Row and column effects
567 Statistical analysis of social networks

Peter Hoff

Statistics, University of Washington
Monk data

\[ \text{sd}(rsum(Y)) \]

\#

\[ [1] 0.9002541 \]

\[ \text{sd}(csum(Y)) \]

\#

\[ [1] 2.698341 \]
Testing the SRG

rERG <- function(n, s)
{
  Y <- matrix(0, n, n) ; diag(Y) <- NA
  Y[!is.na(Y)] <- sample( c(rep(1, s), rep(0, n*(n-1) - s)))
  Y
}

n <- nrow(Y)
s <- sum(Y, na.rm=TRUE)

sdo <- sd(rsum(Y))
sdi <- sd(csum(Y))

DSD <- NULL
for(sim in 1:S)
{
  Ysim <- rERG(n, s)
  DSD <- rbind(DSD, c(sd(rsum(Ysim)), sd(csum(Ysim)))
}
Testing the SRG
Testing the SRG

Compared to any SRG,

- the nodes show less heterogeneity in outdegree;
- the nodes show more heterogeneity in indegree.

No SRG model provides a good representation of $Y$ in terms of its degrees.

What kind of model would provide a better representation of the data?
Compared to any SRG,

- the nodes show less heterogeneity in outdegree;
- the nodes show more heterogeneity in indegree.

No SRG model provides a good representation of $Y$ in terms of its degrees.

What kind of model would provide a better representation of the data?
Testing the SRG

Compared to any SRG,

- the nodes show less heterogeneity in outdegree;
- the nodes show more heterogeneity in indegree.

No SRG model provides a good representation of $\mathbf{Y}$ in terms of its degrees.

What kind of model would provide a better representation of the data?
Testing the SRG

Compared to any SRG,

- the nodes show less heterogeneity in outdegree;
- the nodes show more heterogeneity in indegree.

No SRG model provides a good representation of $Y$ in terms of its degrees.

What kind of model would provide a better representation of the data?
Testing the SRG

Compared to any SRG,

- the nodes show less heterogeneity in outdegree;
- the nodes show more heterogeneity in indegree.

No SRG model provides a good representation of $Y$ in terms of its degrees.

What kind of model would provide a better representation of the data?
ANOVA for factorial data

Consider the data matrix for a generic two-factor layout:

\[
\begin{array}{c|cccc}
 & 1 & 2 & \cdots & K_2 \\
1 & y_{1,1} & y_{1,2} & \cdots & y_{1,K_2} \\
2 & y_{2,1} & y_{2,2} & \cdots & y_{2,K_2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
K_1 & y_{K_1,1} & y_{K_1,2} & \cdots & y_{K_1,K_2} \\
\end{array}
\]

The standard “ANOVA” row and column effects (RCE) model for such data is

\[y_{i,j} = \mu + a_i + b_j + \epsilon_{i,j}\]

- \(\mu\) represents an overall mean;
- \(a_i\) represents the average deviation of row \(i\) from \(\mu\);
- \(b_j\) represents the average deviation of column \(j\) from \(\mu\).

More abstractly,
- \(a_i\) is the “effect” of factor one having level \(i\);
- \(b_j\) is the “effect” of factor two being level \(j\).
ANOVA for factorial data

Consider the data matrix for a generic two-factor layout:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>⋯</th>
<th>$K_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$y_{1,1}$</td>
<td>$y_{1,2}$</td>
<td>⋯</td>
<td>$y_{1,K_2}$</td>
</tr>
<tr>
<td>2</td>
<td>$y_{2,1}$</td>
<td>$y_{2,2}$</td>
<td>⋯</td>
<td>$y_{2,K_2}$</td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
</tr>
<tr>
<td>$K_1$</td>
<td>$y_{K_1,1}$</td>
<td>$y_{K_1,2}$</td>
<td>⋯</td>
<td>$y_{K_1,K_2}$</td>
</tr>
</tbody>
</table>

The standard “ANOVA” row and column effects (RCE) model for such data is

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

- $\mu$ represents an overall mean;
- $a_i$ represents the average deviation of row $i$ from $\mu$;
- $b_j$ represents the average deviation of column $j$ from $\mu$.

More abstractly,
- $a_i$ is the “effect” of factor one having level $i$;
- $b_j$ is the “effect” of factor two being level $j$. 
ANOVA for factorial data

Consider the data matrix for a generic two-factor layout:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>...</th>
<th>$K_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$y_{1,1}$</td>
<td>$y_{1,2}$</td>
<td>...</td>
<td>$y_{1,K_2}$</td>
</tr>
<tr>
<td>2</td>
<td>$y_{2,1}$</td>
<td>$y_{2,2}$</td>
<td>...</td>
<td>$y_{2,K_2}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>$K_1$</td>
<td>$y_{K_1,1}$</td>
<td>$y_{K_1,2}$</td>
<td>...</td>
<td>$y_{K_1,K_2}$</td>
</tr>
</tbody>
</table>

The standard “ANOVA” row and column effects (RCE) model for such data is

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

- $\mu$ represents an overall mean;
- $a_i$ represents the average deviation of row $i$ from $\mu$;
- $b_j$ represents the average deviation of column $j$ from $\mu$.

More abstractly,
- $a_i$ is the “effect” of factor one having level $i$;
- $b_j$ is the “effect” of factor two being level $j$. 
ANOVA for factorial data

Consider the data matrix for a generic two-factor layout:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>⋯</th>
<th>$K_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$y_{1,1}$</td>
<td>$y_{1,2}$</td>
<td>⋯</td>
<td>$y_{1,K_2}$</td>
</tr>
<tr>
<td>2</td>
<td>$y_{2,1}$</td>
<td>$y_{2,2}$</td>
<td>⋯</td>
<td>$y_{2,K_2}$</td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
</tr>
<tr>
<td>$K_1$</td>
<td>$y_{K_1,1}$</td>
<td>$y_{K_1,2}$</td>
<td>⋯</td>
<td>$y_{K_1,K_2}$</td>
</tr>
</tbody>
</table>

The standard “ANOVA” row and column effects (RCE) model for such data is

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

- $\mu$ represents an overall mean;
- $a_i$ represents the average deviation of row $i$ from $\mu$;
- $b_j$ represents the average deviation of column $j$ from $\mu$.

More abstractly,

- $a_i$ is the “effect” of factor one having level $i$;
- $b_j$ is the “effect” of factor two being level $j$. 

6/36
ANOVA for factorial data

Consider the data matrix for a generic two-factor layout:

\[
\begin{array}{c|cccc}
 & 1 & 2 & \cdots & K_2 \\
1 & y_{1,1} & y_{1,2} & \cdots & y_{1,K_2} \\
2 & y_{2,1} & y_{2,2} & \cdots & y_{2,K_2} \\
\vdots & \vdots & \ddots & & \ddots \\
K_1 & y_{K_1,1} & y_{K_1,2} & \cdots & y_{K_1,K_2} \\
\end{array}
\]

The standard “ANOVA” row and column effects (RCE) model for such data is

\[y_{i,j} = \mu + a_i + b_j + \epsilon_{i,j}\]

- \(\mu\) represents an overall mean;
- \(a_i\) represents the average deviation of row \(i\) from \(\mu\);
- \(b_j\) represents the average deviation of column \(j\) from \(\mu\).

More abstractly,

- \(a_i\) is the “effect” of factor one having level \(i\);
- \(b_j\) is the “effect” of factor two being level \(j\).
ANOVA for factorial data

Consider the data matrix for a generic two-factor layout:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>...</th>
<th>$K_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$y_{1,1}$</td>
<td>$y_{1,2}$</td>
<td>...</td>
<td>$y_{1,K_2}$</td>
</tr>
<tr>
<td>2</td>
<td>$y_{2,1}$</td>
<td>$y_{2,2}$</td>
<td>...</td>
<td>$y_{2,K_2}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$K_1$</td>
<td>$y_{K_1,1}$</td>
<td>$y_{K_1,2}$</td>
<td>...</td>
<td>$y_{K_1,K_2}$</td>
</tr>
</tbody>
</table>

The standard “ANOVA” row and column effects (RCE) model for such data is

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

- $\mu$ represents an overall mean;
- $a_i$ represents the average deviation of row $i$ from $\mu$;
- $b_j$ represents the average deviation of column $j$ from $\mu$.

More abstractly,
- $a_i$ is the “effect” of factor one having level $i$;
- $b_j$ is the “effect” of factor two being level $j$. 
ANOVA for factorial data

Consider the data matrix for a generic two-factor layout:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>⋯</th>
<th>$K_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$y_{1,1}$</td>
<td>$y_{1,2}$</td>
<td>⋯</td>
<td>$y_{1,K_2}$</td>
</tr>
<tr>
<td>2</td>
<td>$y_{2,1}$</td>
<td>$y_{2,2}$</td>
<td>⋯</td>
<td>$y_{2,K_2}$</td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
</tr>
<tr>
<td>$K_1$</td>
<td>$y_{K_1,1}$</td>
<td>$y_{K_1,2}$</td>
<td>⋯</td>
<td>$y_{K_1,K_2}$</td>
</tr>
</tbody>
</table>

The standard “ANOVA” row and column effects (RCE) model for such data is

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

- $\mu$ represents an overall mean;
- $a_i$ represents the average deviation of row $i$ from $\mu$;
- $b_j$ represents the average deviation of column $j$ from $\mu$.

More abstractly,

- $a_i$ is the “effect” of factor one having level $i$;
- $b_j$ is the “effect” of factor two being level $j$. 
ANOVA for factorial data

Consider the data matrix for a generic two-factor layout:

\[
\begin{array}{cccc}
1 & 2 & \cdots & K_2 \\
1 & y_{1,1} & y_{1,2} & \cdots & y_{1,K_2} \\
2 & y_{2,1} & y_{2,2} & \cdots & y_{2,K_2} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
K_1 & y_{K_1,1} & y_{K_1,2} & \cdots & y_{K_1,K_2} \\
\end{array}
\]

The standard “ANOVA” row and column effects (RCE) model for such data is

\[y_{i,j} = \mu + a_i + b_j + \epsilon_{i,j}\]

- \(\mu\) represents an overall mean;
- \(a_i\) represents the average deviation of row \(i\) from \(\mu\);
- \(b_j\) represents the average deviation of column \(j\) from \(\mu\).

