Opgave 1: Graft vs. Host disease

Denne opgave er baseret på opgave 12.3 fra DG Altman, p. 361. Data omhandler knoglemarvstransplantation af 37 leukæmipatienter, og outcome er forekomst af graft versus host disease, GvHD. Formålet med opgaven er at forsøge at prediktere, hvem der får GvHD.

Filen `gvhd.txt` indeholder følgende variable:

pnr Patientnummer

rcpage Patientens alder (recipienten)

donage Donors alder

type Leukæmitype:
1:AML (akut myeloid leukæmi)
2:ALL (akut lymfatisk leukæmi)
3:CML (kronisk myeloid leukæmi)

preg Indikator for hvorvidt donor har været gravid (1:ja, 0:nej)

index Index for epidermal celle-lymfocyt reaktion (kvantitativ måling)

gvhd Forekomst af GvHD (1:ja, 0:nej)

Data indlæses direkte fra hjemmesiden:

```
graft <-
read.table("http://publicifsv.sund.ku.dk/~lts/basal/data/gvhd.txt",
header=T)

graft$pnr = NULL

graft$grp = graft$type
graft$type = NULL

graft$type <- factor(graft$grp, levels=1:3, labels=c("AML", "ALL", "CML"))
```

Bemærk, at variablen angivet ovenfor som `type` her overføres til `grp` og bagefter laves om til en mere informativ version (med karakter værdier), som så kaldes `type`. Dette kan sikkert gøres mere elegant....

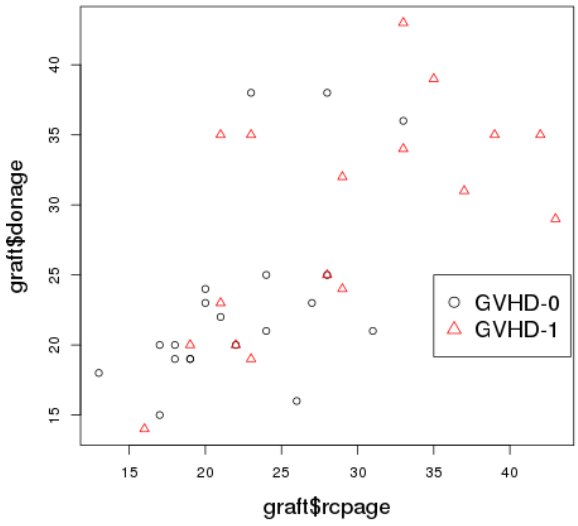
Herefter indeholder data frame'n variabel-kolonnerne:
1: `rcpage`, 2: `donage`, 3: `preg`, 4: `index`, 5: `gvhd`, 6: `grp`, 7: `type`

og vi kan danne os et overblik over data på forskellig vis, som illusteret nedenfor:

```
> summary(graft[,c(1,2,4)])
     rcpage          donage          index
 Min.   :13.00   Min.   :14.00   Min.   : 0.270
 1st Qu.:20.00   1st Qu.:20.00   1st Qu.: 0.920
 Median :23.00   Median :23.00   Median : 2.010
 Mean   :25.43   Mean   :25.81   Mean   : 2.556
 3rd Qu.:29.00   3rd Qu.:34.00   3rd Qu.: 3.730
 Max.   :43.00   Max.   :43.00   Max.   :10.110


> table(graft$gvhd)

 0  1
20 17


> table(graft$type)

AML ALL CML
 11  16  10
```

Vi bemærker, at patienterne er mellem 13 og 43 år, og at donorernes alder synes at stemme godt overens med patienternes alder. Dette er ikke noget tilfælde, idet man forsøger at matche disse, hvilket f.eks. kan ses af et simpelt scatterplot, hvor der også er tilføjet farver alt efter om patienten *har* fået graft vs host disease (rød) eller *ikke har* (sort).

```
symbol=graft$gvhd+1

plot(graft$rcpage, graft$donage, pch=symbol, col=symbol, cex.lab=1.5)
legend(35, 25, legend=c("GVHD-0", "GVHD-1"), pch=1:2,
       col=c("black", "red"), cex=1.5)
```



På denne figur ses i øvrigt, at højere alder synes at disponere for at få graft vs. host disease, men at det måske kan være svært at skille aldrene på de to personer ad (donor og modtager).

Yderligere confounding må påregnes mht. variablen `preg`, der angiver om donoren har været gravid eller ej, da denne naturligt nok også må hænge sammen med donors alder.

Og så begynder vi på spørgsmålene:

1. *Lav først en tabel, der viser hyppigheden af GvHD for de tre sygdomstyper. Er der evidens for en forskel på disse tre hyppigheder?*

   Der er her tale om en simpel 3×2-tabel, så vi benytter koden:

```
> table(graft[,c(7,5)])
      gvhd
type    0  1
  AML   6  5
  ALL  12  4
  CML   2  8
```

Dette er dog en lige lovlig skrabet tabel, så vi sætter lidt mere information på nedenfor: Vi udregner rækkeprocenter og udfører et χ^2-test, samt (da der er tale om en meget *tynd* tabel, også et Fishers eksakt test.

Dette leder til outputtet

```
> spm1 = matrix(table(graft$type, graft$gvhd),nrow=3)
> rownames(spm1) = c("AML","ALL","CML")
> colnames(spm1) = c("GVHD_0","GVHD_1")

> spm1
    GVHD_0 GVHD_1
AML      6      5
ALL     12      4
CML      2      8

> chisq.test(spm1)
Pearson's Chi-squared test

data:  spm1
X-squared = 7.497, df = 2, p-value = 0.02355

Warning message:
In chisq.test(spm1) : Chi-squared approximation may be incorrect


> chisq.test(spm1)$expected
      GVHD_0   GVHD_1
AML 5.945946 5.054054
ALL 8.648649 7.351351
CML 5.405405 4.594595
Warning message:
In chisq.test(spm1) : Chi-squared approximation may be incorrect

> fisher.test(spm1)

Fisher's Exact Test for Count Data

data:  spm1
p-value = 0.02879
alternative hypothesis: two.sided
```

Vi ser af tabellen, at der totalt set observeres 17 tilfælde af graft vs. host disease blandt de 37 personer, samt at der er en tydelig overvægt af tilfælde blandt type de kroniske leukæmier (CML). Vi ser af χ^2-testet, at der faktisk er signifikant forskel på de tre hyppigheder ($P = 0.024$), men det er *meget små antal*, og de forventede værdier (som også fremgår af tabellen) er flere steder lige omkring eller under 5.

