Praktikum 7
R – logistiline regressioon

1.

¤) Salvestage arvutisse Eesti puude andmestik *R*-i failina:

http://ph.emu.ee/~ktanel/DK_0007/puud.rda

¤) Avage *R* ja käivitage lisamoodul *Rcmdr*.

¤) Võtke *R*-is kasutusele salvestatud puude andmestik

– *R Commanderis* käsud *Data -> Load data set …*

– või skripti aknas käsk `load`, **näiteks**

```
load("C:/Documents and Settings/Tanel/DK_0007/puud.rda")
```

¤) Puude andmestikus vastab üks rida ühele puule (st, et uurimisobjektiks on puu), veergudes on vastavalt:

veerus nimega 'A' puu vanused aastates;

veerus 'D' puu diameetersentimeetrites;

veerus 'H' puu kõrgus meetrites;

veerus 'ARENGUKL' puu arenguklass väärtustega A – lage, N – noorendikud, L – latimets (noorendikust järgmine), K – keskealised, V – valmiv, Y – küps, S – selguseta, – puuduv väärtus;

veerus 'PE' puu liik (HB – haab, KS – kask, KU – kuusk, LV – hall lepp, …);

veerus 'KKT' kasvukohatüüp (AN – angervaksa, JM – mustika, …)[1];

…

1.1. Puu raieküpsuse prognoosimine vanuse abil (logistiline regressioon)

Legend. Algajat kasumiahnet metsameest huvitab, millal tema mets valmis saab. Paraku ei suuda ta kuidagi mõista eksperte ja nende hinnanguid metsa küpsusele. Tema tahab lihtsat, ignorandist linnamehele mõistetavat otsustuseeskirja. Siinkohal tuleb appi **logistiline regressioon**.

¤) Esmalt tuleb andmestikku tekitada uus 0-1-tüüpi tunnus, mis näitab,
kas mets on raieküps (uue tunnuse väärtus 1, kui ARENGUKL="Y")
või mitte (uue tunnuse väärtus 0, kui ARENGUKL≠"Y").

Selleks tuleb skripti-aknasse sisestada käsk ('mature' saab olema uue loodava veeru/tunnuse nimi)

```
puud$mature = 1*(puud$ARENGUKL=="Y")
```

> Alternatiivina võib uue tunnuse tekitada ka *R Commanderi* menüüde abil:
> *Data -> Manage variables in active dataset -> Compute new variable...*
> Lahtrisse '*Expression to compute*' trükkige (või kopeerige-kleepige) käsk
> `1*(ARENGUKL=="Y")`

[1] Kasvukohatüüp (ka metsakasvukohatüüp) on mullastikult ja taimestikult ühtlane metsaala. Nimetus tuleb enamesineva taime järgi (näiteks kasvab naadi kasvukohas väga palju naate, jänesekapsa kasvukohatüübis jänesekapsaid jne) – http://et.wikipedia.org/wiki/Kasvukohatüüp.
Täpsemalt vt näiteks http://www.hot.ee/sinumets3/sinumets03-08.pdf.

¤) Edasiste käskude lihtsama esituse huvides võiks puude andmestiku määrata vaikimisi andmestikuks ka skripti-akna tarvis:

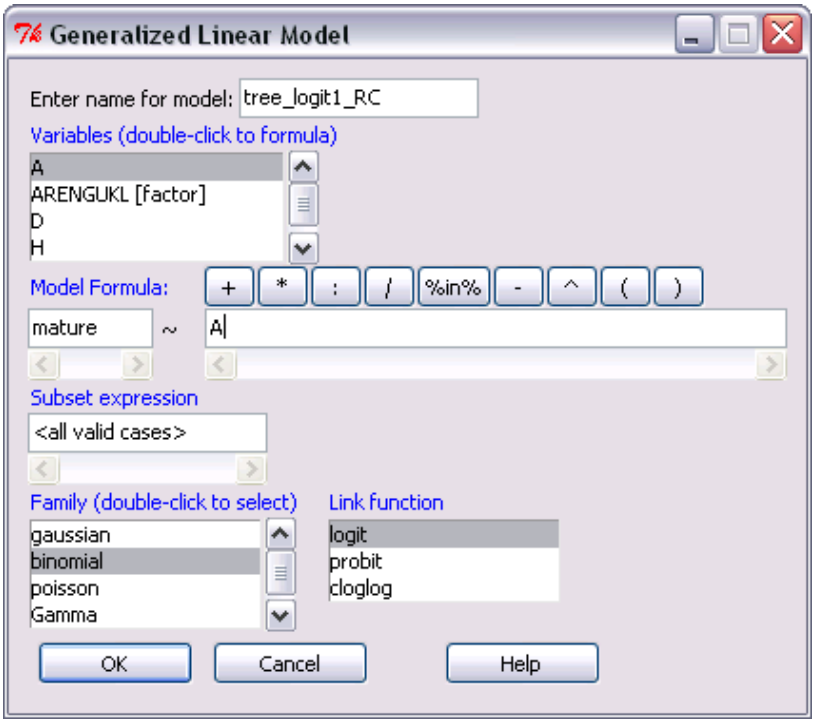
```
attach(puud)
```

¤) Püüame nüüd prognoosida puu raieküpsuse tõenäosust vanuse abil.