More abstractly,

- \(a_i\) is the “effect” of factor one having level \(i\);
- \(b_j\) is the “effect” of factor two being level \(j\).
ANOVA for factorial data

Consider the data matrix for a generic two-factor layout:

\[
\begin{array}{cccc}
 & 1 & 2 & \cdots & K_2 \\
1 & y_{1,1} & y_{1,2} & \cdots & y_{1,K_2} \\
2 & y_{2,1} & y_{2,2} & \cdots & y_{2,K_2} \\
\vdots & & & & \\
K_1 & y_{K_1,1} & y_{K_1,2} & \cdots & y_{K_1,K_2} \\
\end{array}
\]

The standard “ANOVA” row and column effects (RCE) model for such data is

\[
y_{i,j} = \mu + a_i + b_j + \epsilon_{i,j}
\]

- \(\mu\) represents an overall mean;
- \(a_i\) represents the average deviation of row \(i\) from \(\mu\);
- \(b_j\) represents the average deviation of column \(j\) from \(\mu\).

More abstractly,

- \(a_i\) is the “effect” of factor one having level \(i\);
- \(b_j\) is the “effect” of factor two being level \(j\).
ANOVA for factorial designs

ML and OLS estimates are given by

- \( \hat{\mu} = \bar{y} \cdot \);
- \( \hat{a}_i = \bar{y}_{i.} - \bar{y} \cdot \);
- \( \hat{b}_j = \bar{y}_{.j} - \bar{y} \cdot \).

Note that

\[
\text{mean}(\hat{a}_1, \ldots, \hat{a}_{K_1}) = \frac{1}{K_1} \sum (\bar{y}_{i.} - \bar{y} \cdot) = \frac{1}{K_1} \sum \bar{y}_{i.} - \bar{y} \cdot = \frac{1}{K_1 K_2} \sum \sum y_{i,j} - \bar{y} \cdot = \bar{y} \cdot - \bar{y} \cdot = 0.
\]

\[
\text{var} (\hat{a}_1, \ldots, \hat{a}_{K_1}) = \text{var}(\bar{y}_{1.}, \ldots, \bar{y}_{K_1}).
\]
ANOVA for factorial designs

ML and OLS estimates are given by

- $\hat{\mu} = \bar{y}..$
- $\hat{a}_i = \bar{y}_{i.} - \bar{y}..$
- $\hat{b}_j = \bar{y}_{.j} - \bar{y}..$

Note that

$$\text{mean}(\hat{a}_1, \ldots, \hat{a}_{K_1}) = \frac{1}{K_1} \sum (\bar{y}_{i.} - \bar{y}..)$$

$$= \frac{1}{K_1} \sum \bar{y}_{i.} - \bar{y}..$$

$$= \frac{1}{K_1 K_2} \sum \sum y_{i,j} - \bar{y}..$$

$$= \bar{y}.. - \bar{y}.. = 0.$$
ANOVA for factorial designs

ML and OLS estimates are given by

- $\hat{\mu} = \bar{y}..$;
- $\hat{a}_i = \bar{y}_{i.} - \bar{y}..$;
- $\hat{b}_j = \bar{y}_{.j} - \bar{y}..$.

Note that

$$\text{mean}(\hat{a}_1, \ldots, \hat{a}_{K_1}) = \frac{1}{K_1} \sum (\bar{y}_{i.} - \bar{y}..) = \frac{1}{K_1} \sum \bar{y}_{i.} - \bar{y}.. = \frac{1}{K_1 K_2} \sum \sum y_{i,j} - \bar{y}.. = \bar{y}.. - \bar{y}.. = 0.$$  

$$\text{var} (\hat{a}_1, \ldots, \hat{a}_{K_1}) = \text{var}(\bar{y}_{1.}, \ldots, \bar{y}_{K_1.}).$$
ANOVA for factorial designs

ML and OLS estimates are given by

- $\hat{\mu} = \bar{y}..$
- $\hat{a}_i = \bar{y}_{i.} - \bar{y}..$
- $\hat{b}_j = \bar{y}_{.j} - \bar{y}..$

Note that

$$\text{mean}(\hat{a}_1, \ldots, \hat{a}_{K_1}) = \frac{1}{K_1} \sum (\bar{y}_{i.} - \bar{y}..) = \frac{1}{K_1} \sum \bar{y}_{i.} - \bar{y}.. = \frac{1}{K_1 K_2} \sum \sum y_{i,j} - \bar{y}.. = \bar{y}.. - \bar{y}.. = 0.$$  

$$\text{var} (\hat{a}_1, \ldots, \hat{a}_{K_1}) = \text{var}(\bar{y}_{1.}, \ldots, \bar{y}_{K_1}).$$
ANOVA for factorial designs

ML and OLS estimates are given by

- $\hat{\mu} = \bar{y}..$
- $\hat{a}_i = \bar{y}_{i.} - \bar{y}..$
- $\hat{b}_j = \bar{y}_{..j} - \bar{y}..$

Note that

$$\text{mean}(\hat{a}_1, \ldots, \hat{a}_{K_1}) = \frac{1}{K_1} \sum (\bar{y}_{i.} - \bar{y}..)$$

$$= \frac{1}{K_1} \sum \bar{y}_{i.} - \bar{y}..$$

$$= \frac{1}{K_1 K_2} \sum \sum y_{i,j} - \bar{y}..$$

$$= \bar{y}.. - \bar{y}.. = 0.$$
For the RCE model

- the model is parameterized in terms of differences among row means and differences among column means;
- heterogeneity in the observed row/column means is represented by heterogeneity in the $\hat{a}_i$'s and $\hat{b}_j$'s;
- tests for row and column heterogeneity can be made via $F$-tests.

Back to binary directed network data:

- the data take the form of a two-way layout;
- we've seen empirically the existence of row and column heterogeneity.

Is there an analogous RCE model for network data?
For the RCE model

- the model is parameterized in terms of differences among row means and differences among column means;
- heterogeneity in the observed row/column means is represented by heterogeneity in the $\hat{a}_i$’s and $\hat{b}_j$’s;
- tests for row and column heterogeneity can be made via $F$-tests.

Back to binary directed network data:

- the data take the form of a two-way layout;
- we’ve seen empirically the existence of row and column heterogeneity.

Is there an analogous RCE model for network data?
For the RCE model

- the model is parameterized in terms of differences among row means and differences among column means;
- heterogeneity in the observed row/column means is represented by heterogeneity in the $\hat{a}_i$’s and $\hat{b}_j$’s;
- tests for row and column heterogeneity can be made via $F$-tests.

Back to binary directed network data:

- the data take the form of a two-way layout;
- we've seen empirically the existence of row and column heterogeneity.

Is there an analogous RCE model for network data?
RCE model interpretation

For the RCE model

- the model is parameterized in terms of differences among row means and differences among column means;
- heterogeneity in the observed row/column means is represented by heterogeneity in the $\hat{a}_i$’s and $\hat{b}_j$’s;
- tests for row and column heterogeneity can be made via $F$-tests.

Back to binary directed network data:

- the data take the form of a two-way layout;
- we’ve seen empirically the existence of row and column heterogeneity.

Is there an analogous RCE model for network data?
For the RCE model

- the model is parameterized in terms of differences among row means and differences among column means;
- heterogeneity in the observed row/column means is represented by heterogeneity in the $\hat{a}_i$’s and $\hat{b}_j$’s;
- tests for row and column heterogeneity can be made via $F$-tests.

Back to binary directed network data:

- the data take the form of a two-way layout;
- we’ve seen empirically the existence of row and column heterogeneity.

Is there an analogous RCE model for network data?
For the RCE model
- the model is parameterized in terms of differences among row means and differences among column means;
- heterogeneity in the observed row/column means is represented by heterogeneity in the $\hat{a}_i$’s and $\hat{b}_j$’s;
- tests for row and column heterogeneity can be made via $F$-tests.

Back to binary directed network data:
- the data take the form of a two-way layout;
- we’ve seen empirically the existence of row and column heterogeneity.

Is there an analogous RCE model for network data?
For the RCE model

- the model is parameterized in terms of differences among row means and differences among column means;

- heterogeneity in the observed row/column means is represented by heterogeneity in the $\hat{a}_i$’s and $\hat{b}_j$’s;

- tests for row and column heterogeneity can be made via $F$-tests.

Back to binary directed network data:

- the data take the form of a two-way layout;

- we’ve seen empirically the existence of row and column heterogeneity.

Is there an analogous RCE model for network data?
For the RCE model

- the model is parameterized in terms of differences among row means and differences among column means;
- heterogeneity in the observed row/column means is represented by heterogeneity in the $\hat{a}_i$’s and $\hat{b}_j$’s;
- tests for row and column heterogeneity can be made via $F$-tests.

Back to binary directed network data:

- the data take the form of a two-way layout;
- we’ve seen empirically the existence of row and column heterogeneity.

Is there an analogous RCE model for network data?
RCE model interpretation

For the RCE model

- the model is parameterized in terms of differences among row means and differences among column means;
- heterogeneity in the observed row/column means is represented by heterogeneity in the $\hat{a}_i$’s and $\hat{b}_j$’s;
- tests for row and column heterogeneity can be made via $F$-tests.

Back to binary directed network data:

- the data take the form of a two-way layout;
- we’ve seen empirically the existence of row and column heterogeneity.

Is there an analogous RCE model for network data?
Odd and probabilities

Let $\Pr(Y_{i,j} = 1) = \theta$. Then

$$\text{odds}(Y_{i,j} = 1) = \frac{\theta}{1 - \theta}$$

$$\log \text{odds}(Y_{i,j} = 1) = \log \frac{\theta}{1 - \theta} = \mu$$

Now write the probability in terms of the log odds $\mu$:

$$\log \text{odds}(Y_{i,j} = 1) = \mu$$

$$\text{odds}(Y_{i,j} = 1) = e^\mu$$

$$\frac{\theta}{1 - \theta} = e^\mu$$

$$\theta = e^\mu - \theta e^\mu$$

$$\theta(1 + e^\mu) = e^\mu$$

$$\theta = \Pr(Y_{i,j} = 1) = \frac{e^\mu}{1 + e^\mu}$$
Logistic regression

Let \( \{Y_{i,j} : i \neq j\} \) be independent binary random variables with

\[
\Pr(Y_{i,j}) = \theta_{i,j}.
\]

- If \( \mu_{i,j} = \mu \) for all \( i, j \), then we have the SRG\((n, \frac{\mu}{1+e^\mu})\) model.
- Our hypothesis tests generally reject this model.

Thus the hypothesis of homogeneous \( \mu_{i,j} \)'s is rejected.

What heterogeneity in the \( \mu_{i,j} \)'s will help us capture the observed data patterns?
Logistic regression

Let \( \{ Y_{i,j} : i \neq j \} \) be independent binary random variables with

\[
\Pr(Y_{i,j}) = \theta_{i,j}.
\]

- If \( \mu_{i,j} = \mu \) for all \( i, j \), then we have the SRG\((n, \frac{e^\mu}{1+e^\mu})\) model.
- Our hypothesis tests generally reject this model.

