Notes from: Loglinear models for contingency tables

Chapter 8 Agresti (2007)
Saúl Sotomayor Leytón
February 2012

Setup

```
> library(MASS)
> library(survival)
> library(glmmML)
> library(repeated)
> library(vcd)
> options(width=70)
```

1 Introduction

This chapter deals with data that are related in some way, for example data that were taken for the same subjects in two different occasions, or a sample of subjects that are evaluated by two different observers, or two observations (that can be questions) that are given by the same subjects.
Because of this correlation, the construction of the table (and the data frame in R) is different; for the first example in Agresti (2007) about paying higher taxes or cut living standards the construction of the table, because the questions were asked to the same people, is the one on the left, however if it was asked to different people it would be like the one on the right.

Pay Higher taxes	Cut living standards	
	Yes	No
Yes		
No		

	Yes	No
Pay higher taxes		
Cut living standards		

Table 1: Two ways of constructing a table

2 Comparing proportions

Because the data is related in some way, we expect to see some sort of relation in the cells that compose the table. In the example of the two measures that could be taken to help the environment, we expect that most people who responded "yes" to the first question would also respond "yes" to the other, and the same thing for those who responded "no".
Now, when we compare proportions we are comparing whether the first observation was equal, or not, to the second observation. In this particular case, whether people are equally in favor of an increase in taxes as for a cut in living standards in order to help the environment. Note that when the probabilities of a "yes" response for both questions are identical, also are identical the probabilities of a "no" response:

$$\pi_{1+} - \pi_{+1} = (\pi_{11} + \pi_{12}) - (\pi_{11} + \pi_{21}) = \pi_{12} - \pi_{21} \tag{1}$$

2.1 McNemar Test

This is a chi-squared test for the null hypothesis that the marginal probabilities are equal for row and column, $H_0 : \pi_{+1} = \pi_{+1}$, which is equivalently to, $H_0 : \pi_{12} = \pi_{21}$. Regarding the last expression, let $n^* = n_{12} + n_{21}$, under the null hypothesis these n^* observations has a $1/2$ chance of contributing to n_{12} and $1/2$ change to contribute to n_{21}.
When $n^* > 10$, this binomial distribution has a shape similar to the normal distribution with the same mean $1/2n^*$ and a standard deviation of $\sqrt{n^*(1/2)(1/2)}$. Then the standardized normal test equals:

$$z = \frac{n_{12} - n^*(1/2)}{\sqrt{n^*(1/2)(1/2)}} = \frac{n_{12} - n_{21}}{\sqrt{n_{12} + n_{21}}} \tag{2}$$

An associated p-value below the cut-point of 0.05 provides evidence for a difference of proportions.

Beside performing a significant test we can also construct a confidence interval (CI) that it's much more informative. Let $p_{ij} = n_{ij}/n$ be the cell proportions, then the difference, $p_{1+} - p_{+1}$ between the sample marginal proportions is the estimate for the true difference, $\pi_{1+} - \pi_{+1}$. For this difference the sample variance σ^2 and the standard error equals

$$\begin{aligned} \sigma^2 &= [p_{1+}(1 - p_{1+}) + p_{+1}(1 - p_{+1}) - 2(p_{11}p_{22} - p_{12}p_{21})] \\ SE &= \sqrt{(n_{12} + n_{21}) - (n_{12} - n_{21})^2/n}/n \end{aligned}$$

With these values we can, then, construct the CI, as usual:

$$(p_{1+} - p_{+1}) \pm Z_{\alpha/2} SE \tag{3}$$

2.1.1 Calculation in R

To calculate the McNemar test in R we need the data in form of a matrix and then just use the `mcnemar.test` function without the continuity correction (Agresti 2007, see).

```
> # UK prime minister approval at two diff times, Thompson-2012 p177
> table.10.1 <- matrix(c(794, 150, 86, 570), byrow = T, ncol = 2)
> mcnemar.test(table.10.1, correct = FALSE)
```

```
        McNemar's Chi-squared test

data:  table.10.1
McNemar's chi-squared = 17.356, df = 1, p-value = 3.099e-05
```

