Steven Maharaj 695281 Assignment 2, Question 1

Due: Friday 20 September 2019

There are places in this assignment where R code will be required. Therefore set the random seed so assignment is reproducible.

```
set.seed(695281)  #Please change random seed to your student id number.
```

Question One (12 marks)

In generalised linear models, rather than estimating effects from the response data directly, we model through a link function, $\eta(\boldsymbol{\theta})$, and assume $\eta(\boldsymbol{\theta})_i = \mathbf{x}_i'\boldsymbol{\beta}$. The link function can be determined by re-arranging the likelihood of interest into the exponential family format,

$$p(y|\boldsymbol{\theta}) = f(y)g(\boldsymbol{\theta})e^{\eta(\boldsymbol{\theta})'u(y)}. \tag{1}$$

a) Re-arrange the Poisson probability mass function into the exponential family format to determine the canonical link function. The Poisson pmf is

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

Answer:

We have that the Poisson pmf is

$$\begin{aligned} Pr(y|\lambda) &= \frac{\lambda^y e^{-\lambda}}{y!} \\ &= \frac{1}{y!} e^{y\log(\lambda)} e^{-\lambda} \end{aligned}$$

Hence $f(y) = \frac{1}{y!}, u(y) = y, g(\lambda) = e^{-\lambda}$ and the link function

$$\eta(\lambda) = \log(\lambda).$$

To explore some properties of Metropolis sampling, consider the dataset `Warpbreaks.csv`, which is on LMS. This dataset contains information of the number of breaks in a consignment of wool. In addition, Wool type (A or B) and tension level (L, M or H) was recorded.

b) Fit a Poisson regression to the warpbreak data, with Wool type and tension treated as factors using the function glm in R. Report co-efficient estimates and the variance-covariance matrix.

Answer:

```
# read data
WOOL <- read.csv("Warpbreaks.csv")
```

```r
# model poisson regression
mod<-glm(breaks~ ., WOOL, family = poisson(link = "log"))
summary(mod)
```

```
##
## Call:
## glm(formula = breaks ~ ., family = poisson(link = "log"), data = WOOL)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6871  -1.6503  -0.4269   1.1902   4.2616
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.17347    0.05567  57.002  < 2e-16 ***
## woolB       -0.20599    0.05157  -3.994 6.49e-05 ***
## tensionL     0.51849    0.06396   8.107 5.21e-16 ***
## tensionM     0.19717    0.06833   2.885  0.00391 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 297.37  on 53  degrees of freedom
## Residual deviance: 210.39  on 50  degrees of freedom
## AIC: 493.06
##
## Number of Fisher Scoring iterations: 4
```

```r
vcov(mod)
```

```
##               (Intercept)         woolB      tensionL      tensionM
## (Intercept)  0.003099518 -1.193312e-03 -2.564099e-03 -2.564099e-03
## woolB       -0.001193312  2.659585e-03  5.034078e-19  2.454025e-19
## tensionL    -0.002564099  5.034078e-19  4.090810e-03  2.564099e-03
## tensionM    -0.002564099  2.454025e-19  2.564099e-03  4.669354e-03
```

```r
confint(mod,level=0.995)
```

```
## Waiting for profiling to be done...
```

```
##                    0.3 %      99.8 %
## (Intercept)  3.013926429  3.32660462
## woolB       -0.351173215 -0.06152152
## tensionL     0.340192963  0.69951267
## tensionM     0.005811379  0.38973193
```

c) Fit a Bayesian Poisson regression using Metropolis sampling. Assume flat priors for all coefficients. Extract the design matrix $\mathbf{X}$ from the `glm` fitted in a). For the proposal distribution, use a Normal distribution with mean θ^{t-1} and variance-covariance matrix $c^2\hat{\mathbf{\Sigma}}$ where $\mathbf{\Sigma}$ is the variance-covariance matrix from the glm fit. Consider three candidates for c, $1.6/\sqrt{p}, 2.4/\sqrt{p}, 3.2\sqrt{p}$, where p is the number of parameters estimated. Run the Metropolis algorithm for 10,000 iterations, and discard the first 5,000. Report the following:

- Check, using graphs and appropriate statistics, that each chain converges to the same distribution. To do this, you may find installing the R package `coda` helpful.

- The proportion of candidate draws that were accepted.
- The effective sample size for each chain.
- What do you think is the best choice for c. Does this match the results stated in class on efficiency and optimal acceptance rate?