Social relations models for binary and valued relations

567 Statistical analysis of social networks

Peter Hoff

Statistics, University of Washington

Limitations of ERGMs

We've developed, fit and evaluated models of the form

$\mathbf{Y} \sim$ sender covariates + receiver covariates + dyad covariates + network statistics

where "network statistics" includes

- nodal in- and outdegree statistics;
- number of mutual dyads.

Including the above network statistics allows the model to represent

- nodal heterogeneity of in- and outdegrees;
- within-dyad dependence of ties.

The `ergm` software allows for additional network statistics in the model.

Limitations of ERGMs

We've encountered several limitations of ERGMs so far:

1. computationally intensive estimation algorithms;
2. confounding of nodal covariate effects with nodal degree effects;
3. models are limited to binary relational data.

The third one is particularly restrictive - **most relations are not binary**:

- `monk data:` ranked relational outcomes;
- `conflict data:` positive, negative and valued military relations;
- `primate data:` counts of relational events.

Seven of the first ten datasets at `http://moreno.ss.uci.edu/data.html` have valued relations.

A strategy for valued relations

Network patterns can be viewed as dependencies among observations:

- Reciprocity can be viewed as **dependence** between $y_{i,j}$ and $y_{j,i}$.
- Outdegree heterogeneity can be viewed as **dependence** among $\{y_{i,1}, \ldots, y_{i,n}\}$.
- Indegree heterogeneity can be viewed as **dependence** among $\{y_{1,j}, \ldots, y_{n,j}\}$.

Statistical methods for dependent data use **normal random effects models** to represent dependencies among observations.

Linear regression

Suppose our data consist of the following:

- $\mathbf{Y} = \{y_{i,j} : i \neq j\}$, an $n \times n$ sociomatrix of "continuous" entries;
- $\mathbf{X} = \{\mathbf{x}_{i,j} : i \neq j\}$, an $n \times n \times p$ array of covariates or predictors.

A linear regression model posits that

$$y_{i,j} = \boldsymbol{\beta}^T \mathbf{x}_{i,j} + \epsilon_{i,j}$$

where

- $\boldsymbol{\beta}$ is an unknown $p \times 1$ vector of regression parameters;
- $\boldsymbol{\beta}^T \mathbf{x} = \boldsymbol{\beta} \cdot \mathbf{x} = \sum_{k=1}^{p} \beta_k x_k$ is the dot-product of $\boldsymbol{\beta}$ and $\mathbf{x}$.
- $\{\epsilon_{i,j} : i \neq j\}$ are mean-zero disturbance terms or "noise."

Ordinary least squares

$$y_{i,j} = \boldsymbol{\beta}^T \mathbf{x}_{i,j} + \epsilon_{i,j}$$

Given data $\mathbf{Y}$ and $\mathbf{X}$, how should we estimate $\boldsymbol{\beta}$?

Ordinary least squares (OLS):

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{\text{ols}} &= \arg\min_{\beta} \sum_{i \neq j} (y_{i,j} - \boldsymbol{\beta}^T \mathbf{x}_{i,j})^2 \\ &= (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \mathbf{y}\end{aligned}$$

where

- $\boldsymbol{y}$ is the vectorized version of the matrix $\mathbf{Y}$;
- $\tilde{\mathbf{X}}$ is the "matricized" version of the array $\mathbf{X}$.

Maximum likelihood estimation

Suppose we assume $\{\epsilon_{i,j} : i \neq j\} \sim$ i.i.d. normal$(0, \sigma^2)$.

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_{\text{mle}} &= \arg\max_{\beta} \log p(\mathbf{Y}|\mathbf{X}, \beta) \\
&= \arg\max_{\beta} -\tfrac{1}{2} \sum_{i \neq j} (y_{i,j} - \boldsymbol{\beta}^T \mathbf{x}_{i,j})^2 \\
&= \arg\min_{\beta} \sum_{i \neq j} (y_{i,j} - \boldsymbol{\beta}^T \mathbf{x}_{i,j})^2 \\
&= \hat{\boldsymbol{\beta}}_{\text{ols}}
\end{aligned}$$

Under the assumption of i.i.d. normal errors, $\hat{\boldsymbol{\beta}}_{\text{mle}} = \hat{\boldsymbol{\beta}}_{\text{ols}}$.

Properties of OLS estimators

Under very general conditions on $\mathbf{X}$ and $\{\epsilon_{i,j} : i \neq j\}$,

- $\hat{\boldsymbol{\beta}}_{\text{ols}}$ is **unbiased** ;
- $\hat{\boldsymbol{\beta}}_{\text{ols}}$ is **consistent**.

Neither of these properties rely on the $\epsilon_{i,j}$'s being normal or independent.
So why worry about network dependence?

Problem: Generally we are interested in more than a point estimate of $\boldsymbol{\beta}$. We want

- standard errors;
- p-values;
- hypothesis tests.

These items require an accurate model for the dependence among the $\epsilon_{i,j}$'s.

A bit of linear algebra

Matrix form of linear regression:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

The variance of $\hat{\beta}_{\text{ols}}$ can be computed with some matrix algebra:

$$\begin{aligned}
\mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \\
\hat{\boldsymbol{\beta}} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}) \\
&= \boldsymbol{\beta} + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\epsilon}
\end{aligned}$$

From this we see $\mathrm{E}[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta}$ if $\mathbf{X}$ and $\boldsymbol{\epsilon}$ are uncorrelated, and

$$\begin{aligned}
\mathrm{Cov}[\boldsymbol{\beta}] &= \mathrm{E}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}] \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathrm{Cov}[\boldsymbol{\epsilon}]\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}.
\end{aligned}$$

The variance of $\hat{\boldsymbol{\beta}}$, and therefore its standard errors, depends on $\mathrm{Cov}[\boldsymbol{\epsilon}]$.

General linear model

Returning to network and relational data:

Normal linear regression model:

$$y_{i,j} = \boldsymbol{\beta}^T \mathbf{x}_{i,j} + \epsilon_{i,j}$$
$$\{\epsilon_{i,j}\} \sim \text{i.i.d. normal}(0, \sigma^2)$$

General linear regression model:

$$y_{i,j} = \boldsymbol{\beta}^T \mathbf{x}_{i,j} + \epsilon_{i,j}$$
$$\{\epsilon_{i,j}\} \sim \ ?$$

Network covariance: What correlations do we expect among the $\epsilon_{i,j}$'s ?

Network covariance

What were the main deviations from independence we've seen in logistic regression models?

- excess heterogeneity of in- and outdegrees;
- excess reciprocity.

An analogy to continuous variables suggests the possibility of

- within-dyad correlation between $\epsilon_{i,j}$ and $\epsilon_{j,i}$;
- within-sender correlation among $\{\epsilon_{i,1}, \ldots, \epsilon_{i,n}\}$;
- within-receiver correlation among $\{\epsilon_{1,j}, \ldots, \epsilon_{n,j}\}$;

Let's evaluate some relational data for the presence of such dependence.

Example: Effects of conflict on trade

Trade data on 130 countries from 1990-2000

Outcome variable:

- $y_{i,j}$ = average trade between 130 countries from 1990-2000.

Nodal covariates:

- population
- gdp
- polity

Dyadic covariates:

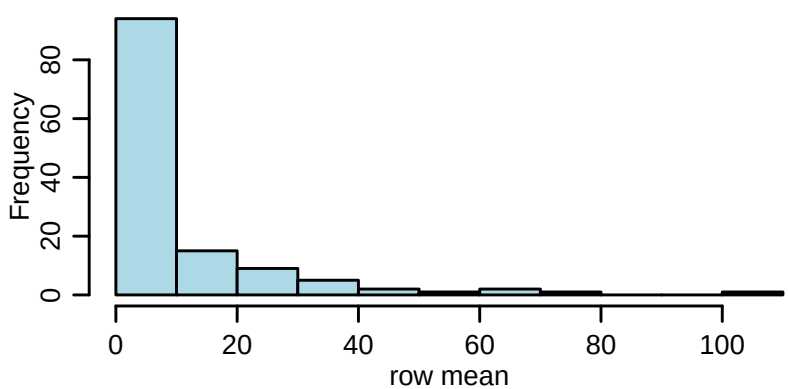
- number of conflicts
- geographic distance
- number of shared intergovernmental organizations

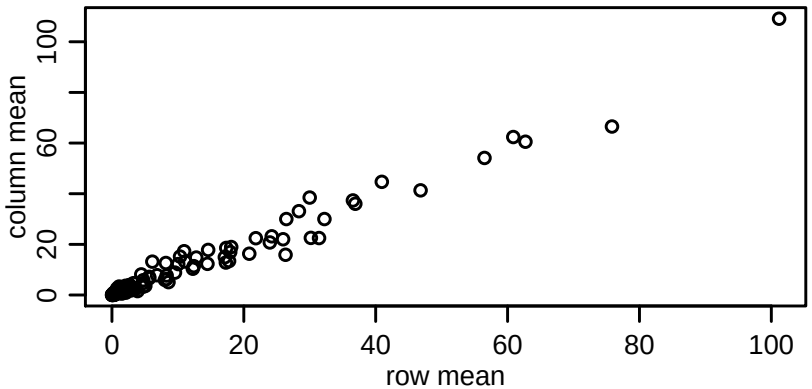
(trade, population, gdp and distance are on the log scale)

