

Generalized linear mixed models

Peter Hoff
Duke STA 610

Non-normal data

Assumption so far:

Within-group heterogeneity is well-represented by a normal distribution.

Reality:

Not a good assumption in many applications.

Examples of data that are generally not normally distributed include

- income, hospital costs or other monetary data (often highly skewed);
- time to an event;
- number of children or other count data;
- binary indicator variables.

Example: Police stop data

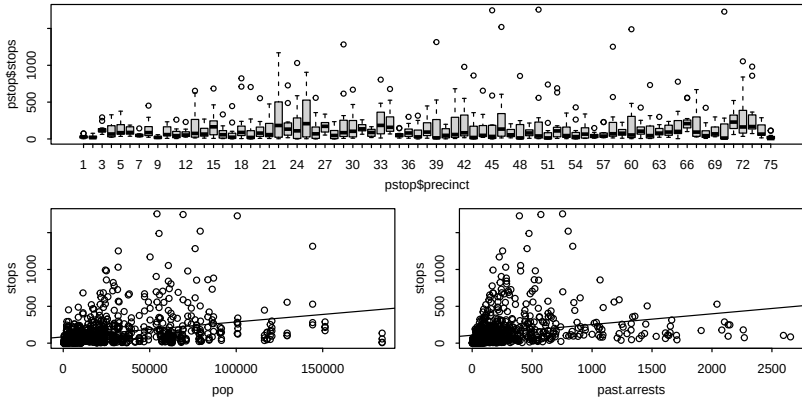
A 1999 study of the NYC police department gathered data on police searches in 75 city precincts, including

- stops: number of stops in a given precinct;
- pop: the population of the precinct;
- past.arrests: number of arrests in the precinct in a previous year.
- eth, crime: stops broken down by ethnicity and crime type.

```
pstop[1:15,]
```

##	stops	pop	past.arrests	precinct	eth	crime
## 1	75	1720	191	1	1	1
## 2	36	1720	57	1	1	2
## 3	74	1720	599	1	1	3
## 4	17	1720	133	1	1	4
## 5	37	1368	62	1	2	1
## 6	39	1368	27	1	2	2
## 7	23	1368	149	1	2	3
## 8	3	1368	57	1	2	4
## 9	26	23854	135	1	3	1
## 10	32	23854	16	1	3	2
## 11	10	23854	107	1	3	3
## 12	13	23854	123	1	3	4
## 13	73	2596	227	2	1	1
## 14	37	2596	56	2	1	2
## 15	9	2596	246	2	1	3

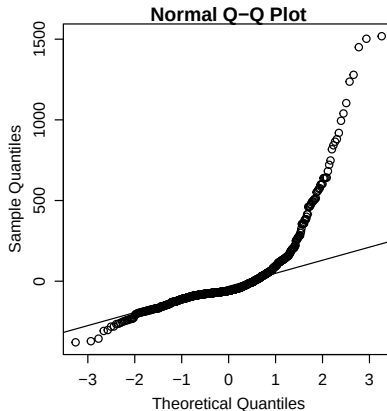
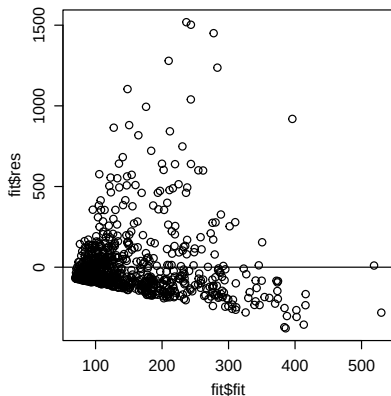
Example: Police stop data



Example: Police stop data

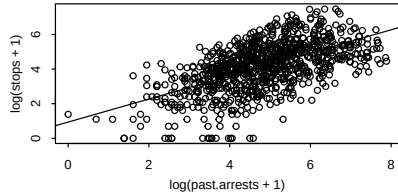
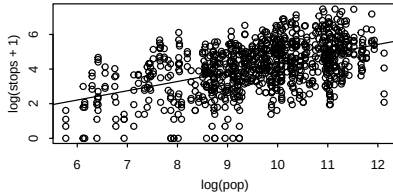
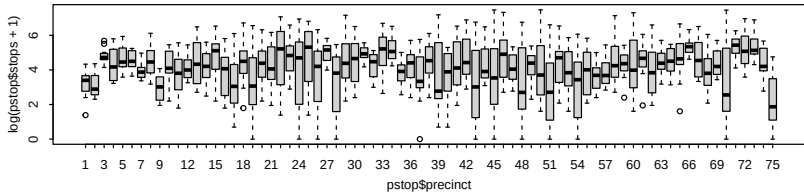
```
fit<-lm(stops~pop+past.arrests,data=pstop)

mpar(mfrow=c(1,2))
plot(fit$res~fit$fit) ; abline(h=0)
qqnorm(fit$res) ; qqline(fit$res)
```



Example: Police stop data

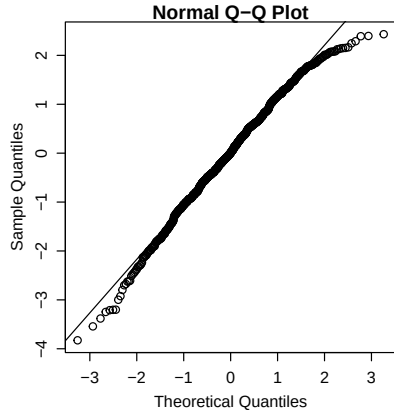
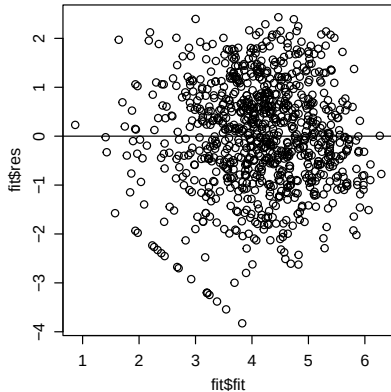
```
mpar()  
layout(matrix(c(1,2,1,3),2,2) )  
boxplot(log(pstop$stops+1)~pstop$precinct)  
plot(log(stops+1)~log(pop),data=pstop) ; abline(lm(log(stops+1)~log(pop),data=pstop))  
plot(log(stops+1)~log(past.arrests+1),data=pstop) ; abline(lm(log(stops+1)~log(past.arrests+1),data=pstop))
```



Example: Police stop data

```
fit<-lm(log(stops+1)~log(pop)+log(past.arrests+1),data=pstop)

mpar(mfrow=c(1,2))
plot(fit$res~fit$fit) ; abline(h=0)
qqnorm(fit$res) ; qqline(fit$res)
```



