Lesson 3 Binomial, Binary and Poisson regression

(Rev. April 2016)

3.1 Introduction

3.1.1 The normal linear model is a special case of the generalized linear model, which allows for situations where an unconstrained regression prediction may not be suitable (e.g. for necessarily positive variables), and where the variance and mean may be interrelated. Under this generalization, the observations are drawn from a density belonging to the exponential family, namely

$$p(y \mid \theta, \phi) = \exp[\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi)],$$

where θ determines the location of the distribution, and $a_i(\phi)$ has the form $a_i(\phi)=\phi/w_i$ with known weights w_i.

3.1.2 For a response y following a density within the exponential family, one has

$$E(y_i)= \mu_i = b'(\theta_i),$$

and

$$var(y_i) = b''(\theta_i)a_i(\phi).$$

This family includes both continuous densities (normal, log-normal, etc.), and discrete distributions (binomial, Bernoulli, Poisson, etc). The mean μ_i is related to a regression term (or linear predictor) η_i via a link function g,

$$g(\mu_i) = \eta_i = X_i\beta.$$

The inverse link function $h(\eta_i)$ maps the value of the linear predictor to the conditional mean μ_i.

3.1.3 Classical (maximum likelihood) estimation for discrete distributions in R uses the glm command (possibly specifying the link) as exemplified in

```
glm(y ~ x1+..+xp, data = D, family = binomial(link="logit"))
glm(y ~ x1+..+xp, data = D, family = poisson)
```

3.2 Binary and binomial regression

3.2.1 Binary and binomial regression are widely applied forms of generalized linear model. Under binary regression (with responses y=1 or y=0), the likelihood is Bernoulli

$$y_i \sim \text{Bern}(\pi_i),$$

where

$$\pi_i = \Pr(y_i = 1 | X_i)$$

is the success probability to be predicted using regressors X_i. Binomial regression involves grouped binary data, with a total of n_i subjects in the i^{th} group, and y_i of these subjects having response 1. Binomial regression predicts the probability of success given the risk profile X_i of the i^{th} group.

3.2.2 Binary and binomial regression use a distribution function F as the inverse link F=h, with

$$\pi_i = F(X_i\beta) = h(X_i\beta),$$

to ensure predicted probabilities are between 0 and 1. The link function is correspondingly $g=F^{-1}$, relating the success probability to the linear predictor, namely

$$g(\pi_i) = \eta_i = X_i\beta .$$

3.2.3 Commonly adopted options for F in binary and binomial regression are the logistic, normal and extreme value, respectively:

$$F(X_i\beta) = \exp(X_i\beta)/[1+\exp(X_i\beta)],$$

$$F(X_i\beta) = \Phi(X_i\beta),$$

$$F(X_i\beta) = 1 - \exp[-e^{X_i\beta}].$$

The corresponding links are the logit, probit and complementary log-log, respectively

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = X_i\beta,$$

$$\pi_i = \Phi(X_i\beta),$$

$$\log(-\log(1-\pi_i)) = X_i\beta.$$

with the logit being the canonical link function.

3.2.4 The interpretation of the β-coefficients depends on the link. Consider the logit link. When x_{ji} increases by 1, $\log(\frac{\pi_i}{1-\pi_i})$ increases by β_j. Therefore $\frac{\pi_i}{1-\pi_i}$ increases by a factor $\exp(\beta_j)$. For a dichotomous risk factor, this quantity is an odds ratio expressing the ratio of probabilities that y=1 for a subject with the risk factor as against a subject without the risk factor. For a continuous risk factor, the odds ratio increases by a factor $\exp(\beta_j)$ for each unit increase in x_{ji}.

3.2.5 By contrast to normal linear regression, under a normal prior

$$\beta \sim N(b_0, B_0),$$

for the regression coefficients, the posterior density $p(\beta|y)$ under binomial and binary regression is not a standard density, though log-concavity of the full conditionals allows Gibbs sampling via adaptive rejection (Gilks and Wild, 1992). While a diffuse prior on β is often convenient in linear regression, numeric stability and prior choice in binomial or binary regression may be assisted by scaling predictors (e.g. standardization). Especially in logit or probit regression a combination of a large predictor value with a large regression coefficient (as may be obtained using a diffuse prior) may cause numeric overflow.

