Social relations models for binary, ranked and ordinal relations

567 Statistical analysis of social networks

Peter Hoff

Statistics, University of Washington
Adapting the normal SRM

Relational data are often valued, non-binary.

The standard SRM is appropriate for normal valued relational data.

However, relational data is often neither binary nor normal:

- Links can be absent or continuous within the same network.
  - meaning $Y_{ij}$ is either 0 or some arbitrary real number;
  - communication networks (time spent communicating);
- Links can be absent or ordinal within the same network.
  - conflict networks (negative, positive or zero relation);
  - ranked nominations (friends are ranked, non-friends are not).

The SRM can be adapted via ordinal probit models to handle such data.
Adapting the normal SRM

Relational data are often valued, non-binary.

The standard SRM is appropriate for normal valued relational data.

However, relational data is often neither binary nor normal:

- Links can be absent or continuous within the same network.
  - meaning $Y_{ij}$ is either 0 or some arbitrary real number;
  - communication networks (time spent communicating);
- Links can be absent or ordinal within the same network.
  - conflict networks (negative, positive or zero relation);
  - ranked nominations (friends are ranked, non-friends are not).

The SRM can be adapted via ordinal probit models to handle such data.
Adapting the normal SRM

Relational data are often valued, non-binary.

The standard SRM is appropriate for normal valued relational data.

However, relational data is often neither binary nor normal:

- Links can be absent or continuous within the same network.
  - meaning $Y_{i,j}$ is either 0 or some arbitrary real number;
  - communication networks (time spent communicating);

- Links can be absent or ordinal within the same network.
  - conflict networks (negative, positive or zero relation);
  - ranked nominations (friends are ranked, non-friends are not).

The SRM can be adapted via ordinal probit models to handle such data.
Adapting the normal SRM

Relational data are often valued, non-binary.

The standard SRM is appropriate for normal valued relational data.

However, relational data is often neither binary nor normal:

- Links can be absent or continuous within the same network.
  - meaning $Y_{i,j}$ is either 0 or some arbitrary real number;
  - communication networks (time spent communicating);
- Links can be absent or ordinal within the same network.
  - conflict networks (negative, positive or zero relation);
  - ranked nominations (friends are ranked, non-friends are not).

The SRM can be adapted via ordinal probit models to handle such data.
Adapting the normal SRM

Relational data are often valued, non-binary.

The standard SRM is appropriate for **normal** valued relational data.

**However**, relational data is often neither binary nor normal:

- Links can be **absent** or **continuous** within the same network.
  - meaning $Y_{i,j}$ is either 0 or some arbitrary real number;
  - communication networks (time spent communicating);

- Links can be **absent** or **ordinal** within the same network.
  - conflict networks (negative, positive or zero relation);
  - ranked nominations (friends are ranked, non-friends are not).

The SRM can be adapted via **ordinal probit models** to handle such data.
Adapting the normal SRM

Relational data are often valued, non-binary.

The standard SRM is appropriate for normal valued relational data.

However, relational data is often neither binary nor normal:

- Links can be absent or continuous within the same network.
  - meaning $Y_{i,j}$ is either 0 or some arbitrary real number;
  - communication networks (time spent communicating);
- Links can be absent or ordinal within the same network.
  - conflict networks (negative, positive or zero relation);
  - ranked nominations (friends are ranked, non-friends are not).

The SRM can be adapted via ordinal probit models to handle such data.
Adapting the normal SRM

Relational data are often valued, non-binary.

The standard SRM is appropriate for normal valued relational data.

However, relational data is often neither binary nor normal:

- Links can be absent or continuous within the same network.
  - meaning $Y_{i,j}$ is either 0 or some arbitrary real number;
  - communication networks (time spent communicating);
- Links can be absent or ordinal within the same network.
  - conflict networks (negative, positive or zero relation);
  - ranked nominations (friends are ranked, non-friends are not).

The SRM can be adapted via ordinal probit models to handle such data.
Adapting the normal SRM

Relational data are often valued, non-binary.

The standard SRM is appropriate for normal valued relational data.

**However**, relational data is often neither binary nor normal:

- Links can be **absent** or **continuous** within the same network.
  - meaning \( Y_{i,j} \) is either 0 or some arbitrary real number;
  - communication networks (time spent communicating);
- Links can be **absent** or **ordinal** within the same network.
  - conflict networks (negative, positive or zero relation);
  - ranked nominations (friends are ranked, non-friends are not).

The SRM can be adapted via **ordinal probit models** to handle such data.
Adapting the normal SRM

Relational data are often valued, non-binary.

The standard SRM is appropriate for **normal** valued relational data.

**However**, relational data is often neither binary nor normal:

- Links can be **absent** or **continuous** within the same network.
  - meaning $Y_{i,j}$ is either 0 or some arbitrary real number;
  - communication networks (time spent communicating);
- Links can be **absent** or **ordinal** within the same network.
  - conflict networks (negative, positive or zero relation);
  - ranked nominations (friends are ranked, non-friends are not).

The SRM can be adapted via **ordinal probit models** to handle such data.
Adapting the normal SRM

Relational data are often valued, non-binary.

The standard SRM is appropriate for normal valued relational data.

However, relational data is often neither binary nor normal:

- Links can be absent or continuous within the same network.
  - meaning $Y_{i,j}$ is either 0 or some arbitrary real number;
  - communication networks (time spent communicating);
- Links can be absent or ordinal within the same network.
  - conflict networks (negative, positive or zero relation);
  - ranked nominations (friends are ranked, non-friends are not).

The SRM can be adapted via ordinal probit models to handle such data.
Ordinal data

An **ordinal variable** has a meaningful ordering to the possible outcomes.

This is in contrast to a **categorical** (non-ordered) variable.

**Ordinal variable**
- continuous: (all real numbers)
- discrete: (counts, ranks, etc.)

**Categorical variable**
- non-orderable categories (religion, ethnicity, etc.)
An **ordinal variable** has a meaningful ordering to the possible outcomes. This is in contrast to a **categorical** (non-ordered) variable.

**Ordinal variable**
- continuous: (all real numbers)
- discrete: (counts, ranks, etc.)

**Categorical variable**
- non-orderable categories (religion, ethnicity, etc.)
Ordinal data

An *ordinal variable* has a meaningful ordering to the possible outcomes.

This is in contrast to a *categorical* (non-ordered) variable.

**Ordinal variable**
- continuous: (all real numbers)
- discrete: (counts, ranks, etc.)

**Categorical variable**
- non-orderable categories (religion, ethnicity, etc.)
An **ordinal variable** has a meaningful ordering to the possible outcomes. This is in contrast to a **categorical** (non-ordered) variable.

**Ordinal variable**
- continuous: (all real numbers)
- discrete: (counts, ranks, etc.)

**Categorical variable**
- non-orderable categories (religion, ethnicity, etc.)
Ordinal data

An **ordinal variable** has a meaningful ordering to the possible outcomes. This is in contrast to a **categorical** (non-ordered) variable.

**Ordinal variable**
- continuous: (all real numbers)
- discrete: (counts, ranks, etc.)

**Categorical variable**
- non-orderable categories (religion, ethnicity, etc.)
An **ordinal variable** has a meaningful ordering to the possible outcomes. This is in contrast to a **categorical** (non-ordered) variable.

**Ordinal variable**
- continuous: (all real numbers)
- discrete: (counts, ranks, etc.)

**Categorical variable**
- non-orderable categories (religion, ethnicity, etc.)
Ordinal data

An **ordinal variable** has a meaningful ordering to the possible outcomes. This is in contrast to a **categorical** (non-ordered) variable.

**Ordinal variable**
- continuous: (all real numbers)
- discrete: (counts, ranks, etc.)

**Categorical variable**
- non-orderable categories (religion, ethnicity, etc.)
Ordinal data

An **ordinal variable** has a meaningful ordering to the possible outcomes. This is in contrast to a **categorical** (non-ordered) variable.

**Ordinal variable**
- continuous: (all real numbers)
- discrete: (counts, ranks, etc.)

**Categorical variable**
- non-orderable categories (religion, ethnicity, etc.)
An **ordinal variable** has a meaningful ordering to the possible outcomes. This is in contrast to a **categorical** (non-ordered) variable.

**Ordinal variable**
- continuous: (all real numbers)
- discrete: (counts, ranks, etc.)

**Categorical variable**
- non-orderable categories (religion, ethnicity, etc.)
The simplest ordinal variable is a binary random variable $Y \in \{0, 1\}$. Let

$$\begin{align*}
\Pr(Y = 1) &= \theta \\
\Pr(Y = 0) &= 1 - \theta
\end{align*}$$

This model for $Y$ has the following latent variable representation:

$$\begin{align*}
\mu &= \Phi^{-1}(\theta) \\
Z &\sim N(\mu, 1) \\
Y &= 1 \times (Z > 0)
\end{align*}$$

Here, $\Phi^{-1}(\theta)$ is the $\theta$-quantile of the standard normal distribution.
The simplest ordinal variable is a binary random variable \( Y \in \{0, 1\} \). Let

\[
\Pr(Y = 1) = \theta \\
\Pr(Y = 0) = 1 - \theta
\]

This model for \( Y \) has the following **latent variable representation**:

\[
\mu = \Phi^{-1}(\theta) \\
Z \sim N(\mu, 1) \\
Y = 1 \times (Z > 0)
\]

Here, \( \Phi^{-1}(\theta) \) is the \( \theta \)-quantile of the standard normal distribution.
Probit representation

To confirm the representation, recall

- If $Z \sim N(\mu, 1)$ then $Z - \mu \sim N(0, 1)$;
- If $Z \sim N(\mu, 1)$ then $Z = \mu + \epsilon$, where $\epsilon \sim N(0, 1)$;
- If $\epsilon \sim N(0, 1)$ then $-\epsilon \sim N(0, 1)$.