Logistilise regressiooni teostamiseks nö käsurealt (mõistlikuma tulemuse saamiseks jätame välja üksikud puud, kelle vanusena on kirjas 0):

```
tree_logit1 = glm(mature ~ A, family=binomial(), data=puud)
summary(tree_logit1)
```

> Sama mudeli saab tellida ka *R Commanderi* menüüdest (uut tunnust 'mature' sisaldava andmestiku *R Commanderis* kasutamiseks võib osutuda vajalikuks selle andmestiku uuesti *R Commanderi* vaikimisi andmestikuks määramine).
>
> *Statistics -> Fit models -> Generalized linear model*



Väljavõtt tulemustest:

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.558268   0.070342  -79.02   <2e-16 ***
A            0.053054   0.000826   64.23   <2e-16 ***
```

Vanusele vastav positiivne regressioonikordaja näitab, et puude vanuse kasvades suureneb ka nende raieküpsuse tõenäosus. Kusjuures eksponent regressioonikordajast,

$$\exp(0{,}053) = 1{,}054,$$

hindab šansside suhet *OR* (*odds ratio*; näitab raieküpsuse šansi suurenemist puu vanuse suurenemisel ühe aasta võrra).

R-i poolt hinnatud logistiline prognoosivõrrand avaldub hinnatud parameetrite kaudu kujul

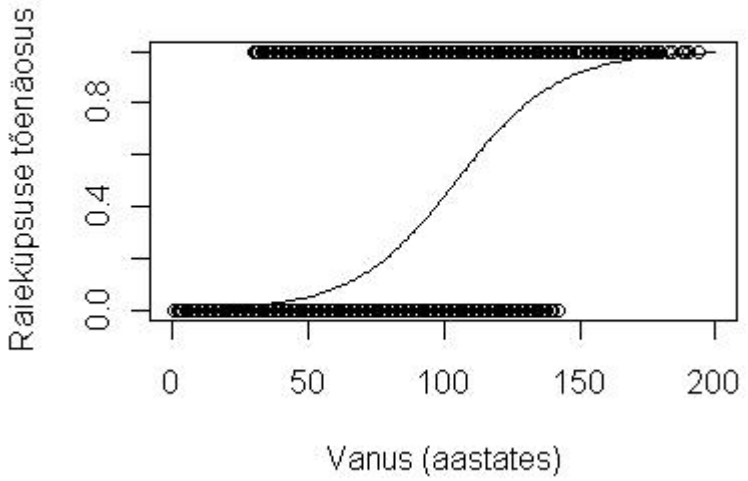
$$P(\text{mature}=1 \mid A) = \exp(-5{,}558 + 0{,}053 \times A) / (1 + \exp(-5{,}558 + 0{,}053 \times A))$$

(prognoositakse tõenäosust puu raieküpsuseks teada oleva vanuse korral).

Tulemused graafiliselt (laseme *R*-l prognoosida raieküpsuse tõenäosust vanuste 0-200 korral):

```
x = seq(0,200)
y = predict(tree_logit1, data.frame(A=x), type="response")
plot(x, y, type="l", xlab="Vanus (aastates)", ylab="Raieküpsuse tõenäosus")
points(A, mature)
```

Viimane rida eelnevas programmis lisab joonisele punktidena tegelikud (empiirilised) väärtused.



Veel võimalusi illustreerimaks logistilise regressioonanalüüsi tulemusi:

```
par(mfrow=c(2,1))
hist(A[mature==1], xlim=c(0,200))
hist(A[mature==0], xlim=c(0,200))
par(mfrow=c(1,1))
```

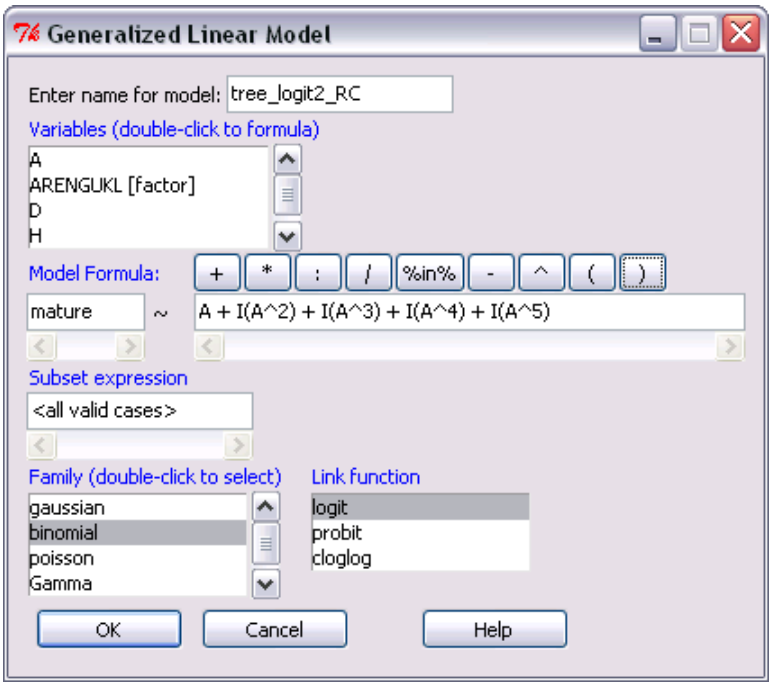
Märkus.
Metsa raieküpsuse saavutamise aeg on ilmselt ülehinnatud (80%-lise raieküpsuse saavutamiseks kulub prognoosi kohaselt ligikaudu 125 aastat) – põhjuseks see, et raieküpsuse saavutanud mets koheselt ka raiutakse, mistõttu nende andmed (raieküpsus teatud vanuses) andmebaasis ei kajastu. Alles jäävad vaid metsad, mis mingil põhjusel pole vaatamata suurele vanusele raieküpseks saanud ja ehk ka üksikud kaitsealused metsad.

1.2.

Kuna linnamees soovib oma metsa raieküpsust täpsemalt prognoosida, üritab ta teha oma logistilise regressiooni mudelit keerukamaks:

```
tree_logit2 = glm(mature~A+I(A^2)+I(A^3)+I(A^4)+I(A^5),family=binomial(),data=puud)
summary(tree_logit2)
```

Sama *R Commanderis*:

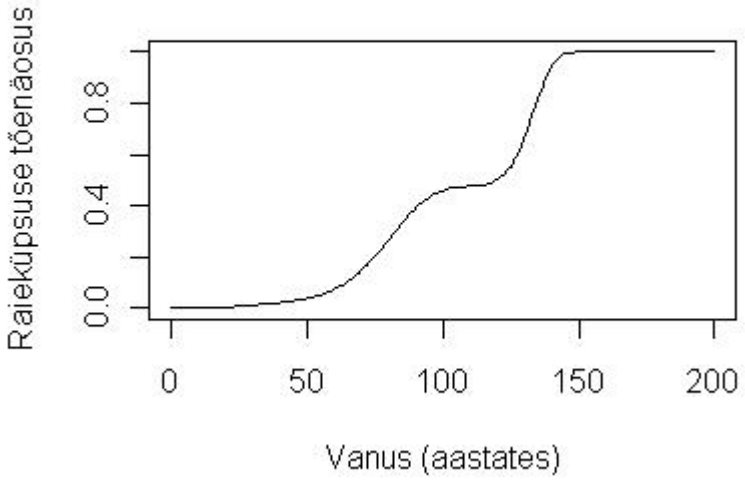


Raieküpsuse prognoos graafiliselt:

```
x = seq(0,200)
y = predict(tree_logit2, data.frame(A=x), type="response")
plot(x, y, type="l", xlab="Vanus (aastates)", ylab="Raieküpsuse tõenäosus")
```

Millega on seletatav jõnks graafikul?

Vastus – tegu on vanusevahemikuga, millal raiutakse metsi väga intensiivselt (alles on jäänud vaid mingil põhjusel raieküpsust mitte saavutanud puud).



1.3.

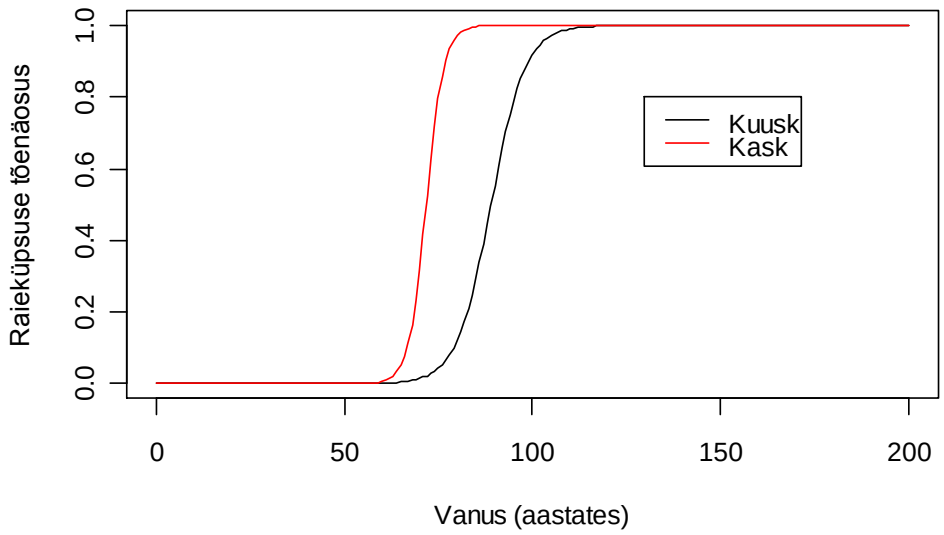
Siiani analüüsisime kõiki puuliike koos, samas on loomulik eeldada, et eri liigid saavutavad raieküpsuse erineval ajal. Seega võiks püüda hinnata erinevat mudelit iga puuliigi tarvis …

