Homework 2

Zach White

1. Agresti 4.5 (30 points)

Consider the following artificial data. When x=0, we have 1 success out of 4 trials. When x=1, we have 2 successes out of 4 trials; when x=2, we have 4 successes out of 4 trials.

Use M_0 to denote an intercept only logistic regression model, and use M_1 to denote the model with an intercept and a linear term in x. Denote the maximized log-likelihood values by L_0 for M_0, L_1 for M_1, and L_s for the saturated model. Consider the data in two formats: (i) ungrouped data, with n_i=1 for $i = 1, 2, ..., 12$, and (ii) grouped data with $n_i = 4$ for \$i=1,2,3.

1a.

Fit M_0 and M_1 for each data format, and report L_0 and L_1 in each case. Note the data format does not affect the values of L_0 and L_1.

Solution

```
# Ungrouped
y = c(1,0,0,0,1,1,0,0,1,1,1,1)
x = c(0,0,0,0,1,1,1,1,2,2,2,2)
Y.ungroup = cbind(y,c(1-y))
int.x = glm(y~x,family = binomial(link = logit))
int.only = glm(y~1, family = binomial(link = logit))
#int.x$fitted.values
#int.only$fitted.values
# Grouped
group.y.succ = c(1,2,4)
group.y.fail = c(3,2,0)
Y.group = cbind(group.y.succ,group.y.fail)
group.x = c(0,1,2)
group.x.mod = glm(cbind(group.y.succ,group.y.fail)~group.x, family = binomial(link = logit))
group.int.only = glm(cbind(group.y.succ,group.y.fail)~1,family = binomial(link = logit))

#logLik(int.x);logLik(int.only)
#logLik(group.x.mod); logLik(group.int.only)

inv.logit <- function(x) { exp(x) / (1+exp(x))}
nloglike <- function(Y,X,a,b) {
  L <- 0
  n = length(X)
  for (i in 1:n){
     L <- L + sum(Y[i,1]*log(inv.logit(a+b*X[i])) +
                Y[i,2]*log(1-inv.logit(a+b*X[i])))
  }
 return(-L)
}
x.un.like = nloglike(Y.ungroup,x,coef(int.x)[1],coef(int.x)[2])
x.group.like = nloglike(Y.group,group.x,coef(group.x.mod)[1],coef(group.x.mod)[2])
```

```r
int.un.like = nloglike(Y.ungroup,x,coef(int.only)[1],0)
int.group.like = nloglike(Y.group,group.x,coef(group.int.only)[1],0)

likelihood.mat = matrix(c(x.un.like,x.group.like,int.un.like,int.group.like),byrow = TRUE,nrow = 2)
colnames(likelihood.mat) = c("Ungrouped","Grouped")
rownames(likelihood.mat) = c("Full","Intercept")
likelihood.mat
```

```
##           Ungrouped  Grouped
## Full       5.514129 5.514129
## Intercept  8.150319 8.150319
```

I created my own function to calculate the log likelihood of the ungrouped and grouped data, and there is no difference between the likelihoods. This can easily be shown using the actual likelihoods of a binomial and bernoulli with the data we are given.

1b.

Show that the deviances for M_0 and M_1 differ for the two data formats. Why is this? (Hint: Consider the number of parameters in the saturated model for each case.)

Solution

```r
int.x$deviance; int.only$deviance
```

```
## [1] 11.02826
```

```
## [1] 16.30064
```

```r
group.x.mod$deviance; group.int.only$deviance
```

```
## [1] 0.9843993
```

```
## [1] 6.25678
```

```r
deviance.mat = matrix(c(int.x$deviance, int.only$deviance,
group.x.mod$deviance, group.int.only$deviance), byrow = FALSE, nrow = 2,dimnames = list(c("Full","Int")
deviance.mat
```

```
##      ungrouped   grouped
## Full  11.02826 0.9843993
## Int   16.30064 6.2567798
```

The deviances will be different because under the grouped data, we only have 3 observations and thus 3 parameters under the grouped data, while under the ungrouped data, we have 12 observations and thus 12 parameters. Thus the deviance will clearly be different because under the grouped data, our models have 1-2 parameters compared to a maximum of 3 in the saturated model, and the ungrouped data have 1-2 parameters compared to a maximum of 12 in the saturated model. We see that as a whole, the deviance is lower for the grouped data than the ungrouped data. And of course, the full model deviance is lower than just the intercept deviance, which is just the global average log-odds for this data.