Now do the calculation:

$$\Pr(Y = 1) = \Pr(Z > 0)$$
$$= \Pr(-Z < 0)$$
$$= \Pr([-Z + \mu] < \mu)$$
$$= \Pr(\epsilon < \mu) = \Phi(\mu) = \theta$$
To confirm the representation, recall

- If \( Z \sim N(\mu, 1) \) then \( Z - \mu \sim N(0, 1) \);
- If \( Z \sim N(\mu, 1) \) then \( Z = \mu + \epsilon \), where \( \epsilon \sim N(0, 1) \);
- If \( \epsilon \sim N(0, 1) \) then \(-\epsilon \sim N(0, 1)\).

Now do the calculation:

\[
\Pr(Y = 1) = \Pr(Z > 0) \\
= \Pr(-Z < 0) \\
= \Pr([-Z + \mu] < \mu) \\
= \Pr(\epsilon < \mu) = \Phi(\mu) = \theta
\]
Probit representation

To confirm the representation, recall

- If $Z \sim N(\mu, 1)$ then $Z - \mu \sim N(0, 1)$;
- If $Z \sim N(\mu, 1)$ then $Z = \mu + \epsilon$, where $\epsilon \sim N(0, 1)$;
- If $\epsilon \sim N(0, 1)$ then $-\epsilon \sim N(0, 1)$.

Now do the calculation:

\[
\Pr(Y = 1) = \Pr(Z > 0) \\
= \Pr(-Z < 0) \\
= \Pr([-Z + \mu] < \mu) \\
= \Pr(\epsilon < \mu) = \Phi(\mu) = \theta
\]
To confirm the representation, recall

- If $Z \sim N(\mu, 1)$ then $Z - \mu \sim N(0, 1)$;
- If $Z \sim N(\mu, 1)$ then $Z = \mu + \epsilon$, where $\epsilon \sim N(0, 1)$;
- If $\epsilon \sim N(0, 1)$ then $-\epsilon \sim N(0, 1)$.

Now do the calculation:

$$
\Pr(Y = 1) = \Pr(Z > 0)
= \Pr(-Z < 0)
= \Pr([-Z + \mu] < \mu)
= \Pr(\epsilon < \mu) = \Phi(\mu) = \theta
$$
To confirm the representation, recall

- If $Z \sim N(\mu, 1)$ then $Z - \mu \sim N(0, 1)$;
- If $Z \sim N(\mu, 1)$ then $Z = \mu + \epsilon$, where $\epsilon \sim N(0, 1)$;
- If $\epsilon \sim N(0, 1)$ then $-\epsilon \sim N(0, 1)$.

Now do the calculation:

$$\Pr(Y = 1) = \Pr(Z > 0) = \Pr(-Z < 0) = \Pr([-Z + \mu] < \mu) = \Pr(\epsilon < \mu) = \Phi(\mu) = \theta$$
Probit representation

\[ \theta = .15, \mu = \Phi^{-1}(.15) = -1.04 \]
Probit representation

\[ \theta = 0.15, \mu = \Phi^{-1}(0.15) = -1.04 \]
Probit regression

Now suppose we have binary data $Y_1, \ldots, Y_n$ we want to relate to explanatory variables $x_1, \ldots, x_n$

Latent variable model:

$$\epsilon_1, \ldots, \epsilon_n \sim \text{i.i.d. } N(0,1)$$

$$Z_i = \beta^T x_i + \epsilon_i$$

$$Y_i = 1 \times (Z_i > 0)$$

Under this latent variable model, the $Y_i$'s are independent and

$$Pr(Y_i = 1) = Pr(Z_i > 0)$$

$$= Pr(\beta^T x_i + \epsilon_i > 0)$$

$$= Pr(-\epsilon_i < \beta^T x_i)$$

$$= Pr(\epsilon_i < \beta^T x_i) = \Phi(\beta^T x_i)$$
Probit regression

Now suppose we have binary data $Y_1, \ldots, Y_n$ we want to relate to explanatory variables $x_1, \ldots, x_n$

**Latent variable model:**

$$\epsilon_1, \ldots, \epsilon_n \sim \text{i.i.d. } N(0, 1)$$

$$Z_i = \beta^T x_i + \epsilon_i$$

$$Y_i = 1 \times (Z_i > 0)$$

Under this latent variable model, the $Y_i$’s are independent and

$$\Pr(Y_i = 1) = \Pr(Z_i > 0)$$

$$= \Pr(\beta^T x_i + \epsilon_i > 0)$$

$$= \Pr(-\epsilon_i < \beta^T x_i)$$

$$= \Pr(\epsilon_i < \beta^T x_i) = \Phi(\beta^T x_i)$$
Probit regression

Now suppose we have binary data $Y_1, \ldots, Y_n$ we want to relate to explanatory variables $x_1, \ldots, x_n$

Latent variable model:

$$
\epsilon_1, \ldots, \epsilon_n \sim \text{i.i.d. } N(0,1) \\
Z_i = \beta^T x_i + \epsilon_i \\
Y_i = 1 \times (Z_i > 0)
$$

Under this latent variable model, the $Y_i$'s are independent and

$$
\Pr(Y_i = 1) = \Pr(Z_i > 0) \\
= \Pr(\beta^T x_i + \epsilon_i > 0) \\
= \Pr(-\epsilon_i < \beta^T x_i) \\
= \Pr(\epsilon_i < \beta^T x_i) = \Phi(\beta^T x_i)
$$
This latent variable model is exactly the same as **probit regression**:

$$
\Pr(Y_1 = y_1, \ldots, Y_n = y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} \Phi(\beta^T x_i)^{y_i} [1 - \Phi(\beta^T x_i)]^{1-y_i}
$$

Compare to **logistic regression**:

$$
\Pr(Y_1 = y_1, \ldots, Y_n = y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} \left( \frac{e^{\beta^T x_i}}{1 + e^{\beta^T x_i}} \right)^{y_i} \left( \frac{1}{1 + e^{\beta^T x_i}} \right)^{1-y_i}
$$

In fact, logistic regression has a latent variable representation also:

$$
\epsilon_1, \ldots, \epsilon_n \sim \text{i.i.d. } L(0, 1) \\
Z_i = \beta^T x_i + \epsilon_i \\
Y_i = 1 \times (Z_i > 0)
$$
This latent variable model is exactly the same as **probit regression**:

\[
\Pr(Y_1 = y_1, \ldots, Y_n = y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} \Phi(\beta^T x_i)^{y_i} [1 - \Phi(\beta^T x_i)]^{1-y_i}
\]

Compare to **logistic regression**:

\[
\Pr(Y_1 = y_1, \ldots, Y_n = y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} \left( \frac{e^{\beta^T x_i}}{1 + e^{\beta^T x_i}} \right)^{y_i} \left( \frac{1}{1 + e^{\beta^T x_i}} \right)^{1-y_i}
\]

In fact, logistic regression has a latent variable representation also:

\[
\epsilon_1, \ldots, \epsilon_n \sim \text{i.i.d. } L(0, 1) \\
Z_i = \beta^T x_i + \epsilon_i \\
Y_i = 1 \times (Z_i > 0)
\]
Probit regression

This latent variable model is exactly the same as **probit regression**:

\[
\Pr(Y_1 = y_1, \ldots, Y_n = y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} \Phi(\beta^T x_i)^{y_i} [1 - \Phi(\beta^T x_i)]^{1-y_i}
\]

Compare to **logistic regression**:

\[
\Pr(Y_1 = y_1, \ldots, Y_n = y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} \left( 1 + e^{\beta^T x_i} \right)^{y_i} \left( \frac{1}{1 + e^{\beta^T x_i}} \right)^{1-y_i}
\]

In fact, logistic regression has a latent variable representation also:

\[
\epsilon_1, \ldots, \epsilon_n \sim \text{i.i.d. } L(0, 1)
\]

\[
Z_i = \beta^T x_i + \epsilon_i
\]

\[
Y_i = 1 \times (Z_i > 0)
\]
Logistic versus probit regression

Sheep dominance data
- counts of dominance encounters between 28 male sheep;
- age of each sheep, in years.

Let’s examine the effect of age difference on dominance.
Let’s examine the effect of age difference on dominance.

Sheep dominance data
- counts of dominance encounters between 28 male sheep;
- age of each sheep, in years.
Logistic versus probit regression

Sheep dominance data
- counts of dominance encounters between 28 male sheep;
- age of each sheep, in years.

