

## Response: len

## Df Sum Sq Mean Sq F value Pr(>F)

## factor(dose) 2 2426.43 1213.22 92.000 < 2.2e-16 \*\*\*

## supp 1 205.35 205.35 15.572 0.0002312 \*\*\*

## factor(dose):supp 2 108.32 54.16 4.107 0.0218603 \*

## Residuals 54 712.11 13.19

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Saime üsna ootuspärase tulemuse – doosi suurusest tulenev odontoblastide keskmise pikkuse kõver on erinevate manustamisiviiside korral erineva kujuga (koosmõju on statistiliselt oluline).

Mida saame kahefaktorilise ANOVA korral välja lugeda funktsiooni summary väljundist? Nagu ikka saame ülevaate jääkide jaotusest tervikuna (kui juba selles jaotuses midagi imelikku on, siis on kindel, et mudel ei ole adekvaatne). Lisaks näeme tervikmudeli olulisuse testi tulemust (p<0.001) ja determinatsioonikordajat (0.79). Leiame ka mudeli mudeli jääkide standardvea hinnangu (3.6). Mudeli parameetrite olulisuse testidest on juba tunduvalt keerulisem asjakohast infot välja lugeda (vaid kahe kahetasemelise faktoriga mudeli korral on need parameetrite testid sisuliselt samaväärsed funktsiooni anova väljundiga).

Tasemete keskmised saab siiski kergesti leida. Antud mudeli korral on baastasemeteks vastavalt doos 0.5 ja apelsinimahl manustamisviisina. Selles grupis on odontoblastide keskmine pikkus 13.2 mm. Kui soovime leida keskmist pikkust doosi 2.0 korral, kusjuures manustamisviisiks on askorbiinhape, siis arvutame lihtsalt 13.2+12.8-5.3+5.3=26.0 ehk siis kaasame arvutusse vabaliikme, vastavad peamõjude liikmed ja vastava koosmõju liikme.

summary(m1)

## Call:

## lm(formula = len ~ factor(dose) \* supp, data = ToothGrowth)

## Residuals:

## Min 1Q Median 3Q Max

## -8.20 -2.72 -0.27 2.65 8.27

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) 13.230 1.148 11.521 3.60e-16 \*\*\*

## factor(dose)1 9.470 1.624 5.831 3.18e-07 \*\*\*

## factor(dose)2 12.830 1.624 7.900 1.43e-10 \*\*\*

## suppVC -5.250 1.624 -3.233 0.00209 \*\*

## factor(dose)1:suppVC -0.680 2.297 -0.296 0.76831

## factor(dose)2:suppVC 5.330 2.297 2.321 0.02411 \*

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

## Residual standard error: 3.631 on 54 degrees of freedom

## Multiple R-squared: 0.7937, Adjusted R-squared: 0.7746

## F-statistic: 41.56 on 5 and 54 DF, p-value: < 2.2e-16

Vaatleme nüüd ka üht teist andmestikku. See pärineb Ri paketist „MASS“ ja kuigi antud pakett ei vaja eraldi installeerimist tuleb see alati peale Ri käivitamist tööle laadida.

library(MASS)

head(cabbages)

## Cult Date HeadWt VitC

## 1 c39 d16 2.5 51

## 2 c39 d16 2.2 55

## 3 c39 d16 3.1 45

## 4 c39 d16 4.3 42

## 5 c39 d16 2.5 53

## 6 c39 d16 4.3 50

Andmestikus on kahe eri kapsakultivaari kapsapeade andmed (pea kaal ja C-vitamiini sisaldus). Mõlemat kultivaari on istutatud kolmel eri kuupäeval. Teeme koosmõjuga mudeli C-vitamiini sisaldusele.

m2 <- lm(VitC~Cult\*Date, data=cabbages)

anova(m2)

## Analysis of Variance Table

## Response: VitC

## Df Sum Sq Mean Sq F value Pr(>F)

## Cult 1 2496.2 2496.15 54.1095 1.089e-09 \*\*\*

## Date 2 909.3 454.65 9.8555 0.0002245 \*\*\*

## Cult:Date 2 144.3 72.15 1.5640 0.2186275

## Residuals 54 2491.1 46.13

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Näeme, et antud juhul ei osutunud koosmõju oluliseks.

Saamaks kiiret visuaalset ülevaadet olukorrast võime kasutada karpdiagrammi abi (seal näeme gruppide mediaane ja varieeruvust). Alternatiivselt võime kasutada funktsiooni interaction.plot, mis ei tunne küll kahjuks valemisüntaksit aga võimaldab kiiret ülevaadet gruppide keskmistest. Paneme kaks graafikut kõrvuti. Seejuures pöörame karpdiagrammi telgede tähistuse telgedega risti ja keelame (ruumi huvides) horisontaaltelje nime kujutamise.

op <- par(mfrow=c(1,2))

boxplot(VitC~Cult+Date, data=cabbages, las=2, xlab= "")

interaction.plot(cabbages$Date, cabbages$Cult, cabbages$VitC)

par(op)



Joonistelt on selge, et koosmõju ei osutunud statistiliselt oluliseks, sest grupisisene varieeruvus on küllaltki suur, samas kui igas grupis on vaid 10 vaatlust.

Kuivõrd peamõjusid me üldjuhul olulise koosmõju korral testida ei soovigi siis väärib veel mainimist, et antud mudeli korral talitaksime edasi järgnevalt. Esmalt jätaksime mudelist koosmõju välja.

m2a <- update(m2, .~.-Cult:Date)

Funktsioon update võimaldab mugavalt mudelit uuendada – esialgne mudel võetakse esimest argumendist ja teise argumendi abil näidatakse, kuidas esialgset mudelit muuta. Selles süntaksis tähendab punkt kõike seda, mis oli seal esialgses mudelis. Miinuse või plussi abil osutame, mida eemaldada või lisada. Niisiis ütleme antud käsuga, et uus mudel peaks olema sama sõltuva tunnusega ning sõltumatute tunnuste osa jääb muidu samaks, ent koosmõju jäetakse välja. Seejärel saame juba testida peamõjude statistilist olulisust (antud juhul on mõlemal mõju olemas).

anova(m2a)

## Analysis of Variance Table

## Response: VitC

## Df Sum Sq Mean Sq F value Pr(>F)

## Cult 1 2496.2 2496.15 53.0411 1.179e-09 \*\*\*

## Date 2 909.3 454.65 9.6609 0.0002486 \*\*\*

## Residuals 56 2635.4 47.06

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

* 1. **Kovariaadi kaasamisega mudel**

Üritame nüüd kapsapea kaalu abil C-vitamiini kogust seletada. Ehk on nt nii, et suuremates peades ongi lihtsalt seda vitamiini vähem ning kultivaaridel on erinev keskmine peade kaal (ehk siis et kultivaaride mõju on pigem näiline). Kuna soovime jääda kahe sõltumatu tunnuse juurde siis keskendume vaid ühele istutuskuupäevale.

m3 <- lm(VitC~HeadWt\*Cult, subset=(Date=="d21"), data=cabbages)

anova(m3)

## Analysis of Variance Table

## Response: VitC

## Df Sum Sq Mean Sq F value Pr(>F)

## HeadWt 1 1523.43 1523.43 45.7302 4.562e-06 \*\*\*

## Cult 1 239.23 239.23 7.1810 0.01644 \*

## HeadWt:Cult 1 10.52 10.52 0.3159 0.58186

## Residuals 16 533.02 33.31

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Antud mudelis on siis mõlemal kultivaaril otsekui oma vitamiini ja pea kaalu vaheline lineaarne seos (ehk mõlemal oma vabaliige ja tõus). Koosmõju mitteolulisus viitab, et tõus võiks olla ühine. Uurime lähemalt

plot(VitC~HeadWt, col=as.numeric(Cult), data=cabbages, subset=(Date=="d21"))



Kasutasime joonistamisel eri kultivaaride erinevaks värvimiseks, et värve võib Ris ette anda ka numbrite abil (nt vaikimisi on 1 must ja 2 punane). Kuivõrd faktortunnuse numbriliseks muutes saamegi just sellised väärtused (esimesest faktori tasemest saab väärtus 1, teisest 2 jne) siis ongi kuvatav graafik just selline. Näeme, et varieeruvus pea kaaludes on kultivaaride vahel on tõepoolest suur. On raske öelda, kas sirgete tõusud (antud juhul langused) peaksid olema erinevad või mitte, ent kuivõrd valimimahud on väga väikesed siis ei ole koosmõju ebaoluliseks osutumine üllatav.

m3a <- update(m3, .~.-HeadWt:Cult)

anova(m3a)

## Analysis of Variance Table

## Response: VitC

## Df Sum Sq Mean Sq F value Pr(>F)

## HeadWt 1 1523.43 1523.43 47.6475 2.558e-06 \*\*\*

## Cult 1 239.23 239.23 7.4821 0.01409 \*

## Residuals 17 543.54 31.97

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Näeme, et antud juhul jääb kultivaar oluliseks ka pea kaalu arvestamisel. Ent nagu ka graafikult nägime on pea kaalu mõju C-vitamiini sisaldusele vägagi arvestatav.

* 1. **Marginaalkeskmiste hinnang (*LS-means* või *EMMs*)**

Antud olulisel võttel puudub kahjuks hea ja üldtuntud eestikeelne vaste. Põhimõte on aga väga lihtne – soovime faktori mõju suurust hinnata olukorras, kus ülejäänud tunnused on mitte ainult arvesse võetud vaid ka nö tasemete vahel võrdseks loetud.

Meie näites soovime leida kultivaari mõju olukorras, kus pea kaal on mõlema kultivaari korral võrdne. Analoogiliselt võime nt püüda sama liiki kalu erinevates püügikohtades ning hinnata kala keskmist kaalu, ent ootuspäraselt erineb meie valimisse sattuv sooline koosseis püügikohati. Meie sooviks võib olla soo mõju püügikohtade võrdlusest eemaldada.

Paketi emmeans sama nimega funktsiooni abil saame selliseid hinnanguid kergesti leida. Funktsioonile anname esimeseks argumendiks mudeli ning teiseks valemi, mis näitab, millise tunnuse mõju eemaldada soovime (vasakule poole püstkriipsu faktortunnus, mille mõju huvi pakub ning paremale poole püstkriipsu faktortunnus või kovariaat, mille mõju meid segab).

library(emmeans)

emmeans(m3a, ~Cult|HeadWt)

## HeadWt = 2.1:

## Cult emmean SE df lower.CL upper.CL

## c39 58.6 2.15 17 54.0 63.1

## c52 68.0 2.15 17 63.5 72.6

## Confidence level used: 0.95

Väljundist leiame nii marginaalkeskmiste hinnangud kui ka nende standardvead ja usalduspiirid.

1. **Kolme sõltumatu tunnusega mudel**
	1. **Kolmefaktoriline ANOVA**

Vaatleme näitena tähk-kukehirsi (*Echinochloa crus-galli*) süsihappegaasi sidumisvõimet käsitlevat Ri näidis andmestikku nimega CO2. Taimed pärinesid kahest erinevast kasvukohast. Osadele taimedele tekitati mõõtmiseelselt külmashokk. Lisaks varieeriti ka ümbritseva keskkonna süsihappegaasi taset.

Kuivõrd needki andmed sisaldavad kordusmõõtmisi, siis peame siingi tavalise ANOVA kasutamiseks osa andmetest analüüsist välja jätma.

head(CO2, 10)

 Plant Type Treatment conc uptake

1 Qn1 Quebec nonchilled 95 16.0

2 Qn1 Quebec nonchilled 175 30.4

3 Qn1 Quebec nonchilled 250 34.8

4 Qn1 Quebec nonchilled 350 37.2

5 Qn1 Quebec nonchilled 500 35.3

6 Qn1 Quebec nonchilled 675 39.2

7 Qn1 Quebec nonchilled 1000 39.7

8 Qn2 Quebec nonchilled 95 13.6

9 Qn2 Quebec nonchilled 175 27.3

10 Qn2 Quebec nonchilled 250 37.1

Näeme, et iga taime kohta on tehtud 7 mõõtmist. Kui me valime võrdluseks keskkonna süsihappegaasi taseme vähima ja suurima väärtuse (vastavalt 95 ja 1000), siis saame meile sobiva alamvalimi kaasates igast 14-realisest plokist (kokku on 6 sellist plokki) esimese ja viimase vaatluse (nii kasutame iga taime kohta vaid üht vaatlust).

abi <- CO2[rep(c(1,14), each=6)+rep(14\*(0:5), 2), ]

Meie andmestikku jääb nüüd vaid 12 vaatlust, kusjuures mõnele faktori tasemete kombinatsioonile vastab vaid üks vaatlus. Kas sellises olukorras on üldse võimalik mudelit sobitada? Ilma probleemideta, sest me eeldame tavalise ANOVA korral, et igal faktori tasemete kombinatsioonil on jääkide jaotus sama dispersiooniga. Muidugi ei saa me seda eeldust ühe vaatlusega kombinatsioonidel kuidagi kontrollida, ent me saame seda teha ülejäänud kombinatsioonide jääkide põhjal.

m1 <- lm(uptake~Treatment\*Type\*factor(conc), data=abi)

anova(m1)

## Analysis of Variance Table

## Response: uptake

## Df Sum Sq Mean Sq F value Pr(>F)

## Treatment 1 6.75 6.75 4.5685 0.0993764

## Type 1 330.75 330.75 223.8579 0.0001162

## factor(conc) 1 1432.21 1432.21 969.3503 6.342e-06

## Treatment:Type 1 23.52 23.52 15.9188 0.0162672

## Treatment:factor(conc) 1 3.84 3.84 2.5990 0.1822259

## Type:factor(conc) 1 105.84 105.84 71.6345 0.0010679

## Treatment:Type:factor(conc) 1 16.34 16.34 11.0558 0.0292408

## Residuals 4 5.91 1.48

Näeme, et kolmene koosmõju on statistiliselt oluline. Vaatleme nüüd alternatiivset mudelit, kus lisaks peamõjudele on kaasatud vaid kasvukoha ja ümbritseva keskkonna süsihappegaasi koosmõju, ülejäänud koosmõjusid aga kaasatud ei ole.

m2 <- lm(uptake~Treatment+factor(conc)\*Type, data=abi)

anova(m2)

## Analysis of Variance Table

## Response: uptake

## Df Sum Sq Mean Sq F value Pr(>F)

## Treatment 1 6.75 6.75 1.725 0.2304636

## factor(conc) 1 1432.21 1432.21 366.006 2.654e-07 \*\*\*

## Type 1 330.75 330.75 84.524 3.713e-05 \*\*\*

## factor(conc):Type 1 128.05 128.05 32.724 0.0007199 \*\*\*

## Residuals 7 27.39 3.91

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Kuivõrd meie valim on väga väike, siis oleks täiesti normaalne mitte proovidagi maksimaalset võimalikku mudelit. Kuidas on aga kahe sobitatud mudeli vahekord ehk kumb on siis mõistlikum mudel? Kuivõrd teine mudel on esimese mudeli kitsendus ja mõlemad on ANOVA mudelid siis saame antud juhul neid F-testi abil võrrelda (me testime niisiis mitme mudeli liikme ühist statistilist olulisust korraga).

anova(m2, m1)

## Analysis of Variance Table

## Model 1: uptake ~ Treatment + factor(conc) \* Type

## Model 2: uptake ~ Treatment \* Type \* factor(conc)

## Res.Df RSS Df Sum of Sq F Pr(>F)

## 1 7 27.392

## 2 4 5.910 3 21.482 4.8464 0.08073 .

Test ütleb niisiis, et kolm lisandunud mudeli liiget ei lisa mudelile piisavalt seletusvõimet, et nende lisamine oleks statistiliselt põhjendatud. Kas tegu pole mitte vastuolulise tulemusega kui arvestame, et eelnevalt nägime kolmese interaktsiooni statistilist olulisust? Vastuolu siiski ei ole, sest meie küsitud küsimus ei olnud täpselt sama. Siin on taaskord tegu mitmese testimisega probleemiga (aga seda tüüpi probleemist väga sageli ei räägita) – ANOVA kaitseb meid küll olukorras, kus mõnel faktoril on palju tasemeid, ent ei kaitse meid kuidagi olukorras, kus meil on palju faktoreid (sarnane on olukord ka teiste statistiliste mudelite korral, mitte ainult ANOVA mudelite puhul). Tegelikult isegi olukorras, kus meil on ANOVA mudel, milles vaid kaks faktorit ja nende koosmõju, on tõenäosus, et leiame olulise pea- või koosmõju olukorras, kus tegelikult ühtegi olulist mõju ei eksisteeri, 0.14 ehk 14%.

