Project1

Erle E. Sandø, Simon Liabø

2022-09-23

Problem 1

a)

The pdf for a Poisson distribution is $f(x) = \frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!}$ where the expected value λ is given by the cononical link function $\lambda_i = e^{\eta_i}$ and opposite $\eta_i = ln(\lambda_i)$.

To find the log likelihood function we first need the likelihood function $L(\beta)$.

$$L(\beta) = \prod_{i=1}^{n} \frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!}$$

.

Log likelihood function:

$$\begin{aligned} l(\beta) &= \ln(\prod_{i=1}^{n} \frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!}) \\ &= \sum_{i=1}^{n} \ln(\frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!}) \\ &= \sum_{i=1}^{n} \ln e^{-\lambda_i} + \ln \lambda_i^{y_i} - \ln y_i! \\ &= \sum_{i=1}^{n} -\lambda_i + y_i \ln \lambda_i - \ln y_i! \\ &= \sum_{i=1}^{n} -e^{x_i^T \beta} + y_i x_i^T \beta - \ln y_i! \end{aligned} \tag{1}$$

Fisher score function:

$$
\begin{aligned}
s(\beta) &= \frac{\partial l}{\partial \beta} \\
&= \frac{\partial}{\partial \beta}(\sum_{i=1}^{n} -e^{x_i^T \beta} + y_i x_i^T \beta - \ln y_i!) \\
&= \sum_{i=1}^{n} \frac{\partial}{\partial \beta}(-e^{x_i^T \beta} + y_i x_i^T \beta - \ln y_i!) \\
&= \sum_{i=1}^{n} -x_i e^{x_i^T \beta} + y_i x_i \\
&= \sum_{i=1}^{n} x_i (y_i - e^{x_i^T \beta})
\end{aligned} \tag{2}
$$

Observed Fisher information:

$$
\begin{aligned}
H(\beta) &= -\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T} \\
&= -\frac{\partial s(\beta)}{\partial \beta^T} \\
&= -\frac{\partial}{\partial \beta^T}(\sum i = 1^n (x_i (y_i - e^{\eta_i}))) \\
&= \sum i = 1^n (\frac{\partial}{\partial \beta^T}(x_i (y_i - e^{\eta_i}))) \\
&= \sum i = 1^n (x_i \frac{\partial}{\partial \beta^T} e^{\eta_i}) \\
&= \sum i = 1^n (x_i \frac{\partial \eta_i}{\partial \beta^T} e^{\eta_i}) \\
&= \sum i = 1^n (x_i x_i^T \lambda_i)
\end{aligned} \tag{3}
$$

Expected Fisher information:

$$
\begin{aligned}
F(\beta) &= E[s_i(\beta) \cdot s_i^T(\beta)] \\
&= E[(y_i - \lambda_i) x_i \cdot (y_i - \lambda_i) x_i^T)] \\
&= E[x_i x_i^T (y_i - \lambda_i)^2] \\
&= x_i x_i^T E[(y_i - \lambda_i)^2] \\
&= x_i x_i^T \cdot Var y_i \\
&= x_i x_i^T \lambda_i
\end{aligned} \tag{4}
$$

b)

```
score = function(y, X, beta)
  {
  eta = as.vector(X %*% beta)
  lmdba = exp(eta)
  score = apply((y - lmdba) * X,2,sum)
```

```
  score
}
expected_fisher = function(X, beta)
  {
  eta = as.vector(X %*% beta)
  W = diag(exp(eta))
  t(X) %*% W %*% X
}
log_likelihood = function(y, X, beta, lmbda = exp(as.vector(X %*% beta)))
  {
  sum(ifelse(lmbda==0,0,y*log(lmbda) ) - lmbda)
}
myglm = function(formula, data, start=rep(0, ncol(model.matrix(formula, data))))
  {
  X = model.matrix(formula, data)
  response = as.character(formula)[2]
  y = data[[response]]
  beta = start

  s=1
  while (s > (1e-10)) {
    eta = as.vector(X %*% beta)
    lmbda = exp(eta)

    score_val = score(y, X, beta)
    f = expected_fisher(X, beta)

    beta = beta + solve(f) %*% score_val
    s = sum(score_val^2)
    }
  #vcov
  cov_mat = solve(f)

  #coefficients
  sd_err = sqrt(diag(cov_mat))
  coeff = cbind(beta, sd_err)
  colnames(coeff) = c("Estimate", "Std.Error")
  rownames(coeff) = paste0("beta_", seq_along(beta)-1)

  #deviance
  dev = 2 * (log_likelihood(y, X, beta, lmbda = y) - log_likelihood(y, X, beta))

  list(coefficients = coeff, deviance = dev, vcov = cov_mat)
}
```