1c.

Show that the difference between the deviances for M_0 and M_1 is the same regardless of data format. Why is this? This result implies that for testing for the effect of x, the data format does not matter, though it does matter for testing goodness of fit versus the saturated model.

Solution

```
anova.ungroup = anova(int.only,int.x,test = "Chisq")
anova.grouped = anova(group.int.only,group.x.mod,test = "Chisq")
anova.ungroup$Deviance[2]
```

```
## [1] 5.27238
```

```
anova.grouped$Deviance[2]
```

```
## [1] 5.27238
```

We can see that the differences in the deviances between the two models within the same grouping status is the same across groups. This is because we are adding the same linear term across the groups. So we can expect the same linear term to explain the same amount of deviance.

2. Cellular differentiation (20 points)

Using the cellular differentiation data from class, fit a Poisson regression model with IFN as a nominal categorical variable and interaction terms between IFN and TNF (a saturated model). Assess whether the interaction terms provide significant improvement to the model and diagnose any problems encountered in estimation. Using the main effects model, evaluate whether IFN is best treated as a nominal categorical variable or by fitting a linear trend in IFN dose.

Solution

```
celldiff = c(11,18,20,39,22,38,52,69)
TNF = c(10,10,10,10,100,100,100,100)
TNF.cat = as.factor(TNF)
IFN = c(0,4,20,100,0,4,20,100)
IFN.cat = as.factor(IFN)
m.cat.full = glm(celldiff~TNF + IFN.cat + TNF:IFN.cat, family = poisson(link = log))
sum.cat.full = summary(m.cat.full)

m.cat = glm(celldiff~TNF + IFN.cat, family = poisson(link = log))
sum.cat = summary(m.cat)

m.cont = glm(celldiff~TNF + IFN, family = poisson(link=log))
summary(m.cont)
```

```
##
## Call:
## glm(formula = celldiff ~ TNF + IFN, family = poisson(link = log))
##
## Deviance Residuals:
##       1        2        3        4        5        6        7        8
## -1.3492   0.3306   0.2400   0.4394  -2.0585   0.6362   1.9851  -0.6686
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 2.698748   0.133992  20.141  < 2e-16 ***
## TNF         0.008013   0.001444   5.549 2.87e-08 ***
## IFN         0.008135   0.001340   6.070 1.28e-09 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 78.877  on 7  degrees of freedom
## Residual deviance: 11.210  on 5  degrees of freedom
## AIC: 58.829
##
## Number of Fisher Scoring iterations: 4
```

```
m.cont.full = glm(celldiff~TNF + IFN + IFN:TNF, family = poisson(link = log))
summary(m.cont.full)
```

```
##
## Call:
## glm(formula = celldiff ~ TNF + IFN + IFN:TNF, family = poisson(link = log))
##
## Deviance Residuals:
##       1        2        3        4        5        6        7        8
## -1.0621   0.6189   0.4358  -0.0761  -2.2542   0.4396   1.8521  -0.3037
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.609e+00  1.750e-01  14.909  < 2e-16 ***
## TNF          9.268e-03  2.097e-03   4.419 9.91e-06 ***
## IFN          1.000e-02  2.589e-03   3.864 0.000112 ***
## TNF:IFN     -2.660e-05  3.163e-05  -0.841 0.400411
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 78.877  on 7  degrees of freedom
## Residual deviance: 10.504  on 4  degrees of freedom
## AIC: 60.123
##
## Number of Fisher Scoring iterations: 4
```

```
# Categorical vs Continuos
anova(m.cat,m.cont,test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: celldiff ~ TNF + IFN.cat
## Model 2: celldiff ~ TNF + IFN
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         3      1.391
## 2         5     11.210 -2   -9.819 0.007376 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Categorical interactions vs. Categorical
anova(m.cat.full,m.cat, test = "Chisq")
```

```
## Analysis of Deviance Table
##
```

```
## Model 1: celldiff ~ TNF + IFN.cat + TNF:IFN.cat
## Model 2: celldiff ~ TNF + IFN.cat
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         0      0.000
## 2         3      1.391 -3   -1.391   0.7076
```

```
# continuous interactions vs. continuous
anova(m.cont.full, m.cont, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: celldiff ~ TNF + IFN + IFN:TNF
## Model 2: celldiff ~ TNF + IFN
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         4     10.504
## 2         5     11.210 -1 -0.70595   0.4008
```