```
tree_logit3 = glm(mature~A*factor(PE),family=binomial(),data=puud)
```

Analüüsi tulemusena saadud hoiatus tähendab, et osade puuliikide tarvis ei olnud võimalik tellitud regressioonivõrrandit hinnata. Andmestikus suurel hulgal esindatud puuliikidega ei tohiks aga probleeme olla.

Joonistame hinnatud mudeli abil graafikud kahe puuliigi – kuuse ja kase jaoks:

```
x = seq(0,200)
yKU = predict(tree_logit3, data.frame(A=x, PE="KU"), type="response")
yKS = predict(tree_logit3, data.frame(A=x, PE="KS"), type="response")
plot(x, yKU, type="l", xlab="Vanus (aastates)", ylab="Raieküpsuse tõenäosus")
lines(x, yKS, col="red")
legend(130, 0.8, c("Kuusk", "Kask"), col=c("black","red"), lty=1)
```



Tulemus on üsna ootuspärane – kased saavutavad raieküpsuse varem kui kuused. Seejuures on kaskede raieküpsus vanusega pisut täpsemalt määratav (logistilise kõvera tõus on järsem).

1.4.

Logistilist regressiooni võib kasutada ka vaatlusandmete lahterdamiseks: kui puu on 75 aastat vana, siis tõenäoliselt on ta raieküps, kui aga puu vanus on 60 aastat, siis arvatavasti ta veel raieküps ei ole.

Et mistahes prognoosid on harva täiesti täpsed, võib vigu esineda ka taolisel lahterdamisel. Prognostiliste testide täpsust kirjeldatakse sageli kahe suurusega – testi tundlikkusega ja spetsiifilisusega.

Testi **tundlikkus** (*sensitivity*) näitab, kui suure tõenäosusega tuvastab test toimunud sündmuse (a'la kui suure tõenäosusega leitakse haigel indiviidil haigus).

Testi **spetsiifilisus** (specificity) näitab, kui suure tõenäosusega tuvastab test õigesti sündmuse mittetoimumise (a'la terve indiviid prognoositakse terveks).

Näiteks puude raieküpsuse analüüsil on loomulik otsustada, et kui mudel prognoosib raieküpsuse tõenäosuseks üle 0,5, siis loeme puu (metsa) raieküpseks.

Uurimaks sellise testi tundlikkust ja spetsiifilisust, tuleb konstrueerida tabel, mis näitaks, kui täppi taolise reegli alusel tehtud otsustused lähevad.
Järgnevas programmis defineeritud muutuja `kyps` sisaldab kõigi andmestikku kuuluvate puude raieküpsuse prognoose mudelist `tree_logit3`.
Semikoolonit kasutatakse eraldamaks ühte ritta kirjutatud erinevaid käske.

```
kyps=predict(tree_logit3, type="response")
tabel=table(kyps>0.5, mature); tabel
spetsiifilisus = tabel[1,1]/sum(tabel[,1])
tundlikkus = tabel[2,2]/sum(tabel[,2])
spetsiifilisus; tundlikkus
```

```
> tabel
       mature
           0    1
 FALSE 21912 1002
 TRUE    532 3209
```