3.2.6 A number of outputs from binary and binomial regression are of interest beyond the regression coefficients. The overall predictive or discriminatory value of a binary regression may be of interest (e.g. in clinical applications). This is summarised by the sensitivity, the proportion of positive (y=1) responses predicted correctly, and the specificity, the proportion of negative (y=0) responses predicted correctly (Tabaei and Herman, 2002; Kuk et al, 2014). In analysis involving interventions of various kinds, one may also be interested in treatment effects according to various combinations of patient background or case-mix.

3.2.7 At individual subject level, predictive checks for binary or binomial

regression involve sampling replicate data (i.e. predictions)

$y_{rep,i} \sim Bern(\pi_i)$,

or

$y_{rep,i} \sim Bin(n_i,\pi_i)$,

and comparing these with the observations, either directly or via posterior predictive tests using discrepancy measures.

3.2.8 For binary data, the probabilities

$Pr(y_{rep,i}=y_i|y)$

that replicate binary data values equal the observed values replace the probabilities

$Pr(y_{rep,i} \geq y_i|y)$

appropriate in linear regression for metric responses. Low probabilities that

$Pr(y_{rep,i}=y_i|y)$

indicate poorly fitted cases, especially when such probabilities are low compared to the overall sensitivity and specificity rates (for cases with y=1 and y=0 respectively). For binomial data the check is

$Pr(y_{rep,i} > y_i|y)+0.5Pr(y_{rep,i} = y_i|y)$

and both low and high probabilities indicate poor fit.

3.2.9 One may also calculate standardized residuals, as in

$r_i = (y_i-\pi_i)/[\pi_i(1-\pi_i)]^{0.5}$

for binary data, and

$r_i = (y_i-n_i\pi_i)/[n_i\pi_i(1-\pi_i)]^{0.5}$

for binomial data. For binomial data we also need to check for overdispersion, or greater variability in the data than would be expected based on a given simple statistical model. Binomial overdispersion occurs if variances exceed the theoretical values $n_i\pi_i(1-\pi_i)$. If there is overdispersion, the standard deviations of the β coefficients will be understated.

3.2.10 One way to do this is compare the binomial deviance D to N where N is the number of cases (see section 8.3.4 of "The Bugs Book" by Lunn

et al). Let $m_i=n_i\pi_i$. We can include commands to calculate

$$D=2\{y_i\log(y_i/m_i)+(n_i-y_i)\log([n_i-y_i]/[n_i-m_i])\}$$

and compare the posterior mean value of D to N. If the mean of D considerably exceeds N then overdispersion is present, and the model would need to be extended (a similar issue occurs with count data and Poisson overdispersion, see section 3.4)

3.3 Example Data: Drug Use Data

3.3.1 This dataset is from a study designed to compare the effectiveness of two treatment programs (TREAT) to reduce drug abuse. Other predictor variables are listed in Table 1. The data are a subsample from the University of Massachusetts AIDS Research Unit (UMARU) IMPACT Study (UIS). This was a collaborative research involving two concurrent randomized trials of residential treatment for drug abuse. The goal was to compare effectiveness of treatment programs of different planned duration in reducing drug abuse and high-risk HIV behavior.

3.3.2 These data inter alia show how categorical predictors are to be treated. In applying the glm command (conventional maximum likelihood) one needs to specify that a predictor is categorical, as exemplified in the sequence

```
# columns dfree, age, beck, ivhx, ndrugtx, race, treat, site
D <- read.table("druguse.txt",header=T)
# create categorical predictor
D$ivhx.gr <- factor(D$ivhx)
# max lkd fit
CF <- glm(dfree ~ age+beck+ivhx.gr+ndrugtx+race+treat+site, data=D, family=binomial(link="logit")).
```

Name	Description	Codes/Values
ID	Identification Code	1-575
AGE	Age at Enrolment	Years
BECK	Beck Depression Score at Admission	0.000-54.000
IVHX	IV Drug Use History at Admission	1=Never 2=Previous 3=Recent
NDRUGTX	Number of Prior Drug Treatments	0-40
RACE	Subject's Race	0=White
TREAT	Treatment Randomization Assignment	0=Short 1=Long
SITE	Treatment Site	0=A 1=B
DFREE*	Remained Drug Free for 12 Months	1=Remained Drug Free 0=Otherwise

* Outcome variable.