Thus the hypothesis of homogeneous \( \mu_{i,j} \)'s is rejected.

What heterogeneity in the \( \mu_{i,j} \)'s will help us capture the observed data patterns?
Logistic regression

Let \( \{ Y_{i,j} : i \neq j \} \) be independent binary random variables with

\[
Pr(Y_{i,j}) = \theta_{i,j}.
\]

- If \( \mu_{i,j} = \mu \) for all \( i, j \), then we have the \( \text{SRG}(n, \frac{e^\mu}{1+e^\mu}) \) model.
- Our hypothesis tests generally reject this model.

Thus the hypothesis of homogeneous \( \mu_{i,j} \)'s is rejected.

What heterogeneity in the \( \mu_{i,j} \)'s will help us capture the observed data patterns?
Logistic regression

Let \( \{ Y_{i,j} : i \neq j \} \) be independent binary random variables with

\[
\Pr(Y_{i,j}) = \theta_{i,j}.
\]

- If \( \mu_{i,j} = \mu \) for all \( i, j \), then we have the SRG(\( n, \frac{e^\mu}{1+e^\mu} \)) model.
- Our hypothesis tests generally reject this model.

Thus the hypothesis of homogeneous \( \mu_{i,j} \)'s is rejected.

What heterogeneity in the \( \mu_{i,j} \)'s will help us capture the observed data patterns?
Logistic regression

Let \( \{ Y_{i,j} : i \neq j \} \) be independent binary random variables with

\[
\Pr(Y_{i,j}) = \theta_{i,j}.
\]

- If \( \mu_{i,j} = \mu \) for all \( i, j \), then we have the \( \text{SRG}(n, \frac{e^{\mu}}{1+e^{\mu}}) \) model.
- Our hypothesis tests generally \textbf{reject} this model.

Thus the hypothesis of homogeneous \( \mu_{i,j} \)'s is rejected.

What heterogeneity in the \( \mu_{i,j} \)'s will help us capture the observed data patterns?
Logistic regression

Let \( \{ Y_{i,j} : i \neq j \} \) be independent binary random variables with

\[
\Pr(Y_{i,j}) = \theta_{i,j}.
\]

- If \( \mu_{i,j} = \mu \) for all \( i, j \), then we have the SRG\((n, \frac{e^{\mu}}{1+e^{\mu}})\) model.
- Our hypothesis tests generally reject this model.

Thus the hypothesis of homogeneous \( \mu_{i,j} \)'s is rejected.

What heterogeneity in the \( \mu_{i,j} \)'s will help us capture the observed data patterns?
Binary RCE model

**RCE model:** \( \{ Y_{i,j} : i \neq j \} \) are independent with

\[
\Pr(Y_{i,j} = 1) = \frac{e^{\mu_{i,j}}}{1 + e^{\mu_{i,j}}}
\]

\[
\mu_{i,j} = \mu + a_i + b_j
\]

or more compactly,

\[
\Pr(Y_{i,j} = 1) = \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}}
\]

**Interpretation:**

odds \( Y_{i,j} = 1 \) = \( e^{\mu + a_i + b_j} \)

odds ratio \( Y_{i,j} = 1 : Y_{k,j} = 1 \) = \( e^{\mu + a_i + b_j} / e^{\mu + a_k + b_j} \)

log odds ratio \( Y_{i,j} = 1 : Y_{k,j} = 1 \) = \([\mu + a_i + b_j] - [\mu + a_k + b_j] = a_i - a_k\)
Binary RCE model

**RCE model:** \( \{Y_{i,j} : i \neq j\} \) are independent with

\[
Pr(Y_{i,j} = 1) = \frac{e^{\mu_{i,j}}}{1 + e^{\mu_{i,j}}}
\]

\[
\mu_{i,j} = \mu + a_i + b_j
\]

or more compactly,

\[
Pr(Y_{i,j} = 1) = \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}}
\]

**Interpretation:**

\[
\text{odds}(Y_{i,j} = 1) = e^{\mu + a_i + b_j}
\]

\[
\text{odds ratio}(Y_{i,j} = 1 : Y_{k,j} = 1) = \frac{e^{\mu + a_i + b_j}}{e^{\mu + a_k + b_j}}
\]

\[
\log \text{odds ratio}(Y_{i,j} = 1 : Y_{k,j} = 1) = [\mu + a_i + b_j] - [\mu + a_k + b_j] = a_i - a_k
\]
Binary RCE model

\[ \Pr(Y_{i,j} = 1) = \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}} \]

The differences among the \(a_i\)'s represents heterogeneity among nodes in terms of the probability of sending a tie, and similarly for the \(b_j\)'s.

- The \(a_i\)'s can represent heterogeneity in nodal outdegree;
- The \(b_j\)'s can represent heterogeneity in nodal outdegree.

Caution: The row and column effects \(a\) and \(b\) are only identified up to differences. Suppose \(c + d + e = 0\):

\[
\Pr(Y_{i,j} = 1) = \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}} = \frac{e^{\mu + a_i + b_j + c + d + e}}{1 + e^{\mu + a_i + b_j + c + d + e}} = \frac{e^{(\mu + c) + (a_i + d) + (b_j + e)}}{1 + e^{(\mu + c) + (a_i + d) + (b_j + e)}} = \frac{e^{\tilde{\mu} + \tilde{a}_i + \tilde{b}_j}}{1 + e^{\tilde{\mu} + \tilde{a}_i + \tilde{b}_j}}
\]

It is standard to restrict the estimates somehow to reflect this:

- Sum to zero side conditions: \(\sum a_i = 0, \sum b_j = 0\)
- Set to zero side conditions: \(a_1 = b_1 = 0\)
Binary RCE model

\[ \Pr(Y_{i,j} = 1) = \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}} \]

The differences among the \(a_i\)'s represents heterogeneity among nodes in terms of the probability of sending a tie, and similarly for the \(b_j\)'s.

- The \(a_i\)'s can represent heterogeneity in nodal outdegree;
- The \(b_j\)'s can represent heterogeneity in nodal outdegree.

Caution: The row and column effects \(a\) and \(b\) are only identified up to differences. Suppose \(c + d + e = 0\):

\[
\begin{align*}
\Pr(Y_{i,j} = 1) &= \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}} \\
&= \frac{e^{\mu + a_i + b_j + c + d + e}}{1 + e^{\mu + a_i + b_j + c + d + e}} \\
&= \frac{e^{(\mu + c) + (a_i + d) + (b_j + e)}}{1 + e^{(\mu + c) + (a_i + d) + (b_j + e)}} = e^{\tilde{\mu} + \tilde{a}_i + \tilde{b}_j} \\
&= \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}}
\end{align*}
\]

It is standard to restrict the estimates somehow to reflect this:

- Sum to zero side conditions: \(\sum a_i = 0\), \(\sum b_j = 0\)
- Set to zero side conditions: \(a_1 = b_1 = 0\)
Binary RCE model

\[ \Pr(Y_{i,j} = 1) = \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}} \]

The differences among the \( a_i \)'s represents heterogeneity among nodes in terms of the probability of sending a tie, and similarly for the \( b_j \)'s.

- The \( a_i \)'s can represent heterogeneity in nodal outdegree;
- The \( b_j \)'s can represent heterogeneity in nodal outdegree.

Caution: The row and column effects \( a \) and \( b \) are only identified up to differences. Suppose \( c + d + e = 0 \):

\[ \Pr(Y_{i,j} = 1) = \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}} = \frac{e^{\mu + a_i + b_j + c + d + e}}{1 + e^{\mu + a_i + b_j + c + d + e}} = \frac{e^{(\mu + c) + (a_i + d) + (b_j + e)}}{1 + e^{(\mu + c) + (a_i + d) + (b_j + e)}} = \frac{e^{\tilde{\mu} + \tilde{a}_i + \tilde{b}_j}}{1 + e^{\tilde{\mu} + \tilde{a}_i + \tilde{b}_j}} \]

It is standard to restrict the estimates somehow to reflect this:

- Sum to zero side conditions : \( \sum a_i = 0, \sum b_j = 0 \)
- Set to zero side conditions : \( a_1 = b_1 = 0 \)
The differences among the $a_i$'s represents heterogeneity among nodes in terms of the probability of sending a tie, and similarly for the $b_j$'s.

- The $a_i$'s can represent heterogeneity in nodal outdegree;
- The $b_j$'s can represent heterogeneity in nodal outdegree.

Caution: The row and column effects $a$ and $b$ are only identified up to differences. Suppose $c + d + e = 0$:

$$
\Pr(Y_{i,j} = 1) = \frac{e^{\mu+a_i+b_j}}{1 + e^{\mu+a_i+b_j}} = \frac{e^{\mu+a_i+b_j+c+d+e}}{1 + e^{\mu+a_i+b_j+c+d+e}} = \frac{e^{(\mu+c)+(a_i+d)+(b_j+e)}}{1 + e^{(\mu+c)+(a_i+d)+(b_j+e)}} = \frac{e^{\tilde{\mu}+\tilde{a}_i+\tilde{b}_j}}{1 + e^{\tilde{\mu}+\tilde{a}_i+\tilde{b}_j}}
$$

It is standard to restrict the estimates somehow to reflect this:

- Sum to zero side conditions: $\sum a_i = 0, \sum b_j = 0$
- Set to zero side conditions: $a_1 = b_1 = 0$
Binary RCE model

\[ \Pr(Y_{i,j} = 1) = \frac{e^{\mu+a_i+b_j}}{1 + e^{\mu+a_i+b_j}} \]

The differences among the \(a_i\)'s represents heterogeneity among nodes in terms of the probability of sending a tie, and similarly for the \(b_j\)'s.

- The \(a_i\)'s can represent heterogeneity in nodal outdegree;
- The \(b_j\)'s can represent heterogeneity in nodal outdegree.

Caution: The row and column effects \(a\) and \(b\) are only identified up to differences. Suppose \(c + d + e = 0\):

\[ \Pr(Y_{i,j} = 1) = \frac{e^{\mu+a_i+b_j}}{1 + e^{\mu+a_i+b_j}} \]

\[ = \frac{e^{\mu+a_i+b_j+c+d+e}}{1 + e^{\mu+a_i+b_j+c+d+e}} \]

\[ = \frac{e^{(\mu+c)+(a_i+d)+(b_j+e)}}{1 + e^{(\mu+c)+(a_i+d)+(b_j+e)}} = \frac{e^{\tilde{\mu}+\tilde{a}_i+\tilde{b}_j}}{1 + e^{\tilde{\mu}+\tilde{a}_i+\tilde{b}_j}} \]

It is standard to restrict the estimates somehow to reflect this:

- Sum to zero side conditions : \(\sum a_i = 0, \sum b_j = 0\)
- Set to zero side conditions : \(a_1 = b_1 = 0\)
Binary RCE model

\[ \Pr(Y_{i,j} = 1) = \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}} \]

The differences among the \( a_i \)'s represents heterogeneity among nodes in terms of the probability of sending a tie, and similarly for the \( b_j \)'s.