Example: Grouseticks

Some variables:

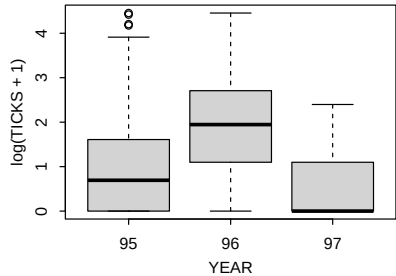
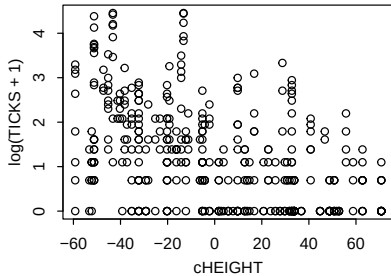
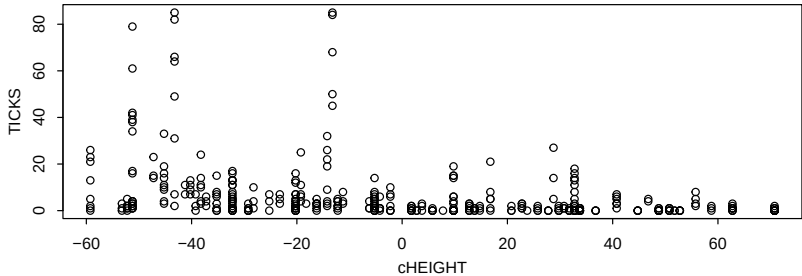
- TICKS: number of ticks on a chicks head
- BROOD: brood number
- cHEIGHT: height above sea level (centered)
- YEAR: year of study

```
grouseticks[1:15,]
```

##	INDEX	TICKS	BROOD	HEIGHT	YEAR	LOCATION	cHEIGHT
## 1	1	0	501	465	95	32	2.759305
## 2	2	0	501	465	95	32	2.759305
## 3	3	0	502	472	95	36	9.759305
## 4	4	0	503	475	95	37	12.759305
## 5	5	0	503	475	95	37	12.759305
## 6	6	3	503	475	95	37	12.759305
## 7	7	2	503	475	95	37	12.759305
## 8	8	0	504	488	95	44	25.759305
## 9	9	0	504	488	95	44	25.759305
## 10	10	2	504	488	95	44	25.759305
## 11	11	0	505	492	95	47	29.759305
## 12	12	0	505	492	95	47	29.759305
## 13	13	0	505	492	95	47	29.759305
## 14	14	0	506	490	95	45	27.759305
## 15	15	0	506	490	95	45	27.759305

Almost 20% of broods have a zero count for all chicks.

Example: Grouseticks

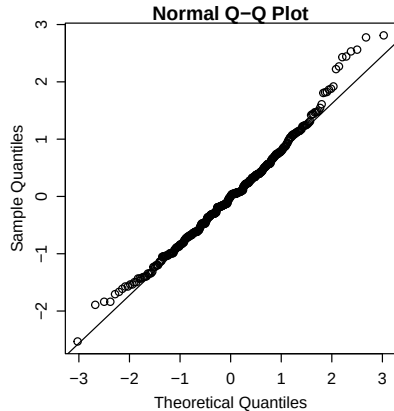
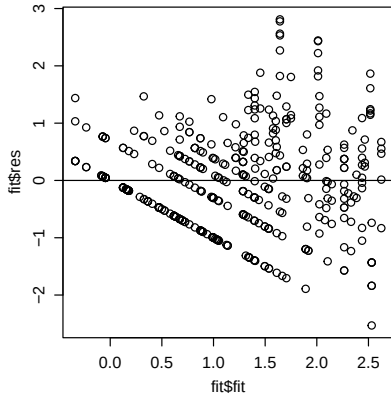


Example: Grouseticks

```
fit<-lm( log(TICKS+1)~ cHEIGHT + as.factor(YEAR) ,data=grouseticks)
```

```
summary(fit)$coef
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	1.06715919	0.07961023	13.404799	4.356064e-24
## cHEIGHT	-0.01331815	0.00120441	-11.057815	5.646538e-25
## as.factor(YEAR)96	0.76833413	0.10607991	7.242975	2.280915e-12
## as.factor(YEAR)97	-0.46374571	0.10926018	-4.244417	2.728417e-05



Transformably normal

Many outcomes can be transformed so that distributional assumptions are almost met:

- mean-variance relationships can be stabilized;
- residuals have a distribution close to normal.

However, sometimes such transformations are not feasible or desirable:

- For some discrete data it is difficult or impossible to find an appropriate transformation (eg., binary data).
- Even if a transformation could be found, there are advantages to analyzing the data on its original scale.

In such cases we will modify our regression models to account for non-normality.

Logistic regression

Logistic regression:

$$\Pr(y_i = 1) = \theta_i = \frac{e^{\beta^T \mathbf{x}_i}}{1 + e^{\beta^T \mathbf{x}_i}}$$
$$\log \frac{\theta_i}{1 - \theta_i} = \beta^T \mathbf{x}_i$$

Data types for logistic regression

Binary data

y	x1	x2
0	2.3	3.2
0	2.3	2.1
1	2.3	5.4
0	4.1	1.6
1	4.1	3.2
1	4.1	1.2
.	.	.
.	.	.

```
fit <- glm( y ~ x1 + x2 , family=binomial )
```

Data types for logistic regression

Binomial data

y	n	x1	x2
4	10	2.3	3.2
2	12	2.3	2.1
5	8	2.3	5.4
3	10	4.1	1.6
6	16	4.1	3.2
8	9	4.1	1.2
.	.	.	.
.	.	.	.

Here, the model is

$$y_i \sim \text{binomial}(\theta_i, n_i)$$

```
fit <- glm( cbind(y,n-y) ~ x1 + x2 , family=binomial )
```

Example: Social network analysis

Network and relational data: Data measured on pairs (dyads) of units.

Friendship study: What characteristics of people lead to friendship ties?

Data:

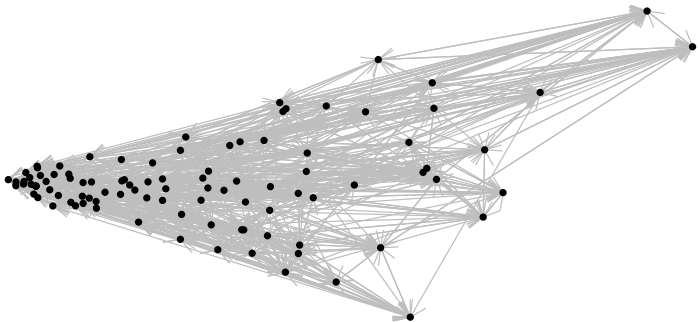
friendship: binary indicator of (directed) friendship

gpa: GPA

smoke: smoking score

grade: year in school

Example: Social network analysis



Example: Social network analysis

