

R package glmm

Christina Knudson

May 22, 2014

1 Bacteria example

The MASS package contains the command `glmmPQL` and the bacteria data-set. The manual describes the data set as follows: “Tests of the presence of the bacteria *H. influenzae* in children with otitis media in the Northern Territory of Australia.”

The data were fit using `glmmPQL`, `glmm` with $m = 10^3$, `glmm` with $m = 10^4$, `glmm` with $m = 10^5$. (The data did need to be reformatted, which took only a couple minutes). The parameter estimates are summarized in the following table. More model details can be seen in the output that follows the table.

	Intercept	trtdrug	trtdrug+	I(week > 2) TRUE	ν
<code>glmmPQL</code>	3.41	-1.25	-.75	-1.61	1.99
<code>glmm</code> $m = 10^3$	3.02	-1.20	-.89	-1.44	.81
<code>glmm</code> $m = 10^4$	3.65	-1.36	-.92	-1.66	2.02
<code>glmm</code> $m = 10^5$	3.49	-1.65	-.90	-1.46	.90

It’s safe to say that the bacteria `glmm` results with a paltry Monte Carlo sample size of $m = 10^3$ are not reliable. We conclude this because the estimates change quite a bit when $m = 10^4$. The results using $m = 10^4$ are very similar to the `glmmPQL` results.

The `glmmPQL` results:

```
> bac.pql<-glmmPQL(y ~ trt + I(week > 2), random = ~ 1 | ID,
+                   family = binomial, data = bacteria)
> summary(bac.pql)
```

Linear mixed-effects model fit by maximum likelihood

Data: bacteria

AIC BIC logLik

NA NA NA

Random effects:

Formula: ~1 | ID

(Intercept) Residual

StdDev: 1.410637 0.7800511

Variance function:

```

Structure: fixed weights
Formula: ~invwt
Fixed effects: y ~ trt + I(week > 2)

```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	3.412014	0.5185033	169	6.580506	0.0000
trtdrug	-1.247355	0.6440635	47	-1.936696	0.0588
trtdrug+	-0.754327	0.6453978	47	-1.168779	0.2484
I(week > 2)TRUE	-1.607257	0.3583379	169	-4.485311	0.0000

```

Correlation:
      (Intr) trtdrg trtdr+
trtdrug      -0.598
trtdrug+     -0.571  0.460
I(week > 2)TRUE -0.537  0.047 -0.001

```

```

Standardized Within-Group Residuals:
      Min      Q1      Med      Q3      Max
-5.1985361  0.1572336  0.3513075  0.4949482  1.7448845

```

```

Number of Observations: 220
Number of Groups: 50

```

The bacteria glmm results with a Monte Carlo sample size of $m = 10^3$:

```

> set.seed(1234)
> bac.glmm1<-glmm(y2~trt+I(week > 2),list(~0+ID),
family=bernoulli.glmm, data=bacteria, m=10^3, varcomps.names=c("ID"))

> summary(bac.glmm1)

```

```

Call:
glmm(fixed = y2 ~ trt + I(week > 2), random = list(~0 + ID), varcomps.names = c("ID"),
data = bacteria, family.glmm = bernoulli.glmm, m = 10^3)

```

```

Fixed Effects:

```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.0199	0.4667	6.471	9.75e-11 ***
trtdrug	-1.1989	0.4473	-2.680	0.007357 **
trtdrug+	-0.8883	0.4742	-1.873	0.061021 .
I(week > 2)TRUE	-1.4407	0.4258	-3.384	0.000716 ***

```

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

Variance Components for Random Effects (P-values are one-tailed):

```

	Estimate	Std. Error	z value	Pr(> z)/2
ID	0.8175	0.1635	4.998	2.89e-07 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

The results from fitting the bacteria dataset with glmm and $m = 10^4$:

```
> set.seed(1234)
> bac.glmm2<-glmm(y2~trt+I(week > 2),list(~0+ID), family=bernoulli.glmm, data=bacteria, m=10^4, varcomps=
> summary(bac.glmm2)
```

