hw1 Johanna Copeland

Johanna Copeland

2/15/2021

Question 4

Load in data

```
AD <- read_csv("../data/AD.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   y = col_double(),
##   x = col_double()
## )
```

Question #4) Implementation of Newton's method on a logistic regression model in R software. Investigators are interested in assessing the association between the baseline cognitive dysfunction and the risk of Alzheimer's disease. In the AD data set (AD.csv file), the investigators collected a random sample of n = 400 participants of an Alzheimer's disease study. The variables in the data set include standardized baseline cognitive dysfunction measurement (the column named x) and the diagnosis of Alzheimer's disease (the column named y). As a statistical consultant, you would like to use a logistic regression model to analyze the data, where the response variable Y is the diagnosis of Alzheimer's disease (1=diagnosis of Alzheimer's disease, 0=no Alzheimer's disease), and the independent variable X is the standardized baseline cognitive dysfunction measurement (the higher, the worse).

a) Summarize the two variables in the data set based on commonly used summary statistics. What is the proportion of Alzheimer's disease p0 in this sample?

```
table(AD$y)
```

```
##
##   0   1
## 281 119
```

```
summary(AD)
```

```
##        y                x
##  Min.   :0.0000   Min.   :-2.88892
##  1st Qu.:0.0000   1st Qu.:-0.57510
##  Median :0.0000   Median :-0.02664
##  Mean   :0.2975   Mean   : 0.03809
##  3rd Qu.:1.0000   3rd Qu.: 0.69590
##  Max.   :1.0000   Max.   : 2.64917
```

The proportion with Alzheimer's disease in the sample is 119/400 or 29.75%

b. Write out the logistic regression model for this data set, using alpha and beta to represent the intercept and the regression coefficient of X.

ln(p/(1-p)) = alpha- beta(x)

c. alpha.0 = ln(p/(1-p)) = ln(0.2975/ (1-0.2975)) = -0.8592 beta.0 = 0

i)

```
score.fun<-function(a0, a1, dat){
###input variables
#a0: the estimate for beta0
#a1: the estimate for beta1
#dat: the data set
 n<-nrow(dat)
 x.mat<-as.matrix(cbind(rep(1,n), dat$x)) #n x 2 design matrix
 beta.0<-as.matrix(c(a0, a1), 2,1) #2x1 regression coefficent vector
 p<-1/(1+exp(-x.mat%*%beta.0))
 #output score function, 2x1 vector
 lkhd.score=t(x.mat)%*%(dat$y - p) #likelihood score function
 obs.info=t(x.mat)%*%(x.mat*cbind(p*(1-p), p*(1-p))) #observed information
 current <- rbind(a0, a1)
#output the likelihood score and observed information as a list
 #list(lkhd.score=lkhd.score, obs.info=obs.info)
 plus1 <<- current + inv(obs.info)%*%lkhd.score
 ans <<- plus1 - current
 print(ans)
}


#v = 0
score.fun(a0 = -0.8592, a1 = 0, dat = AD)
```

```
##            [,1]
## a0 -0.01812689
## a1  0.47509569
```

```
(max(ans[1,], ans[2,]) > 10^-7)
```

```
## [1] TRUE
```

```
#v = 1
score.fun(a0 = -0.8773269, a1 = 0.4750957, dat = AD)
```

```
##            [,1]
## a0 -0.04606738
## a1  0.02476840
```

```
(max(ans[1,], ans[2,]) > 10^-7)
```

```
## [1] TRUE
```

```
#v = 2
score.fun(a0 = -0.9233943, a1 = 0.4998641, dat = AD)
```

```
##             [,1]
## a0 -0.0007251897
## a1  0.0006981776
```

```
(max(ans[1,], ans[2,]) > 10^-7)
```

```
## [1] TRUE
```

```
# v = 3
score.fun(a0 = -0.9241195, a1 = 0.5005623, dat = AD)
```

```
##             [,1]
## a0 -2.593402e-07
## a1  2.822309e-07
```

```
(max(ans[1,], ans[2,]) > 10^-7)
```

```
## [1] TRUE
```

```
plus1
```

```
##          [,1]
## a0 -0.9241198
## a1  0.5005626
```

```
#v= 4
#score.fun(a0 = -0.9241198, a1 = 0.5005626, dat = AD)
#(max(ans[1,], ans[2,]) > 10^-7)
#FALSE
```

(ii) 4 iterations until the algorithm stopped.