c)

```
n = 1000
k = 2
#simulate data
beta = rnorm(k+1)
X = cbind(matrix(1,n),matrix(rnorm(n * k), nrow = n, ncol = k))
eta = as.vector(X %*% beta)
```

```
lmd = exp(eta)
y = rpois(n,lmd)
data_sim = as.data.frame(cbind(y,X[,2:3]))
#fit models
model_myglm = myglm(y~., data = data_sim)
model_glm = glm(y~., data = data_sim, family = poisson(link=log))
#evaluate
coeff_diff = mean((model_myglm$coefficients[,1] -model_glm$coefficients)^2)
coeff_diff
```

```
## [1] 9.860761e-32
```

```
vcov_diff = mean( (model_myglm$vcov - vcov(model_glm))^2)
vcov_diff
```

```
## [1] 4.148394e-21
```

The model looks good. The results are very close to those obtained with glm() and vcov().

Problem 2

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

In problem 2 we will consider a data frame containing data on the bird Great tit in the national park of Hoge Veluwe. The data was collected on 135 female birds in the summer of 2005.

The response variable is the number of fledglings leaving the nest, which relies on the time of initiate breeding and the number of fledglings for each bird, plus the timing of food resources. The number of fledglings follow a poisson distribution with expectation $\lambda_i(t_i)$. This dependence is explained by a gaussian function

$$\lambda_0 \exp(-\frac{(t_i-\theta)^2}{2\omega^2})$$

.

a)

In the expression above λ_0 is the number of fledglings when it is the highest. Θ is the mean time, E(t), that is the time when there is the most fledglings. (?) ω represents how much variance there is in the number of fledglings.

b)

A generalized linear model needs a random component, a systematic component and a link function which can give the relations between the GLM parameters contained in β and $(\lambda_0, \theta, \omega)$.

In this situation the random component is y_i which as said follows a poisson distribution. The systematic component is $\eta_i = t_i^T\beta$, and the relation can be explained by $\eta_i = \ln(\lambda_i)$ which is a canonical link function.

The link function gives the relation between β and $(\lambda_0, \theta, \omega)$:

$$\eta_i = \beta_0 + \beta_1 t_i + \beta_2 t_i^2 = \log(\lambda_i) = \log(\lambda_0 + \exp(-\frac{(t_i-\theta)^2}{2\omega^2})) = \log(\lambda_0) - \frac{(t_i-\theta)^2}{2\omega^2}$$

c)

```
m.birds = myglm(y ~ t + I(t^2), data = data)
summary(m.birds)
```

```
##              Length Class  Mode
## coefficients 6      -none- numeric
## deviance     1      -none- numeric
## vcov         9      -none- numeric
```

```
m.birds_glm = glm(y~t+I(t^2), data = data, family = poisson(link=log))
summary(m.birds_glm)
```

```
##
## Call:
## glm(formula = y ~ t + I(t^2), 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 ***
## t            0.085183   0.034053   2.502  0.01237 *
## I(t^2)      -0.003299   0.001019  -3.236  0.00121 **
## ---
## 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
```

e)

For sufficiently large number of observations, n, the deviance, D, is approximately $\chi^2_{n-p}-$distributed. We do a hypothesis test with H_0: The model is a good fit, and H_1: The model fit is bad.

```
n = length(data$t)
p = nrow(m.birds$coefficients)
df = n-p
D = m.birds$deviance

p_value = 1 - pchisq(D, df)
p_value
```

```
## [1] 2.210676e-12
```