Kuidas siis ikkagi käituda meie olukorras? Antud valimimahu korral on kolmese koosmõju testimine kindlasti ülepakkumine. Võimalik, et see kolmene koosmõju on tegelikult olemas, ent meil ei ole piisavalt andmeid, et seda usaldusväärselt hinnata.

Vaatleme ka viimati sobitatud mudeli eelduste täidetust isegi kui väikese valimi korral on tüüpiliselt eelduste täidetuse rikutust väga keeruline tuvastada (kaasa arvatud olukorras, kus eeldused tegelikult märgatavalt rikutud on). Üheks võimaluseks on siis kõrvutada mudeli põhjal joonistatud graafikuid simuleeritud andmete põhjal joonistatud graafikutega.

Joonistame siis jääkide histogrammi ja lisame sellele juurde ka 11 histogrammi, mis joonistatud samamoodi 12 väärtuse põhjal, ent need väärtused ongi standardsest normaaljaotusest simuleeritud (Ris funktsioon rnorm).

op <- par(mfrow=c(4,3))

hist(resid(m2), xlim=c(-3,3), breaks=-3:3)

for (i in 1:11){

hist(rnorm(12), xlim=c(-3,3), breaks=-3:3)

}

par(op)



Näeme, et jääkide varieeruvusega ei ole ehk asjad päris korras – absoluutväärtuselt suuri jääke on liiga palju. Kuivõrd valimimaht on väike siis on igati ootuspärane, et jääkide dispersiooni hinnang ei pruugi olla kuigi täpne (hajuvuse (nt dispersioon) hindamine on palju raskem ülesanne kui paiknemise (nt keskmine) hindamine ja vajab seetõttu suuremat valimimahtu, et seda täpselt teha).

Vaatame nüüd jääkide ja sobitatud väärtuste hajuvusdiagrammi ja ka antud graafiku teisendatud kuju

plot(m2, c(1,3))





Tundub, et suured jäägid ei paikne süstemaatiliselt.

* 1. **Keskmiste mitmene võrdlus**

ANOVAst kui protseduurist, mille abil saame end kaitsta mitmese testimisega seotud valepositiivsete probleemi eest olukorras, kus faktoril on palju tasemeid oli juba juttu. Kui avastame mõne (enam kui kahe tasemega) faktori olulise mõju siis kindlasti tekib ka küsimus, milles see mõju väljendub (ehk millistel faktori tasemetel on uuritava tunnuse keskmine eristuv). Analoogiline küsimus võib tekkida mistahes faktorite kombinatsioonide korral. Eelnevalt vaadeldud mudeli korral nt „kas madal süsihappegaasi kontsentratsioon koos asukoha mõjuga võib nullida külmatöötluse mõju ehk kas “külmashokki mitte saanud Quebeci taim seob madala süsihappegaasi kontsentratsiooni juures rohkem süsihappegaasi kui külmashoki saanud Missisipi taim kõrge süsihappegaasi kontsentratsiooni juures.

Vaatleme esmalt vastavaid keskmisi koos usalduspiiridega.

emmeans(m2, ~factor(conc)+Type+Treatment)

## conc Type Treatment emmean SE df lower.CL upper.CL

## 95 Quebec nonchilled 15.91 1.21 7 13.04 18.77

## 1000 Quebec nonchilled 45.62 1.40 7 42.31 48.92

## 95 Mississippi nonchilled 11.94 1.21 7 9.08 14.81

## 1000 Mississippi nonchilled 28.58 1.40 7 25.28 31.89

## 95 Quebec chilled 9.68 1.40 7 6.38 12.99

## 1000 Quebec chilled 39.39 1.21 7 36.53 42.26

## 95 Mississippi chilled 5.72 1.40 7 2.41 9.02

## 1000 Mississippi chilled 22.36 1.21 7 19.49 25.22

Meid huvitavad tasemete kombinatsioonid on saadud tabeli esimeses ja viimases reas. Usalduspiirid küll ei kattu, ent sellest ei piisa veel erinevuse statistilise olulisuse kontrolliks. Käsu

pairs(emmeans(m2, ~Treatment+Type+factor(conc)))

abil sooritatakse kõikvõimalikud paariviisilised võrdlused ning vaikimisi tehakse Tukey mitmese võrdlemise korrektsioon. Meid huvitava võrdlus on väljundis seitsmendal real

## contrast estimate SE df t.ratio p.value

## nonchilled Quebec 95 - chilled Mississippi 1000 -6.45 1.81 7 -3.572 0.0936

Näeme, et erinevus ei osutu pärast mitmese võrdluse parandust statistiliselt oluliseks. Kui antud võrdlus oleks meil juba varasemalt ainsana plaanis siis ei peaks me muidugi mitmeste võrdluse korrektsiooni tegema

pairs(emmeans(m2, ~Treatment+Type+factor(conc)), adjust= "none")

## contrast estimate SE df t.ratio p.value

## nonchilled Quebec 95 - chilled Mississippi 1000 -6.45 1.81 7 -3.572 0.0091

ent sellisel juhul oleksime ilmselt tekitanud oma analüüsi ka ainult ühe kahe tasemega faktori ja oleksime otsitava tulemuse leidnud juba otse mudeli väljundist.

Vahel võime soovida läbi viia vaid osasid paariviisilisi võrdlusi. Nt võime soovida kõikvõimalikke võrdlusi kasvukoha siseselt.

pairs(emmeans(m2, ~Treatment+factor(conc)|Type))

## Type = Quebec:

## contrast estimate SE df t.ratio p.value

## nonchilled 95 - chilled 95 6.22 1.21 7 5.139 0.0057

## nonchilled 95 - nonchilled 1000 -29.71 1.66 7 -17.844 <.0001

## nonchilled 95 - chilled 1000 -23.48 1.81 7 -13.004 <.0001

## chilled 95 - nonchilled 1000 -35.93 2.28 7 -15.731 <.0001

## chilled 95 - chilled 1000 -29.71 1.66 7 -17.844 <.0001

## nonchilled 1000 - chilled 1000 6.22 1.21 7 5.139 0.0057

## Type = Mississippi:

## contrast estimate SE df t.ratio p.value

## nonchilled 95 - chilled 95 6.22 1.21 7 5.139 0.0057

## nonchilled 95 - nonchilled 1000 -16.64 1.66 7 -9.996 0.0001

## nonchilled 95 - chilled 1000 -10.42 1.81 7 -5.768 0.0029

## chilled 95 - nonchilled 1000 -22.87 2.28 7 -10.011 0.0001

## chilled 95 - chilled 1000 -16.64 1.66 7 -9.996 0.0001

## nonchilled 1000 - chilled 1000 6.22 1.21 7 5.139 0.0057

## P value adjustment: tukey method for comparing a family of 4 estimates

Sel juhul tehakse mitmese võrdluse parandus samuti kasvukoha siseselt.

1. **Sõltumatute tunnuste testimise järjekord. Alternatiivsed mudelid I**
	1. **Sõltumatute tunnuste testimise järjekord**

Uurime esmalt üht keerulisemat kahefaktorilist dispersioonanalüüsi näidet. Vaatleme tibude kasvuandmestikku ChickWeight, kus vaadeldud tibude kehakaalu erinevate söötade korral.

head(ChickWeight)

## weight Time Chick Diet

## 1 42 0 1 1

## 2 51 2 1 1

## 3 59 4 1 1

## 4 64 6 1 1

## 5 76 8 1 1

## 6 93 10 1 1

summary(ChickWeight)

## weight Time Chick Diet

## Min. : 35.0 Min. : 0.00 13 : 12 1:220

## 1st Qu.: 63.0 1st Qu.: 4.00 9 : 12 2:120

## Median :103.0 Median :10.00 20 : 12 3:120

## Mean :121.8 Mean :10.72 10 : 12 4:118

## 3rd Qu.:163.8 3rd Qu.:16.00 17 : 12

## Max. :373.0 Max. :21.00 19 : 12

## (Other):506

Näeme, et tegu on kordusmõõtmistega (iga tibu kohta 12 andmepunkti ajahetkedel 0 kuni 21). Söötu on kokku 4 erinevat ja valim ei ole tasakaaluline – esimene sööt on rohkem esindatud. Valemisüntaksit tundev funktsioon xtabs võimaldab meil mugavalt sagedustabeleid koostada – siis näeme kohe, kuidas vaatlused jaotuvad. Kuivõrd sõltumatuteks tunnusteks on plaanis võtta aeg ja sööt (ehk küsida, kas eri söödad mõjuvad kaalule ühtemoodi), siis vaatleme jaotumist just nende tunnuste alusel.

xtabs(~Diet+Time, data=ChickWeight)

## Time

## Diet 0 2 4 6 8 10 12 14 16 18 20 21

## 1 20 20 19 19 19 19 19 18 17 17 17 16

## 2 10 10 10 10 10 10 10 10 10 10 10 10

## 3 10 10 10 10 10 10 10 10 10 10 10 10

## 4 10 10 10 10 10 10 10 10 10 10 9 9

Nüüd näeme kohe, et esimesel sööda tasemel alustas 20 tibu aga erinevatel põhjustel oli viimasel mõõtmispäeval alles veel vaid 16 tibu. Analoogiliselt oli neljandal sööda tasemel alguses 10 tibu, lõpus aga 9.

Vaatleme ajahetki 0 ja 10 ning eraldame vastava alamandmestiku. Selleks kasutame funktsiooni subset ja anname ette tingimused, mida alamandmestikku valitavad kirjed peavad rahuldama.

andmed <- subset(ChickWeight, (Time==0)|(Time==10))

Antud juhul ütleme, et sobivad kõik need andmeread, kus aja väärtus on 0 või 10. Loogilise tingimuse „või“ sümboliks on Ris püstkriips | ja loogilise tingimuse „ja“ sümboliks on märk &, selguse huvides paneme tingimuse kõik osad eraldi sulgudesse.

Kuivõrd tegu on kordusmõõtmistega siis oleks vale kasutada ühe tibu andmeid mitu korda ilma, et me kasutaksime vastavat statistilist mudelit. Paariviisilisest t-testist jääb siin aga väheseks, sest sinna ei õnnestu kaasata sööda mõju ja siis tuleks teha eraldi mudel iga sööda kohta. See variant meile ei sobi (siis ei saa me sööda mõju võrrelda).

Oleme siis kavalad. Kuivõrd tibude on andmestikus sööda alusel järjestatud siis valime ajahetkel 0 välja need tibud, kelle identifikaatortunnus on lõpuga 0, 1, 2, 3 või 4. Ajahetkel 10 vaatame aga neid tibusid, kelle identifikaatortunnus on lõpuga 5, 6, 7, 8 või 9. Nii ei kasuta me kordusmõõtmisi ja ei riku ANOVA eeldusi (aga muidugi kaotame nii valimimahus ja seega ka analüüsi võimsuses). Seda saame lihtsasti sõnastada tingimusena, kasutades jääki, mis tekib identifikaatortunnuse kümnega jagamisel (Ris annab selle funktsioon %%). Kuivõrd identifikaatortunnus ei ole meie andmestikus numbriline siis tuleb seda esmalt numbriliseks teisendada.

andmed$Chick <- as.numeric(as.character(andmed$Chick))

Seejuures „factor“ tüüpi tunnuse numbriliseks teisendamisel on otstarbekas talitada just nii. Vastasel juhul teisendab R tunnused numbriliseks vastavalt faktori tasemete järjekorrale, mis ei pruugi aga ühtida (ja antud näidisandmestiku korral ei ühtigi) nende numbrilise järjekorraga. Niisiis

andmed2 <- subset(andmed, ((Time==0)&(Chick%%10<5))|((Time==10)&(Chick%%10>=5)))

Nüüd oleme lõpuks analüüsiks valmis.

m1 <- lm(weight~factor(Time)\*Diet, data=andmed2)

anova(m1)

## Analysis of Variance Table

## Response: weight

## Df Sum Sq Mean Sq F value Pr(>F)

## factor(Time) 1 51830 51830 276.7808 < 2e-16 \*\*\*

## Diet 3 2053 684 3.6553 0.02004 \*

## factor(Time):Diet 3 2187 729 3.8929 0.01547 \*

## Residuals 41 7678 187

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Saime üsna ootuspärase tulemuse – keskmine kaalu kasv on eri söötade korral olnud erinev (koosmõju on statistiliselt oluline). Kas me ei oleks võinud lihtsalt loobuda kaalu algväärtustest ja teha ühefaktorilise ANOVA söödatunnusega? Seda võinuksime teha juhul, kui olnuksime kindlad, et kaalukeskmised alghetkel sööda eri gruppides ei erine.

Nagu teame on mitme sõltumatu tunnusega mudeli mõte võtta samaaegselt arvesse mitme sõltumatu tunnuse mõju. Kui meil on mudelis 2 sõltumatut kategoorilist tunnust ja valim ei ole tasakaaluline (enamasti ei ole) siis on tegelikult mitu erinevat võimalust, kuidas tunnuste olulisust testida. Mäletatavasti on tunnuse olulisuse testimine samaväärne kahe mudeli võrdlemisega. Oletame, et mudelis on lisaks ka kahe tunnuse koosmõju.

Esimene viis on järjestikune võrdlus (ja vahel nimetatakse seda I tüüpi testimiseks). Meie mudeli näitel on tunnuse aeg olulisus saadud ainult vabaliikmega mudeli ja mudeli, kus lisaks ka aeg võrdlemisel. Tunnuse sööt olulisus on saadud mudeli, kus on vabaliige ja aeg võrdlemisel mudeliga, kus lisaks ka sööt. Koosmõju olulisus on saadud mudeli, kus on vabaliige, aeg ja sööt võrdlemisel mudeliga, kus on lisaks ka interaktsioon. Sellist (järjekorda arvestavat) testimist kasutab R vaikimisi. Arusaadavalt on sellises olukorras oluline, kumb faktor (kas aeg või sööt) on mudelis esimesel ja kumb teisel kohal.

Enamasti ei ole aga loogilist sõltumatute muutujate järjekorda ning pigem oleksime huvitatud testima mõlema tunnuse olulisust nii, et teine tunnus on juba arvesse võetud (seda nimetatakse ka II tüüpi testimiseks). See tähendab, et tunnuse sööt olulisuse saaksime mudeli, kus on vabaliige ja aeg võrdlemisel mudeliga, kus on lisaks ka sööt. Tunnuse aeg olulisuse saaksime aga mudeli, kus on vabaliige ja sööt võrdlemisel mudeliga, kus on lisaks ka aeg. Koosmõju testimine on analoogiline I tüüpi testimisega.

Lisaks on olemas veel üks võimalus (mida nimetatakse ka III tüüpi testimiseks). Siin võetakse tunnuse peamõju testimisel samuti arvesse teise tunnuse mõju, ent lisaks võetakse arvesse ka koosmõju. See ei ole eriti loogiline, sest sisuliselt tähendaks see, et me otsekui võrdleksime täismudelit mudeliga, kus on üks sõltumatu tunnus ja selle ning teise sõltumatu tunnuse interaktsioon (aga kahe sõltumatu faktortunnuse korral ei ole selline asi mõeldav). Koosmõju testimine on analoogiline eelnevate variantidega.

Kui mudelis koosmõju ei olegi siis on II ja III tüüpi testimine samaväärsed ning sel juhul on testimist mugav läbi viia funktsiooni drop1 abil (argumentideks tuleb anda mudel ise ning test="F"). Kuigi II tüüpi testimist saab nö käsitsi läbi viia (mudeli defineerimisel sõltumatute tunnuste järjekorda muutes), siis mugavam on seda teha Ri paketi car funktsiooni Anova abil (NB! Ongi suur täht funktsiooninime alguses). See funktsioon võimaldab ka III tüüpi teste, valides argumendiks type=3.

Selle jutuosa lõpuks märgime veel, et kuivõrd antud andmestik on tasakaaluline siis siin langevad I ja II tüüpi testimise tulemused kokku.

Edasi vaatleme, mida teha olukorras, kus kahtlustame, et pidevate tunnuste vaheline seos ei ole lineaarne. Põhimõtteliselt on meil siis kaks võimalust – kasutada mittelineaarset mudelit või kasutada lineaarset mudelit ja teisendada tunnuseid.

