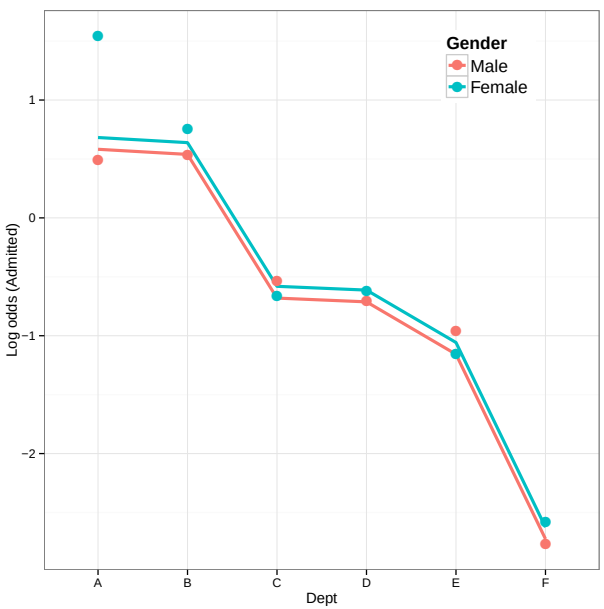
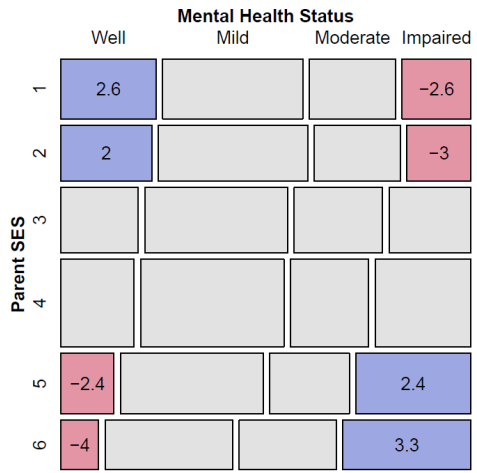
Extending Loglinear Models

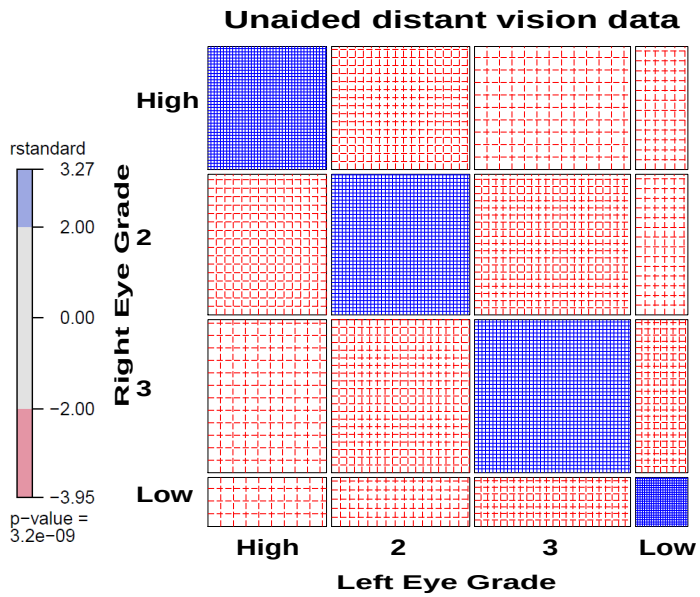
Michael Friendly

Psych 6136

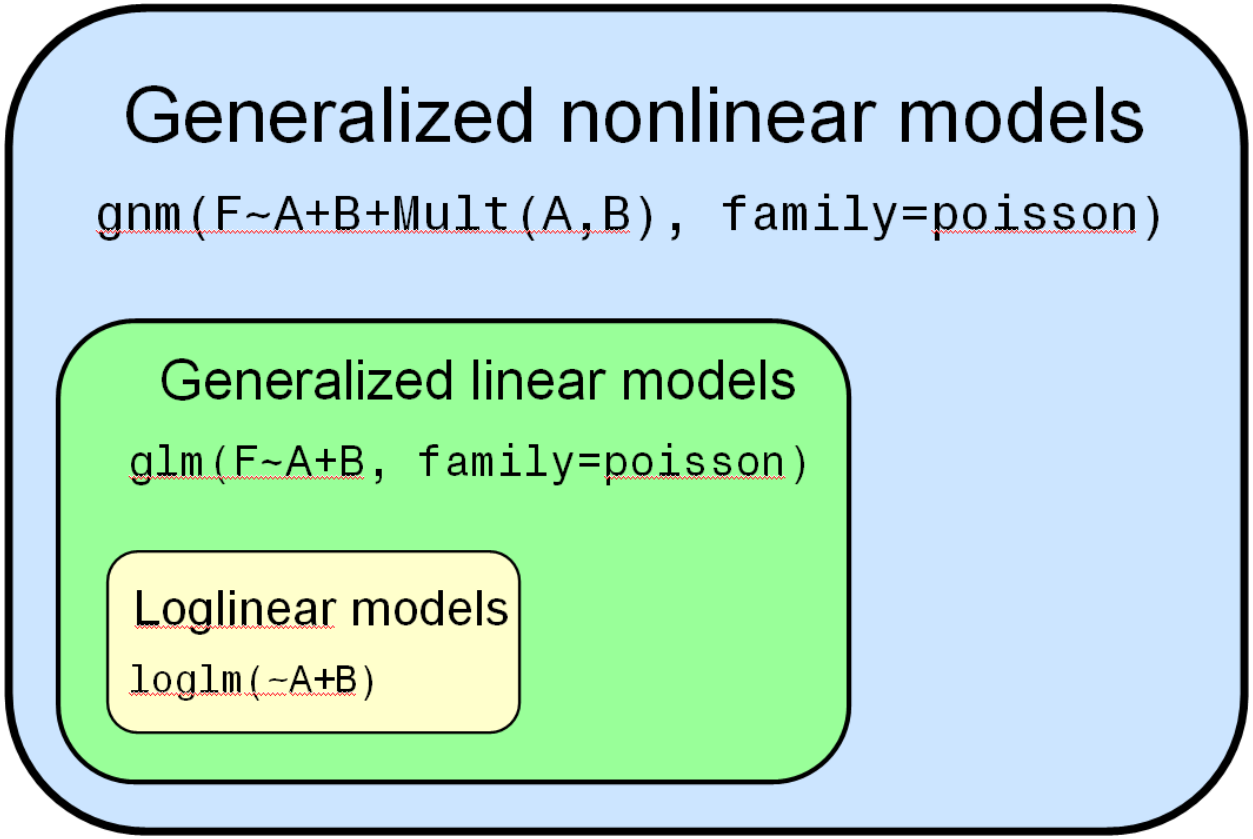
April 6, 2015







Visual overview: Models for frequency tables



- Related models: logistic regression, polytomous regression, log odds models, ...
- Goals: Connect all with visualization methods

Loglinear models: Perspectives

Loglinear approach

Loglinear models were first developed as an analog of classical ANOVA models, where *multiplicative* relations (under independence) are re-expressed in *additive* form as models for log(frequency).

$$\log m_{ij} = \mu + \lambda_i^A + \lambda_j^B \equiv [A][B] \equiv\sim A + B$$

- This expresses the model of independence for a two-way table (no A*B association, or $A \perp B$)
- The notations $[A][B] \equiv\sim A + B$ are shorthands
- Three-way tables: models $[A][B][C]$ (mutual indep.), $[AB][C]$ (joint indep.), $[AB][AC]$ (cond. indep.), ... $[ABC]$ (saturated)

Extended loglinear models

Loglinear models can be extended in a variety of ways:

- Models for ordinal factors allow a more parsimonious description of association
- Specialized models for square tables provide more nuanced hypotheses
- These ideas apply to higher-way tables
- Some of these extensions are more easily understood or used when loglinear models are re-cast in an equivalent, but simpler or more general form

Loglinear models: Perspectives

GLM approach

More generally, loglinear models are also generalized linear models (GLMs) for log(frequency), with a Poisson distribution for the cell counts.

$$\log \boldsymbol{m} = \boldsymbol{X}\beta$$

- This looks just like the general linear ANOVA, regression model, but for log frequency
- This approach allows quantitative predictors and special ways of treating ordinal factors

Loglinear models: Perspectives

Logit models

When one table variable is a binary response, a logit model for that response is equivalent to a loglinear model.

$$\log(m_{1jk}/m_{2jk}) = \alpha + \beta_j^B + \beta_k^C \equiv [AB][AC][BC]$$

- $\log(m_{1jk}/m_{2jk})$ represents the log odds of response category 1 vs. 2
- The model formula includes only terms for the effects on A of variables B and C
- The equivalent loglinear model is [AB] [AC] [BC]
- The logit model assumes [BC] association, and [AB] $\rightarrow \beta_j^B$, [AC] $\rightarrow \beta_k^C$

Logit models

For a *binary* response, each loglinear model is equivalent to a logit model (logistic regression, with categorical predictors)

- e.g., Admit $\perp$ Gender $|$ Dept (conditional independence $\equiv$ [AD][DG])

$$\log m_{ijk} = \mu + \lambda_i^A + \lambda_j^D + \lambda_k^G + \lambda_{ij}^{AD} + \lambda_{jk}^{DG}$$

So, for admitted ($i = 1$) and rejected ($i = 2$), we have:

$$\log m_{1jk} = \mu + \lambda_1^A + \lambda_j^D + \lambda_k^G + \lambda_{1j}^{AD} + \lambda_{jk}^{DG} \quad (1)$$

$$\log m_{2jk} = \mu + \lambda_2^A + \lambda_j^D + \lambda_k^G + \lambda_{2j}^{AD} + \lambda_{jk}^{DG} \quad (2)$$

Thus, subtracting (1)-(2), terms not involving Admit will cancel:

$$\begin{aligned} L_{jk} &= \log m_{1jk} - \log m_{2jk} = \log(m_{1jk}/m_{2jk}) = \text{log odds of admission} \\ &= (\lambda_1^A - \lambda_2^A) + (\lambda_{1j}^{AD} - \lambda_{2j}^{AD}) \\ &= \alpha + \beta_j^{\text{Dept}} \qquad \text{(renaming terms)} \end{aligned}$$

where, α: overall log odds of admission; β_j^{Dept}: effect on admissions of department

Logit models

Other loglinear models have similar, simpler forms as logit models, where only the relations of the response to the predictors appear in the equivalent logit model.

- Admit $\perp$ Gender $\perp$ Dept (mutual independence $\equiv$ [A][D][G])

$$\begin{aligned} \log m_{ijk} &= \mu + \lambda_i^A + \lambda_j^D + \lambda_k^G \\ \equiv L_{jk} &= (\lambda_1^A - \lambda_2^A) = \alpha \qquad \text{(constant log odds)} \end{aligned}$$

- Admit $\perp$ Gender $|$ Dept, except for Dept. A

$$\begin{aligned} \log m_{ijk} &= \mu + \lambda_i^A + \lambda_j^D + \lambda_k^G + \lambda_{ij}^{AD} + \lambda_{jk}^{DG} + \delta_{(j=1)}\lambda_{ik}^{AG} \\ \equiv L_{jk} &= \log(m_{1jk}/m_{2jk}) = \alpha + \beta_j^{\text{Dept}} + \delta_{(j=1)}\beta^{\text{Gender}} \end{aligned}$$

where,

- β_j^{Dept}: effect on admissions for department j,
- $\delta_{(j=1)}\beta^{\text{Gender}}$: 1 df term for effect of gender in Dept. A.

Logit models

- Each logit model for a binary response, C is $\equiv$ a loglinear model
- The loglinear model must include the $[AB]$ association of predictors
- When the response, C has $m > 2$ levels, models for generalized logits have equivalent loglinear form.

Table: Equivalent loglinear and logit models for a three-way table, with C as a binary response variable.