Computing the CI isn't as straightforward, first we need to calculate the table of proportions, then we calculate the marginal proportions for the "yes" responses.

```
> table.10.1.prop <- prop.table(table.10.1)
> prop.diff <- margin.table(table.10.1.prop, 2)[1] - margin.table(table.10.1.prop,
+     1)[1]
```

In the last command, the numbers `1` and `2` specifies whether we want, respectively, rows or columns' marginals, while the index (`[1]`) indicates that we only want the marginals for column and row 1.

As it's noted in the formula for the SE, we only need the cells of the off-diagonal, (*i.e.* n_{12}, n_{21}), therefore it's convenient to store both on a variable.

```
> off.diag <- diag(table.10.1.prop[1:2, 2:1])
```

The indexing values indicate that we want the off-diagonal elements, otherwise, the `diag` function will return the main diagonal.
With those values we can calculate the CI interval:

```
> prop.diff + c(-1, 1) * qnorm(0.975) * sqrt((sum(off.diag) - diff(off.diag)^2)/sum(table.10.1))
```

```
[1] -0.05871612 -0.02128388
```

Note that `qnorm(0.975)` $= Z_{\alpha/2} = 1.96$. I prefer the former expression because it's more meaningful. Also note that the functions `sum` and `diff` compute the sum and difference, respectively.

3 Logistic regression models

3.1 Marginal models for marginal proportions

Section 8.2.1 in Agresti (2007) relates the analysis of marginal proportions to logistic models in two ways, first by using a *identity link*, and second, by using a *logit link* function. In the first case we estimate the difference of proportions by comparing the models:

$$P(Y_1 = 1) = \alpha + \delta, P(Y_2 = 1) = \alpha \tag{4}$$

Where $\delta = P(Y_1 = 1) - P(Y_2 = 1)$. The null hypothesis of equal marginal probabilities is evaluated with the McNemar test: $H_0 : \delta = 0$.
The second case evaluates the odds ratio of marginal distributions through the comparison of the following models:

$$logitP[Y_1 = 1] = \alpha + \beta; logitP[Y_2 = 1] = \alpha \tag{5}$$

Which can be expressed as:

$$P[Y_t = 1] = \alpha + \beta x_t \tag{6}$$

where $x_t = 1$ for t=1 and 0 otherwise.

3.2 Conditional logistic regression for matched pairs

The following section discusses another way of expressing matched data which decomposes all the n observation in n partial tables like table 2, namely, one table for each subject (there the name, *subject specific table*).

Note the differences with the *population-averaged table* showed in section 8.1 of Agresti (2007).

	Response	
Observations	Yes	No
Pay Higher taxes	1	0
Cut living standards	1	0

Table 2: Example of a subject-specific table

Conditional logistic regression compares the models:

$$logitP[Y_{i1} = 1] = \alpha_i + \beta; logitP[Y_{i2} = 1] = \alpha_i \tag{7}$$

where i represents the subject and t=*1 or 2* the observation. As Thompson (2007) notes, "this means that the association within a pair is described completely by α_i, and pairs are only associated via the common effect, β. The sign and magnitude of α_i relative to β will determine the strength of association between the two responses. Also these models assume that the odds of "success" for observation 1 is $exp(\beta)$ times the odds for observation 2.

Agresti (2007) notes that even though this model focuses on parameter β, the occurrence of many α_i parameters difficults the fitting of the model. As a way to deal with this problem, he proposes *conditional logistic regression* that eliminates (conditions) the subject's specific effect and make inferences about β. Through this approach the odds ratio equal $exp(\beta) = n_{12}/n_{21}$; for example for table 8.1 the odds ratio would be $132/107 = 1.23$. Thus, assuming that the model holds, the odds for a "yes"response for the question for raising taxes is 23% higher than a "yes" answer for cutting living standards.
Agresti (2007) notes that the difference of this value from a previous one (of 1.11) reflects the differences that usually exists between marginal and conditional odds ratio. He also notes that just like the McNemar test, the cells n_{12} and n_{21}, provide all the necessary information.