* 1. **Sõltumatu muutuja mittelineaarne teisendus**

Tuleme tagasi puude andmestiku juurde trees. Oletame esmalt, et puude kõrgus ei ole teada ning proovime puu ruumala mudeldada ainult läbimõõdu abil. Proovime kahte kandidaati.

m1 <- lm(Volume~Girth, data=trees)

m2 <- lm(Volume~Girth+I(Girth\*\*2), data=trees)

anova(m1,m2)

## Analysis of Variance Table

## Model 1: Volume ~ Girth

## Model 2: Volume ~ Girth + I(Girth^2)

## Res.Df RSS Df Sum of Sq F Pr(>F)

## 1 29 524.30

## 2 28 311.38 1 212.92 19.146 0.0001524 \*\*\*

Näeme, et ruutliige on mudelis oluline. Kui kasutame vaid ruutliiget, siis on võit reeglina väike, ent suuremate astmete korral on mõistlik mudeli defineerimisel kasutada abifunktsiooni poly, mis vähendab kirjutamisvaeva, aga lisaks kasutab ortogonaalseid polünoome, mis vähendavad tunduvalt võimalikke arvutuslikke probleeme mudeli sobitamisel (probleemid võivad tuleneda sellest, et tunnuse kõrgema astme funktsioonid on üksteisega väga tugevalt korreleeritud). Alternatiivselt seega

m2a <- lm(Volume~poly(Girth, 2), data=trees)

## summary(m2a)

## Call:

## lm(formula = Volume ~ poly(Girth, 2), data = trees)

## Residuals:

## Min 1Q Median 3Q Max

## -5.4889 -2.4293 -0.3718 2.0764 7.6447

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) 30.1710 0.5989 50.374 < 2e-16 \*\*\*

## poly(Girth, 2)1 87.0734 3.3348 26.111 < 2e-16 \*\*\*

## poly(Girth, 2)2 14.5918 3.3348 4.376 0.000152 \*\*\*

## Residual standard error: 3.335 on 28 degrees of freedom

## Multiple R-squared: 0.9616, Adjusted R-squared: 0.9588

## F-statistic: 350.5 on 2 and 28 DF, p-value: < 2.2e-16

ning näeme, et ruutliikme p-väärtus on muutumatu sõltumata kirjapildist. Madalama astmega liikmeid (antud juhul siis nt läbimõõdu lineaarliiget) kõrgema astme statistilise olulisuse korral mudelist ei eemaldata.

* 1. **Mittelineaarne mudel**

Mittelineaarsete mudelite jaoks on tegelikult Ris olemas võimas pakett nimega nlme, ent meie vaatleme järgnevalt lihtsat mittelineaarse funktsiooni sobitamist funktsiooni nls abil. Tuleme tagasi andmestiku CO2 juurde. Olgu meile teooriast teada, et CO2 sidumisvõimet peaks hästi seletama eksponentfunktsioon kujul

sidumisvõime = a \* (1 - exp(-exp(b) \* CO2)),

kus a ja b on mudeli parameetrid, mida hinnata soovime ning CO2 on ümbritseva keskkonna süsihappegaasi tase. Näeme, et seda seost ei ole võimalik logaritmimise abil kuidagi lineaarseks muuta. Ei aita ka sõltumatu muutuja teisendamine.

plot(uptake~conc, data=CO2, col=as.numeric(Plant), pch=19)



Sobitatava funktsioonikuju kohta on teada, et tegu on nullpunkti läbiva positiivse funktsiooniga, mis nö jõuab platoole (ehk ligineb asümptoodile). Parameeter a vastab platoo väärtusele ja parameeter b on võrdne log(log(2)/y), kus y on sõltumatu tunnuse väärtus, mille korral on saavutatud pool platoo kõrgusest. Kuivõrd funktsioon nls vajab sisendina ka parameetrite algväärtusi (tegelike parameetrite hinnangutest mitte liiga kaugel paiknevaid väärtusi) siis saame neid teadmisi edasises ära kasutada.

Jooniselt on näha, et platoo on taimeti selgelt erinev omades väärtusi umbes 15 ja 45 vahel. Pool platoo väärtusest saavutatakse umbes argumendi 100 juures. Seega kui loome mudeli, kus igal taimel on oma platoo, siis sobiks parameetri a algväärtuseks nt 30. Eeldame, et parameeter b on kõigil taimedel sama ja selle algväärtuseks sobib siis log(log(2)/100), mis on umbes -5.

m3 <- nls(uptake ~ a[Plant]\*(1-exp(-exp(b)\*conc)), data=CO2, start=list(a=rep(30, 12), b=-5))

summary(m3)

## Formula: uptake ~ a[Plant] \* (1 - exp(-exp(b) \* conc))

## Parameters:

## Estimate Std. Error t value Pr(>|t|)

## a1 40.66486 1.25082 32.51 <2e-16 \*\*\*

## a2 43.64253 1.26211 34.58 <2e-16 \*\*\*

## a3 46.32442 1.27287 36.39 <2e-16 \*\*\*

## a4 36.95268 1.23774 29.86 <2e-16 \*\*\*

## a5 40.39978 1.24985 32.32 <2e-16 \*\*\*

## a6 40.81618 1.25138 32.62 <2e-16 \*\*\*

## a7 29.66088 1.21536 24.41 <2e-16 \*\*\*

## a8 33.66302 1.22708 27.43 <2e-16 \*\*\*

## a9 32.91569 1.22479 26.88 <2e-16 \*\*\*

## a10 14.66700 1.18405 12.39 <2e-16 \*\*\*

## a11 20.81517 1.19442 17.43 <2e-16 \*\*\*

## a12 21.98482 1.19678 18.37 <2e-16 \*\*\*

## b -5.09493 0.04813 -105.85 <2e-16 \*\*\*

## Residual standard error: 2.597 on 71 degrees of freedom

Vaikimisi sooritatav nulliga võrdlemise test, meid antud olukorras kindlasti ei huvita. Ligikaudse normaaljaotuse eelduse põhjal saame parameetri a hinnangute ja nende standardvigade põhjal koostada nt taimede 1 ja 3 platoo statistilise erinevuse testi (kuigi tegelikult oleks lisaks standardvigadele korrektne arvesse võtta ka hinnangute vaheline kovariatsioon (mis on funktsiooni vcov abil vahetult leitav)).

* 1. **II tüüpi regressioon**

II tüüpi regressioon ehk nn sümmeetriline regressioon on kasutusel olukorras, kus meil on kaks pidevat tunnust ja üht ei ole põhjust teisele eelistada (ehk ei saa konkreetselt öelda, et üks neist kahest on sõltuv ja teine sõltumatu muutuja). II tüüpi regressioon on kasutusel ka olukorras, kus teame, et mõlemad muutujad on mõõdetud ebatäpselt (selline olukord esineb muidugi praktiliselt alati). Peamiseks meetodi eelduseks on, et andmed peaksid (ligikaudu) pärinema kahemõõtmelisest normaaljaotusest (mis tähendab, et erindeid ei tohiks andmestikus olla).

Ris on teist tüüpi regressiooni jaoks pakett lmodel2. Sõltuvalt sellest, kas me eeldame, et mõlema tunnuse korral on mõõtmistäpsus umbes sama (see on siis nö vigade, mitte tunnuste enda varieeruvus) või seotud tunnuse enda varieeruvusega (ehk siis suurema varieeruvusega tunnusel on ka vigade varieeruvus suurem; nt kui tunnusteks on mingid liikide arvud eri ruumipunktides) on antud paketis vaja vaadata analüüsi MA (*major axis*) või SMA (*standard major axis* aga sageli tuntud ka kui *reduced major* axis). Kui eesmärgiks on tõusu hinnangute hilisem omavaheline võrdlemine (sooritatud nt 2 sarnast eksperimenti) siis on mõistlikum MA kasutamine. Antud paketis leiduv analüüs RMA (*ranged major axis*) on veel kolmas II tüüpi regressiooni võimalus, mida paketi autor soovitab SMA asemel kasutada.

Võrdleme esmalt C-vitamiini ja pea kaalu vahelisi sirge tõuse kahel kultivaaril. Kuivõrd lmodel2 ei võimalda alamandmestiku kasutamist siis peame eraldi andmestikud ise esmalt valmis tegema.

abi1 <- subset(cabbages, Cult=="c52")

abi2 <- subset(cabbages, Cult=="c39")

library(lmodel2)

lmodel2(VitC~HeadWt, data=abi1)

## RMA was not requested: it will not be computed.

## No permutation test will be performed

## Model II regression

## Call: lmodel2(formula = VitC ~ HeadWt, data = abi1)

## n = 30 r = -0.6731977 r-square = 0.4531951

## Parametric P-values: 2-tailed = 4.570085e-05 1-tailed = 2.285042e-05

## Angle between the two OLS regression lines = 4.65549 degrees

## Regression results

## Method Intercept Slope Angle (degrees) P-perm (1-tailed)

## 1 OLS 79.55426 -6.646605 -81.44387 NA

## 2 MA 97.65201 -14.584215 -86.07752 NA

## 3 SMA 86.91086 -9.873185 -84.21655 NA

## Confidence intervals

## Method 2.5%-Intercept 97.5%-Intercept 2.5%-Slope 97.5%-Slope

## 1 OLS 72.68506 86.42346 -9.472856 -3.820354

## 2 MA 87.71438 122.18449 -25.344074 -10.225604

## 3 SMA 81.37115 94.25885 -13.095987 -7.443485

## Eigenvalues: 71.82389 0.3991493

## H statistic used for computing C.I. of MA: 0.0008421345

Esimese kultivaari jaoks leiame MA realt tõusu usaldusvahemiku (-25.3; -10.2). Analoogiliselt saame teise kultivaari jaoks käsuga

lmodel2(VitC~HeadWt, data=abi2)

## 2 MA 81.66436 193.77162 -48.946659 -10.377645

ehk tõusu usaldusvahemikuks on (-48.9; -10.4) mistõttu näeme siitki, et tõusud ei tundu eri kultivaaride vahel erinevat.

Kui sooviksime aga kahe kultivaari ühist tõusu hinnangut (võttes arvesse mõõtmiste ebatäpsust), siis eeldaksime ilmselt, et tunnuste mõõtmistäpsus on kooskõlas tunnuste skaalaga (ja seetõttu on C-vitamiini vigade hajuvus märgatavalt suurem (ehk siis MA kasutamine ei oleks õigustatud)). RMA hinnangut vaikimisi ei leita, me peame selle eraldi tellima. Vastavalt sellele, kas antud tunnuse jaoks on nullväärtus tähenduslik või mitte tuleb määrata argumendid range.x ja range.y väärtusega "relative" või "interval".

lmodel2(VitC~HeadWt, data=cabbages, range.x="relative", range.y="relative")

## Method Intercept Slope Angle (degrees) P-perm (1-tailed)

## 4 RMA 81.44972 -9.061587 -83.70255 NA

Kui me soovime, et funktsioon väljastaks ka tõusu statistilise olulisuse testi p-väärtuse siis peame täiendavalt määrama ka argumendi nperm, andes sellele väärtuseks positiivse täisarvu (nii mitu permutatsiooni tehakse permutatsioonitestis).

1. **Alternatiivsed mudelid II**

Kui meie vaatlused ei ole sõltumatud siis peame seda ka mudeli loomisel arvesse võtma. Lihtsaim näide on juba varem vaadeldud paariviisiline t-test. Enamasti nii lihtsast mudelist siiski ei piisa.

* 1. **Hierarhiline mudel**

Esmalt vaatame aga olukorda, kus üks sõltumatu muutuja on teisele allutatud. See tähendab lihtsalt, et allutatud muutuja tasemed omavad tähendust vaid teise muutuja tasemete kontekstis. Sisuliselt tähendab see lihtsalt, et kahe muutuja vahel on interaktsioon (ehk sisuliselt sobitame lihtsalt koosmõjuga ANOVA mudeli). Ainus erinevus on siis, et allutatud muutuja nö peamõju ei ole võimalik eraldi testida (sest see oleks sisuliselt ebamõistlik) – me testime „allutatud tunnuse peamõju ja koosmõju üheskoos“.

Vaatleme näitena andmestikku cabbages (paketist MASS), uurime jätkuvalt C-vitamiini sisaldust kapsapeades ja oletame, et istutamiskuupäev on allutatud kultivaarile (me ei eelda seega, et kuupäeva mõju peaks eri kultivaaride korral olema samasugune).

Ris saab sellise mudeli kirja panna kahel samaväärsel viisil

m1 <- lm(VitC~Cult/Date, data=cabbages)

m1a <- lm(VitC~Cult+Cult:Date, data=cabbages)

Kuivõrd kultivaari peamõju testimisel ei tulekski koosmõju arvesse võtta siis antud juhul saame korrektsed (II tüüpi testimisel vastavad) olulisustõenäosused lihtsasti (lisaks on valim tasakaaluline ning I tüüpi testimistulemused langevad antud juhul kokku II tüüpi testimistulemustega)

anova(m1)

## Analysis of Variance Table

## Response: VitC

## Df Sum Sq Mean Sq F value Pr(>F)

## Cult 1 2496.2 2496.15 54.1095 1.089e-09 \*\*\*

## Cult:Date 4 1053.6 263.40 5.7098 0.0006597 \*\*\*

## Residuals 54 2491.1 46.13

Näeme, et istutamiskuupäeval on C-vitamiini sisaldusele statistiliselt oluline mõju (p<0.001), ent kuivõrd istutamiskuupäev on kultivaarile allutatud siis ei ütle see meile, kas istutuskuupäeva mõju on eri kultivaaride korral samasugune või erinev.

Kui peamõjusid olnuks teisigi (ja valim polnuks tasakaalus) siis saaksime II tüüpi testimist nö käsitsi läbi viia (mudeli defineerimisel sõltumatute tunnuste järjekorda muutes), ent mugavam oleks seda teha Ri paketi car funktsiooni Anova abil.

Analoogiliselt võib allutatud olla ka pidev tunnus. Oletame, et uurime jätkuvalt C-vitamiini sõltuvust ning pea kaal on kultivaarile allutatud. Olgu lisaks mudelis ka istutamiskuupäev (antud näites selle tunnuse allutatust ei eelda).

m2 <- lm(VitC~Date+Cult/HeadWt, data=cabbages)

anova(m2)

## Analysis of Variance Table

## Response: VitC

## Df Sum Sq Mean Sq F value Pr(>F)

## Date 2 909.30 454.65 12.2495 4.105e-05 \*\*\*

## Cult 1 2496.15 2496.15 67.2529 4.684e-11 \*\*\*

## Cult:HeadWt 2 631.14 315.57 8.5023 0.0006164 \*\*\*

## Residuals 54 2004.26 37.12

Näeme, et pea kaalul on mõju C-vitamiini sisaldusele (p<0.001), ent kuivõrd pea kaal on kultivaarile allutatud siis ei ütle see meile, kas see pea kaalu mõju on eri kultivaaride korral samasugune või erinev.

* 1. **Juhusliku faktoriga mudel**

Praktikas kõige tavalisem olukord on selline, kus me arvestame, et meie vaatlused ei ole kõik sõltumatud. Kas on mõõdetud sama objekti (ükskõik kas „madalamal tasemel“ nagu nt lind või „kõrgemal tasemel“ nagu nt kurn) korduvalt (kordusmõõtmised) või on näiteks vaatlused sooritatud erinevatel aegadel (nt eri aastatel) või mõlemad eelmised variandid. Teoreetiliselt võime mõelda, et vaatluste sõltuvuse võiksime ära kirjeldada vaid lisatavate kovariaatide abil (nt võtame arvesse linnu parameetrid või kurna parameetrid või aasta parameetrid), ent praktikas meil seda teavet tavaliselt piisaval määral ei ole (nt me ei tea või ei oska mõõta kõiki vajalikke linnu parameetreid, mis võiksid sõltuvat tunnust mõjutada).

Kõigil eelnevatel juhtudel tuleb mudelisse kaasata juhuslik faktor (või mitu juhuslikku faktorit kui arvesse on vaja võtta nt nii indiviidi kui ka aja mõju) ning idee on siis selles, et juhusliku faktori samal taseme väärtuste jäägid (ja seeläbi siis ka sõltuva tunnuse väärtused ise) on sõltuvad, eri tasemete jäägid (ja seeläbi siis ka sõltuva tunnuse väärtused ise) aga sõltumatud. Samas tuleb mõista, et see ei pea tähendama, et sõltuva tunnuse väärtused on sama juhusliku faktori väärtuse piires tingimata väga sarnased, sest eeldatavalt omavad mõju ka analüüsi kaasatud sõltumatud tunnused.

Kuivõrd juhusliku faktori kaasamisel on mudeli parameetriks sisuliselt faktori dispersioon ehk tasemetevaheline hajuvus (samas kui tavalise faktori korral on selleks tasemetesisene keskmine) siis tuleneb sellest ka nõue, et juhusliku faktori tasemeid peab mudelis olema piisavalt palju (tavalise mudeli korral ei ole probleemiks, et mõne faktori taseme kohta on vaid üks väärtus), sest dispersiooni täpne hindamine on tunduvalt keerulisem ülesanne kui keskmise täpne hindamine.

