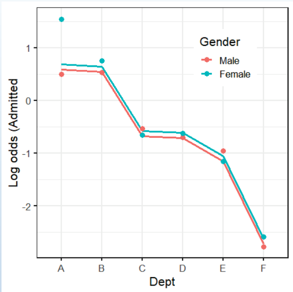
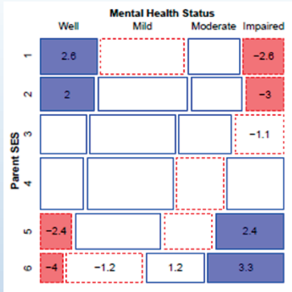
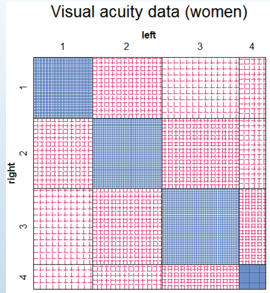






Extending loglinear models

Michael Friendly

Psych 6136

https://friendly.github.io/psy6136

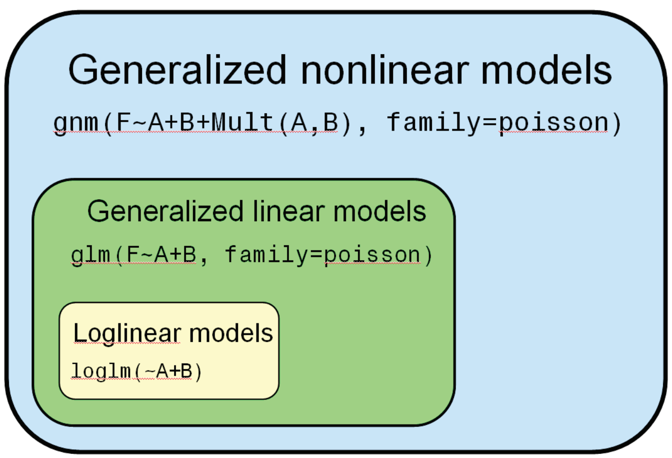
Today's topics

If you thought we were done with loglinear models, think again:

- Overview of extended loglinear models
- Logit models for response variables
- Models for ordinal factors
- RC models, estimating row/col scores
- Models for square tables
- More complex models

- The related languages of model specification and data vis give:
  - Tools for thinking!!
  - New visualization methods

Visual overview: Models for frequency tables



Related models: logistic regression, polytomous regression, log odds models, …
Goal: connect all with visualization methods

Extending loglinear models

Loglinear models can be extended in a variety of ways

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

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)

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

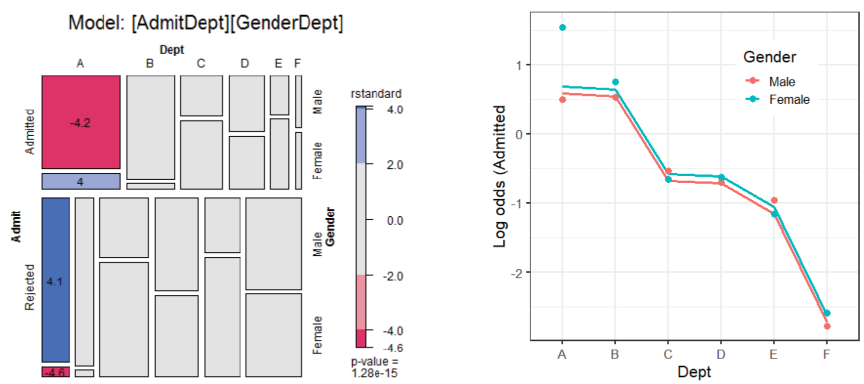
Loglinear models: Perspectives

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

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:

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

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

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

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

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, ≡ a loglinear model
  - The loglin model must include the [AB] association of predictors
  - When the response, C, has *m*>2 levels, multinomial models 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

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
```

Berkeley data: glm() approach

GLM approach, using glm()

- Convert `UCBadmissions` to a frequency data frame
- The `Freq` variable is used at 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
```


Berkeley data: glm() approach

GLM approach, using glm()

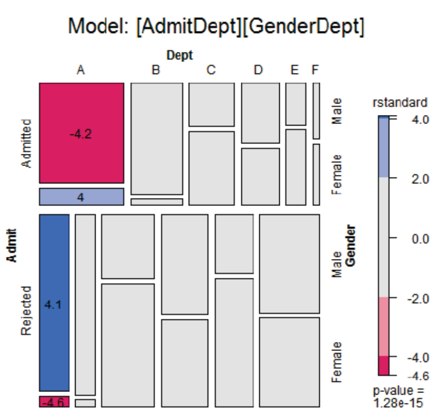
- Fit the same model of conditional independence, [AD][GD]
- This uses family = "poisson" to give model for log(Freq)

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

> 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
```

Hmm, doesn't look like a very good fit!

What does the mosaic plot tell us?

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



For a glm() model, mosaic() uses residuals from that model

Standardized residuals ("rstandard") have better statistical properties

Here, we see that the lack of fit is confined to Dept A

- G^2 = 21.7
- $\text{Resid}(4.2)^2 = 17.6$

Berkeley data: Logit approach

Logit approach, using `glm()` = [AD][GD]

- 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")

> Anova(berk.logit2, test="Wald")
Analysis of Deviance Table (Type II tests)

Response: Admit == "Admitted"
       Df  Chisq Pr(>Chisq)
Dept    5 534.71     <2e-16 ***
Gender  1   1.53       0.22
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

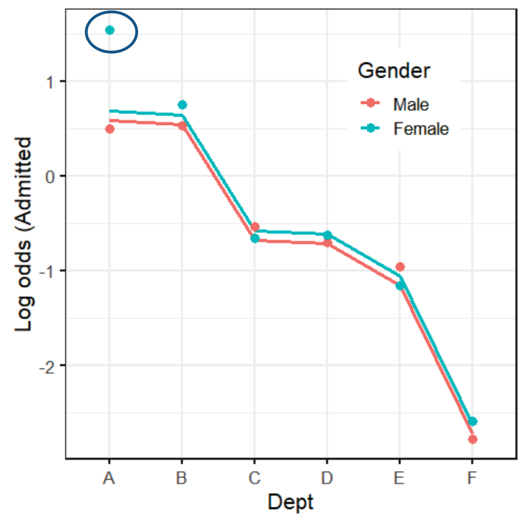
Plots for logit models

- Logit models are easier to interpret because there are fewer parameters
- Easiest to interpret from plots of the fitted & observed 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  0.58  0.49
3  Admitted Female    A  0.68  1.54
5  Admitted   Male    B  0.54  0.53
7  Admitted Female    B  0.64  0.75
9  Admitted   Male    C -0.68 -0.54
11 Admitted Female    C -0.58 -0.66
```



```
ggplot(pred2, aes(x=Dept, y=fit, group=Gender, color=Gender)) +
  geom_line(linewidth=1.4) +
  geom_point(aes(y=obs), size=3) + …
```