I treat the other variable TNF as continuous or this problem. When comparing the categorical to continuous, the only significant category of IFN is 100 U/ml. However, when we treat IFN as continuous, the coefficient is significant according to a Wald test. When I think about reconciling these two ideas, it could be that we really start to see the effects of IFN as the dosage gets to be pretty high. In neither of the models are the interactions significant.

Interactions do decrease the deviance, but that is mostly because it uses up all of the degres of freedom. It doesn't seem that adding these interactions add to the model. Normally, this might be a problem, but since we are dealing with grouped poisson count data, it's not a huge deal. This is for two reasons: the degrees of freedom for grouped data is different than ungrouped data and under the poisson framework, the variance and the mean are the same. And so similarly in the way that we cannot estimate variance of a normal distribution with only one data point, we can estimate variance of a poisson if we know the mean because these are shared.

When we are analyzing the basic model, it seems as though categorical representation of IFN is better than the continuous representation. The difference in deviance between these two models is 9.82, which is significant under a χ^2 test with 2 degrees of freedom.

3. Negative binomial distribution (20 points)

3a. Negative binomial as Poisson-Gamma mixture

Consider the negative binomial distribution,

$$f(y \mid p, r) = \binom{r+y-1}{y} p^r (1-p)^y.$$

Suppose we have a Poisson random variable Y where $y \mid \lambda \sim$ Poisson and $\lambda \sim$ Gamma(α, β). Show the negative binomial distribution can be derived as a Poisson-gamma mixture by integrating λ out of the joint likelihood $Pr(y, \lambda) = Pr(y \mid \lambda) Pr(\lambda)$.

Solution

$$
\begin{aligned}
f(y \mid p, r) &= \binom{r+y-1}{y} p^r (1-p)^y \\
p(y \mid \lambda) = \frac{e^{-\lambda}\lambda^y}{y!} \quad \pi(\lambda) &= \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} \exp -\beta\lambda \\
Pr(y, \lambda) &= \frac{e^{-\lambda}\lambda^y}{y!} \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} \exp\{-\beta\lambda\} \\
Pr(y \mid \alpha, \beta) &= \int_\lambda \frac{e^{-\lambda}\lambda^y}{y!} \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} \exp\{-\beta\lambda\} d\lambda \\
&= \frac{\beta^\alpha}{\Gamma(\alpha)y!} \int_\lambda \lambda^{y+\alpha-1} \exp\{-\lambda(\beta+1)\} d\lambda \\
&= \frac{\beta^\alpha}{\Gamma(\alpha)y!} \frac{\Gamma(y+\alpha)}{(\beta+1)^{y+\alpha}} \int_\lambda \frac{(\beta+1)^{y+\alpha}}{\Gamma(y+\alpha)} \lambda^{y+\alpha-1} \exp\{-\lambda(\beta+1)\} d\lambda \\
&= \frac{\beta^\alpha}{(\beta+1)^y(\beta+1)^\alpha} \frac{\Gamma(y+\alpha)}{\Gamma(\alpha)y!} \\
&= \frac{\Gamma(y+\alpha)}{\Gamma(\alpha)y!} (\frac{\beta}{\beta+1})^\alpha (\frac{1}{\beta+1})^y \\
\text{Note that } \frac{\Gamma(y+\alpha)}{\Gamma(\alpha)y!} &= \frac{(y+\alpha-1)(y+\alpha-2)\dots(y+\alpha-y)\Gamma(\alpha)}{\Gamma(\alpha)y!} = \binom{\alpha+y-1}{y} \\
&= \binom{\alpha+y-1}{y} (\frac{\beta}{\beta+1})^\alpha (1-\frac{\beta}{\beta+1})^y \\
\text{So clearly } \alpha = r \text{ and } p &= \frac{\beta}{\beta+1}
\end{aligned}
$$