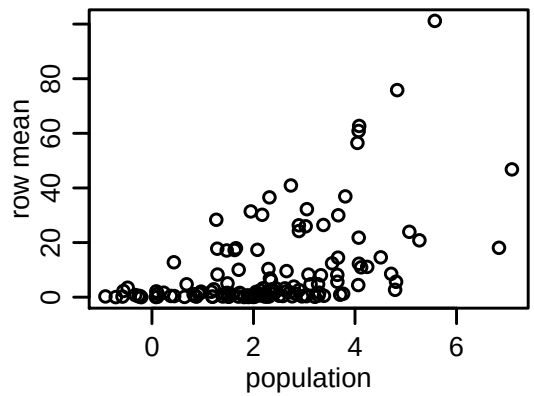
Some exploratory data analysis

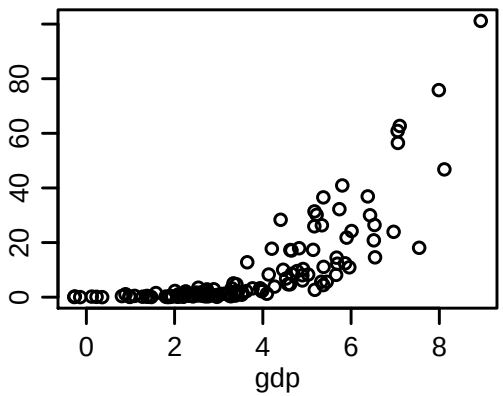
Instead of out and indegrees, we can look at row and column averages:

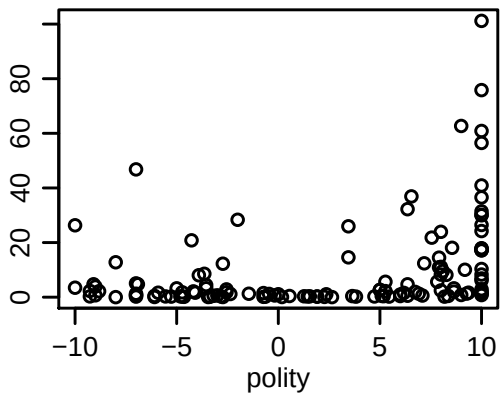
- row mean = average (logged) exports
- column mean = average (logged) imports