Call:

```
glmm(fixed = y2 ~ trt + I(week > 2), random = list(~0 + ID),
     varcomps.names = c("ID"), data = bacteria, family.glmm = bernoulli.glmm,
     m = 10^4)
```

Fixed Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.6514	0.6327	5.771	7.9e-09 ***
trtdrug	-1.3645	0.7461	-1.829	0.067427 .
trtdrug+	-0.9186	0.7619	-1.206	0.227915
I(week > 2)TRUE	-1.6660	0.4645	-3.587	0.000334 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Variance Components for Random Effects (P-values are one-tailed):

	Estimate	Std. Error	z value	Pr(> z)/2
ID	2.0244	0.7515	2.694	0.00353 **

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Results of glmm with $m = 10^5$:

```
> set.seed(1234)
> bac.glmm3<-glmm(y2~trt+I(week > 2),list(~0+ID), family=bernoulli.glmm, data=bacteria, m=10^5, varcomps=
> summary(bac.glmm3)
```

Call:

```
glmm(fixed = y2 ~ trt + I(week > 2), random = list(~0 + ID),
     varcomps.names = c("ID"), data = bacteria, family.glmm = bernoulli.glmm,
```

```
m = 10^5)
```

Fixed Effects:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	3.4888	0.4861	7.178	7.08e-13	***
trtdrug	-1.6526	0.5232	-3.159	0.001586	**
trtdrug+	-0.8968	0.5160	-1.738	0.082196	.
I(week > 2)TRUE	-1.4614	0.4334	-3.372	0.000746	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Variance Components for Random Effects (P-values are one-tailed):

	Estimate	Std. Error	z value	Pr(> z)/2	
ID	0.8999	0.2471	3.641	0.000136	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

2 Herd CBPP Example

The cbpp dataset is located in the lme4 package. The lme4 package describes it thusly: “Contagious bovine pleuropneumonia (CBPP) is a major disease of cattle in Africa, caused by a mycoplasma. This dataset describes the serological incidence of CBPP in zebu cattle during a follow-up survey implemented in 15 commercial herds located in the Boji district of Ethiopia. The goal of the survey was to study the within-herd spread of CBPP in newly infected herds. Blood samples were quarterly collected from all animals of these herds to determine their CBPP status. These data were used to compute the serological incidence of CBPP (new cases occurring during a given time period). Some data are missing (lost to follow-up).”

First, I fit the data using glmer in lme4. Then I fit the data using glmm with $m = 10^4$. This took 16.75 minutes on my netbook. The point estimates and the standard errors were very similar between the two methods of model-fitting. (The data did need to be reformatted, which took only a couple minutes)

First, the results from glmer:

```
> summary(gm1)
Generalized linear mixed model fit by maximum likelihood (Laplace
Approximation) [glmerMod]
Family: binomial ( logit )
Formula: cbind(incidence, size - incidence) ~ period + (1 | herd)
Data: cbpp

            AIC      BIC   logLik deviance df.resid
194.1      204.2    -92.0    184.1      51
```

Scaled residuals:

Min	1Q	Median	3Q	Max
-2.3816	-0.7889	-0.2026	0.5142	2.8791

Random effects:

Groups Name	Variance	Std.Dev.
herd (Intercept)	0.4123	0.6421

Number of obs: 56, groups: herd, 15

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-1.3983	0.2312	-6.048	1.47e-09	***
period2	-0.9919	0.3032	-3.272	0.001068	**
period3	-1.1282	0.3228	-3.495	0.000474	***
period4	-1.5797	0.4220	-3.743	0.000182	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

	(Intr)	perid2	perid3
period2	-0.363		
period3	-0.340	0.280	
period4	-0.260	0.213	0.198

Next, the results using glmm with $m = 10^4$:

```
> summary(herd.glmm1)
```

Call:

```
glmm(fixed = Y ~ period, random = list(~0 + herd), varcomps.names = c("herd"),  
     data = herddat, family.glmm = bernoulli.glmm, m = 10^4)
```