Loglinear model	Logit model	Logit formula
$[AB][C]$	α	`C ~ 1`
$[AB][AC]$	$\alpha + \beta_i^A$	`C ~ A`
$[AB][BC]$	$\alpha + \beta_j^B$	`C ~ B`
$[AB][AC][BC]$	$\alpha + \beta_i^A + \beta_j^B$	`C ~ A + B`
$[ABC]$	$\alpha + \beta_i^A + \beta_j^B + \beta_{ij}^{AB}$	`C ~ A * B`

Example: Berkeley data—loglinear approach

Loglinear approach, using MASS::`loglm()`

- Uses `UCBAdmissions` in table form
- Fit model of conditional independence of gender and admission given department, [*AD*][*GD*]

```
library(MASS)
berk.loglm1 <- loglm(~ Dept * (Gender + Admit), data=UCBAdmissions)
berk.loglm1

## Call:
## loglm(formula = ~Dept * (Gender + Admit), data = UCBAdmissions)
##
## Statistics:
##                       X^2 df  P(> X^2)
## Likelihood Ratio 21.736  6 0.0013520
## Pearson          19.938  6 0.0028402
```

Example: Berkeley data—GLM approach

GLM approach, using `glm()`

- Convert `UCBAdmissions` to a frequency data frame form
- The frequency `Freq` will be used as the response variable

```
berkeley <- as.data.frame(UCBAdmissions)
head(berkeley)

##      Admit Gender Dept Freq
## 1 Admitted   Male    A  512
## 2 Rejected   Male    A  313
## 3 Admitted Female    A   89
## 4 Rejected Female    A   19
## 5 Admitted   Male    B  353
## 6 Rejected   Male    B  207
```

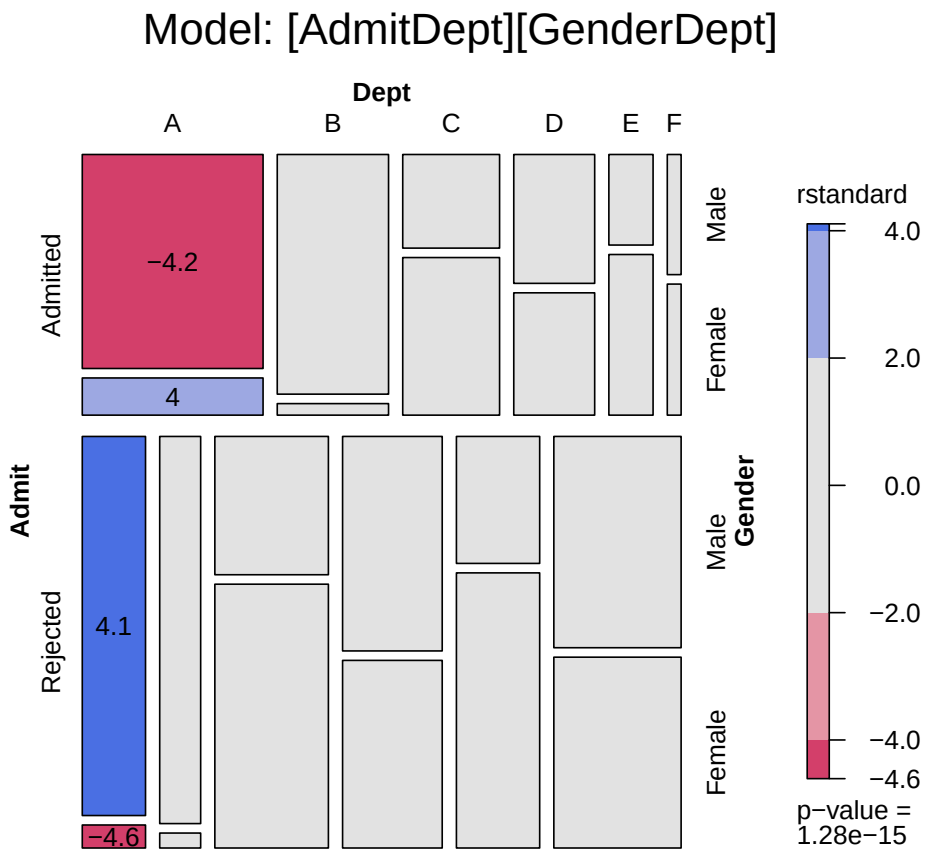
Example: Berkeley data—GLM approach

GLM approach, using `glm()`

- Fit the same model of conditional independence, [*AD*][*GD*]
- This uses `family="poisson"` to give a model for log(`Freq`)

```
berk.glm1 <- glm(Freq ~ Dept * (Gender+Admit),
                 data=berkeley, family="poisson")
library(vcdExtra)
LRstats(berk.glm1)

## Likelihood summary table:
##           AIC BIC LR Chisq Df Pr(>Chisq)
## berk.glm1 217 238     21.7  6     0.0014 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(vcdExtra)
mosaic(berk.glm1, shade=TRUE, formula=~Admit+Dept+Gender,
       residuals_type="rstandard", labeling=labeling_residuals,
       main="Model: [AdmitDept][GenderDept]")
```



Example: Berkeley data—logit approach

Logit approach, using `glm()`

- The equivalent logit model is $L_{ij} = \alpha + \beta_i^{\text{Dept}} + \beta_j^{\text{Gender}}$
- Fit this with `glm()` using `Admit=="Admitted"` as the response, and `family=binomial`
- Need to specify `weights=Freq` with the data in frequency form

```
berk.logit2 <- glm(Admit=="Admitted" ~ Dept + Gender,
                   data=berkeley, weights=Freq, family="binomial")
library(car)
Anova(berk.logit2)

## Analysis of Deviance Table (Type II tests)
##
## Response: Admit == "Admitted"
##        LR Chisq Df Pr(>Chisq)
## Dept         763  5     <2e-16 ***
## Gender         2  1       0.22
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

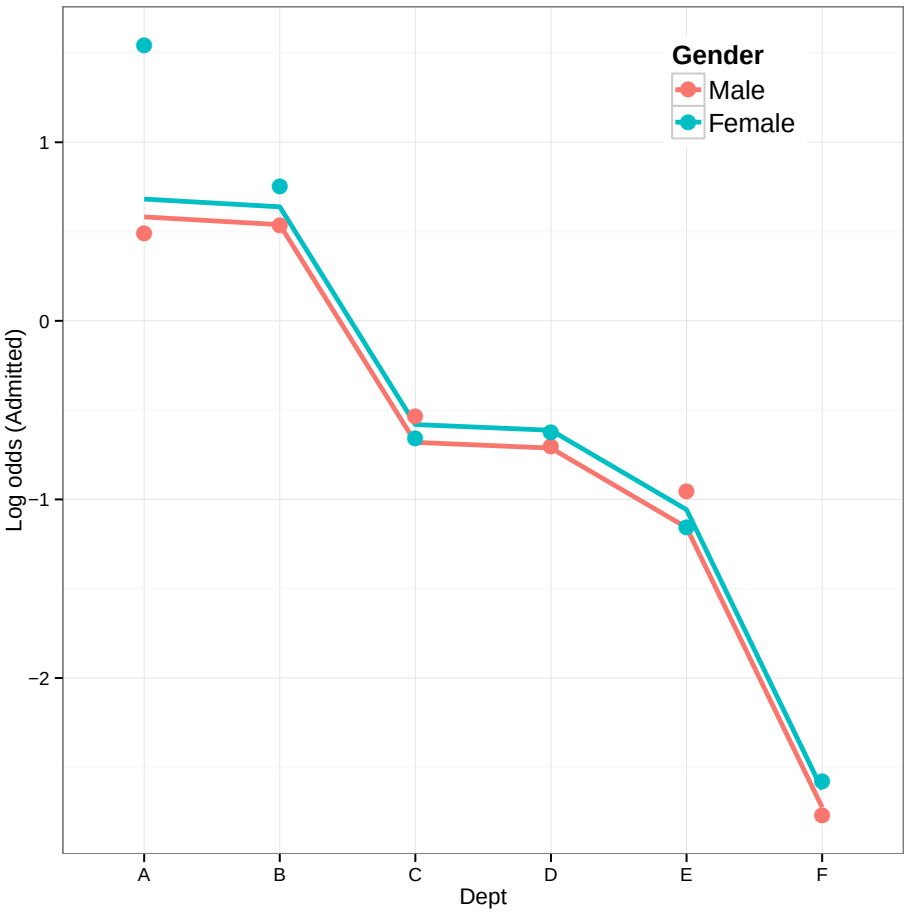
Plots for logit models

- Logit models are easier to interpret because there are fewer parameters
- Easiest to interpret from plots of the fitted log odds
- Get these using the `predict()` method for the model

```
obs <- log(UCBAdmissions[1,,] / UCBAdmissions[2,,])
pred2 <- cbind(berkeley[,1:3], fit=predict(berk.logit2))
pred2 <- cbind(subset(pred2, Admit=="Admitted"), obs=as.vector(obs))
head(pred2)
```

```
##       Admit Gender Dept    fit    obs
## 1  Admitted   Male    A  0.582  0.492
## 3  Admitted Female    A  0.682  1.544
## 5  Admitted   Male    B  0.539  0.534
## 7  Admitted Female    B  0.639  0.754
## 9  Admitted   Male    C -0.681 -0.536
## 11 Admitted Female    C -0.581 -0.660
```

Plots for logit models



- Large effects of Dept on admission
- Small effect of Gender (NS)
- Reason for lack of fit: Dept. A

A better model

Allow an association between *Admit* and *Gender* only in Dept. A

- Loglinear form:

$$\log m_{ijk} = \mu + \lambda_i^A + \lambda_j^D + \lambda_k^G + \lambda_{ij}^{AD} + \lambda_{jk}^{DG} + I(j = 1)\lambda_{ik}^{AG} \ ,$$

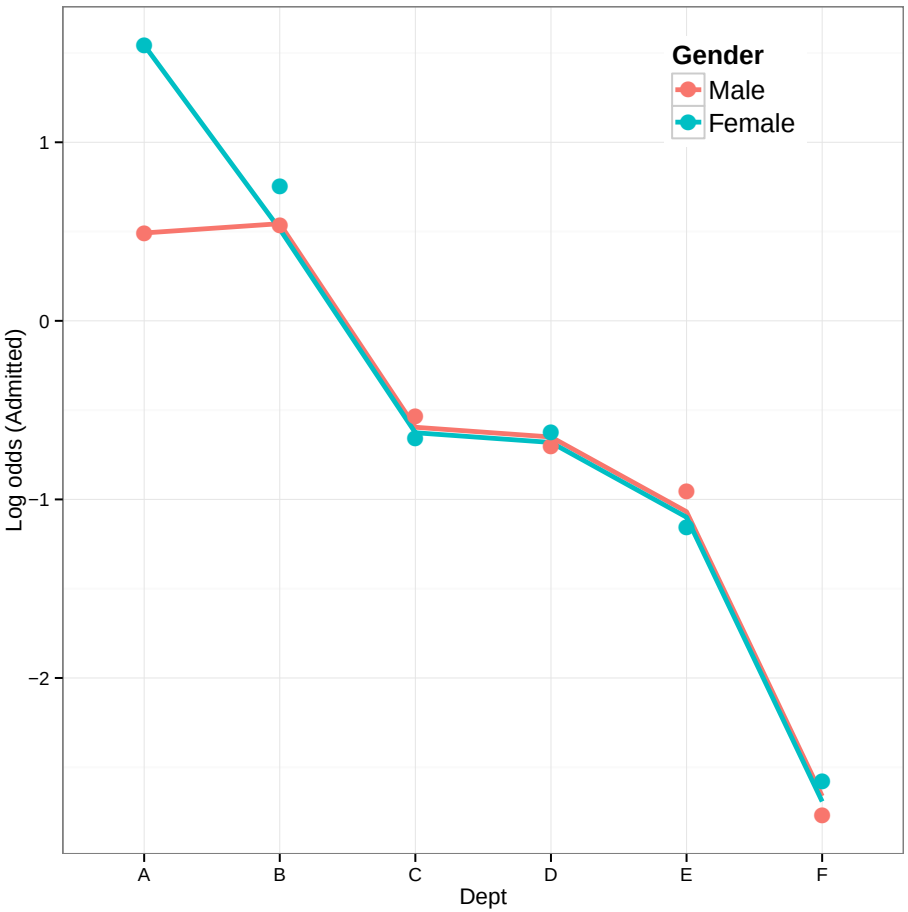
- Equivalent logit form:

$$L_{ij} = \alpha + \beta_i^{\text{Dept}} + I(j = 1)\beta^{\text{Gender}} \ .$$