```
X[1:15,]
```

```
##      rgpa rsmoke cgpa csmoke igrade  igpa ismoke
## [1,]  0.71   0.62 0.41  -0.07      1  0.29 -0.04
## [2,]  0.85   0.98 0.41  -0.07      1  0.35 -0.07
## [3,]  0.86  -0.12 0.41  -0.07      0  0.35  0.01
## [4,]  0.41   0.64 0.41  -0.07      0  0.17 -0.05
## [5,]  0.66   0.63 0.41  -0.07      0  0.27 -0.04
## [6,] -0.54   0.87 0.41  -0.07      0 -0.22 -0.06
## [7,]  0.69   0.66 0.41  -0.07      0  0.28 -0.05
## [8,] -0.29  -0.64 0.41  -0.07      0 -0.12  0.05
## [9,] -0.51  -0.68 0.41  -0.07      1 -0.21  0.05
## [10,]  0.41   0.64 0.41  -0.07      0  0.17 -0.05
## [11,]  2.17  -1.04 0.41  -0.07      0  0.89  0.07
## [12,] -0.28   0.00 0.41  -0.07      0 -0.11  0.00
## [13,]  0.69  -0.61 0.41  -0.07      1  0.28  0.04
## [14,]  1.52  -0.60 0.41  -0.07      0  0.62  0.04
## [15,] -0.30   1.51 0.41  -0.07      0 -0.12 -0.11
```

```
y[1:15]
```

```
## [1] 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0
```

Logistic regression

```
fit0<-glm( y ~ X, family=binomial)
```

```
summary(fit0)$coef
```

##		Estimate	Std. Error	z value	Pr(> z)
##	(Intercept)	-3.3963777162	0.06823249	-49.7765466	0.000000e+00
##	Xrgpa	0.2654852784	0.05086973	5.2189240	1.799656e-07
##	Xrsmoke	0.3108504175	0.05915817	5.2545649	1.483748e-07
##	Xcgpa	0.2752593480	0.05035810	5.4660390	4.602032e-08
##	Xcsmoke	0.2506464322	0.05986543	4.1868306	2.828768e-05
##	Xigrade	1.5465497415	0.08209840	18.8377564	3.703344e-79
##	Xigpa	-0.0007701432	0.04687819	-0.0164286	9.868925e-01
##	Xismoke	0.1119389182	0.07453943	1.5017410	1.331640e-01

Do you think the observations are independent?

Poisson model

A natural model for count data is the *Poisson model*: $Y \sim \text{Pois}(\theta)$ if

$$\Pr(Y = y|\theta) = \text{dpois}(y, \theta) = \theta^y e^{-\theta} / y!.$$

For example, if $\theta = 2.1$ (the 2006 U.S. fertility rate),

$$\begin{array}{lll} \Pr(Y = 0|\theta = 2.1) = & (2.1)^0 e^{-2.1} / (0!) = & .12 \\ \Pr(Y = 1|\theta = 2.1) = & (2.1)^1 e^{-2.1} / (1!) = & .26 \\ \Pr(Y = 2|\theta = 2.1) = & (2.1)^2 e^{-2.1} / (2!) = & .27 \\ \Pr(Y = 3|\theta = 2.1) = & (2.1)^3 e^{-2.1} / (3!) = & .19 \\ & \vdots & \vdots \end{array}$$

Poisson regression

In Poisson regression, we relate the mean θ to explanatory variables x

One possibility would be to just write $\theta(x)$ as a linear regression:

$$\theta(x) = \beta_0 + \beta_1 x.$$

However, this allows for negative values of $\theta(x)$, which doesn't make sense:

$$Y \in \{0, 1, 2, \dots\} \Rightarrow E[Y|x] = \theta(x) \geq 0 \text{ for all } x$$

Multiplicative mean model

One way to impose this constraint is to write $\theta(x)$ in terms of multiplicative effects, via the exponential function:

$$E[Y|x] = \theta(x) = \exp(\beta_0 + \beta_1 x) = e^{\beta_0 + \beta_1 x}$$

Here, β_1 represents the *multiplicative effect* of x on Y :

$$\begin{aligned} E[Y|x+1]/E[Y|x] &= \frac{e^{\beta_0 + \beta_1(x+1)}}{e^{\beta_0 + \beta_1 x}} \\ &= e^{\beta_1} \end{aligned}$$

So an increase of x by one results in the mean being e^{β_1} times as large.

Eg., $\beta_1 = \log 2 = 0.693 \Rightarrow$ a unit increase in x leads to a doubling of $E[Y]$.

Poisson regression with a log-link

Suppose $\{Y_i, \mathbf{x}_i\}$ are independently sampled from a population.

The *Poisson regression model with a log-link* is

$$Y_i | \mathbf{x}_i \sim \text{Poisson}(\exp[\boldsymbol{\beta}^T \mathbf{x}_i]).$$

Log-link: the function linking the regression to the expectation of Y is log:

$$E[Y|\mathbf{x}] = \exp(\boldsymbol{\beta}^T \mathbf{x}) \Leftrightarrow \log E[Y|\mathbf{x}] = \boldsymbol{\beta}^T \mathbf{x}$$

MLE: The log-likelihood, as a function of $\boldsymbol{\beta}$, is given by

$$\begin{aligned} l(\boldsymbol{\beta} : \mathbf{y}, \mathbf{X}) &= \log \Pr(\mathbf{y} | \mathbf{X}, \boldsymbol{\beta}) \\ &= \log \prod_{i=1}^n p(y_i | \mathbf{x}_i, \boldsymbol{\beta}) \\ &= \sum_{i=1}^n \log \text{dpois}(y_i, \exp(\boldsymbol{\beta}^T \mathbf{x}_i)) \\ &= \text{sum}(\log(\text{dpois}(\mathbf{y}, \exp(\mathbf{X} \%* \% \boldsymbol{\beta})))) \end{aligned}$$

Example: Police stop data

```
fit.pstop<-glm( stops ~ log(pop) + log(past.arrests+1) + as.factor(eth) ,  
               family=poisson, data=pstop[pstop$crime==1,])
```

