

Problem 1

a)

We consider a Poisson random variable y , which has density function

$$f(y|\lambda) = \frac{e^{-\lambda} \lambda^y}{y!}$$

which can be rewritten as an exponential family in the GLM framework as

$$f(y|\lambda) = e^{y \ln(\lambda) - \lambda - \ln(y!)} = e^{y\theta - b(\theta) - c(y, \phi, w)}$$

with $\theta = \ln(\lambda)$. Now, if we choose our link function to be exactly this mapping to the canonical parameters, that is $g(\lambda) = \ln(\lambda)$, this is called a canonical link function, also linking the expected value of the response, λ , to the linear predictor, $g(\lambda) = \eta_i = x_i^T \beta$.

Derivation of log-likelihood: The likelihood function for independent poisson random variables y_i is given as

$$L(\lambda) = \prod_{i=1}^n L_i(\lambda_i) = \prod_{i=1}^n f(y_i|\lambda_i) = \prod_{i=1}^n \frac{e^{-\lambda_i} \lambda_i^{y_i}}{y_i!}$$

Taking the logarithm of this expression, and using the fact that the logarithm of products equals the sum of the logarithms, gives us the following log-likelihood function

$$l(\lambda) = \ln(L(\lambda)) = \sum_{i=1}^n l_i(\lambda) = \sum_{i=1}^n y_i \ln(\lambda_i) - \lambda_i - \ln(y_i!)$$

Now however, we would like to express this in the GLM framework. That is, we would like to write the log-likelihood as $l(\beta)$, with β being the coefficients in our linear predictor. So we use the fact that $\ln(\lambda_i) = x_i^T \beta \implies \lambda_i = e^{x_i^T \beta}$ and rewrite the log-likelihood to arrive at our final expression

$$l(\beta) = \sum_{i=1}^n y_i x_i^T \beta - e^{x_i^T \beta} - \ln(y_i!)$$

Derivation of score function: The score function is defined as

$$s(\beta) = \frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^n s_i(\beta) = \sum_{i=1}^n \frac{\partial l_i(\beta)}{\partial \beta}$$

by expressing $s_i(\beta)$ in terms of the linear predictor $\eta_i = x_i^T \beta$ we can write

$$s_i(\beta) = \frac{\partial l_i(\eta_i(\beta))}{\partial \eta_i(\beta)} \cdot \frac{\partial \eta_i(\beta)}{\partial \beta}$$
$$s_i(\beta) = \frac{\partial (y_i \eta_i - e^{\eta_i} - \ln(y_i!))}{\partial \eta_i} \cdot \frac{\partial x_i^T \beta}{\partial \beta} = (y_i - e^{\eta_i}) \cdot x_i = (y_i - \lambda_i) \cdot x_i$$

hence we arrive at the following expression

$$s(\beta) = \sum_{i=1}^n (y_i - \lambda_i) x_i$$

Derivation of expected Fisher information matrix: By definition, the expected Fisher information matrix can be expressed as

$$F(\beta) = \text{Cov}(s(\beta)) = \sum_{i=1}^n \text{Cov}(s_i(\beta))$$

Note that $E[s(\beta)] = 0$ as $E[y_i] = \lambda_i$. Using this, and the definition of covariance we obtain

$$\sum_{i=1}^n \text{Cov}(s_i(\beta)) = \sum_{i=1}^n E\left[\left(s_i(\beta) - E[s_i(\beta)]\right)\left(s_i(\beta) - E[s_i(\beta)]\right)^T\right] = \sum_{i=1}^n E[s_i(\beta)s_i(\beta)^T]$$

using the expression for $s_i(\beta)$ from earlier yields the following

$$F(\beta) = \sum_{i=1}^n E[s_i(\beta)s_i(\beta)^T] = \sum_{i=1}^n x_i x_i^T E[(y_i - \lambda_i)^2]$$

and finally, by exploiting that $E[(y_i - \lambda_i)^2]$ is the variance of y_i , we arrive at our final expression

$$F(\beta) = \sum_{i=1}^n x_i x_i^T \lambda_i$$

b)