Vaatleme kaera saagikuse andmestikku oats paketist MASS. Kaera on kasvatatud kuuel erineval maa-alal, igaühel kolme erinevat kaerasorti. Iga sordi korral on kasutatud nelja erinevat väetamise intensiivsust.

head(oats)

## B V N Y

## 1 I Victory 0.0cwt 111

## 2 I Victory 0.2cwt 130

## 3 I Victory 0.4cwt 157

## 4 I Victory 0.6cwt 174

## 5 I Golden.rain 0.0cwt 117

## 6 I Golden.rain 0.2cwt 114

Ilmselgelt on maa-ala vaja kaasata juhusliku faktorina (ehk teha nn segamudel). Ris on segamudelite tegemiseks otstarbekas kasutada paketti lme4. Kuivõrd segamudelite korral on pea kõik statistilised testid ligikaudsed (ja alati ei ole isegi selge, kui ligikaudsed) siis antud pakett p-väärtusi ei väljasta. Kui me neid vajame siis peame appi võtma paketi lmerTest, mis need ligikaudsed p-väärtused meile arvutab (kasutades vaikimis III tüüpi testimist).

library(lmerTest)

m3 <- lmer(Y~V+N+(1|B), data=oats) #vabaliige sõltub juhuslikust faktorist B

anova(m3)

## Type III Analysis of Variance Table with Satterthwaite's method

## Sum Sq Mean Sq NumDF DenDF F value Pr(>F)

## V 1786.4 893.2 2 61 3.8091 0.02762 \*

## N 20020.5 6673.5 3 61 28.4598 1.239e-11 \*\*\*

Väetamine kindlasti mõjutab saagikust (p<0.001) ja tundub, et ka sordil on mõju (p=0.03). Antud mudelis lubasime niisiis, et igal maa-alal on erinev saagikuse baastase.

summary(m3)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [

## lmerModLmerTest]

## Formula: Y ~ V + N + (1 | B)

## Data: oats

## REML criterion at convergence: 577.3

## Scaled residuals:

## Min 1Q Median 3Q Max

## -2.0301 -0.7531 0.1071 0.6866 2.0736

## Random effects:

## Groups Name Variance Std.Dev.

## B (Intercept) 245.0 15.65

## Residual 234.5 15.31

## Number of obs: 72, groups: B, 6

## Fixed effects:

## Estimate Std. Error df t value Pr(>|t|)

## (Intercept) 79.917 7.771 9.271 10.285 2.25e-06 \*\*\*

## VMarvellous 5.292 4.420 61.000 1.197 0.235908

## VVictory -6.875 4.420 61.000 -1.555 0.125058

## N0.2cwt 19.500 5.104 61.000 3.820 0.000315 \*\*\*

## N0.4cwt 34.833 5.104 61.000 6.824 4.64e-09 \*\*\*

## N0.6cwt 44.000 5.104 61.000 8.620 3.80e-12 \*\*\*

## Correlation of Fixed Effects:

## (Intr) VMrvll VVctry N0.2cw N0.4cw

## VMarvellous -0.284

## VVictory -0.284 0.500

## N0.2cwt -0.328 0.000 0.000

## N0.4cwt -0.328 0.000 0.000 0.500

## N0.6cwt -0.328 0.000 0.000 0.500 0.500

Funktsiooni summary abil saame heita kiirpilgu jääkide jaotusele ning lisaks näeme, et maa-alade vaheline saagikuse varieeruvus (245) on antud juhul üsna sarnane jääkhajuvuse varieeruvusega (235).

Ehk on aga hoopis mõistlik arvata, et sordid ei käitu erinevatel maa-aladel mitte ühtemoodi vaid üks sort võib paremini hakkama saada ühel alal, teine aga teisel (ehk siis sisuliselt esineb sordi ja ala vaheline koosmõju)?

m4 <- lmer(Y~N+V+(V|B), data=oats) #V mõju sõltub juhuslikust faktorist

anova(m4)

## Type III Analysis of Variance Table with Satterthwaite's method

## Sum Sq Mean Sq NumDF DenDF F value Pr(>F)

## N 20020.5 6673.5 3 51 41.0527 1.228e-13 \*\*\*

## V 404.7 202.3 2 5 1.2447 0.3642

 Tundub tõepoolest, et selline fenomen siin aset leiab ja seega ei ole keskmiselt võimalik öelda, et mõnda sorti saaks saagikuse mõttes eelistada (p=0.36).

**8.3**. **Kordusmõõtmistega mudel**

Lihtsamatel juhtudel (ehk olukorras, kus „aeg“ ei ole analüüsi kaasatud) on kordusmõõtmistega mudel lihtsalt tavaline juhusliku faktoriga mudel. Kui vaatluste ajaline järjestus on oluline, siis on mudel veidi keerulisem.

Vaatleme nüüd uuesti tibude kasvuandmestikku ChickWeight, kus vaadeldud tibude kehakaalu erinevate söötade korral. Nüüd on meil olemas tööriist, mis võimaldab andmestikku täies ulatuses kasutada. Eeldame, et kaal kasvab ajas lineaarselt ning igal tibul on oma unikaalne (vabaliikme ja tõusuga) kasvusirge ning ka sööt võib kasvusirge vabaliiget ja tõusu mõjutada.

m5 <- lmer(weight~Time\*Diet+(Time|Chick), data=ChickWeight)

anova(m5)

## Type III Analysis of Variance Table with Satterthwaite's method

## Sum Sq Mean Sq NumDF DenDF F value Pr(>F)

## Time 53171 53171 1 45.445 325.4619 < 2.2e-16 \*\*\*

## Diet 1638 546 3 45.971 3.3415 0.027184 \*

## Time:Diet 2792 931 3 45.541 5.6963 0.002127 \*\*

Väljundist järeldub, et söötade mõjud (tõusud ajas) on erinevad (p=0.002). Aga milliste söötade korral siis tõusud erinevad? Siin aitab meid taaskord pakett emmeans, kus tõusude võrdlemiseks on funktsioon emtrends.

pairs(emtrends(m5, "Diet", var="Time")) #esimene tunnus näitab klasse, teine kovariaati

## contrast estimate SE df t.ratio p.value

## 1 - 2 -2.332 1.30 45.6 -1.788 0.2924

## 1 - 3 -5.146 1.30 45.6 -3.944 0.0015

## 1 - 4 -3.255 1.31 45.7 -2.494 0.0744

 ## 2 - 3 -2.814 1.50 45.0 -1.879 0.2517

 ## 2 - 4 -0.923 1.50 45.0 -0.616 0.9265

 ## 3 - 4 1.891 1.50 45.0 1.262 0.5915

Näeme, et oluliselt erinevad vaid esimese ja kolmanda sööda tõus (p=0.002).

Samuti võime küsida, millise sööda korral on tibude kaal lõpphetkel suurim (ja kui suured need erinevused umbes on). Põhimõtteliselt on tegu keskmiste mitmese võrdlemisega, ent me ei soovi võrdlust teostada keskmise aja korral. Seetõttu on ka käsk veidi keerulisem.

pairs(emmeans(ref\_grid(m5, at=list(Time=24)), ~Diet|Time))

## Time = 24:

## contrast estimate SE df t.ratio p.value

## 1 - 2 -50.9 27 45.5 -1.887 0.2478

## 1 - 3 -108.1 27 45.5 -4.004 0.0013

## 1 - 4 -76.4 27 45.6 -2.828 0.0338

## 2 - 3 -57.1 31 45.0 -1.844 0.2665

## 2 - 4 -25.4 31 45.0 -0.820 0.8445

## 3 - 4 31.7 31 45.0 1.023 0.7366

## Degrees-of-freedom method: kenward-roger

## P value adjustment: tukey method for comparing a family of 4 estimates

Näeme, et katse lõppajaks eristuvad esimest sööta saanud tibud oluliselt kolmandat või neljandat sööta saanud tibudest olles kaalult kergemad (vastavalt p=0.001 ja p=0.03).

**8.4.** **Muutuva sõltuvusega mudel**

Kui teostatud vaatlustel on ruumiline paigutus (nt asukoht kaardil) või on põhjust eeldada, et vaatluste sõltuvus on muutuv (nt ajaliselt lähemad vaatlused on tugevamalt seotud kui ajaliselt kaugemad) siis ei piisa sellise sõltuvuse kirjeldamiseks juhuslikust faktorist (sest selles mudelis on nt sama objekti kõigi mõõtmiste vaheline sõltuvus samasugune). Esimesel juhul räägitakse vigade ruumilisest autokorrelatsioonist ja teisel juhul ajalisest autokorrelatsioonist. Selliseid muutuva sõltuvusega mudeleid nimetatakse ka kovariatsioonistruktuuriga (või korrelatsioonistruktuuriga) mudeliteks.

Selliseid mudeleid võimaldab Ris mugavalt sobitada pakett nlme. Kui mudelisse on vaja kaasata ka juhuslik faktor, siis tuleb kasutada antud paketi funktsiooni lme, kui juhuslikku faktorit vaja ei ole siis kasutatakse funktsiooni gls.

Vaatleme paketis nlme leiduvat nisu saagikuse andmestikku Wheat2. Eri asukohtadesse on regulaarse paigutuse alusel külvatud 56 erinevat nisu sorti ning registreeritud saagikus.

library(nlme)

head(Wheat2)

## Grouped Data: yield ~ variety | Block

## Block variety yield latitude longitude

## 1 1 LANCER 29.25 4.3 19.2

## 2 1 BRULE 31.55 4.3 20.4

## 3 1 REDLAND 35.05 4.3 21.6

## 4 1 CODY 30.10 4.3 22.8

## 5 1 ARAPAHOE 33.05 4.3 24.0

## 6 1 NE83404 30.25 4.3 25.2

Näeme, et lisaks asukoha koordinaatidele on andmestikus ka asukohti põhjalaiust pidi plokkideks jagav tunnus. Võib oletada, et põhja-lõunasuunaline mõju saagikusele on olemas (ja seeläbi ka otsekui ploki mõju), ent ei ole ilmselt mõistlik arvata, et just ploki piir on kuidagi maagiline – pigem toimub saagikuse kasv ilmselt ehk sujuvamalt (ja ka sama ploki piires). Antud andmetele on mõistlik sobitada muutuva sõltuvusega mudel.

m6 <- gls(yield~variety, corr=corGaus(form=~latitude+longitude, nugget=T), data=Wheat2) #nugget=T määrab, et kaks samas punktis tehtud vaatlust ei ole maksimaalselt korreleeritud

anova(m6)

## Denom. DF: 168

## numDF F-value p-value

## (Intercept) 1 97.80435 <.0001

## variety 55 1.85682 0.0014

summary(m6)

## Generalized least squares fit by REML

## Model: yield ~ variety

## Data: Wheat2

 ## AIC BIC logLik

 ## 1185.102 1369.416 -533.5509

## Correlation Structure: Gaussian spatial correlation

## Formula: ~latitude + longitude

## Parameter estimate(s):

## range nugget

## 10.7006062 0.2614215

Näeme, et sortide saagikuse vahel on statistiline erinevus (p=0.001). Kui sooviksime seda erinevust täpsemalt uurida siis aitaks meid pakett emmeans. Funktsiooni summary väljundi algusosast saame aga välja lugeda muutuvat sõltuvust iseloomustavate parameetrite väärtused. Parameeter nugget iseloomustab sõltuvuse tugevust – kahe samast punktist tehtud vaatluse korreleeritusse määr on üks miinus selle parameetri väärtus (antud näites siis 0.74). Parameeter range iseloomustab korreleeritusse määra kahanemise kiirust. Mida suurem on see väärtus, seda kaugemad väärtused on teineteisega veel arvestataval määral korreleeritud. Kuivõrd korrelatsioonistruktuure on erinevaid siis sõltub täpne korreleeritusse määra arvutusvalem just sellest, millise korrelatsioonistruktuuri oleme mudelisse kaasanud (vt ka ?corClasses).

Kuldreegel on, et palju olulisem sellest, millise ruumilise või ajalise korrelatsioonistruktuuri oleme oma mudelisse valinud, on see, et oleme selle üldse mudelisse kaasanud (eeldusel muidugi, et vaatluste vigade ruumiline või ajaline sõltuvus esineb).

1. **Kui sõltuv tunnus ei ole pidev tunnus**

Kui sõltuv tunnus ei ole pidev tunnus siis ei saa me ka kasutada tavalisi lineaarseid mudeleid (nt ANOVA, ANCOVA), mida eelnevalt vaadelnud oleme. Veidi suuremat pilti vaadates mõistame, et seni oleme tegelikult pidevalt modelleerinud jaotuse keskväärtust ja üksikväärtused siis lihtsalt varieeruvad (normaaljaotuse alusel) modelleeritud keskväärtuse ümber.

Oletame nüüd, et meie sõltuv tunnus on nt sugu (millel on seega kaks kategooriat). Võime need väärtused tähistada ka nt kui, et null on isane ja üks on emane. Sellisel juhul oleks emaste osakaal lihtsalt ühtede osakaal või ka vaatluste keskmine. Kui nüüd sooviksime modelleerida osakaalu (ehk kaasata sõltumatuid tunnuseid) siis tuleb teha kahte asja – seos osakaalu kohta tuleb mõistlikult teisendada (sest osakaal saab olla vaid vahemikus nullist üheni) ja vaatluste varieerumine keskmise ümber ei saa olla normaaljaotuse alusel (sest vaatlused saavad olla ju vaid null või üks). Siin tuleb mängu nn üldistatud lineaarne mudel. Üldistatud siis selles mõttes, et normaaljaotuse asemel võib vaatluste jaotuseks olla ka mõni muu jaotus (nt Bernoulli või Poissoni jaotus) ning lisaks modelleeritakse üldiselt jätkuvalt jaotuse keskmisele vastavat parameetrit sõltumatute tunnuste lineaarkombinatsiooni abil, ent sellele parameetrile on rakendatud mõnd teisendust (nt logit või logaritm).

* 1. **Testid sagedustabeli korral**

Vahel võib olla ka nii, et ühtegi sõltumatut tunnust ei olegi ning soovime testida vaid andmete jaotumist vastavalt mingitele teoreetilistele osakaaludele. Nt võime küsida, kas juhuslikult järvest püütud 54 kala, kellest 31 osutus isaseks (ja vastavalt 23 emaseks) on tõestus selle kohta, et antud järves on isaste ja emaste kalade suhe erinev ühest. Appi tuleb hii-ruut test.

chisq.test(c(31,23), p=c(0.5,0.5)) #teise argumendina anname tõenäosused, mille kehtivust kontrollime

## Chi-squared test for given probabilities

## data: c(31, 23)

## X-squared = 1.1852, df = 1, p-value = 0.2763

Näeme, et antud valimi põhjal ei saa väita, et suhe ühest erineks (p=0.28).

Analoogilist testi saame teha ka mitmemõõtmelise sagedustabeli korral. Olgu nt lisaks eristatud ka kala liik (ahven või koger) ning valimis 10 isast ahvenat, 21 isast kokre, 8 emast ahvenat ja 15 emast kokre. Kuivõrd uurijal on teada, et kolmveerand järve kaladest on kogred siis võib ta kontrollida, kas selline valim on kooskõlas tõenäosustega, mis tekivad kui lisaks arvestada, et isaste ja emaste kalade suhe ei erine kummalgi liigil ühest

chisq.test(c(10,21,8,15), p=c(0.5\*0.25,0.5\*0.75,0.5\*0.25,0.5\*0.75))

## Chi-squared test for given probabilities

## data: c(10, 21, 8, 15)

## X-squared = 3.1852, df = 3, p-value = 0.3639

Tundub, et kõik on ootuspärane (p=0.36).

Samas võime sellise mitmemõõtmelise sagedustabeli korral küsida, kas emaste ja isaste suhted on mõlema kalaliigi korral samasugused (ehk tunnused liik ja sugu on sõltumatud). Selle küsimuse korral tulebki funktsioonile sagedustabel mitmemõõtmelisena ette anda.

chisq.test(matrix(c(10,21,8,15), nrow=2))

## Pearson's Chi-squared test with Yates' continuity correction

## data: matrix(c(10, 21, 8, 15), nrow = 2)

## X-squared = 6.1405e-31, df = 1, p-value = 1

Kindlasti ei saa antud testi põhjal väita, liik ja sugu ei ole sõltumatud (p=1).

Kui andmed on andmemaatriksis, siis ei pea me muidugi neid vaadeldud sagedusi ise sisse tippima vaid võime nt funktsiooni xtabs abil sagedustabeli koostada.