Let’s examine the effect of age difference on dominance.
Sheep dominance data

- counts of dominance encounters between 28 male sheep;
- age of each sheep, in years.

Let’s examine the effect of age difference on dominance.
Sheep dominance data

Let

- \( \text{dom}[i,j] = \) indicator that \( i \) has dominated \( j \) at least once;
- \( \text{aged}[i,j] = \text{age}_i - \text{age}_j \).

```R
mean(aged[dom==1],na.rm=TRUE )
## [1] 2.184
mean(aged[dom==0],na.rm=TRUE )
## [1] -1.079051
```

```R
summary( glm(dom~aged,family=binomial) )$coef
```

## Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.8357694 0.08729001 -9.574628 1.022245e-21
## aged 0.2260270 0.02317097 9.754752 1.760303e-22

```R
summary( glm(dom~aged,family=binomial(link=probit)) )$coef
```

## Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.5138332 0.05111585 -10.05232 8.972511e-24
## aged 0.1385751 0.01337034 10.36437 3.601223e-25
Sheep dominance data

Let

- \( \text{dom}[i,j] = \) indicator that \( i \) has dominated \( j \) at least once;
- \( \text{aged}[i,j] = \text{age}_i - \text{age}_j. \)

```r
mean(aged[dom==1], na.rm=TRUE)
## [1] 2.184

mean(aged[dom==0], na.rm=TRUE)
## [1] -1.079051
```

```r
summary( glm(dom~aged, family=binomial) )$coef
## (Intercept)    aged
## -0.836        0.226

summary( glm(dom~aged, family=binomial(link=probit)) )$coef
## (Intercept)    aged
## -0.514        0.139
```
Logit, probit and other binary regression models

Comparing logistic and probit regression output:

- $\hat{β}$'s are on different scales;
- inference (z-scores) are typically similar.

Both models are binary regression models of the form:

$$\Pr(Y_1 = y_1, \ldots, Y_n = y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} g(\beta^T x_i)^{y_i} [1 - g(\beta^T x_i)]^{1-y_i},$$

where $g^{-1}$ is called the inverse-link function.

Link functions:

- logistic regression: $g^{-1}(μ) = \exp(μ)/[1 + \exp(μ)];$
- probit regression: $g^{-1}(μ) = Φ(μ);$
- other binary regression: $g^{-1}(μ)$ is a strictly increasing function.
Comparing logistic and probit regression output:

- $\hat{\beta}$’s are on different scales;
- inference ($z$-scores) are typically similar.

Both models are binary regression models of the form:

$$
\Pr(Y_1 = y_1, \ldots, Y_n = y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} g(\beta^T x_i)^{y_i} \left[ 1 - g(\beta^T x_i) \right]^{1-y_i},
$$

where $g^{-1}$ is called the inverse-link function.

Link functions:

- logistic regression: $g^{-1}(\mu) = \exp(\mu)/[1 + \exp(\mu)]$;
- probit regression: $g^{-1}(\mu) = \Phi(\mu)$;
- other binary regression: $g^{-1}(\mu)$ is a strictly increasing function.
Logit, probit and other binary regression models

Comparing logistic and probit regression output:

- $\hat{\beta}$'s are on different scales;
- inference ($z$-scores) are typically similar.

Both models are binary regression models of the form:

$$\Pr(Y_1 = y_1, \ldots, Y_n = y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} \left[ g(\beta^T x_i)^{y_i} [1 - g(\beta^T x_i)]^{1-y_i} \right],$$

where $g^{-1}$ is called the inverse-link function.

Link functions:

- logistic regression: $g^{-1}(\mu) = \exp(\mu)/[1 + \exp(\mu)];$
- probit regression: $g^{-1}(\mu) = \Phi(\mu);$
- other binary regression: $g^{-1}(\mu)$ is a strictly increasing function.
Logit, probit and other binary regression models

Comparing logistic and probit regression output:

- $\hat{\beta}$’s are on different scales;
- inference (z-scores) are typically similar.

Both models are binary regression models of the form:

$$\Pr(Y_1 = y_1, \ldots, Y_n = y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} g(\beta^T x_i)^{y_i} [1 - g(\beta^T x_i)]^{1-y_i},$$

where $g^{-1}$ is called the inverse-link function.

Link functions:

- logistic regression: $g^{-1}(\mu) = \exp(\mu) /[1 + \exp(\mu)]$;
- probit regression: $g^{-1}(\mu) = \Phi(\mu)$;
- other binary regression: $g^{-1}(\mu)$ is a strictly increasing function.
Comparing logistic and probit regression output:

- $\hat{\beta}$'s are on different scales;
- inference (z-scores) are typically similar.

Both models are **binary regression models** of the form:

$$\Pr(Y_1 = y_1, \ldots, Y_n = y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} g(\beta^T x_i)^{y_i} [1 - g(\beta^T x_i)]^{1-y_i},$$

where $g^{-1}$ is called the **inverse-link function**.

**Link functions:**

- logistic regression: $g^{-1}(\mu) = \exp(\mu) / [1 + \exp(\mu)];$
- probit regression: $g^{-1}(\mu) = \Phi(\mu);$
- other binary regression: $g^{-1}(\mu)$ is a strictly increasing function.
Logit, probit and other binary regression models

Comparing logistic and probit regression output:

- $\hat{\beta}$’s are on different scales;
- inference (z-scores) are typically similar.

Both models are **binary regression models** of the form:

$$\Pr(Y_1 = y_1, \ldots, Y_n = y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} g(\beta^T x_i)^{y_i} [1 - g(\beta^T x_i)]^{1-y_i},$$

where $g^{-1}$ is called the **inverse-link function**.

**Link functions:**

- **logistic regression:** $g^{-1}(\mu) = \exp(\mu)/[1 + \exp(\mu)];$
- **probit regression:** $g^{-1}(\mu) = \Phi(\mu);$  
- **other binary regression:** $g^{-1}(\mu)$ is a strictly increasing function.
Logit, probit and other binary regression models

Comparing logistic and probit regression output:
• $\hat{\beta}$’s are on different scales;
• inference (z-scores) are typically similar.

Both models are **binary regression models** of the form:

$$
Pr(Y_1 = y_1, \ldots, Y_n = y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} \frac{g(\beta^T x_i)^{y_i} [1 - g(\beta^T x_i)]^{1-y_i}}{1 - y_i},
$$

where $g^{-1}$ is called the **inverse-link function**.

**Link functions:**
• **logistic regression**: $g^{-1}(\mu) = \exp(\mu) / [1 + \exp(\mu)];$
• **probit regression**: $g^{-1}(\mu) = \Phi(\mu);$ 
• **other binary regression**: $g^{-1}(\mu)$ is a strictly increasing function.
Comparing logistic and probit regression output:

- $\hat{\beta}$’s are on different scales;
- inference (z-scores) are typically similar.

Both models are binary regression models of the form:

$$\Pr(Y_1 = y_1, \ldots, Y_n = y_n|x_1, \ldots, x_n) = \prod_{i=1}^{n} g(\beta^T x_i)^{y_i} [1 - g(\beta^T x_i)]^{1-y_i},$$

where $g^{-1}$ is called the inverse-link function.

Link functions:

- logistic regression: $g^{-1}(\mu) = \exp(\mu) / [1 + \exp(\mu)];$
- probit regression: $g^{-1}(\mu) = \Phi(\mu);$
- other binary regression: $g^{-1}(\mu)$ is a strictly increasing function.
Comparing logistic and probit regression output:

- $\hat{\beta}$’s are on different scales;
- inference ($z$-scores) are typically similar.

Both models are **binary regression models** of the form:

$$
Pr(Y_1 = y_1, \ldots, Y_n = y_n | x_1, \ldots, x_n) = \prod_{i=1}^{n} g(\beta^T x_i)^{y_i} [1 - g(\beta^T x_i)]^{1-y_i},
$$

where $g^{-1}$ is called the **inverse-link function**.

**Link functions:**

- **logistic regression:** $g^{-1}(\mu) = \exp(\mu)/[1 + \exp(\mu)]$;
- **probit regression:** $g^{-1}(\mu) = \Phi(\mu)$;
- **other binary regression:** $g^{-1}(\mu)$ is a strictly increasing function.
Comparing logistic and probit regression output:

- $\hat{\beta}$’s are on different scales;
- inference (z-scores) are typically similar.

Both models are **binary regression models** of the form:

$$\Pr(Y_1 = y_1, \ldots, Y_n = y_n|x_1, \ldots, x_n) = \prod_{i=1}^{n} g(\beta^T x_i)^{y_i} [1 - g(\beta^T x_i)]^{1-y_i},$$

where $g^{-1}$ is called the **inverse-link function**.

**Link functions:**

- **logistic regression**: $g^{-1}(\mu) = \exp(\mu)/[1 + \exp(\mu)];$
- **probit regression**: $g^{-1}(\mu) = \Phi(\mu);$
- **other binary regression**: $g^{-1}(\mu)$ is a strictly increasing function.
Recall the latent variable representation of probit regression:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]

We could estimate \( \beta \) with a probit regression analysis, but what about network dependence in the data?