```
summary(fit.pstop)$coef
```

##	Estimate	Std. Error	z value	Pr(> z)
## (Intercept)	1.26242170	0.051554721	24.48702	2.030854e-132
## log(pop)	0.07275764	0.009306141	7.81824	5.356706e-15
## log(past.arrests + 1)	0.59150963	0.012420654	47.62307	0.000000e+00
## as.factor(eth)2	-0.32832680	0.014383023	-22.82739	2.451891e-115
## as.factor(eth)3	-1.00964834	0.027073122	-37.29338	2.100727e-304

```
fit.tpstop<-lm(log(stops+1) ~ log(pop) + log(past.arrests+1) + as.factor(eth) ,  
               data=pstop[pstop$crime==1,])
```

```
summary(fit.tpstop)$coef
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	-0.7092861	0.32676005	-2.170664	3.102700e-02
## log(pop)	0.1881834	0.05655829	3.327248	1.027900e-03
## log(past.arrests + 1)	0.7346374	0.07750595	9.478465	4.320869e-18
## as.factor(eth)2	-0.4813409	0.12046354	-3.995739	8.798892e-05
## as.factor(eth)3	-1.3316453	0.15791405	-8.432723	4.472171e-15

Example: Grouse tick data

```
fit.gtick<-glm( TICKS ~ cHEIGHT + as.factor(YEAR) , family=poisson, data=grouseticks)
```

```
summary(fit.gtick)$coef
```

##	Estimate	Std. Error	z value	Pr(> z)
## (Intercept)	1.61599798	0.0401455805	40.253447	0.000000e+00
## cHEIGHT	-0.02145184	0.0007103969	-30.196982	2.594875e-200
## as.factor(YEAR)96	0.40964577	0.0453477934	9.033422	1.663851e-19
## as.factor(YEAR)97	-1.68514105	0.0898007151	-18.765341	1.450635e-78

```
fit.tgtick<-lm(log(TICKS+1) ~ cHEIGHT + as.factor(YEAR) , data=grouseticks)
```

```
summary(fit.tgtick)$coef
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	1.06715919	0.07961023	13.404799	4.356064e-34
## cHEIGHT	-0.01331815	0.00120441	-11.057815	5.646538e-25
## as.factor(YEAR)96	0.76833413	0.10607991	7.242975	2.280915e-12
## as.factor(YEAR)97	-0.46374571	0.10926018	-4.244417	2.728417e-05

Generalized linear models

Generalized linear model (glm): A model in which the mean $E[Y]$ of an outcome is related to some specified function of a linear predictor $\beta^T \mathbf{x}$ via a *link function* g :

$$\begin{aligned} g(E[Y|\mathbf{x}]) &= \beta^T \mathbf{x} \\ \theta = E[Y|\mathbf{x}] &= g^{-1}(\beta^T \mathbf{x}) \\ Y &\sim f(y|\theta, \gamma) \end{aligned}$$

Generalized linear models

Normal, Poisson and logistic regression are all GLMs:

The normal regression model: $Y \sim \text{normal}(\beta^T \mathbf{x}, \sigma^2)$

- $\theta = \beta^T \mathbf{x}$, so g is the identity link
- $\gamma = \sigma^2$
- $f(y|\theta, \gamma) = \text{dnorm}(y, \theta, \gamma)$

The Poisson regression model: $Y \sim \text{Poisson}(\exp[\beta^T \mathbf{x}])$

- $\log \theta = \beta^T \mathbf{x}$, so g is the log link
- γ is not present
- $f(y|\theta) = \text{dpois}(y, \theta)$

The logistic regression model: $Y \sim \text{binomial}(n, \frac{\exp[\beta^T \mathbf{x}]}{1 + \exp[\beta^T \mathbf{x}]})$

- when $n = 1$, $E[Y|x] = \Pr(Y = 1|x) = \frac{\exp[\beta^T \mathbf{x}]}{1 + \exp[\beta^T \mathbf{x}]}$
- $\log(\theta/[1 - \theta]) = \beta^T \mathbf{x}$, so g is the logit link
- γ is not present
- $f(y|\theta) = \text{dbinom}(y, n, \theta)$

GLMMs

Recall the general form of a LME:

$$\begin{aligned}y_{i,j} &\sim N(\theta_{i,j}, \sigma^2) \\ \theta_{i,j} &= \boldsymbol{\beta}^T \mathbf{x}_{i,j} + \mathbf{b}_j^T \mathbf{z}_{i,j} \\ \mathbf{b}_1, \dots, \mathbf{b}_m &\sim \text{i.i.d. mvn}(\mathbf{0}, \boldsymbol{\Psi})\end{aligned}$$

A *generalized linear mixed model* is the same model, but with the normal distribution replaced with a glm:

$$\begin{aligned}y_{i,j} &\sim f(y|\theta_{i,j}, \gamma) \\ \theta_{i,j} &= \boldsymbol{\beta}^T \mathbf{x}_{i,j} + \mathbf{b}_j^T \mathbf{z}_{i,j} \\ \mathbf{b}_1, \dots, \mathbf{b}_m &\sim \text{i.i.d. mvn}(\mathbf{0}, \boldsymbol{\Psi})\end{aligned}$$

In this model,

- the $\text{mvn}(\mathbf{0}, \boldsymbol{\Psi})$ distribution represents across-group heterogeneity
- $f(y|\theta, \gamma)$ represents within-group heterogeneity.

The fixed (non-group specific) parameters to estimate include $\{\boldsymbol{\beta}, \boldsymbol{\Psi}, \gamma\}$.

Estimation

LMMs: Estimation for LMMs is facilitated by the following fact:

$$\left. \begin{array}{l} \mathbf{y}_j = \mathbf{X}_j\beta + \mathbf{Z}_j\mathbf{b}_j + \epsilon_j \\ \mathbf{b}_j \sim N(\mathbf{0}, \Psi) \end{array} \right\} \Rightarrow p(\mathbf{y}|\beta, \mathbf{X}_j, \mathbf{Z}_j) = \int p(\mathbf{y}|\beta, \mathbf{X}_j, \mathbf{Z}_j, \mathbf{b}_j) \times p(\mathbf{b}_j|\Psi) d\mathbf{b}_j$$
$$= \text{dnorm}(\mathbf{X}_j\beta, \Sigma),$$

where Σ depends on Ψ , $\mathbf{Z}_j$ and σ^2 .

The likelihood based on this normal density can be written down and optimized.

Estimation

GLMMs: Estimation for GLMMs is more difficult:

$$\left. \begin{array}{l} \mathbf{y}_{i,j} \sim f(y_{i,j} | \mathbf{x}_{i,j}^\top \boldsymbol{\beta} + \mathbf{z}_{i,j}^\top \mathbf{b}_j) \\ \mathbf{b}_j \sim N(\mathbf{0}, \Psi) \end{array} \right\} \Rightarrow p(\mathbf{y}_j | \boldsymbol{\beta}, \mathbf{X}_j, \mathbf{Z}_j) = \int p(\mathbf{y}_j | \boldsymbol{\beta}, \mathbf{X}_j, \mathbf{Z}_j, \mathbf{b}_j) \times p(\mathbf{b}_j | \Psi) d\mathbf{b}_j$$

is not a normal density

The likelihood can't be written down in a closed form.

Obtaining MLEs requires iteration of the following:

- approximating (derivatives of) the above integral;
- optimization steps to find the MLE.

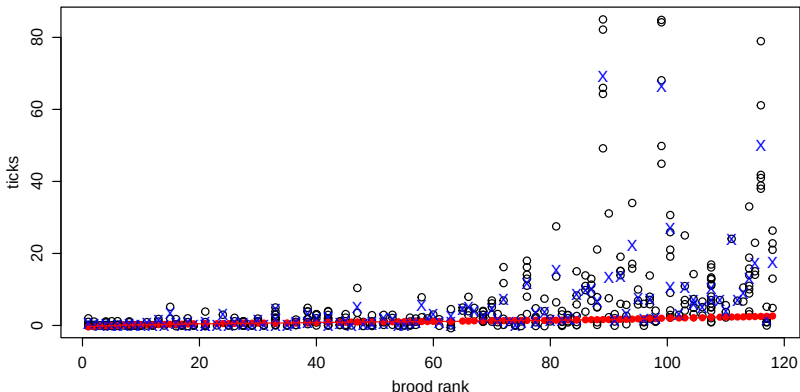
Non-convergence of optimization procedures for parameter estimation in such models is common.

Grousetick example

