STK4900 Vår 2024
Obligatorisk innlevering 2

Adrian Duric

Oppgave 1

a)

Under nullhypotesen H_0 vil psykisk lidelse og kroppstype være uavhengige utfall, dvs.

$$P(A \cap B) = P(A)P(B|A) = P(A)P(B)$$

fordi $P(B|A) = P(B)$ når A og B er uavhengige. Dette viser at

$$p_{i,j} = Pr(A = i, B = j) = Pr(A = i)Pr(B = j) = a_i b_j$$

når $A = i$ og $B = j$ er uavhengige utfall, som må være tilfelle under nullhypotesen.

b)

Vi beregner estimater for a-verdiene som proporsjonene av mennesker kategorisert til hver psykiske lidelse, dvs.

$$a_i = \frac{N_i}{n}$$

der N_i er antallet mennesker som har lidelsen $A = i$. Vi gjør en tilsvarende beregning for b_j der b_j er antallet mennesker med kroppstypen $B = j$.

```r
# Lager tabellen
table = matrix(c(3,8,5,19,7,55,102,21,130,26,23,36,12,64,18), nrow=5)
dimnames(table)=list(
        c("moody","anxiety","autism","hyperkinetic","other"),
        c("thin","normal","overweight")
    )
n = sum(table)

# Estimerer a- og b-verdier
a1 = sum(table[1,]) / n # a_moody
a2 = sum(table[2,]) / n # a_anxiety
a3 = sum(table[3,]) / n # a_autism
a4 = sum(table[4,]) / n # a_hyperkinetic
a5 = sum(table[5,]) / n # a_other

b1 = sum(table[, 1]) / n # b_thin
b2 = sum(table[, 2]) / n # b_normal
b3 = sum(table[, 3]) / n # b_overweight
```

```r
# Beregner 95% konfidensintervall for p_4,2
N42 = table[4,2]
p42 = N42 / n
se_p = sqrt((p42 * (1 - p42)) / n)
# Bruker at sample-proporsjonen er ca. normalfordelt ifølge sentralgrenseteoremet
lb = p42 - 1.96*se_p
ub = p42 + 1.96*se_p

sprintf("Sample-estimat av proporsjon: %.3f", p42)
```

```
## [1] "Sample-estimat av proporsjon: 0.246"
```

```r
sprintf("95%% konfidensintervall: [%.3f, %.3f]", lb, ub)
```

```
## [1] "95% konfidensintervall: [0.209, 0.282]"
```

```r
sprintf("Forventet proporsjon under nullhypotesen: %.3f", a4*b2)
```

```
## [1] "Forventet proporsjon under nullhypotesen: 0.254"
```

Vi ser at forventet proporsjon er godt innenfor 95% konfidensintervallet til sample-estimatet, som vil si at nullhypotesen kan virke rimelig basert på dette resultatet alene, og bør uansett ikke forkastes på bakgrunn av det.

c)

Generelt for diskrete stokastiske variabler X (som vi har i dette eksemplet), har vi at

$$E(X) = \sum_i x_i p(x_i)$$

over alle mulige utfall i. I denne oppgaven kan vi definere $X_{i,j}$ som antallet ganger utfallet $(A = i, B = j)$ forekommer. Denne stokastiske variabelen er binomisk fordelt med $E(X_{i,j}) = np_{i,j}$. Begrunnelsen for dette er at når vi trekker ett individ fra populasjonen, bryr vi oss bare om hvorvidt utfallet for dette individet er enten $(A = i, B = j)$, eller ikke (andre kombinasjoner av klasser er irrelevante). Utfallene er med andre ord binære. Vi antar også at ulike forsøk er uavhengige av hverandre, og at $Pr(A = i, B = j)$ er lik for alle forsøkene. Vi kan derfor anse $X_{i,j}$ som binomisk fordelt. Under antagelsen om uavhengighet har har vi altså:

$$E(X_{i,j}) = np_{i,j} = na_i b_j$$

som følge av resultatet vi viste i oppgave a).

```r
# Beregner estimerte forventede verdier
e11 = n*a1*b1
e12 = n*a1*b2
e13 = n*a1*b3
e21 = n*a2*b1
e22 = n*a2*b2
e23 = n*a2*b3
e31 = n*a3*b1
e32 = n*a3*b2
```

```
e33 = n*a3*b3
e41 = n*a4*b1
e42 = n*a4*b2
e43 = n*a4*b3
e51 = n*a5*b1
e52 = n*a5*b2
e53 = n*a5*b3

sprintf("%.2f   %.2f   %.2f", e11, e12, e13)
```

```
## [1] "6.43   51.14   23.43"
```

```
sprintf("%.2f   %.2f   %.2f", e21, e22, e23)
```

```
## [1] "11.59   92.18   42.23"
```

```
sprintf("%.2f   %.2f   %.2f", e31, e32, e33)
```

```
## [1] "3.02   23.99   10.99"
```

```
sprintf("%.2f   %.2f   %.2f", e41, e42, e43)
```

```
## [1] "16.91   134.48   61.60"
```

```
sprintf("%.2f   %.2f   %.2f", e51, e52, e53)
```

```
## [1] "4.05   32.20   14.75"
```

d)

Bruker `chisq.test` for å beregne de ønskede verdiene:

```
# Beregner forventede verdier
chisq.test(table, correct=F)$expected
```

```
## Warning in chisq.test(table, correct = F): Chi-squared approximation may be
## incorrect
```

```
##                   thin    normal overweight
## moody         6.431002  51.14178   23.42722
## anxiety      11.591682  92.18147   42.22684
## autism        3.017013  23.99244   10.99055
## hyperkinetic 16.911153 134.48393   61.60491
## other         4.049149  32.20038   14.75047
```

```
# Beregner Pearson-statistikk
chisq.test(table, correct=F)
```

```
## Warning in chisq.test(table, correct = F): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table
## X-squared = 11.536, df = 8, p-value = 0.1731
```

Vi leser ut $\chi^2 = 11.536$ for $df = 8$, og ser fra en χ^2-fordelingstabell og/eller P-verdien at denne χ^2-verdien er innenfor rimelighetens grenser gitt en antagelse om uavhengighet. Basert på dette holder antagelsen om uavhengighet, og vi vil altså ikke forkaste nullhypotesen.

Oppgave 2

a)

```
eyes = matrix(scan("retinopathy.txt",skip=15),byrow=T,ncol=10)

x1 = eyes[ ,2] # gender (female 0, male 1)
x2 = eyes[ ,3] # duration since diabetes diagnosis, in years
x3 = eyes[ ,4] # edema present in one or both eyes
x4 = eyes[ ,5] # hemoglobin level
x5 = eyes[ ,6] # body mass index, bmi
x6 = eyes[ ,7] # pulse, heartbeat over 30 seconds
x7 = eyes[ ,8] # urine condition (1) or not (0)
x8 = eyes[ ,9] # diastolic blood pressure
yy = eyes[ ,10] # main outcome, 1 if retinopathy, 0 if not

aux = cbind(x2,x4,x5,x6,x8)
cor(aux)
```

```
##             x2          x4           x5           x6         x8
## x2  1.00000000 -0.06305944  0.192491072  0.038970470 0.07175727
## x4 -0.06305944  1.00000000 -0.078873903  0.193922631 0.10559820
## x5  0.19249107 -0.07887390  1.000000000 -0.005376639 0.27698182
## x6  0.03897047  0.19392263 -0.005376639  1.000000000 0.29126519
## x8  0.07175727  0.10559820  0.276981818  0.291265193 1.00000000
```