```
berkeley <- within(berkeley,
                   dept1AG <- (Dept=='A')*(Gender=='Female'))
berk.logit3 <- glm(Admit=="Admitted" ~ Dept + Gender + dept1AG,
                   data=berkeley, weights=Freq, family="binomial")
Anova(berk.logit3)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Admit == "Admitted"
##          LR Chisq Df Pr(>Chisq)
## Dept          647  5    < 2e-16 ***
## Gender          0  1       0.72
## dept1AG        18  1    2.7e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Plots for logit models

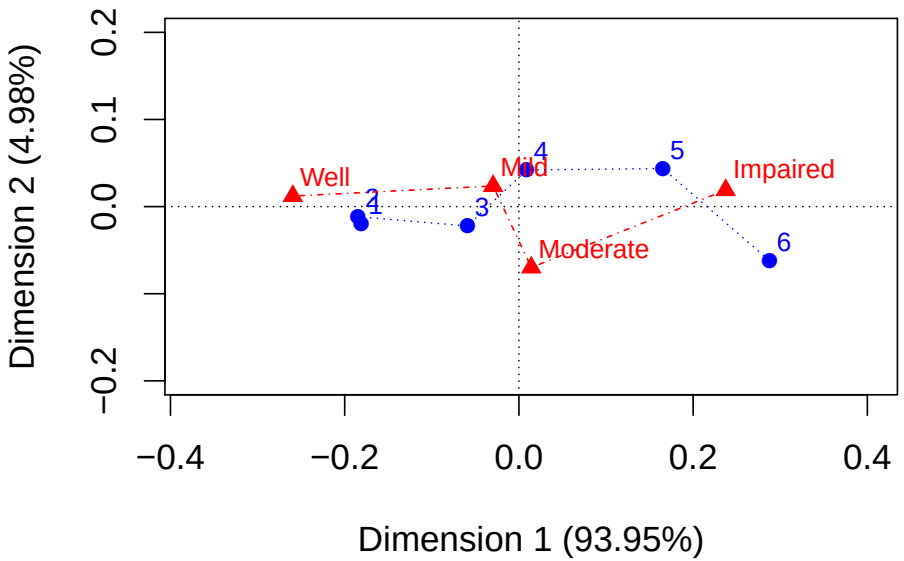


- Large effects of Dept on admission
- No effect of Gender
- Perfect fit now in Dept. A

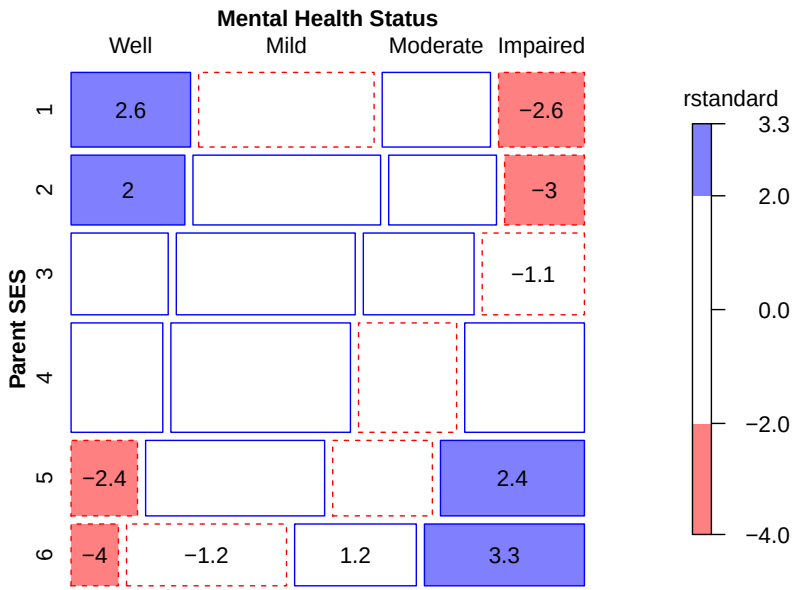
Loglinear models for ordinal variables

Ordinal variables reveal themselves in different ways in exploratory plots:

- With correspondence analysis, one large dimension accounting for most of the association
- With mosaic plots, an opposite-corner pattern of residuals



Mental health data: Independence



Advantages of ordinal models

- More focused tests $\Longrightarrow$ more powerful tests
- Consume fewer df $\Longrightarrow$ can fit unsaturated models in between [A][B] and [AB]
- Fit fewer parameters $\Longrightarrow$ easier interpretation
- Fit fewer parameters (usually) $\Longrightarrow$ smaller standard errors

These are similar to reasons for using

- Cochran-Mantel-Haenzel (CMH) tests
- Testing linear or polynomial trends/contrasts in ANOVA

Models for ordered categories

Consider an $R \times C$ table having ordered categories

- In many cases, the *RC* association may be described more simply by assigning numeric scores to the row & column categories.
- For simplicity, we consider only integer scores, 1, 2, ... here
- These models are easily extended to stratified tables

R:C model	μ_{ij}^{RC}	**df**	**Formula**
Uniform association	$i \times j \times \gamma$	1	`i:j`
Row effects	$a_i \times j$	$(I-1)$	`R:j`
Col effects	$i \times b_j$	$(J-1)$	`i:C`
Row+Col eff	$ja_i + ib_j$	$I+J-3$	`R:j + i:C`
RC(1)	$\phi_i \psi_j \times \gamma$	$I+J-3$	`Mult(R, C)`
Unstructured (R:C)	μ_{ij}^{RC}	$(I-1)(J-1)$	`R:C`

Linear x Linear Model (Uniform association)

- Assume linear ordering of both the row and column variables
- Assign scores (usually integers, 1, 2, ...)

$$\begin{aligned} \boldsymbol{a} &= \{a_i\}, \quad a_1 \leq a_2 \leq \cdots a_I \\ \boldsymbol{b} &= \{b_j\}, \quad b_1 \leq b_2 \leq \cdots b_J \end{aligned}$$

- Then, the ***linear-by-linear model*** ($L \times L$) model is:

$$\log(m_{ij}) = \mu + \lambda_i^A + \lambda_j^B + \gamma\, a_i b_j \ .$$

- The local odds ratios for adjacent 2×2 tables are:

$$\log(\theta_{ij}) = \gamma(a_{i+1} - a_i)(b_{j+1} - b_j) \quad \Longrightarrow \quad \log(\theta_{ij}) = \gamma \text{ for integer scores}$$

- Only one more parameter (γ) than the independence model
- Independence model: special case, $\gamma = 0$

Row effects and column effects models: R, C, R+C

- In the ***row effects model*** (R), the row variable, A, is treated as nominal, but B is assigned scores

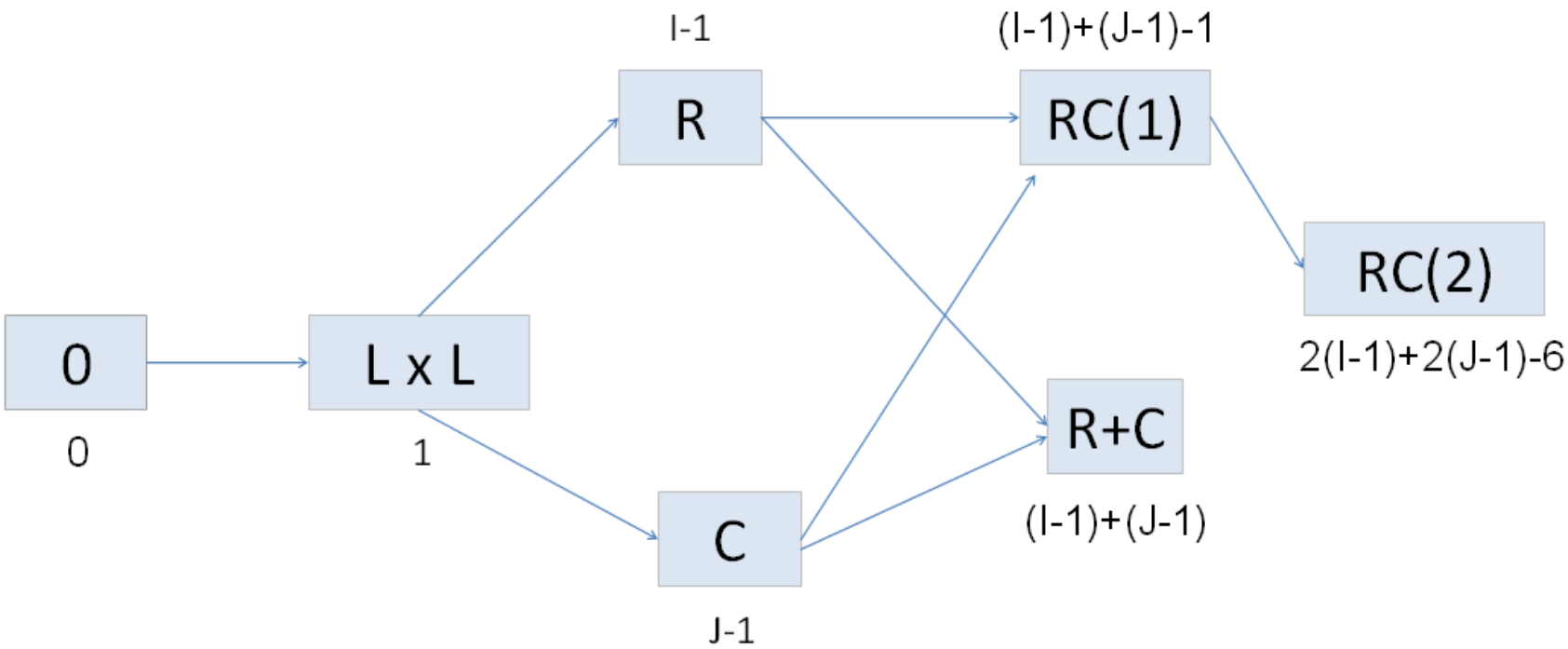
$$\log(m_{ij}) = \mu + \lambda_i^A + \lambda_j^B + \alpha_i b_j \quad \ni \quad \sum_i \alpha_i = 0 \text{ or } \alpha_1 = 0$$

- In the analogous ***column effects model*** (C), the row variable, A, is assigned scores, but B is nominal
- The ***row plus column effects model*** (R+C), assigns scores to both the rows and column variables.

$$\log(m_{ij}) = \mu + \lambda_i^A + \lambda_j^B + (\alpha_i b_j + a_i \beta_j)$$

Models for ordered categories

Nesting relationships among association models for ordinal variables



Any pair connected by an arrow path can be tested by a LR test of the form $G^2(M_2|M_1)$

Example: Mental impariment & SES

Data on mental health status of NYC youth in relation to parents' SES

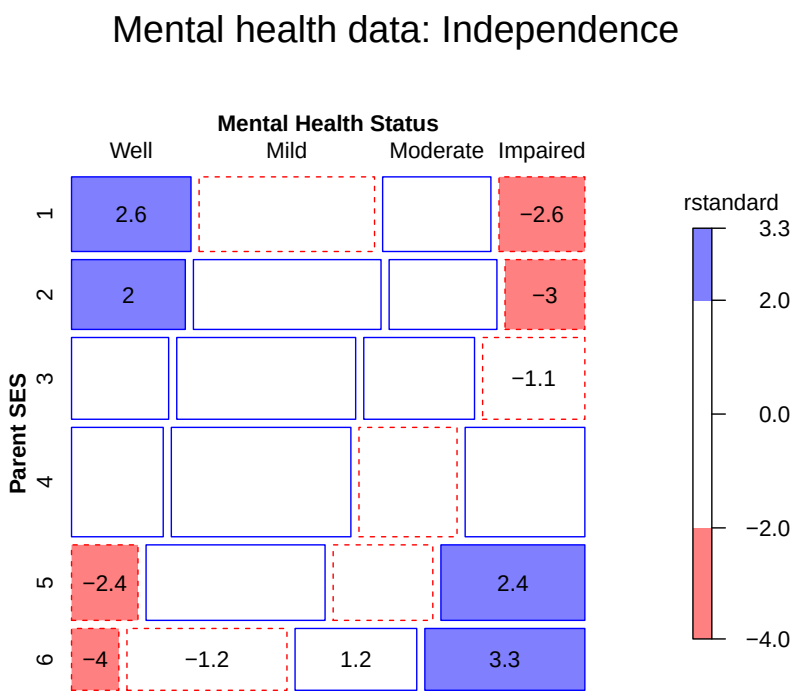
```
(mental.tab <- xtabs(Freq ~ mental+ses, data=Mental))

##             ses
## mental        1   2   3   4   5   6
##   Well       64  57  57  72  36  21
##   Mild       94  94 105 141  97  71
##   Moderate   58  54  65  77  54  54
##   Impaired   46  40  60  94  78  71
```