- The \( a_i \)'s can represent heterogeneity in nodal outdegree;
- The \( b_j \)'s can represent heterogeneity in nodal outdegree.

Caution: The row and column effects \( a \) and \( b \) are only identified up to differences. Suppose \( c + d + e = 0 \):

\[ \Pr(Y_{i,j} = 1) = \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}} = \frac{e^{\mu + a_i + b_j + c + d + e}}{1 + e^{\mu + a_i + b_j + c + d + e}} = \frac{e^{(\mu + c) + (a_i + d) + (b_j + e)}}{1 + e^{(\mu + c) + (a_i + d) + (b_j + e)}} = \frac{e^{\tilde{\mu} + \tilde{a}_i + \tilde{b}_j}}{1 + e^{\tilde{\mu} + \tilde{a}_i + \tilde{b}_j}} \]

It is standard to restrict the estimates somehow to reflect this:

- Sum to zero side conditions : \( \sum a_i = 0, \sum b_j = 0 \)
- Set to zero side conditions : \( a_1 = b_1 = 0 \)
Binary RCE model

\[ \Pr(Y_{i,j} = 1) = \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}} \]

The differences among the \( a_i \)'s represents heterogeneity among nodes in terms of the probability of sending a tie, and similarly for the \( b_j \)'s.

- The \( a_i \)'s can represent heterogeneity in nodal outdegree;
- The \( b_j \)'s can represent heterogeneity in nodal outdegree.

**Caution:** The row and column effects \( a \) and \( b \) are only identified up to differences. Suppose \( c + d + e = 0 \):

\[
\Pr(Y_{i,j} = 1) = \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}} = \frac{e^{\mu + a_i + b_j + c + d + e}}{1 + e^{\mu + a_i + b_j + c + d + e}} = \frac{e^{(\mu + c) + (a_i + d) + (b_j + e)}}{1 + e^{(\mu + c) + (a_i + d) + (b_j + e)}} = \frac{e^{\tilde{\mu} + \tilde{a}_i + \tilde{b}_j}}{1 + e^{\tilde{\mu} + \tilde{a}_i + \tilde{b}_j}}
\]

It is standard to restrict the estimates somehow to reflect this:

- **Sum to zero side conditions** : \( \sum a_i = 0, \sum b_j = 0 \)
- **Set to zero side conditions** : \( a_1 = b_1 = 0 \)
Binary RCE model

\[
\text{Pr}(Y_{i,j} = 1) = \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}}
\]

The differences among the \(a_i\)'s represents heterogeneity among nodes in terms of the probability of sending a tie, and similarly for the \(b_j\)'s.

- The \(a_i\)'s can represent heterogeneity in nodal outdegree;
- The \(b_j\)'s can represent heterogeneity in nodal outdegree.

**Caution:** The row and column effects \(a\) and \(b\) are only identified up to differences. Suppose \(c + d + e = 0\):

\[
\text{Pr}(Y_{i,j} = 1) = \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}}
\]

\[
= \frac{e^{\mu + a_i + b_j + c + d + e}}{1 + e^{\mu + a_i + b_j + c + d + e}}
\]

\[
= \frac{e^{(\mu + c) + (a_i + d) + (b_j + e)}}{1 + e^{(\mu + c) + (a_i + d) + (b_j + e)}}
\]

\[
= \frac{e^{\tilde{\mu} + \tilde{a}_i + \tilde{b}_j}}{1 + e^{\tilde{\mu} + \tilde{a}_i + \tilde{b}_j}}
\]

It is standard to restrict the estimates somehow to reflect this:

- **Sum to zero side conditions:** \(\sum a_i = 0, \sum b_j = 0\)
- **Set to zero side conditions:** \(a_1 = b_1 = 0\)
To make an analogy to the two-way factorial design,
  - sender index = row factor;
  - receiver index = columns factor.

To fit the RCE model with logistic regression in R, we need to
  - code the senders as a row factor;
  - code the receivers as a column factor;
  - convert everything to vectors.
Fitting the binary RCE in R

To make an analogy to the two-way factorial design,

- sender index = row factor;
- receiver index = columns factor.

To fit the RCE model with logistic regression in R, we need to

- code the senders as a row factor;
- code the receivers as a column factor;
- convert everything to vectors.
To make an analogy to the two-way factorial design,

- sender index = row factor;
- receiver index = columns factor.

To fit the RCE model with logistic regression in R, we need to

- code the senders as a row factor;
- code the receivers as a column factor;
- convert everything to vectors.
Fitting the binary RCE in R

To make an analogy to the two-way factorial design,
- sender index = row factor;
- receiver index = columns factor.

To fit the RCE model with logistic regression in R, we need to
- code the senders as a row factor;
- code the receivers as a column factor;
- convert everything to vectors.
Fitting the binary RCE in R

To make an analogy to the two-way factorial design,

- sender index = row factor;
- receiver index = columns factor.

To fit the RCE model with logistic regression in R, we need to

- code the senders as a row factor;
- code the receivers as a column factor;
- convert everything to vectors.
To make an analogy to the two-way factorial design,
- sender index = row factor;
- receiver index = columns factor.

To fit the RCE model with logistic regression in R, we need to
- code the senders as a row factor;
- code the receivers as a column factor;
- convert everything to vectors.
Fitting the binary RCE in R

To make an analogy to the two-way factorial design,
- sender index = row factor;
- receiver index = columns factor.

To fit the RCE model with logistic regression in R, we need to
- code the senders as a row factor;
- code the receivers as a column factor;
- convert everything to vectors.
To make an analogy to the two-way factorial design,

- sender index = row factor;
- receiver index = columns factor.

To fit the RCE model with logistic regression in R, we need to

- code the senders as a row factor;
- code the receivers as a column factor;
- convert everything to vectors.
To make an analogy to the two-way factorial design,
- sender index = row factor;
- receiver index = columns factor.

To fit the RCE model with logistic regression in R, we need to
- code the senders as a row factor;
- code the receivers as a column factor;
- convert everything to vectors.
Fitting the binary RCE in R

Ridx <- matrix((1:nrow(Y)), nrow(Y), nrow(Y))
Cidx <- t(Ridx)

Ridx[1:4, 1:4]

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

Cidx[1:4, 1:4]

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

Y[1:4, 1:4]

## ROMUL BONAVEN AMBROSE BERTH
## ROMUL   NA    1    1    0
## BONAVEN  1    NA    0    0
## AMBROSE  1    1    NA    0
## BERTH    0    0    0    NA
Fitting the binary RCE in R

```r
y <- c(Y)
ridx <- c(Ridx)
cidx <- c(Cidx)

y[1:22]
## [1] NA 1 1 0 1 1 1 1 0 0 0 1 0 1 1 1 0 1 1 NA 1 0

ridx[1:22]
## [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 1  2  3  4

cidx[1:22]
## [1]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  2  2  2
```
Fitting the binary RCE in R

```r
fit <- glm( y ~ factor(ridx) + factor(cidx), family=binomial)
summary(fit)
```

```r
##
## Call:
## glm(formula = y ~ factor(ridx) + factor(cidx), family = binomial)
##
## Deviance Residuals:
## Min 1Q Median 3Q Max
## -1.5975 -0.8044 -0.5725 0.9091 2.2146
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.1154 0.7558 1.476 0.14001
## factor(ridx)2 -0.3175 0.7780 -0.408 0.68319
## factor(ridx)3 -0.4545 0.7823 -0.581 0.56124
## factor(ridx)4 -0.7633 0.8136 -0.938 0.34816
## factor(ridx)5 -0.3557 0.7820 -0.455 0.64920
## factor(ridx)6 -0.8156 0.8103 -1.007 0.31415
## factor(ridx)7 -0.3954 0.7839 -0.504 0.61394
## factor(ridx)8 -0.1036 0.7654 -0.135 0.89232
## factor(ridx)9 -0.7818 0.8128 -0.962 0.33616
## factor(ridx)10 -0.1668 0.7624 -0.219 0.82686
## factor(ridx)11 -0.4545 0.7823 -0.581 0.56124
## factor(ridx)12 -0.7231 0.8133 -0.889 0.37395
## factor(ridx)13 -1.1720 0.8627 -1.358 0.17433
## factor(ridx)14 -0.3954 0.7839 -0.504 0.61394
## factor(ridx)15 -0.1453 0.7641 -0.190 0.84917
## factor(ridx)16 -0.4545 0.7823 -0.581 0.56124
## factor(ridx)17 -0.8156 0.8103 -1.007 0.31415
## factor(ridx)18 -0.1453 0.7641 -0.190 0.84917
## factor(cidx)2 -0.2721 0.7164 -0.380 0.70409
## factor(cidx)3 -2.2164 0.8229 -2.693 0.00707 **
## factor(cidx)4 -1.5578 0.7449 -2.091 0.03650 *
## factor(cidx)5 -0.7603 0.7119 -1.068 0.28555
## factor(cidx)6 -2.7143 0.9160 -2.963 0.00305 **
## factor(cidx)7 -1.2622 0.7272 -1.736 0.08261 .
## factor(cidx)8 -1.5182 0.7444 -2.040 0.04139 *
## factor(cidx)9 -1.8669 0.7738 -2.413 0.01583 *
## factor(cidx)10 -2.6772 0.9159 -2.923 0.00347 **
## factor(cidx)11 -2.2164 0.8229 -2.693 0.00707 **
## factor(cidx)12 -1.0265 0.7169 -1.432 0.15220
## factor(cidx)13 -1.5763 0.7443 -2.118 0.03419 *
## factor(cidx)14 -1.2622 0.7272 -1.736 0.08261 .
## factor(cidx)15 -2.1961 0.8225 -2.670 0.00758 **
## factor(cidx)16 -2.2164 0.8229 -2.693 0.00707 **
## factor(cidx)17 -2.7143 0.9160 -2.963 0.00305 **
## factor(cidx)18 -2.1961 0.8225 -2.670 0.00758 **
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 367.18 on 305 degrees of freedom
## Residual deviance: 327.43 on 271 degrees of freedom
## (18 observations deleted due to missingness)
## AIC: 397.43
##
## Number of Fisher Scoring iterations: 4
```
Fitting the binary RCE in R

```r
fit<-glm( y ~ C(factor(ridx),sum) + C(factor(cidx),sum) , family=binomial)
summary(fit)
```

```
##
## Call:
## glm(formula = y ~ C(factor(ridx), sum) + C(factor(cidx), sum),
##     family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
##-1.5975  -0.8044  -0.5725   0.9091  2.2146
##
## Coefficients:
##            Estimate Std. Error   z value  Pr(>|z|)
## (Intercept) -1.03516    0.14270 -7.25401 4.04e-13 ***
## C(factor(ridx), sum)1  0.47000    0.52409  0.89699  0.36983
## C(factor(ridx), sum)2  0.15250    0.54948  0.27815  0.78137
## C(factor(ridx), sum)3  0.01546    0.55376  0.02844  0.97772
## C(factor(ridx), sum)4 -0.29326    0.59177 -0.49630  0.62020
## C(factor(ridx), sum)5  0.11430    0.55418  0.20593  0.83659
## C(factor(ridx), sum)6 -0.34557    0.58751 -0.58754  0.55641
## C(factor(ridx), sum)7  0.07456    0.55617  0.13392  0.89336
## C(factor(ridx), sum)8  0.36639    0.53352  0.68669  0.49225
## C(factor(ridx), sum)9 -0.31179    0.59078 -0.52756  0.59767
## C(factor(ridx), sum)10 0.30324    0.52919  0.57324  0.56663
## C(factor(ridx), sum)11 0.01546    0.55376  0.02844  0.97772
## C(factor(ridx), sum)12 -0.25315    0.59185 -0.42837  0.66885
## C(factor(ridx), sum)13 -0.70197    0.64986 -1.07994  0.27992
## C(factor(ridx), sum)14 0.07456    0.55617  0.13392  0.89336
## C(factor(ridx), sum)15 0.32468    0.53146  0.61124  0.54125
## C(factor(ridx), sum)16 -0.51560    0.62097 -0.83068  0.40637
## C(factor(ridx), sum)17 -1.03374    0.72727 -1.42141  0.15520
## C(factor(cidx), sum)1  1.68053    0.50580  3.32276  0.00089 ***
## C(factor(cidx), sum)2  1.40844    0.49285  2.85763  0.00427 **
## C(factor(cidx), sum)3 -0.53583    0.62103 -0.86355  0.38825
## C(factor(cidx), sum)4  0.12272    0.52658  0.23276  0.81572
## C(factor(cidx), sum)5  0.92026    0.48615  1.89280  0.05837 .
## C(factor(cidx), sum)6 -1.03374    0.72727 -1.42141  0.15520
## C(factor(cidx), sum)7  0.41828    0.50504  0.82793  0.40754
## C(factor(cidx), sum)8  0.16234    0.52692  0.30771  0.75801
## C(factor(cidx), sum)9 -0.18637    0.56207 -0.33162  0.74021
## C(factor(cidx), sum)10 -0.99664    0.72772 -1.36957  0.17083
## C(factor(cidx), sum)11 -0.53583    0.62103 -0.86355  0.38825
## C(factor(cidx), sum)12  0.65405    0.49161  1.32994  0.18338
## C(factor(cidx), sum)13  0.10426    0.52545  0.19773  0.84271
## C(factor(cidx), sum)14  0.41828    0.50504  0.82793  0.40754
## C(factor(cidx), sum)15 -0.51560    0.62097 -0.82935  0.40637
## C(factor(cidx), sum)16 -0.53583    0.62103 -0.86355  0.38825
## C(factor(cidx), sum)17 -1.03374    0.72727 -1.42141  0.15520
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 367.18 on 305 degrees of freedom
## Residual deviance: 327.43 on 271 degrees of freedom
## (18 observations deleted due to missingness)
## AIC: 397.43
##
## Number of Fisher Scoring iterations: 4
```
Fitting the binary RCE in R