For å få konfidensintervaller for korrelasjonene i tillegg til deres estimater, bruker jeg `cor.test`.

```
# x2
cor.test(x2, x4)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x2 and x4
## t = -1.6585, df = 689, p-value = 0.09766
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.13699926  0.01157918
## sample estimates:
##         cor
## -0.06305944
```

```
cor.test(x2, x5)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x2 and x5
## t = 5.149, df = 689, p-value = 3.422e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1196243 0.2632952
## sample estimates:
##       cor
## 0.1924911
```

```
cor.test(x2, x6)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x2 and x6
## t = 1.0237, df = 689, p-value = 0.3063
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.03571751  0.11322554
## sample estimates:
##        cor
## 0.03897047
```

```
cor.test(x2, x8)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x2 and x8
## t = 1.8884, df = 689, p-value = 0.05939
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.002842106  0.145562389
## sample estimates:
##        cor
## 0.07175727
```

```
# x4
cor.test(x4, x5)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x4 and x5
## t = -2.0768, df = 689, p-value = 0.03819
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.152560593 -0.004315124
## sample estimates:
##        cor
## -0.0788739
```

```
cor.test(x4, x6)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x4 and x6
## t = 5.1887, df = 689, p-value = 2.788e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1210899 0.2646786
## sample estimates:
##       cor
## 0.1939226
```

```
cor.test(x4, x8)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x4 and x8
## t = 2.7874, df = 689, p-value = 0.005459
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.03126024 0.17877434
## sample estimates:
##       cor
## 0.1055982
```

```
# x5
cor.test(x5, x6)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x5 and x6
## t = -0.14113, df = 689, p-value = 0.8878
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.07992875  0.06923529
## sample estimates:
##          cor
## -0.005376639
```

```r
cor.test(x5, x8)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x5 and x8
## t = 7.5665, df = 689, p-value = 1.232e-13
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2066671 0.3444502
## sample estimates:
##       cor
## 0.2769818
```

```r
# x6
cor.test(x6, x8)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x6 and x8
## t = 7.9919, df = 689, p-value = 5.602e-15
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2214927 0.3580707
## sample estimates:
##       cor
## 0.2912652
```

Vi leser ut at de signifikante ikke-null korrelasjonene (hvor konfidensintervallet ikke inneholder 0) er: x_2 og x_5, x_4 og x_5, x_4 og x_6, x_4 og x_8, x_5 og x_8, og x_6 og x_8. Konfidensintervallet for korrelasjonen mellom puls (x_6) og diastolisk blodtrykk (x_8) er: $(0.2214927, 0.3580707)$, som vil si at korrelasjonen er positiv og signifikant. Det vil si at det er et positivt lineært forhold mellom målt puls og blodtrykk.

b)

Utfører logistisk regresjon for hver av kovariatene:

```r
summary(glm(yy~x1, family=binomial))
```

```
##
## Call:
## glm(formula = yy ~ x1, family = binomial)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.3610     0.1340  -10.16   <2e-16 ***
## x1           -0.1291     0.1926   -0.67    0.503
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 679.75  on 690  degrees of freedom
## Residual deviance: 679.30  on 689  degrees of freedom
## AIC: 683.3
##
## Number of Fisher Scoring iterations: 4
```

```
summary(glm(yy~x2, family=binomial))
```

```
##
## Call:
## glm(formula = yy ~ x2, family = binomial)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.78396    0.19875 -14.007   <2e-16 ***
## x2           0.09149    0.01033   8.859   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 679.75  on 690  degrees of freedom
## Residual deviance: 588.69  on 689  degrees of freedom
## AIC: 592.69
##
## Number of Fisher Scoring iterations: 4
```

```
summary(glm(yy~x3, family=binomial))
```

```
##
## Call:
## glm(formula = yy ~ x3, family = binomial)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.6601     0.1070  -15.52  < 2e-16 ***
## x3            2.7587     0.3805    7.25 4.16e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 679.75  on 690  degrees of freedom
## Residual deviance: 616.91  on 689  degrees of freedom
## AIC: 620.91
##
## Number of Fisher Scoring iterations: 4
```

```
summary(glm(yy~x4, family=binomial))
```

```
##
## Call:
## glm(formula = yy ~ x4, family = binomial)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.99532    0.49750  -4.011 6.05e-05 ***
## x4           0.05246    0.04457   1.177    0.239
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 679.75  on 690  degrees of freedom
## Residual deviance: 678.38  on 689  degrees of freedom
## AIC: 682.38
##
## Number of Fisher Scoring iterations: 4
```

```
summary(glm(yy~x5, family=binomial))
```

```
##
## Call:
## glm(formula = yy ~ x5, family = binomial)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.88757    0.52105  -5.542 2.99e-08 ***
## x5           0.06173    0.02133   2.894   0.0038 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 679.75  on 690  degrees of freedom
## Residual deviance: 671.56  on 689  degrees of freedom
## AIC: 675.56
##
## Number of Fisher Scoring iterations: 4
```

```
summary(glm(yy~x6, family=binomial))
```

```
##
## Call:
## glm(formula = yy ~ x6, family = binomial)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.38315    0.59869  -5.651  1.6e-08 ***
## x6           0.04643    0.01380   3.366 0.000764 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 679.75  on 690  degrees of freedom
## Residual deviance: 668.39  on 689  degrees of freedom
## AIC: 672.39
##
## Number of Fisher Scoring iterations: 4
```

```
summary(glm(yy~x7, family=binomial))
```

```
##
## Call:
## glm(formula = yy ~ x7, family = binomial)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.6900     0.1129 -14.975  < 2e-16 ***
## x7            1.4333     0.2366   6.057 1.39e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 679.75  on 690  degrees of freedom
## Residual deviance: 645.30  on 689  degrees of freedom
## AIC: 649.3
##
## Number of Fisher Scoring iterations: 3
```

```
summary(glm(yy~x8, family=binomial))
```

```
##
## Call:
## glm(formula = yy ~ x8, family = binomial)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.821682   0.744020  -6.481 9.14e-11 ***
## x8           0.043167   0.009208   4.688 2.76e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 679.75  on 690  degrees of freedom
## Residual deviance: 656.64  on 689  degrees of freedom
## AIC: 660.64
##
## Number of Fisher Scoring iterations: 4
```