The implementation of myglm is showcased here:

```
dev_func <- function(a,b){
  if(a*b==0){return(-(a-b))}
  else{return(a * log(a/b) - (a-b))}

myglm <- function(formula, data, start = 1){

y <- data$y
X <- model.matrix(formula, data)

if(start){beta <- rep(0, ncol(X))}
else      {beta <- start}

#fisher score
while(TRUE){
  eta      <- as.vector((X %*% beta))
  score    <- t(X) %*% (y - exp(eta))
  E_fisher <- t(X) %*% diag(exp(eta)) %*% X
  delta    <- solve(E_fisher) %*% score
  if (all(abs(score) < 1e-04)){
    break
  }
  beta     <- beta + delta
}

#deviance residuals
yhat = exp(eta)
deviance <- 2 * sum(mapply(dev_func, y, yhat))

#cov matrix
vcoc = solve(E_fisher)

return(list(beta = beta,
            deviance = deviance,
            vcoc = vcoc))
}
```

c)

For testing the myglm function we compare the outputs to the glm functions incorporated in r, applied on a completely arbitrary simulated dataset. The outputs correspond, which indicates a functioning myglm function. The dataset for comparison, as well as outputs from both myglm and glm from r, is presented here.

```
#test simulation:
set.seed(7)
intercept = 1.4 #beta0
b1 = 0.7 #beta1, helt random verdier dette
b2 = 1.2 #beta2
N = 200 #punkter i testsett
y = rep(NA,N)
x1 = rep(NA,N)
x2 = rep(NA,N)
for (i in 1:N){
  x1[i] = rbinom(1,1,0.5)
  x2[i] = rbinom(1,1,0.5)
  expval = exp(intercept+x1[i]*b1+x2[i]*b2)
  y[i] = rpois(1,expval)
}

print("OUR FUNCTION")

## [1] "OUR FUNCTION"

testdata <- data.frame(y,x1,x2)
testresults <- myglm(y ~ x1 + x2, testdata)
testresults

## $beta
##           [,1]
## (Intercept) 1.4421002
## x1          0.7077695
## x2          1.1525622
##
## $deviance
## [1] 188.2504
##
## $vcoc
##           (Intercept)          x1          x2
## (Intercept)  0.002255937 -1.103422e-03 -1.531998e-03
## x1          -0.001103422  1.702604e-03 -1.363044e-05
## x2          -0.001531998 -1.363044e-05  2.054552e-03

print("GLM FUNKSJON")

## [1] "GLM FUNKSJON"

testresults_r <- glm(y ~ x1 + x2, family = poisson(link = log), data = testdata)
summary(testresults_r)

##
## Call:
## glm(formula = y ~ x1 + x2, family = poisson(link = log), data = testdata)
##
## Deviance Residuals:
```

```
##           Min           1Q           Median           3Q           Max
## -2.55458   -0.67473   -0.07099    0.69121    2.79742
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.44210    0.04750   30.36  <2e-16 ***
## x1           0.70777    0.04126   17.15  <2e-16 ***
## x2           1.15256    0.04533   25.43  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 1258.07  on 199  degrees of freedom
## Residual deviance: 188.25  on 197  degrees of freedom
## AIC: 1022.5
##
## Number of Fisher Scoring iterations: 4
vcov(testresults_r)
##           (Intercept)           x1           x2
## (Intercept)  0.002255927 -1.103420e-03 -1.531990e-03
## x1          -0.001103420  1.702602e-03 -1.363042e-05
## x2          -0.001531990 -1.363042e-05  2.054544e-03
```

Problem 2

We were given the dataset **veluwe.Rdata**, containing the reproductive success of 135 great tit females from Hoge Veluwe National park in the summer of 2005. The variable t denotes the day on which breeding was initialized. y is the number of fledglings leaving the nest.

The number of fledglings y_i is Poisson distributed with mean λ_i given by a Gaussian function,

$$\lambda_i = \lambda_0 e^{\frac{-(t_i - \theta)^2}{2\omega^2}}.$$

a)

There are three unknown parameters of interest in the equation above, λ_0 , ω and θ .