Large effect 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:

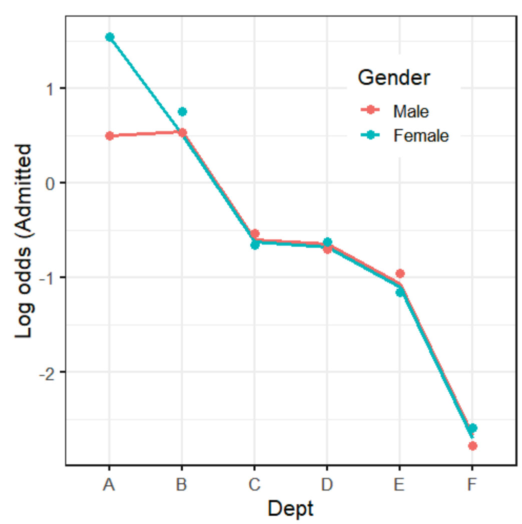
$$\log m_{ijk} = \mu + \lambda_i^A + \lambda_j^D + \lambda_k^G + \lambda_{ij}^{AD} + \lambda_{jk}^{DG} + \boxed{I(j=1)\lambda_{ik}^{AG}} \; ,$$

- 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
```


Plot observed and fitted values from this model



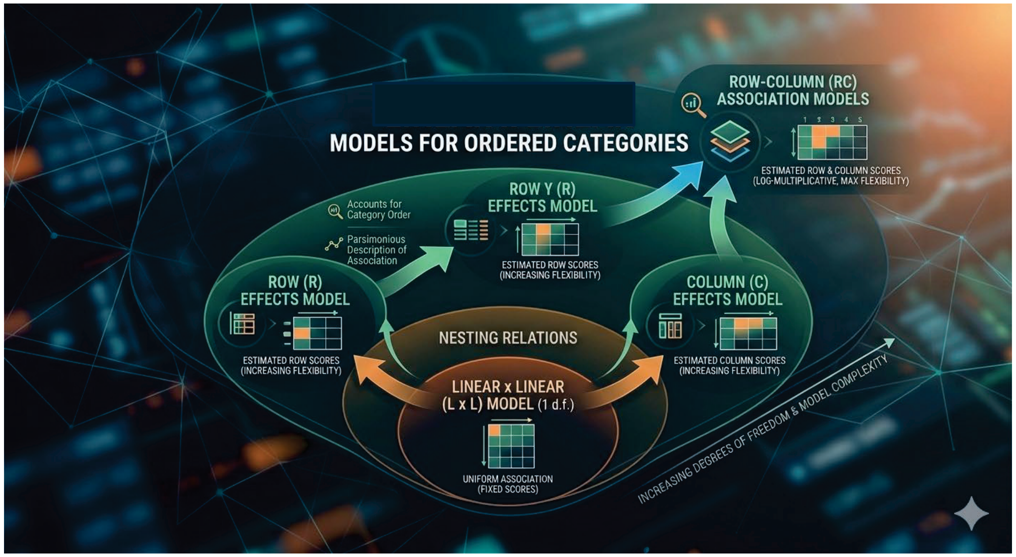
Large effect of Dept on admission

No effect of Gender

Perfect fit now for Dept A (at the expense of 1 df)

Lesson: glm() allows you to fit models to account for unusual cells

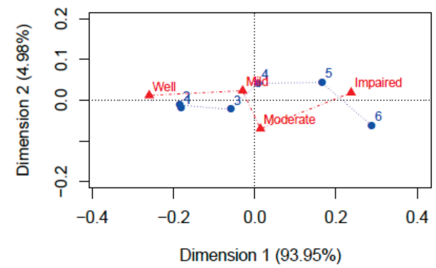
Models for Ordinal Variables



Loglinear models for ordinal variables

Ordinal variables reveal themselves in different ways in exploratory plots

- In correspondence analysis, one large dimension accounting for most of χ^2
- In mosaic plots, an opposite corner pattern of residuals



Advantages of ordinal models

- More focused tests → greater power to detect
- Use fewer df → can fit different models between independence [A][B] and saturated [AB]
  - Fewer parameters → easier interpretation
  - Fewer parameters → smaller std. errors

These are similar to reasons for using:

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

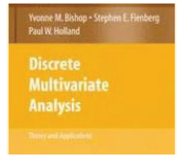
Models for ordered categories

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

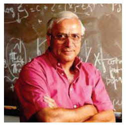
- In many cases, the *RC* association may be described more simply by assigning numeric scores to the row & column categories.
- For simplicity, we consider only integer scores, 1, 2, . . . here
- These models are easily extended to stratified tables

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`

History corner

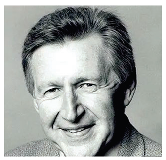


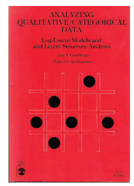
Original ideas for loglinear models came from Bishop, Fienberg, Holland (1974), *Discrete Multivariate Analysis*



Established a language & notation for loglinear models as ~ ANOVA for log(freq)

Leo Goodman, sociologist at U. Chicago worked out the extensions to a wide range of contexts: numeric scores for categories, estimating latent category scores, …





Linear x Linear Model (Uniform association)

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

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

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

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

One more term for association

- 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 & column effects: R, C, R+C

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

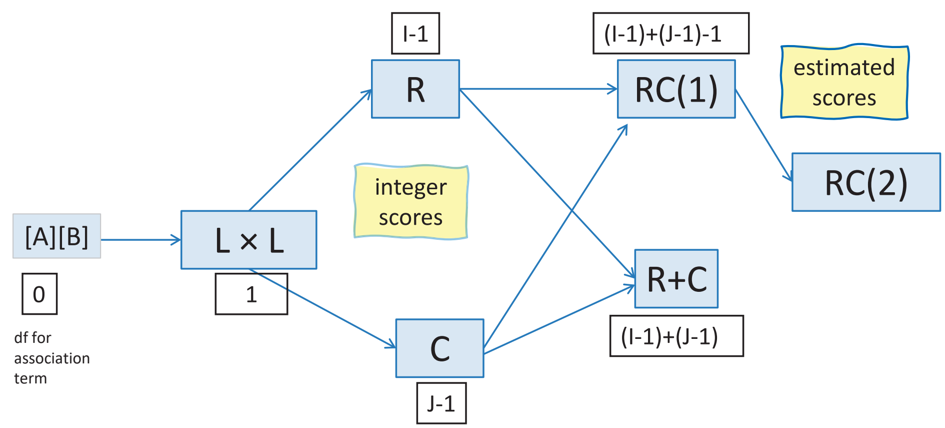
$$\log(m_{ij}) = \mu + \lambda_i^A + \lambda_j^B + \alpha_i b_j \quad \ni \quad \sum_i \alpha_i = 0 \text{ or } \alpha_1 = 0$$

- 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 relations among models for ordinal variables



Example: Mental impairment & SES

Data on mental health status of NYC youth in relation to parents' SES
Note that ses & mental have been declared as ordered factors