Det tilsvarende eksakte test giver dog P=0.029, altså ligeledes signifikans.

2. *Lav nu en logistisk regression med* `gvhd` *som outcome og med de øvrige variable som forklarende variable, idet I dog først logaritmetransformerer* `index`*.*

   Vi ser i første omgang på den logaritmetransformerede variabel `graft$logindex = log(graft$index)`, hvor vi altså har brugt den naturlige logaritme. Husk, at det ikke gør nogen forskel for modellen, kun for de estimater, man vælger at fokusere på.

   Vi bemærker først, at der i alt er 5 potentielle kovariater, hvoraf den ene (`type`) er kategorisk med 3 niveauer, dvs. der skal estimeres 6 parametre i en model, hvor alle medtages. Dette er **klart for meget**, så vores analyse må betegnes som en **fisketur**, hvor evt. resultater skal eftervises ved en senere lejlighed!

   Vi benytter `glm`, og vi afstår fra modelkontrol, da materialet er for lille til, at det vil give mening:

```
fuld = glm(formula=gvhd ~
type+logindex+as.factor(preg)+donage+rcpage,
family=binomial(link="logit"), data=graft)

summary(fuld)
drop1(fuld, test = "Chisq")
```

   Outputtet (beskåret) fra denne *fulde* model bliver:

```
> summary(fuld)

Call:
glm(formula = gvhd ~ type + logindex + as.factor(preg) + donage +
    rcpage, family = binomial(link = "logit"), data = graft)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
```

```
-2.00075  -0.36457  -0.09559   0.57964   1.68281

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      -5.47420    2.59647  -2.108   0.0350 *
typeALL          -0.42824    1.25389  -0.342   0.7327
typeCML           1.61517    1.30743   1.235   0.2167
logindex          1.81874    0.88919   2.045   0.0408 *
as.factor(preg)1  1.66236    1.18103   1.408   0.1593
donage            0.10724    0.08262   1.298   0.1943
rcpage            0.01597    0.08205   0.195   0.8457
---
```

```
> drop1(fuld, test = "Chisq")
Single term deletions

Model:
gvhd ~ type + logindex + as.factor(preg) + donage + rcpage
                Df Deviance    AIC    LRT Pr(>Chi)
<none>               26.252 40.252
type             2   28.959 38.959 2.7064  0.25841
logindex         1   32.343 44.343 6.0904  0.01359 *
as.factor(preg)  1   28.447 40.447 2.1947  0.13849
donage           1   28.082 40.082 1.8292  0.17622
rcpage           1   26.291 38.291 0.0382  0.84495
```

Vi bemærker, at vi i denne model ikke kan se forskel på de 3 sygdomsgrupper (overall test giver P=0.26).

Vi venter med at kommentere på estimaterne for odds ratio til et senere spørgsmål.

3. *Fjern nu de forklarende variable successivt fra modellen, idet den mindst signifikante fjernes først (backwards elimination). Overvej gerne, om dette er en fornuftig fremgangsmåde.*

   (a) *Undervejs i denne proces skal I være opmærksomme på, om estimater og P-værdier for nogle kovariater ændrer sig voldsomt ved fjernelse af andre. Se specielt på forskellen blandt leukæmigrupperne.*

   Man ser, at `rcpage` (patientens alder) har den største P-værdi (P=0.84), så vi starter med at smide denne ud af modellen:. Bemærk, at dette umiddelbart er i modstrid med vores iagttagelse fra figuren s. 3, men som nævnt er der flere variable, der hænger meget sammen og således let kan "dække" over hinanden.

```
> ud1 = glm(formula=gvhd ~
+ type+logindex+as.factor(preg)+donage,
+ family=binomial(link="logit"), data=graft)

> summary(ud1)

Call:
glm(formula = gvhd ~ type + logindex + as.factor(preg) + donage,
    family = binomial(link = "logit"), data = graft)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.05229  -0.36206  -0.09068   0.62322   1.66401

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      -5.29619    2.40716  -2.200   0.0278 *
typeALL          -0.39678    1.24246  -0.319   0.7495
typeCML           1.69128    1.26205   1.340   0.1802
logindex          1.84073    0.88748   2.074   0.0381 *
as.factor(preg)1  1.72187    1.14533   1.503   0.1327
donage            0.11352    0.07606   1.493   0.1356
---

> drop1(ud1, test = "Chisq")
Single term deletions

Model:
gvhd ~ type + logindex + as.factor(preg) + donage
               Df Deviance    AIC    LRT Pr(>Chi)
<none>              26.291 38.291
type            2   29.403 37.403 3.1126  0.21092
logindex        1   32.630 42.630 6.3397  0.01181 *
as.factor(preg) 1   28.812 38.812 2.5218  0.11228
donage          1   28.832 38.832 2.5411  0.11092
---
```

Der skete ingen særlige ændringer af de øvrige estimater, da vi smed `rcpage` ud.

Den mindst signifikante variabel blandt de tilbageværende er nu `type` (P=0.21), men denne har en særstatus, fordi den indeholder to sammenligninger (3 grupper, 2 frihedsgrader). Kigger vi nærmere på forskellene på de 3 grupper, ser vi, at den højeste P-værdi (**af de to viste!**) frekommer ud for `type ALL`, svarende til sammenligningen mellem patienter med ALL og patienter med AML (da denne sidstnævnte svarer til estimatet 0). Den ikke-viste sammenligning er mellem ALL og CML, men ud fra estimaterne kan vi se, at denne forskel må være den største af de 3.

Det er derfor klart, at den mindst signifikante forskel er mellem AML og ALL patienter, dvs. at den eneste betydende prognostiske forskel på leukæmigrupperne mht. GvHD ser ud til at være, hvorvidt der er tale om en akut eller kronisk leukæmi.