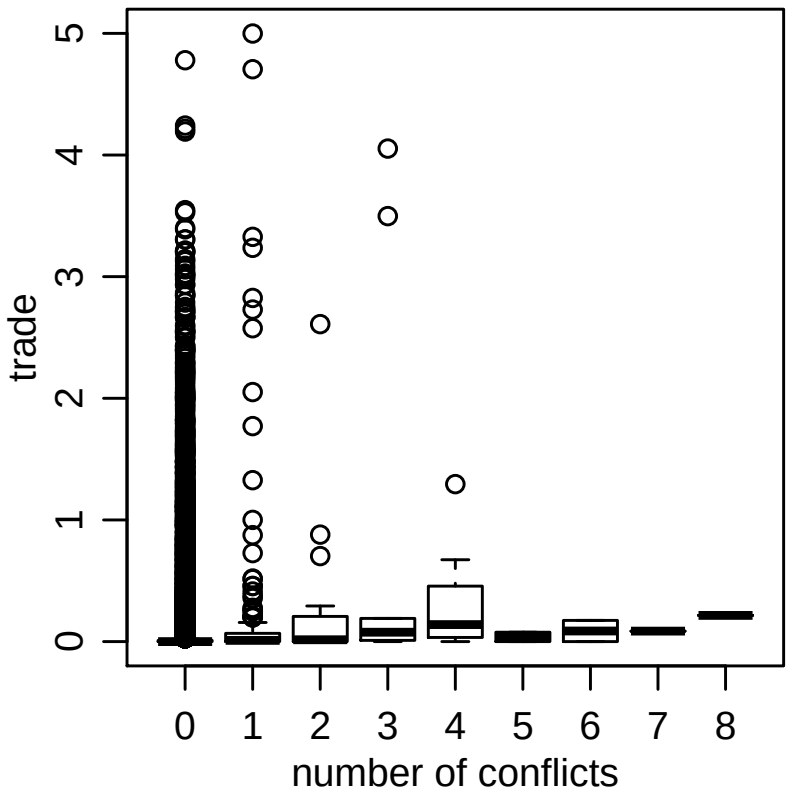


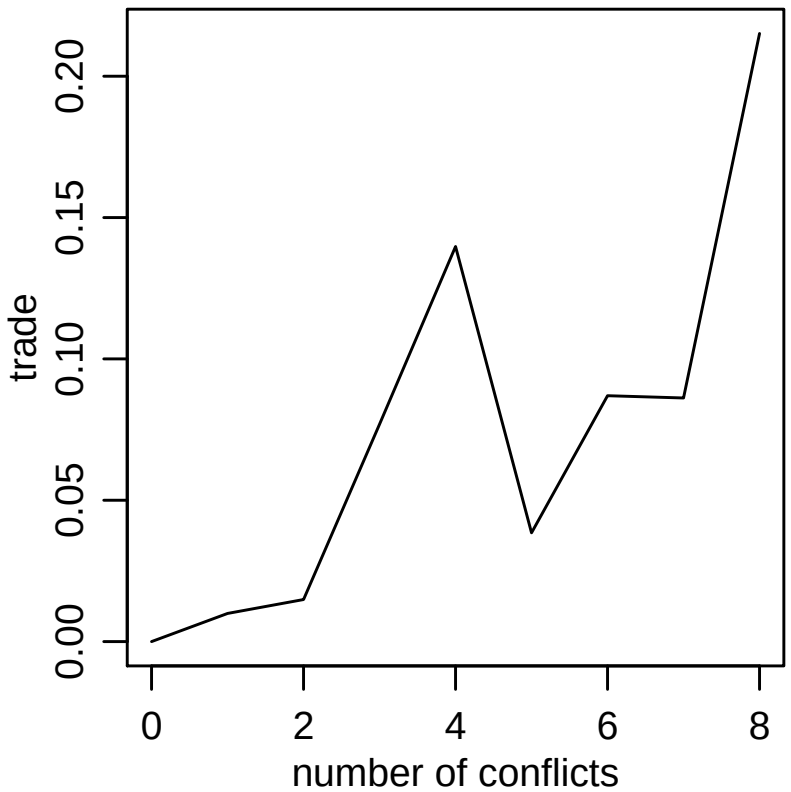






Some exploratory data analysis

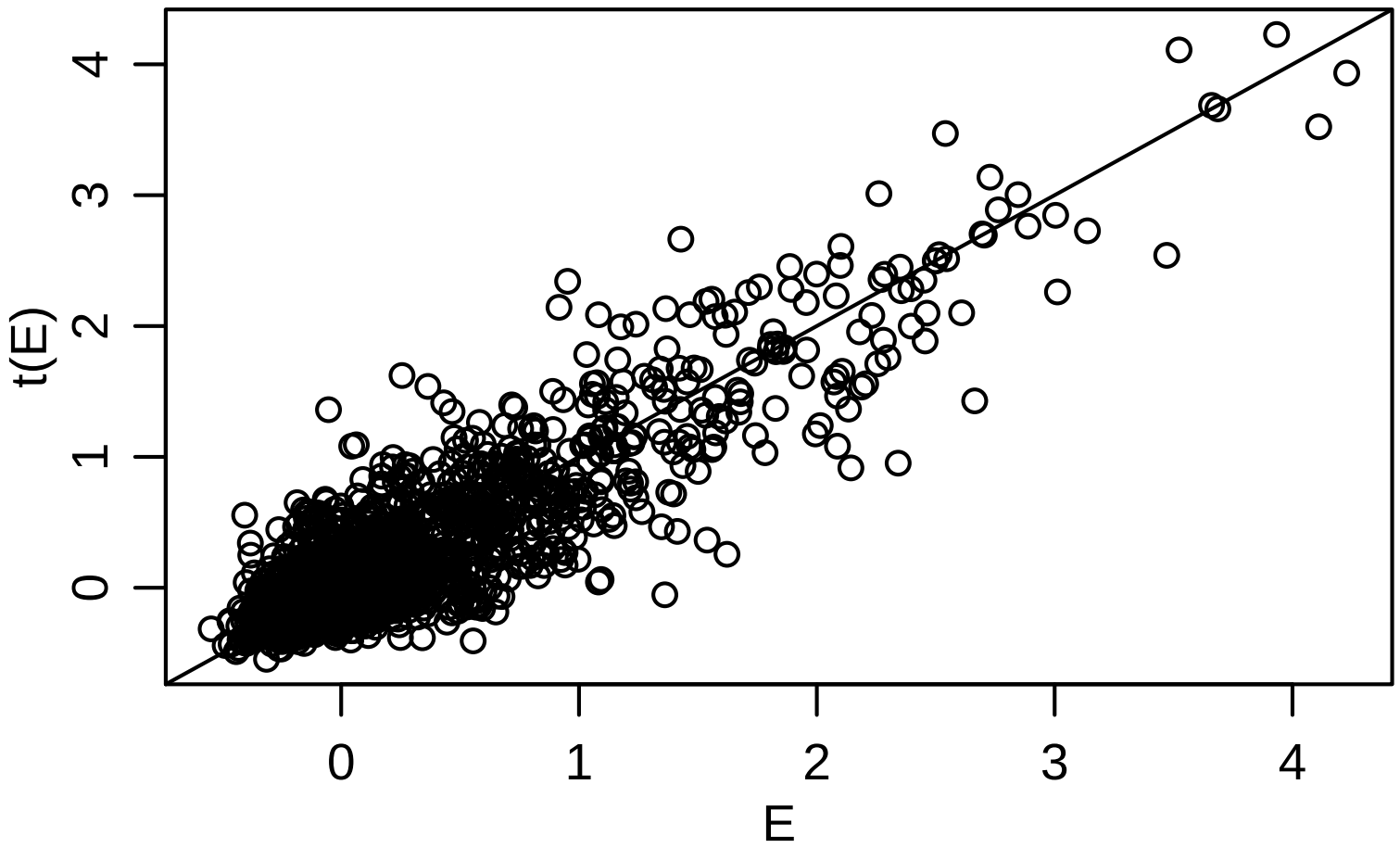




OLS estimation

```
dimnames(X)[[3]]

##  [1] "rpop" "rgdp" "rpty" "cpop" "cgdp" "cpty" "pti"  "sigo" "dst"  "con"

yv<-c(Y)
Xm<-apply(X,3,"c")

fit_ols<-lm(yv~Xm)

summary(fit_ols)$coef

##                 Estimate   Std. Error     t value      Pr(>|t|)
## (Intercept) -0.2984079233 1.118213e-02 -26.6861550 1.091219e-153
## Xmrpop      -0.0260558723 2.194809e-03 -11.8715876  2.241922e-32
## Xmrgdp       0.0584609784 1.888985e-03  30.9483598 1.462845e-204
## Xmrpty      -0.0007722494 3.483605e-04  -2.2168111  2.664938e-02
## Xmcpop      -0.0239575625 2.194458e-03 -10.9173040  1.180708e-27
## Xmcgdp       0.0558042060 1.888935e-03  29.5426872 4.923833e-187
## Xmcpty      -0.0001799325 3.483234e-04  -0.5165674  6.054650e-01
## Xmpti        0.0002902070 4.520667e-05   6.4195622  1.403381e-10
## Xmsigo       0.0053237026 1.979838e-04  26.8895835 5.824836e-156
## Xmdst       -0.0458066907 3.805497e-03 -12.0369790  3.114190e-33
## Xmcon        0.0343657702 9.274728e-03   3.7053128  2.118089e-04
```

Residual diagnostics - dyadic correlation




```
E<-matrix(0,nrow(Y),ncol(Y))
E[!is.na(Y)] <- fit_ols$res

cor(c(E),c(t(E)),use="complete"  )

## [1] 0.9107801
```

Residual diagnostics - dyadic correlation

The residual within-dyad correlation is quite high: $\hat{\rho} = 0.91$.

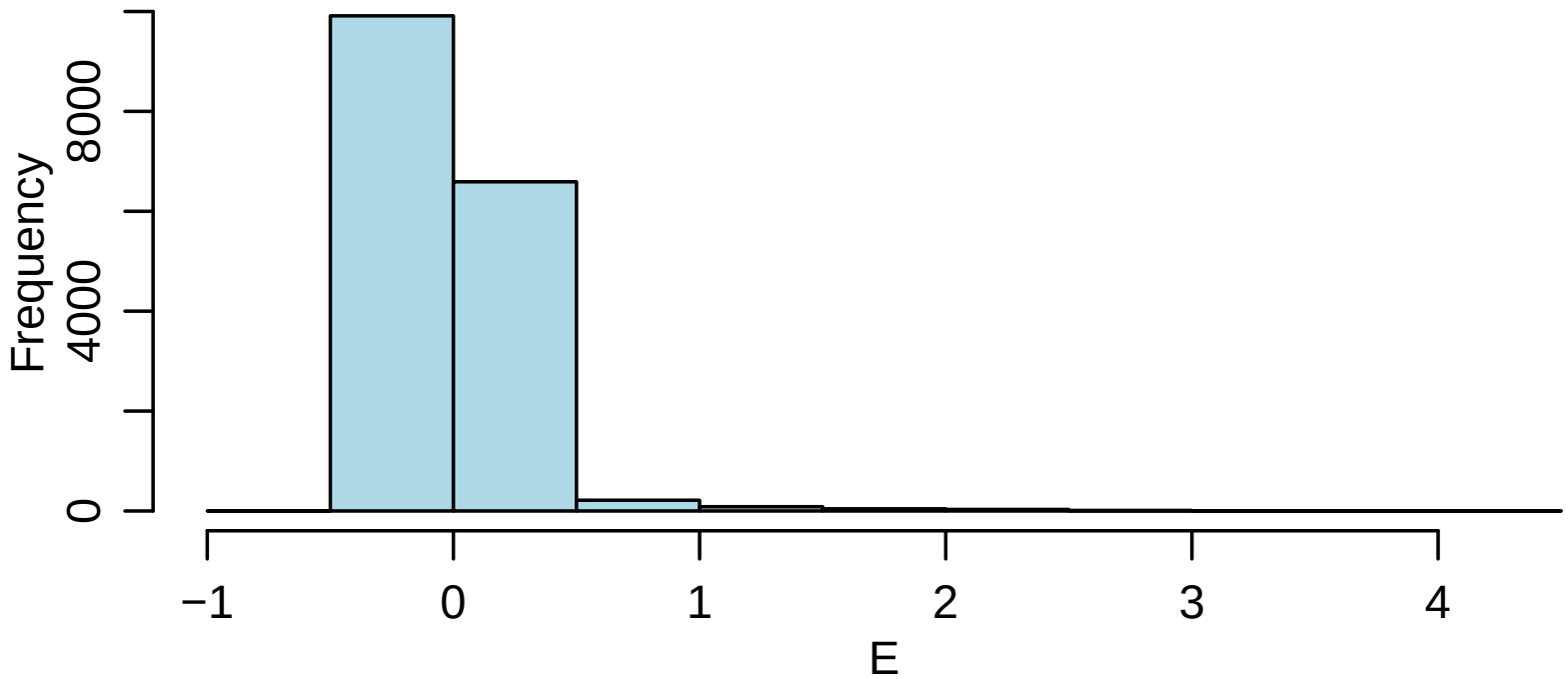
Is it large enough to reject $H : \text{Cor}[\epsilon_{i,j}, \epsilon_{j,i}] = \rho = 0$?

Hypothesis test:

Compare $\hat{\rho}$ with what we would expect under the model

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

A normal theory test for of $H : \rho = 0$ exists, but we might be concerned with the normality assumption:



Permutation tests

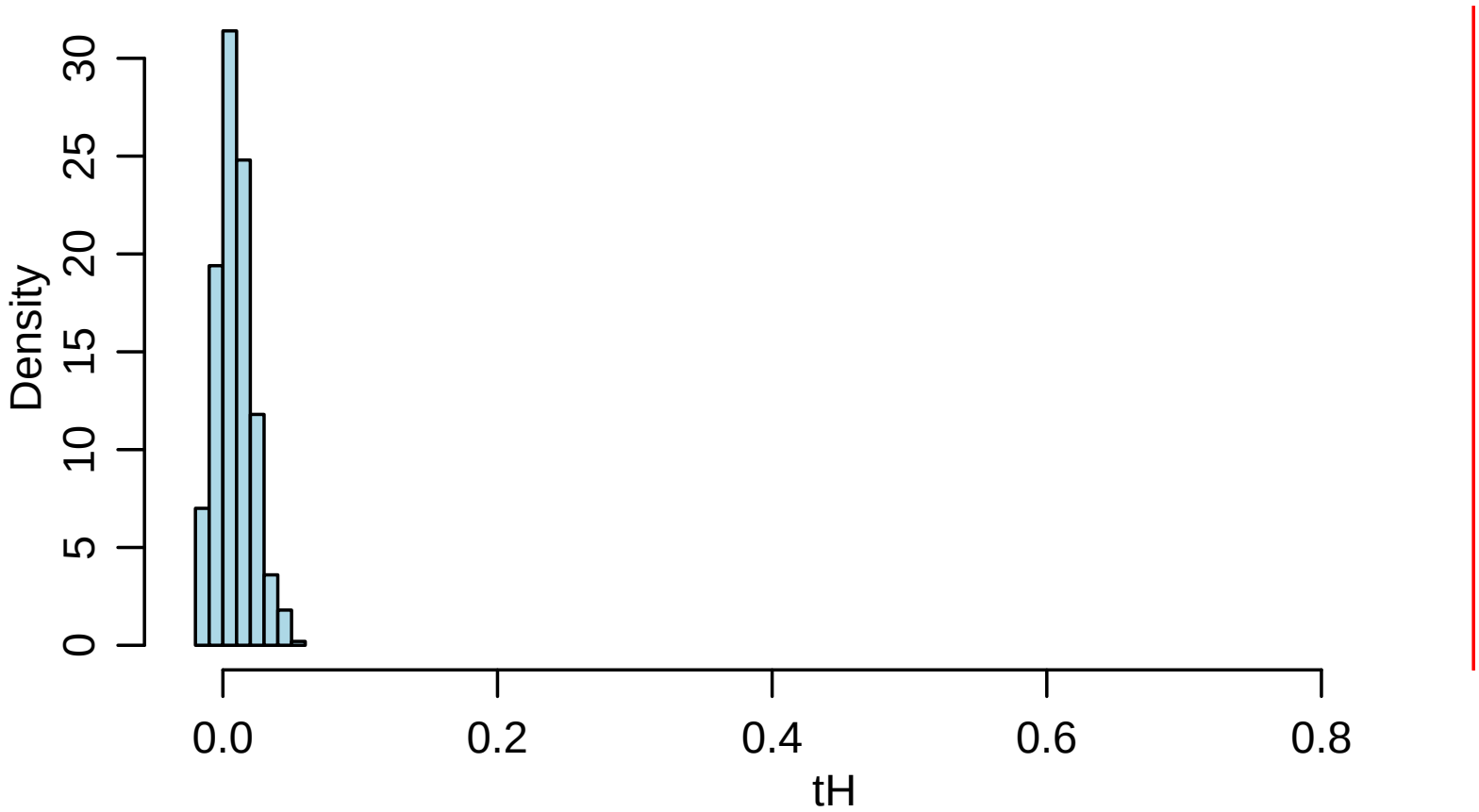
It turns out we can test independence without assuming normality.