```
> str(Mental)
'data.frame':	24 obs. of  3 variables:
 $ ses   : Ord.factor w/ 6 levels "1"<"2"<"3"<"4"<..: 1 1 1 1 2 2 2 2 3 3 ...
 $ mental: Ord.factor w/ 4 levels "Well"<"Mild"<..: 1 2 3 4 1 2 3 4 1 2 ...
 $ Freq  : int  64 94 58 46 57 94 54 40 57 105 ...
```

Display it as a 2-way table

```
> (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
```


Example: Mental impairment & SES

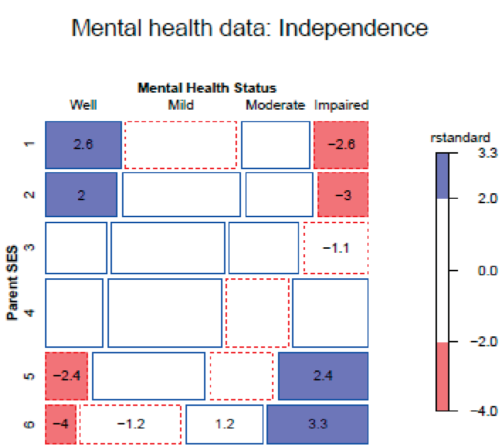
Fit and test the independence model using glm()

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

> vcdExtra::LRstats(indep)
Likelihood summary table:
         AIC    BIC LR Chisq Df Pr(>Chisq)    
indep 209.59 220.19   47.418 15  3.155e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

FIT!
Or FIT NOT!
There is no TRY

Yoda: Look at the mosaic, Luke!

```
> mosaic(indep, residuals_type="rstandard",
         labeling=labeling_residuals,
         main="Mental health data: Independence")
```



Departures from independence show the classic opposite corner pattern

The mosaic uses discrete shading levels, so it is useful to show residuals in the cells

NB: This shading scheme:
`shading=shading_Friendly`
was designed for B/W reproduction.
What would be better?

Local odds ratios

For ordered tables, useful to examine the local log odds ratios for successive 2 x 2 sub-tables
These would all be ≈ 0 under independence

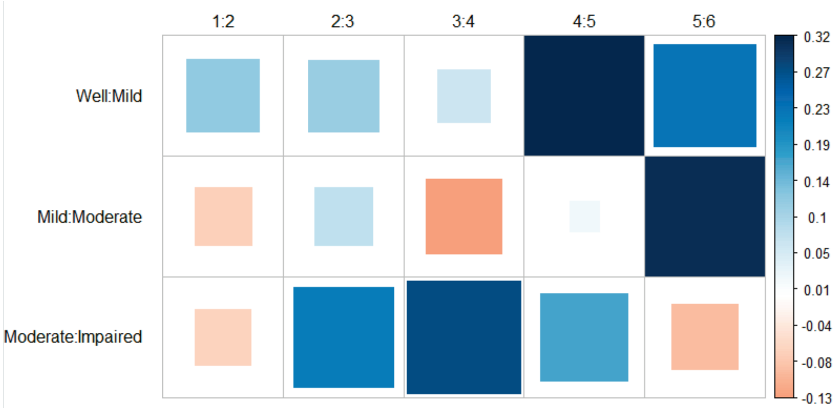
```
> (LMT <- loddsratio(t(mental.tab)))
log odds ratios for mental and ses 

                   ses
mental                  1:2    2:3     3:4    4:5    5:6
  Well:Mild          0.1158 0.1107  0.0612 0.3191  0.227
  Mild:Moderate     -0.0715 0.0747 -0.1254 0.0192  0.312
  Moderate:Impaired -0.0683 0.2201  0.2795 0.1682 -0.094

> mean(LMT$coefficients)
[1] 0.103
> mean(LMT$coefficients) |> exp()
[1] 1.11
```

On average, a one-unit step down the SES scale multiplies the odds of one worse mental health classification by exp(0.103) = 1.11 (11% increase)

Local odds ratios

We can plot these as area- and color-proportional shaded squares using **corrplot()**