Section 8.2.4 shows how to use logistic regression for case-control pairs. In these cases the variable of response is fixed by design, matching one "positive" case with a "control". As it was explained in previous chapters, by doing this we invert the relationship: Y given X to X given Y, however, because we are working with odds ratios the value will be the same.

Finally, section 8.2.5 relates the McNemar test to the Cochran-Mantel-Haenzel (CMH) test, being the former a especial case of the later. More over, it's mentioned that with the CMH test we can expand the analysis of marginal homogeneity from 2x2xn data sets to Tx2xn (where T represents the observations or questions) through a test called *Cochran's Q*, or even a TxIxn data sets (where I represents different categories for the observations).

3.2.1 Conditional logistic models in R

To fit conditional logistic models in R we need to work with ungrouped data. As illustrated in Thompson (2007) for the myocardial infarction example, we need to construct a data frame with 3 columns, one indicating the pairing (the subject-specific table), and the rest , indicating the corresponding combination of factors, in this case `Diabetes` and `Myocardial Infarction`.

```
> table.10.3 <- data.frame(pair = rep(1:144, rep(2, 144)), MI = rep(c(0,
+     1), 144), diabetes = c(rep(c(1, 1), 9), rep(c(1, 0), 16),
+     rep(c(0, 1), 37), rep(c(0, 0), 82)))
```

Note that the command, `rep(1:144,rep(2,144))` means that the vector of numbers, from 1 to 144, will be repeated two times each element. I think it would be simpler to enter `rep(1:144,each=2)`. Also note that the `diabetes` factor is a concatenation of the numbers of subject specific tables (see tables 8.3 and 8.4 in Agresti 2007).

Once we have the data frame, Thompson (2007) describes 4 ways to fit the conditional logistic model. All these forms are similar in that a formula needs to be provided, plus a specification of the pairing vector.

Option 1

```
> ## library(survival)
> fit.CLR <- clogit(MI ~ diabetes + strata(pair), method = "exact",
+     data = table.10.3)
> summary(fit.CLR)
```

```
Call:
coxph(formula = Surv(rep(1, 288L), MI) ~ diabetes + strata(pair),
    data = table.10.3, method = "exact")

  n= 288, number of events= 144

           coef exp(coef) se(coef)     z Pr(>|z|)
diabetes 0.8383    2.3125   0.2992 2.802  0.00508 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

         exp(coef) exp(-coef) lower .95 upper .95
diabetes     2.313     0.4324     1.286     4.157

Concordance= 0.573  (se = 0.035 )
Likelihood ratio test= 8.55  on 1 df,   p=0.003
Wald test            = 7.85  on 1 df,   p=0.005
Score (logrank) test = 8.32  on 1 df,   p=0.004
```

The other 3 forms use a generalized linear mixed model and the estimate of β is a bid different from the previous method.

Option 2

```
> ## library(MASS)
> fit.glmmPQL <- glmmPQL(MI ~ diabetes, random = ~1 | pair, family = binomial,
+     data = table.10.3)
> summary(fit.glmmPQL)
```

```
Linear mixed-effects model fit by maximum likelihood
  Data: table.10.3
  AIC BIC logLik
   NA  NA     NA
```

```
Random effects:
 Formula: ~1 | pair
         (Intercept) Residual
StdDev: 5.169457e-05        1

Variance function:
 Structure: fixed weights
 Formula: ~invwt
Fixed effects:  MI ~ diabetes
                 Value Std.Error  DF   t-value p-value
(Intercept) -0.1941560 0.1368852 143 -1.418385  0.1583
diabetes     0.8039216 0.2844441 143  2.826290  0.0054
 Correlation:
         (Intr)
diabetes -0.481

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-1.35646609 -0.90748519 -0.08513768  1.10194631  1.10194631

Number of Observations: 288
Number of Groups: 144
```

The argument `1|pair` is used to specify a random intercept as it's characteristic of the GLMMs.