```
> spetsiifilisus; tundlikkus
[1] 0.9762966
[1] 0.7620518
```

Kõigist raieküpsetest puudest prognoositakse raieküpseks 3209 ja mitteraieküpseks 1002. Järelikult on puu vanuse ja liigi alusel õigesti raieküpseks määratud puude osakaal 3209/(3209+1002) = 0,762 (76,2 %). Saadud arv kujutab enesest raieküpsuse testi tundlikkust.

Kõigist mitteraieküpsetest puudest prognoosib logistiline mudel õigesti mitteraieküpseteks 21912 ja valesti raieküpseteks 532. Seega on raieküpsuse testi spetsiifilisus 21912/(21912+532) = 0,976.

1.5.

Viimati konstrueeritud mudel hindas küll raieküpsuse ja vanuse vahelise seose kõigile puuliikidele eraldi, aga prognoosid ning testi tundlikkuse ja spetsiifilisuse saime ikkagi kõigile puuliikidele korraga. Kuigi ka eelneva analüüsi tulemustest on välja nopitavad tulemused puuliikide kaupa, on vahel lihtsam teha uus analüüs üksnes mingi alamandmestiku põhjal.

Hindame järgnevalt vaid kuuskede ja kaskede (eraldi) raieküpsust nende vanuse alusel ning leiame testide tundlikkuse ja spetsiifilisuse.

Esmalt moodustama puude andmestikust üksnes teadaoleva vanusega kuuski või kaski sisaldavad alamandmestikud ja teostame analüüsid nendega.

```
kuusk = subset(puud, subset = A>0 & PE=="KU")
kask = subset(puud, subset = A>0 & PE=="KS")
kuusk_logit = glm(mature~A, family=binomial(), data=kuusk)
kask_logit = glm(mature~A, family=binomial(), data=kask)
```

Tulemusi võib soovi korral illustreerida jälle ka pildiga, mis tuleb identne paar lehekülge tagasi konstrueerituga.

```
x = seq(0,200)
yKU = predict(kuusk_logit, data.frame(A=x), type="response")
yKS = predict(kask_logit, data.frame(A=x), type="response")
plot(x, yKU, type="l", xlab="Vanus (aastates)", ylab="Raieküpsuse tõenäosus")
lines(x, yKS, col="red")
legend(130, 0.8, c("Kuusk", "Kask"), col=c("black","red"), lty=1)
```

Leiame ka testide tundlikkuse ja spetsiifilisuse.

```
kyps_kuusk=predict(kuusk_logit, type="response")
tabel_kuusk=table(kyps_kuusk>0.5, kuusk$mature); tabel_kuusk
spets_kuusk = tabel_kuusk[1,1]/sum(tabel_kuusk[,1])
tund_kuusk = tabel_kuusk[2,2]/sum(tabel_kuusk[,2])
spets_kuusk; tund_kuusk
```

```
kyps_kask=predict(kask_logit, type="response")
tabel_kask=table(kyps_kask>0.5, kask$mature); tabel_kask
spets_kask = tabel_kask[1,1]/sum(tabel_kask[,1])
tund_kask = tabel_kask[2,2]/sum(tabel_kask[,2])
spets_kask; tund_kask
```

```
> spets_kuusk; tund_kuusk
[1] 0.9860474
[1] 0.8386207
```

```
> spets_kask; tund_kask
[1] 0.9786216
[1] 0.8607955
```

Tulemustest võib järeldada, et kui mitteraieküpsuse prognoosimise täpsus (testi spetsiifilisus) on kuuski ja kaski eraldi vaadates enam-vähem sama suur, kui kõiki puid koos vaadates – ligikaudu 98%, siis raieküpsuse prognoosimise osas (testi tundlikkus) annab mudel nii kuuskede kui ka kaskede puhul paremad tulemused kui kõigil puudel kokku – raieküpsetest kaskedest saab vaid nende vanuse järgi õigesti tuvastada 86%, raieküpsetest kuuskedest 84%.

1.6.

Aga, kas hinnatud tõenäosus 0,5 on see kõige õigem piir klassifitseerimaks puid raieküpseteks ja mitteraieküpseteks?

Optimaalseima lõikepunkti (*cutpoint*, prognoositud raieküpsuse tõenäosus, millest alates puu ka tegelikult raieküpseks lugeda) leidmiseks on kasutatavad II Maailmasõja ajal inglaste poolt radarisignaalidest omade ja võõraste lennukite eristamise algoritmide testimiseks välja töötatud **ROC-kõverad** (ROC = *Receiver Operating Characteristic*).

Kuigi *R*-s on olemas mitmeid vastavate analüüside jaoks mõeldud pakette, ei sisaldu neist ühtki vaikimisi koos *R*-ga (ja *R Commander*-iga) installeeritavate pakettide hulgas. Siiski on ROC-kõverad konstrueeritavad ka ilma vastavate pakettideta, lihtsalt hüpoteeside testimisest ja optimaalseima lõikepunkti headust mõõtvate karakteristikute hindamisest tuleb loobuda.

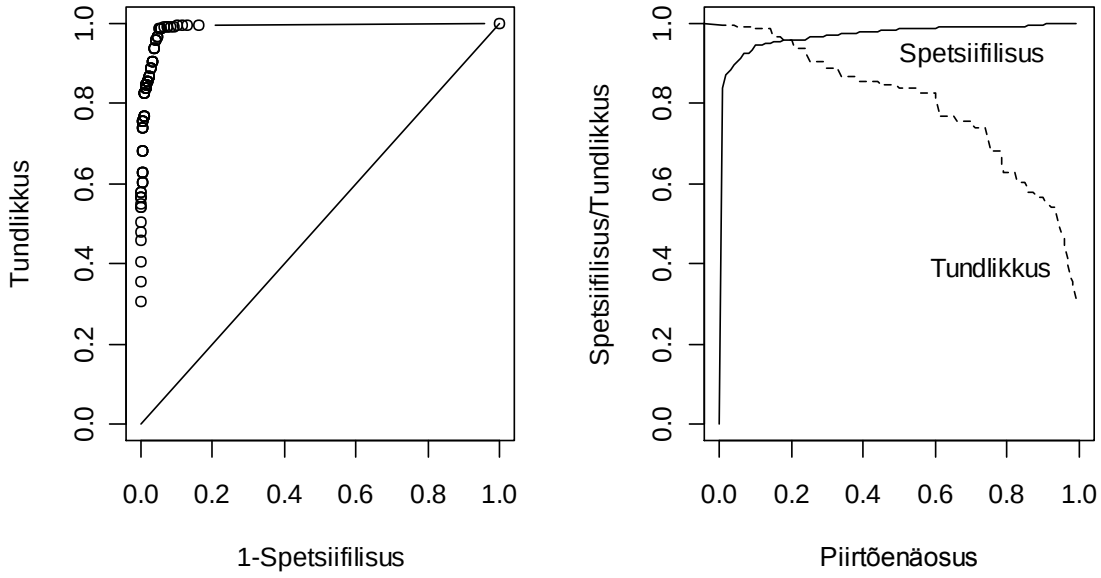
Järgnev programm teostab vajalikud arvutused ja joonistab välja ROC-kõverad kuuskede raieküpsuse tarvis.