Could there be

- across-row heterogeneity/within row correlation?
- across-column heterogeneity/within column correlation?
- within dyad correlation?
SRM for binary relational data

Recall the latent variable representation of probit regression:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]

\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]

We could estimate \( \beta \) with a probit regression analysis, but what about network dependence in the data?

Could there be

• across-row heterogeneity/within row correlation?
• across-column heterogeneity/within column correlation?
• within dyad correlation?
Recall the latent variable representation of probit regression:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]

We could estimate \( \beta \) with a probit regression analysis, but what about network dependence in the data?

Could there be

- across-row heterogeneity/within row correlation?
- across-column heterogeneity/within column correlation?
- within dyad correlation?
SRM for binary relational data

Recall the latent variable representation of probit regression:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]

We could estimate \( \beta \) with a probit regression analysis, but what about network dependence in the data?

Could there be

- across-row heterogeneity/within row correlation?
- across-column heterogeneity/within column correlation?
- within dyad correlation?
Recall the latent variable representation of probit regression:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]

\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]

We could estimate \( \beta \) with a probit regression analysis, but what about network dependence in the data?

Could there be

- across-row heterogeneity/within row correlation?
- across-column heterogeneity/within column correlation?
- within dyad correlation?
SRM for binary relational data

Recall the latent variable representation of probit regression:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]

We could estimate \( \beta \) with a probit regression analysis, but what about network dependence in the data?

Could there be

- across-row heterogeneity/within row correlation?
- across-column heterogeneity/within column correlation?
- within dyad correlation?
SRM for binary relational data

Recall the latent variable representation of probit regression:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]

Dependence on the \( Y \)-scale can be induced by dependence on the \( Z \)-scale:

\[ \epsilon_{i,j} = a_i + b_j + \epsilon_{i,j} \]
\[ \{(a_1, b_1), \ldots, (a_n, b_n)\} \sim \text{i.i.d. } N(0, \Sigma_{ab}) \]
\[ \{(\epsilon_{i,j}, \epsilon_{j,i}) : i \neq j\} \sim \text{i.i.d. } N(0, \Sigma_e) \]

\[ \Sigma_{ab} = \begin{pmatrix} \sigma_a^2 & \sigma_{ab} \\ \sigma_{ab} & \sigma_b^2 \end{pmatrix} \]
\[ \Sigma_e = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \]

Note that the variance of \( \epsilon_{i,j} \) is fixed at 1.

The scales of \( \epsilon_{i,j} \) and \( \beta \) are not separately identifiable.
Recall the latent variable representation of probit regression:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]

Dependence on the \( Y \)-scale can be induced by dependence on the \( Z \)-scale:

\[ \epsilon_{i,j} = a_i + b_j + e_{i,j} \]
\[ \{(a_1, b_1), \ldots, (a_n, b_n)\} \sim \text{i.i.d. } N(0, \Sigma_{ab}) \]
\[ \{(e_{i,j}, e_{j,i}) : i \neq j\} \sim \text{i.i.d. } N(0, \Sigma_e) \]

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

Note that the variance of \( e_{i,j} \) is fixed at 1.

The scales of \( e_{i,j} \) and \( \beta \) are not separately identifiable.
SRM for binary relational data

Recall the latent variable representation of probit regression:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]

\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]

Dependence on the \( Y \)-scale can be induced by dependence on the \( Z \)-scale:

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

\[ \{(a_1, b_1), \ldots, (a_n, b_n)\} \sim \text{i.i.d. } N(0, \Sigma_{ab}) \]

\[ \{(e_{i,j}, e_{j,i}) : i \neq j\} \sim \text{i.i.d. } N(0, \Sigma_e) \]

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

Note that the variance of \( e_{i,j} \) is fixed at 1.

The scales of \( e_{i,j} \) and \( \beta \) are not separately identifiable.
Recall the latent variable representation of probit regression:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]

Dependence on the \( Y \)-scale can be induced by dependence on the \( Z \)-scale:

\[ \epsilon_{i,j} = a_i + b_j + e_{i,j} \]
\[ \{(a_1, b_1), \ldots, (a_n, b_n)\} \sim \text{i.i.d. } N(0, \Sigma_{ab}) \]
\[ \{(e_{i,j}, e_{j,i}) : i \neq j\} \sim \text{i.i.d. } N(0, \Sigma_e) \]

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

Note that the variance of \( e_{i,j} \) is fixed at 1.

The scales of \( e_{i,j} \) and \( \beta \) are not separately identifiable.
SRM for binary relational data

**SRM probit model:**

\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]
\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ \epsilon_{i,j} = a_i + b_j + e_{i,j}, \]

and \( \{a_i, b_j, e_{i,j}\} \) are random effects as described previously.

**Parameter estimation:**

- The likelihood can’t be expressed in closed form;
- Bayesian parameter estimates can be obtained via MCMC.

The latter is provided in the package amen.
SRM for binary relational data

**SRM probit model:**

\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]
\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ \epsilon_{i,j} = a_i + b_j + e_{i,j}, \]

and \(\{a_i, b_j, e_{i,j}\}\) are random effects as described previously.

**Parameter estimation:**

- The likelihood can’t be expressed in closed form;
- Bayesian parameter estimates can be obtained via MCMC.

The latter is provided in the package `amen`. 
SRM for binary relational data

SRM probit model:

\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]
\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ \epsilon_{i,j} = a_i + b_j + e_{i,j}, \]

and \( \{a_i, b_j, e_{i,j}\} \) are random effects as described previously.

Parameter estimation:

- The likelihood can’t be expressed in closed form;
- Bayesian parameter estimates can be obtained via MCMC.

The latter is provided in the package \texttt{amen}.
SRM for binary relational data

SRM probit model:

\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]
\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ \epsilon_{i,j} = a_i + b_j + e_{i,j}, \]

and \( \{a_i, b_j, e_{i,j}\} \) are random effects as described previously.

Parameter estimation:

- The likelihood can’t be expressed in closed form;
- Bayesian parameter estimates can be obtained via MCMC.

The latter is provided in the package amen.
SRM for binary relational data

SRM probit model:

\[ Y_{i,j} = 1 \times (Z_{i,j} > 0) \]
\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ \epsilon_{i,j} = a_i + b_j + e_{i,j}, \]

and \( \{a_i, b_j, e_{i,j}\} \) are random effects as described previously.

Parameter estimation:
- The likelihood can’t be expressed in closed form;
- Bayesian parameter estimates can be obtained via MCMC.

The latter is provided in the package amen.
SRM probit in amen

\[ \text{XD} <- \text{outer}(X, X, ",") \]

\[
\text{fit}_\text{ame}_\text{bin} <- \text{ame}(YB, XD, \text{model} = ",\text{bin}\")
\]

---

The images show various visualizations of data, including histograms and scatter plots, with axes labeled appropriately.
The summary command provides a canned summary of the fitted model:

```r
summary(fit_ame_bin)
```

```
##
## beta:
## pmean   psd  z-stat    p-val
## intercept -0.654 0.149 -4.374 0
## .dyad     0.200 0.034  5.835 0

## Sigma_ab pmean:
## a   b
## a  0.603 -0.057
## b -0.057  0.172

## rho pmean:
## -0.359
```
Confidence intervals can be obtained via the `quantile` command:

```r
apply( fit_ame_bin$BETA , 2 , quantile, prob=c(.025,.5, .975))
## intercept .dyad
## 2.5% -0.9423416 0.1398794
## 50%  -0.6554213 0.1985873
## 97.5% -0.4028070 0.2637526

apply( fit_ame_bin$SABR , 2 , quantile, prob=c(.025,.5, .975))
## va    cab     vb    rho    ve
## 2.5%  0.3111828 -0.26359446 0.07500074 -0.5818362 1
## 50%   0.5511227 -0.05253038 0.15423887 -0.3646930 1
## 97.5% 1.2468605  0.09875416 0.33123579 -0.1495458 1
```
Summary of results:

Strong evidence of an age effect:
- Older sheep dominate younger ones.

Row and column heterogeneity is not substantial:
- Compare $\hat{\sigma}_a^2 = 0.55$ and $\hat{\sigma}_b^2 = 0.15$ to the error variance $\sigma_e^2 = 1$.

There is evidence that dominance tends to go one-way in a dyad:
- The 95% CI for $\rho$ is $(-0.58, -0.15)$. 
**Summary of results:**

**Strong evidence of an age effect:**
- Older sheep dominate younger ones.

Row and column heterogeneity is not substantial:
- Compare $\hat{\sigma}_a^2 = 0.55$ and $\hat{\sigma}_b^2 = 0.15$ to the error variance $\sigma_e^2 = 1$.

There is evidence that dominance tends to go one-way in a dyad:
- The 95% CI for $\rho$ is $(-0.58, -0.15)$. 
Summary of results:

Strong evidence of an age effect:

• Older sheep dominate younger ones.

Row and column heterogeneity is not substantial:

• Compare $\hat{\sigma}_a^2 = 0.55$ and $\hat{\sigma}_b^2 = 0.15$ to the error variance $\sigma_e^2 = 1$.

