Assignment1

Regina Crespo Lopez Oliver (20000322-8806) & Malin Mueller (20011115-T460)

2024-09-27

```
  # Byt ÅÅMMDD mot ditt födelsedatum
  set.seed(011115) # - Malin
  # set.seed(000322) # - Regina
```

```
load("proj_data.Rdata")
modell <- glm(Resultat ~ Alder + Kon + Utbildare,
              data = data_individ,
              family = "binomial")
summary(modell)
```

```
##
## Call:
## glm(formula = Resultat ~ Alder + Kon + Utbildare, family = "binomial",
##     data = data_individ)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.151971   0.249661   0.609 0.542716
## Alder                -0.031394   0.008677  -3.618 0.000297 ***
## KonMan                0.090185   0.136394   0.661 0.508476
## UtbildareTrafikskola  0.916541   0.157659   5.813 6.12e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1353  on 999  degrees of freedom
## Residual deviance: 1297  on 996  degrees of freedom
## AIC: 1305
##
## Number of Fisher Scoring iterations: 4
```

```
  source("funktioner.R")
  y <- matrix(data_individ$Resultat, ncol = 1)
  X <- model.matrix(Resultat ~ Alder + Kon + Utbildare,
                    data = data_individ)
```

Task 1:

Verify, using functions I and NR from Part I, that the z value column of the output are Wald statistics (see the textbook page 128).

```r
theta0 <- c(0, 0, 0, 0)
NR_estimate <- NR(theta0, 5, y, X)
I_estimate <-I(NR_estimate, y, X) # Is NR the z that we hear about?
```

```r
# According to the book
true_wald_statistics <- (NR_estimate- theta0)/(sqrt(diag(solve(I(NR_estimate, y, X))))) #wald statist
inv_diag <- diag(solve(I(NR_estimate, y, X))) #the calculation of our standard error
wald <- NR_estimate/sqrt(inv_diag) # wald statistic from our method
print(true_wald_statistics)
```

```
##                             [,1]
## (Intercept)           0.6087103
## Alder                -3.6179510
## KonMan                0.6612126
## UtbildareTrafikskola  5.8134506
```

```r
print(wald)
```

```
##                             [,1]
## (Intercept)           0.6087103
## Alder                -3.6179510
## KonMan                0.6612126
## UtbildareTrafikskola  5.8134506
```

```r
# They are the same!!
```

Task 2:

Compute the generalized likelihood ratio statistics (see textbook chapter 5.5) that corresponds to the Wald statistics in Task 1 and determine the corresponding P -values. Note that your likelihood ratio statistics should be of the same order of magnitude as the squared Wald statistics (why?)

Our answer: Note that your likelihood ratio statistics should be of the same order of magnitude as the squared Wald statistic - this is because the wald statistic follows a normal distribution, whereas the likelihood ratio follows a chi square distribution.

```r
theta1 <- c(0, 0, 0)
## Iterating through X's columns, comparing and obtaining the ratio
x_name = c('intercept', 'alder', 'kon', 'utbildare')
results <- matrix(NA, nrow = length(x_name), ncol = 3)

colnames(results) <- c("x_name", "gLikelihood", "p_value")
Lp_null <-L(NR_estimate, y, X)

for (i in 1:4) { #Why 4? we have 4 columns in X

  eta <- NR(theta1, niter = 10, y = y, X = X[, -i])
  Lp_ML <- L(eta, y, X[,-i])
  gLikelihood <-  2*(log(Lp_null) - log(Lp_ML))

  p_value <- pchisq(gLikelihood, 1, lower.tail = FALSE)
```

```
  results[i, ] <- c(x_name[i], gLikelihood,p_value)
}

print(results)
```

```
##      x_name      gLikelihood        p_value
## [1,] "intercept" "0.37102834642269" "0.542444264176871"
## [2,] "alder"     "13.5762882028575" "0.000229060734894706"
## [3,] "kon"       "0.43811398878438" "0.508034160097674"
## [4,] "utbildare" "34.4805166374149" "4.30539405089157e-09"
```

Comparing the results of the generalized ratio statistics (above), with the Wald statistics (below), shows the expected outcome of the task. The generalized likelihood statistics is in the same order of magnitude as the wald statistics, and even displays similar values.

```
true_wald_statistics^2
```

```
##                             [,1]
## (Intercept)            0.3705282
## Alder                 13.0895692
## KonMan                 0.4372022
## UtbildareTrafikskola 33.7962084
```