(b) *I stedet for at se på leukæmitype som en kategorisk variabel med 3 niveauer kunne man have set på de to forskellige aspekter: 'kronisk versus akut' og 'lymfatisk versus myeloid'. Definer to nye 0/1-variable, der kan bruges til at vurdere hvert af de to aspekter.*

Erstat den kategoriske variabel `type` *med de to nye variable, og foretag en ny backwards elimination.*

Vi definerer de to nye variable `cml` og `all` i stedet for `type`, og starter forfra, men her vises kun yderst skrabet output fra de to første kørsler, idet det er de selvsamme modeller som dem, vi så på ovenfor:

```
graft$all = as.numeric(graft$grp == 2)
graft$cml = as.numeric(graft$grp == 3)

fuld1 = glm(formula=gvhd ~
all+cml+logindex+as.factor(preg)+donage+rcpage,
family=binomial(link="logit"), data=graft)
summary(fuld1)

ud1 = glm(formula=gvhd ~
all+cml+logindex+as.factor(preg)+donage,
family=binomial(link="logit"), data=graft)
summary(ud1)
```

og vi finder

```
> summary(fuld1)

Call:
glm(formula = gvhd ~ all + cml + logindex + as.factor(preg) +
    donage + rcpage, family = binomial(link = "logit"), data = graft)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.00075  -0.36457  -0.09559   0.57964   1.68281
```

```
Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -5.47420    2.59647  -2.108   0.0350 *
all              -0.42824    1.25389  -0.342   0.7327
cml               1.61517    1.30743   1.235   0.2167
logindex          1.81874    0.88919   2.045   0.0408 *
as.factor(preg)1  1.66236    1.18103   1.408   0.1593
donage            0.10724    0.08262   1.298   0.1943
rcpage            0.01597    0.08205   0.195   0.8457
---

> summary(ud1)

Call:
glm(formula = gvhd ~ all + cml + logindex + as.factor(preg) +
    donage, family = binomial(link = "logit"), data = graft)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.05229  -0.36206  -0.09068   0.62322   1.66401

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -5.29619    2.40716  -2.200   0.0278 *
all              -0.39678    1.24246  -0.319   0.7495
cml               1.69128    1.26205   1.340   0.1802
logindex          1.84073    0.88748   2.074   0.0381 *
as.factor(preg)1  1.72187    1.14533   1.503   0.1327
donage            0.11352    0.07606   1.493   0.1356
---
```

Herfra er det naturligt at gå videre med at fjerne variablen `all`, hvilket svarer til at slå de to akutte former for leukæmi sammen, og kun skelne mellem akut og kronisk leukæmi.

```
ud2 = glm(formula=gvhd ~
cml+logindex+as.factor(preg)+donage,
family=binomial(link="logit"), data=graft)
summary(ud2)
```

```
> summary(ud2)

Call:
glm(formula = gvhd ~ cml + logindex + as.factor(preg) + donage,
    family = binomial(link = "logit"), data = graft)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0726  -0.4070  -0.0957   0.5725   1.7072

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -5.41209    2.33766  -2.315   0.0206 *
cml               1.87021    1.14214   1.637   0.1015
```

```
logindex          1.89722    0.86718   2.188   0.0287 *
as.factor(preg)1  1.73459    1.14852   1.510   0.1310
donage            0.10922    0.07354   1.485   0.1375
---
```

Vi bemærker, at udeladelsen af `all` fra modellen bevirker, at `cml` får en noget øget betydning, og en lavere P-værdi. Men husk, at sammenligningen nu er *en anden en før*! Før var det sammenligningen af CML og AML, men nu er det sammenligningen af CML med *både* AML og ALL. At slå disse to sammen giver samtidig større styrke i sammenligningen.

I denne model er der tre variable med P-værdier omkring 10-15%, `cml`, `preg` og `donage`, så det er ikke umiddelbart klart hvilken af dem, man skal fjerne først. Man skal være meget stærk i troen for at insistere på, at det skal være `donage`, der fjernes, så i stedet har vi her valgt at fitte alle tre mulige modeller, hvor en af dem er fjernet:

```
mod1 = glm(formula=gvhd ~
logindex+as.factor(preg)+donage,
family=binomial(link="logit"), data=graft)

mod2 = glm(formula=gvhd ~
cml+logindex+donage,
family=binomial(link="logit"), data=graft)

mod3 = glm(formula=gvhd ~
cml+logindex+as.factor(preg),
family=binomial(link="logit"), data=graft)
```

Herved får vi outputtet:

```
> summary(mod1)

Call:
glm(formula = gvhd ~ logindex + as.factor(preg) + donage, family = binomial(link = "logit"),
    data = graft)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0395  -0.5230  -0.1074   0.6282   1.8942
```

```
Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      -5.25867    2.13355  -2.465   0.0137 *
logindex          2.10604    0.82273   2.560   0.0105 *
as.factor(preg)1  1.29125    1.03016   1.253   0.2100
donage            0.12453    0.06579   1.893   0.0584 .
---

> summary(mod2)

Call:
glm(formula = gvhd ~ cml + logindex + donage, family = binomial(link = "logit"),
    data = graft)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2413  -0.5873  -0.1087   0.5314   1.5529

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.85304    2.26239  -2.587  0.00968 **
cml          1.51424    1.07876   1.404  0.16041
logindex     2.05917    0.82810   2.487  0.01290 *
donage       0.14877    0.06938   2.144  0.03202 *
---


> summary(mod3)

Call:
glm(formula = gvhd ~ cml + logindex + as.factor(preg), family = binomial(link = "logit"),
    data = graft)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0680  -0.4158  -0.1651   0.7017   1.4405

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       -2.5464     0.9485  -2.685  0.00726 **
cml                2.2506     1.1060   2.035  0.04187 *
logindex           1.4877     0.7197   2.067  0.03872 *
as.factor(preg)1   2.4955     1.1012   2.266  0.02344 *
---
```

Det fremgår af ovenstående, at når en af variablene fjernes, så bliver mindst en af de andre signifikante. I den sidste model er såvel cml som preg signifikante, så det vil næppe være en god ide at udelade nogen af disse. Hvis vi får kniven for struben og **skal** vælge en af modellerne, vil vi derfor vælge denne.

(c) *Overvej forskellen på resultaterne fra de to analyser i 3a og 3b.*

Nu stoppede vi jo sådan set den baglæns elimination i 3a inden vi var nået til vejs ende. Hvis vi ikke havde erstattet `type` med `cml`

og `all`, ville vi have fortsat eliminationen ved at fjerne `type` helt fra modellen, og så ville vi have fortsat således:

```
try1 = glm(formula=gvhd ~
logindex+as.factor(preg)+donage,
family=binomial(link="logit"), data=graft)
summary(try1)

try2 = glm(formula=gvhd ~
logindex+donage,
family=binomial(link="logit"), data=graft)
summary(try2)
```

hvorved vi ville have fået

```
> summary(try1)

Call:
glm(formula = gvhd ~ logindex + as.factor(preg) + donage, family = binomial(link = "logit"),
    data = graft)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0395  -0.5230  -0.1074   0.6282   1.8942

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -5.25867    2.13355  -2.465   0.0137 *
logindex          2.10604    0.82273   2.560   0.0105 *
as.factor(preg)1  1.29125    1.03016   1.253   0.2100
donage            0.12453    0.06579   1.893   0.0584 .
---

> summary(try2)

Call:
glm(formula = gvhd ~ logindex + donage, family = binomial(link = "logit"),
    data = graft)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8298  -0.6412  -0.1189   0.6440   1.7503

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.45399    2.08147  -2.620  0.00879 **
logindex     2.17773    0.78986   2.757  0.00583 **
donage       0.14594    0.06465   2.257  0.02399 *
```

Som det ses, ville vi altså ende med en helt anden konklusion end ovenfor. Index-variablen ville godt nok stadig være signifikant,

men med en væsentlig større estimeret effekt, men `preg` er ikke med i modellen mere, og vi fik ikke opdaget, at kronisk og akut leukæmi giver forskellig risiko. Til gengæld vil vi nu vurdere ældre donorer til at være mere risikable, hvilket måske kan skyldes, at ældre donorer oftere har været gravide.....