There is evidence that dominance tends to go one-way in a dyad:

• The 95% CI for $\rho$ is $(-0.58, -0.15)$. 
Summary of results:

Strong evidence of an age effect:
- Older sheep dominate younger ones.

Row and column heterogeneity is not substantial:
- Compare $\hat{\sigma}_a^2 = 0.55$ and $\hat{\sigma}_b^2 = 0.15$ to the error variance $\sigma_e^2 = 1$.

There is evidence that dominance tends to go one-way in a dyad:
- The 95% CI for $\rho$ is $(-0.58, -0.15)$. 
Summary of results:

Strong evidence of an age effect:

- Older sheep dominate younger ones.

Row and column heterogeneity is not substantial:

- Compare $\hat{\sigma}_a^2 = 0.55$ and $\hat{\sigma}_b^2 = 0.15$ to the error variance $\sigma_e^2 = 1$.

There is evidence that dominance tends to go one-way in a dyad:

- The 95% CI for $\rho$ is $(-0.58, -0.15)$. 
Summary of results:

Strong evidence of an age effect:
- Older sheep dominate younger ones.

Row and column heterogeneity is not substantial:
- Compare $\hat{\sigma}_a^2 = 0.55$ and $\hat{\sigma}_b^2 = 0.15$ to the error variance $\sigma_e^2 = 1$.

There is evidence that dominance tends to go one-way in a dyad:
- The 95% CI for $\rho$ is $(-0.58, -0.15)$. 
Summary of results:

Strong evidence of an age effect:
- Older sheep dominate younger ones.

Row and column heterogeneity is not substantial:
- Compare $\hat{\sigma}_a^2 = 0.55$ and $\hat{\sigma}_b^2 = 0.15$ to the error variance $\sigma_e^2 = 1$.

There is evidence that dominance tends to go one-way in a dyad:
- The 95% CI for $\rho$ is $(-0.58, -0.15)$. 
Summary of results:

Strong evidence of an age effect:
- Older sheep dominate younger ones.

Row and column heterogeneity is not substantial:
- Compare $\hat{\sigma}_a^2 = 0.55$ and $\hat{\sigma}_b^2 = 0.15$ to the error variance $\sigma_e^2 = 1$.

There is evidence that dominance tends to go one-way in a dyad:
- The 95% CI for $\rho$ is $(-0.58, -0.15)$. 
Goodness of fit

The default amen plot provides four goodness of fit plots:

- Outdegree distribution: Comparing outdegree distribution to simulated;
- Indegree distribution: Comparing indegree distribution to simulated;
- Reciprocity: Comparing fraction reciprocated ties to simulated fraction;
- Transitivity: Comparing number of triangles to simulated number.

These statistics are computed with the commands

- \texttt{t\_degree}
- \texttt{t\_recip}
- \texttt{t\_trans}
The default amen plot provides four goodness of fit plots:

Outdegree distribution: Comparing outdegree distribution to simulated;
Indegree distribution: Comparing indegree distribution to simulated;
Reciprocity: Comparing fraction reciprocated ties to simulated fraction;
Transitivity: Comparing number of triangles to simulated number.

These statistics are computed with the commands

t_degree
t_recip
t_trans
The default `amen` plot provides four goodness of fit plots:

**Outdegree distribution**: Comparing outdegree distribution to simulated;
**Indegree distribution**: Comparing indegree distribution to simulated;
**Reciprocity**: Comparing fraction reciprocated ties to simulated fraction;
**Transitivity**: Comparing number of triangles to simulated number.

These statistics are computed with the commands

t_degree
t_recip
t_trans
Goodness of fit

The default amen plot provides four goodness of fit plots:

**Outdegree distribution**: Comparing outdegree distribution to simulated;
**Indegree distribution**: Comparing indegree distribution to simulated;
**Reciprocity**: Comparing fraction reciprocated ties to simulated fraction;
**Transitivity**: Comparing number of triangles to simulated number.

These statistics are computed with the commands:

```
t_degree
```
```
t_recip
```
```
t_trans
```
Goodness of fit

The default amen plot provides four goodness of fit plots:

**Outdegree distribution:** Comparing outdegree distribution to simulated;
**Indegree distribution:** Comparing indegree distribution to simulated;
**Reciprocity:** Comparing fraction reciprocated ties to simulated fraction;
**Transitivity:** Comparing number of triangles to simulated number.

These statistics are computed with the commands

t_degree
t_recip
t_trans
Goodness of fit

The default amen plot provides four goodness of fit plots:

**Outdegree distribution:** Comparing outdegree distribution to simulated;

**Indegree distribution:** Comparing indegree distribution to simulated;

**Reciprocity:** Comparing fraction reciprocated ties to simulated fraction;

**Transitivity:** Comparing number of triangles to simulated number.

These statistics are computed with the commands

\[
t\_\text{degree} \\
t\_\text{recip} \\
t\_\text{trans}
\]
The default amen plot provides four goodness of fit plots:

**Outdegree distribution:** Comparing outdegree distribution to simulated;
**Indegree distribution:** Comparing indegree distribution to simulated;
**Reciprocity:** Comparing fraction reciprocated ties to simulated fraction;
**Transitivity:** Comparing number of triangles to simulated number.

These statistics are computed with the commands

t_degree
t_recip
t_trans
Goodness of fit

The default amen plot provides four goodness of fit plots:

**Outdegree distribution**: Comparing outdegree distribution to simulated;
**Indegree distribution**: Comparing indegree distribution to simulated;
**Reciprocity**: Comparing fraction reciprocated ties to simulated fraction;
**Transitivity**: Comparing number of triangles to simulated number.

These statistics are computed with the commands

```r
t_degree
t_recip
t_trans
```
Let’s use these statistics to evaluate the fit of three models:

**fit_bin_000**: the probit model, fix $\sigma_a^2 = \sigma_b^2 = \rho = 0$.

**fit_bin_001**: dyadic correlation $\rho$ estimated, $\sigma_a^2 = \sigma_b^2 = 0$.

**fit_bin_111**: full SRM covariance model.
Let’s use these statistics to evaluate the fit of three models:

**fit_bin_000**: the probit model, fix $\sigma_a^2 = \sigma_b^2 = \rho = 0$.

**fit_bin_001**: dyadic correlation $\rho$ estimated, $\sigma_a^2 = \sigma_b^2 = 0$.

**fit_bin_111**: full SRM covariance model.
Let’s use these statistics to evaluate the fit of three models:

**fit_bin_000:** the probit model, fix $\sigma^2_a = \sigma^2_b = \rho = 0$.

**fit_bin_001:** dyadic correlation $\rho$ estimated, $\sigma^2_a = \sigma^2_b = 0$.

**fit_bin_111:** full SRM covariance model.
Let’s use these statistics to evaluate the fit of three models:

- **fit_bin_000**: the probit model, fix $\sigma_a^2 = \sigma_b^2 = \rho = 0$.
- **fit_bin_001**: dyadic correlation $\rho$ estimated, $\sigma_a^2 = \sigma_b^2 = 0$.
- **fit_bin_111**: full SRM covariance model.
Let’s use these statistics to evaluate the fit of three models:

**fit_bin_000**: the probit model, fix $\sigma_a^2 = \sigma_b^2 = \rho = 0$.

**fit_bin_001**: dyadic correlation $\rho$ estimated, $\sigma_a^2 = \sigma_b^2 = 0$.

**fit_bin_111**: full SRM covariance model.
Model comparison

```r
fit_bin_000 <- ame(YB, XD, model = "bin", rvar = FALSE, cvar = FALSE, dcor = FALSE)
```
Model comparison

```r
fit_bin_001 <- ame(YB, XD, model = "bin", rvar = FALSE, cvar = FALSE, dcor = TRUE)
```
fit_bin_111<-ame(YB,XD,model="bin")

Model comparison
The SRM model `fit_bin_111` looks best based on these statistics, except possibly for the transitivity statistic (more on that soon).
Ordered probit models for ordinal data

The original sheep data consists of counts:

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
<th>V6</th>
<th>V7</th>
<th>V8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NA</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>NA</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
<td>7</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>NA</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>NA</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>NA</td>
</tr>
</tbody>
</table>

Ideally, this information should be taken into consideration.

- In principle, throwing away information is not efficient.
- Effects of some covariates might distinguish high values of the relation.
Ordered probit models for ordinal data

The original sheep data consists of counts:

```
YV[1:8,1:8]
```

```
##
## V1 V2 V3 V4 V5 V6 V7 V8
## [1,] NA 0 0 0 0 0 0 1
## [2,] 0 NA 0 0 5 2 1 0
## [3,] 0 0 NA 0 7 4 0 0
## [4,] 0 0 8 NA 0 0 0 0
## [5,] 0 0 0 0 NA 1 0 0
## [6,] 0 0 0 7 0 NA 0 0
## [7,] 0 0 1 0 0 0 NA 0
## [8,] 0 1 1 4 5 3 0 NA
```

Ideally, this information should be taken into consideration.

