First we install the glmbayes library.

```
library(glmbayes)
```

To understand how the output of the glmb function mirrors that for the glm function, it useful to take a look at the first portion of the example that is provided for the glm function. The data is based on Randomized Controlled Trial data from Dobson (1990). Here is a view of the data.

```
## Dobson (1990) Page 93: Randomized Controlled Trial :
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3,1,9)
treatment <- gl(3,3)
print(d.AD <- data.frame(treatment, outcome, counts))

##   treatment outcome counts
## 1         1       1     18
## 2         1       2     17
## 3         1       3     15
## 4         2       1     20
## 5         2       2     10
## 6         2       3     20
## 7         3       1     25
## 8         3       2     13
## 9         3       3     12
```

The example code for the glm function specifies a Poisson regression model for this data as follows:

```
glm.D93 <- glm(counts ~ outcome + treatment,
              family = poisson())
```

The printed output from the call to glm looks as follows (note there are 5 variables in the model). One of the coefficients represents the intercept, while the others reprent the effect o outcomes and treatment on the counts.

```
print(glm.D93)

##
## Call:  glm(formula = counts ~ outcome + treatment, family = poisson())
##
## Coefficients:
## (Intercept)     outcome2     outcome3   treatment2   treatment3
##   3.045e+00   -4.543e-01   -2.930e-01    1.338e-15    1.421e-15
##
## Degrees of Freedom: 8 Total (i.e. Null);  4 Residual
## Null Deviance:     10.58
## Residual Deviance: 5.129  AIC: 56.76
```

To run a Bayesian version of this model, we first need to add a prior. As the output above had 5 columns, we need a prior mean with 5 components. For now, we use log(mean(counts)) as a prior point estimate for the intercept and 0 as point estimates for the other components.

```
mu<-matrix(0,5)
mu

##      [,1]
## [1,]    0
## [2,]    0
## [3,]    0
## [4,]    0
## [5,]    0

mu[1,1]=log(mean(counts))
mu

##          [,1]
## [1,] 2.813411
## [2,] 0.000000
## [3,] 0.000000
## [4,] 0.000000
## [5,] 0.000000
```

For now, we give all of the components a prior standard deviation of 1 as use it to populate a diagonal prior Variance matrix.

```
mysd<-1
V=((mysd)^2)*diag(5)
V

##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
```

We are now ready to call the glmb function using similar code to that for glm. In addition to the two prior components, we also tell the function to generate 1000 random samples for the analysis (similar to how functions like rnorm would be called). [This part of the code does not seem to currently be getting printed to the *.pdf file.]

```r
n<-1000
glmb.D93<-glmb(counts ~ outcome + treatment,family = poisson(),n=n,mu=mu,Sigma=V)
```

```
## Standardizing the model:
## Starting Envelope Creation:
## Gridtype is :1
## Number of Variables in model are :5
## Number of points in Grid are :243
## Finding Values of Log-posteriors:
## Finding Value of Gradients at Log-posteriors:
## Finished Log-posterior evaluations:
## Finished Envelope Creation:
```

```r
print(glmb.D93)
```

```
##
## Call:  glmb(n = n, formula = counts ~ outcome + treatment, family = poisson(),
##     mu = mu, Sigma = V)
##
## Posterior Mean Coefficients:
## (Intercept)     outcome2     outcome3   treatment2   treatment3
##    3.013728    -0.436761    -0.269464     0.006341    -0.003915
##
## Effective Number of Parameters: 4.626228
## Expected Residual Deviance: 9.82226
## DIC: 56.08067
```

```r
summary(glm.D93)
```

```
##
## Call:
## glm(formula = counts ~ outcome + treatment, family = poisson())
##
## Deviance Residuals:
##        1         2         3         4         5         6         7         8
## -0.67125   0.96272  -0.16965  -0.21999  -0.95552   1.04939   0.84715  -0.09167
##        9
## -0.96656
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.045e+00  1.709e-01  17.815   <2e-16 ***
## outcome2    -4.543e-01  2.022e-01  -2.247   0.0246 *
## outcome3    -2.930e-01  1.927e-01  -1.520   0.1285
```

```
## treatment2   1.338e-15  2.000e-01   0.000   1.0000
## treatment3   1.421e-15  2.000e-01   0.000   1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 10.5814  on 8  degrees of freedom
## Residual deviance:  5.1291  on 4  degrees of freedom
## AIC: 56.761
##
## Number of Fisher Scoring iterations: 4
```

```
summary(glmb.D93)
```

```
## Call
## glmb(n = n, formula = counts ~ outcome + treatment, family = poisson(),
##     mu = mu, Sigma = V)
##
## Expected Deviance Residuals:
##        1        2        3        4        5        6        7        8
## -0.55729  0.99103 -0.16224 -0.13021 -0.95305  1.03238  0.99178 -0.05149
##        9
## -0.94523
##
## Prior and Maximum Likelihood Estimates with Standard Deviations
##
##             Prior Mean   Prior.sd  Max Like. Like.sd
## (Intercept)  2.813e+00  1.000e+00  3.045e+00   0.171
## outcome2     0.000e+00  1.000e+00 -4.543e-01   0.202
## outcome3     0.000e+00  1.000e+00 -2.930e-01   0.193
## treatment2   0.000e+00  1.000e+00  1.338e-15   0.200
## treatment3   0.000e+00  1.000e+00  1.421e-15   0.200
##
## Bayesian Estimates Based on 1000 iid draws
##
##             Post.Mode Post.Mean   Post.Sd MC Error Pr(tail)
## (Intercept)  3.027191  3.013728  0.174809        0   0.1269
## outcome2    -0.428953 -0.436761  0.204607        0   0.0130 *
## outcome3    -0.272569 -0.269464  0.189596        0   0.0779 .
## treatment2   0.004042  0.006341  0.194856        0   0.4695
## treatment3   0.004042 -0.003915  0.186309        0   0.4875
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Distribution Percentiles
##
##                  1.0%      2.5%      5.0%    Median     95.0%     97.5% 99.0%
## (Intercept)  2.617811  2.675934  2.726062  3.014661  3.284156  3.341939 3.387
## outcome2    -0.936338 -0.863943 -0.794507 -0.436180 -0.093240 -0.038541 0.015
## outcome3    -0.679785 -0.620496 -0.575448 -0.272055  0.037269  0.116785 0.187
## treatment2  -0.442861 -0.374234 -0.326694  0.019567  0.305070  0.343982 0.439
## treatment3  -0.409679 -0.366995 -0.313983 -0.007799  0.295986  0.352706 0.410
##
## Effective Number of Parameters: 4.626228
## Expected Residual Deviance: 9.82226
## DIC: 56.08067
##
## Mean Likelihood Subgradient Candidates Per iid sample: 1.815
```