Faktisk er det ikke let at svare på, hvad der *egentlig* har betydning for forekomsten af GvHD, fordi vi kun har sølle 17 tilfælde alt i alt, og derfor absolut ikke burde bruge mere end 2 parametre på kovariater!!

Vi må (endnu en gang) konstatere, at vi har været på fisketur!!

4. *Find odds-ratioerne associeret med de binære forklarende variable i slutmodellen fra 3b, med tilhørende 95% konfidensintervaller.*

   For at få de estimerede odds-ratioer i slutmodellen (`mod3`) fra spørgsmål 3b, skriver vi:

```
> exp(cbind(coef(mod3), confint(mod3)))
Waiting for profiling to be done...
                                    2.5 %      97.5 %
(Intercept)       0.07836183 0.007712384   0.3584982
cml               9.49307468 1.256305406 109.1452495
logindex          4.42699116 1.288578179  24.2894317
as.factor(preg)1 12.12781735 1.666952110 141.9878238
```

   og vi får altså konklusionerne

   - Folk med kronisk leukæmi har en odds for GvHD, der er 9.49 gange højere end folk med akut leukæmi, CI=(1.26, 109.1).
   - Patienter, der modtager knoglemarv fra en donor, der har været gravid, har en odds for GvHD, der er 12.13 gange højere end de, der modtager den fra en donor, der ikke har været gravid, CI=(1.67, 141.99).

   Man ser, at selv om de estimerede odds-ratioer er ret store og signifikant forskellige fra 1, er den nedre grænse for begge ret tæt på 1, så der er ikke megen evidens for substantielle effekter af disse to variable.

5. *Giv en verbal fortolkning af koefficienten til log(index), gerne en, du kunne anvende, hvis du talte til en kongres.*
Hvis du på baggrund af dette hellere ville have anvendt en anden logaritmefunktion, så skift den ud nu.

   Den odds ratio, der er angivet for `logindex` (altså 4.427) svarer til OR mellem to patienter der adskiller sig med 1 i værdien af logindex, dvs. mellem to patienter, hvis forhold mellem indexværdier er $e = 2.7183$. Det er næppe en intuitivt rimelig måde at rapportere effekten af index på.

   Disse fortolkningsproblemer gør, at vi gerne vil skifte logaritme, f.eks. til 2-tals logaritmen, `log2index=log2(index)`.

   En forskel på 1 enhed i `log2index` svarer til en fordobling af index-værdien, hvilket er en noget mere mundret fortolkning.

   Hvis vi fitter en model med `log2index`, får vi:

```
> slut = glm(formula=gvhd ~
+ cml+log2index+as.factor(preg),
+ family=binomial(link="logit"), data=graft)

> exp(cbind(coef(slut), confint(slut)))
Waiting for profiling to be done...
                                    2.5 %      97.5 %
(Intercept)       0.07836183 0.007712384   0.3584982
cml               9.49307468 1.256305406 109.1452495
log2index         2.80445446 1.192128228   9.1263713
as.factor(preg)1 12.12781735 1.666952110 141.9878238
```

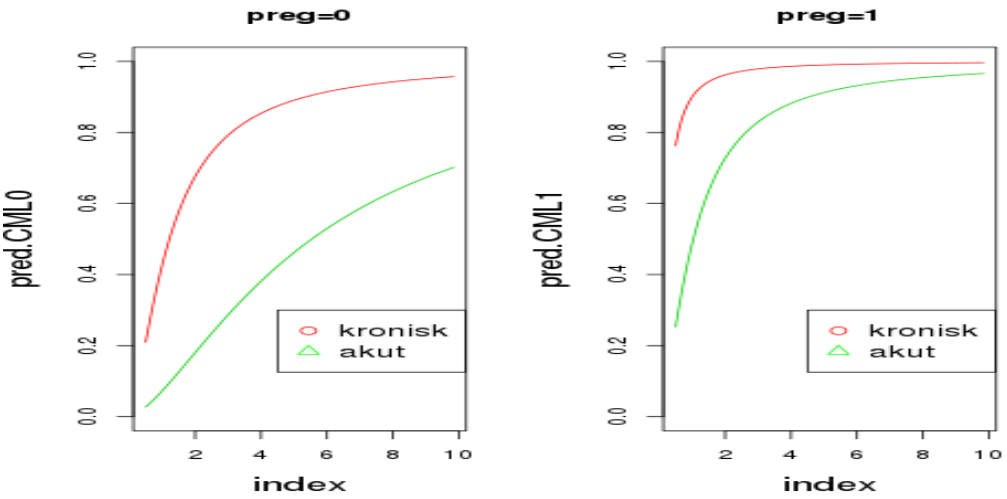
   Estimatet for odds-ratio associeret med log2(index), dvs. med en fordobling af index-værdien, bliver 2.80, med 95% konfidensintervallet (1.19 , 9.13).

   Bemærk, at fittet af de to modeller er fuldstændigt det samme, det er kun koefficienten til `log2index`, der er anderledes, testet og P-værdien er den samme. I princippet kunne man i stedet for at vælge 2-talslogaritmen have valgt 10-talslogaritmen. Den estimerede odds-ratio ville da svare til effekten af en 10-dobling af index-værdien. Det

ville svare nogenlunde til range af index, så en reference til fordobling af værdien vil nok i dette materiale være mest klinisk relevant, mens man i andre materialer vil støde på variable som i populationen varierer meget mere, og hvor effekten af en 10-dobling ville kunne være relevant. Hvilken logaritme, man med fordel kan anvende i forbindelse med en kovariat, afhænger altså helt af den konkrete sammenhæng, og er *ikke* et statistisk spørgsmål.