```
corrplot(as.matrix(LMT), method="square", is.corr = FALSE,
         tl.col = "black", tl.srt = 0, tl.offset=1)
```



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)
```

Recall: in R, an interaction term, **A:B** is represented by the product, $a_i \times b_j$ of the parameters, a_i, b_j, for the factors.
`Rscore, Cscore` here are just numbers, so are not estimated parameters

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

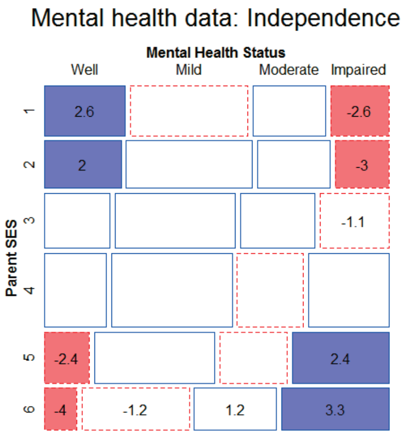
When overall tests are unclear, you can carry out tests of nested sets of models using **anova()**, giving tests of ΔG^2.
The indep, linlin and row effect models are one nested set:

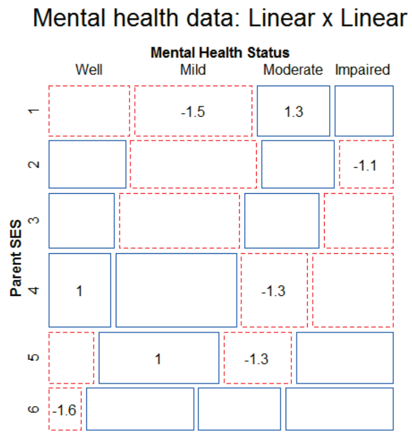
```
anova(indep, linlin, roweff, test="Chisq")

## 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

Comparing models: Mosaic plots

Beyond statistical tests, mosaic plots show the remaining structure in the residuals, unaccounted for in a given model.





Interpreting the $L \times L$ model

In the $L \times L$ model, the parameter γ is the constant local odds ratio. e^{γ} is the multiplier of the odds for a one-step change in mental or ses

```
> coef(linlin)[["Rscore:Cscore"]]
[1] 0.09069

> coef(linlin)[["Rscore:Cscore"]] |> exp()
[1] 1.095
```

- $\hat{\gamma} = 0.0907 \implies$ 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

- 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, α and β 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–

$$\sum_i \alpha_i = \sum_j \beta_j = 0$$
$$\sum_i \alpha_i^2 = \sum_j \beta_j^2 = 1$$

Log-multiplicative (RC) models

- 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) model 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 with GOF tests and AIC, BIC

```
> vcdExtra::LRstats(indep, linlin, roweff, coleff, RC1, RC2)
Likelihood summary table:
       AIC BIC LR Chisq Df Pr(>Chisq)
indep  210 220     47.4 15    3.2e-05 ***
linlin 174 186      9.9 14       0.77
roweff 174 189      6.3 12       0.90
coleff 179 196      6.8 10       0.74
RC1    180 199      3.6  8       0.89
RC2    187 211      0.5  3       0.91
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```


Comparing models

`anova()` again gives tests of $\Delta\chi^2$ for nested models

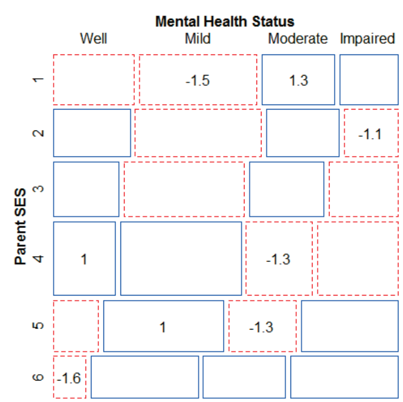
- Are estimated RC scores better than integer scores in the L x L model?
- If so, do we need more than one dimension?

```
> 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(mental, 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
```

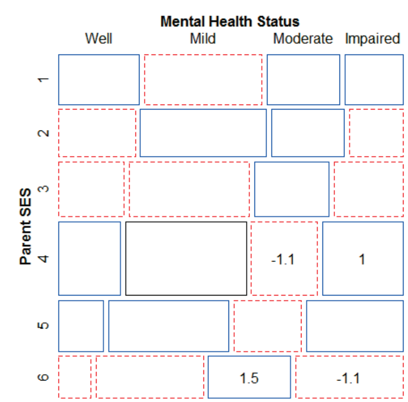
Neither RC model shows a significant advantage over the L x L model

Comparing models: Mosaic plots

Mental health data: Linear x Linear

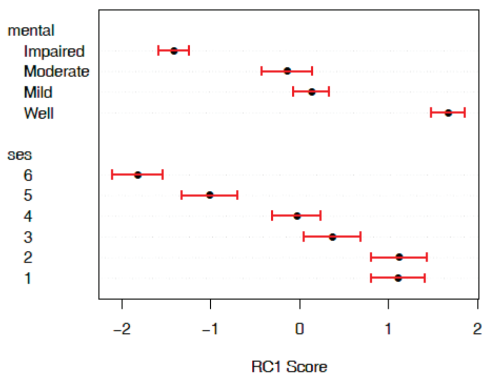


Mental health data: RC(1) model



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



mental: mild & moderate not that different, but ordered correctly

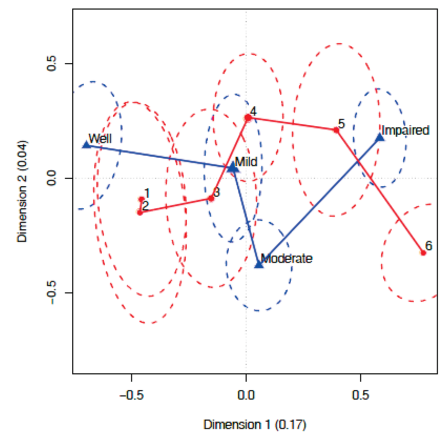
ses: approx. linear, except for ses = (1,2), which don't differ