Table 1 Drug Use Data

3.3.3 In JAGS, the code (in yellow below) in effect treats the lowest level of the variable ivhx as a reference category, and so parallels the approach used in the maximum likelihood fit. For numeric stability, we scale age, beck and ndrugtx by dividing the original predictor values by 10. We use N(0,100) priors for the unknown coefficients. The full calling sequence is then

```
library(rjags)
library(jagstools)
setwd("C://R files")
# columns dfree, age, beck, ivhx, ndrugtx, race, treat, site
D <- read.table("druguse.txt",header=T)
# create categorical predictor
D$ivhx.gr <- factor(D$ivhx)
# classical fit
CF <- glm(dfree ~ age+beck+ivhx.gr+ndrugtx+race+treat+site, data=D, family=binomial(link="logit"))
summary(CF)
# JAGS
cat("model { for (i in 1:575) {dfree[i] ~ dbern(p[i])
dfree.rep[i] ~ dbern(p[i])
check[i] <- equals(dfree.rep[i],dfree[i])
check.y1[i] <- equals(dfree.rep[i],dfree[i])*equals(dfree[i],1)
```

```
check.y0[i] <- equals(dfree.rep[i],dfree[i])*equals(dfree[i],0)
logit(p[i]) <- beta0+beta[1]*age[i]/10+beta[2]*beck[i]/10+beta[3]*equals(ivhx[i],2)
+beta[4]*equals(ivhx[i],3)+beta[5]*ndrugtx[i]/10+beta[6]*race[i]+beta[7]*treat[i]+beta[8]*site[i]}
# predct'd probs at combination of covariates
logit(P[1]) <- beta0+beta[1]*3.7+beta[2]*1.7+beta[4]+beta[5]*0.1+beta[7]
logit(P[2]) <- beta0+beta[1]*2.7+beta[2]*1.7+beta[4]+beta[5]*0.6
totch.y1 <- sum(check.y1[]);totch.y0 <- sum(check.y0[])
# priors
beta0 ~ dnorm(0,0.001)
for (j in 1:8) {beta[j] ~ dnorm(0,0.001)
                                    sig.beta[j] <- step(beta[j])}}", file="logit.jag")
INI <- list(list(beta0=0,beta=c(0,0,0,0,0,0,0,0)),
                          list(beta0=-2,beta=c(0,0,-0.5,-0.5,0,0,0,0)))
M <- jags.model(data=D,inits=INI,n.chains=2,n.adapt=500, file="logit.jag")
R <- coda.samples(M,c("beta"),n.iter=25000)
summary(R)
# trace plot and Kernel density
plot(R)
gelman.diag(R)
gelman.plot(R)
# extract predicted probabilities, predictive concordance rates
R1 <- coda.samples(M,c("check","p"),n.iter=25000)
check.R1 <- jagsresults(R1, c("check"))
p.R1 <- jagsresults(R1, c("p"))
```

Questions

Q1 Consider the code in section 3.3.3. Which is the correct interpretation of totch.y1? Is it (a) the number of subjects with y=1, or (b) the number of subjects with actual response y=1 whose predicted response matches the actual response, or (c) the number of subjects with actual response y=0 whose predicted response matches the actual response.

Q2 Suggest how the sensitivity in the code in section 3.3.3 could be calculated from totch.y1. Dividing totch.y1 by (a) the grand total of subjects, or (b) the number of subjects with actual response y=1, or (c) the number of subjects with actual response y=0.