Option 3

```
> ## library(glmmML)
> ## Data on myocardial infection for 144 positive patients and their controls
> table.10.3 <- data.frame(pair = rep(1:144, rep(2, 144)), MI = rep(c(0, 1), 144),
+                          diabetes = c(rep(c(1, 1), 9), rep(c(1, 0), 16), rep(c(0, 1), 37),
+                                       rep(c(0, 0), 82)))
> glmmML(MI ~ diabetes, cluster = table.10.3$pair, family = binomial,
+     data = table.10.3)
```

```
Call:  glmmML(formula = MI ~ diabetes, family = binomial, data = table.10.3,      cluster = table.


               coef se(coef)      z Pr(>|z|)
(Intercept) -0.1942   0.1364 -1.423  0.15500
diabetes     0.8039   0.2835  2.836  0.00457

Scale parameter in mixing distribution:  7.534e-07 gaussian
Std. Error:                              0.1214

        LR p-value for H_0: sigma = 0:  0.5

Residual deviance: 390.9 on 285 degrees of freedom         AIC: 396.9
```

Note that the `cluster` is used to specify the pairing variable and that we need to specify it as part of the data frame, *i.e.* `table.10.3$pair`.

Option 4

```
> ## library(repeated)
> fit.glmm <- glmm(MI ~ diabetes, family = binomial, data = table.10.3,
+     nest = pair)
> summary(fit.glmm)
```

```
Call:
glmm(MI ~ diabetes, family = binomial, data = table.10.3, nest = pair)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.942e-01  1.365e-01  -1.423   0.1548
diabetes     8.039e-01  2.837e-01   2.834   0.0046 **
sd           6.829e-07  1.196e-01   0.000   1.0000
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 399.25  on 287  degrees of freedom
Residual deviance: 390.80  on 285  degrees of freedom
AIC: 396.8

Number of Fisher Scoring iterations: 4
```

The `nest` argument is used to specify the pairing vector.

4 Multicategory models

This section expands the logistic model for more than two categories. The test of marginal homogeneity is:

$$P(Y_1 = i) = P(Y_2 = i) for i = 1, \ldots, I \tag{8}$$

Agresti (2007) describes two types of data that can be used in this model, first, for categories that are unordered or *nominal*, for example coffee brands; and second, for categories that have a particular order. In the former, we test the marginal homogeneity hypothesis versus *any* departure of that, while in the later we compare the null hypothesis versus a particular departure from that. Agresti (2007) notes that the former case is less powerful, because the many possibilities of departure from homogeneity, reflected in the degrees of freedom, df=I-1, than the later, which has 1 degree of freedom.

4.1 Nominal multicategory models

4.1.1 Nominal models in R

For nominal responses, Thompson (2007, p.183–185) describes a way to calculate the deviance value under the null hypothesis. For this we need a vector of counts and a matrix of dummy values that

agrees with the null hypothesis (see Agresti 2007, p. 184). At first sight this matrix this matrix seems complicated but once we analyze it, we realize that it isn't; first we need to keep in mind that, of the I categories, I-1 are non redundant. Another thing is that the matrix has rows equal to the number of cells in the table (in this case 16). Finally, the number of columns equal $(I-1)^2+1+(I-1)$. They represent the first $I-1$ cells for rows 1 to I-1 (m_{ij}), plus the last cell (m_{II}), and finally, plus the I-1 row marginals (which, under the null hypothesis are equal to the column marginals).