\begin{verbatim}
mu.hat<-fit$coef[1]
a.hat<- fit$coef[1+1:(nrow(Y)-1)] ; a.hat<-c(a.hat,-sum(a.hat) )
b.hat<- fit$coef[nrow(Y)+1:(nrow(Y)-1)] ; b.hat<-c(b.hat,-sum(b.hat) )
\end{verbatim}

- Why do nodes with the same outdegrees have different $\hat{a}_i$'s?
- Why do nodes with the same indegrees have different $\hat{b}_j$'s?
Fitting the binary RCE in R

\[
\mu_{\hat{}} <- \text{fit$coef[1]} \\
a_{\hat{}} <- \text{fit$coef[1+1:(\text{nrow(Y)-1})]} \; ; \; a_{\hat{}} <- c(a_{\hat{}}, -\text{sum(a_{\hat{}})}) \\
b_{\hat{}} <- \text{fit$coef[\text{nrow(Y)+1:(\text{nrow(Y)-1})}]} \; ; \; b_{\hat{}} <- c(b_{\hat{}}, -\text{sum(b_{\hat{}})})
\]

- Why do nodes with the same outdegrees have different \(\hat{a}_i\)'s?
- Why do nodes with the same indegrees have different \(\hat{b}_j\)'s?
Fitting the binary RCE in R

mu.hat<-fit$coef[1]
a.hat<- fit$coef[1+1:(nrow(Y)-1)] ; a.hat<-c(a.hat,-sum(a.hat) )
b.hat<- fit$coef[nrow(Y)+1:(nrow(Y)-1)] ; b.hat<-c(b.hat,-sum(b.hat) )

• Why do nodes with the same outdegrees have different $\hat{a}_i$'s?
• Why do nodes with the same indegrees have different $\hat{b}_j$'s?
Fitting the binary RCE in R

\[
\mu.hat <- \text{fit$coef}[1] \\
a.hat <- \text{fit$coef}[1 + 1:(\text{nrow}(Y) - 1)] ; a.hat <- c(a.hat, -\text{sum}(a.hat)) \\
b.hat <- \text{fit$coef}[\text{nrow}(Y) + 1:(\text{nrow}(Y) - 1)] ; b.hat <- c(b.hat, -\text{sum}(b.hat))
\]

- Why do nodes with the same outdegrees have different \( \hat{a}_i \)'s?
- Why do nodes with the same indegrees have different \( \hat{b}_j \)'s?
Understanding the MLEs

\[
\text{muij.mle} \leftarrow \mu.hat + \text{outer}(a.hat, b.hat, "\+)
\]
\[
p.mle \leftarrow \exp(\text{muij.mle})/(1+\exp(\text{muij.mle})) \; \text{; diag}(\text{p.mle}) \leftarrow \text{NA}
\]
Understanding MLEs

$$\Pr(Y|\mu, a, b) = \prod_{i \neq j} \frac{e^{(\mu + a_i + b_j)y_{i,j}}}{1 + e^{\mu + a_i + b_j}}$$

$$= \exp(\mu y.. + \sum a_i y_{i.} + \sum b_j y_{.j}) \prod_{i \neq j} (1 + e^{\mu + a_i + b_j})^{-1}$$

Let’s find the maximizer of the log-likelihood in $a_i$:

$$\log \Pr(Y|\mu, a, b) = \mu y.. + \sum a_i y_{i.} + \sum b_j y_{.j} - \sum_{i \neq j} \log(1 + e^{\mu + a_i + b_j})$$

$$\frac{\partial}{\partial a_i} \log \Pr(Y|\mu, a, b) = y_{i.} - \sum_{i \neq j} \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}}$$

The MLE occurs at values $(\hat{\mu}, \hat{a}, \hat{b})$ for which

$$y_{i.} = \sum_{j: i \neq j} \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}}$$

i.e., $y_{i.} = \sum_{j} \hat{p}_{i,j}$
Understanding MLEs

\[
\Pr(Y|\mu, a, b) = \prod_{i \neq j} \frac{e^{(\mu + a_i + b_j) y_{i,j}}}{1 + e^{\mu + a_i + b_j}}
\]

\[
= \exp(\mu y.. + \sum a_i y_{i.} + \sum b_j y_{.j}) \prod_{i \neq j} (1 + e^{\mu + a_i + b_j})^{-1}
\]

Let’s find the maximizer of the log-likelihood in \(a_i\):

\[
\log \Pr(Y|\mu, a, b) = \mu y.. + \sum a_i y_{i.} + \sum b_j y_{.j} - \sum_{i \neq j} \log(1 + e^{\mu + a_i + b_j})
\]

\[
\frac{\partial}{\partial a_i} \log \Pr(Y|\mu, a, b) = y_{i.} - \sum_{i \neq j} \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}}
\]

The MLE occurs at values \((\hat{\mu}, \hat{a}, \hat{b})\) for which

\[
y_{i.} = \sum_{j: i \neq j} \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}}
\]

i.e., \(y_{i.} = \sum_j \hat{p}_{i,j}\)
Understanding MLEs

\[
\Pr(Y|\mu,a,b) = \prod_{i \neq j} \frac{e^{(\mu+a_i+b_j)y_{i,j}}}{1 + e^{\mu+a_i+b_j}}
\]

\[
= \exp(\mu y.. + \sum a_i y_{i.} + \sum b_j y_{.j}) \prod_{i \neq j} (1 + e^{\mu+a_i+b_j})^{-1}
\]

Let’s find the maximizer of the log-likelihood in \(a_i\):

\[
\log \Pr(Y|\mu,a,b) = \mu y.. + \sum a_i y_{i.} + \sum b_j y_{.j} - \sum_{i \neq j} \log(1 + e^{\mu+a_i+b_j})
\]

\[
\frac{\partial}{\partial a_i} \log \Pr(Y|\mu,a,b) = y_{i.} - \sum_{i \neq j} \frac{e^{\mu+a_i+b_j}}{1 + e^{\mu+a_i+b_j}}
\]

The MLE occurs at values \((\hat{\mu}, \hat{a}, \hat{b})\) for which

\[
y_{i.} = \sum_{j \neq i} \frac{e^{\mu+a_i+b_j}}{1 + e^{\mu+a_i+b_j}}
\]

i.e., \(y_{i.} = \sum_j \hat{p}_{i,j}\)
Understanding MLEs

\[
\Pr(Y|\mu, a, b) = \prod_{i \neq j} \frac{e^{(\mu + a_i + b_j) y_{i,j}}}{1 + e^{\mu + a_i + b_j}}
\]

\[
= \exp(\mu y_\cdot + \sum a_i y_{i.} + \sum b_j y_{.j}) \prod_{i \neq j} (1 + e^{\mu + a_i + b_j})^{-1}
\]

Let’s find the maximizer of the log-likelihood in \(a_i\):

\[
\log \Pr(Y|\mu, a, b) = \mu y_\cdot + \sum a_i y_{i.} + \sum b_j y_{.j} - \sum_{i \neq j} \log(1 + e^{\mu + a_i + b_j})
\]

\[
\frac{\partial}{\partial a_i} \log \Pr(Y|\mu, a, b) = y_{i.} - \sum_{i \neq j} \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}}
\]

The MLE occurs at values \((\hat{\mu}, \hat{a}, \hat{b})\) for which

\[
y_{i.} = \sum_{j: i \neq j} \frac{e^{\mu + a_i + b_j}}{1 + e^{\mu + a_i + b_j}}
\]

i.e., \(y_{i.} = \sum_j \hat{p}_{i,j}\)
Model evaluation: Best case scenario

How does the “best” RCE model compare to the data?

\[ \log \text{odds}(Y_{i,j} = 1) = \mu_{i,j} \]
\[ \mu_{i,j} = \mu + a_i + b_j \]
\[ \hat{\mu}_{i,j} = \hat{\mu} + \hat{a}_i + \hat{b}_j \]

muij.mle <- mu.hat + outer(a.hat, b.hat, "+")
p.mle <- exp(muij.mle) / (1 + exp(muij.mle))

t.M <- NULL
for(s in 1:S)
{
  Ysim <- matrix(rbinom(n^2, 1, p.mle), n, n); diag(Ysim) <- NA
  t.M <- rbind(t.M, c(mmean(Ysim), sd(rsum(Ysim)), sd(csum(Ysim))))
}
Model evaluation: Best case scenario

- improvement in terms of indegree heterogeneity;
- still a discrepancy in terms of outdegree heterogeneity (why?)
Comparison to the i.i.d. model
Model comparison

Often it is desirable to have a numerical comparisons of two models.