Idea: If $\{\epsilon_{i,j}\}$ are independent, then the null distribution of the residuals $\{\epsilon_{i,j}\}$ given $\hat{\boldsymbol{\beta}}$ is (approximately) obtained by sampling from the observed residuals without replacement.

Analogy: Recall under the SRG, the distribution of the graph given the total number of edges was obtained by sampling the edges uniformly without replacement.

```
s.obs<-cor(c(E),c(t(E)),use="complete"  )

s.SIM<-NULL
Esim<-E

for(s in 1:S)
{
  Esim[ !is.na(E) ] <- sample( E[!is.na(E)] )
  s.SIM<-c(s.SIM, cor(c(Esim),c(t(Esim)),use="complete") )
}
```

Permutation tests



We reject the assumption of independence, based on residual dyadic correlation.

Residual diagnostics - row and column heterogeneity

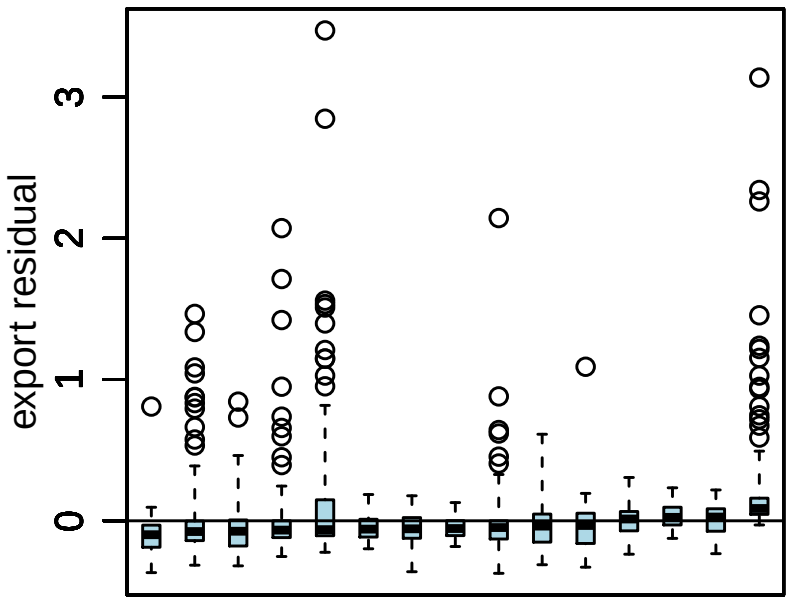
Under independence,

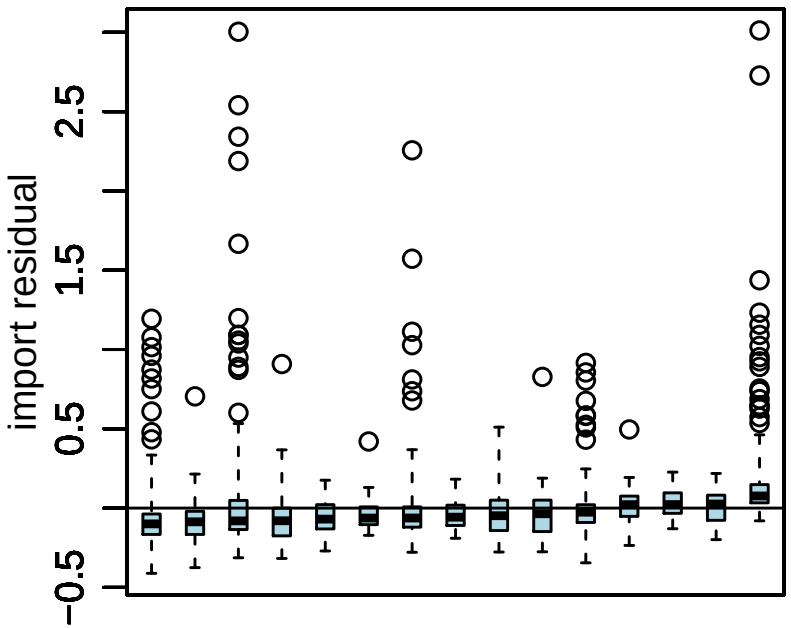
- all residuals should be centered around zero;
- all residuals **within a row** should be centered around zero;
- all residuals **within a column** should be centered around zero.

Failure of the latter two items suggests

- within-row correlation, which is equivalent to
- across-row heterogeneity.

Residual checks





- Some countries have residuals that are mostly below zero;
- Some countries have residuals that are mostly above zero.

Is such a pattern likely under independence?

F statistics for row and column heterogeneity

The standard way to evaluate for systematic row and column variation is with an ANOVA:

$$\epsilon_{i,j} = a_i + b_j + e_{i,j}$$
$$H_R : a_1 = \cdots a_n = 0$$
$$H_C : b_1 = \cdots b_n = 0$$

Under normality, these hypotheses can be evaluated with F-tests:

```
rowidx<-matrix(1:nrow(Y),nrow(Y),nrow(Y))
colidx<-t(rowidx)
anova(lm(c(E)~as.factor(c(rowidx))+as.factor(c(colidx)) ))

## Analysis of Variance Table
##
## Response: c(E)
##                          Df Sum Sq Mean Sq F value    Pr(>F)
## as.factor(c(rowidx))    129 121.90 0.94497  20.990 < 2.2e-16 ***
## as.factor(c(colidx))    129 119.10 0.92328  20.509 < 2.2e-16 ***
## Residuals             16641 749.16 0.04502
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

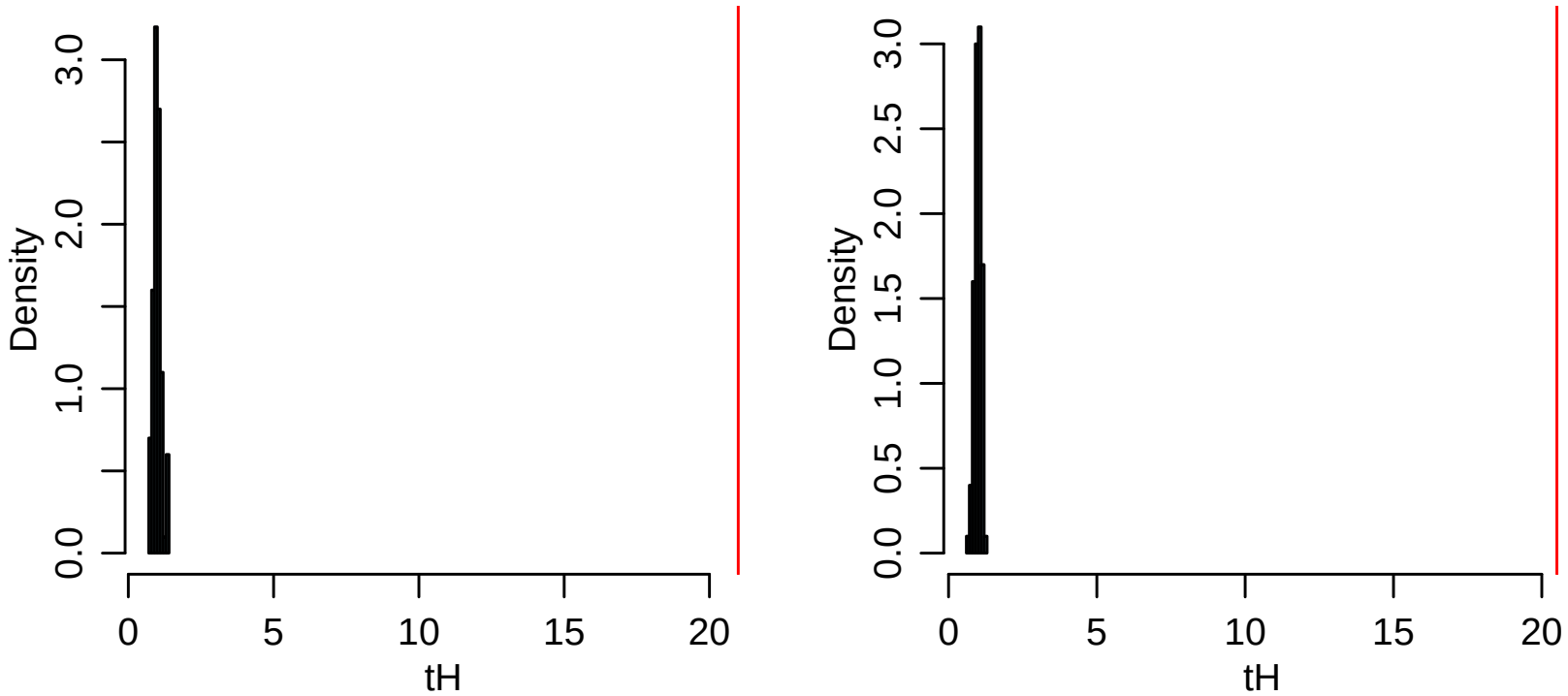
Permutation tests

The validity of the p-values depends on normality.

To evaluate independence without assuming normality, use a permutation test:

```
s.obs<-anova(lm(c(E)~as.factor(c(rowidx))+as.factor(c(colidx)) ))$F[1:2]
s.SIM<-NULL
Esim<-E

for(s in 1:100)
{
  Esim[ !is.na(E) ] <- sample( E[!is.na(E)] )
  s.SIM<-rbind(s.SIM,anova(lm(c(Esim)~as.factor(c(rowidx))+as.factor(c(colidx))))$F[1:2])
}
```

Permutation tests



We reject the assumption of independence, based on the row and column F-statistics.

Row, column and dyadic effects

Consider a normal RCE model of the form

$$y_{i,j} = \boldsymbol{\beta}^T \mathbf{x}_{i,j} + \epsilon_{i,j}$$
$$\epsilon_{i,j} = a_i + b_j + e_{i,j}$$

Our residual diagnostics suggest that

- a_i and b_i might be correlated;
- $e_{i,j}$ and $e_{j,i}$ might be correlated.

Analysis of the covariance of $\{a_i\}$ $\{b_j\}$, $\{e_{i,j}\}$ was studied by Warner, Kenny and Stoto (1979), and their "error decomposition"

$$\epsilon_{i,j} = a_i + b_j + e_{i,j}$$

is called the **social relations model**.

Social relations model

Original SRM:

$$y_{i,j} = \mu + a_i + b_j + e_{i,j}.$$

Original motivation: Decompose variance around μ into parts describing

- heterogeneity across rows means (outdegrees);
- heterogeneity across column means (indegrees);
- correlation between row and column means;
- correlation within dyads.

Note that this is nearly a direct analogue to the p_1 model.

$$\Pr(Y_{i,j} = 1 | Y_{j,i} = 1) = \frac{\exp(\mu + a_i + b_j + \gamma y_{j,i})}{1 + \exp(\mu + a_i + b_j + \gamma y_{j,i})}$$

The social relations model

Random effects version: (Wong [1982], Li and Loken[2002])

$$
\begin{aligned}
y_{i,j} &= \mu + \epsilon_{i,j} \\
\epsilon_{i,j} &= a_i + b_j + e_{i,j} \\
\{(a_1, b_1), \ldots, (a_n, b_n)\} &\sim \text{i.i.d. } N(0, \Sigma_{ab}) \\
\{(e_{i,j}, e_{j,i}) : i \neq j\} &\sim \text{i.i.d. } N(0, \Sigma_e)
\end{aligned}
$$

$$
\Sigma_{ab} = \begin{pmatrix} \sigma_a^2 & \sigma_{ab} \\ \sigma_{ab} & \sigma_b^2 \end{pmatrix} \quad \Sigma_e = \sigma_e^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}
$$

Understanding random effects

What does the SRM imply about the correlation between relational measurements?

Recall,

$$\text{Cov}[U, V] = \text{E}[(U - \mu_U)(V - \mu_V)]$$

Let's compute the covariance between $y_{i,j}$ and $y_{i,k}$, $j \neq k$:

$$\begin{aligned}
\text{Cov}[y_{i,j}, y_{i,k}] &= \text{E}[(y_{i,j} - \mu)(y_{i,k} - \mu)] \\
&= \text{E}[(a_i + b_j + e_{i,j})(a_i + b_k + e_{i,k})] \\
&= \text{E}[a_i^2] + \text{E}[a_i b_k] + \text{E}[a_i e_{i,j}] + \cdots + \text{E}[e_{i,j} e_{i,k}] \\
&= \sigma_a^2
\end{aligned}$$

Covariances in the SRM

Similar calculations give the following non-zero covariances:

$$\begin{aligned}
\text{Cov}[y_{i,j}, y_{i,k}] &= \sigma_a^2 && \text{(within-row covariance)} \\
\text{Cov}[y_{i,j}, y_{k,j}] &= \sigma_b^2 && \text{(within-column covariance)} \\
\text{Cov}[y_{i,j}, y_{j,k}] &= \sigma_{ab} && \text{(row-column covariance)} \\
\text{Cov}[y_{i,j}, y_{j,i}] &= 2\sigma_{ab} + \rho\sigma_e^2 && \text{(row-column covariance plus reciprocity)}
\end{aligned}$$

All other covariances are zero:

$$\text{Cov}[y_{i,j}, y_{k,l}] = 0 \text{ if } i, j, k, l \text{ are distinct}$$

Social relations regression modeling

Just like with the p_1 model, we may want to include regressors in the network dependence model

Regression modeling:

$$\begin{aligned} y_{i,j} &= \beta^T x_{i,j} + a_i + b_j + e_{i,j} \\ \{(a_1, b_1), \ldots, (a_n, b_n)\} &\sim \text{i.i.d. } N(0, \Sigma_{ab}) \\ \{(e_{i,j}, e_{j,i}) : i \neq j\} &\sim \text{i.i.d. } N(0, \Sigma_e) \end{aligned}$$

We call such a model a normal **social relations regression model (SRRM)**.

An alternative formulation

Suppose our regressors can be categorized as sender, receiver and dyadic:

$$\beta^T x_{i,j} = \beta_r^T x_i^r + \beta_c^T x_j^c + \beta_d^T x_{i,j}^d$$

The SRRM can be equivalently formulated as follows:

$$\begin{aligned} y_{i,j} &= \beta_d^T x_{i,j}^d + r_i + c_j + e_{i,j} \\ r_i &= \beta_r^T x_i^r + a_i \\ c_j &= \beta_c^T x_j^c + b_j, \end{aligned}$$

where the variance/covariance of the error terms $a_i, b_j, e_{i,j}$ is as before.

This is exactly the same model as the SRRM, but can assist with interpretation.

Fitting SRRMs

amen-package package:amen R Documentation

Additive and Multiplicative Effects Models for Networks and Relational Data

Description:

Analysis of network and relational data using additive and multiplicative effects (AME) models. The basic model includes regression terms, the covariance structure of the social relations model (Warner, Kenny and Stoto (1979), Wong (1982)), and multiplicative factor effects (Hoff(2009)). Four different link functions accommodate different relational data structures, including binary/network data (bin), normal relational data (nrm), ordinal relational data (ord) and data from fixed-rank nomination schemes (frn). Several of these link functions are discussed in Hoff, Fosdick, Volfovsky and Stovel (2013). Development of this software was supported in part by NICHD grant R01HD067509.

Authors: Peter Hoff, Bailey Fosdick, Alex Volfovsky, Yanjun He

Package amen

```
Usage:

     ame(Y, Xdyad=NULL, Xrow=NULL, Xcol=NULL, rvar = !(model=="rrl") ,
     cvar = TRUE, dcor = TRUE, R = 0, model="nrm",
     intercept=!is.element(model,c("rrl","ord")),
     odmax=rep(max(apply(Y>0,1,sum,na.rm=TRUE)),nrow(Y)), seed = 1, nscan =
     50000, burn = 500, odens = 25, plot=TRUE, print = TRUE, gof=TRUE)

Arguments:

       Y: an n x n square relational matrix of relations. See model
          below for various data types.

   Xdyad: an n x n x pd array of covariates

    Xrow: an n x pr matrix of nodal row covariates

    Xcol: an n x pc matrix of nodal column covariates

    rvar: logical: fit row random effects?

    cvar: logical: fit column random effects?

    dcor: logical: fit a dyadic correlation?
```

Using amen

```
library(amen)
fit_srrm<-ame(Y,X)
```

```
summary(fit_srrm)

##
## beta:
##              pmean   psd z-stat p-val
## intercept -0.601 0.064 -9.379 0.000
## rpop.dyad -0.028 0.016 -1.758 0.079
## rgdp.dyad  0.045 0.014  3.219 0.001
## rpty.dyad -0.002 0.002 -0.792 0.428
## cpop.dyad -0.026 0.015 -1.788 0.074
## cgdp.dyad  0.042 0.014  3.022 0.003
## cpty.dyad -0.001 0.002 -0.505 0.614
## pti.dyad   0.000 0.000 -3.952 0.000
## sigo.dyad  0.015 0.000 41.212 0.000
## dst.dyad   0.013 0.006  2.268 0.023
## con.dyad  -0.001 0.004 -0.228 0.819
##
## Sigma_ab pmean:
##       a     b
## a 0.021 0.013
## b 0.013 0.022
##
## rho pmean:
##  0.883
```

Alternative model specification

```
dim(Xn)

## [1] 130   3

colnames(Xn)

## [1] "pop"    "gdp"    "polity"

dim(Xd)

## [1] 130 130   4

dimnames(Xd)[[3]]

## [1] "polity_int"  "shared_igos" "distance"    "conflicts"
```

```
fit_srrm_drc<-ame(Y,Xdyad=Xd,Xrow=Xn,Xcol=Xn)
```

```
summary(fit_srrm_drc)