Q3 On the basis of your answer in Q2, include extra commands as necessary in the code to calculate the sensitivity. Is the posterior mean

sensitivity (a) under 0.25, or (b) over 0.25.

Q4 The existing code in section 3.3.3 compares predicted responses under two scenarios regarding subject background and treatment. P[1] is for white subjects at site A, aged 37, with Beck score of 17, recent IV drug use, a ndrugtx value 1 (=0.1 for covariate values divided by 10), and with longer duration treatment. P[2] is for white subjects at site A, aged 27, Beck score 17, recent IV drug use, a ndrugtx value of 6 (=0.6 for covariate values divided by 10), and with shorter duration treatment. Include a command to calculate the ratio of P[1] to P[2], and then monitor it. Is the posterior mean for this ratio (a) over 2.5 or (b) under 2.5?

Q5 Change the code in section 3.3.3 so that P[1] refers to subjects never making IV drug use. So P[1] is now to be defined for subjects aged 37, Beck score of 17, never IV drug use, ndrugtx value of 1, and with longer duration treatment. Is the posterior mean for the ratio of P[1] to P[2] now (a) over 4.5 or (b) under 4.5?

Q6 Add a command to the JAGS code in section 3.3.3 to calculate standardized residuals,

$$r_i = (y_i - \pi_i)/[\pi_i(1-\pi_i)]^{0.5}$$

It is suggested to then use the jagsresults command to extract the residuals (supposing these are coded as r[i] in JAGS), as in

```
R1 <- coda.samples(M,c("r"),n.iter=1000)
r.R1 <- jagsresults(R1, c("r"))
resid <- r.R1[,1]
```

Which cases have the highest and lowest posterior mean standardized residuals? Hint: use which.max(resid) and which.min(resid).

3.4 A Binomial Regression Example: Snoring Levels, Coronary Heart Disease and Raised Blood Pressure (Hypertension)

3.4.1 Consider data from Norton and Dunn (Br Med J, 1985) available at http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1417471/ (see Table II). Totals numbers of subjects (n_1,..,n_4) at different snoring levels are 1379 (never snore), 638 (occasional), 213 (snore nearly every night) and 254 (snore every night). These levels are assigned scores 1,3,5,6. Numbers (y_1,..,y_4) with CHD at the four snoring levels are 24, 35, 21 and 30.

Numbers with hypertension (z_1,..,z_4) at the four snoring levels are 32, 56, 26 and 47. We wish to assess a trend in levels of CHD and hypertension as snoring level increases.

3.4.2 To analyse the trend for CHD we can read the data directly into R. Also in the jags.model command the data=D element is no longer needed. The full sequence is

```
library(rjags)
D <- list(n = c(1379,638,213,254), y = c(24, 35, 21,30), z = c(32, 56, 26,47), score = c(1,3,5,6))
not.y <- n-y
not.z <- n-z
# classical binomial regression fit
CF1=glm(cbind(y,not.y) ~ score,family=binomial)
CF2=glm(cbind(z,not.z) ~ score,family=binomial)
summary(CF1)
summary(CF2)
# JAGS
cat("model { for (i in 1:4) {y[i] ~ dbin(p[i],n[i])
y.rep[i] ~ dbin(p[i],n[i])
check[i] <- step(y.rep[i]-y[i])-0.5*equals(y.rep[i],y[i])
logit(p[i]) <- beta[1]+beta[2]*score[i]}
# priors
for (j in 1:2) {beta[j] ~ dnorm(0,0.001)}}", file="binom.jag")
INI <- list(list(beta=c(0,0)),
                        list(beta=c(0,0.5)))
M <- jags.model(inits=INI,data=D,n.chains=2,n.adapt=500, file="binom.jag")
R <- coda.samples(M,c("beta"),n.iter=5000)
summary(R)
gelman.diag(R)
```

The trend slope coefficient for CHD is significant and very similar for both classical and Bayesian analysis, around 0.40.