	m11	m12	m13	m21	m22	m23	m31	m32	m33	m44	m1+	m2+	m3+
m11	1	0	0	0	0	0	0	0	0	0	0	0	0
m12	0	1	0	0	0	0	0	0	0	0	0	0	0
m13	0	0	1	0	0	0	0	0	0	0	0	0	0
m14	-1	-1	-1	0	0	0	0	0	0	0	1	0	0
m21	0	0	0	1	0	0	0	0	0	0	0	0	0
m22	0	0	0	0	1	0	0	0	0	0	0	0	0
m23	0	0	0	0	0	1	0	0	0	0	0	0	0
m24	0	0	0	-1	-1	-1	0	0	0	0	0	1	0
m31	0	0	0	0	0	0	1	0	0	0	0	0	0
m32	0	0	0	0	0	0	0	1	0	0	0	0	0
m33	0	0	0	0	0	0	0	0	1	0	0	0	0
m34	0	0	0	0	0	0	-1	-1	-1	0	0	0	0
m41	-1	0	0	-1	0	0	-1	0	0	0	1	0	0
m42	0	-1	0	0	-1	0	0	-1	0	0	0	1	0
m43	0	0	-1	0	0	-1	0	0	-1	0	0	0	1
m44	1	0	0	0	0	0	0	0	0	1	0	0	0

Now let's explain the values. Assume that we need to match the names of the rows with those from the columns, for those values that appear in both, column and row, the value of the matching cell should be one, and zero other wise. This is true for, $m_{11}, m_{12}, m_{13}, m_{21}, m_{22}, m_{23}, m_{31}, m_{32}, m_{33}$ and m_{44}. For the other cases we can infer the value from those of the others, for example, for m_{14} we see that the first three columns have the value `-1` and the cell corresponding to the column m_{1+} has the value `1`. This notation is equivalent to m_{14} as the following equation shows,

$$m_{14} = m_{1+} - m_{11} - m_{12} - m_{13} \tag{9}$$

The same thing applies to m_{24} and m_{34}.

With the explanation of the fitted values we can show the commands that fit the model:

```
> dummies <- matrix(c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+     1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
+     0, 0, 0, 0, 0, 0, -1, -1, -1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
+     0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
+     0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
+     0, 0, 0, 0, 0, -1, -1, -1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
+     0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
+     0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
+     0, 0, 0, 0, -1, -1, -1, 0, 0, 0, 1, -1, 0, 0, -1, 0, 0, -1,
+     0, 0, 0, 1, 0, 0, 0, -1, 0, 0, -1, 0, 0, -1, 0, 0, 0, 1,
+     0, 0, 0, -1, 0, 0, -1, 0, 0, -1, 0, 0, 0, 1, 0, 0, 0, 0,
+     0, 0, 0, 0, 0, 1, 0, 0, 0), nrow = 16, ncol = ((4 - 1)^2) +
```

```
+     1 + 3, dimnames = list(c("m11", "m12", "m13", "m14", "m21",
+     "m22", "m23", "m24", "m31", "m32", "m33", "m34", "m41", "m42",
+     "m43", "m44"), c("m11", "m12", "m13", "m21", "m22", "m23",
+     "m31", "m32", "m33", "m44", "m1+", "m2+", "m3+")), byrow = TRUE)
> dummies <- data.frame(counts = c(11607, 100, 366, 124, 87, 13677,
+     515, 302, 172, 225, 17819, 270, 63, 176, 286, 10192), dummies)
> fit <- glm(counts ~ . - 1, family = poisson(identity), data = dummies)
```

The coefficients from the model are the fitted values, however, note that not all the cells are expressed, thus the missing ones have to be calculated through subtracting. The residual deviance and its associated p-value provide evidence for the null hypothesis.

4.2 Ordinal multicategory models

4.2.1 Ordinal models in R

Regarding data that has ordered categories, Agresti (2007, p. 254–255) describe the models:

$$logitP(Y_{i1} \leq J) = \alpha_{ij} + \beta; logitP(Y_{i2} \leq J) = \alpha_{ij} \tag{10}$$

which are cumulative logit models that "expresses the logit in terms of a subject effects (α_{ij}) and a margin effect (β)", the latter under the proportional odds assumption, *i.e.* the same for all equations. Agresti (2007, p. 255) describes two ways of calculating the estimate of β ($\hat{\beta}$) and its standard error, which will help, later, to test the null hypothesis that $\beta = 0$.

