Project 1

olarr@ntnu.no: 10031 ; tizianom@ntnu.no: 10006

1

a)

Let $Y \sim Poisson(\lambda)$ where

$$y_i \underset{iid.}{\sim} Poisson(\lambda_i)$$

The Canonical choice of link function for the Poisson distribution is

$$\begin{aligned}\eta_i &= g(\lambda_i) \\ &= log(\lambda_i) \\ &= X_i^T \beta\end{aligned}$$

Log-likelihood:

The likelihood is given by:

$$\begin{aligned}L(\lambda) &= P(Y_1 = y_1, ..., Y_n = y_n) \\ &= [y_i\text{'s are indep.}] \\ &= P(Y_1 = y_1) \cdot ... \cdot P(Y_n = y_n) \\ &= \frac{\lambda_i^{-y_1}}{y_1!} e^{-\lambda_i} \cdot ... \cdot \frac{\lambda_i^{-y_n}}{y_n!} e^{-\lambda_i} \\ &= \prod_{i=1}^{\infty} \frac{\lambda_i^{-y_i}}{y_i!} e^{-\lambda_i}\end{aligned}$$

Taking the log of the likelihood gives:

$$\begin{aligned}l(\beta) &= ln(L(\beta)) \\ &= ln(\prod_{i=1}^{\infty} \frac{\lambda_i^{-y_i}}{y_i!} e^{-\lambda_i}) \\ &= \sum_{i=1}^{n} (y_i ln(\lambda_i) - \lambda_i - ln(y_i!)) \\ &= \sum_{i=1}^{n} (y_i ln(e^{\eta_i}) - e^{\eta_i} - ln(y_i!)) \\ &= \sum_{i=1}^{n} (y_i ln(e^{X_i^T \beta}) - e^{X_i^T \beta} - ln(y_i!))\end{aligned}$$

Because y_i do not depend on β_i, we can omit $ln(y_i!)$ when finding the score function.

Score function:

$$\begin{aligned} s(\beta) &= \frac{\partial}{\partial\beta} l(\beta) \\ &= \frac{\partial}{\partial\beta}(\sum_{i=1}^{n}(y_i ln(e^{x_i^T\beta}) - e^{x_i^T\beta})) \\ &= \sum_{i=1}^{n}(y_i x_i - x_i e^{x_i^t\beta}) \\ &= \sum_{i=1}^{n}(y_i - e^{x_i^T\beta})x_i \end{aligned}$$

Observed Fisher information:

$$\begin{aligned} H(\beta) &= -\frac{\partial^2}{\partial\beta\partial\beta^T} l(\beta) \\ &= -\frac{\partial}{\partial\beta} s^T(\beta) \\ &= -\sum_{i=1}^{n}\frac{\partial}{\partial\beta}((y_i - e^{x_i^T\beta})x_i^T) \\ &= \sum_{i=1}^{n} x_i x_i^T e^{x_i^T\beta} \\ &= X^T W X \end{aligned}$$

where $W = diag(e^{X_1^T\beta}, ..., e^{X_n^T\beta}) = diag(e^{\eta_1}, ..., e^{\eta_n})$

Expected Fisher information:

$$\begin{aligned} F(\beta) &= Var[s(\beta)] \\ &= Var[\sum_{i=1}^{n}(y_i - e^{x_i^T\beta})x_i] \\ &= \sum_{i=1}^{n} x_i Var[y_i - e^{x_i^T\beta}]x_i^T \\ &= \sum_{i=1}^{n} x_i Var[y_i]x_i^T \\ &= \sum_{i=1}^{n} x_i \lambda_i x_i^T \\ &= X^T W X \end{aligned}$$

where W is the same as in the observed fisher information

b)

```r
myglm <- function(formula, data, start = 0) {

   # Defining y and X:
   y <- data$y
   X <- model.matrix(formula, data)

   # Defining beta:
   beta_0 <- rep(start, ncol(X))
   beta <- solve(t(X) %*% X) %*% t(X) %*% y

   # Fisher scoring algorithm:
   while (TRUE) {
     beta_0 <- beta
     eta <- as.vector((X %*% beta))
     scoreF <- t(X) %*% (y - exp(eta))
     A <- diag(exp(eta))
     fisherInf <- t(X) %*% A %*% X

     if (all(abs(scoreF) < 1e-04)) {
       break}

     beta <- beta + solve(fisherInf) %*% scoreF
   }

   # Deviance:
   eta <- as.vector((X %*% beta))
   yhat <- exp(eta)
   deviance <- 2 * sum(dpois(y, y, TRUE) - dpois(y, yhat, TRUE))

   # Estimated variance matrix:
   vcov <- solve(fisherInf)

   # Coefficients
   coefficients <- matrix(, nrow = 3, ncol = 2)
   colnames(coefficients) <- c("Estimate", "Std. Error")
   rownames(coefficients) <- c("(Intercept)", "t^2", "t")
   coefficients[,1] <- beta
   coefficients[,2] <- sqrt(diag(vcov))

   return (list(coefficients = coefficients,deviance = deviance,vcov = vcov))
}
```