Hii-ruut test eeldab, et ühegi lahtri oodatav sagedus ei ole alla ühe ja enamuste lahtrite oodatav sagedus on üle viie. Kui need eeldused ei ole täidetud, ent sagedustabeli äärejaotused on fikseeritud (mitte ei selgu katse tulemusena) siis saame kasutada Fisheri testi, mis oodatavatele sagedustele piire ei sea. Loengus toodud paaritumisnäite korral saame

fisher.test(matrix(c(24,12,8,16), nrow=2))

## Fisher's Exact Test for Count Data

## data: matrix(c(24, 12, 8, 16), nrow = 2)

## p-value = 0.01727

## alternative hypothesis: true odds ratio is not equal to 1

## 95 percent confidence interval:

## 1.182306 13.917285

## sample estimates:

## odds ratio

## 3.900965

Testi tulemusena võime öelda, et paaritumisel ei valita kaaslast sõltumatult (p=0.02)

Kui tegu on 2x2 sagedustabeliga ja üks äärejaotustest on fikseeritud siis paketis Barnard on funktsioon barnard.test, mis võimaldab samuti tunnustevahelise assotsiatsiooni olemasolu testida.

Suuremamõõtmeliste sagedustabelite korral saame tunnuste sõltumatust testida log-lineaarse mudeli abil. Vaatleme näitena paketi MASS andmestikku snails. Katses testiti kahe teoliigi elulemust erinevate keskkonnatingimuste korral.

head(snails)

## Species Exposure Rel.Hum Temp Deaths N

## 1 A 1 60.0 10 0 20

## 2 A 1 60.0 15 0 20

## 3 A 1 60.0 20 0 20

## 4 A 1 65.8 10 0 20

## 5 A 1 65.8 15 0 20

## 6 A 1 65.8 20 0 20

Valime analüüsiks vaid 4 nädalat vangistuses hoitud teod ning korraldame andmetabeli ümber nii, et saaksime seda kasutada. Selleks tekitame eraldi read surnud ja elusate tigude jaoks.

abi <- subset(snails, Exposure==4)

abi2 <- abi

abi2$Deaths <- abi2$N-abi2$Deaths #ellujäänute arv

abi$elus <- "ei"

abi2$elus <- "jah"

andmed <- rbind(abi, abi2) #andmestikud üksteise alla

tabel <- xtabs(Deaths~Species+Temp+elus, data=andmed)

tabel

## , , elus = ei

## Temp

## Species 10 15 20

## A 16 17 22

## B 31 38 44

## , , elus = jah

## Temp

## Species 10 15 20

## A 64 63 58

## B 49 42 36

Nüüd võime küsida, kas seos temperatuuri ja elulemuse vahel sõltub liigist, mis log-linearse mudeli mõttes on samaväärne temperatuuri, elulemuse ja liigi vahelise interaktsiooni statistilise olulisusega. Loome 2 mudelit – ühe, kus antud interaktsiooni pole (aga kõik madalama astme asjad on), ja teise, kus lisaks ka vastav interaktsioon.

m1 <- loglm(~(Temp+Species+elus)\*\*2, data=tabel)

m2 <- loglm(~Temp\*Species\*elus, data=tabel)

anova(m1,m2)

## LR tests for hierarchical log-linear models

## Model 1:

## ~(Temp + Species + elus)^2

## Model 2:

## ~Temp \* Species \* elus

## Deviance df Delta(Dev) Delta(df) P(> Delta(Dev)

## Model 1 0.3614829 2

## Model 2 0.0000000 0 0.3614829 2 0.83465

## Saturated 0.0000000 0 0.0000000 0 1.00000

Näeme, et antud juhul ei saa väita, et seos temperatuuri ja elulemuse vahel sõltub liigist (p=0.83).

* 1. **Logistiline regressioon**

Tavaliselt ei ole meil tunnused sümmeetrilises olukorras (nagu nt sagedustabelis) vaid meil on selge ettekujutus, et üks tunnustest on sõltuv tunnus. Kahe kategooriaga sõltuva tunnuse modelleerimiseks on logistiline mudel.

Vaatame uuesti eelmist tigude näidet, ent kaasame nüüd temperatuuri pideva tunnusena.

m3 <- glm(cbind(Deaths,N-Deaths)~Species\*Temp, data=snails, subset=(Exposure==4), family="binomial")

Kui andmestikus oleks iga teo jaoks eraldi andmerida, siis asuks vasakul pool lainekese märki vaid vastav null-üks tunnuse nimi. Antud süntaksit (ühtede arv, nullide arv) saame kasutada siis kui andmed on eelnevalt grupeeritud. Kui grupeerimine on võimalik (ehk esineb sama sõltumatute tunnuste väärtusekombinatsioonidega vaatlusi) siis tuleks seda kindlasti teha. Täpsemalt tuleb sellest juttu ülehajuvuse teema juures.

summary(m3)

## Call:

## glm(formula = cbind(Deaths, N - Deaths) ~ Species \* Temp, family = "binomial", data = snails, subset = (Exposure == 4))

## Deviance Residuals:

## Min 1Q Median 3Q Max

## -2.0283 -1.0996 -0.2385 1.0388 2.3256

## Coefficients:

## Estimate Std. Error z value Pr(>|z|)

## (Intercept) -1.86120 0.60301 -3.087 0.00202 \*\*

## SpeciesB 0.75513 0.78395 0.963 0.33543

## Temp 0.04266 0.03790 1.126 0.26024

## SpeciesB:Temp 0.02315 0.04967 0.466 0.64121

## (Dispersion parameter for binomial family taken to be 1)

## Null deviance: 77.533 on 23 degrees of freedom

## Residual deviance: 40.711 on 20 degrees of freedom

## AIC: 125.13

## Number of Fisher Scoring iterations: 4

Funktsiooni summary väljundist saame logistilise regressiooni korral küll ettekujutuse jääkide jaotusest (logistilise regressiooni korral on jääke võimalik defineerida erinevat moodi, enimkasutavad on hälbimusjäägid (*deviance residuals*), ent see ei ole õige koht sõltumatute tunnuste olulisuse üle otsustamiseks, sest Waldi test (see, mida väljastab summary) on üldistatud lineaarsete mudelite korral ligikaudne ja seetõttu on korrektsem kasutada tõepäras suhte testi (see, mida väljastavad anova ja drop1). Kui me ei soovi tunnuste järjekorra muutmisega vaeva näha siis saame ka siin II tüüpi testimise tulemused mugavalt kätte paketi car funktsiooniga Anova.

Väljundis on näha ka mudeli hälbimus (see on üldistatud lineaarsete mudelite korral analoogiline asi ruutude summaga tavalises lineaarses mudelis). Nii on siis antud andmete korral nullhälbimus (vaid vabaliiget sisaldav mudel) 77.5 ja meie mudeli korral on hälbimuse väärtuseks 40.7. Mida nendest arvudest järeldada võib, sellest tuleb juttu ülehajuvuse teema juures.

Anova(m3)

## Analysis of Deviance Table (Type II tests)

## Response: cbind(Deaths, N - Deaths)

## LR Chisq Df Pr(>Chisq)

## Species 31.631 1 1.864e-08 \*\*\*

## Temp 5.318 1 0.0211 \*

## Species:Temp 0.217 1 0.6414

Näeme, et (nii nagu log-lineaarse analüüsi põhjalgi selgus) liigi ja temperatuuri koosmõju ei näi esinevat (p=0.64). Küll aga sõltub elulemus eraldi nii liigist (p<0.001) kui katsel kasutatud temperatuurist (p=0.02).

Mudeli sobivust saab visuaalselt kontrollida andes plot käsule argumendiks valmis mudeli (analoogiliselt funktsiooni lm mudelitega).

Regressioonijoone (see ei ole nüüd enam sirge) joonistamisel võime võrrandi ise sisse toksida (võrrandi parameetrid leiame summary tabelist), ent ohutum on jätta arvutused Ri hooleks. Tekitame lihtsalt sobivate sõltumatute tunnuste väärtustega uue andmestiku ja laseme mudelil selle põhjal välja ennustada (funktsiooni predict kasutades tuleb üldistatud lineaarsete mudelite korral ka teada anda, et ennustused peavad olema sõltuva tunnuse originaalskaalal, vaikimisi kasutatakse teisendatud skaalat).

Muidugi saaksime joonisele lisada ka valimi vaatlused, ent kuna antud katses oli temperatuur fikseeritud, siis asuksid paljud vaatlused joonisel lihtsalt üksteise peal ning isegi nende vähene liigutamine (funktsiooni jitter abil) ei annaks ilmselt head tulemust.

tempjada <- 5:25

uued\_a <- data.frame(Species="A", Temp=tempjada)

uued\_b <- data.frame(Species="B", Temp=tempjada)

plot(y=predict(m3, uued\_a, type="response"), x=tempjada, type="l", ylim=c(0,1), xlab="Temperatuur", ylab="surma tõenäosus")

lines(y=predict(m3, uued\_b, type="response"), x=tempjada, col=2) #lisame joone juba valmis graafikule



* 1. **Poissoni regressioon**

Kui sõltuv tunnus on naturaalarvuliste väärtustega aga need ei väljenda mitte õnnestumiste arvu katseseerias vaid mingit arvukust, siis on korrektne kasutada üldistatud lineaarset mudelit Poissoni jaotusega.

Kasutame uuesti tigude andmestikku snails end kujutame nüüd ette, et katse toimus teisiti – olgu „N“ nüüd katse kestuse aeg ja „Deaths“ väljendagu loendatud hukkunud tigude arvu.

m4 <- glm(Deaths~Species\*Temp, data=snails, subset=(Exposure==4), family="poisson")

summary(m4)

## Call:

## glm(formula = Deaths ~ Species \* Temp, family = "poisson", data = snails, subset = (Exposure == 4))

## Deviance Residuals:

## Min 1Q Median 3Q Max

## -1.5542 -0.8574 -0.2072 0.7649 1.4365

## Coefficients:

## Estimate Std. Error z value Pr(>|z|)

## (Intercept) 1.020324 0.534214 1.910 0.0561 .

## SpeciesB 0.691866 0.651910 1.061 0.2886

## Temp 0.032874 0.033252 0.989 0.3228

## SpeciesB:Temp 0.001812 0.040555 0.045 0.9644

## (Dispersion parameter for poisson family taken to be 1)

## Null deviance: 47.456 on 23 degrees of freedom

## Residual deviance: 23.781 on 20 degrees of freedom

## AIC: 119.78

## Number of Fisher Scoring iterations: 4

Anova(m4)

## Analysis of Deviance Table (Type II tests)

## Response: Deaths

## LR Chisq Df Pr(>Chisq)

## Species 20.442 1 6.147e-06 \*\*\*

## Temp 3.231 1 0.07226 .

## Species:Temp 0.002 1 0.96437

Funktsiooni summary väljund on ülesehituselt analoogiline logistilise regressiooni korral nähtuga (ent Poissoni jaotuse korral on küllalt lihtne ka parameetrite hinnangutest aru saada – parameetri väärtus tuleb analoogiliselt lineaarse mudeliga tunnuse väärtusega läbi korrutada ent seejärel tuleb see korrutis tõsta e astmesse ning nii saame väärtuse, mis näitab mitu korda (mitte võrra!) antud tunnus keskmist mõjutab). Ka Poissoni regressiooni korral on mudeli sobitamise aluseks mudeli hälbimus ja ka siin on sõltumatute tunnuste olulisusi korrektne vaadata funktsiooni Anova abil. Näeme, et liigil on mõju (p<0.001), ent temperatuuril mitte (p=0.07).

Antud andmete korral olid kõik vaatlusajad võrdse pikkusega. Mis saanuks aga juhul kui see nii ei olnuks? Üks võimalus oleks muidugi leida uue tunnusena surmade arv vaadeldud aja kohta, ent sel juhul tuleks mudeldamisel kindlasti arvesse võtta ka selle uue tunnuse täpsuse erinevus (kui vaatleme ühel juhul ühe tunni ja teisel 2 tundi, siis teisel juhul on uue tunnuse dispersioon 2 korda väiksem). Selline lähenemine on tehniliselt keerukas ja lihtsam on kaasata nn vaatluspingutus otse Poissoni mudelisse valiku offset abil (tulebki lisada just logaritm).

m4a <- glm(Deaths~Species\*Temp, offset=log(N), data=snails, subset=(Exposure==4), family="poisson")

* 1. **Ülehajuvus**

Logistilise regressiooni ja Poissoni regressiooni korral on erinevalt tavalisest regressioonist modelleerimisel kasutusel ainult üks jaotust kirjeldav suurus (keskmine). Tavalise regressiooni korral pannakse sõltumatud tunnused samuti mõjutama sõltuva tunnuse keskmist, ent lisaks sisaldub mudelis ka hajuvus (ehk jaotust kirjeldavaid suurusi on kaks). Binoomjaotuse ja Poissoni jaotuse korral on aga jaotuse dispersioon lihtsalt jaotuse keskmise funktsioon ja seega seda ei oleks otsekui vaja eraldi arvesse võtta.

Praktikas on aga olukord tüüpiliselt selline, et andmetes esinev hajuvus ületab hajuvust, mida mudel ette näeb ning sellist olukorda tuntakse kui ülehajuvust (*overdispersion*). Ülehajuvus tekib, kuna vaatlused on sõltuvad, kuid me pole seda mudeli struktuuris korrektselt arvestanud (nt puuduvad meil mudelist mõned olulised sõltumatud tunnused või ka juhuslik faktor; sõltumatute tunnuste teisendus võib samuti aidata). Sageli ei ole meil praktikas võimalik mudeli struktuuri korrektseks muuta (nt me ei ole mõne olulise sõltumatu tunnuse väärtusi katse käigus registreerinud)

Ülehajuvus toob kaasa situatsiooni, kus mudeli põhjal leitud sõltumatute tunnuste parameetrite standardvigade hinnangud on liiga väikesed ja vastavad p-väärtused seega samuti liiga väikesed. Selle korrigeerimiseks tuuakse ka binoom- ja Poissoni jaotuse korral mängu lisasuurus nn ülehajuvuse parameeter. Selle abil korrigeeritakse standardvigasid ning ühtlasi siis ka p-väärtusi.

Ülehajuvuse olemasolu on võimalik ka statistiliselt testida, ent Poissoni regressiooni ja piisavalt pikkade seeriatega logistilise regressiooni korral annab küllalt hea tulemuse rusikareegel: kui summary väljundis toodud jääkhälbimus (*residual deviance*) ületab märkimisväärselt samal real toodud vabadusastmete arvu, siis on tegu ülehajuvusega ja antud mudelit kasutada (või mudeli põhjal järeldusi teha) ei tohiks.

Rusikareegel töötab seda paremini, mida suurem on iga meie valimi andmerea sõltuva tunnuse oodatav väärtus. Kui me teeme logistilist regressiooni ja igas reas on sõltuva tunnuse väärtus kas 0 või 1, siis ei ole võimalik seda rusikareeglit kasutada ja ülehajuvus jääb üldjuhul diagnoosimata. Siit tuleb välja põhjus, miks logistilises regressioonis tuleks võimalusel alati andmed seeriatesse grupeerida.

Viime nüüd 2 eelnenud analüüsi korrektselt läbi võttes arvesse ka ülehajuvuse (vajaduse ülehajuvust arvestada leidnuksime funktsiooni summary väljunditest m3 ja m4 korral).

m3a <- glm(cbind(Deaths,N-Deaths)~Species\*Temp, data=snails, subset=(Exposure==4), family="quasibinomial")

summary(m3a)

## Call:

## glm(formula = cbind(Deaths, N - Deaths) ~ Species \* Temp, family = "quasibinomial", data = snails, subset = (Exposure == 4))

## Deviance Residuals:

## Min 1Q Median 3Q Max

## -2.0283 -1.0996 -0.2385 1.0388 2.3256

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) -1.86120 0.85155 -2.186 0.0409 \*

## SpeciesB 0.75513 1.10708 0.682 0.5030

## Temp 0.04266 0.05352 0.797 0.4347

## SpeciesB:Temp 0.02315 0.07014 0.330 0.7448

## (Dispersion parameter for quasibinomial family taken to be 1.994238)

## Null deviance: 77.533 on 23 degrees of freedom

## Residual deviance: 40.711 on 20 degrees of freedom

## AIC: NA

## Number of Fisher Scoring iterations: 4

Väljundist näeme, et dispersioon hinnati keskmisest umbes 2 korda suuremaks – umbes niisama arv kordi on jääkhälbimus (40.7) suurem vabadusastmete arvust (20).

Anova(m3a)

## Analysis of Deviance Table (Type II tests)

## Response: cbind(Deaths, N - Deaths)

## LR Chisq Df Pr(>Chisq)

## Species 15.8612 1 6.816e-05 \*\*\*

## Temp 2.6668 1 0.1025

## Species:Temp 0.1088 1 0.7415

Oodatult ei ole temperatuuri mõju nüüd enam statistiliselt oluline (p=0.10).

m4 <- glm(Deaths~Species\*Temp, data=snails, subset=(Exposure==4), family="quasipoisson")

summary(m4)

## Call:

## glm(formula = Deaths ~ Species \* Temp, family = "quasipoisson", data = snails, subset = (Exposure == 4))

## Deviance Residuals:

## Min 1Q Median 3Q Max

## -1.5542 -0.8574 -0.2072 0.7649 1.4365

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) 1.020324 0.581902 1.753 0.0948 .