Questions

Q7 Monitor the predictive checks in the code in 3.4.2, which derive

$$Pr(y_{rep,i} > y_i|y)+0.5Pr(y_{rep,i} = y_i|y)$$

For which observation does the check raise most concerns? Does the

model tend to (a) overpredict or (b) underpredict for this observation.

Q8 Include an extra line in the observation loop to obtain the elements needed to derive the deviance (see section 3.2.10), something like

dv[i] <- y[i]*log(…)

and an extra statement outside the loop to obtain the total deviance, something like

ScD <- 2*sum(dv[]).

Is the posterior mean deviance (a) above 5, or (b) below 5.

Q9 Modify the code in section 3.4.2 to include a second regression of hypertension rates against snoring levels. The code could look something like

```
model { for (i in 1:4) {y[i] ~ dbin(p1[i],n[i])
                        z[i] ~ dbin(p2[i],n[i])
logit(p1[i]) <- beta[1]+beta[2]*score[i]
logit(p2[i]) <- gam[1]+gam[2]*score[i]}
…}
```

Find the probability that the trend slope for hypertension exceeds that for CHD. Is this probability (a) over 0.75, or (b) under 0.75

3.5 Poisson Regression

3.5.1 Poisson regression is a natural choice when the response is a small count, though also applicable for large counts when these are observed in relation to an offset t_i, as in

$$y_i \sim \text{Po}(t_i \rho_i)$$

and the underlying rate ρ_i is low. Poisson regression has a relative rate interpretation: the effect of a predictor is multiplicative on the rate, leading to increases or decreases in the event rate (relative to the average rate) as the predictor varies. The canonical link function is the natural logarithm, namely

$$y_i \sim \text{Po}(\rho_i),$$

$$\log(\rho_i) = X_i \beta,$$

with a unit increase in predictor x_{ji} then associated with a multiplicative increase $\exp(\beta_j)$ in the rate ρ_i.

3.5.2 When there is an offset, the rate ρ_i will be corrected for extent or time of exposure. For example, Bedrick et al (1996) consider number of fish caught as Poisson with mean $\rho_i t_i$ where t_i is hours spent fishing, so that ρ_i is then the average number of fish caught per hour. In epidemiological applications, the exposure is often expected disease events E_i, such that the sum of actual and expected events are equal,

$$\sum_i y_i = \sum_i E_i .$$

Then the Poisson mean μ_i such that

$$y_i \sim Po(\mu_i),$$

will be $\mu_i = \rho_i E_i$, where ρ_i is now a measure of relative risk (with average 1) in different units i, such as patient categories or areas.

3.5.3 Predictive assessments for Poisson regression involve sampling replicates

$$y_{rep,i} \sim Po(\rho_i)$$

or

$$y_{rep,i} \sim Po(t_i \rho_i)$$

and comparing these with the observations, either directly or via posterior predictive tests using discrepancy measures. The relevant probabilities are

$$Pr(y_{rep,i} > y_i \mid y) + 0.5 Pr(y_{rep,i} = y_i \mid y),$$

and both low and high probabilities indicate poor fit .

3.5.4 Overdispersion is commonly present in count data, which if not allowed for will means that the standard deviations of regression coefficients will be understated. Overdispersion can be caused by different mechanisms, and requires more specialized regression methods. To assess whether there is overdispersion present, one may simply evaluate the Poisson deviance D and compare it to N where N is the number of cases (see section 8.3.4 of "The Bugs Book" by Lunn et al). We can include commands to calculate

$D=2\{y_i\log(y_i/\rho_i)-(y_i-\rho_i)\}$

or

$D=2\{y_i\log(y_i/[\rho_i t_i])-(y_i-\rho_i t_i)\}$ (offsets case)

and compare the mean value to N. If the mean of D considerably exceeds N then overdispersion is present and the model would need to be extended. Predictive checks for overdispersion in Poisson regression may extend to posterior predictive p-tests using criteria such as those in Dean and Lawless (1989) as discrepancy functions, evaluated both for actual and replicate data. For example, one possibility involves comparing

$$D_{obs} = 0.5\sum_{i=1}^{n}\{(y_i-\rho_i)^2 - y_i\}$$

with its counterpart

$$D_{rep} = 0.5\sum_{i=1}^{n}\{(y_{rep,i}-\rho_i)^2 - y_{rep,i}\}$$

based on replicate data (this can be done in JAGS/BUGS using a step command). If the probability is low (e.g. under 0.1) or high (over 0.9) that $D_{rep} > D_{obs}$ then the model is not accurately reproducing the data, and the source of the discrepancy includes un-modelled overdispersion.