```
fitkuusk=kuusk_logit$fitted.values
kuusk_punkt=seq(min(fitkuusk), 1, by=0.01)
nn=length(kuusk_punkt)
tun_kuusk=rep(NA,nn); spe_kuusk=rep(NA,nn)
for (i in 1:nn) {
   kuusk_progn=(fitkuusk>kuusk_punkt[i])
   kuusk_tabel=table(kuusk_progn, kuusk$mature)
   tun_kuusk[i]=kuusk_tabel[2,2]/sum(kuusk_tabel[,2])
   spe_kuusk[i]=kuusk_tabel[1,1]/sum(kuusk_tabel[,1])  }

par(mfrow=c(1,2))
plot(1-spe_kuusk, tun_kuusk, type="b", xlab="1-Spetsiifilisus",
 ylab="Tundlikkus", ylim=c(0,1), xlim=c(0,1))
lines(c(0,1), c(0,1))
plot(kuusk_punkt, spe_kuusk, type="l", xlab="Piirtõenäosus",
ylab="Spetsiifilisus/Tundlikkus", ylim=c(0,1), xlim=c(0,1))
lines(kuusk_punkt,tun_kuusk, lty=2)
text(0.7, 0.93, "Spetsiifilisus")
text(0.75, 0.4, "Tundlikkus")
par(mfrow=c(1,1))
```

Tulemuseks on kaks joonist (vt järgmine lk), millest vasakpoolne illustreerib tundlikkuse ja spetsiifilisuse vahekorda kõigi võimalike lõikepunktide korral (minimaalsest hinnatud raieküpsuse tõenäosusest kuni üheni sammuga 0,01). Diagonaalil paiknev joon märgib olukorda, mille korral test mõtet ei oma, reaalsete lõikepunktide alusel välja joonistatud kõvera kaugus diagonaalist näitab testi headust. Antud juhul paiknevad punktid peaaegu täisnurgana (peaaegu maksimaalselt eemal diagonaalist), mis tähendab testi väga head sobivust – kuuskede vanus võimaldab suurepäraselt prognoosida kuuskede raieküpsust.

Parempoolsel joonisel on vertikaalteljel tundlikkus ja spetsiifilisus ning horisontaalteljel võimalikud raieküpsuse tõenäosused. Punkt, kus tundlikkuse ja spetsiifilisuse graafikud lõikuvad, märgib optimaalseimat testi tulemust (logistilise regressiooniga hinnatud tõenäosust, millest alates lugeda puu raieküpseks). Ligikaudu on vastav tõenäosus 0,2.



Kontrollime, kas piirtõenäosus 0,2 annab paremad tundlikkuse ja spetsiifilisuse väärtused, kui eelnevalt kasutatud väärtus 0,5.