SRG : Simple model, shows lack of fit.
RCE : Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”?

Selecting between two false models requires a balance between

model fit and model complexity

- Overly simple model: high explanatory power, poor fit to data, poor
generalizability;
- Overly complex model: low explanatory power, great fit to data, poor
generalizability;
- Sufficiently complex model: good explanatory power, good fit to data, and
good generalizability.

These notions can be made precise in various contexts:
- communication and information theory;
Model comparison

Often it is desirable to have a numerical comparison of two models.

SRG : Simple model, shows lack of fit.
RCE : Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”? Selecting between two false models requires a balance between

model fit and model complexity

- Overly simple model: high explanatory power, poor fit to data, poor generalizability;
- Overly complex model: low explanatory power, great fit to data, poor generalizability;
- Sufficiently complex model: good explanatory power, good fit to data, and good generalizability.

These notions can be made precise in various contexts:

- communication and information theory;
Model comparison

Often it is desirable to have a numerical comparison of two models.

SRG : Simple model, shows lack of fit.
RCE : Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”?  

Selecting between two false models requires a balance between 

**model fit** and **model complexity**

- Overly simple model: high explanatory power, poor fit to data, poor generalizability;
- Overly complex model: low explanatory power, great fit to data, poor generalizability;
- Sufficiently complex model: good explanatory power, good fit to data, and good generalizability.

These notions can be made precise in various contexts:
- communication and information theory;
Model comparison

Often it is desirable to have a numerical comparison of two models.

**SRG** : Simple model, shows lack of fit.

**RCE** : Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”?  

Selecting between two false models requires a balance between

- **model fit** and **model complexity**

- Overly simple model: high explanatory power, poor fit to data, poor generalizability;

- Overly complex model: low explanatory power, great fit to data, poor generalizability;

- Sufficiently complex model: good explanatory power, good fit to data, and good generalizability.

These notions can be made precise in various contexts:

- communication and information theory;

Model comparison

Often it is desirable to have a numerical comparisons of two models.

SRG : Simple model, shows lack of fit.

RCE : Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”? Selecting between two false models requires a balance between

model fit and model complexity

• Overly simple model: high explanatory power, poor fit to data, poor generalizability;
• Overly complex model: low explanatory power, great fit to data, poor generalizability;
• Sufficiently complex model: good explanatory power, good fit to data, and good generalizability.

These notions can be made precise in various contexts:

• communication and information theory;
• out-of-sample predictive performance.
Model comparison

Often it is desirable to have a numerical comparison of two models.

**SRG**: Simple model, shows lack of fit.

**RCE**: Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”?

Selecting between two false models requires a balance between

- **model fit** and **model complexity**

  - Overly simple model: high explanatory power, poor fit to data, poor generalizability;
  - Overly complex model: low explanatory power, great fit to data, poor generalizability;
  - Sufficiently complex model: good explanatory power, good fit to data, and good generalizability.

These notions can be made precise in various contexts:

- communication and information theory;
Model comparison

Often it is desirable to have a numerical comparisons of two models.

**SRG**: Simple model, shows lack of fit.

**RCE**: Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”? Selecting between two false models requires a balance between

- **model fit** and **model complexity**

- Overly simple model: high explanatory power, poor fit to data, poor generalizability;

- Overly complex model: low explanatory power, great fit to data, poor generalizability;

- Sufficiently complex model: good explanatory power, good fit to data, and good generalizability.

These notions can be made precise in various contexts:

- communication and information theory;

Model comparison

Often it is desirable to have a numerical comparison of two models.

**SRG**: Simple model, shows lack of fit.

**RCE**: Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”? Selecting between two false models requires a balance between

- *model fit* and *model complexity*

- Overly simple model: high explanatory power, poor fit to data, poor generalizability;

- Overly complex model: low explanatory power, great fit to data, poor generalizability;

- Sufficiently complex model: good explanatory power, good fit to data, and good generalizability.

These notions can be made precise in various contexts:

- communication and information theory;

Model comparison

Often it is desirable to have a numerical comparisons of two models.

SRG : Simple model, shows lack of fit.
RCE : Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”? 

Selecting between two false models requires a balance between 

**model fit and model complexity**

- Overly simple model: high explanatory power, poor fit to data, poor generalizability;
- Overly complex model: low explanatory power, great fit to data, poor generalizability;
- Sufficiently complex model: good explanatory power, good fit to data, and good generalizability.

These notions can be made precise in various contexts:

- communication and information theory;
Model comparison

Often it is desirable to have a numerical comparisons of two models.

**SRG**: Simple model, shows lack of fit.

**RCE**: Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”?

Selecting between two false models requires a balance between

- **model fit** and **model complexity**

- Overly simple model: high explanatory power, poor fit to data, poor generalizability;
- Overly complex model: low explanatory power, great fit to data, poor generalizability;
- Sufficiently complex model: good explanatory power, good fit to data, and good generalizability.

These notions can be made precise in various contexts:

- Communication and information theory;
- Out-of-sample predictive performance.
Model comparison

Often it is desirable to have a numerical comparisons of two models.

SRG : Simple model, shows lack of fit.
RCE : Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”? Selecting between two false models requires a balance between

model fit and model complexity

- Overly simple model: high explanatory power, poor fit to data, poor generalizability;
- Overly complex model: low explanatory power, great fit to data, poor generalizability;
- Sufficiently complex model: good explanatory power, good fit to data, and good generalizability.

These notions can be made precise in various contexts:

- communication and information theory;
Model comparison

Often it is desirable to have a numerical comparisons of two models.

SRG : Simple model, shows lack of fit.
RCE : Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”? 

Selecting between two false models requires a balance between

**model fit** and **model complexity**

- Overly simple model: high explanatory power, poor fit to data, poor generalizability;
- Overly complex model: low explanatory power, great fit to data, poor generalizability;
- Sufficiently complex model: good explanatory power, good fit to data, and good generalizability.

These notions can be made precise in various contexts:

- communication and information theory;
Model comparison

Often it is desirable to have a numerical comparison of two models.

SRG : Simple model, shows lack of fit.
RCE : Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”? 

Selecting between two false models requires a balance between

model fit and model complexity

- Overly simple model: high explanatory power, poor fit to data, poor generalizability;
- Overly complex model: low explanatory power, great fit to data, poor generalizability;
- Sufficiently complex model: good explanatory power, good fit to data, and good generalizability.

These notions can be made precise in various contexts:
- communication and information theory;
Model comparison

Often it is desirable to have a numerical comparison of two models.

**SRG**: Simple model, shows lack of fit.

**RCE**: Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”? Selecting between two false models requires a balance between

- **model fit** and **model complexity**

- Overly simple model: high explanatory power, poor fit to data, poor generalizability;
- Overly complex model: low explanatory power, great fit to data, poor generalizability;
- Sufficiently complex model: good explanatory power, good fit to data, and good generalizability.

These notions can be made precise in various contexts:

- communication and information theory;
Model comparison

Often it is desirable to have a numerical comparison of two models.

**SRG**: Simple model, shows lack of fit.

**RCE**: Complex model, shows less lack of fit.

Is the RCE model better? Is the added complexity “worth it”?

Selecting between two false models requires a balance between

- **model fit** and **model complexity**

- Overly simple model: high explanatory power, poor fit to data, poor generalizability;
- Overly complex model: low explanatory power, great fit to data, poor generalizability;
- Sufficiently complex model: good explanatory power, good fit to data, and good generalizability.

These notions can be made precise in various contexts:

- communication and information theory;
Model comparison

model performance = model fit - model complexity

Various numerical criteria have been suggested for fit and complexity. One proposal is

- fit = maximized log likelihood
- complexity = number of parameters

which leads to

$$AIC = \log p(Y|\hat{\theta}) - p$$

For historical reasons, this is often multiplied by $-2$:

$$AIC = -2 \log p(Y|\hat{\theta}) + 2p,$$

so that a low value of the AIC is better.
Model comparison

model performance = model fit - model complexity

Various numerical criteria have been suggested for \textit{fit} and \textit{complexity}. One proposal is

- \textit{fit} = maximized log likelihood
- \textit{complexity} = number of parameters

which leads to

\[ \hat{AIC} = \log p(Y|\hat{\theta}) - p \]

For historical reasons, this is often multiplied by \(-2\):

\[ AIC = -2 \log p(Y|\hat{\theta}) + 2p, \]

so that a \textit{low} value of the AIC is \textit{better}. 
Model comparison

model performance = model fit - model complexity

Various numerical criteria have been suggested for fit and complexity. One proposal is

- fit = maximized log likelihood
- complexity = number of parameters

which leads to

\[ \hat{AIC} = \log p(Y|\hat{\theta}) - p \]

For historical reasons, this is often multiplied by $-2$:

\[ AIC = -2 \log p(Y|\hat{\theta}) + 2p, \]

so that a low value of the AIC is better.
Model comparison

model performance = model fit - model complexity

Various numerical criteria have been suggested for fit and complexity. One proposal is

- fit = maximized log likelihood
- complexity = number of parameters

which leads to

\[ \hat{AIC} = \log p(\mathbf{Y}|\hat{\theta}) - p \]

For historical reasons, this is often multiplied by \(-2\):

\[ AIC = -2 \log p(\mathbf{Y}|\hat{\theta}) + 2p, \]

so that a low value of the AIC is better.
Model comparison

```r
fit.0 <- glm( y ~ 1, family = binomial)
fit.rc <- glm( y ~ C(factor(ridx),sum) + C(factor(cidx),sum), family = binomial)
AIC(fit.0)
## [1] 369.183
AIC(fit.rc)
## [1] 397.4326
```

This isn't too surprising:

- `fit.rc` gives a better fit, but
- `fit.rc` has $n + n - 2 = 34$ more parameters than `fit.0`.