## SpeciesB 0.691866 0.710104 0.974 0.3415

## Temp 0.032874 0.036220 0.908 0.3749

## SpeciesB:Temp 0.001812 0.044175 0.041 0.9677

## (Dispersion parameter for quasipoisson family taken to be 1.186504)

## Null deviance: 47.456 on 23 degrees of freedom

## Residual deviance: 23.781 on 20 degrees of freedom

## AIC: NA

## Number of Fisher Scoring iterations: 4

Näeme, et Poissoni mudelis ei ole ülehajuvuse probleem suur (dispersioon ületab keskmist u 1.2 korda; jääkhälbimus on 23.8 ja vabadusastmete arv 20)

Anova(m4)

## Analysis of Deviance Table (Type II tests)

## Response: Deaths

## LR Chisq Df Pr(>Chisq)

## Species 17.2286 1 3.314e-05 \*\*\*

## Temp 2.7231 1 0.0989 .

## Species:Temp 0.0017 1 0.9673

Muutus p-väärtustes on seega ülehajuvuse arvestamisel ka väiksem.

Loendusandmete korral kasutatakse Poissoni jaotuse asemel sageli negatiivset binomiaaljaotust (*negative binomial distribution*) – sellel jaotusel on juba endal kaks parameetrit ja seeläbi on ülehajuvust kergem arvesse võtta (nt funktsioon glm.nb paketis MASS) .

Veelkord, ülehajuvus andmetes on pigem reegel kui erand!

* 1. **Liigsed nullid sõltuva tunnuse väärtustes**

Loendusandmete korral on tavaline, et nulle on registreeritud väärtuste korral rohkem kui mudeljaotus seda ette näeks (*zero-inflation*). Ühelt poolt võib see põhjustada ka ülehajuvust, ent isegi kui ülehajuvuse arvesse võtame siis ei ole mudel sellegipoolest sobilik.

Statistiliselt on liigsete nullide olemasolu GLM mudeli korral võimalik testida Clarke’i testiga (nt paketis GJRM funktsioon VuongClarke), argumentideks tuleb anda mudel, mis ei arvesta võimalikke liigseid nulle ja muidu täpselt identne, ent võimalikke liigseid nulle arvestav mudel. Praktikas on aga küllalt mugav võrrelda sobitatud mudeli poolt ennustatud nullide arvu tegelikult valimis esinevate nullide arvuga.

Kasutame nüüd tervet tigude andmestikku ja kaasame mudelisse ka vangistuse kestuse.

m5 <- glm(Deaths~Species\*Temp\*Exposure, data=snails, family="poisson")

summary(m5)

## Call:

## glm(formula = Deaths ~ Species \* Temp \* Exposure, family = "poisson", data = snails)

## Deviance Residuals:

## Min 1Q Median 3Q Max

## -1.9007 -0.9128 -0.4148 0.4325 3.1573

## Coefficients:

## Estimate Std. Error z value Pr(>|z|)

## (Intercept) -9.38883 3.24267 -2.895 0.00379 \*\*

## SpeciesB 5.33644 3.56530 1.497 0.13445

## Temp 0.35412 0.17976 1.970 0.04885 \*

## Exposure 2.55548 0.85506 2.989 0.00280 \*\*

## SpeciesB:Temp -0.21675 0.20009 -1.083 0.27868

## SpeciesB:Exposure -1.10653 0.94679 -1.169 0.24252

## Temp:Exposure -0.07603 0.04775 -1.592 0.11133

## SpeciesB:Temp:Exposure 0.05187 0.05354 0.969 0.33268

## (Dispersion parameter for poisson family taken to be 1)

## Null deviance: 448.08 on 95 degrees of freedom

## Residual deviance: 100.53 on 88 degrees of freedom

## AIC: 293.32

## Number of Fisher Scoring iterations: 5

Tundub, et ülehajuvus probleemiks ei ole. Kuidas on aga lood nullidega? Leiame mudeli poolt ennustatava nullide arvu ja tegeliku nullide arvu andmestikus.

sum(dpois(0, exp(predict(m5, type="link"))))

## 34.96197

sum(snails$Deaths==0)

## 41

Näeme, et nulle on küll rohkem kui teoreetiliselt ennustatud, ent erinevus ei ole suur.

Anova(m5)

## Analysis of Deviance Table (Type II tests)

## Response: Deaths

## LR Chisq Df Pr(>Chisq)

## Species 46.107 1 1.120e-11 \*\*\*

## Temp 15.430 1 8.561e-05 \*\*\*

## Exposure 280.475 1 < 2.2e-16 \*\*\*

## Species:Temp 0.646 1 0.42169

## Species:Exposure 1.402 1 0.23643

## Temp:Exposure 2.831 1 0.09244 .

## Species:Temp:Exposure 0.978 1 0.3225

Valimimaht on eelmiste mudelitega võrreldes oluliselt kasvanud. Statistiliselt olulised on nii liik (p<0.001), temperatuur (p<0.001) kui ka vangistuse kestus (p<0.001).

Teeme nüüd mudeli, mis lubab ka liigseid nulle. Samas kasutame negatiivset binomiaaljaotust, mis pakub kaitset ülehajuvuse vastu. Paketi glmmTMB sama nimega funktsioon võimaldab samaaegselt nii juhuslike faktorite kasutamist, erinevaid sõltuva tunnuse jaotusi kui ka liignullide kaasamist mudelisse.

m5a <- glmmTMB(Deaths~Species\*Temp\*Exposure, ziformula=~1, data=snails, family="nbinom1")

Valiku ziformula abil on nullide esinemise tõenäosus võimalik seada sõltuvusse ka erinevatest sõltumatutest tunnustest.

summary(m5a)

## Family: nbinom1 ( log )

## Formula: Deaths ~ Species \* Temp \* Exposure

## Zero inflation: ~1

## Data: snails

## AIC BIC logLik deviance df.resid

## 293.4 319.1 -136.7 273.4 86

## Overdispersion parameter for nbinom1 family (): 0.389

## Conditional model:

## Estimate Std. Error z value Pr(>|z|)

## (Intercept) -9.01523 3.49950 -2.576 0.00999 \*\*

## SpeciesB 4.76958 3.90332 1.222 0.22173

## Temp 0.33445 0.19471 1.718 0.08585 .

## Exposure 2.46369 0.92713 2.657 0.00788 \*\*

## SpeciesB:Temp -0.19052 0.21966 -0.867 0.38576

## SpeciesB:Exposure -0.96704 1.04021 -0.930 0.35255

## Temp:Exposure -0.07087 0.05194 -1.364 0.17241

## SpeciesB:Temp:Exposure 0.04511 0.05895 0.765 0.44410

## Zero-inflation model:

## Estimate Std. Error z value Pr(>|z|)

## (Intercept) -21.87 9314.48 -0.002 0.998

Väljundist näeme, et liignullide esinemist modelleeriv liige ei erine statistiliselt oluliselt nullist (p=1) ehk siis ei ole usaldusväärset infot, et nulle oleks oluliselt rohkem (või vähem) kui negatiivne binomiaaljaotuse ette näeb.

Anova(m5a)

## Analysis of Deviance Table (Type II Wald chisquare tests)

## Response: Deaths

## Chisq Df Pr(>Chisq)

## Species 27.8420 1 1.316e-07 \*\*\*

## Temp 10.8161 1 0.001006 \*\*

## Exposure 139.2106 1 < 2.2e-16 \*\*\*

## Species:Temp 0.4183 1 0.517804

## Species:Exposure 0.7589 1 0.383673

## Temp:Exposure 2.1349 1 0.143982

## Species:Temp:Exposure 0.5857 1 0.444102

Ootuspäraselt ei ole võrreldes mudeliga m5 toimunud kvalitatiivseid muutusi (sest antud andmete korral ei ole ülehajuvuse ega liignullide probleem kuigi suur). Statistiliselt olulised on nii liik (p<0.001), temperatuur (p=0.001) kui ka vangistuse kestus (p<0.001).

1. **Andmetevaheline ajaline ja ruumiline sõltuvus**

Ei ajas ega ruumis tehtud vaatlused ei ole sõltumatud kui vaatleme väga väikest aja või ruuminihet. Kui nihe on suurem (nt aasta või erinev prooviala) siis võime mõnikord sellest sõltuvusest lihtsasti mööda saada (nt kaasama aasta/ala juhusliku faktorina olukorras kus iga aasta/ala kohta on mitmeid vaatlusi). Kui meie tunnuseks on aga nt regulaarsete ajavahemike järel toimunud ühekordne mõõtmine siis ei ole juhusliku faktori kasutamine kindlasti lahendus.

* 1. **Ajalise autokorrelatsiooni tuvastamine**

Põhimõtteliselt saame muidugi alati tekitada ise sobivalt nihutatud tunnused ja arvutada nendevahelised korrelatsioonid, ent mugavam on kasutada juba valmis funktsioone. Kuigi Ris on olemas ka eraldi aegrea objektid siis eeldame meie, et meie andmed on siiski tavalise andmevektorina (nt iga-aastased vaatlused kronoloogilises järjekorras). Olgu vaatluse all Niiluse aastane vooluhulk Aswani lähistel, mis on registreeritud andmestikus Nile.

andmed <- as.numeric(Nile) #muudame tavaliseks andmevektoriks

(acf(andmed, lag.max=10)) #sulgude abil kuvatakse ka objekti sisu, mitte ainult graafik

## Autocorrelations of series ‘andmed’, by lag

## 0 1 2 3 4 5 6 7 8 9 10 ## 1.000 0.498 0.385 0.328 0.239 0.228 0.227 0.222 0.300 0.142 0.090



Sinise katkendjoonega märgitakse statistilise olulisuse piiri. Niisiis on nt viieaastase nihke korral (statistiliselt oluline) autokorrelatsioon (*lagged correlation*) andmestikus olemas (korrelatsioonikordaja r=0.23).

* 1. **Ajalise autokorrelatsiooni arvestamine mudelis**

Vaatleme nüüd loengu näidet jäneste ja rebaste arvukuse kohta. Tegu on sünteetilise (ehk simuleeritud mitte tegeliku) andmestikuga. Kuigi juhuslike arvude genereaator annab nn pseudojuhuslikke arve siis fikseerides generaatorisse mineva algväärtuse saame iga kord sama tulemuse (mis on näidete tegemiseks väga mugav).

Rebaste iga-aastane arvukus (tunnus „x“) on täiesti juhuslik ehk ajaline autokorrelatsioon puudub (vaatlused normaaljaotusest, mille keskväärtus on 27 ja standardhälve 4). Jäneste arvukus (tunnus „y“) on (negatiivses) lineaarses sõltuvuses rebaste arvukusest, ent lisatud on veel ka ajaliselt autokorreleeritud juhuslik vealiige (keskväärtusega 0, standardhälbega 5 ning järjestikuste väärtuste vahelise korrelatsiooniga 0.56).

set.seed(200) #juhuslike arvude generaatori algväärtuse fikseerimine

x <- rnorm(100,27,4)

y <- 49-0.17\*x+5\*arima.sim(model=list(ar=0.56), n=100)

cor.test(x,y)

## Pearson's product-moment correlation

## data: x and y

## t = -2.8284, df = 98, p-value = 0.005674

## alternative hypothesis: true correlation is not equal to 0

## 95 percent confidence interval:

## -0.44701410 -0.08276434

## sample estimates:

## cor

## -0.2747166

Tunnustevaheline korrelatsioonikordaja (aegridade kontekstis *cross correlation*) on statistiliselt oluline (p=0.006), ent see test ei ole ilmselt korrektne. Mõistlikum oleks teha mudel, kus jäneste arvukus sõltub rebaste arvukusest ning uurida selle mudeli jääke. Kui oskame luua mudeli, kus jääkides autokorrelatsiooni ei esine siis saame sellest mudelist välja lugeda korrektse tulemuse.

Esmalt kujutame toorandmed aegridadena graafikul.

op <- par(mfrow=c(2,1), mar=c(4,4,1,1))

plot(y, xlab="time", ylab="jäneste arvukus")

plot(x, xlab="time", ylab="rebaste arvukus", type="l")

par(op)



Teeme esmalt tavalise lineaarse mudeli.

andmed2 <- data.frame(y=as.numeric(y), x)

m1 <- lm(y~x, data=andmed2)

Uurime nüüd selle mudeli jääkide autokorrelatsiooni.

abi <- acf(resid(m1), lag.max=8, plot=F)

plot(abi, xlab="nihe", main="", ylab="mudeli jääkide autokorrelatsioon")



Oodatult on jääkides ajalise nihke üks aasta korral tugev autokorrelatsioon. Konstrueerime siis mudeli, kus arvestatakse jääkidevahelist autokorrelatsiooni.

library(nlme)

m2 <- gls(y~x, correlation=corAR1(), data=andmed)

summary(m2)

## Generalized least squares fit by REML

## Model: y ~ x

## Data: andmed

## AIC BIC logLik

## 578.4297 588.7695 -285.2148

## Correlation Structure: AR(1)

## Formula: ~1

## Parameter estimate(s):

## Phi

## 0.477944

## Coefficients:

## Value Std.Error t-value p-value

## (Intercept) 51.41723 2.9275456 17.563256 0.0000

## x -0.26024 0.1038474 -2.506018 0.0139

## Correlation:

## (Intr)

## x -0.962

## Standardized residuals:

## Min Q1 Med Q3 Max

## -2.11892052 -0.77518761 0.07638169 0.79773850 1.95765385

## Residual standard error: 4.78289

## Degrees of freedom: 100 total; 98 residual

Näeme, et mudelis leitud hinnang jääkidevahelisele autokorrelatsioonile nihke üks korral on 0.48 (valimi suurusest ja genereeritud väärtuste juhuslikkusest tulenevalt ei peagi see genereerimisel kasutatud väärtusega 0.56 täpselt ühtima). Sõltuvus rebaste arvukusest on jätkuvalt statistiliselt oluline (p=0.01).

Enne seose tõestatuks lugemist peame aga veenduma, kas antud mudeli jääkidega on kõik korras. Kui me kasutame paketi nlme mudelit, siis saame õiged jäägid kätte lisavalikuga type="normalized".

op <- par(mfrow=c(2,1), mar=c(4,4,1,1))

plot(resid(m2, type="normalized"), type="l", xlab="aeg", ylab="jääk")

abi2 <- acf(resid(m2, type="normalized"), lag.max=8, plot=F)

plot(abi2, xlab="nihe", main="", ylab="mudeli jääkide autokorrelatsioon")

par(op)



Näeme, et autokorrelatsioon on mudeli jääkidest kadunud ja seega võime tõepoolest väita, et jäneste arvukuse ja rebaste arvukuse vahel on seos.

* 1. **Ruumilise autokorrelatsiooni tuvastamine**

Kui me modelleerime ruumilisi andmeid siis üldiselt on ruumilise vigade struktuuri kaasamine mudelisse pigem reegel kui erand. Oleme juba sellekohast näidet vaadelnud (nisu saagikuste andmestik). Kas selles näites oli ruumilise struktuuri kaasamine mudelisse aga üldse tarvilik?

Üks võimalus sellisele küsimusele vastata on leida korrelatsioonikordaja ruumiline analoog ehk Morani I. Selle leidmine eeldab aga vaatluspaaride (käiakse läbi kõikvõimalikud paarid) kaalude maatriksi defineerimist, mis omakorda peaks põhinema vaatlustevahelisel kaugustel. Teeme nii, et esmalt võtame arvesse kõik paarid, kus vaatlused paiknevad üksteisele lähedal (nendel paaridel kaalud 1, teistel 0). Järgmiseks võtame arvesse veidi suurema kaugusega paarid, siis veel suurema kaugusega paarid jne. Selliselt tekib ruumilise autokorrelatsioonikordaja väärtuste jada, mis kirjeldab siis tunnuses esinevat ruumilist autokorrelatsiooni erineva vaatlustevahelise kauguse korral.

library(ape) #annab meile vajaliku funktsiooni Moran.I

distances <- as.matrix(dist(Wheat2[,4:5])) #tunnuste longitude ja latitude põhjal leitud eukleidilised kaugused paariliste vahel

plot(x=1, xlim=c(0,10), ylim=c(-1,1), type="n", xlab="kaugusklass", ylab="ruumiline autokorrelatsioon") #teeme valmis õigete telgede ulatusega graafiku, ent ei kuva veel ainsatki väärtust

for(i in 1:10){ #kümne sammuga tsükkel

dist <- (distances>(5\*(i-1))&distances<=5\*i)\*1 #kaalu 1 saavad vastavasse kaugusvahemikku jäävad paarid

points(x=i, y=Moran.I(Wheat2$yield, dist)$observed) #kanname graafikule vastavate paaride põhjal leitud ruumilise autokorrelatsiooni väärtuse

}

abline(h=0) #abijoon



Näeme, et autokorrelatsioon on vaatlustes kindlasti olemas. Analoogiliselt võiksime nüüd kontrollida ruumilist paigutust arvestava mudeli jääke veendumaks, et need on autokorrelatsioonist vabad.