Fra utskriften leser vi at x_2, x_3, x_5, x_6 x_7 og x_8 har tilhørende signifikante $\hat{\beta}$-estimater. I dette tilfellet er alle disse $\hat{\beta}$-verdiene også positive, som vil si at alle når en av disse kovariatene øker med 1, øker log-odds ratioen for at y inntreffer med den tilhørende $\hat{\beta}$-verdien.

c)

```
augen = glm(yy ~ x1+x2+x3+x4+x5+x6+x7+x8, family=binomial)
summary(augen)
```

```
##
## Call:
## glm(formula = yy ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8, family = binomial)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.29420    1.26670  -5.758 8.49e-09 ***
## x1           0.07829    0.23170   0.338  0.73544
## x2           0.08526    0.01136   7.502 6.27e-14 ***
## x3           2.00179    0.41659   4.805 1.55e-06 ***
## x4           0.06526    0.05442   1.199  0.23046
## x5           0.02074    0.02726   0.761  0.44681
## x6           0.02851    0.01753   1.626  0.10396
## x7           0.84327    0.29186   2.889  0.00386 **
## x8           0.02308    0.01179   1.958  0.05023 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 679.75  on 690  degrees of freedom
## Residual deviance: 520.51  on 682  degrees of freedom
## AIC: 538.51
##
## Number of Fisher Scoring iterations: 5
```

Formelen bak denne regresjonsmodellen er:

$$P(y = 1|x_1, \ldots, x_8) = \frac{exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_8 x_8)}{1 + exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_8 x_8)}$$

Dette kan også uttrykkes som:

$$log(\frac{P(y = 1|x_1, \ldots, x_8)}{1 - P(y = 1|x_1, \ldots, x_8)}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_8 x_8$$

hvor y er et binært utfall (her: har retinopati inntruffet eller ikke), x-verdiene er kovariater, og β-verdiene er regresjonskoeffisienter tilknyttet hver sin kovariat.

Fra å lese utskriften ser vi at utenom skjæringspunktet er $\hat{\beta}_2$, $\hat{\beta}_3$ og $\hat{\beta}_7$ signifikante på 95% konfidensnivå eller høyere, mens $\hat{\beta}_8$ er nesten også det (P-verdien er så vidt høyere enn 0,05). De er alle sammen positive, som vil si at om en av de tilhørende kovariatene øker med 1 mens alle andre kovariatverdier holdes fast, er log-odds ratioen estimert å øke med den tilsvarende $\hat{\beta}$-verdien, slik vi kan lese fra den andre formelen over.