Test the independence model:

```
indep <- glm(Freq ~ mental + ses,
             family = poisson, data = Mental)
vcdExtra::LRstats(indep)

## Likelihood summary table:
##       AIC BIC LR Chisq Df Pr(>Chisq)
## indep 210 220     47.4 15    3.2e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Example: Mental impariment & SES

```
mosaic(indep, gp=shading_Friendly, residuals_type="rstandard",
       main="Mental health data: Independence")
```



- The mosaic shows the classic opposite-corner pattern for ordered factors
- Standardized residuals (`rstandard`) have better statistical properties
- Cells are labeled with residual values

Fitting ordinal models

To fit ordinal models, use `as.numeric()` on a factor variable to assign integer scores (or other numeric scores)

```
Cscore <- as.numeric(Mental$ses)
Rscore <- as.numeric(Mental$mental)
```

Then, add the appropriate $L \times L$, R, or C terms to the independence model:

```
linlin <- update(indep, . ~ . + Rscore:Cscore)
roweff <- update(indep, . ~ . + mental:Cscore)
coleff <- update(indep, . ~ . + Rscore:ses)
```

Comparing models

```
LRstats(indep, linlin, roweff, coleff, sortby="AIC")

## Likelihood summary table:
##          AIC   BIC LR Chisq Df Pr(>Chisq)
## indep  209.6 220.2    47.42 15   3.16e-05 ***
## coleff 179.0 195.5     6.83 10      0.741
## roweff 174.4 188.6     6.28 12      0.901
## linlin 174.1 185.8     9.90 14      0.770
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- All ordinal models are acceptable by LR tests
- The $L \times L$ model is judged the best by both AIC and BIC.
- This has only 1 more parameter than the independence model

Comparing models

When overall tests are unclear, you can carry out tests of nested sets of models using `anova()`, giving tests of ΔG^2.
For example the `indep`, `linlin` and `roweff` models are one nested set:

```
anova(indep, linlin, roweff, test="Chisq")

## Analysis of Deviance Table
##
## Model 1: Freq ~ mental + ses
## Model 2: Freq ~ mental + ses + Rscore:Cscore
## Model 3: Freq ~ mental + ses + mental:Cscore
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        15       47.4
## 2        14        9.9  1     37.5    9e-10 ***
## 3        12        6.3  2      3.6     0.16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The $L \times L$ model is a signif. improvement; the R model is not.

Interpereting the $L \times L$ model

In the $L \times L$ model, the parameter γ is the constant local odds ratio:

```
# interpret linlin association parameter
coef(linlin)[["Rscore:Cscore"]]

## [1] 0.090687

exp(coef(linlin)[["Rscore:Cscore"]])

## [1] 1.0949
```

- $\hat{\gamma} = 0.0907 \implies$ local odds ratio, $\hat{\theta}_{ij} = \exp(0.0907) = 1.095$.
- each step down the SES scale increases the odds of being classified one step poorer in mental health by 9.5%.
- a very simple interpretation of association!

Log-multiplicative (RC) models I

- The $L \times L$, R, and C models are all simpler to interpret than the saturated model
- But, all depend on assigning fixed scores to the categories
- The ***row-and-column effects model*** (RC(1)) makes these parameters

$$\log(m_{ij}) = \mu + \lambda_i^A + \lambda_j^B + \gamma\, \alpha_i \beta_j \quad \text{or, } \lambda_{ij}^{AB} = \gamma\, \alpha_i \beta_j$$

  where γ, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ comprise additional parameters to be estimated beyond the independence model.
- γ here is $\sim$ to γ in the $L \times L$ model
- The ordering and spacing of the categories is estimated from the data (as in CA)
- Requires some constraints to be identifiable: e.g., unweighted solution–

$$\begin{aligned} \sum_i \alpha_i = \sum_j \beta_j &= 0 \\ \sum_i \alpha_i^2 = \sum_j \beta_j^2 &= 1 \end{aligned}$$

Log-multiplicative (RC) models II

- This generalizes to multiple bilinear terms, the RC(M) model

$$\lambda_{ij}^{AB} = \sum_{k=1}^{M} \gamma_k \, \alpha_{ik} \beta_{jk} \qquad M = \min(I-1, J-1)$$

- e.g., the RC(2) model has two bilinear terms (like a 2D CA solution)

$$\lambda_{ij}^{AB} = \gamma_1 \, \alpha_{i1} \beta_{j1} + \gamma_2 \, \alpha_{i2} \beta_{j2}$$

- RC models are not loglinear– contain multiplicative terms
  - Can't use `glm()`
  - The `gnm()` function in gnm fits a wide variety of such ***generalized nonlinear models***
  - The `rc()` function in logmult uses `gnm()` and makes plotting easier.

Generalized nonlinear models

The gnm package provides fully general ways to specify nonlinear GLMs

- Basic nonlinear functions: `Exp()`, `Inv()`, `Mult()`
- The RC(1) model: `gnm(Freq ~A + B + Mult(A,B)`
- The RC(2) model:
  `gnm(Freq ~A + B + instances(Mult(A,B),2)`
- Models for mobility tables— the UNIDIFF model

$$\log m_{ijk} = \alpha_{ik} + \beta_{jk} + \exp(\gamma_k)\delta_{ij}$$

  the exponentiated multiplier is specified as `Mult(Exp(C), A:B)`
- User-defined functions allow further extensions

Example: Mental impairment & SES

Fit the RC(1) and RC(2) models by adding terms using `Mult()` to the independence model

```
library(gnm)
indep <- gnm(Freq ~ mental + ses,
             family = poisson, data = Mental, verbose=FALSE)
RC1 <- update(indep, . ~ . + Mult(mental, ses))
RC2 <- update(indep, . ~ . + instances(Mult(mental, ses),2))
```

Compare models:

```
vcdExtra::LRstats(indep, linlin, roweff, coleff, RC1, RC2)

## Likelihood summary table:
##           AIC   BIC LR Chisq Df Pr(>Chisq)
## indep   209.6 220.2    47.42 15   3.16e-05 ***
## linlin  174.1 185.8     9.90 14      0.770
## roweff  174.4 188.6     6.28 12      0.901
## coleff  179.0 195.5     6.83 10      0.741
## RC1     179.7 198.6     3.57  8      0.894
## RC2     186.7 211.4     0.52  3      0.914
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

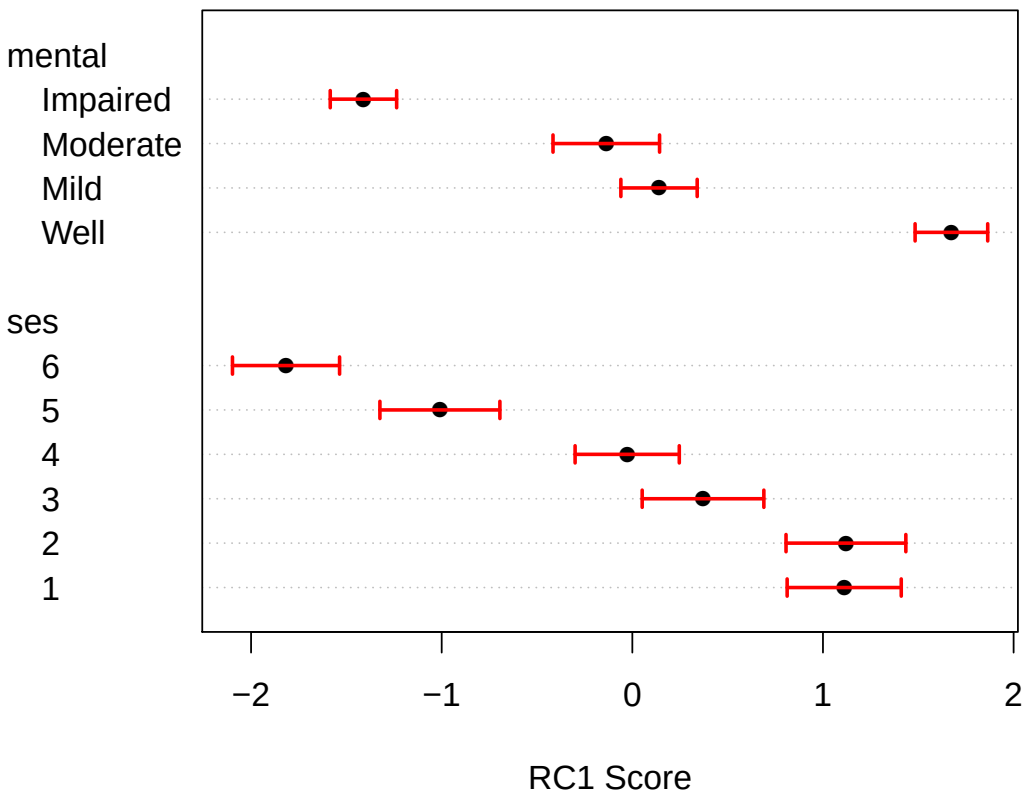
Comparing models

- Are estimated RC scores better than integer scores?
- If so, do we need more than one dimension?

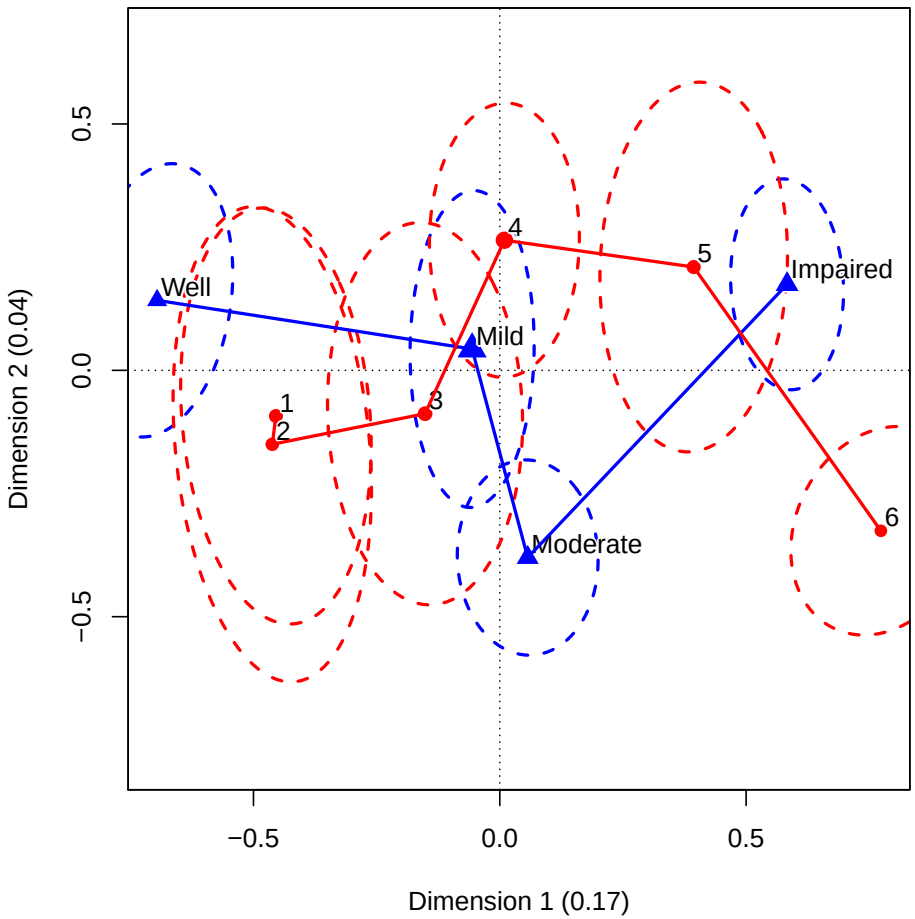
```
anova(linlin, RC1, RC2, test="Chisq")

## Analysis of Deviance Table
##
## Model 1: Freq ~ mental + ses + Rscore:Cscore
## Model 2: Freq ~ mental + ses + Mult(mental, ses)
## Model 3: Freq ~ mental + ses + Mult(mental, ses, inst = 1) + Mult(menta
##     ses, inst = 2)
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        14       9.90
## 2         8       3.57  6     6.32     0.39
## 3         3       0.52  5     3.05     0.69
```