En illustration af de predikterede sandsynligheder for graft-vs-host disease kan fås ved at prediktere sandsynligheder for en ny data frame, med et passende antal værdier af `index` eller `log2index`. Vi skal dog gøre dette for alle 4 kombinationer af de to andre kovariater, `cml` og `preg`:

```
ny.CML0 = data.frame(log2index=seq(-1,3.3,0.1),cml=1,preg=0)
pred.CML0 = predict(slut, ny.CML0, type="response")

ny.CML1 = data.frame(log2index=seq(-1,3.3,0.1),cml=1,preg=1)
pred.CML1 = predict(slut, ny.CML1, type="response")

ny.AKUT0 = data.frame(log2index=seq(-1,3.3,0.1),cml=0,preg=0)
pred.AKUT0 = predict(slut, ny.AKUT0, type="response")

ny.AKUT1 = data.frame(log2index=seq(-1,3.3,0.1),cml=0,preg=1)
pred.AKUT1 = predict(slut, ny.AKUT1, type="response")

index=2^(seq(-1,3.3,0.1))

par(mfrow=c(1,2))
plot(index, pred.CML0, main="preg=0", ylim=c(0,1),
     col=2, type="l", cex.lab=1.5)
lines(index, pred.AKUT0, col=3, type="l", cex.lab=1.5)
plot(index, pred.CML1, main="preg=1", ylim=c(0,1),
     col=2, type="l", cex.lab=1.5)
lines(index, pred.AKUT1, col=3, type="l", cex.lab=1.5)
```



6. *Stod der mon noget i protokollen om at undersøge nogle interessante interaktioner? Prøv f.eks. interaktionen mellem det logaritmerede index og* `preg`*, og giv en beskrivelse af resultatet.*

   Her forestiller vi os, at index-værdien kunne have forskellig prognostisk betydning, afhængig af, om donor har været gravid eller ej.

   Vi tester denne interaktion *som sædvanlig*, ved at inkludere leddet `preg*log2index`:

```
int = glm(formula=gvhd ~
cml+log2index+as.factor(preg)+as.factor(preg)*log2index,
family=binomial(link="logit"), data=graft)
summary(int)
```

   og finder outputtet:

```
> int = glm(formula=gvhd ~
+ cml+log2index+as.factor(preg)+as.factor(preg)*log2index,
+ family=binomial(link="logit"), data=graft)
> summary(int)

Call:
glm(formula = gvhd ~ cml + log2index + as.factor(preg) + as.factor(preg) *
    log2index, family = binomial(link = "logit"), data = graft)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.41155  -0.27268  -0.05696   0.44911   1.28816
```

```
Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -3.5104     1.4643  -2.397   0.0165 *
cml                          2.5028     1.2661   1.977   0.0481 *
log2index                    1.7238     0.8507   2.026   0.0427 *
as.factor(preg)1             6.3005     2.8768   2.190   0.0285 *
log2index:as.factor(preg)1  -2.8780     1.7085  -1.684   0.0921 .
---
```

Med en P-værdi på 9% er der en svag indikation af forskel på effekten af index-variablen afhængig af om donor har været gravid eller ej.

For at estimere de to effekter (for `preg` hhv 0 og 1), kører vi modellen uden hovedvirkning af `log2index`, og med de to separate effekter af denne ved at benytte `as.factor(preg):log2index` i stedet for `as.factor(preg)*log2index`, altså med kolon i stedet for stjerne:

```
int2 = glm(formula=gvhd ~
cml+as.factor(preg)+as.factor(preg):log2index,
family=binomial(link="logit"), data=graft)
summary(int2)
exp(cbind(coef(int2), confint(int2)))
```

hvorved vi i stedet får

```
> int2 = glm(formula=gvhd ~
+ cml+as.factor(preg)+as.factor(preg):log2index,
+ family=binomial(link="logit"), data=graft)
> summary(int2)

Call:
glm(formula = gvhd ~ cml + as.factor(preg) + as.factor(preg):log2index,
    family = binomial(link = "logit"), data = graft)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.41155  -0.27268  -0.05696   0.44911   1.28816

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -3.5104     1.4643  -2.397   0.0165 *
cml                          2.5028     1.2661   1.977   0.0481 *
as.factor(preg)1             6.3005     2.8768   2.190   0.0285 *
as.factor(preg)0:log2index   1.7238     0.8507   2.026   0.0427 *
as.factor(preg)1:log2index  -1.1542     1.4700  -0.785   0.4324
---

> exp(cbind(coef(int2), confint(int2)))
Waiting for profiling to be done...
                                          2.5 %       97.5 %
(Intercept)                0.02988464 0.0005364857 2.505458e-01
cml                       12.21605402 1.3345076061 2.594046e+02
```

```
as.factor(preg)1           544.86828442 6.8056809676 1.720470e+06
as.factor(preg)0:log2index   5.60598270 1.6126269770 5.976259e+01
as.factor(preg)1:log2index   0.31531752 0.0055253095 3.527806e+00
```

Vi ser, at effekten af index-variablen ser ser ud til at være modsat for patienter med og uden gravid donor, men som nævnt er dette altså ikke signifikant.

Opgave 2: Mordsager

Over en årrække har man indsamlet materiale om mordsager i USA. Vi skal her se på en del af dette materiale, nemlig de sager, hvor den tiltalte blev dømt for mord.

Der var i alt 191 sorte og 483 hvide, der blev dømt for mord, og af disse fik 15 sorte og 53 hvide en dødsdom.

1. *Opstil en* 2×2*-tabel til sammenligning af sandsynlighederne for en dødsdom for hvide hhv. sorte, og kvantificer forskellen, med konfidensinterval. Ser konklusionen fornuftig ud?*

   Vi kan se, at der må være 191-15=176 sorte, der *ikke* fik dødsstraf, og tilsvarende 483-53=430 hvide. Vi indlæser derfor et datasæt med 4 linier, og laver en 2×2-tabel:

```
mord = matrix(c(176,430,15,53), nrow=2)
rownames(mord) = c("black","white")
colnames(mord) = c("nej","ja")

> mord
      nej ja
black 176 15
white 430 53
```

Vi analyserer tabellen med nedenstående kommandoer:

```
prop.table(mord,1)*100
chisq.test(mord)
chisq.test(mord,correct=F)
chisq.test(mord)$expected
fisher.test(mord)
```

Herved får vi outputtet:

```
> prop.table(mord,1)*100
           nej        ja
black 92.14660  7.853403
white 89.02692 10.973085


> chisq.test(mord)

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

data:  mord
X-squared = 1.1447, df = 1, p-value = 0.2847


> chisq.test(mord)$expected
         nej       ja
black 171.73 19.27003
white 434.27 48.72997


> fisher.test(mord)

Fisher's Exact Test for Count Data

data:  mord
p-value = 0.2578
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.777655 2.837046
sample estimates:
odds ratio
  1.445462
```

Vi ser, at såvel χ^2-testet som Fishers eksakte test giver P-værdier over 0.2, og der er altså *ikke* evidens for forskelsbehandling af hvide og sorte.