Størrelsen på $\hat{\beta}$-verdiene må tolkes ut fra hva som er vanlige størrelser for de tilhørende kovariatene. Blant de signifikante $\hat{\beta}$-verdiene ser vi f.eks. at $\hat{\beta}_3$ og $\hat{\beta}_7$ er betydelig større enn $\hat{\beta}_2$ og $\hat{\beta}_8$, men de to førstnevnte tilhører også binære kovariater, dvs. x_3 og x_7 kan aldri ta verdier større enn 1.

```r
sprintf("Forventet bidrag fra kovariant x2: %.2f", mean(x2)*augen$coefficients[3])
```

```
## [1] "Forventet bidrag fra kovariant x2: 1.08"
```

```r
sprintf("Forventet bidrag fra kovariant x8: %.2f", mean(x8)*augen$coefficients[9])
```

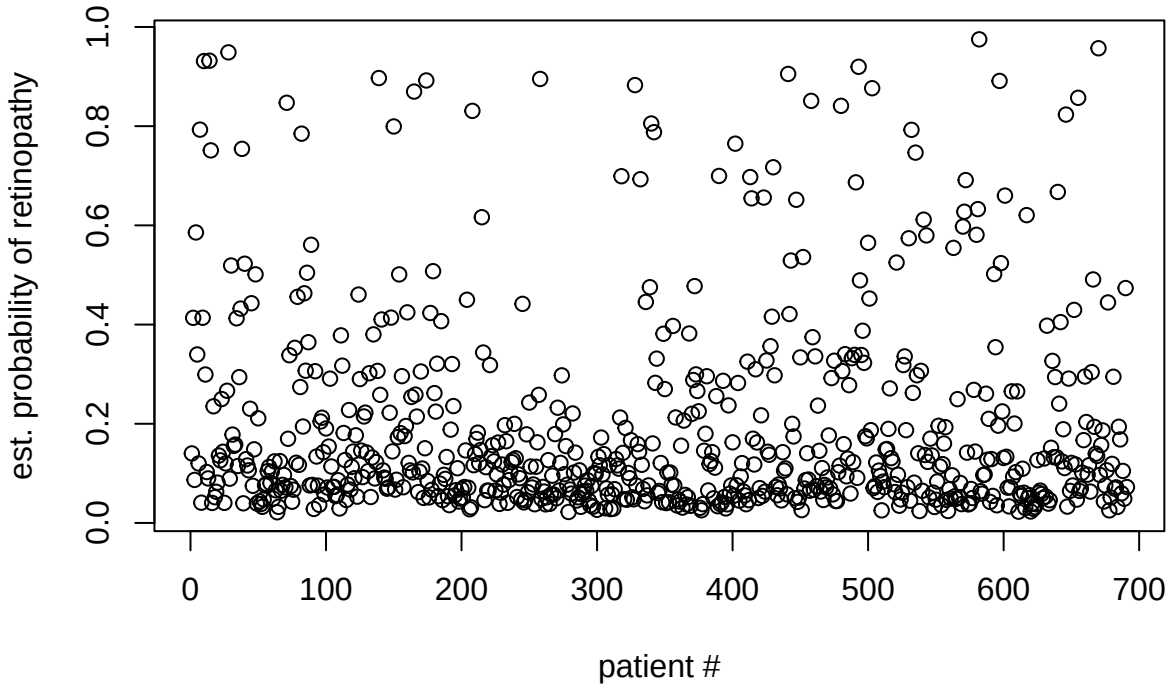
```
## [1] "Forventet bidrag fra kovariant x8: 1.78"
```

Fra å regne $\hat{\beta}_2\bar{x}_2$ og $\hat{\beta}_8\bar{x}_8$ får vi en idé av hvor stor økning i log-odds ratioen vi kan forvente å få fra en gjennomsnittlig kovariatverdi for x_2 og x_8 mtp. vanlige størrelsesforhold for kovariatene. Vi ser da at alle de signifikante regresjonskoeffisientene har sammenlignbart store bidrag til økning i log-odds ratioen. Spesielt x_3, tilstedeværelsen av ødem i ett eller begge øynene, ser ut til å øke log-odds ratioen mest, og har dessuten veldig høy signifikans.