$\lambda_i(t_i)$ reaches its maximal value when the exponent is equal to 0, as the exponent is negative at all times. This occurs when $t_i = \theta$ (assuming $\omega > 0$), thus θ can be interpreted as the optimal time for initiating breeding. Larger values for ω implies that the function values decreases slower as the term $(t_i - \theta)^2$ increases, and ω can thus be interpreted as the amount of ‘slack’ for not starting at the optimal time θ . When the exponent is equal to 0, and the function value attains it maximum, we get $\lambda_i = \lambda_0$. This means that the parameter λ_0 is the maximal expected number of fledglings leaving the nest, and is reached when breeding is initiated at the optimal time θ .

b)

Again, we can express the problem in the poisson GLM framework, which was introduced earlier. If we choose the canonical link we have that our parameter of interest is $\ln \lambda_i$ is linked to the systematical component

$\eta = x_i^T \beta$ in the following manner,

$$x_i^T \beta = \ln \lambda_i = \ln \left(\lambda_0 e^{\frac{-(t_i - \theta)^2}{2\omega^2}} \right)$$

Now, from this, we can find the relationship between the β 's in the linear predictor, and the parameters contained in λ_i ,

$$x_i^T \beta = \ln \lambda_0 - \frac{\theta^2}{2\omega^2} + \frac{t_i \theta}{\omega^2} - \frac{t_i^2}{2\omega^2}$$

$$x_i^T \beta = \beta_0 + \beta_1 t - \beta_2 t^2$$

with

$$\beta_0 = \ln \lambda_0 - \frac{\theta^2}{2\omega^2}, \quad \beta_1 = \theta \omega^2, \quad \beta_2 = \frac{1}{2\omega^2}.$$

c)

We use the given dataset and fit it with myglm, as presented in 1b).

```
load(url("https://www.math.ntnu.no/emner/TMA4315/2020h/hoge-veluwe.Rdata"))
```

```
#2c)
```

```
results3 = myglm(y ~ I(-t^2) + t, data=data)
print(results3)
```

```
## $beta
##           [,1]
## (Intercept) 1.420130462
## I(-t^2)      0.003298608
## t           0.085183057
##
## $deviance
## [1] 277.4613
##
## $vcoc
##           (Intercept)      I(-t^2)      t
## (Intercept)  0.0797693783 -2.550195e-04 -9.308596e-03
## I(-t^2)      -0.0002550195  1.039306e-06  3.369024e-05
## t           -0.0093085957  3.369024e-05  1.159672e-03
```

```
results4 = glm(y ~ I(-t^2) + t, family = poisson(link = log), data = data)
summary(results4)
```

```
##
## Call:
## glm(formula = y ~ I(-t^2) + t, family = poisson(link = log),
##      data = data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.7797  -0.7788   0.3430   0.7438   2.2312
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.420130    0.282427   5.028 4.95e-07 ***
## I(-t^2)      0.003299    0.001019   3.236 0.00121 **
## t           0.085183    0.034053   2.502 0.01237 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 300.11  on 134  degrees of freedom
## Residual deviance: 277.46  on 132  degrees of freedom
## AIC: 740.67
##
## Number of Fisher Scoring iterations: 5
```

```
vcov(results4)
```

```
##              (Intercept)          I(-t^2)              t
## (Intercept)  0.0797650582 -2.549981e-04 -9.307971e-03
## I(-t^2)      -0.0002549981  1.039189e-06  3.368697e-05
## t            -0.0093079707  3.368697e-05  1.159579e-03
```

From the r output, we observe that the parameters of the linear predictor in the GLM framework are estimated to be

$$\hat{\beta}_0 = 1.42, \quad \hat{\beta}_1 = 0.085, \quad \hat{\beta}_2 = 0.0033.$$

d)

Here we are using the Wald-test for analysing the significance of the t^2 term. The test is formulated followingly

$$H_0 : \hat{\beta}_2 = 0 \quad vs \quad H_1 : \hat{\beta}_2 \neq 0$$

the test statistic is under the assumption of H_0 the following

$$Z = \frac{\hat{\beta}_2}{\sigma_{\hat{\beta}_2}} \sim N(0, 1)$$

where $\sigma_{\hat{\beta}_2}$ is the estimated standard deviation of the regression coefficient. The test statistic is standard normally distributed, and using a 0.05 significance level, H_0 is rejected for $|Z| > 1.96$. For our model, the Z-value was calculated to 3.24, hence we reject H_0 and conclude that there seems to be evidence of a quadratic effect of t .

```
#2d
b2_se = sqrt(results3$vcoc[2,2])
b2_t = results3$beta[2] / b2_se
cat("Wald-test value, Z: ", b2_t)
```

```
## Wald-test value, Z:  3.235631
```

e)