```
fit0<-glm( TICKS~ cHEIGHT+as.factor(YEAR),family=poisson,data=grouseticks)
```

```
summary(fit)$coef
```

	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	1.06715919	0.07961023	13.404799	4.356064e-34
## cHEIGHT	-0.01331815	0.00120441	-11.057815	5.646538e-25
## as.factor(YEAR)96	0.76833413	0.10607991	7.242975	2.280915e-12
## as.factor(YEAR)97	-0.46374571	0.10926018	-4.244417	2.728417e-05



GLMMs with glmer

```
fit1<-glmer( TICKS ~ cHEIGHT + as.factor(YEAR) + (1|BROOD) ,family=poisson,data=grouseticks)
```

```
BIC(fit1)
```

```
## [1] 2008.07
```

```
BIC(fit0)
```

```
## [1] 4398.807
```

```
summary(fit1)$coef
```

##	Estimate	Std. Error	z value	Pr(> z)
## (Intercept)	0.50919618	0.186616637	2.728568	6.360998e-03
## cHEIGHT	-0.02386615	0.003010643	-7.927260	2.240344e-15
## as.factor(YEAR)96	1.13587983	0.242387482	4.686215	2.783040e-06
## as.factor(YEAR)97	-1.00113687	0.269685515	-3.712238	2.054345e-04

Nested nests

Another variable is **LOCATION**, specifying spatial location of each brood.

Question: What are the grouping factors, and how are they related?

Answer: BROOD is nested within LOCATION
(but note, each brood already has a unique identifier)

```
fit2<-glmer( TICKS ~ cHEIGHT + as.factor(YEAR) + (1|BROOD) + (1|LOCATION) ,family=poisson,  
BIC(fit2)  
## [1] 2011.87  
BIC(fit1)  
## [1] 2008.07
```

We'll continue with `fit1`.

Checking assumptions

The fitted mean for each brood j is given by

$$\hat{\theta}_j = \exp(\hat{\beta}_1 + \hat{\beta}_2 \times \text{cHEIGHT}_j + \widehat{\text{YEAR}_j} + \hat{b}_j)$$

The fitted model for data within brood j is

$$\hat{p}(y_{i,j}) = \text{dpois}(\hat{\theta}_j)$$

$$\hat{E}[y_{i,j}] = \hat{\theta}_j$$

$$\hat{V}[y_{i,j}] = \hat{\theta}_j$$

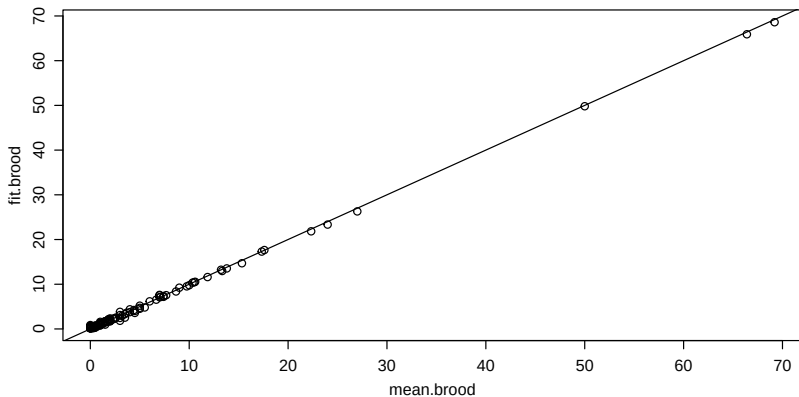
Question: How can we check (one aspect of) this assumption?

Answer: Check to see if sample variance corresponds to fitted variance:

- If some Poisson model is correct, sample mean $\approx$ sample variance.
- If the Poisson regression is correct, fitted mean $\approx$ sample variance.

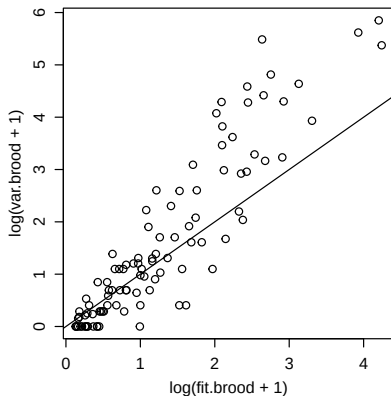
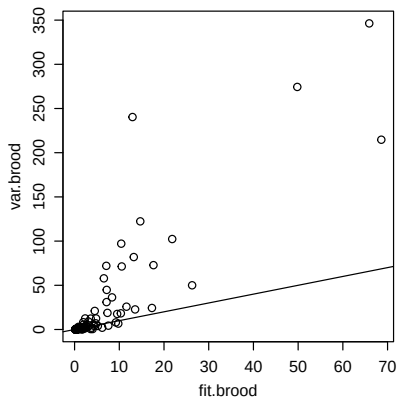
Checking assumptions