Visualizing RC scores

- The RC(1) model can be interpreted visually using a dotplot of the scaled category scores together with error bars.
- This allows you to see where this model differs from the $L \times L$ model with integer spacing



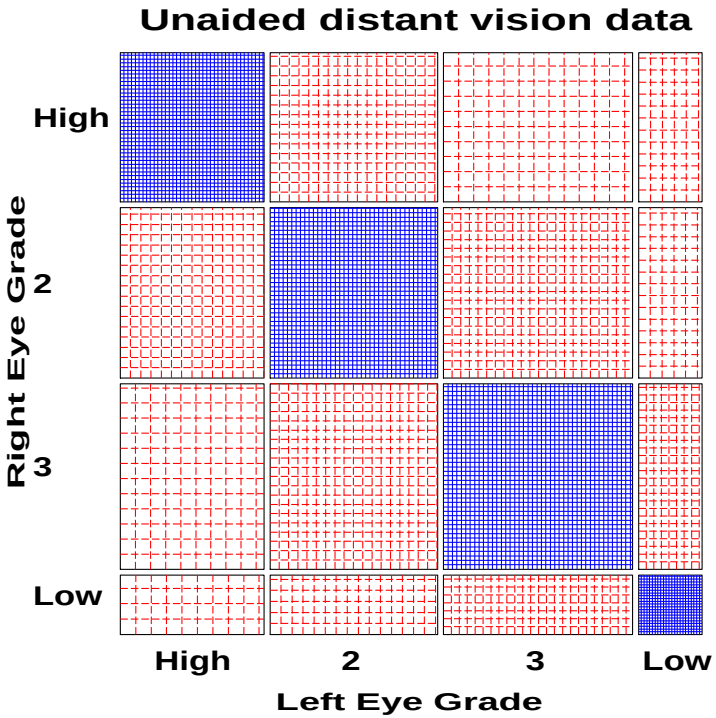
Visualizing RC scores



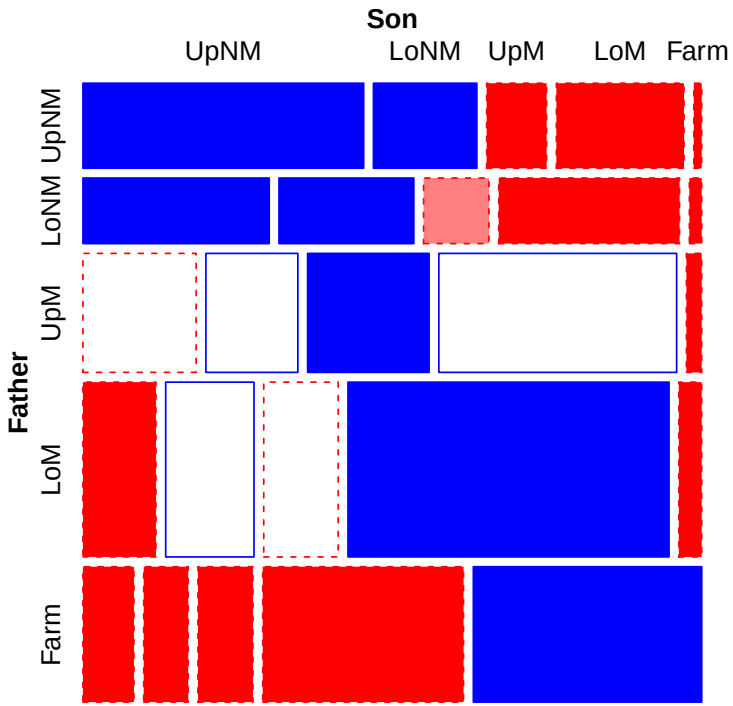
- For the RC(2) model, plot the category scores for dim. 1 and 2
- The logmult package makes these plots much easier
- Also, provides bivariate confidence ellipses

Square tables

Square tables arise when the row and column variables have the same categories, often ordered



Visual acuity data



Hauser social mobility data

Square tables: Models

In such cases, general association is a given, because of the diagonal cells. More interesting models concern associations in the off-diagonal cells

- ***Quasi-independence***: ignore the diagonal cells

$$\log m_{ij} = \mu + \lambda_i^A + \lambda_j^B + \delta_i I(i = j) \ .$$

  This model adds one parameter, δ_i, for each diagonal cell, which fits those frequencies perfectly.
- ***Symmetry***: $\pi_{ij} = \pi_{ji}$, but this implies marginal homogeneity, $\pi_{i+} = \sum_j \pi_{ij} = \sum_j \pi_{ji} = \pi_{+i}$ for all i.
- ***Quasi-symmetry***:

$$\log m_{ij} = \mu + \lambda_i^A + \lambda_j^B + \lambda_{ij} \ , \quad \lambda_{ij} = \lambda_{ji}$$

- It can be shown that

$$\begin{aligned} \text{symmetry} &= \text{quasi-symmetry} + \text{marginal homogeneity} \\ G^2(S) &= G^2(QS) + G^2(MH) \end{aligned}$$

Square tables: Models

For these models, the essential idea is to construct factor levels corresponding to the unique parameters representing association

$$\text{Diag}_{4\times4} = \begin{bmatrix} 1 & . & . & . \\ . & 2 & . & . \\ . & . & 3 & . \\ . & . & . & 4 \end{bmatrix} \qquad \text{Symm}_{4\times4} = \begin{bmatrix} 11 & 12 & 13 & 14 \\ 12 & 22 & 23 & 24 \\ 13 & 23 & 33 & 34 \\ 14 & 24 & 34 & 44 \end{bmatrix}$$

More general topological models allow any arbitrary pattern:

$$\text{Topo}_{4\times4} = \begin{bmatrix} 2 & 3 & 4 & 4 \\ 3 & 3 & 4 & 4 \\ 4 & 4 & 5 & 5 \\ 4 & 4 & 5 & 1 \end{bmatrix}$$

Square tables: Using gnm

Some models for structured associations in square tables:

- quasi-independence (ignore diagonals)

```
gnm(Freq ~ row + col + Diag(row, col), family=poisson)
```

- symmetry ($\lambda_{ij}^{RC} = \lambda_{ji}^{RC}$)

```
gnm(Freq ~ Symm(row, col), family=poisson)
```

- quasi-symmetry = quasi + symmetry

```
gnm(Freq ~ row + col + Symm(row, col), family=poisson)
```

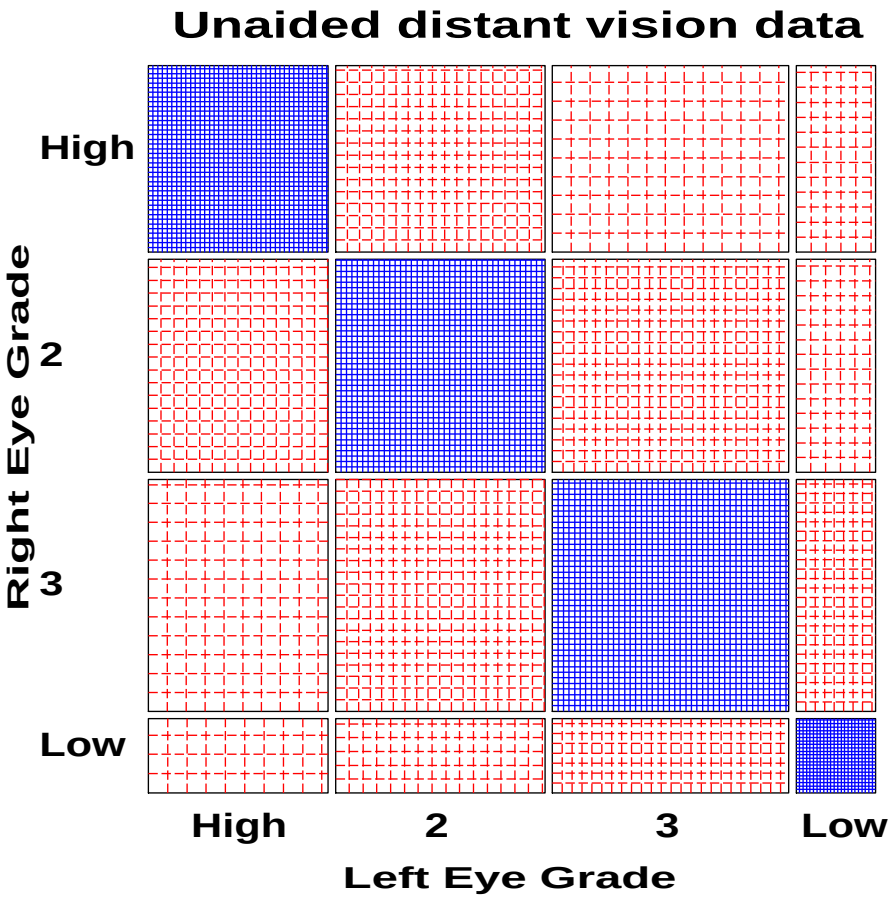
- fully-specified "topological" association patterns

```
gnm(Freq ~ row + col + Topo(row, col, spec=RCmatrix), ...)
```

All of these are actually GLMs, but the gnm package provides convienence functions `Diag`, `Symm`, and `Topo` to facilitate model specification.

Example: Visual acuity

```
data("VisualAcuity", package="vcd")
women <- subset(VisualAcuity, gender=="female", select=-gender)
```



- The diagonal cells clearly dominate
- What associations remain, ignoring these?
- Is there evidence for quasi-symmetry?

Example: Visual acuity— fitting models

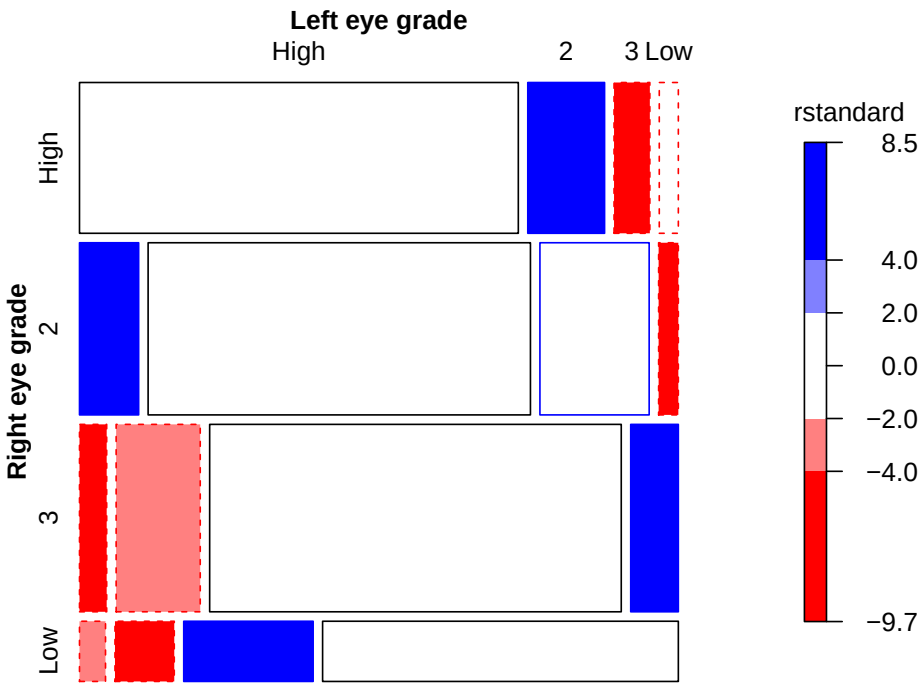
```
indep <- glm(Freq ~ right + left,  data = women, family = poisson)
quasi <- update(indep, . ~ . + Diag(right, left))

symm <- glm(Freq ~ Symm(right, left), data = women, family = poisson)
qsymm <- update(symm, . ~ right + left + .)
```

The QS model fits reasonably well, but none of the others do by likelihood-ratio tests or AIC or BIC.