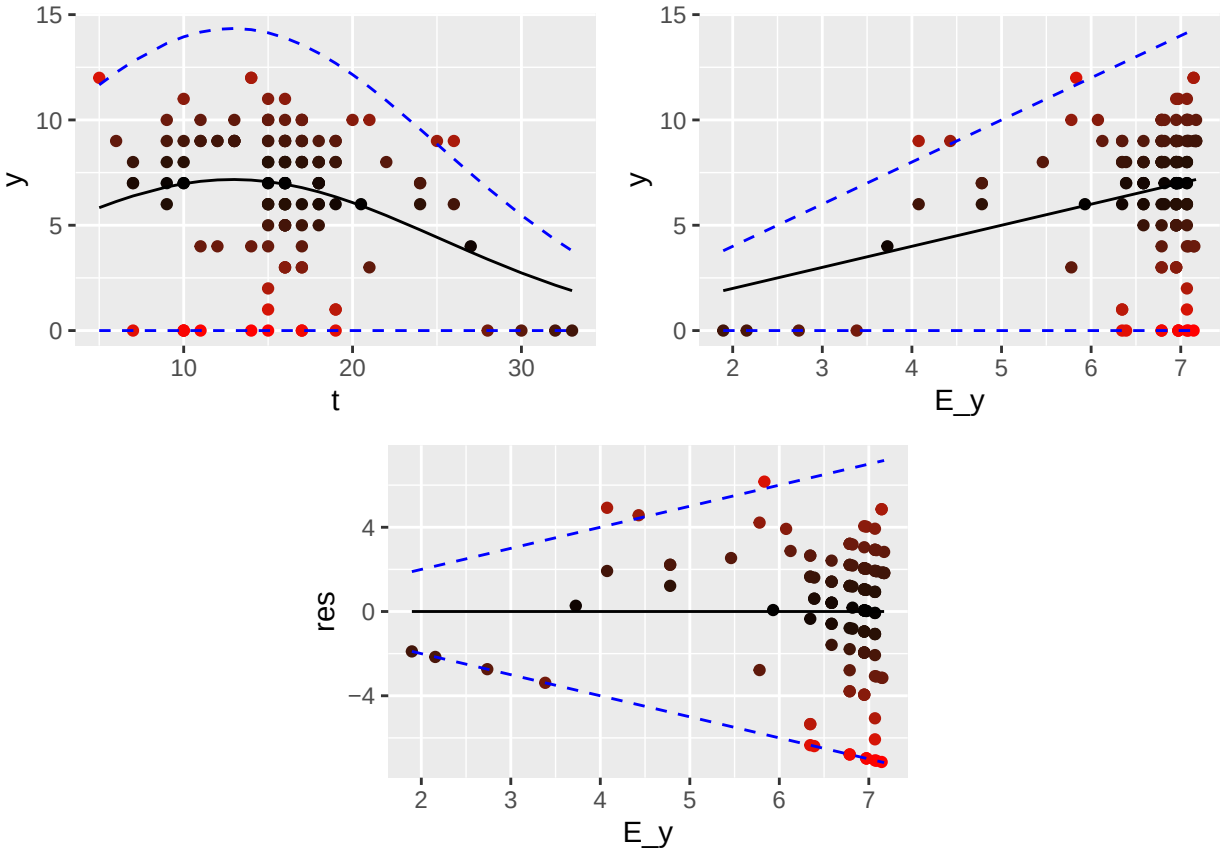
The p-value is smaller than any significance level we might want to choose. We therefore reject H_0 and conclude that the model is not a good fit.

We take a closer look at our data to examine why our model is not satisfying.

The plots below show the data in a few different ways: y against t, y against the expected value (from the

fitted model), and the residuals against the expected value. For all of the plots the black line is the expected value of y, and the dotted blue lines show the variance.

```
#calculating expected values, lambda, and residuals, based on the model fit.
beta = m.birds$coefficients[,1]
lmda = exp(beta[1])*exp(beta[2]*data$t)*exp(beta[3]*(data$t)^2)
E_y = lmda
data = cbind(data, E_y)
names(data)[3] = "E_y"
res = data$y-data$E_y
data = cbind(data, res)
names(data)[4] = "res"
```