Task 3

The score statistic can, like the likelihood ratio statistic, be generalized to the case with a nuisance parameter eta. The generalized score statistic is

TS(theta0) S(theta0, lambda_ML(theta0))ˆ I(theta0, lambda_ML(theta(-1) S(theta0 lambda_ML(theta0))

an asymptotic chiˆ2(q) distribution (notation following the textbook chapter 5.5). An advantage of this statistic is that the ML-estimate only needs to be computed under the null hypothesis. Compute the ML estimate of lambda = (theta_Alder, theta_Utbare) under H0: theta = (theta_intercept, theta_Kon) = (0,0) and use this to determine a P-value based on the generalized score statistic (a model without intercept is somewhat weird for this case, so the intercept should be included regardless of its significance).

If we want to maximize lambda L(theta, lambda) for a fixed theta != 0, the function NR needs to be modified. Instead of doing so, we use R's glm function with a so-called offset. An offset is a variable oi that is added to the linear component xitheta without a coefficient For the logistic regression with offset oi, we then get p(xi) = (1 + exp(-xitheta + oi))ˆ(-1).

```
#walder and utbilder are columns 2 and 4 - so we exclude them
#from restricted X - we only fit model under H0
eta <- NR(c(0,0), niter = 10, y = y, X = X[, c(-2,-4)])
#compute MLE for nuisance intercept and kon
print(eta)
```

```
##                    [,1]
## (Intercept) -0.31749634
## KonMan      -0.09252459
```

```
  #scover, fisher and t at mle for incercept and kon, and null values for
  # alder and utbildare
  score <- S(eta, y, X[, c(-2,-4)])
  print(score)
```

```
##                       [,1]
## (Intercept) -4.440892e-15
## KonMan      -1.243450e-14
```

```
  fisher <- I(eta, y, X[, c(-2,-4)])
```

```
  print(fisher)
```

```
##             (Intercept)   KonMan
## (Intercept)    241.5950 131.6393
## KonMan         131.6393 131.6393
```

```
  T <- t(score)%*%solve(fisher)%*%score
  print(T)
```

```
##              [,1]
## [1,] 1.755671e-30
```

```
  p_val_T <- pchisq(T, 1, lower.tail = FALSE)
  print(p_val_T)
```

```
##      [,1]
## [1,]    1
```

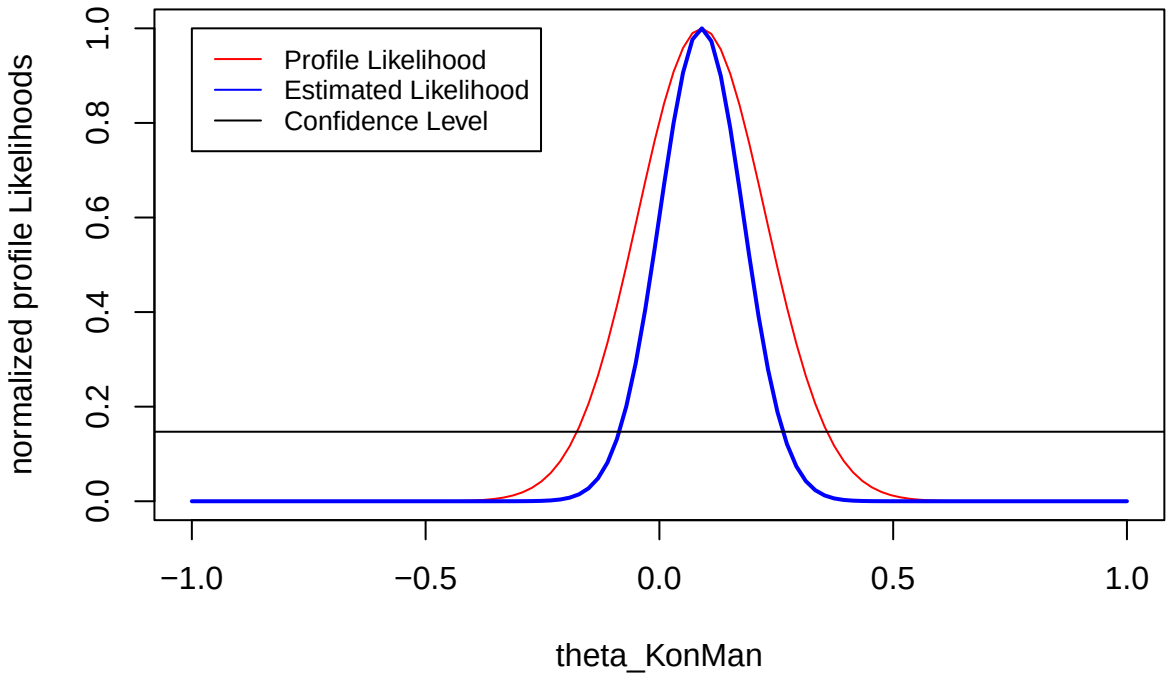
Task 4:

Compute the profile likelihood (textbook definition 5.4) for parameter theta_Kon, Lp(theta_Kon), on a suitable grid of parameter values Use these graph Lp together with the corresponding estimated likelihood. In order to determine eta_ML(theta_Kon) you may for example use the glm.fit function with an extra offset as in which gives estimates of the other coefficients when theta_Kon=.5 (as an example value). Decide a 95% confidence interval based on the profile likelihood visually from the figure by drawing a horizontal line at a suitable level (c.f. Figure 5.3b in the textbook). The choice of level should be motivated and the result compared with the corresponding Wald interval.

```
theta.Kon <- 0.5 # example value
profil <- glm.fit(x = X[, -3], y = y,
                  offset = theta.Kon * X[, 3],
                  family = binomial())
profil$coeff
```

```
##          (Intercept)                Alder UtbildareTrafikskola
##          -0.09857325          -0.03165486           1.03004754
```

```r
# likelihood profile for L(theta, eta)
  compute_profile_likelihood <- function(theta_kon, y, X){

    #set theta
    theta.Kon <- theta_kon
    #fit logistic regression model
    profile <- glm.fit(x = X[, -3], y = y,
                       offset = theta.Kon * X[, 3],
                       family = binomial())
    # extract MLEs for intercep, age, education
    eta_ML <- profile$coeff

    #create a vector with all of our parameter values
    full_params <- c(eta_ML[1], eta_ML[2], theta_kon, eta_ML[3])
    profile_L <- L(full_params, y, X)
    return(profile_L)
  }

  # 100 values of theta.kon from -1 to 1
  theta_grid <- seq(-1, 1, length.out = 100)  # Values of theta_Kon from -1 to 1

  #profile likelihood
  profile_likelihood_values <- sapply(theta_grid, FUN = function(theta_kon)
    compute_profile_likelihood(theta_kon, y, X))

  # normalize profile likelihood
  normalized_profile_likelihood <- (profile_likelihood_values -
                                      min(profile_likelihood_values)) /
    (max(profile_likelihood_values) - min(profile_likelihood_values))


  #estimated likelihood of full model L(theta_kon)
  estimated_likelihood_values <- sapply(theta_grid, function(theta) {
    L(c(NR_estimate[1], NR_estimate[2], theta, NR_estimate[4]), y, X)
  })

  normalized_estimated_likelihood <- (estimated_likelihood_values -
                                        min(estimated_likelihood_values)) /
    (max(estimated_likelihood_values) - min(estimated_likelihood_values))


  #plot
  plot(theta_grid, normalized_profile_likelihood ,
       ylab="normalized profile Likelihoods",
       xlab="theta_KonMan",type="l", col="red", ylim=c(0,1))
  lines(theta_grid, normalized_estimated_likelihood, col = "blue", lwd = 2)
  abline(h=0.147) #  The value corresponds to the likelihood threshold for a
                  # 95% confidence interval
  legend(-1,1, legend=c("Profile Likelihood", "Estimated Likelihood",
                        "Confidence Level"), col= c("red", "blue", "black"),
         lty=1, cex=0.8)
```



Where the likelihood crosses horizontal black line gives you a 95% CI for theta kon.

```
# Wald confidence interval
 wald_ci_lower <-  NR_estimate[3] - 1.96 * inv_diag[3]
 wald_ci_upper <-  NR_estimate[3] + 1.96 * inv_diag[3]
 print(wald_ci_lower)
```

```
##     KonMan
## 0.05372289
```

```
 print(wald_ci_upper)
```

```
##    KonMan
## 0.1266476
```

The wald confidence interval is smaller than the profile likelihood interval. This means, it is not accurately capturing the uncertainty in the parameter estimate. In such a case, it would prove wiser to use the profile likelihood interval.