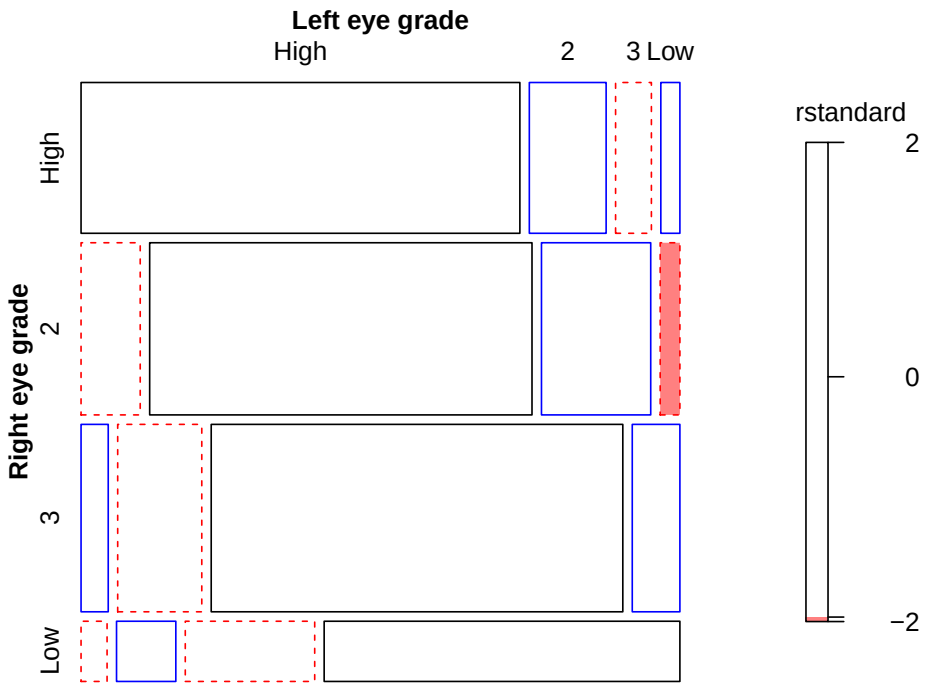
```
vcdExtra::LRstats(indep, quasi, symm, qsymm)

## Likelihood summary table:
##        AIC  BIC LR Chisq Df Pr(>Chisq)
## indep 6803 6808     6672  9     <2e-16 ***
## quasi  338  347      199  5     <2e-16 ***
## symm   157  164       19  6     0.0038 **
## qsymm  151  161        7  3     0.0638 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Example: Visual acuity— visualizing model fit

Quasi−Independence (women)



Quasi−Symmetry (women)



More complex models

- Extensions of these methods arise in a variety of contexts:
  - Panel surveys, where given attitude items are analyzed over time and space
  - Social mobility data, where occupational status of parents and children may admit subtly different models
  - Migration data, where geographical and political factors require some special treatment (e.g., mover-stayer models)
- These often involve:
  - ordinal variables: support for abortion, occumpational status
  - square tables: husbands/wives, fathers/sons, ...
  - strata or layers to control for other factors or analyze change over time or differences over geography

More complex models

- For example, the ***log-multiplicative uniform difference*** (UNIDIFF) model, for factors R, C, with layer variable L:

$$\log m_{ijk} = \mu + \lambda_i^R + \lambda_j^C + \lambda_k^L + \lambda_{ik}^{RL} + \lambda_{jk}^{CL} + \gamma_k \delta_{ij}^{RC}$$

  - The term for the three-way association [RCL] pertains to how the [RC] association varies with layer (L)
  - The UNIDIFF model says there is a multiplier γ_k for a common δ_{ij}^{RC} association
  - Special cases: R, C, RC(1) models for the [RC] association;
  - Special cases: homogeneous associations ($\gamma_k = 0$) for layers
- `gnm()` notation uses `Exp(L)`, so layer effects are on a log scale.
- The logmult package provides a `unidiff()` function that makes this easier.

Models for stratified mobility tables

Baseline models:

- Perfect mobility: `Freq ~(R+C)*L`
- Quasi-perfect mobility: `Freq ~(R+C)*L + Diag(R, C)`

Layer models:

- Homogeneous: no layer effects– $\gamma_k = 0$
- Heterogeneous: e.g., $\mu_{ijk}^{RCL} = \exp(\gamma_k^L)\, \delta_{ij}^{RC}$

Extended models: Baseline $\oplus$ Layer model(R:C model)

R:C model	**Layer model**	
	Homogeneous	log multiplicative
Row effects	`~.+ R:j`	`~.+ Mult(R:j, Exp(L))`
Col effects	`~.+ i:C`	`~.+ Mult(i:C, Exp(L))`
Row+Col eff	`~.+ R:j + i:C`	`~.+ Mult(R:j + i:C, Exp(L))`
RC(1)	`~.+ Mult(R, C)`	`~.+ Mult(R, C, Exp(L))`
Full `R:C`	`~.+ R:C`	`~.+ Mult(R:C, Exp(L)`

Example: Social mobility in US, UK & Japan

Data from Yamaguchi (1987): Cross-national comparison of occupational mobility in the U.S., U.K. and Japan.

```
Yama.tab <- xtabs(Freq ~ Father + Son + Country, data=Yamaguchi87)
structable(Country+Son~Father, Yama.tab[,,1:2])

##         Country   US                          UK
##         Son      UpNM LoNM  UpM  LoM Farm UpNM LoNM  UpM  LoM Farm
## Father
## UpNM             1275  364  274  272   17  474  129   87  124   11
## LoNM             1055  597  394  443   31  300  218  171  220    8
## UpM              1043  587 1045  951   47  438  254  669  703   16
## LoM              1159  791 1323 2046   52  601  388  932 1789   37
## Farm              666  496 1031 1632  646   76   56  125  295  191
```

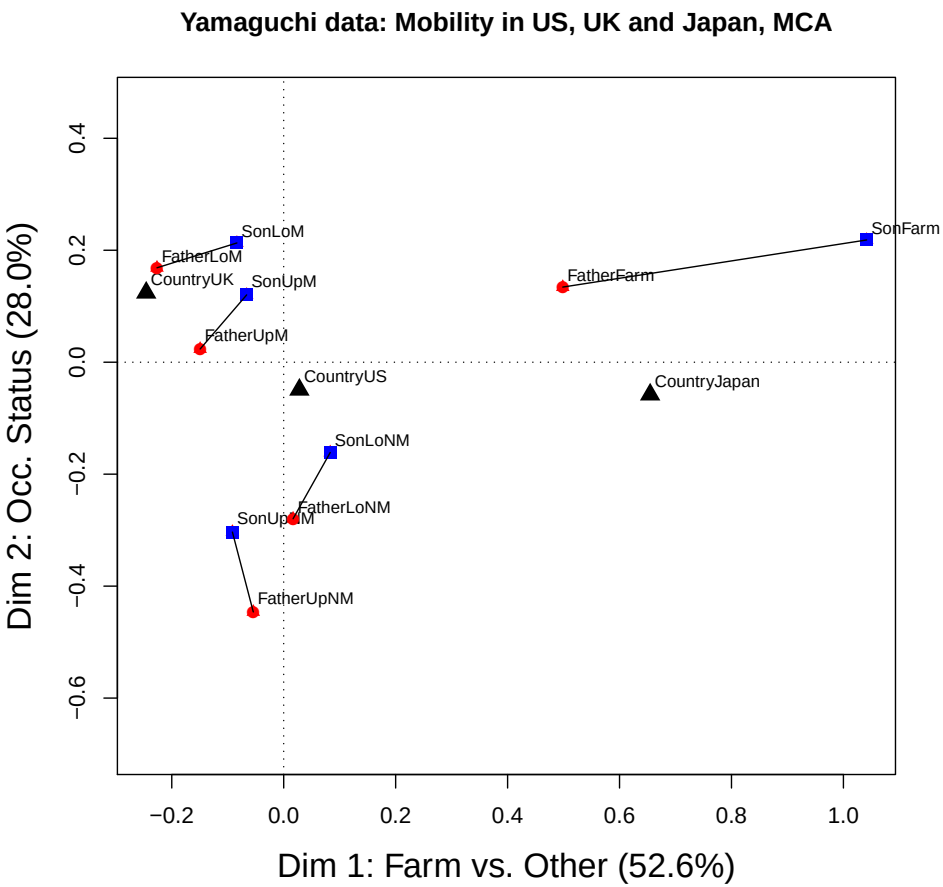
Questions:

- Is occupational mobility the same for all countries?
- If not, how do they differ?
- Are there simple models that describe mobility?

See: `demo("yamaguchi-xie", package="vcdExtra")`

First thought: try MCA

```
library(ca)
Yama.dft <- expand.dft(Yamaguchi87)
yama.mjca <- mjca(Yama.dft)
plot(yama.mjca, what=c("none","all"))
```



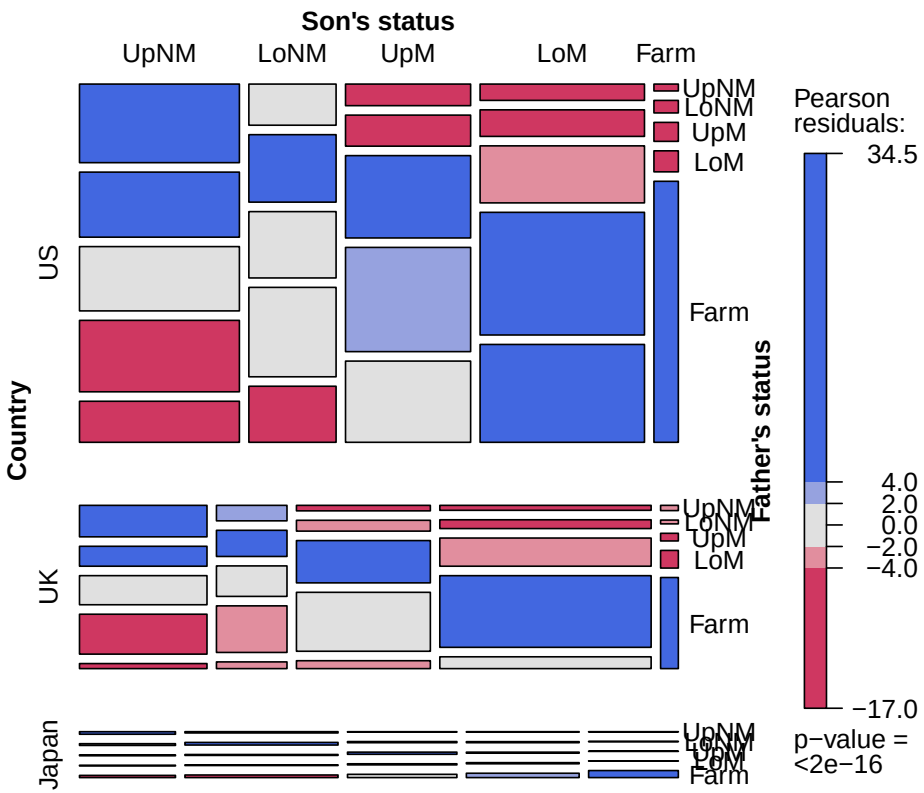
- Dimensions seem to have reasonable interpretations
- Farm differs from others
- All sons seem to move up!
- But, how do dims relate to theories of social mobility?
- How to understand Country effects?

Yamaguchi data: Baseline models

Minimal, null model asserts Father ⊥ Son | Country