Kui me aga nagunii oleme paketi nlme abil mudeli loonud, siis on analoogilise graafiku saamiseks lihtsam kasutada selle paketi funktsiooni Variogram (variogrammil kujutatatud väärtused on sisuliselt üks miinus korrelatsioonikordaja). Graafikule lisatakse vaikimisi ka lokaalne silumisjoon – kui see on ligikaudu horisontaalne sirge siis viitab see, et ruumilise autokorrelatsiooni probleemi ei ole.

m3 <- gls(yield~variety, corr=corGaus(form=~latitude+longitude, nugget=T), data=Wheat2)

plot(Variogram(m3, form=~latitude+longitude, resType="n")) #joonistame graafiku mudeli normeeritud jääkide põhjal



Punktide põhjal joonistatud silumisjoon on peaaegu horisontaalne sirge – probleemi enam ei ole. Sama tulemuse saame ka esmase koodi abil.

distances <- as.matrix(dist(Wheat2[,4:5])) #tunnuste longitude ja latitude põhjal leitud eukleidilised kaugused paariliste vahel

plot(x=1, xlim=c(0,10), ylim=c(-1,1), type="n", xlab="kaugusklass", ylab="ruumiline autokorrelatsioon") #teeme valmis õigete telgede ulatusega graafiku, ent ei kuva veel ainsatki väärtust

for(i in 1:10){ #kümne sammuga tsükkel

dist <- (distances>(5\*(i-1))&distances<=5\*i)\*1 #kaalu 1 saavad vastavasse kaugusvahemikku jäävad paarid

points(x=i, y=Moran.I(resid(m3, type="normalized", dist)$observed) #kanname graafikule vastavate jääkide paaride põhjal leitud ruumilise autokorrelatsiooni väärtuse

}

abline(h=0) #abijoon



Jääkides olev autokorrelatsioon on minimaalne.

Kuigi mudeli usaldatavuseks piisab sellest, et mudeli jäägid on autokorrelatsioonist vabad siis on hea teada, et kui sõltuvat tunnust mõjutavad tegelikkuses nii sellised tunnused, milles ruumilist autokorrelatsiooni ei esine ja sellised, milles ruumiline autokorrelatsioon esineb, siis kipuvad viimaste p-väärtused tulema tunduvalt väiksemad (nn *red-shifts*) aga me ei saa selles suhtes eriti midagi ette võtta (ainult seda teadvustada).

* 1. **Ruumilise autokorrelatsiooni arvestamine mudelis**

Kuivõrd üht klassikalist sellekohast näidet oleme juba vaadelnud, siis nüüd mõtleme natuke teistmoodi. Kujutame ette fülogeneesipuud. Ilmselt on hilisemalt lahknenud liigid üksteisega sarnasemad. Seega kui vaatluseks on liik ja me ei võta liikide seotust arvesse, siis võime nt kahe liiki kirjeldava tunnuse vahel leida seose, mis tegelikkuses tuleneb vaid ühise eellase mõjust (liigitunnustes on fülogeneetiline autokorrelatsioon). Kui me aga oskame mõõta kahe liigi kaugust fülogeneesipuus (siin on küll palju erinevaid võimalusi, kuidas seda teha) siis saame seda kaugust kasutada fülogeneetilise autokorrelatsiooni arvestamisel nii nagu kasutasime asukohtade põhjal leitud eukleidilist kaugust ruumilise autokorrelatsiooni arvestamisel.

Vaatleme näitena paketist ade4 pärit sisalikuliste andmestikku lizards. Antud paketi andmestikud tuleb funktsiooniga data esmalt kättesaadavaks teha.

library(ade4)

data(lizards)

Andmestiku abifailist näeme, et andmestikus on kokku 3 objekti: elukäigutunnuste andmemaatriks traits ning 2 eri alustel loodud fülogeneesipuud hprA ja hprB. Andmed on 18 liigi kohta. Uurime pikkuse (tunnus „matur.L“) sõltuvust vanusest (tunnus „age.mat“) täiskasvanuks saamisel.

plot(matur.L~age.mat, data=lizards$traits)



Tundub, et eksisteerib positiivne seos. Aga me ei ole arvestanud sellega, et fülogeneesipuus lähemal paiknevad liigid võivad anda kattuvat infot.

Salvestame esmalt esimese fülogeneesipuu eraldi objektina ja muudame selle Rile arusaadavaks.

library(ape)

puu <- read.tree(text=lizards$hprA)

Tekitame ka elukäigutunnustest eraldi objekti ning lisame sinna liiginimede tunnuse.

andmed <- lizards$traits

andmed$liiginimed <- rownames(andmed)

Paketist ape pärinevad mitmed korrelatsioonistruktuurid, millele saab fülogeneesipuu otse argumendiks anda (me ei pea siis ise liikidevahelist kaugustemaatriksit leidma). Üks selline on nt corBrownian, mis leiab kahe liigi vahelise kauguse kui suhte „koos evolutsioneerutud aeg“ jagatud „kogu evolutsioneerutud aeg“. Loomaks vastavust andmemaatriksi ja fülogeneesipuu vahel (liigid ei pruugi neis olla samas järjestuses) tuleb argumendiks anda ka liiginimede vektor. NB! Antud tunnus peab olema tüüpi „character“, mitte tüüpi „factor“!

m4 <- gls(matur.L~age.mat, correlation=corBrownian(phy=puu, form=~liiginimed), data=andmed)

Kuna mudelis on vaid üks sõltumatu tunnus, siis saame selle p-väärtuse välja lugeda ka otse summary väljundist.

summary(m4)

## Generalized least squares fit by REML

## Model: matur.L ~ age.mat

## Data: andmed

## AIC BIC logLik

## 172.4416 174.7593 -83.22079

## Correlation Structure: corBrownian

## Formula: ~liiginimed

## Parameter estimate(s):

## numeric(0)

## Coefficients:

## Value Std.Error t-value p-value

## (Intercept) 26.416810 56.71983 0.4657421 0.6477

## age.mat 3.017715 3.78122 0.7980791 0.4365

## Correlation:

## (Intr)

## age.mat -0.727

## Standardized residuals:

## Min Q1 Med Q3 Max

## -0.21802874 -0.12696727 -0.03392951 0.05454821 0.63323838

## Residual standard error: 76.19028

## Degrees of freedom: 18 total; 16 residual

Kuivõrd vaatlusi (liike) on andmestikus vähe, siis on igal andmepunktil suur kaal ning antud väljundi juures teeb murelikuks jääkide jaotus – suurim jääk on vähimaga võrreldes absoluutväärtuselt väga suur (0.63 vs 0.22). Seega tuleb mudeli väljundisse suhtuda teatava ettevaatusega. Väljundist näeme, et toorandmete jooniselt nähtud „kindel“ seos ei ole statistiliselt oluline (p=0.44).

1. **Mitmemõõtmelised meetodid**
	1. **Peakomponentanalüüs**

Kui meie andmestikus on mitmeid korreleeruvaid tunnuseid siis võib nende samaaegne mudelisse kaasamine sõltumatute tunnustena olla problemaatiline. Samas ei ole sageli head põhjust, miks üht neist tunnustest teistele eelistada. Nii võib olla mõistlik luua antud tunnuste baasil uus tunnus, mis kannab endas korreleeruvate tunnuste ühisosa.

Vaatleme iiriste andmestikku iris, kus registreeritud kolme liigi (igast võetud 50 esindajat) õie suurust väljendavad tupp- ja kroonlehtede pikkus ja laius.

cor(iris[,1:4])

## Sepal.Length Sepal.Width Petal.Length Petal.Width

## Sepal.Length 1.0000000 -0.1175698 0.8717538 0.8179411

## Sepal.Width -0.1175698 1.0000000 -0.4284401 -0.3661259

## Petal.Length 0.8717538 -0.4284401 1.0000000 0.9628654

## Petal.Width 0.8179411 -0.3661259 0.9628654 1.0000000

Näeme, et kroonlehtede pikkus ja laius on tugevalt (ja positiivselt) korreleeritud. Nendega on korreleeritud ka tupplehtede pikkus. Tupplehtede laius on kolme ülejäänud tunnusega nõrgas negatiivses seoses. Ilmselt piisaks meile õie suuruse kirjeldamiseks siis kahest tunnusest.

Kuigi kõik tunnused on samades mõõtühikutes (sentimeetrites) ja ka nende varieeruvused on sarnased siis standardiseerime (lahutame maha keskväärtuse ja jagame läbi standardhälbega) tunnused enne peakomponentanalüüsi (valikuga scale.=T). Antud andmete korral ei oma see suurt tähendust. Olukorras, kus mõni tunnus on teistest palju suurema varieeruvusega ei oma aga standardiseerimata andmete põhjal teostatud peakomponentanalüüs tüüpiliselt mingit mõtet (esimeseks peakomponendiks valitakse sisuliselt lihtsalt seesama suurima varieeruvusega tunnus).

pca1 <- prcomp(iris[,1:4], scale.=T)

plot(pca1)



Ootuspäraselt kirjeldab, esimene peakomponent ära suurema osa (umbes kolmveerandi) andmestikus esinevast varieeruvusest (tulpade kõrguste kogusumma). Teise peakomponendi osaks jääb umbes veerand andmestiku koguvarieeruvusest.

pca1

## Standard deviations (1, .., p=4):

## [1] 1.7083611 0.9560494 0.3830886 0.1439265

## Rotation (n x k) = (4 x 4):

## PC1 PC2 PC3 PC4

## Sepal.Length 0.5210659 -0.37741762 0.7195664 0.2612863

## Sepal.Width -0.2693474 -0.92329566 -0.2443818 -0.1235096

## Petal.Length 0.5804131 -0.02449161 -0.1421264 -0.8014492

## Petal.Width 0.5648565 -0.06694199 -0.6342727 0.5235971

Näeme, et esimene peakomponent PC1 (esimeses veerus paiknevad kordajad) moodustatakse tõepoolest peaasjalikult eelnevalt räägitud tunnustekolmiku põhjal (nendel tunnustel on absoluutväärtuselt suuremad kordaja väärtused). Teine peakomponent PC2 (teises veerus paiknevad kordajad) on aga suuresti määratud just tupplehtede laiuse tunnusest. Kordajad vastavad muidugi standardiseeritud tunnustele.

Visualiseerime nüüd esimesed kaks peakomponenti ning lisame värvina ka liigitunnuse.

plot(pca1$x[,1], pca1$x[,2], col=as.numeric(iris[,5]))



Näeme, et kui kirjeldame õie suurust vaid esimese peakomponendi alusel, siis eristub üks liik selgelt ülejäänud kahest. Kui võtame appi ka teise peakomponendi siis on ehk veidi rohkem lootust omavahel eristada ka neid kahte. Samas peakomponentanalüüsis me liigitunnust ei kasutanud ja seega antud analüüsi tulemus ei ole otseselt mõeldud liikide eristamiseks.

* 1. **Diskriminantanalüüs**

Antud iiriste andmestiku kohta võime ka küsida, kuivõrd edukalt õnnestub liigi määramine õielehtede pikkuste ja laiuste alusel ja kuidas seda siis kõige paremini teha? Viimati tehtud joonise põhjal oleme saanud ligikaudse vastuse (tundub, et eristada enam-vähem saab). Täpse vastuse annab diskriminantanalüüs.

Diskriminantanalüüsil on erinevaid kujusid, ent meie vaatame siin lihtsaimat lineaarset diskriminantanalüüsi. Sisuliselt võime mõelda, et meil on mudel, kus sõltuv tunnus on kategooriline ning sõltumatud tunnused on pidevad. Seega sarnane olukord logistilisele regressioonile, ent diskriminantanalüüsil ei ole piirangut sõltuva tunnuse kategooriate arvule. Küll aga on piiranguks, et sõltumatud tunnused peavad olema normaaljaotusega (lineaarse diskriminantanalüüsi korral).

library(MASS)

m1 <- lda(Species~., data=iris)

m1

## Call:

## lda(Species ~ ., data = iris)

## Prior probabilities of groups:

## setosa versicolor virginica

## 0.3333333 0.3333333 0.3333333

## Group means:

## Sepal.Length Sepal.Width Petal.Length Petal.Width

## setosa 5.006 3.428 1.462 0.246

## versicolor 5.936 2.770 4.260 1.326

## virginica 6.588 2.974 5.552 2.026

## Coefficients of linear discriminants:

## LD1 LD2

## Sepal.Length 0.8293776 0.02410215

## Sepal.Width 1.5344731 2.16452123

## Petal.Length -2.2012117 -0.93192121

## Petal.Width -2.8104603 2.83918785

## Proportion of trace:

## LD1 LD2

## 0.9912 0.0088

Analüüsi väljundiks on antud juhul nelja sisendtunnuse teisendus kaheks tunnuseks LD1 ja LD2 (analoogiliselt peakomponentanalüüsiga, kus valisime ise kaks peakomponenti). Neid nn diskrimineerivaid tunnuseid leitakse sõltuva tunnuse kategooriate arvust ühe võrra vähem (või niipalju kui on sõltumatuid tunnuseid, kui see arv peaks olema väiksem). Väljundi lõpus olevad kaks arvu näitavad kui suure osa (protsentides) sõltuva tunnuse kategooriatevahelisest varieeruvusest leitud diskrimineerivad tunnused ära kirjeldavad. Näeme, et teine tunnus on sisuliselt kasutu.

plot(predict(m1)$x, col=as.numeric(iris[,5]))



Näeme, et tunnus LD1 ei erine oluliselt varem leitud tunnusest PC1 (suund on küll vastupidine, ent see ongi mõlemas analüüsis fikseerimata ehk ei oma tähtsust millises suunas väärtused kasvavad ja millises kahanevad).

1. **Liigilise koosseisu analüüsid**
	1. **Liigirikkuse hindamine**

Kui meie käsutuses on vaatluskohtades leitud liikide arvud koos vastavate isendite arvudega siis on selle põhjal võimalik hinnata leitud liikide koguarvu sõltuvust vaatluskohtade arvust. Vastava kõvera ekstrapoleerimisel saame hinnata leitavate liikide arvu olukorras, kus me sooritaksime vaatlusi rohkemates vaatluskohtades.

Pakett vegan võimaldab selliseid kõveraid koosluse andmetele sobitada. Liikide sagedused peavad olema andmemaatriksis nõnda, et iga rida on eraldi vaatluskoht ja iga veerg eraldi liik.

library(vegan)

Meil on kasutada kahlajate andmestik waders, kus 15 vaatluskohas on loendatud kokku 19 liigi esindajad. Vaatame esmalt kuidas registreeritud liikide arv vaatluskohtade lisandumisel kasvanud on.

m1 <- specaccum(waders)

m1

## Species Accumulation Curve

## Accumulation method: exact

## Call: specaccum(comm = waders)

## Sites 1.000000 2.000000 3.000000 4.00000 5.000000 6.000000 7.000000 8.000000 9.000000 10 11 12 13 14 15

## Richness 15.533333 17.790476 18.509890 18.80147 18.925075 18.974825 18.992852 18.998446 18.999800 19 19 19 19 19 19

## sd 2.895207 1.935761 1.271142 0.80732 0.480724 0.260867 0.127706 0.048634 0.014134 0 0 0 0 0 0

Näeme, et liigid on üldiselt hästi esindatud (või siis on vaatluskohad sobivalt valitud) ja juba viie vaatluskohaga on lootust üles leida kõik 19 liiki (keskmine 18.9, standardhälve 0.5). Eeldades, et kõigis vaatluskohtades on kõik leitud kahlajate liigid ka registreeritud, võime antud andmete põhjal ennustada kuimitu vaatluskohta oleks vaja vähemalt 20 liigi leidmiseks. Selleks sobitama andmetele mittelineaarse mudeli ja ennustame seejärel antud mudeli abil leitud liikide arvu.

m2 <- fitspecaccum(m1, model="arrhenius")

plot(1:30,predict(m2,1:30))



Näeme, et 30 vaatluskohta võiks olla piisav 20 liigi registreerimiseks. Kui andmestikus on aga registreeritud vaid levinumad (mitte kõik) liigid siis ei ole muidugi võimalik selliseid järeldusi teha.