Dette betyder ikke nødvendigvis, at der ikke *er* forskelsbehandling, og kigger vi på de rå hyppigheder, ser vi, at hvide har en hyppighed af dødsstraf på 10.97%, medens sorte har 7.85%. Der er således en lille **overhyppighed af dødsstraf blandt hvide gerningsmænd**, og denne kan kvantificeres på forskellig vis.

Vi har allerede i Fishers eksakte test ovenfor fundet odds ratio for dødsstraf for hvide i forhold til sorte til 1.45, med CI=(0.78, 2.84).

Vi udregner nu differensen mellem de to sandsynligheder for dødsstraf til

```
> 0.9214660-0.8902692
[1] 0.0311968
```

og for at få konfidensgrænser på, skriver vi:

```
> prop.test(mord)

2-sample test for equality of proportions with continuity correction

data:  mord
X-squared = 1.1447, df = 1, p-value = 0.2847
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.01970467  0.08209831
sample estimates:
   prop 1    prop 2
0.9214660 0.8902692
```

Vi får altså en risikodifferens på 3.12%-point, med CI=(-1.97, 8.21) %-point.

For at udregne den relative ririko, er vi nødt til at benytte `add-on`-pakken `epitools`:

```
install.packages("epitools")
library("epitools")
```

Herved får vi outputtet:

```
> epitab(mord,method="riskratio")
$tab
      nej        p0 ja         p1 riskratio     lower    upper   p.value
black 176 0.9214660 15 0.07853403  1.000000        NA       NA        NA
white 430 0.8902692 53 0.10973085  1.397239 0.8075977 2.417389 0.2577816

$measure
[1] "wald"

$conf.level
[1] 0.95

$pvalue
[1] "fisher.exact"
```

Sammenfattende får vi således:

- Differens mellem hyppigheder:
  3.12%-point, med 95% CI=(-1.61%-point, 7.84%-point).
- Odds Ratio for dødsstraf blandt hvide i forhold til sorte:
  1.45 (0.78, 2.84).
- Relativ risiko for dødsstraf blandt hvide i forhold til sorte:
  1.40, med CI=(0.81, 2.42).

Hvis vi opdeler data efter offerets race, får vi nedenstående to 2×2*-tabeller:*

Sort offer:

	Dødsstraf?		
Dømte	nej	ja	Total
Sort	139	4	143
Hvid	16	0	16

Hvidt offer:

	Dødsstraf?		
Dømte	nej	ja	Total
Sort	37	11	48
Hvid	414	53	467

2. *Kan offerets race tænkes at være en confounder for problemstillingen?*

   En confounder er pr. definition en variabel, der hænger sammen med såvel "ekspositionen" (her gerningsmandens race) som outcome (risikoen for dødsstraf).

   Man kan let forestille sig, at der er en sammenhæng mellem offer og gerningsmands race, og man kan (desværre) også godt forestille sig, at der er en effekt af offerets race på resultatet af dommen. Så **ja**, offerets race kan sagtens tænkes at være en confounder i denne problematik.

   Vi udnytter nu de nye oplysninger til at undersøge, hvorvidt offerets race virkelig ser ud til at være en confounder.

   Først indlæser vi det udbyggede datasæt, nu med 8 linier, fordi vi opdeler efter 3 binære variable (offerets race, gerningsmandens race og udfaldet af dommen):

```
o = c(rep("white",4),rep("black",4))
g = c(rep(c(rep("white",2),rep("black",2)),2))
d = rep(c("yes","no"),4)
antal = c(53, 414, 11, 37, 0, 16, 4, 139)
```

```
> cbind(o, g, d, antal)
     o       g       d     antal
[1,] "white" "white" "yes" "53"
[2,] "white" "white" "no"  "414"
[3,] "white" "black" "yes" "11"
[4,] "white" "black" "no"  "37"
[5,] "black" "white" "yes" "0"
[6,] "black" "white" "no"  "16"
[7,] "black" "black" "yes" "4"
[8,] "black" "black" "no"  "139"
```

   Herefter laver vi den lange version af data:

```
offer=rep(o,antal)
gerningsmand=rep(g,antal)
death=rep(d,antal)
```

og så er vi klar til at lave vi to tabeller:

- En tabel, der undersøger, om der er sammenhæng mellem offerets race og gerningsmandens race (kovariaten i fokus)

```
> table(offer, gerningsmand)
        gerningsmand
offer   black white
  black   143    16
  white    48   467
```

- En tabel, der undersøger, om der er sammenhæng mellem offerets race og selve dommen (outcome)

```
> table(offer, death)
        death
offer    no yes
  black 155   4
  white 451  64
```

De tilhørende test for uafhængighed ser således ud:

```
> conf1 = table(offer, gerningsmand)

> prop.table(conf1,1)*100

            black     white
  black 89.937107 10.062893
  white  9.320388 90.679612

> chisq.test(conf1)

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

data:  conf1
X-squared = 384.85, df = 1, p-value < 2.2e-16

> fisher.test(conf1)

Fisher's Exact Test for Count Data

data:  conf1
p-value < 2.2e-16
```

```
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
  46.53595 168.26534
sample estimates:
odds ratio
    85.971
```

```
> conf2 = table(offer, death)

> prop.table(conf2,1)*100

               no       yes
  black 97.484277  2.515723
  white 87.572816 12.427184

> chisq.test(conf2)

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

data:  conf2
X-squared = 12.087, df = 1, p-value = 0.0005077

> fisher.test(conf2)

Fisher's Exact Test for Count Data

data:  conf2
p-value = 0.0001177
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
  1.992784 21.101424
sample estimates:
odds ratio
  5.490185
```

Baseret på ovenstående tabel-analyser er vi blevet overbevist om, at offerets race kan agere som confounder for gerningsmandens race, når man vurderer sidstnævntes betydning for en evt. dødsdom.

Vi ser nemlig fra den første tabel, at der i de fleste tilfælde (faktisk

$\frac{143+467}{674} = 90.5\%$ er overensstemmelse mellem offers og gerningsmands race.