3.5.5 One way to model over-dispersed count data is to assume a negative binomial (NB) distribution for y_i which can arise as a gamma mixture of Poisson distributions. One parameterization of the negative binomial probability density function (Zeileis et al, 2008) is

$$p(y|\mu,\theta)=\frac{\Gamma(y+\theta)}{\Gamma(\theta)y!}\frac{\mu^y\theta^\theta}{(\mu+\theta)^{y+\theta}}$$

with regression mean $\mu=X\beta$, a positive shape parameter θ, and with $\Gamma()$ denoting the gamma function. The variance function is

$$V(y_i|X_i)=\mu_i+\mu_i^2/\theta.$$

In a Bayesian analysis one would set a positive prior (e.g. gamma, exponential) on θ.

3.6 Outpatient Visits

3.6.1Deb and Trivedi (1997) analyze data on 4406 individuals covered by Medicare, a public insurance program. The objective is to model the demand for medical care (number of physician office visits, ofp) as the response with predictors:

hosp (number of hospital stays),
excellent self-perceived health status,
poor self-perceived health status,
numchron (number of chronic conditions),
male gender,
school (number of years of education),
privins (private insurance indicator)

3.6.2 We can analyse these data using classical methods, with the glm and glm.nb commands, fitting Poisson and negative binomial (NB) regressions respectively. Note that you will need to install the MASS library. Note also that in the classical fits, the health status variable (with 3 levels) is a categorical factor variable, with coefficients reported for excellent health (health=2) and poor health (health=3), with average health (health=1) as the reference. The Poisson fit suggests overdispersion, with the residual deviance 23168 compared to 4398 d.f.

3.6.3 We also fit a Bayesian model using both Poisson and NB models, the latter with a dnegbin density. Because of the large sample size we run a batch of relatively few iterations (say 1000) and use the Gelman-Rubin checks after each batch to check convergence. If there is no convergence then another batch of 1000 iterations can be run. In fact convergence is obtained after 1000 iterations, under both Poisson and NB regressions.

3.6.4 We find similar estimates of θ from the Bayesian and classical fits (around 1.2). As would be expected, the regression coefficients have less precision under the NB fit. For example, the coefficient on hosp has posterior standard deviation 0.022 under the NB model as compared to 0.006 under the Poisson model.