Can you think of a better model?

Recall `fit.rc` only seemed to improve the indegree fit.

```r
fit.r <- glm( y ~ C(factor(ridx),sum), family = binomial)
fit.c <- glm( y ~ C(factor(cidx),sum), family = binomial)
AIC(fit.r)
## [1] 399.1471
AIC(fit.c)
## [1] 368.308
```
Model comparison

\[
\begin{align*}
\text{fit.0} & \leftarrow \text{glm}( \, y \sim 1, \, \text{family=binomial} \) \\
\text{fit.rc} & \leftarrow \text{glm}( \, y \sim C(\text{factor(ridx)},\text{sum})+C(\text{factor(cidx)},\text{sum}), \, \text{family=binomial} \)
\end{align*}
\]

\[\text{AIC(fit.0)}\]

\[
\begin{align*}
\text{AIC(fit.rc)}
\end{align*}
\]

This isn't too surprising:
• \text{fit.rc} gives a better fit, but
• \text{fit.rc} has \(n + n - 2 = 34\) more parameters than \text{fit.0}.

Can you think of a better model?
Recall \text{fit.rc} only seemed to improve the indegree fit.

\[
\begin{align*}
\text{fit.r} & \leftarrow \text{glm}( \, y \sim C(\text{factor(ridx)},\text{sum}), \, \text{family=binomial} \) \\
\text{fit.c} & \leftarrow \text{glm}( \, y \sim C(\text{factor(cidx)},\text{sum}), \, \text{family=binomial} \)
\end{align*}
\]

\[\text{AIC(fit.r)}\]

\[
\begin{align*}
\text{AIC(fit.c)}
\end{align*}
\]
Model comparison

```
fit.0<-glm( y ~ 1, family=binomial)
fit.rc<-glm( y ~ C(factor(ridx),sum)+C(factor(cidx),sum), family=binomial)
AIC(fit.0)
## [1] 369.183
AIC(fit.rc)
## [1] 397.4326
```

This isn't too surprising:
- `fit.rc` gives a better fit, but
- `fit.rc` has $n + n - 2 = 34$ more parameters than `fit.0`.

Can you think of a better model?  
Recall `fit.rc` only seemed to improve the indegree fit.

```
fit.r<-glm( y ~ C(factor(ridx),sum), family=binomial)
fit.c<-glm( y ~ C(factor(cidx),sum), family=binomial)
AIC(fit.r)
## [1] 399.1471
AIC(fit.c)
## [1] 368.308
```
Model comparison

```
fit.0 <- glm(y ~ 1, family = binomial)
fit.rc <- glm(y ~ C(factor(ridx), sum) + C(factor(cidx), sum), family = binomial)
AIC(fit.0)
## [1] 369.183
AIC(fit.rc)
## [1] 397.4326
```

This isn't too surprising:
- `fit.rc` gives a better fit, but
- `fit.rc` has $n + n - 2 = 34$ more parameters than `fit.0`.

Can you think of a better model?
Recall `fit.rc` only seemed to improve the indegree fit.

```
fit.r <- glm(y ~ C(factor(ridx), sum), family = binomial)
fit.c <- glm(y ~ C(factor(cidx), sum), family = binomial)
AIC(fit.r)
## [1] 399.1471
AIC(fit.c)
## [1] 368.308
```
Model comparison

```r
fit.0 <- glm(y ~ 1, family=binomial)
fit.rc <- glm(y ~ C(factor(ridx),sum) + C(factor(cidx),sum), family=binomial)
AIC(fit.0)
## [1] 369.183
AIC(fit.rc)
## [1] 397.4326
```

This isn't too surprising:
- `fit.rc` gives a better fit, but
- `fit.rc` has \( n + n - 2 = 34 \) more parameters than `fit.0`.

Can you think of a better model?
Recall `fit.rc` only seemed to improve the indegree fit.

```r
fit.r <- glm(y ~ C(factor(ridx),sum), family=binomial)
fit.c <- glm(y ~ C(factor(cidx),sum), family=binomial)
AIC(fit.r)
## [1] 399.1471
AIC(fit.c)
## [1] 368.308
```
Model comparison with BIC

model performance = model fit - model complexity

- fit = maximized log likelihood
- complexity increasing in the number of parameters

An alternative complexity measure results in Bayes Information Criterion (BIC):

$$\tilde{BIC} = \log p(Y|\hat{\theta}) - \frac{1}{2} p \times \log n$$

Again, for historical reasons, this is often multiplied by $-2$:

$$BIC = -2 \log p(Y|\hat{\theta}) + p \times \log n,$$

so that a low value of the BIC is better.

As compared to the AIC,
- the BIC penalizes complexity more;
- the BIC is generally better at model selection, AIC at prediction.
Model comparison with BIC

\[
\text{model performance} = \text{model fit} - \text{model complexity}
\]

- \( \text{fit} = \text{maximized log likelihood} \)
- \( \text{complexity increasing in the number of parameters} \)

An alternative complexity measure results is \textit{Bayes Information Criterion (BIC)}:

\[
BIC = \log p(Y|\hat{\theta}) - \frac{1}{2} p \times \log n
\]

Again, for historical reasons, this is often multiplied by \(-2\):

\[
BIC = -2 \log p(Y|\hat{\theta}) + p \times \log n,
\]

so that a \textbf{low} value of the BIC is \textbf{better}.

As compared to the AIC,

- the BIC penalizes complexity more;
- the BIC is generally better at model selection, AIC at prediction.
Model comparison with BIC

model performance = model fit - model complexity

- fit = maximized log likelihood
- complexity increasing in the number of parameters

An alternative complexity measure results is Bayes Information Criterion (BIC):

\[ \hat{BIC} = \log p(Y|\hat{\theta}) - \frac{1}{2} p \times \log n \]

Again, for historical reasons, this is often multiplied by $-2$:

\[ BIC = -2 \log p(Y|\hat{\theta}) + p \times \log n, \]

so that a low value of the BIC is better.

As compared to the AIC,
- the BIC penalizes complexity more;
- the BIC is generally better at model selection, AIC at prediction.
Model comparison with BIC

model performance = model fit - model complexity

- fit = maximized log likelihood
- complexity increasing in the number of parameters

An alternative complexity measure results is Bayes Information Criterion (BIC):

\[ \tilde{BIC} = \log p(Y|\hat{\theta}) - \frac{1}{2} p \times \log n \]

Again, for historical reasons, this is often multiplied by −2:

\[ BIC = -2 \log p(Y|\hat{\theta}) + p \times \log n, \]

so that a low value of the BIC is better.

As compared to the AIC,
- the BIC penalizes complexity more;
- the BIC is generally better at model selection, AIC at prediction.
Model comparison with BIC

model performance = model fit - model complexity

- fit = maximized log likelihood
- complexity increasing in the number of parameters

An alternative complexity measure results is *Bayes Information Criterion (BIC)*:

\[
\tilde{\text{BIC}} = \log p(Y|\hat{\theta}) - \frac{1}{2} p \times \log n
\]

Again, for historical reasons, this is often multiplied by \(-2\):

\[
\text{BIC} = -2 \log p(Y|\hat{\theta}) + p \times \log n,
\]

so that a **low** value of the BIC is **better**.

As compared to the AIC,

- the BIC penalizes complexity more;
- the BIC is generally better at model selection, AIC at prediction.
Model comparison with BIC

model performance = model fit - model complexity

- fit = maximized log likelihood
- complexity increasing in the number of parameters

An alternative complexity measure results is *Bayes Information Criterion (BIC)*:

\[ \tilde{BIC} = \log p(Y|\hat{\theta}) - \frac{1}{2} p \times \log n \]

Again, for historical reasons, this is often multiplied by \(-2\):

\[ BIC = -2 \log p(Y|\hat{\theta}) + p \times \log n, \]

so that a **low** value of the BIC is **better**.

As compared to the AIC,

- the BIC penalizes complexity more;
- the BIC is generally better at model selection, AIC at prediction.
Model comparison with BIC

model performance = model fit - model complexity

- fit = maximized log likelihood
- complexity increasing in the number of parameters

An alternative complexity measure results in *Bayes Information Criterion (BIC)*:

\[
\tilde{BIC} = \log p(Y|\hat{\theta}) - \frac{1}{2} p \times \log n
\]

Again, for historical reasons, this is often multiplied by \(-2\):

\[
BIC = -2 \log p(Y|\hat{\theta}) + p \times \log n,
\]

so that a **low** value of the BIC is **better**.

As compared to the AIC,

- the BIC penalizes complexity more;
- the BIC is generally better at model selection, AIC at prediction.
Model comparison with BIC

\texttt{BIC(fit.0)}

## [1] 372.9065

\texttt{BIC(fit.rc)}

## [1] 527.7581

\texttt{BIC(fit.r)}

## [1] 466.1716

\texttt{BIC(fit.c)}

## [1] 435.3325

\textbf{Caution:} Theoretical justification of the BIC assumes one of the models considered is correct.
model fit

\[
\begin{align*}
\text{mu.hat} &\leftarrow \text{fit.c$coeff[1]} \\
\text{b.hat} &\leftarrow \text{c(fit.c$coeff[2:18],-sum(fit.c$coeff[2:18]))} \\
\text{muij.mle} &\leftarrow \text{mu.hat + outer(rep(0,n),b.hat,"+"}) \\
\text{p.mle} &\leftarrow \text{exp(muij.mle)/(1+exp(muij.mle))}
\end{align*}
\]

\[
\text{t.M}\leftarrow \text{NULL} \\
\text{for(s in 1:S)} \\
\text{\{ \}
\text{Ysim}\leftarrow \text{matrix(rbinom(n^2,1,p.mle),n,n)} \text{; diag(Ysim)}\leftarrow \text{NA} \\
\text{t.M}\leftarrow \text{rbind(t.M, c(mmean(Ysim),sd(rsum(Ysim)),sd(csum(Ysim)))})
\]

Model comparison

For these monk data,

- among these models, the column effects model has best AIC;
- all models show lack of fit in terms of outdegree heterogeneity.

The lack of fit is not surprising - we know an independence model is wrong:

- monks were limited to three nominations per time period.