Totalt sett anser de fire kovariatene x_2, x_3, x_7 og x_8 for å være de viktigste, da alle er innenfor et ca. 95% konfidensnivå og gir sammenlignbart store økninger i log-odds ratio. Det å utføre den fullstendige regresjonen gir fordelen av at modellen blir mer presis spesielt mtp. å identifisere konfunderende variabler og det sanne bidraget til kovariatene. I dette eksempelet så vi f.eks. i a) at x_2 og x_5 har signifikant korrelasjon, og at begges koeffisienter var signifikante i logistisk regresjon med kun én kovariat. Men i den fulle regresjonen var kun $\hat{\beta}_2$ signifikant, som tyder på at x_2 i større grad er en kausal variabel for y, mens x_5 hadde en konfunderende effekt.

d)

```r
one = 1 + 0*(1:length(yy))
X = cbind(one,x1,x2,x3,x4,x5,x6,x7,x8)
betahat = augen$coef
phats = exp(X %*% betahat)/(1 + exp(X %*% betahat))
plot(1:c(length(yy)), phats, xlab="patient #", ylab="est. probability of retinopathy")
```



Grafen viser estimert sannsynlighet for retinopati for hver pasient i datasettet basert på modellen vår i c). Vi kan ikke tolke typiske kjennetegn på pasienter med høy risiko fra grafen selv, men heller fra å se på regresjonsanalysen slik vi gjorde i c). Da så vi at noen signifikante risikofaktorer som kunne øke risikoen for retinopati er:

- Antall år pasienten har vært diagnotisert med diabetes (flere år øker risikoen)
- Om pasienten har ødem i ett eller begge øynene (hvis ja, øker risikoen)
- Om pasienten har en urinrelatert sykdom (hvis ja, øker risikoen)
- Pasientens diastoliske blodtrykk (høyere blodtrykk øker risikoen)

En pasient med høy risiko for retinopati vil altså typisk være en med høye verdier for en eller flere av disse faktorene, f.eks. en som har hatt diabetes lenge, har ødem i minst ett øye, har en urinrelatert sykdom og/eller har høyt diastolisk blodtrykk. En med lav risiko vil derimot ha lave verdier for disse faktorene. Andre mulige risikofaktorer i analysen har vist seg å ikke være signifikante, så vi kan ikke si med sikkerhet om de påvirker risikoen for retinopati eller ikke.

e)

Gitt $\sigma_{i,j}$ som kovariansen mellom U_i og U_j ($Var(U_i) = \sigma_{i,i} = \sigma_i^2$) har vi:

$$\begin{aligned} cov(a_iU_i, a_jU_j) &= E[(a_iU_i - a_i\mu_i)(a_jU_j - a_j\mu_j)] \\ &= E[a_ia_j(U_i - \mu_i)(U_j - \mu_j)] \\ &= a_ia_j\ E[(U_i - \mu_i)(U_j - \mu_j)] \\ &= a_ia_j\ cov(U_i, U_j) \\ &= a_ia_j\sigma_{i,j} \end{aligned}$$

Sammen med hvordan varians for lineære kombinasjoner av tilfeldige variabler er definert, har vi:

$$\begin{aligned} Var(a_iU_i + a_jU_j) &= a_i^2\ Var(U_i) + a_j^2\ Var(U_j) + 2a_ia_j\ cov(U_i, U_j) \\ &= cov(a_iU_i, a_iU_i) + cov(a_jU_j, a_jU_j) + 2\ cov(a_iU_i, a_jU_j) \end{aligned}$$

Dette kan utvides til flere enn to stokastiske variabler. Generelt kan vi beskrive dette som at variansen i vår lineære kombinasjon er lik summen av kovariansene av alle mulige par av stokastiske variabler i den lineære kombinasjonen (inkludert variansen til den enkelte stokastiske variabelen, som vil si å pare den med seg selv).

For vår lineære kombinasjon $a^{tr}U$ følger det da at vi har:

$$\begin{aligned} Var(a^{tr}U) &= Var(a_0U_0 + \cdots + a_8U_8) \\ &= \sum_{i=0}^{8}\sum_{j=0}^{8} cov(a_iU_i, a_jU_j) \\ &= \sum_{i=0}^{8}\sum_{j=0}^{8} a_ia_j\sigma_{i,j} \end{aligned}$$