c)

We now compare the "myglm" function to the built in "glm" and "vcov" functions in R: This will be done with the simulated data obtained from the "rpois" function.

Data:

```
set.seed(999)
t1 <- c(16,15,17,18,15,14,16,17,18,15) # t^2
t2 <- c(16,14,19,18,15,14,16,17,18,15) # t
testData <- data.frame(y = rpois(10, 10),t1,t2)
```

Our function:

```
testOur <- myglm(y ~ ., testData)
testOur
```

```
## $coefficients
##                   Estimate Std. Error
## (Intercept)  3.3556208985  1.3908842
## t^2         -0.0730511905  0.2089459
## t            0.0001284135  0.1631588
##
## $deviance
## [1] 8.166999
##
## $vcov
##             (Intercept)          t1          t2
## (Intercept)  1.93455873 -0.19419883  0.07346046
## t1          -0.19419883  0.04365840 -0.03134958
## t2           0.07346046 -0.03134958  0.02662080
```

Built in functions:

```
testBuilt <- glm(y ~ ., family = poisson(link = log), data = testData)

summary(testBuilt)$coefficients
```

```
##                   Estimate Std. Error       z value   Pr(>|z|)
## (Intercept)  3.3556200137  1.3908835  2.4125816552 0.01583999
## t1          -0.0730511496  0.2089459 -0.3496175622 0.72662573
## t2           0.0001284232  0.1631588  0.0007871056 0.99937198
```

```
summary(testBuilt)$deviance
```

```
## [1] 8.166999
```

```
vcov(testBuilt)
```

```
##             (Intercept)          t1          t2
## (Intercept)  1.93455691 -0.19419866  0.07346041
## t1          -0.19419866  0.04365838 -0.03134957
## t2           0.07346041 -0.03134957  0.02662079
```

We get the same results.

2

We are given a Poisson distribution with $\lambda_i = \lambda_0 e^{\frac{-(t_i-\theta)^2}{2\omega^2}}$. λ_i are the number of fledglings at time i (number of days after April 1)

a)

When $\theta = t_i$, the exponent of the e is zero, so $\lambda_i = \lambda_0$. This means that λ_0 is the maximum expected number of fledglings leaving the nest when $t_i = \theta$.

The ω determine the growth of the function. If it is large the function will decrease slower.

The function reaches its maximum value when the exponent is 0, i.e $t_i = \theta$. This means that the θ is the optimal time for breeding.

b)

This is a GLM because it is Poisson distributed.

Know that $y \sim Poisson(\lambda_0 e^{\frac{-(t_i-\theta)^2}{2\omega^2}})$

$$\begin{aligned}\lambda_i &= \lambda_0 e^{\frac{-(t_i-\theta)^2}{2\omega^2}} \\ &= e^{\eta} \\ &= e^{x_i^T \beta}\end{aligned}$$

Taking ln on both sides:

$$\begin{aligned}x_i^T\beta &= ln(\lambda_0) - \frac{(t_i-\theta)^2}{2\omega^2} \\ &= ln(\lambda_0) - \frac{1}{2}(\frac{t_i^2}{\omega^2} - \frac{2t_i\theta}{\omega^2} + \frac{\theta^2}{\omega^2}) \\ &= ln(\lambda_0) - \frac{\theta^2}{2\omega^2} + \frac{\theta}{\omega^2}t_i - \frac{1}{2\omega^2}t_i^2 \\ &= \beta_0 + \beta_1 t + \beta_2 t^2\end{aligned}$$