Similar to what we saw in CA

When this matters, RC models provide the statistical machinery for inference

Visualizing RC scores

```
rc2 <- rc(Mental.tab, nd=2, weighting="marginal", se="jackknife")
coords  <- plot(rc2, conf.ellipses=0.68, cex=1.5,
                rev.axes=c(TRUE, FALSE))
```

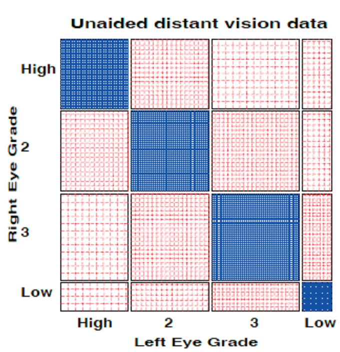


- For the RC(2) model, plot the category scores for dim. 1 and 2
- The logmult package makes these plots much easier
- Also, provides bivariate confidence ellipses

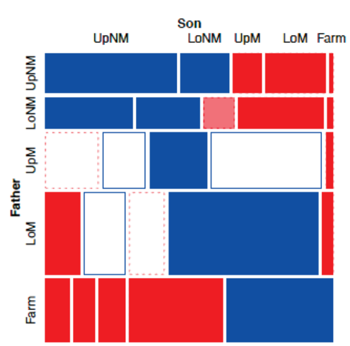
Square tables

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

Special loglinear models allow us to tease apart different reasons for association



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 the nature of association in 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

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

Diag adds *k* parameters to fit diagonal cells, beyond independence
Symm adds k x (k+1) parameters to fit a symmetric pattern of association

More general topological models allow an arbitrary pattern of association, but more parsimonious than the independence model

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

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

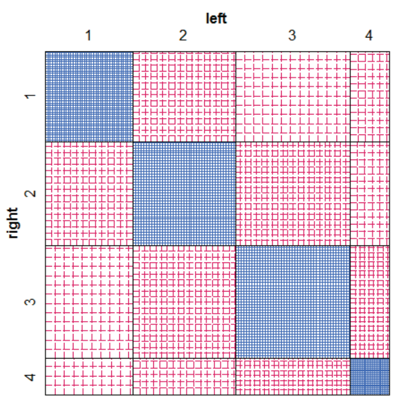
```
gnm(Freq ~ row + col + Topo(row, col, spec=RCmatrix), ...)
```

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)
sieve(Freq ~ right + left, data=women, shade = TRUE,
      main = "Unaided distance vision data")
```

Visual acuity data (women)



Diagonal cells clearly dominate

What associations remain, ignoring these?

Is there evidence for quasi-symmetry?

A more complete analysis could examine gender in relation to these associations

Fitting models

Start with the independence model, then **`update()`** to add other terms

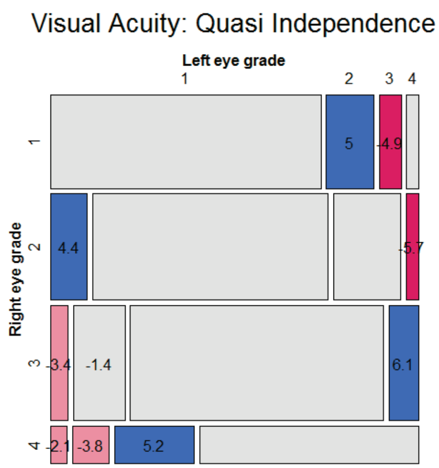
```
> indep <- glm(Freq ~ right + left,  data = women, family = poisson)
> quasi <- update(indep, . ~ . + Diag(right, left))

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

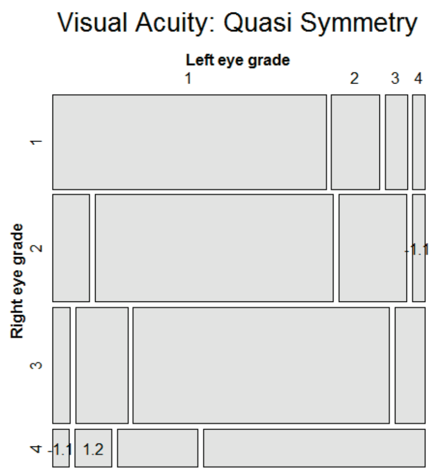
The quasi-symmetry model (qsymm) fits reasonably well; none of the others do by LR G^2 tests or AIC, BIC; qsymm is best by AIC, BIC

```
> 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
```

Visualizing model fits

Quasi-independence: The diagonal cells are forced to fit exactly.
Lack-of-fit appears in the symmetrically opposite cells





Hauser79 data: Occupational mobility

Square tables of occupational categories are a prime example of where various models provide substantiative interpretations of aspects of occupational mobility

```
> data(Hauser79,
package="vcdExtra")
> head(Hauser79)
   Son Father Freq
1 UpNM   UpNM 1414
2 LoNM   UpNM  521
3  UpM   UpNM  302
4  LoM   UpNM  643
5 Farm   UpNM   40
6 UpNM   LoNM  724
```

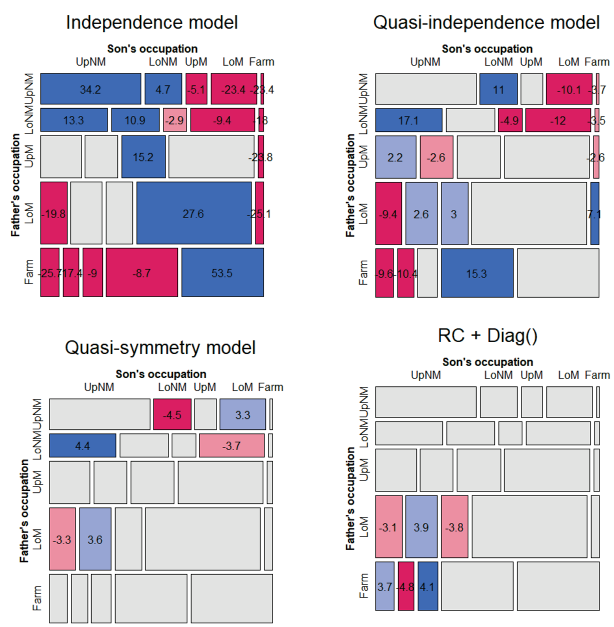
```
> structable(~Father+Son, data=Hauser79)
       Son UpNM LoNM  UpM  LoM Farm
Father
UpNM       1414  521  302  643   40
LoNM        724  524  254  703   48
UpM         798  648  856 1676  108
LoM         756  914  771 3325  237
Farm        409  357  441 1611 1832
```