```
var.brood<-tapply(grouseticks$TICKS,grouseticks$BROOD,var)
mean.brood<-tapply(grouseticks$TICKS,grouseticks$BROOD,mean)
fit.brood<-tapply(fitted(fit1),grouseticks$BROOD,mean)
```



Does this plot surprise you? Why or why not?

Checking assumptions



These data are generally *overdispersed* relative to a Poisson model.
There is more *within-brood* variance than can be ascribed to Poisson variation.

Models for overdispersion

Consider the following model:

$$\begin{aligned}\mu_j &= \boldsymbol{\beta}^T \mathbf{x}_j + \mathbf{b}_j^T \mathbf{z}_j \\ y_{i,j} &\sim \text{Poisson}(e^{\mu_j + \epsilon_{i,j}}) \\ \epsilon_{i,j} &\sim i.i.d. \text{ from some distribution}\end{aligned}$$

- If $\text{Var}[\epsilon_{i,j}] = 0$ then the model is the Poisson GLMM;
- If $\text{Var}[\epsilon_{i,j}] > 0$ then the model allows for overdispersion.

Common overdispersion models:

$$\begin{aligned}\epsilon_{i,j} &\sim i.i.d. N(0, \sigma^2) \\ \epsilon_{i,j} &\sim i.i.d. \text{ log gamma distribution}\end{aligned}$$

The latter corresponds to *negative binomial regression*.
Both can be fit in with `glmer`.

Overdispersed Poisson via random effects

```
grouseticks[1:5,]
```

##	INDEX	TICKS	BROOD	HEIGHT	YEAR	LOCATION	cHEIGHT
## 1	1	0	501	465	95	32	2.759305
## 2	2	0	501	465	95	32	2.759305
## 3	3	0	502	472	95	36	9.759305
## 4	4	0	503	475	95	37	12.759305
## 5	5	0	503	475	95	37	12.759305

```
fit.o1<-glmer( TICKS ~ cHEIGHT + as.factor(YEAR) + (1|BROOD) + (1|INDEX),family=poisson,data=grouseticks)
```

```
fit.o1
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: poisson ( log )
## Formula: TICKS ~ cHEIGHT + as.factor(YEAR) + (1 | BROOD) + (1 | INDEX)
## Data: grouseticks
##      AIC      BIC    logLik deviance df.resid
## 1794.040 1818.034 -891.020 1782.040      397
## Random effects:
## Groups Name      Std.Dev.
## INDEX  (Intercept) 0.5435
## BROOD  (Intercept) 0.9085
## Number of obs: 403, groups: INDEX, 403; BROOD, 118
## Fixed Effects:
##      (Intercept)      cHEIGHT  as.factor(YEAR)96  as.factor(YEAR)97
##      0.40999      -0.02405      1.14892      -0.99061
```

Overdispersed Poisson via random effects

Comparison to Poisson regression

```
BIC(fit1)
```

```
## [1] 2008.07
```

```
BIC(fit.o1)
```

```
## [1] 1818.034
```

```
summary(fit1)$coef
```

##	Estimate	Std. Error	z value	Pr(> z)
## (Intercept)	0.50919618	0.186616637	2.728568	6.360998e-03
## cHEIGHT	-0.02386615	0.003010643	-7.927260	2.240344e-15
## as.factor(YEAR)96	1.13587983	0.242387482	4.686215	2.783040e-06
## as.factor(YEAR)97	-1.00113687	0.269685515	-3.712238	2.054345e-04

```
summary(fit.o1)$coef
```

##	Estimate	Std. Error	z value	Pr(> z)
## (Intercept)	0.40998718	0.190354324	2.153811	3.125502e-02
## cHEIGHT	-0.02404906	0.003044808	-7.898382	2.825465e-15
## as.factor(YEAR)96	1.14891702	0.246301687	4.664674	3.091070e-06
## as.factor(YEAR)97	-0.99061439	0.273457963	-3.622547	2.917162e-04

Overdispersed Poisson via the negative binomial distribution

```
fit.o2<-glmer.nb(TICKS ~ cHEIGHT + as.factor(YEAR) + (1|BROOD),data=grouseticks)
```

```
BIC(fit.o2)
```

```
## [1] 1815.938
```

```
summary(fit.o2)$coef
```

##	Estimate	Std. Error	z value	Pr(> z)
## (Intercept)	0.52854237	0.188250992	2.807647	4.990486e-03
## cHEIGHT	-0.02390358	0.003014457	-7.929647	2.197695e-15
## as.factor(YEAR)96	1.13183765	0.243944397	4.639736	3.488542e-06
## as.factor(YEAR)97	-0.99381870	0.270617604	-3.672410	2.402741e-04

Random effects logistic regression

Logistic regression:

$$\Pr(y_i = 1) = \theta_i = \frac{e^{\beta^T \mathbf{x}_i}}{1 + e^{\beta^T \mathbf{x}_i}}$$
$$\log \frac{\theta_i}{1 - \theta_i} = \beta^T \mathbf{x}_i$$

Mixed effects logistic regression:

$$\Pr(y_{i,j} = 1) = \theta_{i,j} = \frac{e^{\beta^T \mathbf{x}_{i,j} + \mathbf{b}_j^T \mathbf{z}_{i,j}}}{1 + e^{\beta^T \mathbf{x}_{i,j} + \mathbf{b}_j^T \mathbf{z}_{i,j}}}$$
$$\log \frac{\theta_{i,j}}{1 - \theta_{i,j}} = \beta^T \mathbf{x}_{i,j} + \mathbf{b}_j^T \mathbf{z}_{i,j}$$

Data types for logistic regression

Binary data

y	g	x1	x2
0	1	2.3	3.2
0	1	2.3	2.1
1	1	2.3	5.4
0	2	4.1	1.6
1	2	4.1	3.2
1	2	4.1	1.2
.	.	.	.
.	.	.	.

