Lineaarsed mudelid
7. praktikum

Usaldusellips(oid)

Näide 1

Vaatame lihtsat regressioonanalüüsi mudelit

```
# Genereerime andmed
      RNGkind(sample.kind = "Rejection")
      set.seed(1)
      x=runif(100,0,6)
      y=2+1.9*x+rnorm(100)
      and=data.frame(x,y)

# Hindame mudeli
      m=lm(y~x, data=and)
      summary(m)
```

Huvitagu meid näiteks küsimus, kas nii vabaliige kui ka sirge tõus võiksid mõlemad olla võrdsed 2-ga. Võime sellele küsimusele muidugi proovida leida vastust tavaviisil:

```
> confint(m)
                2.5 %   97.5 %
(Intercept) 1.412247 2.229102
x           1.835153 2.068962
```

Kust näeme, et 2 jääb kenasti mõlema parameetri korral 95%-usaldupiiridesse. Arvestades muidugi, et vaatame kahte parameetrit korraga, peaksime kasutama Bonferroni korrektsiooni (ühe usalduspiiri jaoks lubatud vea tõenäosuse 0,05 asemel lubama vea tõenäosust 0,025):

```
> confint(m, level=0.975)
               1.25 %  98.75 %
(Intercept) 1.352172 2.289177
x           1.817957 2.086157
```

Mis teeks usalduspiire muidugi vaid laiemaks. Proovime aga antud hüpoteesipaari koos testida F-testi abil:

$$\left(\boldsymbol{\Lambda}^T\hat{\boldsymbol{\beta}} - \boldsymbol{\Lambda}^T\boldsymbol{\beta}_0\right)^T \left(\boldsymbol{\Lambda}^T(\mathbf{X}^T\mathbf{X})^-\boldsymbol{\Lambda}\cdot\mathrm{MSE}\right)^{-1}\left(\boldsymbol{\Lambda}^T\hat{\boldsymbol{\beta}} - \boldsymbol{\Lambda}^T\boldsymbol{\beta}_0\right)/\mathrm{rank}(\boldsymbol{\Lambda}) \overset{H_0}{\sim} F_{\mathrm{df}_1=\mathrm{rank}(\boldsymbol{\Lambda});\mathrm{df}_2=n-\mathrm{rank}(\mathbf{X})}.$$

Ehk R-i programmina:

```
Lambda=rbind(c(1,0), c(0,1))
stat=t(Lambda%*%coef(m)-c(2,2))%*%
          (solve(Lambda%*%vcov(m)%*%t(Lambda))/2)%*%
                                    (Lambda%*%coef(m)-c(2,2))
stat
1-pf(stat, df1=2, df2=98)
```

Mis annab meile tulemuseks pisikese p-väärtuse (järelikult pole võimalik, et nii vabaliige kui ka sirge tõus oleksid 2-d).

Mitut hüpoteesi saab korraga testida (kuidagimoodi) ka estimable-käsu abil. Viimane käsk küll eeldab, et teame tegelikku vaatluste hajuvust (ignoreerib seda, et me tegelikult hindame jääkide dispersiooni):

```
Lambda=rbind(c(1,0), c(0,1))

library(gmodels)
estimable(m, Lambda, beta0=c(2,2),  joint.test=TRUE)
```

Märkus: sama tulemuse saaksime ise arvutusi tehes siis, kui kasutaksime valemit:

$$\left(\boldsymbol{\Lambda}^T\hat{\boldsymbol{\beta}}-\boldsymbol{\Lambda}^T\boldsymbol{\beta}_0\right)^T\left(\boldsymbol{\Lambda}^T(\mathbf{X}^T\mathbf{X})^-\boldsymbol{\Lambda}\sigma^2\right)^{-1}\left(\boldsymbol{\Lambda}^T\hat{\boldsymbol{\beta}}-\boldsymbol{\Lambda}^T\boldsymbol{\beta}_0\right)\overset{H_0}{\sim}\chi^2_{\mathrm{df}=\mathrm{rank}(\boldsymbol{\Lambda})}$$

ja asendaksime selles valimis jääkide dispersiooni jääkide dispersiooni hinnanguga:

```
1-pchisq(stat*2, df=2)
```

Seega antud juhul on näha, et ise arvutades on võimalik saada vahel ka täpsemaid tulemusi võrreldes R-i standardfunktsioonidega.