(iii) The estimate for for alpha is -0.9241198 and the estimate for beta is 0.5005626.

(iv) p = e^(alpha + betax)/ (1+e^(alpha + betax)) p = e^(-0.9241198 + 0.5005626(1))/ (1+e^(-0.9241198 + 0.5005626(1))) p = 0.3955

d)

```r
score.fun<-function(a0, a1, dat){
###input variables
#a0: the estimate for beta0
#a1: the estimate for beta1
#dat: the data set
 n<-nrow(dat)
 x.mat<-as.matrix(cbind(rep(1,n), dat$x)) #n x 2 design matrix
 beta.0<-as.matrix(c(a0, a1), 2,1) #2x1 regression coefficent vector
 p<-1/(1+exp(-x.mat%*%beta.0))
 #output score function, 2x1 vector
 lkhd.score=t(x.mat)%*%(dat$y - p) #likelihood score function
 obs.info=t(x.mat)%*%(x.mat*cbind(p*(1-p), p*(1-p))) #observed information
#output the likelihood score and observed information as a list
 #list(lkhd.score=lkhd.score, obs.info=obs.info)
 inv.obs.info <- inv(obs.info)
 list(lkhd.score=lkhd.score, obs.info=obs.info, inv.obs.info = inv.obs.info)
}

score.fun(a0 = -0.9241195, a1 = 0.5005623, dat = AD)
```

```
## $lkhd.score
##               [,1]
## [1,] -1.587911e-05
## [2,]  1.567626e-05
##
## $obs.info
##          [,1]     [,2]
## [1,] 79.83338 17.09553
## [2,] 17.09553 71.25305
##
## $inv.obs.info
##             [,1]        [,2]
## [1,]  0.01320451 -0.00316812
## [2,] -0.00316812  0.01479460
```

```r
sqrt(0.01320451)
```

```
## [1] 0.1149109
```

```r
sqrt(0.01479460)
```

```
## [1] 0.1216331
```

The standard errors for alpha and beta are sqrt(0.01320451) = 0.1149109 and sqrt(0.01479460) = 0.1216331 respectively. The obserserved information matrix is printed above under obs.info and the variance-covariance matrix is the inv.obs.info matrix.

e)

```r
(logreg <- glm(y ~ x, family="binomial", data = AD))
```

```
##
## Call:  glm(formula = y ~ x, family = "binomial", data = AD)
##
## Coefficients:
## (Intercept)            x
##     -0.9241       0.5006
##
## Degrees of Freedom: 399 Total (i.e. Null);  398 Residual
## Null Deviance:       487
## Residual Deviance: 468.8     AIC: 472.8
```

```r
summary(logreg)
```

```
##
## Call:
## glm(formula = y ~ x, family = "binomial", data = AD)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2770  -0.8786  -0.7147   1.2835   2.0764
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.9241     0.1149  -8.042 8.83e-16 ***
## x             0.5006     0.1216   4.115 3.87e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 486.98  on 399  degrees of freedom
## Residual deviance: 468.81  on 398  degrees of freedom
## AIC: 472.81
##
## Number of Fisher Scoring iterations: 4
```

The estimate for alpha is -0.9241 and the estimate for beta is 0.5006. These are the same as the estimates I found using the Newton Method above. The standard errors are the also the same I found in part d, those being: 0.1149 for alpha and 0.1216 for beta.