In Thompson (2007, p.181–182) it's described a method to calculate the deviance, Wald statistic and Pearson statistic for the $H_0 : \beta = 0$ but it uses two functions that are not easily available, namely `mph.fit` and `Marg.fct`, both pertain to J. Lang from the university of Iowa[1], the major problem would be to specify the function $\Sigma\Sigma_{i<j}(j-i)n_{ij}$.

5 Loglinear models: Symmetry and quasi-symmetry

This section introduces the concepts of *symmetry* and *quasi-symmetry*, the former is expressed as $\pi_{ij} - \pi_{ji}$ which means that all the cells above the main diagonal are mirror images of those cells below the main diagonal. When this assumption holds, marginal homogeneity must hold too.
For tables where I>2, marginal homogeneity can occur without symmetry, and this can be tested using logistic models.

Symmetry can be expressed as a logistic model of the form: $log(\pi_{ij}/\pi_{ji}) = 0$ for all i and j. The expressed cell frequencies are $\mu_{ij} = (n_{ij} + n_{ji})/2$. The residual degrees of freedom for the χ^2 test of goodness of fit equal $I(I-1)/2$. The standardized residuals for the symmetry model are: $r_{ij} = (n_{ij} - n_{ji})/\sqrt{(n_{ij} + n_{ji})}$.

Agresti (2007) states that the symmetry model is so simple that the model rarely fits well. As an alternative one can fit a quasi-symmetry model:

$$log(\pi_{ij}/\pi_{ji}) = \beta_i - \beta_j \tag{11}$$

where the symmetry model is a special case.

[1]Note.- I think it isn't difficult to input equation 8.7 from Agresti (2007) as a function in R

5.1 Models for nominal categories

5.1.1 Symmetry and quasi-symmetry models in R

Although Agresti (2007, section 8.4.2) describes the fitting of quasi-symmetry models using logistic regression, Thompson (2007, p.186–187) fits both models (symmetry and quasi-symmetry) through loglinear models.

Using the data on migration among 4 regions in the US, Thompson (2007) builds a data frame with 3 columns, 1 for each factor and a column of counts. Then she creates a vector of characters (factors) that represent the cell of the main diagonal and those above them. Remember that under the null hypothesis these cells are mirror images of those below the main diagonal.

```
> residence80 <- residence85 <- (c("NE", "MW", "S", "W"))
> table.10.6 <- expand.grid(res80 = residence80, res85 = residence85)
> table.10.6$counts <- c(11607, 100, 366, 124, 87, 13677, 515, 302, 172,
+                        225, 17819, 270,  63, 176, 286, 10192)
> table.10.6$symm <- paste(pmin(as.numeric(table.10.6$res80),
+                               as.numeric(table.10.6$res85)),
+                          pmax(as.numeric(table.10.6$res80),
+                               as.numeric(table.10.6$res85)), sep = ",")
```

The function `pmin` selects among the two factors, treated as numbers, the lower one, while the function `pmax` selects among the same factors the higher one. The result, as noted before, is a vector of characters that represent the cell of the main diagonal and the above diagonal.
Next, as usual, Thompson (2007) reverses the order of the factor `symm` to set the last category to zero, and fits the loglinear model.

```
> table.10.6$symm <- factor(table.10.6$symm, levels = rev(unique(table.10.6$symm)))
> fit.symm <- glm(counts ~ symm, family = poisson(log), data = table.10.6)
```

Later, for the quasi-symmetry model, Thompson (2007) adds another variable that "differentiates the main effect for the rows and columns", this is just `residence80` factor with its levels reversed, this to match those for the `symm`

```
> table.10.6$res80a <- factor(table.10.6$res80, levels = rev(residence80))
> fit.qsymm <- glm(counts ~ symm + res80a, family = poisson(log),
+     data = table.10.6)
```

As for any loglinear model, the interaction term λ_{ij}^{XY} represents the log odds ratio for the "yes" category of X, relative to Y.