Sama järelduseni oleksime võinud jõuda muidugi ka parameetritele joonistatud järgmise võrratuse poolt defineeritud usaldusellipsoidi:

$$\left(\boldsymbol{\Lambda}^T\hat{\boldsymbol{\beta}}-\boldsymbol{\Lambda}^T\boldsymbol{\beta}\right)^T\left(\boldsymbol{\Lambda}^T(\mathbf{X}^T\mathbf{X})^-\boldsymbol{\Lambda}\cdot\mathrm{MSE}\right)^{-1}\left(\boldsymbol{\Lambda}^T\hat{\boldsymbol{\beta}}-\boldsymbol{\Lambda}^T\boldsymbol{\beta}\right)/\mathrm{rank}(\boldsymbol{\Lambda})\leq \leq f_{1-\alpha;\mathrm{df}_1=\mathrm{rank}(\boldsymbol{\Lambda});\mathrm{df}_2=n-\mathrm{rank}(\mathbf{X})}$$

Kuidas ellipseid arvuti abil joonistada? Ühte võimalust kirjeldatakse näiteks siin: http://www-1.ms.ut.ee/mart/linmud2021/EllipsiJoonistamisest.pdf

Ülatoodud arvutuseeskiri R-i programmina näeks välja siis selline:

```
omavek=   eigen(vcov(m))$vectors
omavaart= eigen(vcov(m))$values

r= seq(0, 2*pi, length=1000)

k= sqrt(qf(0.95,2,98))
a= omavek%*%diag(sqrt(omavaart))%*%t(omavek)
abi= sqrt(2)*a%*%(k*rbind(sin(r), cos(r)))+coef(m)

plot(abi[1,], abi[2,], type="l")
polygon(abi[1,], abi[2,], col="gray90", border="black", lwd=3)

points(2,2, pch=20, col=2, cex=2)
```

Näeme, et punkt 2,2 ei asu 95%-usaldusellipsis, seega võime 95%-kindlusega väita, et vabaliige ja sirge tõus ei saa samaaegselt olla 2-d.

Usaldusellipsit mudeli parameetritele saab leida ka lisamooduli ellipse abil, näiteks lisa oma joonisele ellipse-käsu abil leitud usaldusellips (punasega, et näha oleks):

```
library(ellipse)
lines(ellipse(m, npoint=1000), col="red2", lwd=2)
```

Näide 2

Vaatame veel ühte (veidi realistlikumat) olukorda, kus mitme hüpoteesi koos testimine võib kasulikuks osutuda.

Soovime näiteks kirjeldada y-tunnuse ja x-tunnuse vahelist seost. Teeme mõned eksperimendid erinevate x-tunnuse väärtuste korral (mõõdame, millise y-tunnuse väärtuse tulemuseks saame – aga mõõtmised on paraku ikka mõõtmisveaga). Koostame seose kirjeldamiseks lineaarse mudeli ja joonistame tulemuse endale silmailuks välja:

```
print(load(url("http://www-1.ms.ut.ee/mart/linmud2021/kambatest.RData")))

naide[1:3,]
m=lm(y~poly(x,3, raw=TRUE),data=naide)
plot(naide$x, naide$y, pch=20, col="gray60")

xx=seq(0, 200, length=200)
abi=predict(m, data.frame(x=xx), interval="confidence")
lines(xx, abi[,1], lwd=3)
lines(xx, abi[,2], lty=2)
lines(xx, abi[,3], lty=2)
```

Oletame, et keerukate teoreetiliste arvutuste tulemusel on õnnestunud meil välja rehkendada milline peaks y-tunnuse täpne väärtus teatud x-tunnuse väärtuste korral. Näiteks teooria järgi x=30 korral peaks täpne (mõõtmisveata) y-tunnuse väärtus olema 6,1; x=90 korral peaks y=23,3 ja x=150 korral peaks y=31.

Kas antud teooria peab paika?

Võiksime muidugi vaadata oma mudeli prognoose nende kolme x-tunnuse väärtuse korral:

```
# Teooria: E(y|x=30)=6,1
predict(m, data.frame(x=30), interval="confidence")
# Teooria: E(y|x=90)=23,3
predict(m, data.frame(x=90), interval="confidence")
# Teooria: E(y|x=150)=31
predict(m, data.frame(x=150), interval="confidence")
```

Näeme, et teoreetiliselt ennustatud väärtused jäävad alati leitud usaldusintervallidesse – kõik võiks olla seega kontrollitava teooriaga nagu korras…

Võime teha ka kolm testi, kontrollida hüpoteese

H0: E(y|x=30)=6 ,1 vs H1: E(y|x=30)≠6,1;
H0: E(y|x=90)=23,3; vs H1: E(y|x=30)≠23,3;
H0: E(y|x=150)=31; vs H1: E(y|x=30)≠31;