3b. Exponential family formulation when r known

Show the negative binomial distribution is a member of the exponential family when r is known but not when r is unknown.

Solution

To begin, I show that when r is known, the negative binomial is in the eponential family. Note that the form of the exponential family is as follows:

$$h(x) \exp\{\eta(\theta)T(X) - A(\eta)\}$$

$$
\begin{aligned}
f(y \mid p, r) &= \binom{r+y-1}{y} p^r (1-p)^y \\
&= \binom{r+y-1}{y} \exp\{r \log p + y \log(1-p)\} \\
\text{Let } \theta = log(1-p) \rightarrow p &= 1 - e^\theta \\
&= \binom{r+y-1}{y} \exp\{r \log(1-e^\theta) + y\theta\} \\
&= \binom{r+y-1}{y} \exp\{y\theta \ - (-r \log(1-e^\theta))\}
\end{aligned}
$$

So note $\eta(\theta) = \log(1-p)$, $T(X) = y$, $A(\eta) = -r\log(1-e^{\theta})$, and $h(x) = \binom{r+y-1}{y}$. So when r is known, this is an exponential family. However, when r is not known, this cannot be shown, as follows.

$$\begin{aligned}
f(y \mid p, r) &= \binom{r+y-1}{y} p^r (1-p)^y \\
&= \exp\{\log(\binom{r+y-1}{y} p^r (1-p)^y)\} \\
&= \exp\{\log \binom{r+y-1}{y} + r\log(p) + y\log(1-p)\} \\
&= \exp\{\log(\frac{(r+y-1)!}{(r-1)!y!})\} \exp\{r\log(p) + y\log(1-p)\} \\
&= \frac{(y+r-1)\dots(y+1)}{(r-1)!} \exp\{r\log(p) + y\log(1-p)\}
\end{aligned}$$

We can't reconcile the factorial to get it into an exponential form.

4. Hospital Performance (30 points)

The Centers for Medicare and Medicaid Services (CMS) is required by Congress to evaluate hospital performance. We consider data on outcomes of cardiac surgical procedures in New York State from 2008-2011. The outcome of interest is mortality within 30 days of the procedure, and variables available in the dataset include hospital name, procedure type (the CABG procedure is simpler than valve replacement), number of cases/procedures, number of deaths, and expected mortality rate per 100 cases (this variable adjusts for case mix or perceived difficulty of the procedures related to patient characteristics, e.g. some hospitals may see rich, healthy patients while others see patients who have limited resources to devote to medical care). Assume each line in the dataset represents data contributed by a different physician (physician names are not included). The data are in the file cardiacsurgery.csv.

4a. Model fitting and interpretation

Treating each row of the excel spreadsheet as an observation from a binomial distribution with number of failures equal to the number of deaths and number of trials equal to the number of cases/procedures, fit a GLM exploring the contributions of the hospital, region, procedure, and case mix to the mortality rate. Use analysis of deviance to select the most appropriate model for the data, specify your preferred model clearly in equation form, and provide clear interpretations of all parameter estimates.