So

$$\begin{aligned}\beta_0 &= ln(\lambda_0) - \frac{\theta^2}{2\omega^2} \\ \beta_1 &= \frac{\theta}{\omega^2} \\ \beta_2 &= \frac{1}{2\omega^2}\end{aligned}$$

c)

```
load(url("https://www.math.ntnu.no/emner/TMA4315/2022h/hoge-veluwe.Rdata")) # Defaults to "data"

testOur <- myglm(y ~ I(t^2) + t, data)
testOur
```

```
## $coefficients
##                 Estimate  Std. Error
## (Intercept)  1.420130462 0.282434733
## t^2         -0.003298608 0.001019464
## t            0.085183057 0.034053955
##
## $deviance
## [1] 277.4613
##
## $vcov
##                (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
```

```
# Just to check if it works on this data aswell
# testBuilt <- glm(y ~ I(t^2) + t, family = poisson(link = log), data)
# summary(testBuilt)
# vcov(testBuilt)
```

We see that:

$$\hat{\beta}_0 = 1.42, \qquad \hat{\beta}_1 = 0.085, \qquad \hat{\beta}_2 = -0.0033$$

d)

To test if there is a quadratic effect of t (at a 0.05-level of significance), we test

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

using the Wald-test.

The test statistic under H_0 is then: $z = \frac{\hat{\beta}_2}{\hat{\sigma}_{\beta_2}} \sim N(0,1)$, where $\hat{\sigma}_{\beta_2}$ is the estimated standard deviation. At a 0.05-level of significance, H_0 is rejected if $|z| > 1.96$.

Testing:

```
z = testOur$coefficients[2,1] / (sqrt(testOur$vcov[2,2]))

# z-value from Wald-test
abs(z)
```

```
## [1] 3.235631
```

Here we see that H_0 is rejected, so this indicates that there is in fact a quadratic effect of t (at a 0.05-level of significance)

e)

We have that the deviance for our model is 277.4613. To test if this model is OK, we have the test:

$$H_0: \textit{ model ok} \qquad vs \qquad H_1: \textit{ model not ok}$$

Under H_0, the deviance is approx. χ^2_{n-p}, where n is the amount of observations and p is the number of parameters.

In our case, $n = 135$ and $p = 3$, and H_0 is then rejected if $D > \chi^2_{0.95,n-p}$

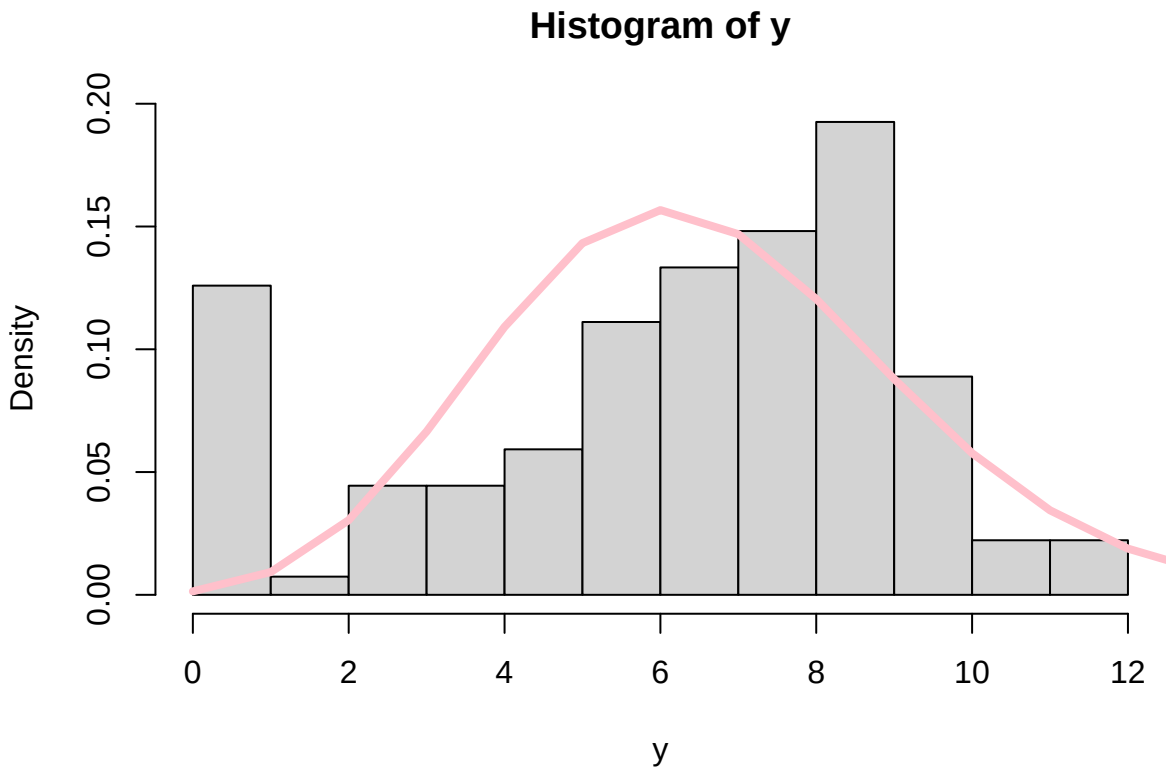
```
criticalValue <- qchisq(0.95, dim(data)[1] - length(testOur$coefficients[,1]))
criticalValue
```

```
## [1] 159.8135
```