More models, more mosaics

For the Hauser79 data on occupational mobility, there are a wide variety of models to consider

```
library(gnm)
hauser.indep <- gnm(Freq ~ Father + Son,
                    data=Hauser79, family=poisson)
hauser.quasi <- update(hauser.indep, ~ . + Diag(Father,Son))
hauser.qsymm <- update(hauser.indep, ~ . + Diag(Father,Son) + Symm(Father,Son) )

# numeric scores
Fscore <- as.numeric(Hauser79$Father)
Sscore <- as.numeric(Hauser79$Son)
hauser.UA     <- update(hauser.indep, ~ . + Fscore:Sscore)
hauser.roweff <- update(hauser.indep, ~ . + Father:Sscore)
hauser.UAdiag <- update(hauser.UA, ~ . + Diag(Father,Son))

# RC models, estimating category scores
hauser.RC <- update(hauser.indep, ~ . + Mult(Father, Son), verbose=FALSE)
hauser.RCdiag <- update(hauser.RC, ~ . + Diag(Father, Son), verbose=FALSE)

# crossings models
hauser.CR <- update(hauser.indep, ~ . + Crossings(Father,Son))
hauser.CRdiag <- update(hauser.CR, ~ . + Diag(Father,Son))
```


More models, more mosaics



Mosaic plots reveal the pattern of lack-of-fit

For more sensitive comparisons, we need model fit statistics

Q:

- How to interpret quasi-independence?
- Quasi-symmetry?
- RC?
- RC+Diag()?

Model comparisons

Collect the models in a glmlist() and compare them using LRstats():

```
modlist <- glmlist(hauser.indep, hauser.roweff, hauser.UA,
                   hauser.UAdiag, hauser.quasi, hauser.qsymm,
                   hauser.topo, hauser.RC, hauser.CR, hauser.CRdiag)

LRstats(modlist, sortby = "BIC")
```

Sorting by BIC shows the best models at the bottom:

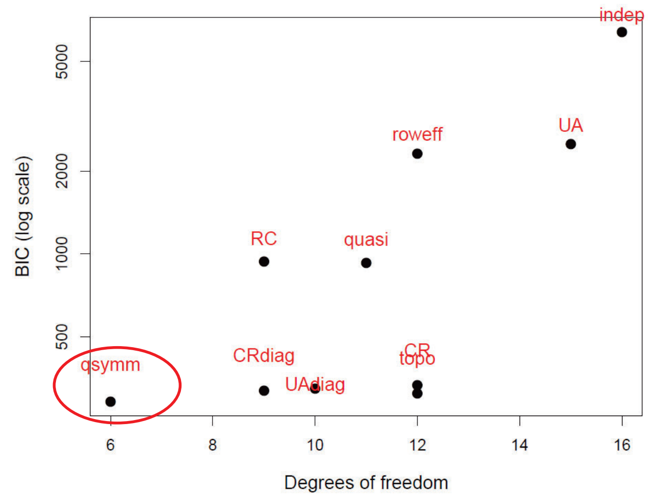
```
Likelihood summary table:
                 AIC    BIC LR Chisq Df Pr(>Chisq)
hauser.indep  6390.8 6401.8   6170.1 16  < 2.2e-16 ***
hauser.UA     2503.4 2515.6   2280.7 15  < 2.2e-16 ***
hauser.roweff 2308.9 2324.7   2080.2 12  < 2.2e-16 ***
hauser.RC      920.2  939.7    685.4  9  < 2.2e-16 ***
hauser.quasi   914.1  931.1    683.3 11  < 2.2e-16 ***
hauser.CR      318.6  334.5     89.9 12  5.131e-14 ***
hauser.UAdiag  305.7  324.0     73.0 10  1.161e-11 ***
hauser.CRdiag  298.9  318.5     64.2  9  2.030e-10 ***
hauser.topo    295.3  311.1     66.6 12  1.397e-09 ***
hauser.qsymm   268.2  291.3     27.4  6  0.0001193 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The quasi-symmetry model is best, but still shows some lack of fit

Model comparison plots

When there are more than a few models, a model comparison plot can show the trade-off between goodness-of-fit and parsimony

- This sorts the models by both fit & complexity



Plot BIC vs. df

Can also use AIC, or G^2 / df in this plot

Plot on log scale to emphasize diff[ce] among better models

And, the winner is: Quasi-symmetry!

More complex models

- Extensions of these methods occur in a variety of contexts:
  - Panel surveys, where attitude items are analyzed over time & space
  - Social mobility data, where occupational status of parents and children may admit subtly different models across strata
  - Migration data, where geographical & political factors require special treatment (e.g., mover-stayer models)
- These often involve:
  - ordinal variables: support for abortion, occupational 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 ⊕ Layer model(R:C model)

R:C model	Layer model: Homogeneous	Layer model: 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

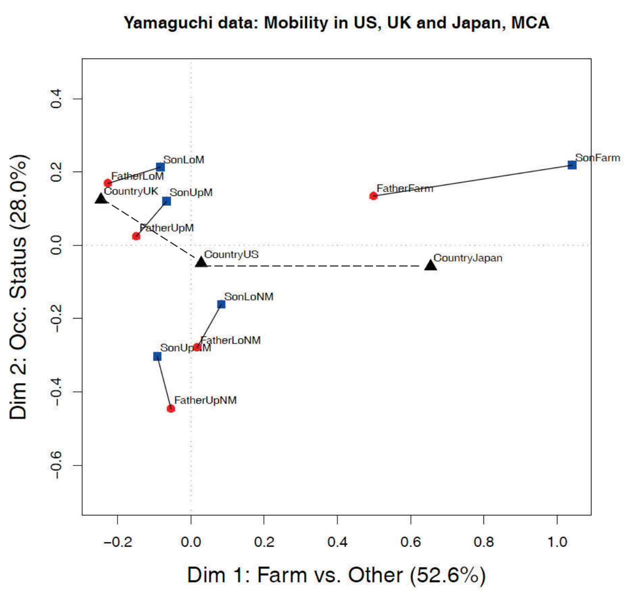
```
> data(Yamaguchi87, package="vcdExtra")
> Yama.tab <- xtabs(Freq ~ Father + Son + Country, data=Yamaguchi87)
> structable(Country+Son~Father, Yama.tab[,,1:2])
       Country   US                            UK
       Son       UpNM LoNM  UpM  LoM Farm |UpNM LoNM  UpM  LoM Farm
Father
UpNM             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? (No layer effects)
- If not, how do they differ?
- Are there simple models that describe mobility?