Dette viser alt unntatt det siste leddet, nemlig at vi kan uttrykke den kvadratiske formen som en matrise-multiplikasjon. Jeg er ikke sikker på hva som er den mest elegante måten å demonstrere dette, men forsøker å utlede det direkte i det følgende:

$$a^{tr}\Sigma = \begin{bmatrix} a_0 & a_1 & \dots & a_8 \end{bmatrix} \begin{bmatrix} \sigma_0^2 & \sigma_{0,1} & \dots & \sigma_{0,8} \\ \sigma_{1,0} & \sigma_1^2 & & \vdots \\ \vdots & & \ddots & \\ \sigma_{8,0} & \dots & & \sigma_8^2 \end{bmatrix}$$

$$= \begin{bmatrix} a^{tr}\boldsymbol{\sigma}_0 & \dots & a^{tr}\boldsymbol{\sigma}_8 \end{bmatrix} \quad hvor \quad \boldsymbol{\sigma}_j = \begin{bmatrix} \sigma_{0,j} \\ \vdots \\ \sigma_{8,j} \end{bmatrix}, \quad a^{tr}\boldsymbol{\sigma}_j = \sum_{i=0}^{8} a_i\sigma_{i,j}$$

$$\begin{aligned} a^{tr}\Sigma a &= \begin{bmatrix} a^{tr}\boldsymbol{\sigma}_0 & \dots & a^{tr}\boldsymbol{\sigma}_8 \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_8 \end{bmatrix} \\ &= a_0a^{tr}\boldsymbol{\sigma}_0 + \cdots + a_8a^{tr}\boldsymbol{\sigma}_8 \\ &= a_0\sum_{i=0}^{8} a_i\sigma_{i,0} + \cdots + a_8\sum_{i=0}^{8} a_i\sigma_{i,8} = \sum_{i=0}^{8}\sum_{j=0}^{8} a_ia_j\sigma_{i,j} \quad q.e.d. \end{aligned}$$

f)

Finner den lineære prediksjonen (log-odds ratioen) $\hat{\gamma}_{jones}$ og sender den gjennom den logistiske funksjonen for å regne ut estimert $\hat{p}_{jones}$.

```r
# Estimert lineær prediktor (log-odds ratio)
pred = predict(augen, newdata=data.frame(x1=0, x2=10, x3=1, x4=10.6, x5=23.0, x6=41, x7=0, x8=77), type=
sprintf("Estimert lineær prediktor: %.4f", pred)
```

```
## [1] "Estimert lineær prediktor: -0.3253"
```

```r
# Estimert sannsynlighet
prob = predict(augen, newdata=data.frame(x1=0, x2=10, x3=1, x4=10.6, x5=23.0, x6=41, x7=0, x8=77), type=
sprintf("Estimert sannsynlighet: %.2f%%", prob*100)
```

```
## [1] "Estimert sannsynlighet: 41.94%"
```

Ved å sette `a` lik Mrs. Jones' data kan vi bruke resultatet fra e) til å regne variansen til estimatet.

```r
a = c(x0=1, x1=0, x2=10, x3=1, x4=10.6, x5=23.0, x6=41, x7=0, x8=77)
Sigma = vcov(augen)
tausq = t(a) %*% Sigma %*% a
sprintf("Varians: %.4f", tausq)
```

```
## [1] "Varians: 0.1954"
```

```r
tau = sqrt(tausq)
sprintf("Standardavvik: %.4f", tau)
```

```
## [1] "Standardavvik: 0.4420"
```

Beregner 90% konfidensintervallet for den sanne lineære prediktoren, og transformerer til 90% konfidensintervallet for den sanne sannsynligheten.

```r
lb_pred = pred - qnorm(0.95) * tau
lb_prob = exp(lb_pred) / (1 + exp(lb_pred))
ub_pred = pred + qnorm(0.95) * tau
ub_prob = exp(ub_pred) / (1 + exp(ub_pred))

sprintf("90%% konfidensintervall for lineær prediktor: (%f, %f)", lb_pred, ub_pred)
```

```
## [1] "90% konfidensintervall for lineær prediktor: (-1.052326, 0.401804)"
```

```r
sprintf("90%% konfidensintervall for sannsynlighet: (%f, %f)", lb_prob, ub_prob)
```

```
## [1] "90% konfidensintervall for sannsynlighet: (0.258779, 0.599121)"
```