Ideally, such design features of the data would be incorporated into the model. We will discuss such models towards the end of the quarter.
Model comparison

For these monk data,

- among these models, the column effects model has best AIC;
- all models show lack of fit in terms of outdegree heterogeneity.

The lack of fit is not surprising - we know an independence model is wrong:
- monks were limited to three nominations per time period.

Ideally, such design features of the data would be incorporated into the model. We will discuss such models towards the end of the quarter.
For these monk data,

- among these models, the column effects model has best AIC;
- all models show lack of fit in terms of outdegree heterogeneity.

The lack of fit is not surprising - we know an independence model is wrong:

- monks were limited to three nominations per time period.

Ideally, such design features of the data would be incorporated into the model. We will discuss such models towards the end of the quarter.
Model comparison

For these monk data,

- among these models, the column effects model has best AIC;
- all models show lack of fit in terms of outdegree heterogeneity.

The lack of fit is not surprising - we know an independence model is wrong:

- monks were limited to three nominations per time period.

Ideally, such design features of the data would be incorporated into the model. We will discuss such models towards the end of the quarter.
Model comparison

For these monk data,

• among these models, the column effects model has best AIC;
• all models show lack of fit in terms of outdegree heterogeneity.

The lack of fit is not surprising - we know an independence model is wrong:
• monks were limited to three nominations per time period.

Ideally, such design features of the data would be incorporated into the model. We will discuss such models towards the end of the quarter.
Model comparison

For these monk data,

- among these models, the column effects model has best AIC;
- all models show lack of fit in terms of outdegree heterogeneity.

The lack of fit is not surprising - we *know* an independence model is wrong:

- monks were limited to three nominations per time period.

Ideally, such design features of the data would be incorporated into the model. We will discuss such models towards the end of the quarter.
For these monk data,

- among these models, the column effects model has best AIC;
- all models show lack of fit in terms of outdegree heterogeneity.

The lack of fit is not surprising - we *know* an independence model is wrong:

- monks were limited to three nominations per time period.

Ideally, such design features of the data would be incorporated into the model.

We will discuss such models towards the end of the quarter.
For these monk data,

- among these models, the column effects model has best AIC;
- all models show lack of fit in terms of outdegree heterogeneity.

The lack of fit is not surprising - we know an independence model is wrong:

- monks were limited to three nominations per time period.

Ideally, such design features of the data would be incorporated into the model. We will discuss such models towards the end of the quarter.
Conflict in the 90s

sd(rsum(Y))
## [1] 3.589398

sd(csum(Y))
## [1] 1.984451
Model selection

```
fit.0<-glm( y ~ 1, family=binomial)
fitt.r<-glm( y ~ C(factor(ridx),sum), family=binomial)
fitt.c<-glm( y ~ C(factor(cidx),sum), family=binomial)
fitt.rc<-glm( y ~ C(factor(ridx),sum)+C(factor(cidx),sum), family=binomial)

AIC(fit.0)
## [1] 2197.674

AIC(fit.r)
## [1] 1947.604

AIC(fit.c)
## [1] 2176.021

AIC(fit.rc)
## [1] 1897.398
```

For these data, the full RCE model is best among these four.
Model selection with BIC

\[
\text{BIC}(\text{fit.0})
\]

## [1] 2205.401

\[
\text{BIC}(\text{fit.r})
\]

## [1] 2952.159

\[
\text{BIC}(\text{fit.c})
\]

## [1] 3180.576

\[
\text{BIC}(\text{fit.rc})
\]

## [1] 3898.78
Best case comparison
Row effects only

\[
\begin{align*}
\mu.hat & \leftarrow \text{fit.r$coef[1]} \\
a.hat & \leftarrow \text{fit.r$coef[1+1:(nrow(Y)-1)]} \ ; \ a.hat \leftarrow -c(a.hat,-\text{sum(a.hat)}) \\
\theta.mle & \leftarrow \mu.hat + \text{outer(a.hat,rep(0,nrow(Y)),"+")} \\
p.mle & \leftarrow \exp(\theta.mle)/(1+\exp(\theta.mle))
\end{align*}
\]
Fit comparison
Recall the “best case scenario” evaluation is somewhat ad-hoc.

Compare to the conditional evaluation:

Suppose \( Y \sim SRG(n, \theta) \):
- \( \{Y|y..\} \not\sim SRG(n, \theta) \);
- \( \{Y|y..\} \sim ERG(n, y..) \).

Similarly, suppose \( Y \sim RCE(\mu, a, b) \):
- \( \{Y|y.., \{y_i\}, \{y_i\}\} \not\sim RCE(\hat{\mu}, \hat{a}, \hat{b}) \);
- \( \{Y|y.., \{y_i\}, \{y_i\}\} \sim? \)
Conditional testing

Recall the “best case scenario” evaluation is somewhat ad-hoc.

Compare to the conditional evaluation:

Suppose $Y \sim SRG(n, \theta)$:
- $\{Y|y..\} \not\sim SRG(n, \theta)$;
- $\{Y|y..\} \sim ERG(n, y..)$.

Similarly, suppose $Y \sim RCE(\mu, a, b)$:
- $\{Y|y.., \{y_i\}, \{y_i\}\} \not\sim RCE(\hat{\mu}, \hat{a}, \hat{b})$.
- $\{Y|y.., \{y_i\}, \{y_i\}\} \sim ?$
Recall the “best case scenario” evaluation is somewhat ad-hoc.

Compare to the conditional evaluation:

Suppose $\mathbf{Y} \sim SRG(n, \theta)$:

- $\{\mathbf{Y}|y.\} \not\sim SRG(n, \theta)$;
- $\{\mathbf{Y}|y.\} \sim ERG(n, y.)$.

Similarly, suppose $\mathbf{Y} \sim RCE(\mu, a, b)$:

- $\{\mathbf{Y}|y., \{y_i\}, \{y_i\}\} \not\sim RCE(\hat{\mu}, \hat{a}, \hat{b})$;
- $\{\mathbf{Y}|y., \{y_i\}, \{y_i\}\} \sim$?
Conditional testing

Recall the “best case scenario” evaluation is somewhat ad-hoc.

Compare to the conditional evaluation:

Suppose $\mathbf{Y} \sim SRG(n, \theta)$:
- $\{\mathbf{Y} | y_{..}\} \not\sim SRG(n, \theta)$;
- $\{\mathbf{Y} | y_{..}\} \sim ERG(n, y_{..})$.

Similarly, suppose $\mathbf{Y} \sim RCE(\mu, a, b)$:
- $\{\mathbf{Y} | y_{..}, \{y_{i..}\}, \{y_{.i}\}\} \not\sim RCE(\hat{\mu}, \hat{a}, \hat{b})$.
- $\{\mathbf{Y} | y_{..}, \{y_{i..}\}, \{y_{.i}\}\} \sim ?$
Recall the “best case scenario” evaluation is somewhat ad-hoc.

Compare to the conditional evaluation:

Suppose $\mathbf{Y} \sim SRG(n, \theta)$:
- $\{\mathbf{Y}|y..\} \sim SRG(n, \theta)$;
- $\{\mathbf{Y}|y..\} \sim ERG(n, y..)$.

Similarly, suppose $\mathbf{Y} \sim RCE(\mu, a, b)$:
- $\{\mathbf{Y}|y.., \{y_{i}\}, \{y_{j}\}\} \sim RCE(\hat{\mu}, \hat{a}, \hat{b})$.
- $\{\mathbf{Y}|y.., \{y_{i}\}, \{y_{j}\}\} \sim$?
Recall the “best case scenario” evaluation is somewhat ad-hoc.

Compare to the conditional evaluation:

Suppose \( Y \sim SRG(n, \theta) \):
- \( \{Y|y.\} \nabla \sim SRG(n, \theta) \);
- \( \{Y|y.\} \sim ERG(n, y..) \).

Similarly, suppose \( Y \sim RCE(\mu, a, b) \) :
- \( \{Y|y., \{y_i\}, \{y_i\}\} \nabla \sim RCE(\hat{\mu}, \hat{a}, \hat{b}) \).
- \( \{Y|y., \{y_i\}, \{y_i\}\} \sim? \).
Recall the “best case scenario” evaluation is somewhat ad-hoc.

Compare to the conditional evaluation:

Suppose $Y \sim SRG(n, \theta)$:
- $\{Y|y_{..}\} \not\sim SRG(n, \theta)$;
- $\{Y|y_{..}\} \sim ERG(n, y_{..})$.

Similarly, suppose $Y \sim RCE(\mu, a, b)$:
- $\{Y|y_{..}, \{y_i\}, \{y_{.i}\}\} \not\sim RCE(\hat{\mu}, \hat{a}, \hat{b})$.
- $\{Y|y_{..}, \{y_i\}, \{y_{.i}\}\} \sim ?$
Recall the “best case scenario” evaluation is somewhat ad-hoc.

Compare to the conditional evaluation:

Suppose $Y \sim SRG(n, \theta)$:
- $\{Y | y..\} \not\sim SRG(n, \theta)$;
- $\{Y | y..\} \sim ERG(n, y..)$.

Similarly, suppose $Y \sim RCE(\mu, a, b)$:
- $\{Y | y.., \{y_i\}, \{y.i\}\} \not\sim RCE(\hat{\mu}, \hat{a}, \hat{b})$.
- $\{Y | y.., \{y_i\}, \{y.i\}\} \sim ?$
Recall the “best case scenario” evaluation is somewhat ad-hoc.

Compare to the conditional evaluation:

Suppose $Y \sim SRG(n, \theta)$:
- $\{Y | y.. \} \not\sim SRG(n, \theta)$;
- $\{Y | y.. \} \sim ERG(n, y..)$.

Similarly, suppose $Y \sim RCE(\mu, a, b)$:
- $\{Y|y.., \{y_i\}, \{y_i\}\} \not\sim RCE(\hat{\mu}, \hat{a}, \hat{b})$.
- $\{Y|y.., \{y_i\}, \{y_i\}\} \sim ?$
Recall the “best case scenario” evaluation is somewhat ad-hoc.

Compare to the conditional evaluation:

Suppose $\mathbf{Y} \sim SRG(n, \theta)$:
- $\{\mathbf{Y}|y_{..}\} \not\sim SRG(n, \theta)$;
- $\{\mathbf{Y}|y_{..}\} \sim ERG(n, y_{..})$.

Similarly, suppose $\mathbf{Y} \sim RCE(\mu, a, b)$:
- $\{\mathbf{Y}|y_{..}, \{y_{i.}\}, \{y_{..i}\}\} \not\sim RCE(\hat{\mu}, \hat{a}, \hat{b})$.
- $\{\mathbf{Y}|y_{..}, \{y_{i.}\}, \{y_{..i}\}\} \sim \ ?$