##
## beta:
##                    pmean   psd z-stat p-val
## intercept         -0.601 0.064 -9.379 0.000
## pop.row           -0.028 0.016 -1.758 0.079
## gdp.row            0.045 0.014  3.219 0.001
## polity.row        -0.002 0.002 -0.792 0.428
## pop.col           -0.026 0.015 -1.788 0.074
## gdp.col            0.042 0.014  3.022 0.003
## polity.col        -0.001 0.002 -0.505 0.614
## polity_int.dyad    0.000 0.000 -3.952 0.000
## shared_igos.dyad   0.015 0.000 41.212 0.000
## distance.dyad      0.013 0.006  2.268 0.023
## conflicts.dyad    -0.001 0.004 -0.228 0.819
##
## Sigma_ab pmean:
##       a     b
## a 0.021 0.013
## b 0.013 0.022
##
## rho pmean:
##  0.883
```

Comparison to OLS estimates

```
beta_srrm<-apply(fit_srrm$BETA,2,mean)
sd_srrm<-apply(fit_srrm$BETA,2,sd)

beta_ols<- summary(fit_ols)$coef[,1]
sd_ols<-summary(fit_ols)$coef[,2]
```

```
cbind( beta_ols,beta_srrm,beta_ols/beta_srrm)

##                  beta_ols     beta_srrm
## (Intercept) -0.2984079233 -0.6012177123   0.4963392
## Xmrpop      -0.0260558723 -0.0275508090   0.9457389
## Xmrgdp       0.0584609784  0.0447958819   1.3050525
## Xmrpty      -0.0007722494 -0.0019185452   0.4025182
## Xmcpop      -0.0239575625 -0.0259785864   0.9222042
## Xmcgdp       0.0558042060  0.0424384226   1.3149453
## Xmcpty      -0.0001799325 -0.0012544244   0.1434383
## Xmpti        0.0002902070 -0.0002283823  -1.2707074
## Xmsigo       0.0053237026  0.0150056557   0.3547797
## Xmdst       -0.0458066907  0.0127626130  -3.5891311
## Xmcon        0.0343657702 -0.0009969012 -34.4725927
```

Comparison to normal theory sds and z-scores

```
cbind( sd_ols,sd_srrm,sd_ols/sd_srrm)

##                    sd_ols      sd_srrm
## (Intercept) 1.118213e-02 6.410096e-02 0.1744455
## Xmrpop      2.194809e-03 1.567079e-02 0.1400574
## Xmrgdp      1.888985e-03 1.391537e-02 0.1357481
## Xmrpty      3.483605e-04 2.422795e-03 0.1437845
## Xmcpop      2.194458e-03 1.452797e-02 0.1510506
## Xmcgdp      1.888935e-03 1.404442e-02 0.1344971
## Xmcpty      3.483234e-04 2.484017e-03 0.1402258
## Xmpti       4.520667e-05 5.779421e-05 0.7822007
## Xmsigo      1.979838e-04 3.641084e-04 0.5437497
## Xmdst       3.805497e-03 5.626285e-03 0.6763784
## Xmcon       9.274728e-03 4.363038e-03 2.1257502

cbind( beta_ols/sd_ols,beta_srrm/sd_srrm)