g)

```r
indexM = (1:n)[x1 == 1] # 348 men
indexW = (1:n)[x1 == 0] # 343 women
eyesM = eyes[indexM, ] # dataset for the men
eyesW = eyes[indexW, ] # dataset for the women

x2M = eyesM[ ,3] # duration since diabetes diagnosis, in years
x3M = eyesM[ ,4] # edema present in one or both eyes
x4M = eyesM[ ,5] # hemoglobin level
x5M = eyesM[ ,6] # body mass index, bmi
x6M = eyesM[ ,7] # pulse, heartbeat over 30 seconds
x7M = eyesM[ ,8] # urine condition (1) or not (0)
x8M = eyesM[ ,9] # diastolic blood pressure
yyM = eyesM[ ,10] # main outcome, 1 if retinopathy, 0 if not

x2W = eyesW[ ,3] # duration since diabetes diagnosis, in years
x3W = eyesW[ ,4] # edema present in one or both eyes
x4W = eyesW[ ,5] # hemoglobin level
x5W = eyesW[ ,6] # body mass index, bmi
x6W = eyesW[ ,7] # pulse, heartbeat over 30 seconds
x7W = eyesW[ ,8] # urine condition (1) or not (0)
x8W = eyesW[ ,9] # diastolic blood pressure
yyW = eyesW[ ,10] # main outcome, 1 if retinopathy, 0 if not

augenM = glm(yyM ~ x2M+x3M+x4M+x5M+x6M+x7M+x8M, family=binomial)
augenW = glm(yyW ~ x2W+x3W+x4W+x5W+x6W+x7W+x8W, family=binomial)
summary(augenM)
```

```
##
## Call:
## glm(formula = yyM ~ x2M + x3M + x4M + x5M + x6M + x7M + x8M,
##     family = binomial)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.923845   2.199239  -3.603 0.000315 ***
## x2M          0.085989   0.021942   3.919 8.89e-05 ***
## x3M          1.686284   0.698512   2.414 0.015774 *
## x4M          0.169967   0.093884   1.810 0.070235 .
## x5M          0.067159   0.048431   1.387 0.165536
## x6M          0.008354   0.031427   0.266 0.790368
## x7M          1.412521   0.485296   2.911 0.003607 **
## x8M          0.013226   0.019076   0.693 0.488099
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 253.74  on 264  degrees of freedom
## Residual deviance: 187.50  on 257  degrees of freedom
##   (83 observations deleted due to missingness)
## AIC: 203.5
##
## Number of Fisher Scoring iterations: 5
```

```r
summary(augenW)
```

```
##
## Call:
## glm(formula = yyW ~ x2W + x3W + x4W + x5W + x6W + x7W + x8W,
##     family = binomial)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.77057    2.02015  -3.352 0.000804 ***
## x2W          0.08672    0.01828   4.743  2.1e-06 ***
## x3W          2.97796    0.83162   3.581 0.000342 ***
## x4W         -0.01102    0.08533  -0.129 0.897246
## x5W         -0.02002    0.04497  -0.445 0.656262
## x6W          0.04736    0.02674   1.771 0.076534 .
## x7W          0.39256    0.49365   0.795 0.426480
## x8W          0.02927    0.01951   1.501 0.133464
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 272.85  on 263  degrees of freedom
## Residual deviance: 209.38  on 256  degrees of freedom
##   (79 observations deleted due to missingness)
## AIC: 225.38
##
## Number of Fisher Scoring iterations: 5
```

Alle regresjonskoeffisientene sammenlignes med utgangspunkt i hvordan de var i den kombinerte regresjonen for menn og kvinner.

- $\hat{\beta}_2$: Omtrent lik som før.
- $\hat{\beta}_3$: Høyere verdi og fortsatt veldig signifikant for kvinner. Lavere verdi og signifikant på lavere konfidensnivå for menn.
- $\hat{\beta}_4$: Høyere verdi og signifikant på 90% konfidensnivå for menn. Lavere (negativ) verdi og fortsatt insignifikant for kvinner.
- $\hat{\beta}_5$: Fortsatt insignifikant for begge kjønn. Negativ verdi for kvinner.
- $\hat{\beta}_6$: Høyere verdi og signifikant på 90% konfidensnivå for kvinner. Lavere verdi og fortsatt insignifikant for menn.
- $\hat{\beta}_7$: Høyere verdi og fortsatt signifikant for menn. Lavere verdi og insignifikant for kvinner.
- $\hat{\beta}_8$: Så vidt høyere verdi for kvinner, lavere verdi for menn. Insignifikant for begge kjønn.