```
yamaNull <- gnm(Freq ~ (Father + Son) * Country, data = Yamaguchi87,
     family = poisson)
mosaic(yamaNull, ~Country + Son + Father, condvars = "Country", ...)
```

[FC][SC] Null [FS] association (perfect mobility)

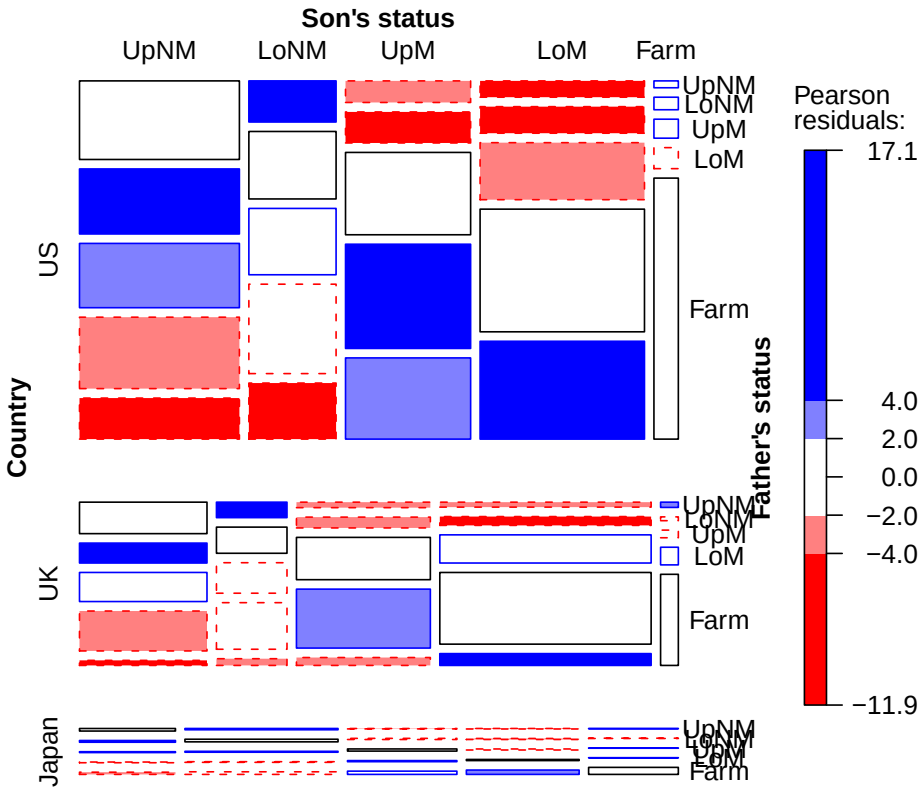


Yamaguchi data: Baseline models

But, for better theory $\Longrightarrow$ ignore diagonal cells

```
yamaDiag <- update(yamaNull, ~. + Diag(Father, Son):Country)
mosaic(yamaDiag, ~Country + Son + Father, condvars = "Country", ...)
```

[FC][SC] Quasi perfect mobility, +Diag(F,S)

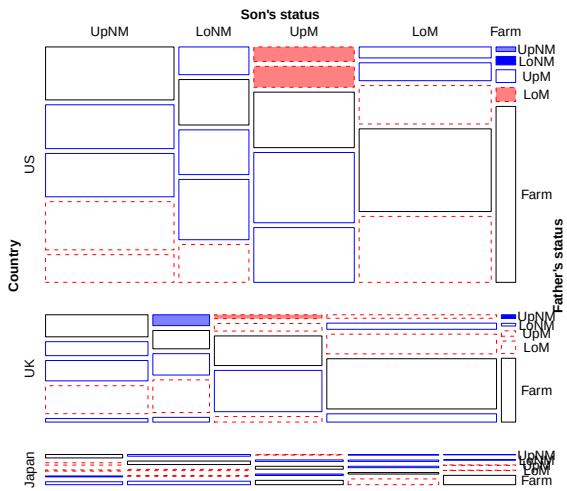


Models for homogeneous association

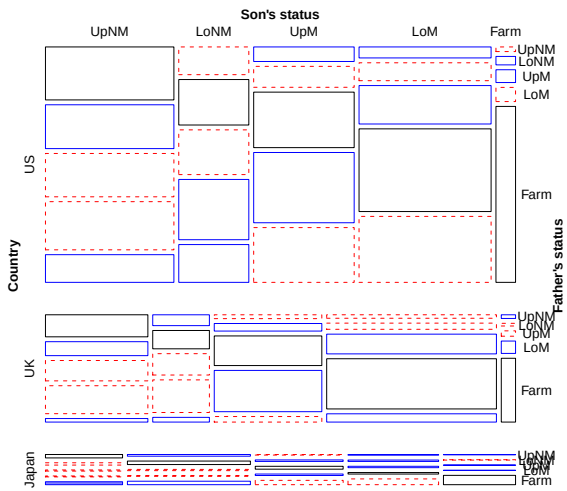
gnm makes it easy to fit collections of models, with simple `update()` methods

```
Rscore <- as.numeric(Yamaguchi87$Father)
Cscore <- as.numeric(Yamaguchi87$Son)

yamaRo <- update(yamaDiag, ~ . + Father:Cscore)
yamaCo <- update(yamaDiag, ~ . + Rscore:Son)
yamaRpCo <- update(yamaDiag, ~ . + Father:Cscore + Rscore:Son)
yamaRCo <- update(yamaDiag, ~ . + Mult(Father,Son))
yamaFIo <- update(yamaDiag, ~ . + Father:Son)
```

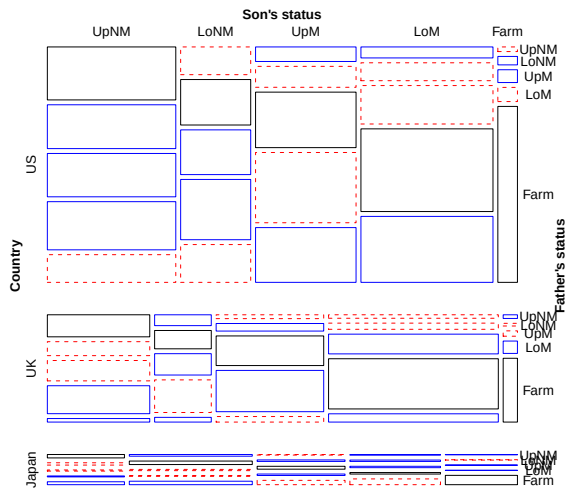
Model Ro: homogeneous row effects, +Father:j



Model Co: homogeneous col effects, +i:Son



Model RCo: homogeneous RC(1)



Models for heterogeneous association

Can combine these with models allowing layer effects
Log-multiplicative (UNIDIFF) models:

```
yamaRx <- update(yamaDiag, ~ . + Mult(Father:Cscore, Exp(Country)))
yamaCx <- update(yamaDiag, ~ . + Mult(Rscore:Son, Exp(Country)))
yamaRpCx <- update(yamaDiag, ~ . + Mult(Father:Cscore +
                                        Rscore:Son, Exp(Country)))
yamaRCx <- update(yamaDiag, ~ . + Mult(Father,Son, Exp(Country)))
yamaFIx <- update(yamaDiag, ~ . + Mult(Father:Son, Exp(Country)))
```

GNM model methods:

- Summary methods: `print(model)`, `summary(model)`, ...
- Extractor methods: `coef(model)`, `residuals(model)`, ...

Visualization:

- Diagnostics: `plot(model)`
- Mosaics, etc: `mosaic(model)`

Yamaguchi data: Comparing models

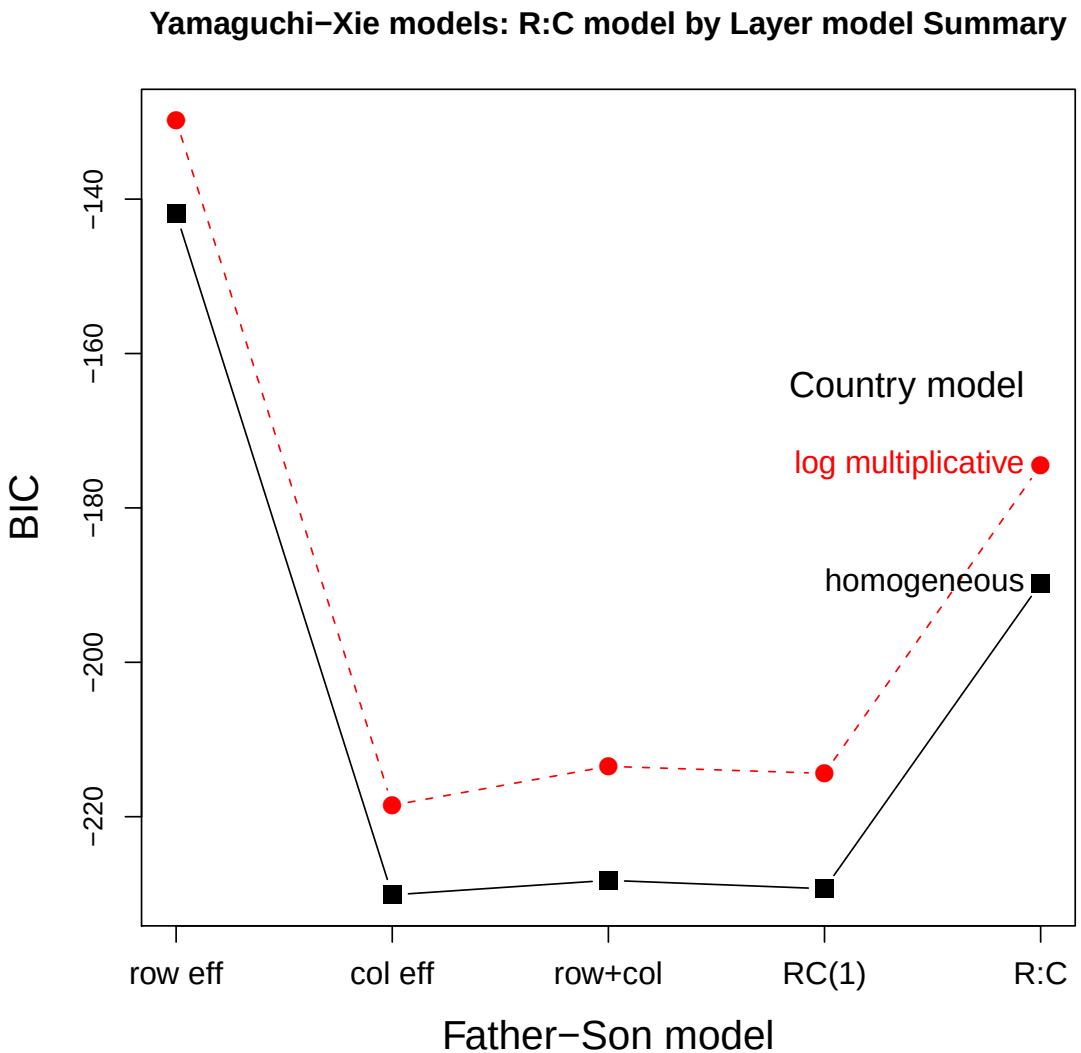
`LRstats()` and related methods facilitate model comparison

```
models <- glmlist(yamaNull, yamaDiag,
                  yamaRo, yamaRx, yamaCo, yamaCx, yamaRpCo,
                  yamaRpCx, yamaRCo, yamaRCx, yamaFIo, yamaFIx)
LRstats(models)