We see that our deviance is larger than the critical value, so our model is probably not the best fit for our data.

Our assumptions for y is that it is Poisson distributed. Looking at the observed density and the poisson distribution given :

```
hist(data[,1], freq=F, main = "Histogram of y", xlab = "y")
lines(x = seq(0,20), dpois(x = seq(0,20), lambda = mean(data$y)), col="pink", lwd = 4)
```



Here the pink line is $Poisson(\lambda = \bar{y})$ and the bar-plot is the density of our data. Looking at this histogram we see that the assumption that y is Poisson distributed does not hold.

f)

We see from 2b) that

$$\hat{\beta_1} = \frac{\theta}{\omega^2}, \qquad \hat{\beta_2} = \frac{1}{2\omega^2}$$

This can be made into:

$$\hat{\omega} = \sqrt{\frac{1}{2\hat{\beta}_2}} \qquad and \qquad \hat{\theta} = \hat{\beta}_1 \hat{\omega}^2$$

Delta method is the following:

For omegas

$$\begin{aligned} Var[\hat{\omega}] &= (\frac{\partial \hat{\omega}}{\partial \hat{\beta}_2})^2 \; Var[\hat{\beta}_2] \\ &= (-\frac{1}{2\sqrt{2\hat{\beta}_2^3}})^2 \; Var[\hat{\beta}_2] \end{aligned}$$

Using this, the standard error for the omegas is then given by $SE[\hat{\omega}] = \sqrt{Var[\hat{\omega}]}$

For thetas

$$\begin{aligned} Var[\hat{\theta}] &= (\frac{\partial \hat{\theta}}{\partial \hat{\beta}_1})^2 \; Var[\hat{\beta}_1] \; + (\frac{\partial \hat{\theta}}{\partial \hat{\beta}_2})^2 \; Var[\hat{\beta}_2] \; + 2 \; (\frac{\partial \hat{\theta}}{\partial \hat{\beta}_1}) \; (\frac{\partial \hat{\theta}}{\partial \hat{\beta}_2}) \; Cov[\hat{\beta}_1, \hat{\beta}_2] \\ &= (\frac{1}{2\hat{\beta}_2})^2 \; Var[\hat{\beta}_1] \; + (-\frac{\hat{\beta}_1}{2\hat{\beta}_2^2})^2 \; Var(\hat{\beta}_2) \; + 2 \; (\frac{1}{2\hat{\beta}_2}) \; (-\frac{\hat{\beta}_1}{2\hat{\beta}_2^2})^2 \; Cov[\hat{\beta}_1, \hat{\beta}_2] \end{aligned}$$

Using this, the standard error for the thetas is then given by $SE[\hat{\theta}] = \sqrt{Var[\hat{\theta}]}$

```
omegaHat <- sqrt(abs(1/(2*testOur$coefficients[2,1])))
thetaHat <- omegaHat^2 * testOur$coefficients[3,1]
cat("omega hat:", omegaHat, "theta hat:", thetaHat)
```

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

```
omegaHat_var <- (-1/(2 * sqrt(abs(2 * testOur$coefficients[2,1]^3))))^2 * testOur$vcov[2,2]
omegaHat_se <- sqrt(omegaHat_var)

thetaHat_var <- (1/(2*testOur$coefficients[2,1]))^2*testOur$vcov[3,3] + ((testOur$coefficients[3,1])/(2
thetaHat_se <- sqrt(thetaHat_var)

cat("SE of omega:", omegaHat_se, "SE of theta:", thetaHat_se)
```

```
## SE of omega: 1.902526 SE of theta: 1.609386
```