Solution

```
cardiac = read.csv("C:/Users/Zachary/Desktop/Fall_2017_Projects/STA_841/STA_841_cat/HW02/cardiacsurgery
cardiac$success = cardiac$Ncases - cardiac$Ndeaths
#cardiac.glm = glm(cbind(success,Ndeaths)~ Hospital+Region+Procedure+ExpectedMortalityRate, data = card
#summary(cardiac.glm)

# Interactions
base.glm = glm(cbind(success,Ndeaths)~ Hospital+Region+Procedure+ExpectedMortalityRate, data = cardiac,
proc.emr.int = glm(cbind(success,Ndeaths)~ Hospital+Region+Procedure+ExpectedMortalityRate + Procedure:
reg.proc.int = glm(cbind(success,Ndeaths)~Hospital+Region+Procedure+ExpectedMortalityRate + Hospital:Pr
anova(base.glm,proc.emr.int, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(success, Ndeaths) ~ Hospital + Region + Procedure + ExpectedMortalityRate
## Model 2: cbind(success, Ndeaths) ~ Hospital + Region + Procedure + ExpectedMortalityRate +
##     Procedure:ExpectedMortalityRate
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       319     371.18
## 2       318     370.95  1  0.23892    0.625
```

```
anova(base.glm,reg.proc.int,test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(success, Ndeaths) ~ Hospital + Region + Procedure + ExpectedMortalityRate
## Model 2: cbind(success, Ndeaths) ~ Hospital + Region + Procedure + ExpectedMortalityRate +
##     Hospital:Procedure
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       319     371.18
## 2       280     346.95 39   24.231   0.9692
```

```
# lmer models
hosp.hier.glm = glmer(cbind(success,Ndeaths) ~ Region+Procedure+ExpectedMortalityRate + (1|Hospital), da
reg.hier.glm = glmer(cbind(success,Ndeaths) ~ Hospital + (1|Region) + Procedure + ExpectedMortalityRate
  data = cardiac, family = binomial(link=logit))

anova(hosp.hier.glm,reg.hier.glm,test = "Chisq")
```

```
## Data: cardiac
## Models:
## hosp.hier.glm: cbind(success, Ndeaths) ~ Region + Procedure + ExpectedMortalityRate +
## hosp.hier.glm:     (1 | Hospital)
## reg.hier.glm: cbind(success, Ndeaths) ~ Hospital + (1 | Region) + Procedure +
## reg.hier.glm:     ExpectedMortalityRate
##                Df     AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## hosp.hier.glm 14  449.35  503.8 -210.68   421.35
## reg.hier.glm  43 1333.40 1500.6 -623.70  1247.40     0     29          1
```

I begin by fitting basic generalize linear models for this binomial data of cardiac surgery success rates. We are interested in the effects of hospital, region, procedure, and epected mortality rate on the success probability of cardiac surgery. After doing this, I test out to see if any reasonable interactions represent significant improvements in the model. However, the interactions don't seem to offer significant improvements.

Something of note though that I can tell is that, it makes sense that since we are interested in ranking hospitals, it makes sense to build a hierarchical model around hospital or possible region. And thus, under this paradigm, I build two hierarchical generalized linear models with random effects on either hospital. Now these aren't immediately comparable to the basic glm's I use previously. However, we can compare them to each other, and in so doing, it seems clear that hospital is more appropriate hierarchy in terms of deviance and also in terms of our model goals.

Knowing that hierarchical models make intuitive sense for our problem, I choose to use a hierarchical Bayesian model with random effects for hospital. The following formula shows this model:

$$y_i|p_i \sim \text{Bin}(n_i, p_i)$$

$$log(\frac{p_i}{1-p_i}) = \gamma + \alpha_{ji} + \beta_{\text{r}} * \text{r}_i + \beta_{\text{p}} * \text{p}_i + \beta_{\text{emr}} * \text{emr}_i$$

where $\beta_{\text{r}},\ \beta_{\text{p}},$ and β_{emr} all have diffuse priors.