3.6.5 The full command sequence is

```
library(rjags); library(jagstools)
setwd("C://R files")
# columns headed ofp, hosp, health,numchron,gendermale,school,privins
D <- read.table("debtriv.txt",header=T)
D$health.gr <- factor(D$health)
CM1 <- glm(ofp ~ hosp+health.gr+numchron+gendermale+school+privins, data = D, family = poisson)
library(MASS)
CM2 <- glm.nb(ofp ~ hosp+health.gr+numchron+gendermale+school+privins, data = D)
# JAGS Poisson
cat("model { for (i in 1:4406) { ofp[i] ~ dpois(mu[i])
   log(mu[i]) <- beta0+beta[1]*hosp[i]+beta[2]*equals(health[i],2)+beta[3]*equals(health[i],3)
                 +beta[4]*numchron[i]+beta[5]*gendermale[i]+beta[6]*school[i]+beta[7]*privins[i]}
                                  beta0 ~ dnorm(0,0.000001)
for (i in 1:7){ beta[i] ~ dnorm(0,0.001)}} ", file="poisson.jag")
INI <- list(list(beta0=0,beta=c(0,0,0,0,0,0,0)),
                          list(beta0=1,beta=c(0.2,0.2,-0.2,0,0,0,0.2)))
M <- jags.model(data=D,inits=INI,n.chains=2,n.adapt=500, file="poisson.jag")
R <- coda.samples(M,c("beta"),n.iter=1000)
summary(R); gelman.diag(R)
# JAGS Neg-Bin
cat("model { for (i in 1:4406) { ofp[i] ~ dnegbin(p[i],theta)
                                  p[i] <- theta/(theta+mu[i])
      log(mu[i]) <- beta0+beta[1]*hosp[i]+beta[2]*equals(health[i],2)+beta[3]*equals(health[i],3)
                 +beta[4]*numchron[i]+beta[5]*gendermale[i]+beta[6]*school[i]+beta[7]*privins[i]}
                                  theta ~ dgamma(1,0.01)
                                  beta0 ~ dnorm(0,0.000001)
for (i in 1:7){ beta[i] ~ dnorm(0,0.001)}} ", file="negbin.jag")
INI <- list(list(beta0=0,beta=c(0,0,0,0,0,0,0),theta=1),
                          list(beta0=1,beta=c(0.2,0.2,-0.2,0,0,0,0.2),theta=2))
M <- jags.model(data=D,inits=INI,n.chains=2,n.adapt=100, file="negbin.jag")
R <- coda.samples(M,c("beta","theta"),n.iter=1000)
summary(R); gelman.diag(R)
```

Questions

Q10. Include commands in the code in section 3.6.5 to calculate the Poisson deviance ScD under the Poisson regression option in JAGS (see section 3.5.4). Is the posterior mean for ScD (a) under 24000 or (b) over 24000.

Q11 Include a command in the code in section 3.6.5 to obtain

standardized residuals under the negative binomial option, namely

$$r_i=(y_i-\mu_i)/V_i^{0.5}$$

where $V_i=V(y_i|X_i)=\mu_i+\mu_i^2/\theta$. Remember that in the code the response is denoted ofp[i].

Which two cases have the highest posterior mean standardized residuals, obtained from $r_i=(y_i-\mu_i)/V_i^{0.5}$? Note it is best when extracting large arrays to first ensure convergence in the main model parameters (e.g. regression coefficients). You will need to calculate the residuals r[i] within the JAGS code, and then use a subsidiary coda.samples command to monitor the large array, and also use jagsresults, as in

```
R1 <- coda.samples(M,c("r"),n.iter=500)
r.R1 <- jagsresults(R1, c("r"))
resid <- r.R1[,1]
```

Note that the posterior mean residuals are contained in r.R1[,1]. Note that the summary(R1) and gelman.diag(R1) commands need not be done, as they are likely to be time-consuming. One can use the command

```
order(resid)
```

to find the largest residuals.

Q12 Modify the prior on θ in the NB regression code to be

$$\theta \sim U(0,10).$$

Run the model for 5000 iterations. Is the posterior mean for θ now (a) over 1.25, or (b) under 1.25.

References (optional background reading for interest, not required)

Bedrick, E., Christensen, R., Johnson, W (1996). A new perspective on priors for generalized linear models. J. Amer Stat Assoc, 91, 1450-1460.

Gilks, W, Wild, P. (1992) Adaptive rejection sampling for Gibbs sampling. Applied Statistics 41, pp 337-348.

Dean C, Lawless J (1989) Tests for Detecting Overdispersion in Poisson Regression Models. J Amer Stat Assoc, 84: 467-472

Deb P, Trivedi PK (1997). Demand for Medical Care by the Elderly: A Finite Mixture Approach. Journal of Applied Econometrics, 12, 313-336.

Kuk A, Li J, Rush A (2014) Variable and threshold selection to control predictive accuracy in logistic regression. J Roy Stat Soc, Ser C
Tabaei B, Herman W. A multivariate logistic regression equation to screen for diabetes: development and validation. Diabetes Care. 2002 25(11):1999-2003.
Zeileis A, Kleiber C, Jackman S (2008) Regression Models for Count Data in R, J Stat Software, 27(8)