```
fit <- glmmer( y ~ x1 + x2 + (x2|g) , family=binomial )
```

Data types for logistic regression

Binomial data

y	n	g	x1	x2
4	10	1	2.3	3.2
2	12	1	2.3	2.1
5	8	1	2.3	5.4
3	10	2	4.1	1.6
6	16	2	4.1	3.2
8	9	2	4.1	1.2
.	.	.	.	.
.	.	.	.	.

Here, the model is

$$y_{i,j} \sim \text{binomial}(\theta_{i,j}, n_{i,j})$$

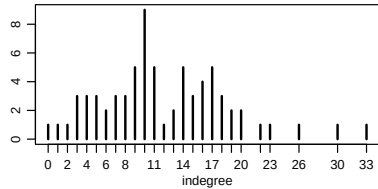
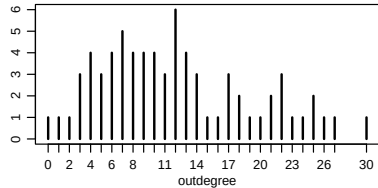
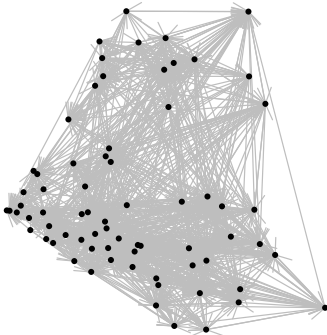
```
fit <- glmer( cbind(y,n-y) ~ x1 + x2 + (x2|g) , family=binomial )
```

Example: Social network analysis

Network and relational data: Data measured on pairs (dyads) of units.

Friendship study: What characteristics of people lead to friendship ties?

Example: Social network analysis



Example: Social network analysis

```
dim(Y)
## [1] 71 71

dim(X)
## [1] 71 71 2

dimnames(X)[[3]]
## [1] "advice"      "samepractice"
```

Logistic regression

```
X[1:10,]  
  
##          advice samepractice  
## [1,]         1             0  
## [2,]         0             1  
## [3,]         1             0  
## [4,]         1             1  
## [5,]         0             1  
## [6,]         0             0  
## [7,]         1             1  
## [8,]         0             0  
## [9,]         0             0  
## [10,]        1             1  
  
y[1:10]  
  
## [1] 0 0 0 0 0 0 0 0 0 0  
  
fit0<-glm( y ~ X, family=binomial)
```

```
summary(fit0)$coef
```

```
##          Estimate Std. Error   z value    Pr(>|z|)  
## (Intercept) -2.3043528 0.07069337 -32.59645 4.604979e-233  
## Xadvice      0.3059047 0.10852842   2.81866  4.822457e-03  
## Xsamepractice 1.1521500 0.08320930  13.84641  1.337308e-43
```

Network dependence

Do you think the 71×70 network ties are independent?

What sort of dependence might you expect? What are the grouping factors?

- sender (row)
- receiver (column)
- dyad (pair)

Simulate and compare

```
ysim<-rbinom(length(y),1,fit0$fitted)

mean(ysim)

## [1] 0.178672

mean(y)

## [1] 0.171831

fit0sim<-glm( ysim ~ X, family=binomial)
fit0sim$coef

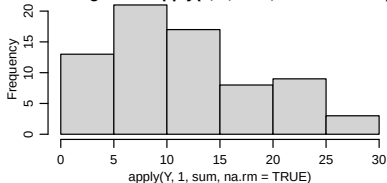
##      (Intercept)      Xadvice Xsamepractice
##      -2.2481936      0.2360881      1.1571658

fit0$coef

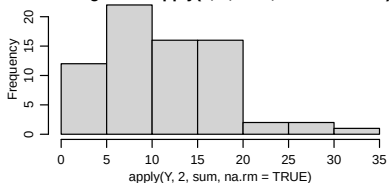
##      (Intercept)      Xadvice Xsamepractice
##      -2.3043528      0.3059047      1.1521500
```


Network dependence

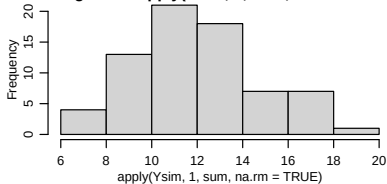
Histogram of `apply(Y, 1, sum, na.rm = TRUE)`



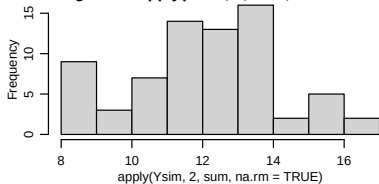
Histogram of `apply(Y, 2, sum, na.rm = TRUE)`



Histogram of `apply(Ysim, 1, sum, na.rm = TRUE)`



Histogram of `apply(Ysim, 2, sum, na.rm = TRUE)`



```
cor( Y[upper.tri(Y)],t(Y)[upper.tri(Y)],use="pairwise.complete")  
## [1] 0.1926326  
  
cor( Ysim[upper.tri(Y)],t(Ysim)[upper.tri(Y)],use="pairwise.complete")  
## [1] 0.06496955
```

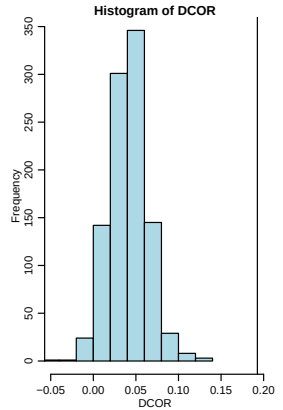
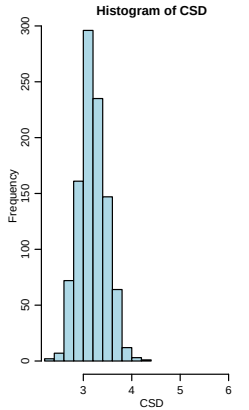
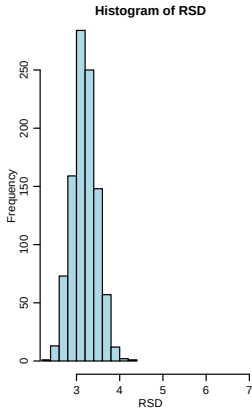
Simulation study

```
RSD<-CSD<-DCOR<-NULL
for(s in 1:1000){

  ysim<-rbinom(length(y),1,fit0$fitted)
  Ysim<-Y ; Ysim[!is.na(Ysim)]<-ysim

  RSD<-c(RSD,sd(apply(Ysim,1,sum,na.rm=TRUE)))
  CSD<-c(CSD,sd(apply(Ysim,2,sum,na.rm=TRUE)))
  DCOR<-c(DCOR,cor(Ysim[upper.tri(Y)],t(Ysim)[upper.tri(Y)],
    use="pairwise.complete"))
}
```

Lack of fit



Social relations model

We would like a model that allows for

- across-sender heterogeneity in outdegrees
- across-receiver heterogeneity in indegrees
- within dyad reciprocity

$$\log \frac{\theta_{i,j}}{1-\theta_{i,j}} = \beta^T \mathbf{x}_{i,j} + a_i + b_j + \epsilon_{\{i,j\}}$$

$$a_1, \dots, a_n \sim \text{i.i.d. } N(0, \tau_a^2)$$

$$b_1, \dots, b_n \sim \text{i.i.d. } N(0, \tau_b^2)$$

$$\{\epsilon_{\{i,j\}}\} \sim \text{i.i.d. } N(0, \sigma^2)$$

Note $\theta_{i,j}$ and $\theta_{j,i}$ share *the same* $\epsilon_{\{i,j\}}$.

- $\{a_i\}$ represent heterogeneity across senders of ties (sociability)
- $\{b_j\}$ represent heterogeneity across receivers of ties (popularity)
- $\{\epsilon_{\{i,j\}}\}$ represents similarity within a dyad (reciprocity)

Fitting the SRM-lite with glmer