Fra den anden tabel ses, at der langt oftere gives dødsdom, når der er tale om et hvidt offer.

3. *Lav en logistisk regressionsmodel for sandsynligheden for en dødsdom, med såvel offerets race som gerningsmandens race som forklarende variable.*

Da vi nu har to kovariater (race for såvel offer som gerningsmand), kan vi ikke klare os med tabel-analyse, men må gå over til logistisk regression:

```
> fuld = glm(formula=straf ~ offer + gerningsmand,
+ family=binomial(link="logit"))
```

```
> summary(fuld)

Call:
glm(formula = straf ~ offer + gerningsmand, family = binomial(link = "logit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7283  -0.4899  -0.4899  -0.2326   2.6919

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       -3.5961     0.5069  -7.094 1.30e-12 ***
offerwhite         2.4044     0.6006   4.003 6.25e-05 ***
gerningsmandwhite -0.8678     0.3671  -2.364   0.0181 *
---
```

Vi finder de tilhørende odds ratioer til fortolkning af effekterne ved at tilbagetransformere med `exp`:

```
> exp(cbind(coef(fuld), confint(fuld)))
Waiting for profiling to be done...
                                      2.5 %      97.5 %
(Intercept)        0.02743038 0.008433309  0.06489753
offerwhite        11.07226549 3.694532608 41.16558028
gerningsmandwhite  0.41987565 0.209436976  0.89221877
```

Vi ser her, at OR for dødsstraf er mere end 11 gange højere, hvis offeret er hvidt i forhold til, hvis offeret er sort.

Til gengæld ses, at hvide gerningsmænd i mindre grad for dødsstraf, når der korrigeres for offerets race (OR=0.42). Vi kan vende denne sidste om ved at skifte forten på estimaterne, hvorved vi får:

```
> exp(cbind(-coef(fuld), -confint(fuld)))
Waiting for profiling to be done...
                                    2.5 %      97.5 %
x(Intercept)       36.45592427 118.5774151 15.40890582
offerwhite          0.09031575   0.2706702  0.02429214
gerningsmandwhite   2.38165751   4.7747061  1.12080135
```

Af dette ses, at Odds Ratio for dødsstraf for sorte i forhold til hvide (gerningsmænd) estimeres til 2.38, med CI=(1.12, 4.77), altså at sorte *oftere* får dødsstraf end hvide! Og resultatet er signifikant ($P = 0.018$).

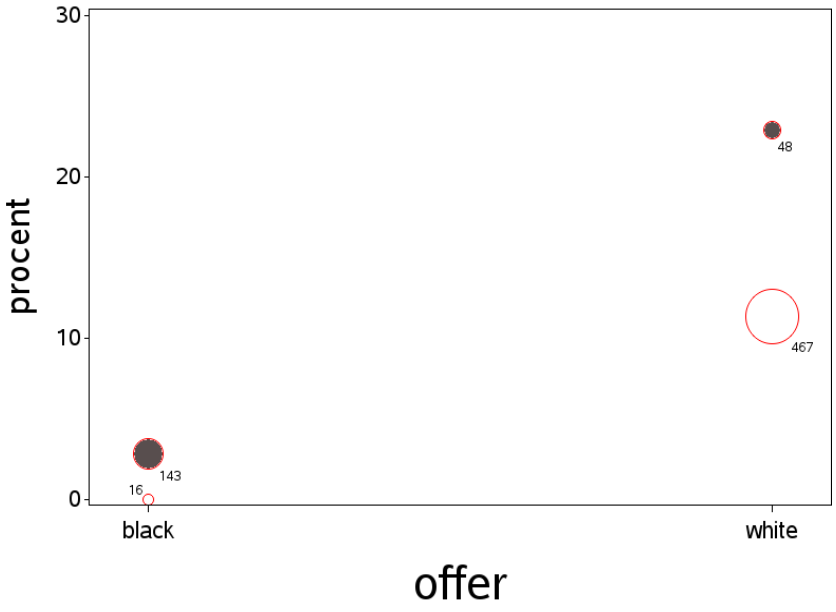
Det er i **kraftig modsætning** til det, vi fandt i spørgsmål 1!

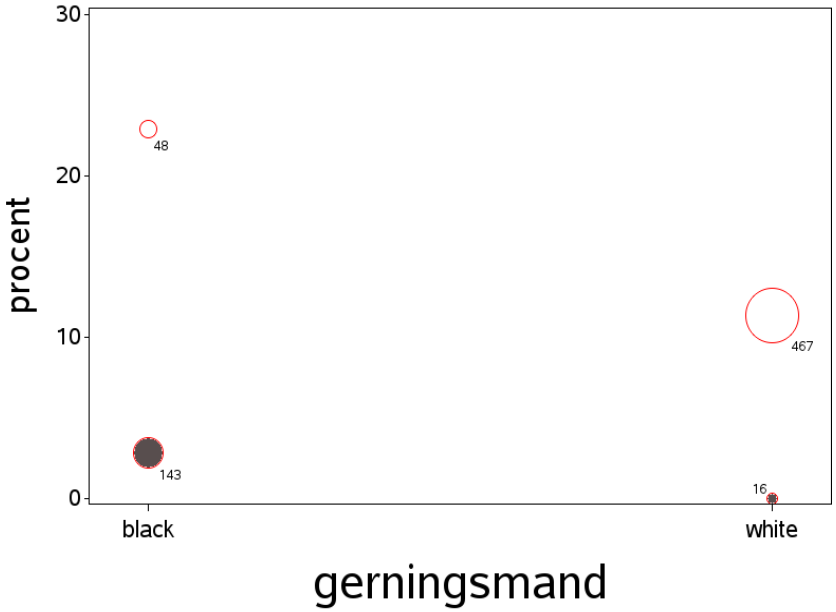
Sammenlign til resultatet i spørgsmål 1.

I spørgsmål 1 fandt vi, at Odds Ratio for dødsstraf for hvide gerningsmænd i forhold til sorte gerningsmænd var 1.45, med CI=(0.78, 2.84), svarende til, at *hvide* oftere får dødsstraf.

Vi har altså totalt fået vendt op og ned på konklusionen, fordi vi tog hensyn til offerets race. Og vi kan pludselig se en signifikans, som vi ikke kunne se tidligere. **Hvad foregår der??**

Vi kan prøve at anskueliggøre situationen ved hjælp af et par figurer, hvor vi afbilder procenten af dødsstraf, som funktion af hver af de to kovariater, med symboler svarende til den anden kovariat, og hvor *størrelsen* af symbolerne (circklerne) svarer til antallet af mordsager i den pågældende kategori:





På den første af disse tegninger ses, at hvide ofre oftere giver anledning til dødsdom, samt at sorte gerningsmænd oftere bliver dødsdømt, uanset offerets race (de sorte cirkler ligger højest).