```
kyps_kuusk=predict(kuusk_logit, type="response")
tabel_kuusk=table(kyps_kuusk>0.2, kuusk$mature); tabel_kuusk
spets_kuusk = tabel_kuusk[1,1]/sum(tabel_kuusk[,1])
tund_kuusk = tabel_kuusk[2,2]/sum(tabel_kuusk[,2])
spets_kuusk; tund_kuusk
```

```
> tabel_kuusk=table(kyps_kuusk>0.2, kuusk$mature); tabel_kuusk

          0    1
FALSE 5222   31
TRUE   225  694
```

```
> spets_kuusk; tund_kuusk
[1] 0.9586929
[1] 0.9572414
```

Tõepoolest, spetsiifilisuse väärtus küll 2,8% võrra langes, seevastu tundlikkus suurenes peaaegu 12% (0,839-lt 0,957-ni). Seega oleks mõistlik lugeda kuusk raieküpseks juba siis, kui logistilise regressiooni võrrandist vanuse alusel leitud prognoos raieküpsuse tõenäosusele on >0,2.

PS. Kui te tahate teada, milline see kuuskede raieküpsuse tõenäosusele hinnatud logistilise regressiooni võrrand ikka on, siis tuleb rakendada käsku `summary` kuuskedele sobitatud mudelile:

```
summary(kuusk_logit)
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -19.64508    0.87718  -22.40   <2e-16 ***
A             0.22066    0.01005   21.96   <2e-16 ***
```

Seega

P(kuusk vanuses *A* on raieküps) = exp(-19,645 + 0,221×*A*) / (1 + exp(-19,645 + 0,221×*A*))

ja kuuse vanuse suurenemisel 1 aasta võrra suureneb šanss osutuda raieküpseks $e^{0,221} = 1,25$ korda.

1.7.

Milline on aga vanus, millal kuusk on 90%-lise tõenäosusega raieküps?

Arvutamiseks võime viimati kirja pandud logistilise regressiooni võrrandist avaldada vanuse *A* raieküpsuse tõenäosuse *p* kaudu:

$$A = [\ln(p/(1-p)) - (-19,645)] / 0,221, \quad p = P(\text{kuusk on raieküps}) = 0,9.$$

R-s:

```
> prob=0.9
> (log(prob/(1-prob))-(-19.64508))/0.22066
[1] 98.98624
```

Tegelikult oskab *R* funktsiooni `glm` rakendamise järel vastavat tõenäosust ka ise arvutada. Selleks on olemas käsk `dose.p`.

Käsu esimses pooles olev sõna „dose“ viitab ülesandele, milleks antud funktsioon välja töötati – so leida etteantud tõenäosusega surmavalt mõjuv doos (näiteks LD90 – *90% lethal dose*).

Käsu süntaks püstitatud küsimusele vastamiseks on kujul

```
dose.p(kuusk_logit, cf = c(1,2), p = 0.9)
```

Esimene argument on mudel, mille alusel soovitud väärtust hinnata. Teine argument `cf` ütleb *R*-le, mitmendal kohal on mudeli parameetrite hulgas vabaliige ja mitmendal kohal vanuse mõjule vastav kordaja (kui mudelis on rohkem argumente, ei vastavad suurused enam paikneda kohtadel 1 ja 2). Kolmas argument `p` näitab, millisele tõenäosusele vastavat argumendi väärtust soovitakse hinnata.

Soovi korral võib korraga hinnata ka mitmele erinevale raieküpsuse tõenäosusele vastavad vanused:

```
dose.p(kuusk_logit, cf = c(1,2), p = c(0.5, 0.6, 0.7, 0.8, 0.9))
```

```
> dose.p(kuusk_logit, cf = c(1,2), p = c(0.5, 0.6, 0.7, 0.8, 0.9))
             Dose        SE
p = 0.5: 89.02755 0.3862613
p = 0.6: 90.86504 0.4150930
p = 0.7: 92.86733 0.4619738
p = 0.8: 95.30996 0.5348745
p = 0.9: 98.98493 0.6648602
```

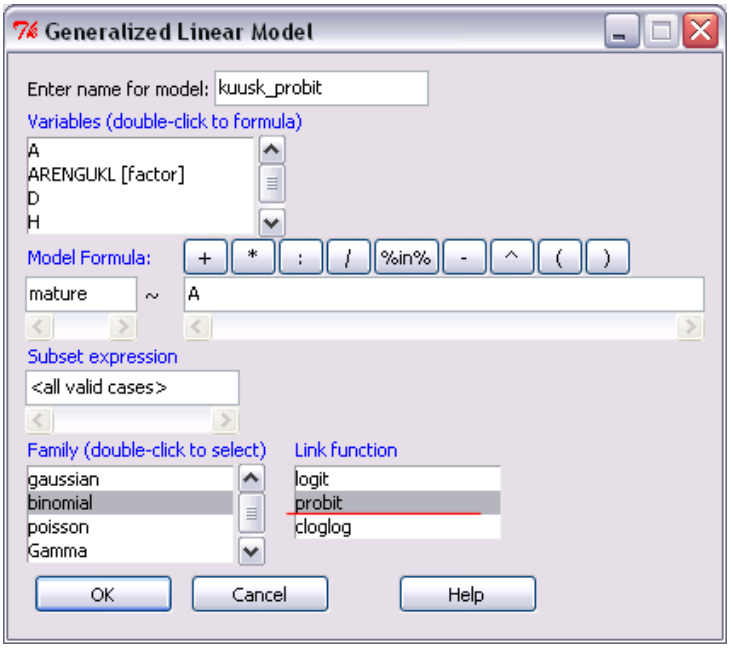
Lisaks etteantud raieküpsuse tõenäosustele vastavatele hinnangulistele vanustele väljastab *R* ka hinnangute täpsust näitavad standardvead.

1.8.

Lõpetuseks võiks uurida, kuivõrd erinevad logistilise mudeli alusel prognoositud kuuskede raieküpsuse tõenäosused probit-mudeli abil leituist?

Probit-mudeli rakendamiseks võib uuesti sisestada (või valida *R Commander*'i menüüst) vastava võrrandi ja määrata seosefunktsiooniks (*link function*) *probit*:

```
kuusk_probit <- glm(mature ~ A, family=binomial(probit), data=kuusk)
```



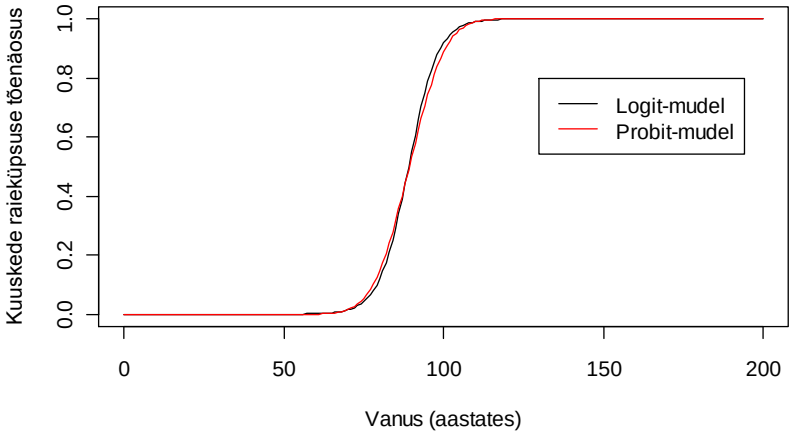
Aga *R*-s on olemas võimalus ka juba olemasoleva mudeli muutmiseks. Funktsiooniks on `update`, mille esimene argument on muudetav mudel, mille järel tuleb kirja panna kõik muudetavad argumendid. Uue mudeli võib soovi korral salvestada uue nimega.

Seega võib mudeli nimega 'kuusk_probit' tekitada ka käsuga

```
kuusk_probit <- update(kuusk_logit, family=binomial(probit))
```

Mudelite võrdlemiseks võib teha joonise:

```
x = seq(0,200)
pred_logit = predict(kuusk_logit, data.frame(A=x), type="response")
pred_probit = predict(kuusk_probit, data.frame(A=x), type="response")
plot(x, pred_logit, type="l", xlab="Vanus (aastates)",
ylab="Kuuskede raieküpsuse tõenäosus")
lines(x, pred_probit, col="red")
legend(130, 0.8, c("Logit-mudel", "Probit-mudel"),
col=c("black","red"), lty=1)
```



Ega vahet tegelikult peaaegu polegi …

Aga kas hinnangulised vanused 90%-lise (vm %-lise) raieküpsuse saavutamiseks on erinevad?

```
Matprop_logit = dose.p(kuusk_logit,cf=c(1,2),p=c(0.5,0.6,0.7,0.8,0.9))
Matprop_probit = dose.p(kuusk_probit,cf=c(1,2),p=c(0.5,0.6,0.7,0.8,0.9))
cbind(Matprop_logit,Matprop_probit)
```

Ega eriti ei ole:

```
> cbind(Matprop_logit,Matprop_probit)
         Matprop_logit Matprop_probit
p = 0.5:      89.02755       89.23885
p = 0.6:      90.86504       91.47535
p = 0.7:      92.86733       93.86816
p = 0.8:      95.30996       96.66852
p = 0.9:      98.98493      100.55213
```

See, miks mõnel erialal eelistatakse binaarsete tunnuste modelleerimisel kasutada logistilisi mudeleid ja mõnel erialal probit-mudeleid, on enamasti kinni traditsioonides.