##                     [,1]       [,2]
## (Intercept) -26.6861550 -9.3792316
## Xmrpop      -11.8715876 -1.7580997
## Xmrgdp       30.9483598  3.2191662
## Xmrpty       -2.2168111 -0.7918726
## Xmcpop      -10.9173040 -1.7881777
## Xmcgdp       29.5426872  3.0217277
## Xmcpty       -0.5165674 -0.5049984
## Xmpti         6.4195622 -3.9516464
## Xmsigo       26.8895835 41.2120543
## Xmdst       -12.0369790  2.2683909
## Xmcon         3.7053128 -0.2284879
```

Bayesian inference and estimation

Estimates and standard errors in `amen` are obtained using **Bayesian inference**

Bayesian inference is closely related to **maximum likelihood inference**, with a few distinctions:

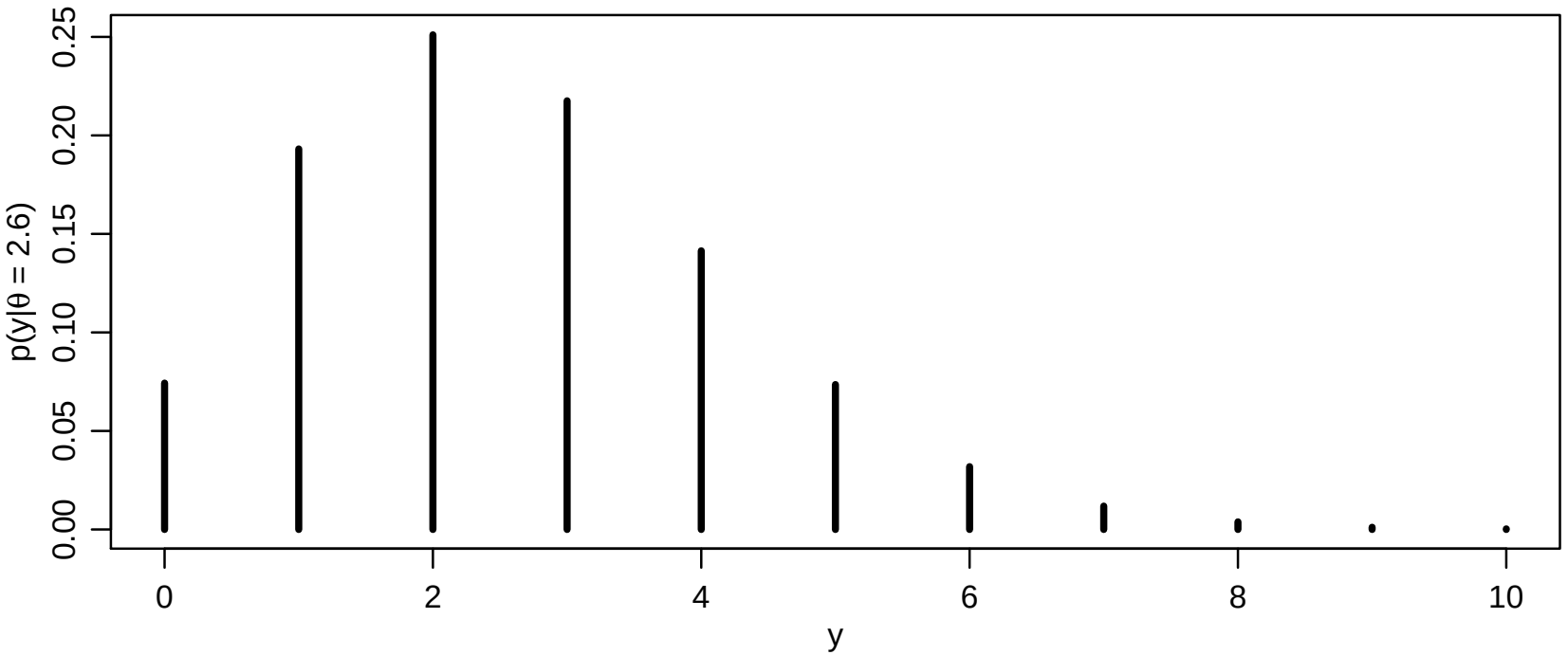
- In Bayesian inference, probability is treated as a measure of uncertainty;
- Bayesian inference combines
  - **data information** with
  - **prior information** to produce
  - **posterior information**.
- For many problems, Bayes and ML inference produce similar results;
- Computational methods for Bayes inference are more universally available.

A simple example

Scenario: A nation-wide survey will ask people about the number of friends they socialize with on average per week. The number of friends per person is assumed to follow a Poisson distribution with mean θ:

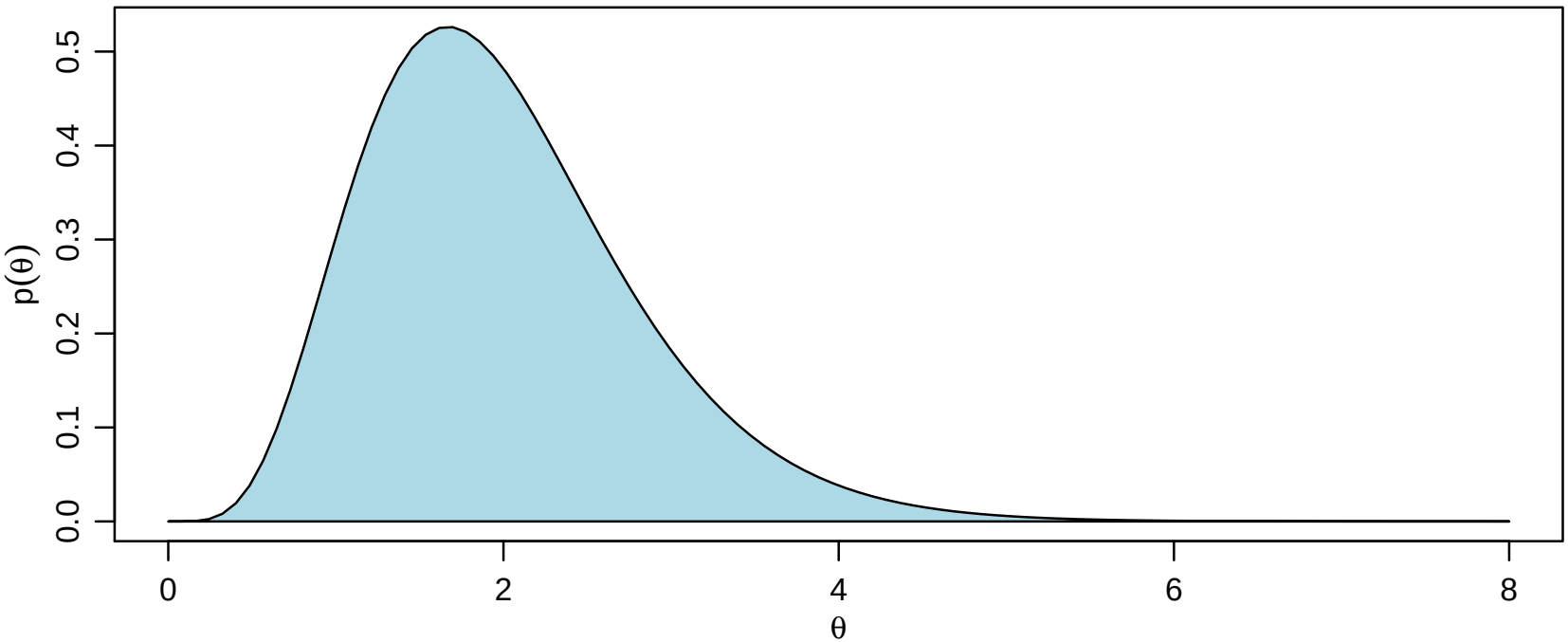
$$Y_1, \ldots, Y_n \sim \text{i.i.d. Poisson}(\theta).$$

Researchers observe values for $Y_1, \ldots, Y_n$, and then wish to obtain an estimate of and confidence interval for θ.



Prior information

Prior information: Researchers have been studying friendship data for a long time, and have some pretty good guesses about what the value of θ is. They summarize this information with a **prior distribution**.



Under this prior distribution $\mathrm{E}[\theta] = 2$.

Posterior information

After having observed $\{Y_1 = y_1, \ldots, Y_n = y_n\}$ The researchers combine their prior information with their data information by computing the

conditional distribution of θ, given $\{y_1, \ldots, y_n\}$,

also known as

the posterior distribution of θ.

Conditional probability

First consider

- $A = \{Y_1 = y_1, \ldots, Y_n = y_n\}$, the observed data;
- $B = \{\theta < 3\}$ a statement about the unknown parameter.

Recall Bayes' rule,

$$\begin{aligned}\Pr(B|A) &= \frac{\Pr(A|B)\Pr(B)}{\Pr(A)} \\ &= \frac{\Pr(A|B)\Pr(B)}{\Pr(A|B)\Pr(B) + \Pr(A|B^c)\Pr(B^c)}\end{aligned}$$

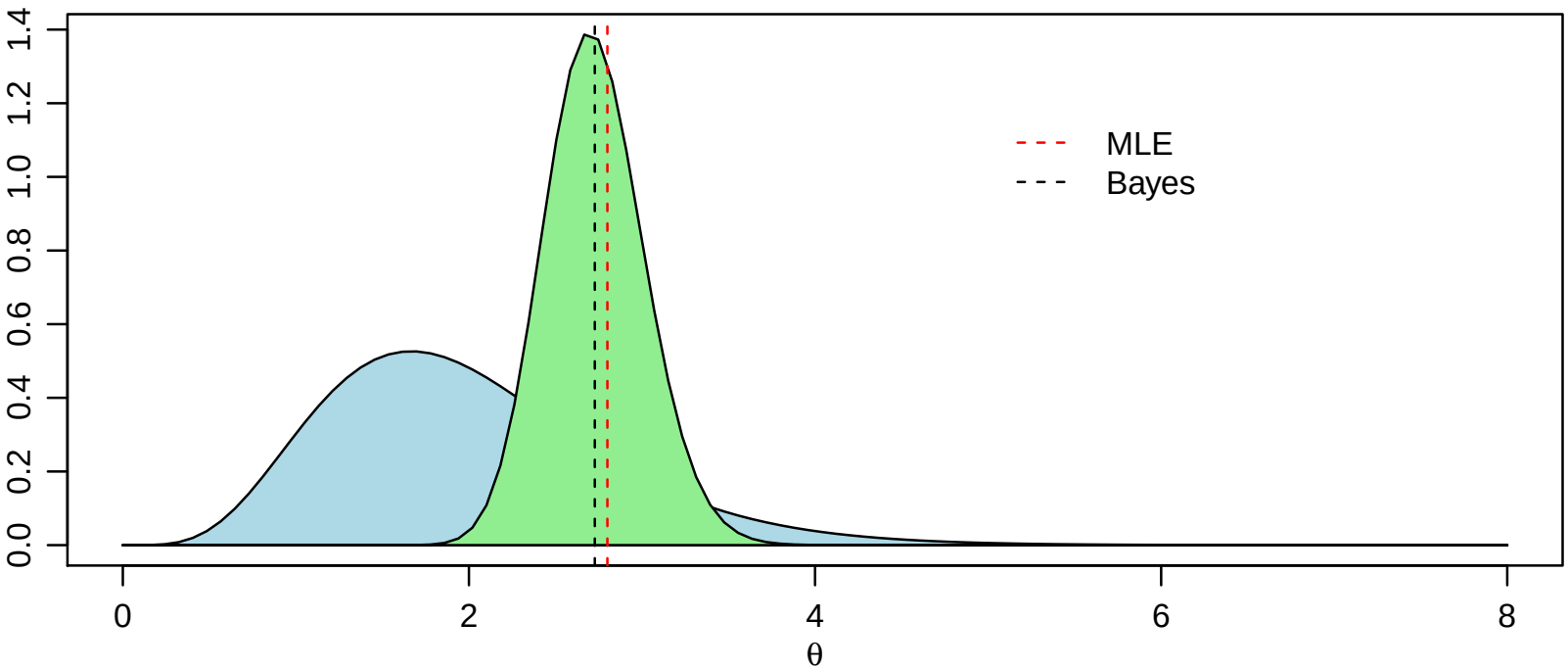
Conditional probability distribution

More generally, the posterior distribution $p(\theta|y_1, \ldots, y_n)$ may be found using a version of Bayes rule. Letting $\mathbf{y} = (y_1, \ldots, y_n)$, we have

$$p(\theta|\mathbf{y}) = \frac{p(\mathbf{y}|\theta)p(\theta)}{\int p(\mathbf{y}|\theta)p(\theta)d\theta}.$$

Data and posterior distribution

Data: Suppose $n = 30$ and $\bar{y} = 2.8$. The MLE is $\hat{\theta}_{\text{MLE}} = \bar{y} = 2.8$. What is the posterior distribution and Bayes estimator?



Given the prior and these data, $\text{E}[\theta|\mathbf{y}] = 2.72$.

Posterior inference via Monte Carlo

Inference cannot always be obtained from a simple plot of the posterior.

- often the posterior distribution is a very high-dimensional object;
- often we can't compute the posterior distribution exactly.

However, for a large number of models we can **simulate** from $p(\theta|\mathbf{y})$.

Simulations $\theta^{(1)}, \ldots, \theta^{(S)}$ from $p(\theta|\mathbf{y})$ can be used to approximate $p(\theta|\mathbf{y})$.

This is known as the **Monte Carlo method**.

Monte Carlo approximation

Posterior descriptions involve integrals we'd often like to avoid:

$$\mathrm{E}[\theta|y] = \int \theta p(\theta|y)\ d\theta$$

$$\text{median}[\theta|y] = \theta_{1/2} : \int_{-\infty}^{\theta_{1/2}} p(\theta|y) = 1/2$$

$$\Pr(\theta_1 < \theta_2|y_1, y_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\theta_2} p(\theta_1, \theta_2|y_1, y_2)\ d\theta_1\ d\theta_2$$

We can easily approximate such integrals arbitrarily closely with **Monte Carlo approximation**.

Monte Carlo approximation

The basic principle: If $\theta^{(1)}, \ldots, \theta^{(S)}$ iid $p(\theta|y)$, then

$$\text{histogram}\{\theta^{(1)}, \ldots, \theta^{(S)}\} \approx p(\theta|y)$$

This implies that

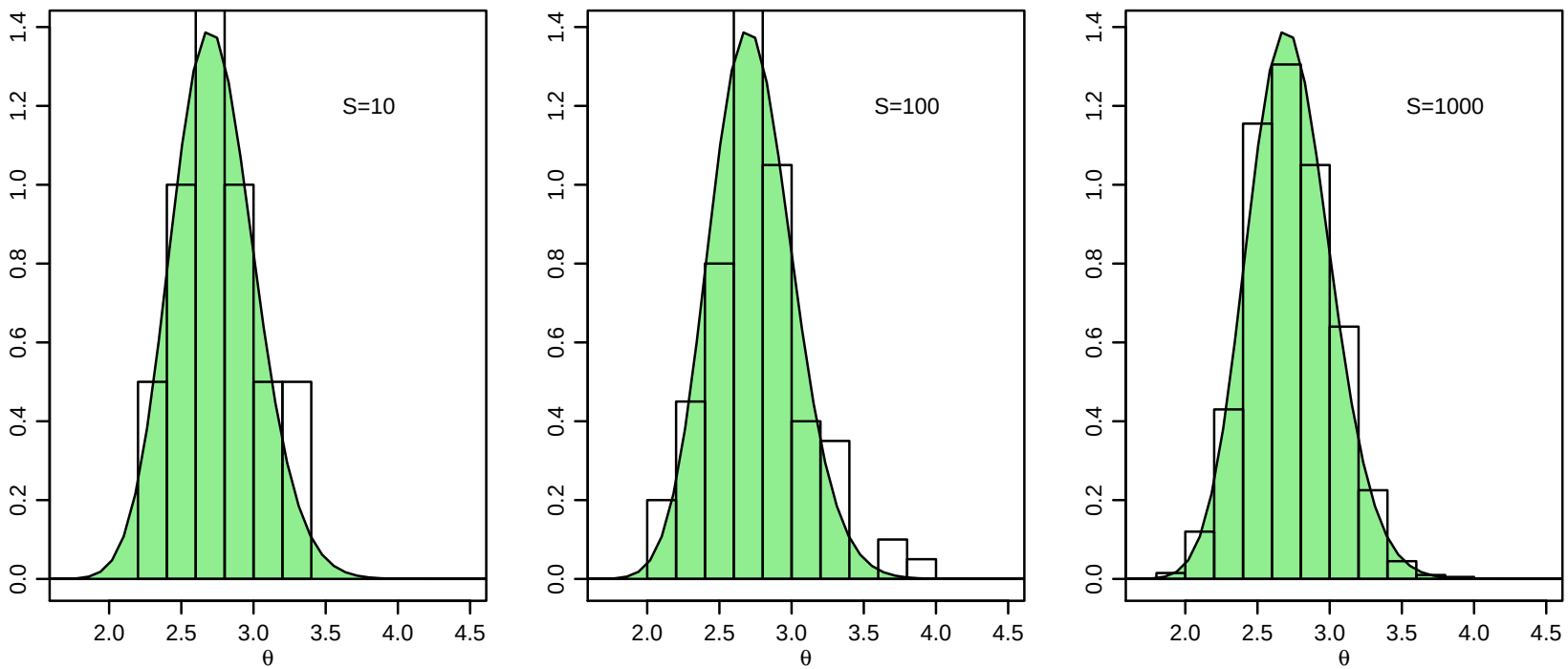
$$\frac{1}{S}\sum \theta^{(s)} \approx \text{E}[\theta|y]$$

$$\frac{\#\{\theta^{(s)} < c\}}{S} \approx \text{Pr}(\theta < c|y)$$

$$\text{median}\{\theta^{(1)}, \ldots, \theta^{(S)}\} \approx \text{median}[\theta|y]$$

etc. with the approximation improving with increasing S.

Monte Carlo approximation



Quick summary of Bayesian inference

- Probability distributions encapsulate information
  - $p(\theta)$ describes prior information
  - $p(y|\theta)$ describes information about y for each θ
  - $p(\theta|y)$ describes posterior information
- Posterior distributions can be calculated via Bayes' rule

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)\ d\theta}$$

- For some models, posteriors can be plotted/summarized with R commands
  - dbeta, dgamma, dnorm
  - qbeta, qgamma, qnorm
- Posterior distributions can be approximated with Monte Carlo methods.
  1. simulate $\theta^{(1)}, \ldots, \theta^{(S)}$ iid $p(\theta|y)$
  2. posterior means, variances, quantiles can be approximated with simulation means, variances, quantiles.

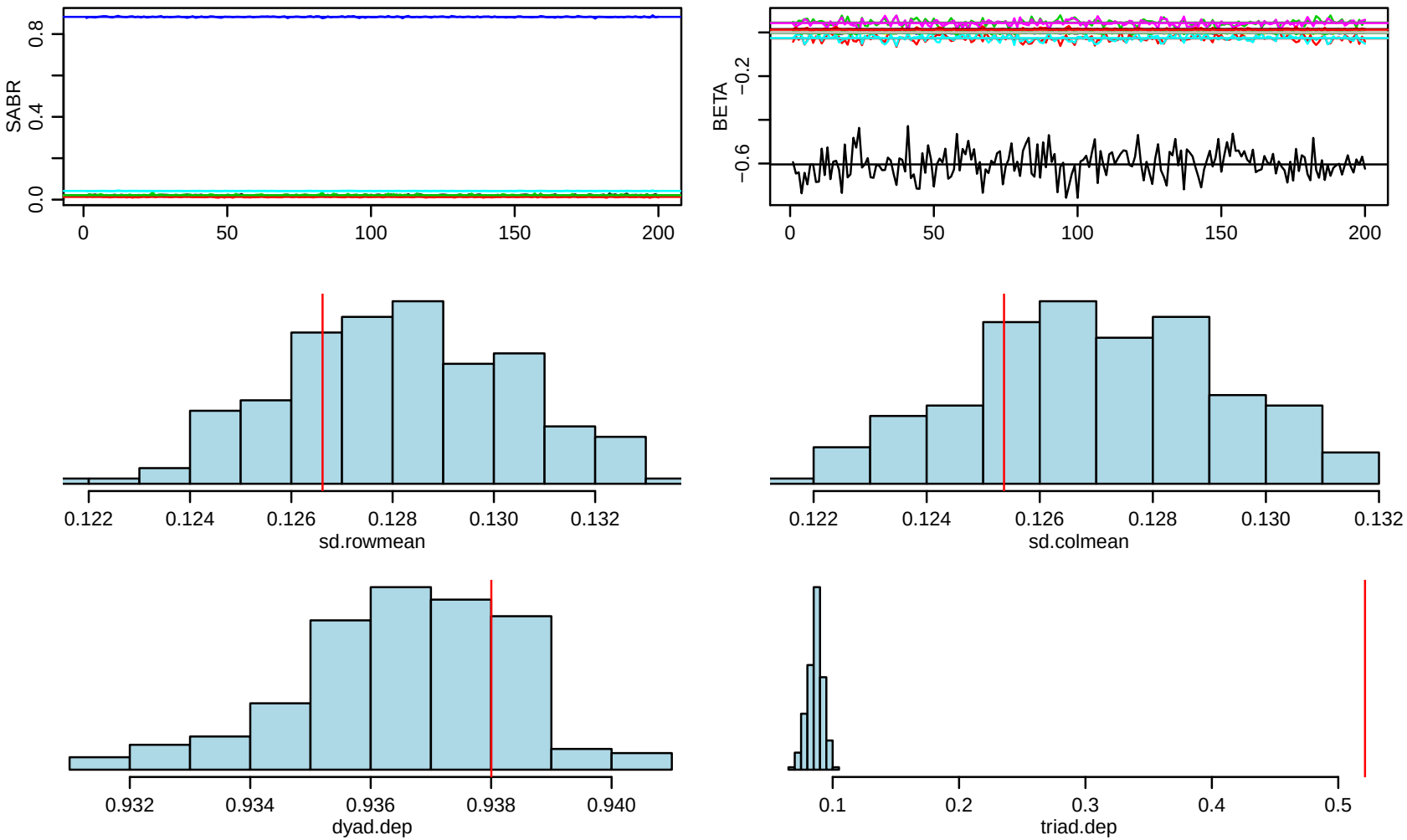
Bayesian inference in amen

The amen package generates a Monte Carlo approximation to the posterior distribution of the parameters in the SRM:

- Regression parameters: $\boldsymbol{\beta}$
- Row and column effects: $\boldsymbol{a}$, $\boldsymbol{b}$
- Covariance parameters: $(\sigma_a^2, \sigma_b^2, \sigma_{ab})$, (σ_e^2, ρ).

Bayesian inference in amen