We are checking whether our candidate model is a good model for the data compared to the saturated model. We have the following null-hypothesis.

$$H_0 : \text{Candidate model ok} \quad vs \quad H_1 : \text{Candidate model not ok}$$

This is done by calculating the deviance of the model. Under the assumption of H_0 , the deviance is approximately chi-squared distributed, that is

$$D \sim \chi^2_{n-p_0}$$

with n being the amount of observations and p_0 is the number of parameters in the model. Using significance level $\alpha = 0.05$, we reject the null-hypothesis if

$$D > \chi^2_{0.95, n-p_0}$$

```
crit <- qchisq(.95, dim(data)[1]-length(results3$beta))
dev=results3$deviance
cat("Deviance: ", dev)
```

```
## Deviance: 277.4613
```

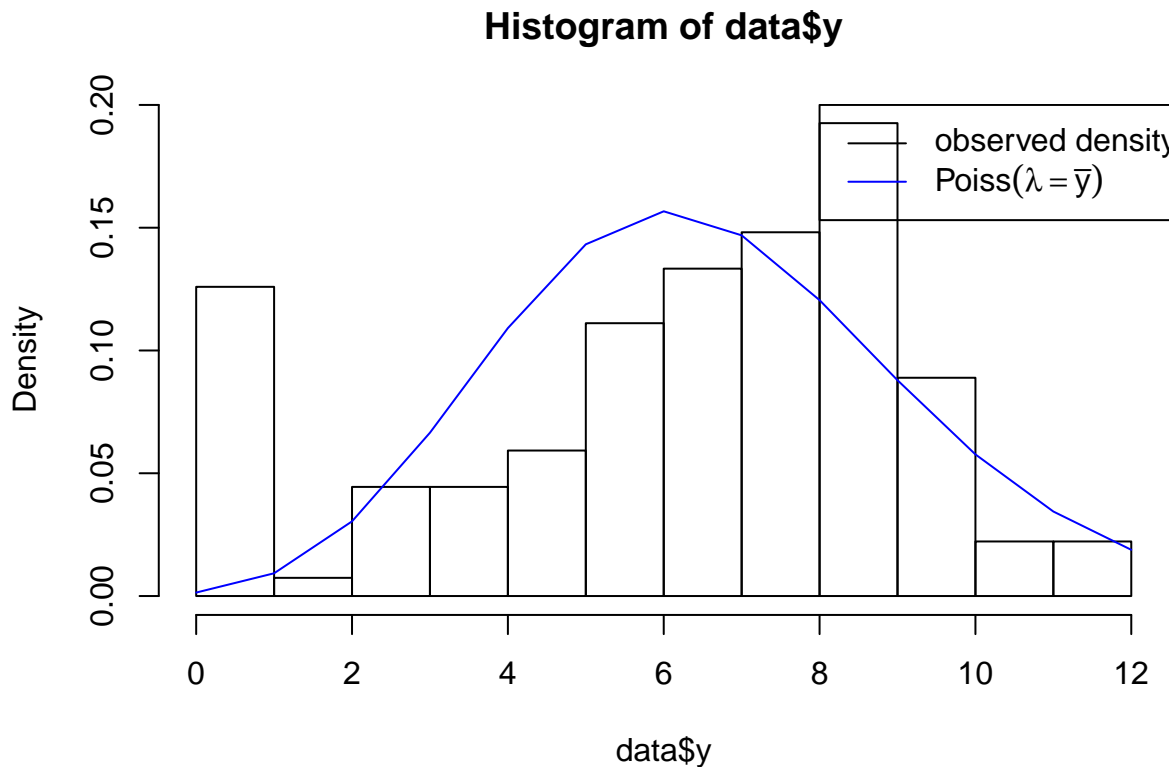
```
cat("Critical value for chisquare distribution: ", crit)
```

```
## Critical value for chisquare distribution: 159.8135
```

We observe that $D > \chi^2_{0.95, n-p_0}$ for our model, hence we reject H_0 and conclude that the model probably is not a good fit for the data.