## Likelihood summary table:
##            AIC  BIC LR Chisq Df Pr(>Chisq)
## yamaNull  6168 6231     5592 48    < 2e-16 ***
## yamaDiag  1943 2040     1336 33    < 2e-16 ***
## yamaRo     771  877      156 29    < 2e-16 ***
## yamaRx     766  877      148 27    < 2e-16 ***
## yamaCo     682  789       68 29    6.1e-05 ***
## yamaCx     677  789       59 27    0.00038 ***
## yamaRpCo   659  773       39 26    0.05089 .
## yamaRpCx   658  776       33 24    0.10341
## yamaRCo    658  772       38 26    0.06423 .
## yamaRCx    657  775       32 24    0.12399
## yamaFIo    665  788       36 22    0.02878 *
## yamaFIx    664  791       31 20    0.05599 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Yamaguchi data: Comparing models

`LRstats()` and related methods facilitate model comparison

```
BIC <- matrix(LRstats(models)$BIC[-(1:2)], 5, 2, byrow=TRUE)
```

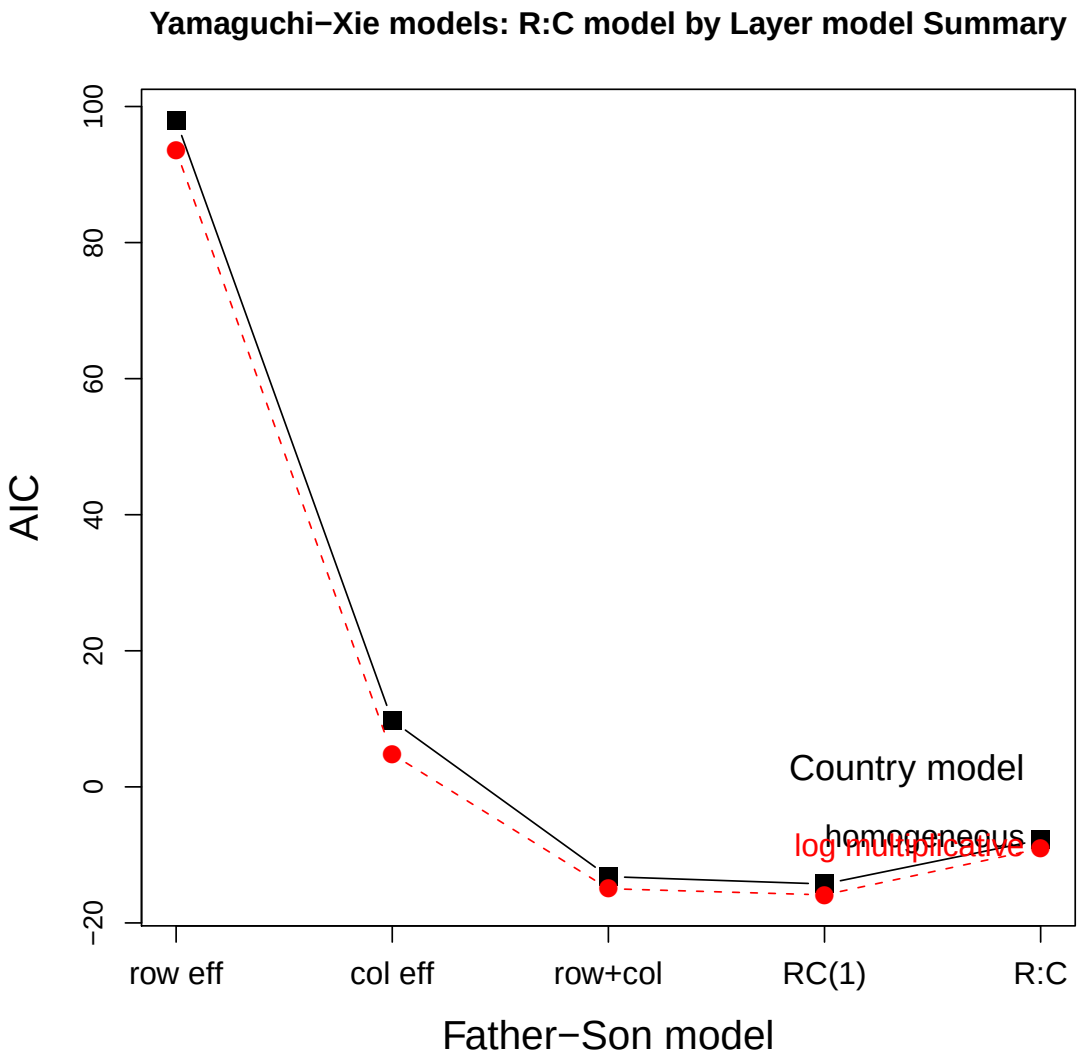


- Homogeneous models all preferred by BIC
- (Xie preferred heterogeneous models)
- Little diff^{ce} among Col, Row+Col and RC(1) models
- $\Longrightarrow$ `R:C` association $\sim$ Row scores (Father's status)

Yamaguchi data: Comparing models

`LRstats()` and related methods facilitate model comparison

```
AIC <- matrix(LRstats(models)$AIC[-(1:2)], 5, 2, byrow=TRUE)
```



- AIC prefers heterogeneous models
- Row+Col and RC(1) model fit best
- $\Longrightarrow$ `R:C` association $\sim$ Father's status estimates
- Model summary plots provide sensitive comparisons!

Yamaguchi data: Interpreting associations

unidiff() in logmult uses **gnm()**, but makes summaries and plotting easier

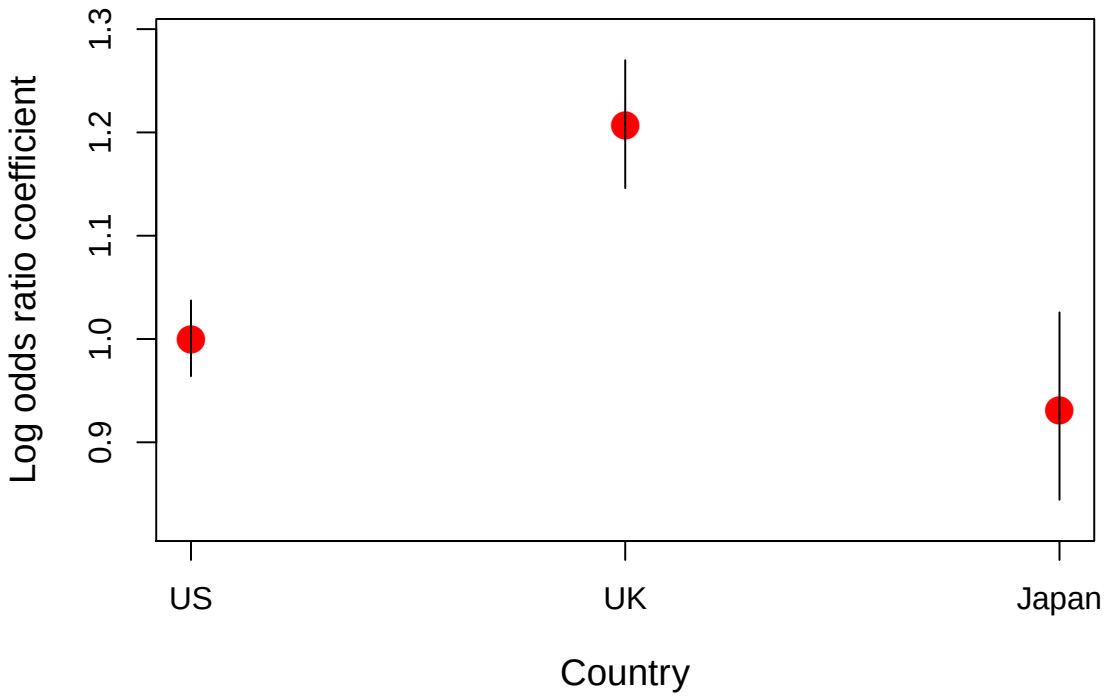
```
library(logmult)
yamaUni <- unidiff(Yama.tab)
```

```
yamaUni

## Call:
## unidiff(tab = Yama.tab)
##
##
## Layer coefficients:
##    US     UK  Japan
## 1.000  1.206  0.931
##
## Layer intrinsic association coefficients:
##    US     UK  Japan
## 0.412  0.497  0.383
##
## Full two-way interaction coefficients:
##        Son
## Father  UpNM     LoNM     UpM      LoM      Farm
##   UpNM   1.0063   0.3024  -0.4399  -0.6048  -0.4394
...
```

Yamaguchi data: Interpreting associations

Plotting the `"unidiff"` object plots the layer coefficients

```
plot(yamaUni, cex=2, col="red", pch=16)
```



Father – Son occupational association is ordered *UK* > *US* > *Japan*

Yamaguchi data: Visualizing associations

The common association parameters, δ_{ij}^{RC}, are contained in the `"unidiff"` object

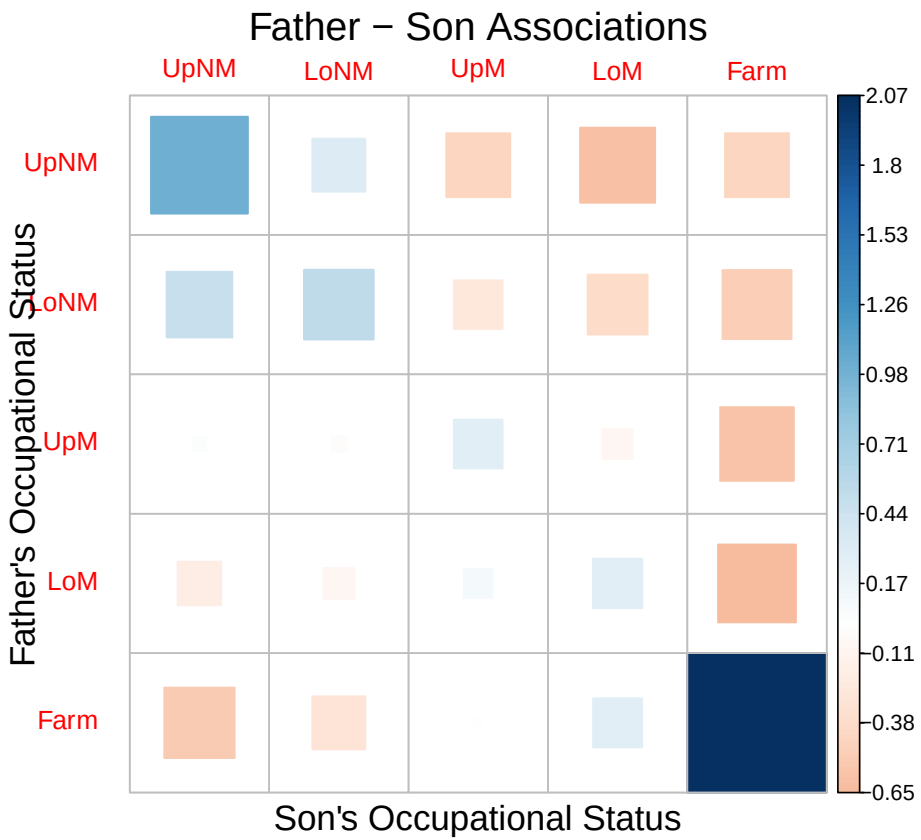
```
inter <- yamaUni$unidiff$interaction
inter.mat <- matrix(inter$Estimate, 5, 5,
                    dimnames=dimnames(Yama.tab)[1:2])
inter.mat
```

```
##        Son
## Father     UpNM     LoNM      UpM      LoM   Farm
##   UpNM   1.0063   0.3024 -0.4399 -0.6048 -0.439
##   LoNM   0.4644   0.5228 -0.2547 -0.3856 -0.512
##   UpM    0.0214 -0.0268  0.2557 -0.0972 -0.583
##   LoM   -0.2056 -0.1028  0.0891  0.2632 -0.650
##   Farm  -0.5320 -0.3026  0.0101  0.2592  2.075
```

Yamaguchi data: Visualizing associations

Plot these as a shaded-square plot using `corrplot()`

```
library(corrplot)
corrplot(inter.mat, method="square", is.corr=FALSE, ...)
```



Yamaguchi data: Visualizing associations

Plot these as a line plot using `matplot()`

```
matplot(t(inter.mat), type="b", pch=15:19, cex=1.2, xaxt="n",
        xlab="Father's status", ylab="Association estimate")
```