```
rows<-matrix(1:nrow(Y),nrow(Y),nrow(Y))
rlab<-c(rows)[!is.na(c(Y))]  
clab<-c(t(rows))[!is.na(c(Y))]  
dlab<-t(apply(cbind(rlab,clab) ,1,sort) )  
dlab<-paste(dlab[,1],dlab[,2],sep=".")  
G<-as.data.frame(cbind(rlab,clab,dlab) )
```

```
G[1:10,]
```

##	rlab	clab	dlab
## 1	2	1	1.2
## 2	3	1	1.3
## 3	4	1	1.4
## 4	5	1	1.5
## 5	6	1	1.6
## 6	7	1	1.7
## 7	8	1	1.8
## 8	9	1	1.9
## 9	10	1	1.10
## 10	11	1	1.11

Fitting the SRM-lite with glmer

Question: How can we fit the logistic SRM?

```
fit.srm<-glmer( y ~ X + (1|G$rlab) + (1|G$clab) + (1|G$dlab) , family=binomial)
```

Unfortunately, this stumps glmer. The best we can do is the following:

```
fit.ab<-glmer( y ~ X + (1|G$rlab) + (1|G$clab) , family=binomial)
```

Fitting the SRM-lite with glmer

```
BIC(fit.ab)
```

```
## [1] 4078.228
```

```
BIC(fit0)
```

```
## [1] 4363.334
```

```
VarCorr(fit.ab)
```

```
## Groups Name Std.Dev.
```

```
## G$rlab (Intercept) 0.74527
```

```
## G$clab (Intercept) 0.67392
```

```
summary(fit.ab)$coef
```

##	Estimate	Std. Error	z value	Pr(> z)
## (Intercept)	-2.61297936	0.14374547	-18.1778200	7.734797e-74
## Xadvice	0.02158876	0.12699036	0.1700031	8.650077e-01
## Xsamepractice	1.25084242	0.08953187	13.9709187	2.345810e-44

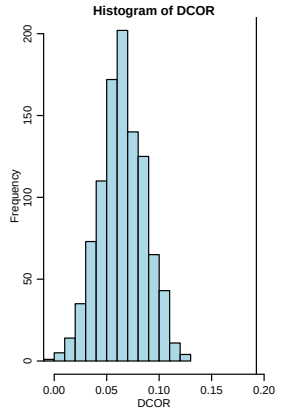
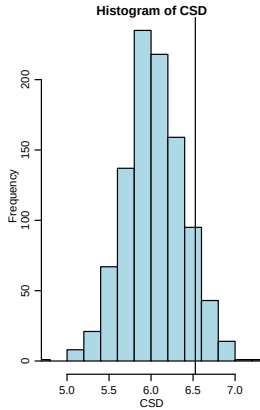
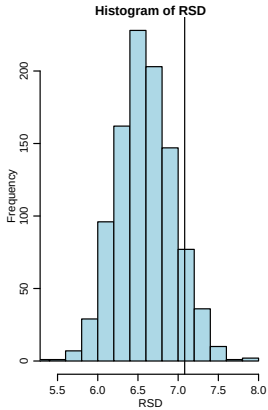
Simulation study

```
RSD<-CSD<-DCOR<-NULL
for(s in 1:1000){

  ysim<-rbinom(length(y),1,fitted(fit.ab))
  Ysim<-Y ; Ysim[!is.na(Ysim)]<-ysim

  RSD<-c(RSD,sd(apply(Ysim,1,sum,na.rm=TRUE)))
  CSD<-c(CSD,sd(apply(Ysim,2,sum,na.rm=TRUE)))
  DCOR<-c(DCOR,cor(Ysim[upper.tri(Y)],t(Ysim)[upper.tri(Y)],
    use="pairwise.complete"))
}
```


Lack of fit



AME model fitting routine

Description:

An MCMC routine providing a fit to an additive and multiplicative effects (AME) regression model to relational data of various types

Usage:

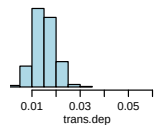
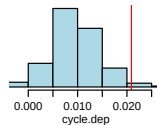
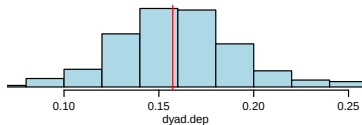
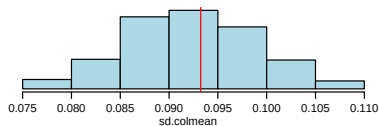
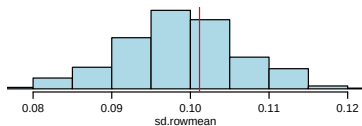
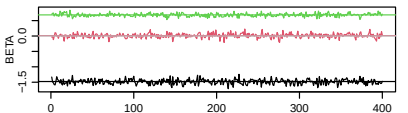
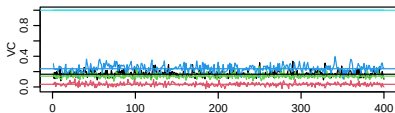
```
ame(Y, Xdyad=NULL, Xrow=NULL, Xcol=NULL, family, R=0, rvar = !(family %in% c("rrl", "ord")),  
    cvar = TRUE, dcor = !symmetric, nvar=TRUE,  
    intercept=!is.element(family,c("rrl","ord")),  
    symmetric=FALSE,  
    odmax=rep(max(apply(Y>0,1,sum,na.rm=TRUE)),nrow(Y)), seed = 1, nsc=10000,  
    burn = 500, odens = 25, plot=TRUE, print = TRUE, gof=TRUE,  
    prior=list())
```

Arguments:

Y: an $n \times n$ square relational matrix of relations. See family

Fitting the SRM

```
library(amen)  
fitSRM<-ame(Y, Xd=X, family="bin")
```



Fitting the SRM

```
summary(fitSRRM)
```

```
##
```

```
## Regression coefficients:
```

	pmean	psd	z-stat	p-val
## intercept	-1.483	0.081	-18.301	0.000
## advice.dyad	0.014	0.072	0.194	0.846
## samepractice.dyad	0.685	0.052	13.039	0.000

```
##
```

```
## Variance parameters:
```

	pmean	psd
## va	0.171	0.039
## cab	0.035	0.022
## vb	0.144	0.031
## rho	0.241	0.048
## ve	1.000	0.000