This can be due to several assumption in the data being violated, for example the assumption that y is Poisson-distributed does not seem to hold, as evident in the histogram below.

```
hist(data$y, freq=F)
lines(x=seq(0,12), dpois(x=seq(0,12), lambda=mean(data$y)), col="blue")
legend(8,0.2, c("observed density", expression(Poiss(lambda == bar(y)))), col=c(1,4), lty=c(1,1))
```



f)

The coefficients $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\beta}_2$ are found by maximum likelihood estimation. Hence we can use the relationships found in 2b) to map the coefficient values to the maximum likelihood estimates of the original parameters.

We see from 2b) that we have

$$\beta_2 = \frac{1}{2\omega^2} \rightarrow \hat{\omega} = \sqrt{\frac{1}{2\hat{\beta}_2}}$$

which then can be used to find θ thorough β_1

$$\hat{\theta} = \hat{\beta}_1 \hat{\omega}^2$$

Using these relations we obtained

$$\hat{\omega} = 12.31, \quad \hat{\theta} = 12.91$$

Now, we are using the delta method to estimate the standard deviations of the parameters. First, for $\hat{\omega}$, again writing

$$\hat{\omega} = \sqrt{\frac{1}{2\hat{\beta}_2}}$$

the delta method is constructed in this fashion

$$Var(\hat{\omega}) = \left(\frac{\partial \hat{\omega}}{\partial \hat{\beta}_2} \right)^2 \cdot Var(\hat{\beta}_2)$$

$$Var(\hat{\omega}) = \left(-\frac{1}{2\sqrt{2\hat{\beta}_2^3}} \right)^2 \cdot Var(\hat{\beta}_2)$$

Using already obtained numerical values and $SE(\hat{\omega}) = \sqrt{Var(\hat{\omega})}$, we obtain

$$SE(\hat{\omega}) = 1.9$$

Similarly for $\hat{\theta}$, we have

$$\hat{\theta} = \hat{\beta}_1 \hat{\omega}^2 = \frac{\hat{\beta}_1}{2\hat{\beta}_2}$$

which yields the following procedure for the delta method

$$Var(\hat{\theta}) = \left(\frac{\partial \hat{\theta}}{\partial \hat{\beta}_1} \right)^2 \cdot Var(\hat{\beta}_1) + \left(\frac{\partial \hat{\theta}}{\partial \hat{\beta}_2} \right)^2 \cdot Var(\hat{\beta}_2) + 2 \cdot \left(\frac{\partial \hat{\theta}}{\partial \hat{\beta}_1} \right) \cdot \left(\frac{\partial \hat{\theta}}{\partial \hat{\beta}_2} \right) \cdot Cov(\hat{\beta}_1, \hat{\beta}_2)$$

$$Var(\hat{\theta}) = \left(\frac{1}{2\hat{\beta}_2} \right)^2 \cdot Var(\hat{\beta}_1) + \left(-\frac{\hat{\beta}_1}{2\hat{\beta}_2^2} \right)^2 \cdot Var(\hat{\beta}_2) + 2 \cdot \left(\frac{1}{2\hat{\beta}_2} \right) \cdot \left(-\frac{\hat{\beta}_1}{2\hat{\beta}_2^2} \right) \cdot Cov(\hat{\beta}_1, \hat{\beta}_2)$$

Again, using known numerical values and the square root, we obtain

$$SE(\hat{\theta}) = 1.61$$

#2f

```
omega = sqrt(1/(2*results3$beta[2]))
theta = omega^2 * results3$beta[3]
cat("omega: ", omega, "theta: ", theta)
```

```
## omega: 12.31175 theta: 12.91197
```

```
omega_var = (-1/(2*sqrt(2*results3$beta[2]^3)))^2 * results3$vcoc[2,2]
omega_se = sqrt(omega_var)
```

```
theta_var = (1/(2*results3$beta[2]))^2*results3$vcoc[3,3] + ((results3$beta[3])/(2*results3$beta[2]^2))
theta_se = sqrt(theta_var)
```

```
cat("SE(omega): ", omega_se, "SE(theta): ", theta_se)
```

```
## SE(omega): 1.902526 SE(theta): 1.609386
```


g)

The optimal breeding time θ was estimated to 12.91. Analyzing the data set, we observe that the mean breeding time was 15.94. We see that indeed the breeding times seem to be lagging behind the optimal time, supporting the claim that the evolutionary response is slower than the enviromental change.