Det samme ses i princippet på den anden figur, så det er nok mest et spørgsmål om smag og behag, hvilken man bedst kan lide.

4. *Er der grund til at tro, at der kan være interaktion mellem offerets race og gerningsmandens race? Og er der evidens for en sådan i disse data?*

   Baseret på figurerne ovenfor og kommentarerne til dem, er der intet, der tyder på en interaktion mellem gerningsmands og offers race. Alligevel vil vi undersøge sagen.

   Vi kan først prøve at analysere to 2×2-tabeller, idet vi opdeler efter offerets race, for at vurdere effekten af gerningsmandens race i hver af de to situationer.

```
white = table(gerningsmand[offer=="white"], death[offer=="white"])

black = table(gerningsmand[offer=="black"], death[offer=="black"])
```

   og vi finder så

```
> white

        no yes
  black  37  11
  white 414  53

> prop.table(white,1)*100

             no      yes
  black 77.08333 22.91667
  white 88.65096 11.34904

> chisq.test(white)

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

data:  white
X-squared = 4.3416, df = 1, p-value = 0.03719
```

```
> fisher.test(white)

Fisher's Exact Test for Count Data

data:  white
p-value = 0.03496
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.2003158 0.9955945
sample estimates:
odds ratio
 0.4315103


-----------------------------------------------------------------------

> black

        no yes
  black 139   4
  white  16   0

> prop.table(black,1)*100

                no        yes
  black  97.202797   2.797203
  white 100.000000   0.000000

> chisq.test(black)

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

data:  black
X-squared = 1.4793e-30, df = 1, p-value = 1

Warning message:
In chisq.test(black) : Chi-squared approximation may be incorrect

> fisher.test(black)

Fisher's Exact Test for Count Data

data:  black
```

```
p-value = 1
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
  0.00000 14.14727
sample estimates:
odds ratio
         0
```

Disse tabeller viser, at hvis offeret er sort, er der ingen signifikant forskel på sorte og hvide gerningsmænd (idet det dog bemærkes, at der er *yderst få* dødsdomme i denne tabel, og slet ingen for hvide gerningsmænd). Sådan som tabellen er stillet op, udregnes OR til 0.

Hvis offeret derimod er hvidt, ses der en signifikant forskel på hyppighederne af dødsdom for de to racer, idet sorte dobbelt så hyppigt får en dødsdom. Relativ risiko (ikke vist her) estimeres til 2.0 (1.1, 3.6), og odds ratio estimeres (ved at invertere) til 2.32 (1.12, 4.83).

På grund af de ganske få dødsdomme for sager med sort offer, kan vi ikke på denne baggrund sige, at der er interaktion mellem gerningsmands og offers race.

Vi kan også prøve at test det direkte i en samlet logistisk regression, ved blot at tilføje interaktionsleddet `offer*gerningsmand`:

```
int = glm(formula=straf ~ offer + gerningsmand + offer*gerningsmand,
family=binomial(link="logit"))
summary(int)
drop1(int, test = "Chisq")
exp(cbind(coef(int), confint(int)))
```

Herved får vi så:

```
> summary(int)

Call:
glm(formula = straf ~ offer + gerningsmand + offer * gerningsmand,
    family = binomial(link = "logit"))
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7215  -0.4908  -0.4908  -0.2382   2.6745

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -3.5482     0.5071  -6.996 2.63e-12 ***
offerwhite                      2.3352     0.6125   3.813 0.000137 ***
gerningsmandwhite             -13.0179   599.8864  -0.022 0.982687
offerwhite:gerningsmandwhite   12.1753   599.8865   0.020 0.983807
---
```

eller med estimation af effekten af `gerningsmand`, opdelt efter `offer`:

```
int2 = glm(formula=straf ~ offer + offer:gerningsmand,
family=binomial(link="logit"))
```

der giver:

```
> summary(int2)

Call:
glm(formula = straf ~ offer + offer:gerningsmand, family = binomial(link = "logit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7215  -0.4908  -0.4908  -0.2382   2.6745

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -3.5482     0.5071  -6.996 2.63e-12 ***
offerwhite                      2.3352     0.6125   3.813 0.000137 ***
offerblack:gerningsmandwhite  -13.0179   599.8864  -0.022 0.982687
offerwhite:gerningsmandwhite   -0.8426     0.3731  -2.258 0.023938 *
---
```

```
> exp(cbind(coef(int2), confint(int2)))
Waiting for profiling to be done...
                                                 2.5 %       97.5 %
(Intercept)                   2.877698e-02 0.008844565 6.812655e-02
offerwhite                    1.033108e+01 3.325177933 3.900769e+01
offerblack:gerningsmandwhite 2.220254e-06          NA 1.211579e+26
offerwhite:gerningsmandwhite 4.306105e-01 0.212952989 9.312923e-01
There were 18 warnings (use warnings() to see them)
```

eller ved at skifte fortegn for at få de modsatte odds ratio'er:

```
> exp(cbind(-coef(int2), -confint(int2)))
Waiting for profiling to be done...
                                                 2.5 %       97.5 %
(Intercept)                     3.475000e+01 113.0637881 1.467856e+01
offerwhite                      9.679529e-02   0.3007358 2.563597e-02
offerblack:gerningsmandwhite 4.503989e+05          NA 8.253692e-27
offerwhite:gerningsmandwhite 2.322285e+00   4.6958721 1.073777e+00
There were 18 warnings (use warnings() to see them)
```

Vi ser, at der ganske rigtigt *ikke* kan påvises nogen interaktion mellem gerningsmands og offers racer ($P = 0.98$).

For hvide ofre fås (ligesom for de opdelte analyser ovenfor) en odds ratio for dødsdom for sorte vs. hvide på 2.32 (1.07, 4.70).

For sorte ofre angives den tilsvarende odds ratio til 4.5×10^5, hvilket er en måde at indikere, at der nok er divideret med 0.

Konklusionen af vores analyser er altså:

1. Der gives langt oftere dødsdom, når offeret er hvidt i forhold til, når offeret er sort.

2. Da de fleste bliver myrdet af en af deres egen race, bevirker punkt 1, at et højt antal hvide dødsdømmes.

3. **For given race af offeret**, får flere sorte end hvide en dødsdom.