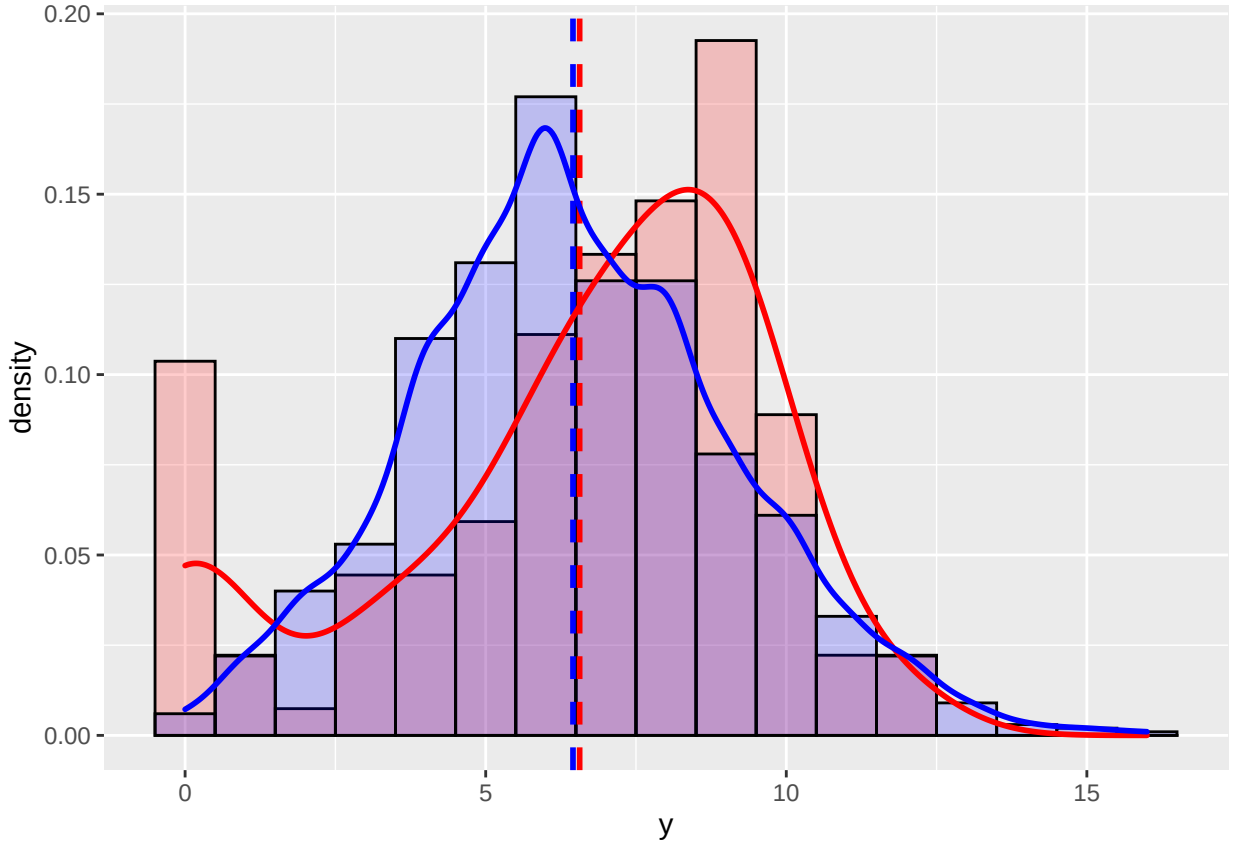
We observe, especially from the third plot, that the variance assumption seems fine. The residuals "fan out" for larger expected values as expected. The data might vary a bit more than expected, but lack of an over-dispersion parameter is probably not our main problem here.

We see that the data, y, tend to miss more frequently above what the model predicts than below. y is also 0 more frequently than we would expect from the model, both when the model predict low and high values.

We make another plot to get a better idea of what might be wrong. We simulate data from the model, using bootstrapping to sample t, and plot density plots of the data and simulated data. Red is the actual data, blue is the simulated data, and the vertical, dotted lines are the means.

```
#simulate data from bootstrap sample of t
n = 1000
t_sim = sample(data$t, n, replace=TRUE)
lmdbd_sim = exp(beta[1])*exp(beta[2]*t_sim)*exp(beta[3]*(t_sim)^2)
y_sim = rpois(n,lmdbd_sim)
```

```
d.sim = as.data.frame(cbind(y_sim,t_sim))
```



Again we see that the actual data is 0 way more than the model expects. Our model skews too much to the left to line up the means. Except for this peak at 0 however, the assumption of a poisson-distribution does not look too bad.

We speculate that the process of "whether any fledglings leave the nest" (binary with y=0 or y>0) and the process of "how many fledgelings leave the nest" might be better modeled separately.