$j = 1 \ldots 40$ levels of the hospitals

$i = 1 \ldots n$ total number of observations

The interpretation of these paramaters are as follows:

e^{γ} represents the global average odds of success give procedure is CABG and region is Brox

e^{α_j} represents the multiplicative change of odds of changing hospitals in the region bronx and CABG operation. These repesent the random effects by hospital.

e^{β_r} represents the multiplicative change of odds of changing regions relative to the region of Bronx.

e^{β_p} represents the multiplcative change of odds of switching from a CABG to a "Valve or Valve/CABG"

$e^{\beta_{emr}}$ represents the multiplicative change of odds of a 1 increase of Expected Mortality Rate while holding all other factors constant.

4b. Hospital rankings

If you are tasked with recognizing "high-achieving" and "low-achieving" hospitals for CMS, which hospitals would you select, and why? Provide a detailed statistical justification for your choice.

Solution

```
jags.model = function(){
  for(n in 1:N){
    logit(p[n]) <- alpha[hospital[n]] + beta.r * region[n] + beta.p * procedure[n] + beta.emr * emr[n]
    Y[n] ~ dbin(p[n],success[n] + ndeaths[n])
  }
  alpha[1] <- 0
  for(h in 2:H){
    alpha[h] ~ dnorm(alpha.mu,alpha.phi)
  }

  beta.r ~ dnorm(0,.00001)
  beta.p ~ dnorm(0,.00001)
  beta.emr ~ dnorm(0,.00001)

  alpha.mu ~ dnorm(0,.00001)
  alpha.phi <- pow(alpha.sigma,-2)
  alpha.sigma ~ dunif(0,100)
}

success = cardiac$success
ndeaths = cardiac$Ndeaths
hospital = cardiac$Hospital
procedure = cardiac$Procedure
emr = cardiac$ExpectedMortalityRate
region = cardiac$Region
```

```
data.list = list(success = success,
            ndeaths = ndeaths,
            Y = success,
            hospital = hospital,
            procedure = procedure,
            emr = emr,
            region = region,
            H = length(unique(hospital)),
            N = length(success)
            )

params = c("alpha", "beta.r", "beta.p","beta.emr","alpha.mu","alpha.sigma")

sims = jags(data.list,inits = NULL, parameters.to.save = params,model = jags.model,n.chains = 2,n.iter =
```

```
## module glm loaded
```