Let us define a variable $\hat{z} = \mu - \hat{\theta}$, where μ is the mean of the breeding times, and $\hat{\theta}$ is still the estimated optimal breeding time. For analysing whether the lag is significant, we do a hypothesis test, namely

$$H_0 : \hat{z} = 0 \quad vs \quad H_1 : \hat{z} \neq 0$$

In order to do this test, we make the assumption that our new variable \hat{z} is normally distributed. Then we can use the Z-test, which under the assumption of H_0 reads

$$Z = \frac{\hat{z}}{\hat{\sigma}_z}$$

where \hat{z} is calculated numerically from our previous estimates and dataset. We also assume that μ and $\hat{\theta}$ are independent, yielding the following expression for the standard error $\hat{\sigma}_z$

$$\hat{\sigma}_z = \sqrt{Var(\mu) + Var(\hat{\theta})}$$

where $Var(\mu)$ is estimated from the dataset, and the previously calculated value $Var(\hat{\theta})$ is used. Again using a 0.05 significance level, H_0 is rejected for $|Z| > 1.96$. For our test, we obtain a value

$$Z = 1.821$$

and hence we do not reject the null-hypothesis. Hence we conclude based on this that the mean value of t is not significantly different from the optimal breeding time.

To check the actual expected affect of this lag, we calculated the exponential

$$e^{\frac{-(t-\theta)^2}{2\omega^2}}$$

for this specific lag, and the parameter values from the model. This resulted in a value of 0.97. This means that the optimal expected number of fledglings leaving the nest, λ_0 , is decreased by 3% due to this lag. This intuitively also supports the hypothesis test, in that this lag doesnt seem to be significant.

```
#2g
meant = mean(data$t)
var = var(data$t)
theta_minus_mu = meant - theta
se = sqrt(var/nrow(data) + theta_var)
Zvalue = theta_minus_mu / se
cat("Z value: ", Zvalue)

## Z value: 1.821029

evf <- function(x,a,b,c){a*exp(-((x-b)^2)/(2*c^2))}
ev = evf(meant,1,theta,omega)
cat("Value of exponential: ", ev)

## Value of exponential: 0.9702654
```

Problem 3

The parameter bootstrapping procedure is implemented and presented below.

```

#3
lambda_0 = exp(results3$beta[1]+(theta^2)/(2*omega^2))
N_b = 1000
beta0s = rep(NA,N_b)
beta1s = rep(NA,N_b)
beta2s = rep(NA,N_b)
for (i in 1:N_b){
  expval_b = mapply(evf,data$t,lambda_0,theta,omega)
  y = rpois(length(expval_b),expval_b)
  data_b = data.frame(y,data$t)
  names(data_b)[2] = "t"
  testresults_b <- myglm(y ~ I(-t^2) + t, data=data_b)
  beta0s[i] = testresults_b$beta[1]
  beta1s[i] = testresults_b$beta[3]
  beta2s[i] = testresults_b$beta[2]
}
beta0var = var(beta0s)
beta1var = var(beta1s)
beta2var = var(beta2s)

```

```
## Variances (Bootstrap):
```

```
## Beta_0 : 0.08213565
```

```
## Beta_1 : 0.00120749
```

```
## Beta_2 : 1.090143e-06
```

```
## Variances (myglm):
```

```
## Beta_0 : 0.07976938
```

```
## Beta_1 : 0.001159672
```

```
## Beta_2 : 1.039306e-06
```

Also, the variances were compared to the ones calculated through the expected Fisher information. We observe that the estimates are more or less the same.