- In principle, throwing away information is not efficient.
- Effects of some covariates might distinguish high values of the relation.
Ordered probit models for ordinal data

The original sheep data consists of counts:

```r
YV[1:8,1:8]
##
## V1 V2 V3 V4 V5 V6 V7 V8
## [1,] NA 0 0 0 0 0 0 1
## [2,] 0 NA 0 0 5 2 1 0
## [3,] 0 0 NA 0 7 4 0 0
## [4,] 0 0 8 NA 0 0 0 0
## [5,] 0 0 0 0 NA 1 0 0
## [6,] 0 0 0 7 0 NA 0 0
## [7,] 0 0 1 0 0 0 NA 0
## [8,] 0 1 1 4 5 3 0 NA
```

Ideally, this information should be taken into consideration.

- In principle, throwing away information is not efficient.
- Effects of some covariates might distinguish high values of the relation.
Ordered probit models for ordinal data

The original sheep data consists of counts:

```
YV[1:8,1:8]
```

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
<th>V6</th>
<th>V7</th>
<th>V8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NA</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>NA</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
<td>7</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>NA</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>NA</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>NA</td>
</tr>
</tbody>
</table>

Ideally, this information should be taken into consideration.

- In principle, throwing away information is not efficient.
- Effects of some covariates might distinguish high values of the relation.
Ordered probit models for ordinal data

The original sheep data consists of counts:

```r
YV[1:8,1:8]
```

```
## V1 V2 V3 V4 V5 V6 V7 V8
## [1,] NA 0 0 0 0 0 0 1
## [2,] 0 NA 0 0 5 2 1 0
## [3,] 0 0 NA 0 7 4 0 0
## [4,] 0 0 8 NA 0 0 0 0
## [5,] 0 0 0 0 NA 1 0 0
## [6,] 0 0 0 7 0 NA 0 0
## [7,] 0 0 1 0 0 0 NA 0
## [8,] 0 1 1 4 5 3 0 NA
```

Ideally, this information should be taken into consideration.

- In principle, throwing away information is not efficient.
- Effects of some covariates might distinguish high values of the relation.
Ordered probit models for ordinal data

Latent variable model:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ \epsilon_{i,j} = a_i + b_j + e_{i,j} \]

Binary probit:

\[ Y_{i,j} = \begin{cases} 
0 & \text{if } Z_{i,j} < 0 \\
1 & \text{if } Z_{i,j} > 0 
\end{cases} \]

What if \( Y_{i,j} \in \{y_1, \ldots, y_K\} = \mathcal{Y} \)?

Here, \( \mathcal{Y} \) is an ordered, countable set of possible values of \( Y_{i,j} \).
Ordered probit models for ordinal data

Latent variable model:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ \epsilon_{i,j} = a_i + b_j + e_{i,j} \]

Binary probit:

\[ Y_{i,j} = \begin{cases} 
0 & \text{if } Z_{i,j} < 0 \\
1 & \text{if } Z_{i,j} > 0 
\end{cases} \]

What if \( Y_{i,j} \in \{y_1, \ldots, y_K\} = \mathcal{Y} \)?

Here, \( \mathcal{Y} \) is an ordered, countable set of possible values of \( Y_{i,j} \).
Ordered probit models for ordinal data

Latent variable model:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ \epsilon_{i,j} = a_i + b_j + e_{i,j} \]

Binary probit:

\[ Y_{i,j} = \begin{cases} 0 & \text{if } Z_{i,j} < 0 \\ 1 & \text{if } Z_{i,j} > 0 \end{cases} \]

What if \( Y_{i,j} \in \{y_1, \ldots, y_K\} = \mathcal{Y} \)?

Here, \( \mathcal{Y} \) is an ordered, countable set of possible values of \( Y_{i,j} \).
Ordered probit models for ordinal data

Latent variable model:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ \epsilon_{i,j} = a_i + b_j + e_{i,j} \]

Binary probit:

\[ Y_{i,j} = \begin{cases} 
0 & \text{if } Z_{i,j} < 0 \\
1 & \text{if } Z_{i,j} > 0 
\end{cases} \]

What if \( Y_{i,j} \in \{ y_1, \ldots, y_K \} = \mathcal{Y} \)?

Here, \( \mathcal{Y} \) is an ordered, countable set of possible values of \( Y_{i,j} \).
Ordered probit models for ordinal data

Latent variable model:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ \epsilon_{i,j} = a_i + b_j + e_{i,j} \]

Ordered probit:

\[ Y_{i,j} = \begin{cases} 
  y_1 & \text{if } Z_{i,j} \in (-\infty, c_1) \\
  y_2 & \text{if } Z_{i,j} \in (c_1, c_2) \\
  \vdots \\
  y_{K-1} & \text{if } Z_{i,j} \in (c_{K-2}, c_{K-1}) \\
  y_K & \text{if } Z_{i,j} \in (c_{K-1}, \infty) 
\end{cases} \]
Ordered probit models for ordinal data

Latent variable model:

\[ Z_{i,j} = \beta^T x_{i,j} + \epsilon_{i,j} \]
\[ \epsilon_{i,j} = a_i + b_j + e_{i,j} \]

Ordered probit:

\[ Y_{i,j} = \begin{cases} 
  y_1 & \text{if } Z_{i,j} \in (\infty, c_1) \\
  y_2 & \text{if } Z_{i,j} \in (c_1, c_2) \\
  \vdots \\
  y_{K-1} & \text{if } Z_{i,j} \in (c_{K-2}, c_{K-1}) \\
  y_K & \text{if } Z_{i,j} \in (c_{K-1}, \infty) 
\end{cases} \]
Ordered probit models for ordinal data

table(c(YB))

##
## 0 1
## 506 250

table(c(YV))

##
## 0 1 2 3 4 5 6 7 8 9 10 11 12
## 506 100 59 34 20 13 5 7 5 1 3 1 2

round( table(c(YV))/sum(table(c(YV))) ,3 )

##
## 0 1 2 3 4 5 6 7 8 9 10 11
## 0.669 0.132 0.078 0.045 0.026 0.017 0.007 0.009 0.007 0.001 0.004 0.001
## 12
## 0.003
Link function for ordered probit
SRM probit in amen

```r
fit_ame_ord <- ame(YV, XD, model = "ord")
```
Summary:

```r
summary(fit_ame_ord)
```

```r
##
## beta:
## pmean psd z-stat p-val
## .dyad 0.162 0.028 5.72 0
##
## Sigma_ab pmean:
## a b
## a 0.731 0.125
## b 0.125 0.224
##
## rho pmean:
## -0.388
```
Confidence intervals

```r
apply( fit_ame_ord$BETA , 2 , quantile, prob=c(.025,.5, .975))
```

## .dyad
## 2.5% 0.1138031
## 50% 0.1615003
## 97.5% 0.2231967

```r
apply( fit_ame_ord$SABR , 2 , quantile, prob=c(.025,.5, .975))
```

## va cab vb rho ve
## 2.5% 0.3728079 -0.06367146 0.1060518 -0.5601718 1
## 50% 0.6811580 0.12026448 0.2057822 -0.3863772 1
## 97.5% 1.4126096 0.37453941 0.4239469 -0.2343322 1

These results are very similar to those obtained from the binary probit analysis using the dichotomized data.
The role of covariance

Regression modeling:

\[ Y_{i,j} = \beta^T x_{i,j} + \epsilon_{ij} \]
\[ Y = \langle X, \beta \rangle + E \]

OLS estimation:

\[ \hat{\beta} = (X^T X)^{-1} X^T y \]

Precision of OLS estimators:

Let \( C = \text{Cov}[E] \)

\[ \text{Cov}[\hat{\beta}] = (X^T X)^{-1} X^T C X (X^T X)^{-1} \]
\[ = (X^T X)^{-1} \sigma^2 \text{ if } C = \sigma^2 I \]

For networks and relational data, typically \( C \neq \sigma^2 I \).
Accurate standard errors can’t be obtained unless we know/estimate \( \text{Cov}[E] \).
The role of covariance

Regression modeling:

\[ Y_{i,j} = \beta^T x_{i,j} + \epsilon_{ij} \]
\[ Y = \langle X, \beta \rangle + E \]

OLS estimation:

\[ \hat{\beta} = (X^T X)^{-1} X^T y \]

Precision of OLS estimators:
Let \( C = \text{Cov}[E] \)

\[ \text{Cov}[\hat{\beta}] = (X^T X)^{-1} X^T C X (X^T X)^{-1} \]
\[ = (X^T X)^{-1} \sigma^2 \text{ if } C = \sigma^2 I \]

For networks and relational data, typically \( C \neq \sigma^2 I \).
Accurate standard errors can’t be obtained unless we know/estimate \( \text{Cov}[E] \).
The role of covariance

Regression modeling:

\[ Y_{i,j} = \beta^T x_{i,j} + \epsilon_{ij} \]
\[ Y = \langle X, \beta \rangle + E \]

OLS estimation:

\[ \hat{\beta} = (X^T X)^{-1} X^T y \]

Precision of OLS estimators:

Let \( C = \text{Cov}[E] \)

\[ \text{Cov}[\hat{\beta}] = (X^T X)^{-1} X^T CX (X^T X)^{-1} \]
\[ = (X^T X)^{-1} \sigma^2 \text{ if } C = \sigma^2 I \]

For networks and relational data, typically \( C \neq \sigma^2 I \).