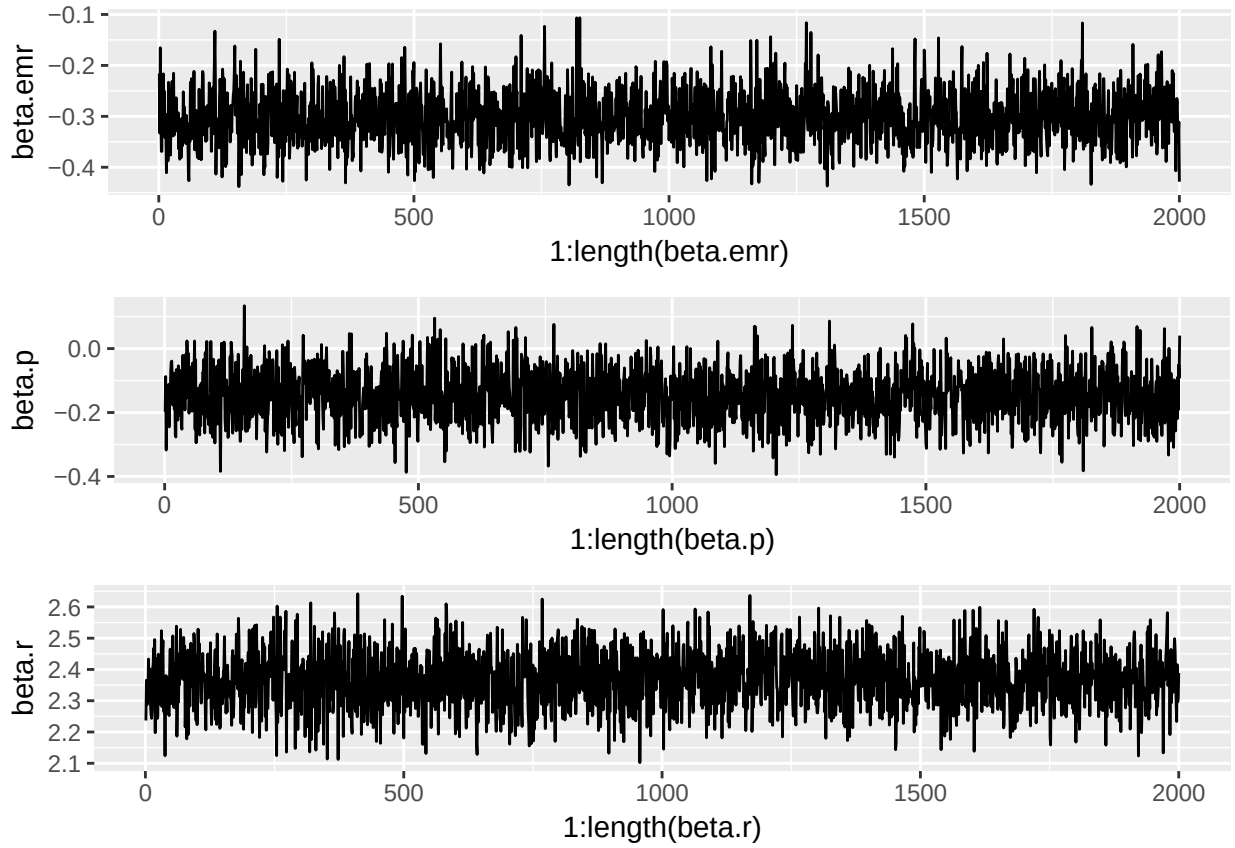
```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 361
##    Unobserved stochastic nodes: 44
##    Total graph size: 3807
##
## Initializing model
```

```
samples = sims$BUGSoutput$sims.matrix

alpha.samps = samples[,1:40]
med = quantile(alpha.samps,.5)
upper = which(colMeans(alpha.samps) > med)
lower = which(colMeans(alpha.samps) <= med)
med.alpha.samps = quantile(colMeans(alpha.samps),.5)
exp.alpha = exp(alpha.samps)
perc.alpha = exp(alpha.samps) / (1+exp(alpha.samps))
colMeans(alpha.samps)
```

```
##     alpha[1]     alpha[2]     alpha[3]     alpha[4]     alpha[5]
##   0.00000000 -21.30351467  -7.16645537  -7.15259849 -19.07889295
##     alpha[6]     alpha[7]     alpha[8]     alpha[9]    alpha[10]
##  -0.19519990   0.07481507 -18.56684291 -11.09222364  -7.05523775
##    alpha[11]    alpha[12]    alpha[13]    alpha[14]    alpha[15]
## -13.56334831  -0.23878508  -4.76218931 -18.86140081 -18.80700917
##    alpha[16]    alpha[17]    alpha[18]    alpha[19]    alpha[20]
##   2.34750184   2.17831354  -7.47011823  -9.30706468 -13.00373827
##    alpha[21]    alpha[22]    alpha[23]    alpha[24]    alpha[25]
##  -4.69063689  -7.33736113  -6.61211392  -7.15767103 -21.58803227
##    alpha[26]    alpha[27]    alpha[28]    alpha[29]    alpha[30]
##  -8.84658368  -2.70199551  -9.47343815  -2.36817063  -7.62070646
##    alpha[31]    alpha[32]    alpha[33]    alpha[34]    alpha[35]
##   0.25935270 -16.67597139 -21.64594491  -2.20609920  -4.71903537
##    alpha[36]    alpha[37]    alpha[38]    alpha[39]    alpha[40]
##  -9.63265427  -2.06675214 -11.43262136 -11.24154935  -9.34800699
```

```
data.frame.samp = as.data.frame(samples)
par(mfrow = c(3,1))
t1 = ggplot(data.frame.samp,aes(y = beta.emr, x = 1:length(beta.emr))) + geom_path()
t2 = ggplot(data.frame.samp,aes(y = beta.p, x = 1:length(beta.p))) + geom_path()
t3 = ggplot(data.frame.samp,aes(y = beta.r, x = 1:length(beta.r))) + geom_path()
grid.arrange( t1,t2,t3, ncol=1)
```