We get

$$\hat{\omega} = 12.31175 \qquad and \qquad \hat{\theta} = 12.91197$$

and

$$SE[\hat{\omega}] = 1.902526 \qquad and \qquad SE[\hat{\theta}] = 1.609386$$

g)

From point $2.f$ we know that the estimate of the optimal breeding time $\hat{\theta}$ is 12.91. Instead, looking at the data we can easily compute the mean value of the breeding dates (i.e. by extracting data\$t) that is equal to 15.937. This difference can be interpreted as the result of a faster globally environmental change on the weather and a faster increase in temperature in the spring months compared to the evolutionary response. According to this intuitive idea, we can make a test in order to verify if the mean value of t is significantly different from the estimated optimal date based on the fitted model. So, we define a new variable $\hat{z} = \mu - \hat{\theta}$, where μ is the mean value of the column representing the breeding times and $\hat{\theta}$ is the same item as above. We can proceed with the test: $H_0 : \hat{z} = 0 \quad vs \quad H_1 : \hat{z} \neq 0$ Taking into account the assumption of gaussianity of $\hat{z}$ we can use the Z-test under H_0. We then have

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

with $Z \sim N(0, 1)$. As regards the computation of $\hat{\sigma}_z$ we need an extra hypothesis, we assume that μ and $\hat{\theta}$ are independent, so from basic theory of statistic we know that:

$$\hat{\sigma}_z = \sqrt{Var[\mu] + Var[\hat{\theta}]}$$

From this equation we can find the value of our statistic:

```
mu = mean(data$t)
n = nrow(data)
var_mu = var(data$t)/n
std_err = sqrt(var_mu + thetaHat_var)
Z_val = (mu - thetaHat) / std_err
Z_val
```

```
## [1] 1.821029
```

That is $Z = 1.821$. It is known that for a test with 0.05 significance level the null hypothesis is rejected for a value of $|Z| > 1.96$. Hence we do not have evidence to reject H_0 and to assert that there exist a significant difference between μ and $\hat{\theta}$.

h)

```
set.seed(999)

lambda <- function(t, thetaHat, c){
   exp(testOur$coefficients[1,1] + (thetaHat^2)/(2 * omegaHat^2)) * exp(-((t - thetaHat)^2)/(2 * omegaHa
}

B = 1000

bootpred <- rep(NA, B)
beta0s <- rep(NA,B)
beta1s <- rep(NA,B)
beta2s <- rep(NA,B)

for (i in 1:B){
   lambdab <- mapply(lambda, data$t, thetaHat, omegaHat)
   y <- rpois(nrow(data), lambdab)
   data_new <- data.frame(y,data$t)
   names(data_new)[2] <- "t"
   testOur_new <- myglm(y ~ I(t^2) + t, data_new)
   beta0s[i] <- testOur_new$coefficients[1,1]
   beta1s[i] <- testOur_new$coefficients[3,1]
   beta2s[i] <- testOur_new$coefficients[2,1]
}

betabootstrap <- t(cbind(var(beta0s), var(beta2s), var(beta1s)))
colnames(betabootstrap) <- c("Variance")
rownames(betabootstrap) <- c("(intercept) bootstrap", "I(t^2) bootstrap", "t bootstrap")
betabootstrap
```

```
##                            Variance
## (intercept) bootstrap 7.982436e-02
## I(t^2) bootstrap      1.042069e-06
## t bootstrap           1.151188e-03
```

Comparing this result to the variances obtained from 2c):

```
betaOur0 <- testOur$vcov[1,1]
betaOur2 <- testOur$vcov[2,2]
betaOur1 <- testOur$vcov[3,3]

betaOur <- t(cbind(betaOur0, betaOur2, betaOur1))
colnames(betaOur) <- c("Variance")
rownames(betaOur) <- c("(intercept)", "I(t^2)", "t")
betaOur
```

```
##                 Variance
## (intercept) 7.976938e-02
## I(t^2)      1.039306e-06
## t           1.159672e-03
```

Calculating the differences:

```
diff <- abs(betaOur - betabootstrap)
rownames(diff) <- c("diff (intercept)", "diff I(t^2)", "diff t")
colnames(diff) <- c("Difference")
diff
```

```
##                     Difference
## diff (intercept) 5.498462e-05
## diff I(t^2)      2.762294e-09
## diff t           8.483787e-06
```

We see that we get pretty similar results.