Fixed Effects:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-1.4166	0.2410	-5.879	4.14e-09	***
period2	-0.9921	0.3078	-3.223	0.001271	**
period3	-1.1289	0.3277	-3.445	0.000571	***
period4	-1.5789	0.4293	-3.678	0.000235	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Variance Components for Random Effects (P-values are one-tailed):

	Estimate	Std. Error	z value	Pr(> z)/2
herd	0.4354	0.2488	1.75	0.0401 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

3 Salamander

```
> set.seed(1234)
```

```
> sal.glmm3<-glmm(Mate~Cross,random=list(~0+Female,~0+Male),varcomps.names=c("F","M"),data=salamander,fa
```

```
> summary(sal.glmm3)
```

Call:

```
glmm(fixed = Mate ~ Cross, random = list(~0 + Female, ~0 + Male),  
     varcomps.names = c("F", "M"), data = salamander, family.glmm = bernoulli.glmm,  
     m = 10^4)
```

Fixed Effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.1123	0.2593	4.290	1.79e-05 ***
CrossRW	-0.5545	0.3558	-1.558	0.119
CrossWR	-3.1701	0.4029	-7.867	3.62e-15 ***
CrossWW	-0.2122	0.3681	-0.577	0.564

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Variance Components for Random Effects (P-values are one-tailed):

	Estimate	Std. Error	z value	Pr(> z)/2
F	1.2087	0.2534	4.769	9.24e-07 ***
M	0.9485	0.1809	5.244	7.86e-08 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Here are the results with $m = 10^4$ but I forgot to set the seed.

```
> summary(sal.glmm2)
```

Call:

```
glmm(fixed = Mate ~ Cross, random = list(~0 + Female, ~0 + Male),  
     varcomps.names = c("F", "M"), data = salamander, family.glmm = bernoulli.glmm,  
     m = 10^4)
```

Fixed Effects:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	0.93879	0.26963	3.482	0.000498	***
CrossRW	-0.76157	0.37376	-2.038	0.041593	*
CrossWR	-2.72491	0.40736	-6.689	2.24e-11	***
CrossWW	0.02256	0.37902	0.060	0.952541	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Variance Components for Random Effects (P-values are one-tailed):

	Estimate	Std. Error	z value	Pr(> z)/2	
F	1.5991	0.2965	5.394	3.45e-08	***
M	0.9908	0.1885	5.255	7.41e-08	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

4 Reformatting the Datasets

4.1 Bacteria Reformatting

The issue with the bacteria dataset is the response was y/n rather than 1/0. I created a new response that changed the y to 1 and the n to 0.

```
bacteria$y2<-as.integer(bacteria$y)-1
```

4.2 CBPP Reformatting

The cbpp dataset was created for binomial but my package was written for Bernoulli responses. In other words, my package needs a row for each success or failure. I did this in the following way:

```
cbpp$nonincidence<-cbpp$size-cbpp$incidence #number of "failures"
herddat<-matrix(data=NA,nrow=842,ncol=3)
colnames(herddat)<-c("Y","period","herd")
rowid<-1
for(i in 1:nrow(cbpp)){
  #make a row for each one of the incidences
  ntimes<-cbpp[i,2]
  if(ntimes>0){
    for(j in 1:ntimes){
      herddat[rowid,]<-c(1,cbpp[i,4],cbpp[i,1])
      rowid<-rowid+1
    }
  }
}
```

```

}

#make a row for each of the nonincidences
ntimes<-cbpp[i,5]
if(ntimes>0){
  for(j in 1:ntimes){
    herddat[rowid,]<-c(0,cbpp[i,4],cbpp[i,1])
    rowid<-rowid+1
  }
}
}

herddat<-as.data.frame(herddat)
herddat$herd<-as.factor(herddat$herd)
herddat$period<-as.factor(herddat$period)

```