According to this model, if we want to classify high-achieving and low-achieving hopsitals, we basically just need to focus on the random effects model. Although this is ad-hoc, the way I choose to do this is by finding the posterior median across all of the random effects, and then I see which hospitals have their entire 95% credible interval above it that posterior median.

```
cred.int = apply(alpha.samps,2,quantile,c(.025,.975))
lgl.credible = cred.int > med
which(apply(lgl.credible,2,sum) == 2)
```

```
##  alpha[1]  alpha[6]  alpha[7] alpha[12] alpha[13] alpha[16] alpha[17]
##         1         6         7        12        13        16        17
## alpha[21] alpha[23] alpha[27] alpha[29] alpha[31] alpha[34] alpha[35]
##        21        23        27        29        31        34        35
## alpha[37]
##        37
```

Doing it this way, the following indexes for the hospitals that are "high-achieving" are the following: 1, 6, 7, 12, 13, 16, 17, 21, 27, 29, 31, 34, 35, and 37. These represent these hospitals:

```
levels(cardiac$Hospital)[which(apply(lgl.credible,2,sum) == 2)]
```

```
##  [1] "Albany Medical Center"  "Champ.Valley Phys Hosp"
##  [3] "Ellis Hospital"         "M I Bassett Hospital"
##  [5] "Maimonides Medical Ctr" "Montefiore - Moses"
##  [7] "Montefiore - Weiler"    "NY Methodist Hospital"
##  [9] "NYP- Weill Cornell"     "St. Elizabeth Med Ctr"
## [11] "St. Josephs Hospital"   "St. Peters Hospital"
## [13] "Unitd Hlth Svcs-Wilson" "Univ.Hosp-Brooklyn"
## [15] "Univ.Hosp-SUNY Upstate"
```

While the above hospitals can be considered high achieving, the remaining hospitals can be considered low achieving. Technically, we could have divided this into multiple groups because we only have low and high achieving. According to my framework, only 14 of the 40 hospitals are considered high-achieving.

```
levels(cardiac$Hospital)[which(apply(lgl.credible,2,sum) != 2)]
```

```
##  [1] "Arnot Ogden Med Ctr"    "Bellevue Hospital Ctr"
##  [3] "Beth Israel Med Ctr"    "Buffalo General Hosp"
##  [5] "Erie County Med Ctr"    "Good Sam - Suffern"
##  [7] "Lenox Hill Hospital"    "LIJ Medical Center"
##  [9] "Mercy Hospital"         "Millard Fillmore Hosp"
## [11] "Mount Sinai Hospital"   "North Shore Univ Hosp"
## [13] "NY Hospital - Queens"   "NYP- Columbia Presby."
## [15] "NYU Hospitals Center"   "Rochester General Hosp"
## [17] "Southside Hospital"     "St. Francis Hospital"
## [19] "St. Lukes at St. Lukes" "Staten Island UnivHosp"
## [21] "Strong Memorial Hosp"   "Univ.Hosp-Stony Brook"
## [23] "Vassar Bros. Med Ctr"   "Westchester Med Ctr"
## [25] "Winthrop Univ. Hosp"
```