Accurate standard errors can’t be obtained unless we know/estimate Cov[E].
The role of covariance

Regression modeling:

\[ Y_{i,j} = \beta^T x_{i,j} + \epsilon_{ij} \]
\[ Y = \langle X, \beta \rangle + E \]

OLS estimation:

\[ \hat{\beta} = (X^T X)^{-1} X^T y \]

Precision of OLS estimators:
Let \( C = \text{Cov}[E] \)

\[ \text{Cov}[\hat{\beta}] = (X^T X)^{-1} X^T C X (X^T X)^{-1} \]
\[ = (X^T X)^{-1} \sigma^2 \text{ if } C = \sigma^2 I \]

For networks and relational data, typically \( C \neq \sigma^2 I \).
Accurate standard errors can’t be obtained unless we know/estimate \( \text{Cov}[E] \).
Justifying the SRM

The social relations covariance model can by a symmetry principle.

**Exchangeability:**

1. randomly sample $n$ individuals from a population;
2. observe $\epsilon_{ij} =$ directed relation between $i$th and $j$th person.

Consider a probability model $\Pr(E)$ for the possible outcomes of $E = \{\epsilon_{i,j}\}$

$$
\Pr \left( \begin{pmatrix} NA & -0.94 & 0.15 \\ -0.7 & NA & 0.63 \\ 0.63 & -0.42 & NA \end{pmatrix} \right) = ?
$$

$$
\Pr \left( \begin{pmatrix} NA & 0.63 & -0.42 \\ 0.15 & NA & -0.94 \\ 0.63 & -0.7 & NA \end{pmatrix} \right) = ?
$$

Note the second matrix is the same as the first with (1,2,3) relabeled as (2,3,1).
The social relations covariance model can by a symmetry principle.

**Exchangeability:**

1. randomly sample $n$ individuals from a population;
2. observe $\epsilon_{i,j} =$ directed relation between $i$th and $j$th person.

Consider a probability model $\Pr(E)$ for the possible outcomes of $E = \{\epsilon_{i,j}\}$

$$\begin{bmatrix}
NA & -0.94 & 0.15 \\
-0.7 & NA & 0.63 \\
0.63 & -0.42 & NA \\
\end{bmatrix}$$

$\Pr(\cdot) =$ ?

$$\begin{bmatrix}
NA & 0.63 & -0.42 \\
0.15 & NA & -0.94 \\
0.63 & -0.7 & NA \\
\end{bmatrix}$$

$\Pr(\cdot) =$ ?

Note the second matrix is the same as the first with (1,2,3) relabeled as (2,3,1).
Justifying the SRM

The social relations covariance model can by a symmetry principle.

**Exchangeability:**

1. randomly sample \( n \) individuals from a population;
2. observe \( \epsilon_{i,j} = \) directed relation between \( i \)th and \( j \)th person.

Consider a probability model \( \Pr(E) \) for the possible outcomes of \( E = \{\epsilon_{i,j}\} \)

\[
\begin{pmatrix}
NA & -0.94 & 0.15 \\
-0.7 & NA & 0.63 \\
0.63 & -0.42 & NA
\end{pmatrix}
\]

\[\Pr(\cdot) = ?\]

\[
\begin{pmatrix}
NA & 0.63 & -0.42 \\
0.15 & NA & -0.94 \\
0.63 & -0.7 & NA
\end{pmatrix}
\]

\[\Pr(\cdot) = ?\]

Note the second matrix is the same as the first with \( (1,2,3) \) relabeled as \( (2,3,1) \).
Justifying the SRM

The social relations covariance model can by a symmetry principle.

**Exchangeability:**

1. randomly sample \( n \) individuals from a population;
2. observe \( \epsilon_{i,j} \) = directed relation between \( i \)th and \( j \)th person.

Consider a probability model \( \Pr(E) \) for the possible outcomes of \( E = \{\epsilon_{i,j}\} \)

\[
\begin{pmatrix}
  \text{NA} & -0.94 & 0.15 \\
  -0.7 & \text{NA} & 0.63 \\
  0.63 & -0.42 & \text{NA}
\end{pmatrix}
\]

\[
\begin{pmatrix}
  \text{NA} & 0.63 & -0.42 \\
  0.15 & \text{NA} & -0.94 \\
  0.63 & -0.7 & \text{NA}
\end{pmatrix}
\]

Note the second matrix is the same as the first with \((1,2,3)\) relabeled as \((2,3,1)\).
Justifying the SRM

The social relations covariance model can by a symmetry principle.

**Exchangeability:**

1. randomly sample \( n \) individuals from a population;
2. observe \( \epsilon_{i,j} = \) directed relation between \( i \)th and \( j \)th person.

Consider a probability model \( \Pr(E) \) for the possible outcomes of \( E = \{\epsilon_{i,j}\} \)

\[
\begin{pmatrix}
NA & -0.94 & 0.15 \\
-0.7 & NA & 0.63 \\
0.63 & -0.42 & NA
\end{pmatrix}
\]

\[= ?\]

\[
\begin{pmatrix}
NA & 0.63 & -0.42 \\
0.15 & NA & -0.94 \\
0.63 & -0.7 & NA
\end{pmatrix}
\]

Note the second matrix is the same as the first with (1,2,3) relabeled as (2,3,1).
The social relations covariance model can by a symmetry principle.

**Exchangeability:**

1. randomly sample $n$ individuals from a population;  
2. observe $\epsilon_{i,j} =$ directed relation between $i$th and $j$th person.

Consider a probability model $Pr(E)$ for the possible outcomes of $E = \{\epsilon_{i,j}\}$

$$Pr \left( \begin{pmatrix} NA & -0.94 & 0.15 \\ -0.7 & NA & 0.63 \\ 0.63 & -0.42 & NA \end{pmatrix} \right) = ?$$

$$Pr \left( \begin{pmatrix} NA & 0.63 & -0.42 \\ 0.15 & NA & -0.94 \\ 0.63 & -0.7 & NA \end{pmatrix} \right) = ?$$

Note the second matrix is the same as the first with (1,2,3) relabeled as (2,3,1).
The social relations covariance model can by a symmetry principle.

**Exchangeability:**

1. randomly sample $n$ individuals from a population;
2. observe $\epsilon_{i,j} =$ directed relation between $i$th and $j$th person.

Consider a probability model $\Pr(E)$ for the possible outcomes of $E = \{\epsilon_{i,j}\}$

$$
\Pr \left( \begin{pmatrix}
NA & -0.94 & 0.15 \\
-0.7 & NA & 0.63 \\
0.63 & -0.42 & NA \\
\end{pmatrix} \right) = ?
$$

$$
\Pr \left( \begin{pmatrix}
NA & 0.63 & -0.42 \\
0.15 & NA & -0.94 \\
0.63 & -0.7 & NA \\
\end{pmatrix} \right) = ?
$$

Note the second matrix is the same as the first with (1,2,3) relabeled as (2,3,1).
The social relations covariance model can be justified by a symmetry principle.

**Exchangeability:**

1. randomly sample $n$ individuals from a population;
2. observe $\epsilon_{i,j} =$ directed relation between $i$th and $j$th person.

Consider a probability model $\Pr(E)$ for the possible outcomes of $E = \{\epsilon_{i,j}\}$

$$\Pr \left( \begin{pmatrix} NA & -0.94 & 0.15 \\ -0.7 & NA & 0.63 \\ 0.63 & -0.42 & NA \end{pmatrix} \right) = ?$$

$$\Pr \left( \begin{pmatrix} NA & 0.63 & -0.42 \\ 0.15 & NA & -0.94 \\ 0.63 & -0.7 & NA \end{pmatrix} \right) = ?$$

Note the second matrix is the same as the first with (1,2,3) relabeled as (2,3,1).
Justifying the SRM

Let $\pi$ be some permutation of $\{1, \ldots, n\}$.

$$E = \{\epsilon_{i,j} : i \neq j\}$$
$$E_\pi = \{\epsilon_{\pi_i,\pi_j} : i \neq j\}$$

**Exchangeability:** A probability distribution $\Pr(E)$ is exchangeable if

$$\Pr(E) = \Pr(E_\pi)$$

for all $E$ and permutations $\pi$.

Exchangeability can be justified by

- random sampling of nodes from a population;
- symmetry in the uncertainty in the $\epsilon_{i,j}$'s.
Let $\pi$ be some permutation of $\{1, \ldots, n\}$.

$E = \{\epsilon_{i,j} : i \neq j\}$

$E_\pi = \{\epsilon_{\pi_i,\pi_j} : i \neq j\}$

**Exchangeability:** A probability distribution $\Pr(E)$ is exchangeable if

$$\Pr(E) = \Pr(E_\pi)$$

for all $E$ and permutations $\pi$.

Exchangeability can be justified by

- random sampling of nodes from a population;
- symmetry in the uncertainty in the $\epsilon_{i,j}$'s.
Let $\pi$ be some permutation of $\{1, \ldots, n\}$.

$$E = \{\epsilon_{i,j} : i \neq j\}$$
$$E_\pi = \{\epsilon_{\pi_i,\pi_j} : i \neq j\}$$

**Exchangeability:** A probability distribution $\Pr(E)$ is exchangeable if

$$\Pr(E) = \Pr(E_\pi)$$

for all $E$ and permutations $\pi$.