```
plot(fit_srrm)
```



Posterior inference via Monte Carlo

```
fit_srrm$SABR[1:12,]

##                va         cab          vb       rho          ve
##  [1,] 0.01967634 0.01341578 0.02066698 0.8777030 0.04098923
##  [2,] 0.02499210 0.01627907 0.02443084 0.8831558 0.04123798
##  [3,] 0.02170353 0.01406014 0.02303109 0.8816819 0.04195064
##  [4,] 0.02479180 0.01503979 0.02147478 0.8798281 0.04125919
##  [5,] 0.02089989 0.01529636 0.02267957 0.8832281 0.04190235
##  [6,] 0.02173060 0.01499526 0.02650665 0.8848382 0.04202978
##  [7,] 0.01971525 0.01311672 0.02187827 0.8866916 0.04232291
##  [8,] 0.02227328 0.01332624 0.02052721 0.8861467 0.04217969
##  [9,] 0.01936770 0.01080374 0.01735727 0.8823997 0.04125212
## [10,] 0.02184757 0.01174441 0.01739378 0.8797601 0.04107131
## [11,] 0.02339106 0.01554288 0.02186145 0.8845296 0.04210824
## [12,] 0.01743037 0.01190051 0.02137431 0.8881189 0.04416590
```

These numbers represent simulations from $p(\sigma_a^2, \sigma_b^2, \sigma_{ab}, \sigma_e^2, \rho | \mathbf{Y}, \mathbf{X})$.

Consider addressing the following inferential questions about ρ:

- What is a point estimate of ρ?
- What is the uncertainty in ρ?
- What is a 95% confidence interval for ρ?

Posterior inference via Monte Carlo

```
rho_mcmc<-fit_srrm$SABR[,4]
```

- What is a point estimate of ρ? $\mathrm{E}[\rho|\mathbf{Y},\mathbf{X}]$

```
mean(rho_mcmc)

## [1] 0.8829668
```

- What is our uncertainty in ρ? $\mathrm{sd}(\rho|\mathbf{Y},\mathbf{X})$

```
sd(rho_mcmc)

## [1] 0.002454778
```

- What is a 95% confidence interval for ρ? $(\rho_{.025}, \rho_{.975})$

```
quantile(rho_mcmc,prob=c(.025,.975))

##      2.5%     97.5%
## 0.8776932 0.8874168
```