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

Explore: Try MCA




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

Dimensions have reasonable interpretations
Farm differs from others
All sons seem to move up!

How does this relate to theories of mobility?

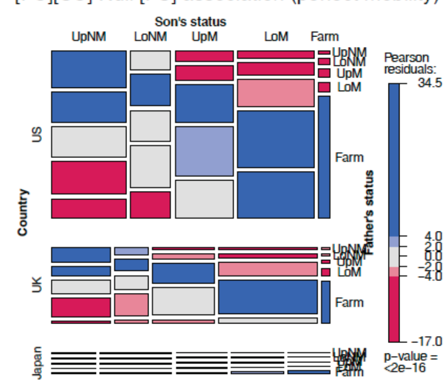
How to understand country effects?

Yamaguchi data: Baseline models

The minimal, null model asserts Father ⊥ Son | Country = [FC][SC] = (F+S)*C

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

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



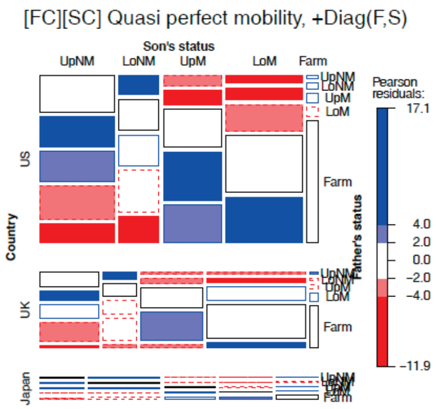
Within country, diagonal cells for F=S dominate

Much more data for US; least for Japan

Yamaguchi data: Baseline models

We expect F ≈ S. Ignore diagonal cells with quasi-independence → Quasi-perfect mobility

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

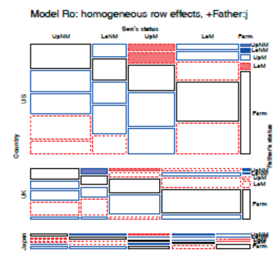


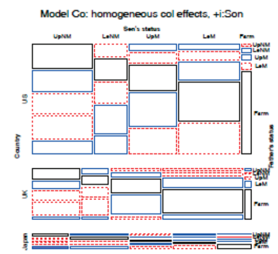
The term Diag(F, S):Country fits diagonal cells perfectly w/in each country

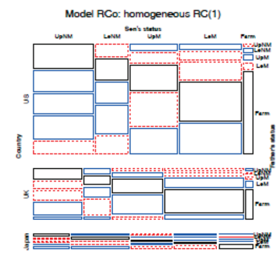
Models for homogeneous associations

gnm(): easy to fit collections of models using update() to the yamaDiag model.
These have no Country term, so they assert same associations for all countries

```
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)
```







Models for heterogeneous associations

Can combine these with models including layer (Country) effects
Log-multiplicative (UNIDIFF) models add a term Mult(…, Exp(Country))

```
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)))
```

We now have quite a collection of alternative models

- How to compare them?
- How to interpret the associations they imply about Father, Son mobility across countries?

Yamaguchi data: Comparing models

LRstats() and related methods facilitate model comparisons

```
> 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
```

Baseline models (yamaNull, yamaDiag)

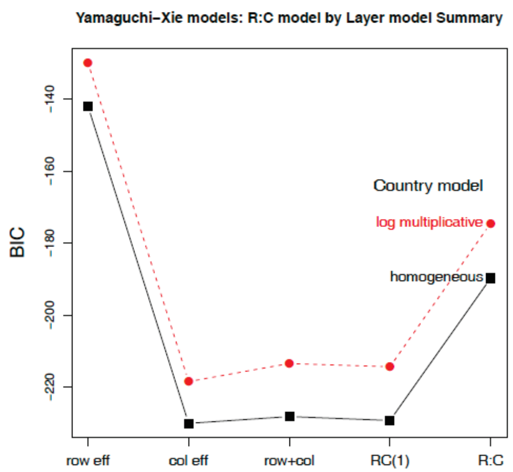
Homogeneous, Father:Son models (yamaRo, yamaRx, yamaCo, yamaCx)

Heteroogeneous, Father:Son models (yamaRpCo to yamaFIx)

Yamaguchi data: Comparing models

Easier to understand by plotting the criteria for these models

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



BIC strongly prefers homogeneous models

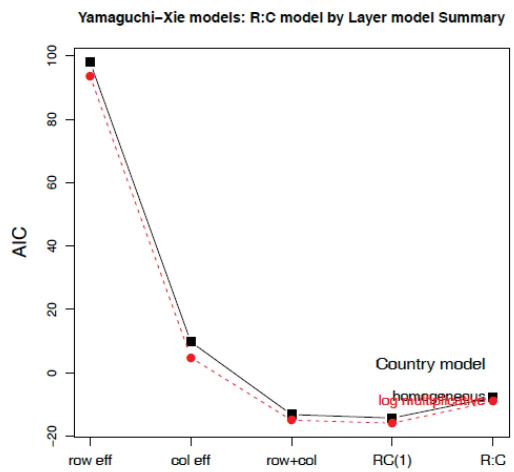
Little diffce among Col, Row+Col, RC(1) models