Exchangeability can be justified by

- random sampling of nodes from a population;
- symmetry in the uncertainty in the $\epsilon_{i,j}$'s.
Let $\pi$ be some permutation of \{1, $\ldots$, $n$\}.

$$E = \{\epsilon_{i,j} : i \neq j\}$$

$$E_\pi = \{\epsilon_{\pi i, \pi j} : i \neq j\}$$

**Exchangeability:** A probability distribution $Pr(E)$ is exchangeable if

$$Pr(E) = Pr(E_\pi)$$

for all $E$ and permutations $\pi$.

Exchangeability can be justified by

- random sampling of nodes from a population;
- symmetry in the uncertainty in the $\epsilon_{i,j}$'s.
Justifying the SRM

Let $\pi$ be some permutation of $\{1, \ldots, n\}$.

$$
\mathbf{E} = \{\epsilon_{i,j} : i \neq j\} \\
\mathbf{E}_\pi = \{\epsilon_{\pi_i,\pi_j} : i \neq j\}
$$

**Exchangeability:** A probability distribution $\Pr(\mathbf{E})$ is exchangeable if

$$
\Pr(\mathbf{E}) = \Pr(\mathbf{E}_\pi)
$$

for all $\mathbf{E}$ and permutations $\pi$.

Exchangeability can be justified by

- random sampling of nodes from a population;
- symmetry in the uncertainty in the $\epsilon_{i,j}$'s.
Justifying the SRM

Let $\pi$ be some permutation of $\{1, \ldots, n\}$.

$$E = \{\epsilon_{i,j} : i \neq j\}$$

$$E_\pi = \{\epsilon_{\pi_i,\pi_j} : i \neq j\}$$

**Exchangeability:** A probability distribution $\Pr(E)$ is exchangeable if

$$\Pr(E) = \Pr(E_\pi)$$

for all $E$ and permutations $\pi$.

Exchangeability can be justified by

- random sampling of nodes from a population;
- symmetry in the uncertainty in the $\epsilon_{i,j}$'s.
Justifying the SRM

Let $\pi$ be some permutation of $\{1, \ldots, n\}$.

$$E = \{\epsilon_{i,j} : i \neq j\}$$
$$E_\pi = \{\epsilon_{\pi_i,\pi_j} : i \neq j\}$$

**Exchangeability:** A probability distribution $\Pr(E)$ is exchangeable if

$$\Pr(E) = \Pr(E_\pi)$$

for all $E$ and permutations $\pi$.

Exchangeability can be justified by

- random sampling of nodes from a population;
- symmetry in the uncertainty in the $\epsilon_{i,j}$’s.
Interesting result:

Suppose

1. $\epsilon_{i,j}$'s are normal and mean zero;
2. our probability for $E = \{\epsilon_{i,j}\}$ is exchangeable.

Then

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

$\{ (a_1, b_1), \ldots, (a_n, b_n) \} \sim \text{i.i.d. } N(0, \Sigma_{ab})$

$\{ (e_{i,j}, e_{j,i}) : i \neq j \} \sim \text{i.i.d. } N(0, \Sigma_e)$

for some $\Sigma_{ab}$ and $\Sigma_e$ (Li and Loken, 2002).

Interpretation:

normality + exchangeability $\Rightarrow$ SRM
Interesting result:
Suppose
1. \( \epsilon_{i,j} \)’s are normal and mean zero;
2. our probability for \( \mathbf{E} = \{\epsilon_{i,j}\} \) is exchangeable.

Then

\[
\epsilon_{i,j} = a_i + b_j + e_{i,j}
\]
\[
\{(a_1, b_1), \ldots, (a_n, b_n)\} \sim \text{i.i.d. } \mathcal{N}(0, \Sigma_{ab})
\]
\[
\{(e_{i,j}, e_{j,i}) : i \neq j\} \sim \text{i.i.d. } \mathcal{N}(0, \Sigma_e)
\]

for some \( \Sigma_{ab} \) and \( \Sigma_e \) (Li and Loken, 2002).

Interpretation:

normality + exchangeability \( \Rightarrow \) SRM
Interesting result:

Suppose

1. $\epsilon_{i,j}$'s are normal and mean zero;
2. our probability for $E = \{\epsilon_{i,j}\}$ is exchangeable.

Then

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

$$\{(a_1, b_1), \ldots, (a_n, b_n)\} \sim \text{i.i.d. } N(0, \Sigma_{ab})$$

$$\{(e_{i,j}, e_{j,i}) : i \neq j\} \sim \text{i.i.d. } N(0, \Sigma_e)$$

for some $\Sigma_{ab}$ and $\Sigma_e$ (Li and Loken, 2002).

Interpretation:

normality + exchangeability $\Rightarrow$ SRM
Interesting result:
Suppose

1. $\epsilon_{i,j}$'s are normal and mean zero;
2. our probability for $E = \{\epsilon_{i,j}\}$ is exchangeable.

Then

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

$$
\{(a_1, b_1), \ldots, (a_n, b_n)\} \sim \text{i.i.d. } N(0, \Sigma_{ab})
$$

$$
\{(e_{i,j}, e_{j,i}) : i \neq j\} \sim \text{i.i.d. } N(0, \Sigma_e)
$$

for some $\Sigma_{ab}$ and $\Sigma_e$ (Li and Loken, 2002).

Interpretation:

normality $+$ exchangeability $\Rightarrow$ SRM
Interesting result:

Suppose

1. $\epsilon_{i,j}$'s are normal and mean zero;
2. our probability for $\mathbf{E} = \{\epsilon_{i,j}\}$ is exchangeable.

Then

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

$\{(a_1, b_1), \ldots, (a_n, b_n)\} \sim \text{i.i.d. } \mathcal{N}(0, \Sigma_{ab})$

$\{(e_{i,j}, e_{j,i}) : i \neq j\} \sim \text{i.i.d. } \mathcal{N}(0, \Sigma_e)$

for some $\Sigma_{ab}$ and $\Sigma_e$ (Li and Loken, 2002).

Interpretation:

normality + exchangeability $\Rightarrow$ SRM
There is a stronger result than this: If $E$ is exchangeable, then

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

$$\text{Cov}[(a_i, b_i)] = \Sigma_{ab}$$

$$\text{Cov}[(e_{i,j}, e_{j,i})] = \Sigma_e$$

for some $\Sigma_{ab}, \Sigma_e$.

**Interpretation:**
exchangeability $\Rightarrow$ SRM covariance structure

**Implication:**

$$\text{Cov}[\hat{\beta}] = (X^TX)^{-1}X^T C X (X^TX)^{-1},$$

where $C = \text{Cov}[(\epsilon_{i,j}, \epsilon_{j,i})].$

Under exchangability, the SRM provides appropriate SEs and CIs for $\beta$. 

Justifying the SRM
Justifying the SRM

There is a stronger result than this: If $\mathbf{E}$ is exchangeable, then

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

$$
\text{Cov}[(a_i, b_i)] = \Sigma_{ab}
$$

$$
\text{Cov}[(e_{i,j}, e_{j,i})] = \Sigma_e
$$

for some $\Sigma_{ab}, \Sigma_e$.

**Interpretation:**

exchangeability $\Rightarrow$ SRM covariance structure

**Implication:**

$$
\text{Cov}[\hat{\beta}] = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{C} \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1},
$$

where $\mathbf{C} = \text{Cov}[(\epsilon_{i,j}, \epsilon_{j,i})]$.

Under exchangability, the SRM provides appropriate SEs and CIs for $\beta$. 
Justifying the SRM

There is a stronger result than this: If \( \mathbf{E} \) is exchangeable, then

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

\[
\text{Cov}[(a_i, b_i)] = \Sigma_{ab}
\]

\[
\text{Cov}[(e_{i,j}, e_{j,i})] = \Sigma_e
\]

for some \( \Sigma_{ab}, \Sigma_e \).

**Interpretation:**
exchangeability \( \Rightarrow \) SRM covariance structure

**Implication:**

\[
\text{Cov}[\hat{\beta}] = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{C} \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1},
\]

where \( \mathbf{C} = \text{Cov}[(\epsilon_{i,j}, \epsilon_{j,i})] \).

Under exchangability, the SRM provides appropriate SEs and CIs for \( \beta \).
Justifying the SRM

There is a stronger result than this: If $E$ is exchangeable, then

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

$$\text{Cov}[(a_i, b_i)] = \Sigma_{ab}$$

$$\text{Cov}[(e_{i,j}, e_{j,i})] = \Sigma_e$$

for some $\Sigma_{ab}, \Sigma_e$.

**Interpretation:**

exchangeability $\Rightarrow$ SRM covariance structure

**Implication:**

$$\text{Cov}[\hat{\beta}] = (X^T X)^{-1} X^T C X (X^T X)^{-1},$$

where $C = \text{Cov}[(\epsilon_{i,j}, \epsilon_{j,i})]$.

Under exchangability, the SRM provides appropriate SEs and CIs for $\beta$. 