* 1. **Koosluste võrdlemine**

Uurime nüüd koosluste võimalikku erinevust mõne vaatluskoha tunnuse alusel. Vaadeldava kahlajate andmestiku korral on abifailist võimalik näha, et vaatluskohtade tüübiks on, kas „coast“ (rannik) või „wetland“ (märgala). Tekitame vastavat tunnust sisaldava andmestiku.

sites <- data.frame(loc=rep(c("coast","wetland"), 8)[c(1:13,15:16)]) #lühiduse huvides tekitame ühe liigse väärtusega vektori ja siis jätame selle kaasamata

Seejärel teeme PERMANOVA mudeli kasutades funktsiooni adonis2 paketist vegan. Funktsioonis saab kasutada valemisüntaksit. Sageduste maatriks peab paiknema asukohti iseloomustavatest tunnustest eraldi.

adonis2(waders~loc, data=sites, method="bray")

## Permutation test for adonis under reduced model

## Terms added sequentially (first to last)

## Permutation: free

## Number of permutations: 999

## adonis2(formula = waders ~ loc, data = sites, method = "bray")

## Df SumOfSqs R2 F Pr(>F)

## loc 1 1.1366 0.26893 4.7821 0.001 \*\*\*

## Residual 13 3.0899 0.73107

## Total 14 4.2265 1.00000

Näeme, et eri tüüpi vaatluskohtades on ka kooslused erinevad (p=0.001), kusjuures kogu sagedusandmestiku varieeruvusest on seletatud u 27%.

**12.3 Koosluste visualiseerimine**

Teades, et vaatluskoha tüübil on mõju, üritame luua ordinatsiooni (koosluste ja vaatluskohtade väikesemõõtmelisse ruumi paigutamise protsessi nimetakase ordineerimiseks), mida oleks võimalik siis visualiseerida. Loome mittemeetrilise mitmemõõtmelise skaleerimise abil (*NMDS*) sagedusmaatriksi kahemõõtmelise esituse.

m3 <- metaMDS(waders)

m3

## Call:

## metaMDS(comm = waders)

## global Multidimensional Scaling using monoMDS

## Data: wisconsin(sqrt(waders))

## Distance: bray

## Dimensions: 2

## Stress: 0.08305045

## Stress type 1, weak ties

## Two convergent solutions found after 20 tries

## Scaling: centring, PC rotation, halfchange scaling

## Species: expanded scores based on ‘wisconsin(sqrt(waders))’

Funktsioon metaMDS üritab vaikimisi teha andmetega teisendused, mis garanteerivad võimalikult asjaliku ordinatsiooni. Antud andmestiku korral võeti algselt sagedusmaatriksi kõigist elementidest ruutjuur (vähendamaks väga arvukate liikide mõju) ning seejärel teostati nn Wisconsini topeltstandardiseerimine (iga maatriksi element jagatakse esimesel sammul läbi vastava veeru maksimumiga ja teisel sammul vastava rea summaga). Kuivõrd alla 0.1 jääv stressiskoor viitab, et skaleerimise tulemus on hea, siis visualiseerime selle tulemuse.

plot(m3, type="text")



Samale graafikule asetatakse nii liigid (S1—S19) kui ka vaatluskohad (A—O) ja seeläbi võime graafikult välja lugeda, millised liigid on millistele vaatluskohtadele iseloomulikud. Näeme, et vaatluskohad on klasterdunud kahte rühma. Vaid kohad D ja B on kahe klastri vahekohas.

Lisame nüüd juba olemasolevale graafikule ka vaatluskoha tüübi (seda me ordineerimisel ei kasutanud).

ordihull(m3, groups=sites$loc, draw="polygon", col="grey", label=T)



Näeme, et ordinatsioon kajastab vaatluskoha tüüp väga edukalt (eri tüüpi vaatluskohad paiknevad eri klastrites, ülekate klastrite vahel puudub).

1. **Veel kasulikku**
	1. **Informatsioonikriteerium AIC**

Sageli on meil samade andmete jaoks võimalik teha mitu erinevat mudelit, mis kõik tunduvad mõistlikud. Kui need mudelid on sellised, et üks on saadud teisest mudeli täiendamise teel (lisatud sõltumatuid tunnuseid) siis on neid mudeleid võimalik statistiliste testide abil võrrelda ja seeläbi otsustada, millist mudelit tuleks eelistada. Kui kaks mudelit on aga sellised, et kumbki pole teise täiendus (*non-nested models*, nt esimeses sisaldub tunnus A ja ei sisaldus tunnus B, teises sisaldub tunnus B ja ei sisaldu tunnus A ning tunnuseid A ja B polegi üldse mõistlik samasse mudelisse kaasata, sest need on tugevalt korreleeritud), siis saame appi võtta informatsioonikriteeriumi, milledest tuntum on AIC.

Alustuseks mitu hoiatust. Esiteks, me võime võrrelda AIC väärtusi vaid nende mudelite korral, milles on kasutatud sama sõltuvat tunnust (ja kui seda on teisendatud siis sama teisendust) ja täpselt samu andmeridu (muutub oluliseks puuduvate väärtuste korral andmestikus, nt kui mingil konkreetsel valimi punktil on tunnuse A väärtus on olemas, aga tunnuse B oma mitte, siis on meie ainsaks võimaluseks see vaatlus mõlemast mudelist välja jätta). Teiseks, AIC on välja töötatud juhuslikke faktoreid mittesisaldavate (üldistatud) lineaarsete mudelite jaoks. Seda saab küll kasutada ka segamudelite korral, ent nende puhul ei ole täpselt selge, milline on piisav erinevus kahe mudeli AIC väärtuse vahel, et üht teisele eelistada (küll aga on teada, et segamudelite korral omab see võrdlus üldse mõtet vaid olukorras, kus mudeli parameetrite hindamisel kasutatakse ML mitte REML (vaikimisi) metoodikat). Kolmandaks, AIC on samuti juhuslik (valmist sõltuv) ja seega ei võimalda väga vähe erinevad AIC väärtused üht mudelit teisele eelistada. Kuldreegel on, et erinevus, mis on vähem kui kaks ühikut, ei võimalda mitte midagi öelda ja erinevus, mis on rohkem kui kümme ühikut, annab selge eelistuse. Lõpuks, üldiselt on AIC alusel sobivaima mudeli otsimine ja statistilised testid kaks alternatiivi ja mõlemat samaaegselt ei kasutata (st. ei ole korrektne leida esmalt AIC alusel parimat mudelit ja seejärel hakata testima selles mudelis sisalduvate sõltumatute tunnuste statistilisi olulisusi (juhul kui me ei võtta arvesse, et esimesel sammul juba toimus mudeli selekteerimine)). Erandiks on olukord, kus AIC alusel leitud parimas mudelis ei tule ükski sõltumatu tunnus statistiliselt oluliseks – see on (kaudseks) tõestuseks nii sellele, et ka parim mudel ei ole hea mudel ja et sõltumatud tunnused tõepoolest mõju ei oma.

Vaatame juba tuttavat puude andmestikku ja küsime, kas puu tüve ruumala ennustamiseks on parem kasutada tüve läbimõõtu või tüve pikkust. Kuivõrd ruumala on silindrilise keha korral (mida puu tüvi ligikaudu on) leitav ristlõike pindala ja kõrguse korrutisena ning ristlõike pindala on leitav diameetri ruudu abil, siis on ilmselt õigem kõik tunnused logaritmida.

m1 <- lm(log(Volume)~log(Girth), data=trees)

m2 <- lm(log(Volume)~log(Height), data=trees)

AIC(m1)

## -42.21102

AIC(m2)

## 36.23131

Näeme, et esimene mudel on teisest mäekõrguselt üle (deltaAIC=78.0). Valimi suurusega korrekteeritud AIC väärtuse (AICc), ja mugavama väljundi saame paketi MuMIn funktsiooni model.sel abil.

library(MuMIn)

model.sel(m1,m2)

## Model selection table

## (Int) log(Grt) log(Hgh) df logLik AICc delta weight

## m1 -2.353 2.2 3 24.106 -41.3 0.00 1

## m2 -13.960 3.982 3 -15.116 37.1 78.44 0

## Models ranked by AICc(x)

Loetavuse huvides on konspektis väljundtabelist ära jäetud veerg, mis kajastab sõltuva tunnuse jaotust mudelis. Väljundist näeme, et antud suur ülekaal põhjustab olukorra, kus diameetrit sisaldav mudel saab kaaluks ühe ja kõrgust sisaldav mudel saab kaaluks nulli. Küll ei ütle selline mudelite võrdlus sugugi, et kui kaasaksime diameetrit sisaldavasse mudelisse ka kõrguse, siis ei oleks see mudel omakorda selgelt parem kui m1 (mõistagi antud juhul see täpselt nii ongi).

Vaatame ka meile tuttavat kapsaste andmestikku paketist MASS. Olgu meil üheks kandidaatmudeliks vaid pea kaalu sisaldav mudel, teiseks aga kultivaari sisaldav mudel.

m3 <- lm(VitC~HeadWt, data=cabbages)

m4 <- lm(VitC~Cult, data=cabbages)

model.sel(m3,m4)

## Model selection table

## (Intrc) HedWt Cult df logLik AICc delta weight

## m3 77.57 -7.567 3 -206.343 419.1 0.00 0.761

## m4 51.50 + 3 -207.502 421.4 2.32 0.239

Models ranked by AICc(x)

Näeme, et pea kaalu sisaldav mudel on ehk veidi informatiivsem.

Tunnuste „tähtsuse“ alusel järjestamisel kasutatakse sageli viisi, kus sobitatakse kõikvõimalikud mudelid ning seejärel liidetakse iga tunnuse jaoks kokku seda sisaldavate mudelite kaalud.

m5<-lm(VitC~HeadWt+Date+Cult, data=cabbages, na.action="na.fail")

dredge(m5)

## Fixed term is "(Intercept)"

## Global model call: lm(formula = VitC ~ HeadWt + Date + Cult, data = cabbages, na.action = "na.fail")

## ---

## Model selection table

## (Intrc) Cult Date HedWt df logLik AICc delta weight

## 8 63.33 + + -4.412 6 -190.420 394.4 0.00 0.772

## 6 67.93 + -5.652 4 -194.069 396.9 2.44 0.228

## 4 49.95 + + 5 -198.610 408.3 13.91 0.001

## 5 77.57 -7.567 3 -206.343 419.1 24.69 0.000

## 2 51.50 + 3 -207.502 421.4 27.01 0.000

## 7 75.18 + -6.904 5 -205.439 422.0 27.57 0.000

## 3 56.40 + 4 -218.601 445.9 51.50 0.000

## 1 57.95 2 -223.495 451.2 56.78 0.000

## Models ranked by AICc(x)

importance(dredge(m5))

## Fixed term is "(Intercept)"

## Cult HeadWt Date

## Sum of weights: 1.00 1.00 0.77

## N containing models: 4 4 4

Kui tunnuste järjestamiseks on eelnev teguviis aktsepteeritav siis parima mudeli leidmiseks ei ole kõikvõimalike mudelite läbi proovimine üldiselt sugugi hea mõte. Kui mudeldusprotsessi eesmärgiks on ennustamine siis kasutatakse küll sageli kõikvõimalike mudelite põhjal leitud keskmistatud mudelit.

model.avg(dredge(m5))

## Fixed term is "(Intercept)"

## Call:

## model.avg(object = dredge(m5))

## Component models:

## ‘123’ ‘13’ ‘12’ ‘3’ ‘1’ ‘23’ ‘2’ ‘(Null)’

## Coefficients:

## (Intercept) Cultc52 Dated20 Dated21 HeadWt

## full 64.37068 9.959991 -0.9376165 3.235072 -4.691449

## subset 64.37068 9.960032 -1.2141008 4.189030 -4.694916

Keskmistatud mudelis on vastavad parameetrite väärtused mudeli kaalude abil keskmistatud. Kui arvestada vaid neid mudeleid, kus vastav sõltumatu tunnus sisaldub siis on tulemuseks real „subset“ leiduvad parameetrite hinnangud, kui kõiki mudeleid (kui tunnust mudelis pole siis on tema kordaja 0), siis real „full“ leiduvad parameetrite hinnangud. Enamasti kasutatakse just neid.

* 1. **Bayesi statistika**

Bayesi statistikas tehakse veidi teistsuguseid eeldusi – nimelt eeldatakse, et parameetrid ei ole mitte konstandid vaid hoopis juhuslikud suurused (ehk siis varieeruvust omavad). Samas ei ole vähevarieeruv juhuslik suurus muidugi konstandist märkimisväärselt erinev. Kui tavastatistikas üritame konstanti (nt üldkeskmist) hinnata siis anname lisaks tavahinnangule (valimikeskmine) üldjuhul ka usaldusvahemiku. Bayesi statistikas on nn usaldusväärtusintervall ehk usutavusvahemik (*credible intervall*) kohustuslik, sest juhuslikku suurust vaid ühe väärtuse abil kirjeldada ei ole üldiselt mõistlik. Seepärast loetaksegi Bayesi statistika eeliseks, et varieeruvuse hindamine on alati osa protsessist samas kui tavastatistikas võib see oluline samm tahaplaanile jääda ning esile kerkib vaid dihhotoomia „on nullist erinev“/“ei ole nullist erinev“.

Kui me kasutame parameetri jaoks mitteinformatiivset eeljaotust, siis ei ole põhjust eeldada, et selle parameetri järeljaotuse keskmine erineks märkimisväärselt tavastatistikaga saadud parameetri hinnangust. Bayesi statistika eripäraks on asjaolu, et järeljaotus leitakse simuleerimise teel ning me peame olema täiesti kindlad, et simulatsiooniahel on olnud „piisavalt“ pikk. Seega on esimene samm alati iteratsioonide hulga piisavuse hindamine. Üldjuhul saab seda hinnata tekitades mitu simulatsiooniahelat ning võrreldes eri ahelatest saadavate jaotuste kokkulangevust.

Kaasaegne Bayesi statistika tegemine käib tarkvara Stan abil. Ris võimaldab selle kasutamist pakett rstan. Selle paketi installeerimine ei ole aga sama lihtne nagu tavalistel Ri pakettide korral. Midagi hullu ka ei ole – täpse juhise leiad kui kirjutad otsingumootorisse „RStan Getting Started“ ja järgid esimese vastena leitava veebilehe juhiseid.

library(rstan)

Vaatleme uuesti paketist ade4 pärit sisalikuliste andmestikku lizards ning tekitame vajalikud objektid.

library(ade4)

data(lizards)

library(ape)

puu <- read.tree(text=lizards$hprA)

andmed <- lizards$traits

andmed$liiginimed <- rownames(andmed)

covmat=vcv.phylo(puu) #fülogeneesipuul põhinev kovariatsioonimaatriks

library(brms)

Viimase sammuna laadisime tööle paketi, mille abil on mugav rstan paketti kasutada – sisendina antud Ri valemisüntaksist tehakse valmis vajalik jooksutatav mudel.

m5 <- brm(matur.L~age.mat+(1|gr(liiginimed, cov=A)), data=andmed, family=gaussian(), data2=list(A=covmat), control=list(adapt\_delta=0.999, max\_treedepth=20), iter=30000)

Kuna andmestikus on vähe liike ning teame juba varasemast, et üks liik on ülejäänutega halvas kooskõlas (väga suur mudeli jääk) siis on juba varakult tehtud täiendavad valikud, tagamaks ahelate parem koonduvus. Bayesi statistika abil mudelite sobitamine on reeglina küllalt aeganõudev. Antud koodile vastava mudeli sobitamine kestab umbes viis minutit. Jooksutamise käigus antakse väljundina muuhulgas hoiatus

## There were xx divergent transitions after warmup

mis viitab, et asjad ei ole hästi (liigi „Pg“ halb kooskõla ülejäänutega). Liigi kõrvalejätmise abil oleks võimalik antud koonduvusprobleem lahendada.

summary(m5)

## Family: gaussian

## Links: mu = identity; sigma = identity

## Formula: matur.L ~ age.mat + (1 | gr(liiginimed, cov = A))

## Data: andmed (Number of observations: 18)

## Samples: 4 chains, each with iter = 30000; warmup = 15000; thin = 1; total post-warmup samples = 60000

## Group-Level Effects:

## ~liiginimed (Number of levels: 18)

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## sd(Intercept) 3.04 1.57 0.20 6.08 1.00 3665 13986

## Population-Level Effects:

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## Intercept 10.99 16.88 -20.43 46.10 1.00 14145 17480

## age.mat 4.27 1.41 1.32 6.89 1.00 11221 15938

## Family Specific Parameters:

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## sigma 9.55 4.20 3.30 18.39 1.00 3316 9192

Veerg Rhat iseloomustab ahelate segunemist. Tehtavate iteratsioonide kasvatamise abil tuleb saavutada olukord, kus Rhat erineb ühest minimaalselt. Näeme, et sedakorda on regressioonisirge tõusu usutavusvahemikuks (1.32; 6.89). Kuna vahemik ei sisalda nulli siis on see viide sellele, et seos tunnuste vahel on olemas.