→ R:C association ~ Row scores (fathers' status)

Yamaguchi data: Comparing models

Easier to understand by plotting the criteria for these models

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



AIC slightly prefers heterogeneous models

Row + Col & RC(1) fit best

→ R:C association ~ ordinal scores

Model summary plots make sense of multiple models

Interpreting associations

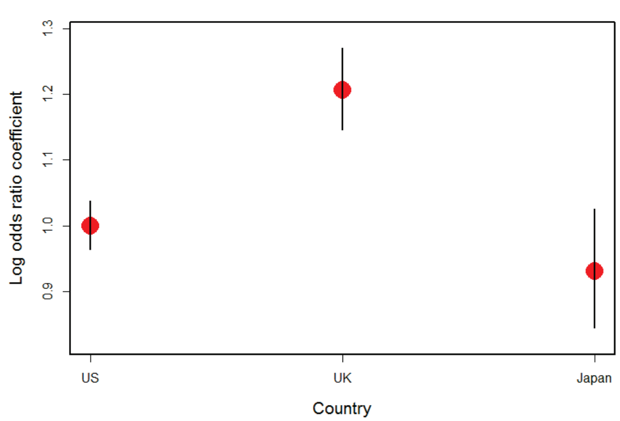
`logmult::unidiff()` uses gnm() for fitting, but makes summaries & plotting easier

```
> library(logmult)
> (yamaUni <- unidiff(as.table(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
  LoNM   0.4644   0.5228  -0.2547  -0.3856  -0.5121
  UpM    0.0214  -0.0268   0.2557  -0.0972  -0.5828
  LoM   -0.2056  -0.1028   0.0891   0.2632  -0.6504
  Farm  -0.5320  -0.3026   0.0101   0.2592   2.074
```


Visualizing associations

Plotting the unidiff object plots the layer association coefficients

```
plot(yamaUni, cex=3, col="red", pch=16)
```



Father – Son association is ordered UK > US > Japan

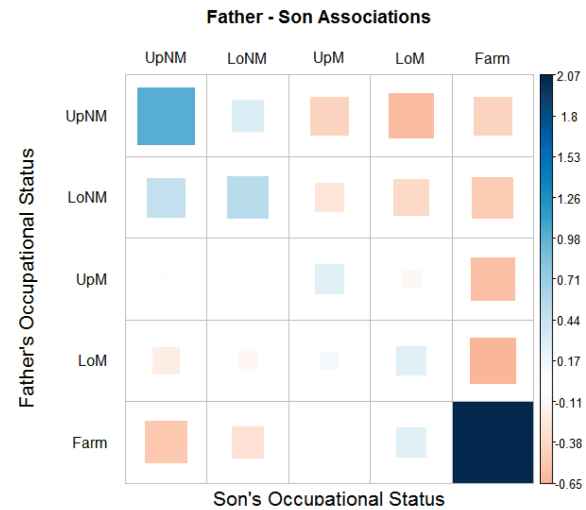
Visualizing associations

The common association parameters, δ_{ij}^{RC} are contained in the unidiff object.
Can extract these and plot in various ways

```
> inter <- yamaUni$unidiff$interaction
> names(inter)
[1] "Estimate"   "Std. Error"

> 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
```


Visualizing associations

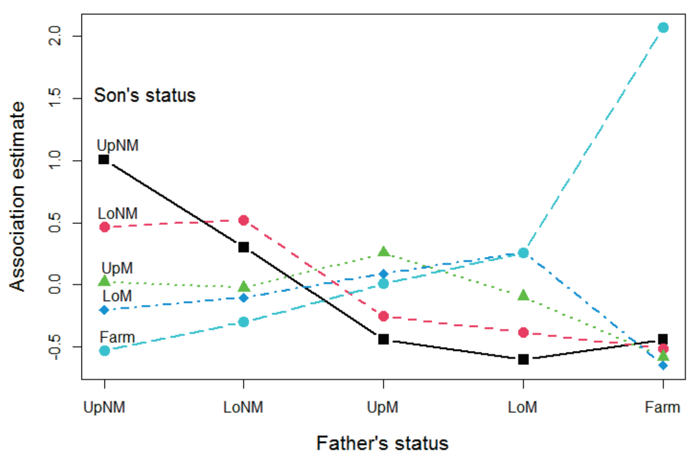
Plot these as shaded squares using corrplot()



Visualizing associations

Plot these as a line plot using matplot()

```
matplot(t(inter.mat), type="b", pch=15:19, cex=1.5, xaxt="n"
        xlab="Father's status", ylab="Association estimate" )
```



Summary

- Loglinear models, as originally formulated, were quite general, but treated all table variables as unordered factors
  - The GLM perspective is more general, allowing quantitative predictors and handling ordinal factors
  - The logit model give a simplified approach when one variable is a response
- Models for ordered factors give more powerful & focused tests
  - L × L, R, C and R+C models assign scores to the factors
  - RC(1) and RC(2) models estimate the scores from the data
- Models for square tables allow testing structured questions
  - Quasi-independence: ignoring diagonals
  - symmetry & quasi-symmetry
  - theory-specific "topological" models
- These methods can be readily combined to analyze complex tables

Looking ahead

- We can go further by considering count data models, g(y) = log(y) = X β + ε, where ε ~ Poisson()
  - Allows for interactions, nonlinearity,
  - Diagnostics for outliers, leverage, influence
- But perhaps poisson assumption, mean = var() fails
  - Quasi-poisson: Estimate overdispersion parameter
  - Neg. binomial model, ε ~ Nbin() is more general
- Excess zeros: sometimes reason for lots of zeros
  - Zero-inflated models
  - Hurdle models