Selliseid hüpoteese võime kontrollida näiteks estimable käsu abil:

```
library(gmodels)

# Variant 1:
estimable(m, c(1,30, 30^2, 30^3), beta0=6.1)
estimable(m, c(1,90, 90^2, 90^3), beta0=23.3)
estimable(m, c(1,150, 150^2, 150^3), beta0=31)

# Variant 2:
estimable(m, rbind(c(1,30, 30^2, 30^3),
                   c(1,90, 90^2, 90^3),
                   c(1,150, 150^2, 150^3)),
          beta0=c(6.1, 23.3, 31) )
```

Näeme, et kõigil juhtudel saadud p-väärtused on suuremad kui 0,05 (kuigi paaril juhul üsna lähedal 0,05-le). Samas peaksime paljude testide korral kasutama mõnda mitmese testimise meetodit – näiteks Bonferroni meetodit – ja võrdlema saadud p-väärtuseid hoopis kriitilise väärtusega 0,05/3, millest kõik leitud p-väärtused jäävad üsna kaugele.

Mis saab aga siis, kui nõuame, et kõik mainitud hüpoteesid peaksid kehtima samaaegselt (vaid siis on ju kontrollitav teooria õige!)? Teeme siis meid huvitava küsimuse kontrollimiseks ühe kambatesti:

H0: E(y|x=30)=6 ,1 ja E(y|x=90)=23,3 ja E(y|x=150)=31
vs
H1: vähemalt mõni neist tingimustest ei kehti

Kambatesti võime teha kas estimable-käsu abil (märka märksõna joint.test=TRUE):

```
estimable(m, rbind(c(1,30, 30^2, 30^3),
                   c(1,90, 90^2, 90^3),
                   c(1,150, 150^2, 150^3)),
          beta0=c(6.1, 23.3, 30),
          joint.test=TRUE )
```

või veidi täpsema tulemuse saame ise käsitsi arvutades (ise arvutades arvestame, et jääkide hajuvus on hionnatud):

```
# Kambatest korrektsemalt:

Lambda=rbind(c(1,30, 30^2, 30^3),
                   c(1,90, 90^2, 90^3),
                   c(1,150, 150^2, 150^3))
beta0=c(7, 24.5, 30.8)
stat=t(Lambda%*%coef(m)-beta0)%*%
    (solve(Lambda%*%vcov(m)%*%t(Lambda))/3)%*%
       (Lambda%*%coef(m)-beta0)
stat
1-pf(stat, df1=3, df2=96)
```

Näeme, et üheskoos kambakat tehes õnnestub nullhüpotees kummutada – kontrollitav teooria ei ole vastavuses vaatlusandmetega.

Sarnast analüüsi võib vaja minna ka siis, kui soovime näiteks võrrelda puude kasvukiiruseid Eestis puude kasvukiirustega Lõuna-Rootsis. Leiame ühe teadusartikli, kus on mainitud teatud vanusega puude suuruseid Rootsis (leitud hiigelsuurte valimite pealt – neil on ju selliste uuringute jaoks raha jalaga segada). Mõõdame ka Eestis mõnede puude vanused ja suurused. Kas Eesti andmete pealt leitud puude kasvukõver on vastavuses Lõuna-Rootsis tehtud mõõtmistega või mitte?

Ülesanne 1

Muudame näites 1 kasutatud andmete genereerimise mehhanismi veidi. Mille poolest erinevad mudeli parameetritele joonistatud usaldusellipsid variant 1 ja variant 2 korral? Oskad sa arvata, miks antud erinevus tekib? Ehk suudad välja tuua mõne tingimuse, millal võiksid usaldusellipsi teljed ühtida koordinaattelgedega – koos põhjendusega?

Variant 1

```
x=runif(100,0,6)
x=x-mean(x)
y=2+1.9*x+rnorm(100)
and=data.frame(x,y)
m=lm(y~x, data=and)
```

Variant 2

```
x=runif(100,0,6)+10
y=2+1.9*x+rnorm(100)
and=data.frame(x,y)
m=lm(y~x, data=and)
```

Ülesanne 2

Genereerime teise andmestiku – dispersioonianalüüsi mudel. Joonista usaldusellips mudeli 2. ja 3. parameetri jaoks (grupp 2 mõju ja grupp 3 mõju). Selgita, kuidas usaldusellipsi pealt vaadata, kas mõlemad mõjud võivad olla ka 0-id. Anna ka oma järeldus kasutades joonist: kas need mõjud võivad olla samaaegselt nullid?

```
set.seed(1)
grupp=rep(1:3, c(10, 100, 1000))
y=2.6*(grupp==1)+
     2.7*(grupp==2)+2.3*(grupp==3)+rnorm(length(grupp))
m2=lm(y~factor(grupp))
summary(m2)
```