Because marginal homogeneity is a special case of quasi-symmetry models, we can test if that is true with a chi-square test comparing these models. Remember that a high deviance difference, with its corresponding p-value, provides evidence against the marginal homogeneity.

```
> anova(fit.symm, fit.qsymm, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: counts ~ symm
Model 2: counts ~ symm + res80a
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1         6    243.550
2         3      2.986  3   240.56 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

5.2 Models for ordered categories

The symmetry model for ordered categories is:

$$log(\pi_{ij}/\pi_{ji}) = \beta(\mu_j - \mu_i) \tag{12}$$

where μ_i corresponds to the row and column scores.

In this case, the symmetry model corresponds to $\beta = 0$. The greater the absolute value of β, the greater the difference among π_{ij} and π_{ji}. With categories $\mu_i = i$ the probability that the first observation is x categories higher than the second observation is $exp(\beta)$. Based on this I think there is an error in Thompson (2007, p. 190).

For the quasi-symmetry models for ordered categories, Thompson (2007) uses the same approach as for nominal categories but adding one column that represents the scores.

```
> table.10.5 <- data.frame(expand.grid(PreSex = factor(1:4),
+                                      ExSex = factor(1:4)),
+                          counts = c(144, 33,  84,  126,  2,  4 , 14 , 29 ,
+                                     0 , 2 , 6 , 25,  0,  0,  1,  5))
> table.10.5$symm <- paste(pmin(as.numeric(table.10.5$PreSex),
+     as.numeric(table.10.5$ExSex)), pmax(as.numeric(table.10.5$PreSex),
+     as.numeric(table.10.5$ExSex)), sep = ",")
> table.10.5$scores <- rep(1:4, each = 4)
> fit.oqsymm <- glm(counts ~ symm + scores, data = table.10.5,
+     family = poisson(link = log))
```

6 Measuring agreement between subjects

As its name suggest, this section focuses on separate rating of a sample by two observers using the same categorical or ordinal scale. It centers its attention to the cell in the main diagonal, which represent the agreement between observers. As Agresti (2007) points out, *agreement* is different form *association*, the former requires association, while the latter can exist without agreement.

One way to see if agreement occurs is to calculate the standardized residuals from a model (loglinear) of independence. Cells with positive standardized residuals have higher frequencies than what would be expected under independence, and, if the largest residuals appear in the main diagonal it's a sign that agreement may be present.

6.0.1 Measuring agreement in R

For the example of diagnosis of carcinoma, Thompson (2007) calculates the standardized residuals as follows:

```
> load('supp_data/pathologist.dat.rda')
> pathologist <- pathologist.dat
> names(pathologist)<- c("y", "B", "A")
> pathologist$y[pathologist$A==4 & (pathologist$B==3 | pathologist$B==4)] <- c(17, 10)
> pathologist$A <- factor(pathologist$A, levels=5:1)
> pathologist$B<-factor(pathologist$B, levels=5:1)
> fit.ind <- glm(y ~ A + B, data = pathologist, family = poisson(log),
+     subset = (A != 5) & (B != 5))
> fit.ind.res <- resid(fit.ind, type = "pearson")/
+     sqrt(1 - lm.influence(fit.ind)$hat)
```

Because there seem to be some degree of agreement, we can fit a model that takes into account the degree of agreement. This type of model is the *quasi-independence* model:

$$log\mu_{ij} = \lambda + \lambda_i^X + \lambda_j^Y + \sigma_i I(i = j) \tag{13}$$

The loglinear model above shows that a parameter $\sigma_i I$ is added to the independence model. Note the indicator variable I, this variable equals 1 when i=j. When $\sigma_i > 0$ more agreements regarding outcome i occur than would be expected under independence.

Thompson (2007) adds these main diagonal elements with the following commands:

```
> pathologist$D1 <- as.numeric((pathologist$A == 1) & (pathologist$B ==
+     1))
> pathologist$D2 <- as.numeric((pathologist$A == 2) & (pathologist$B ==
+     2))
> pathologist$D3 <- as.numeric((pathologist$A == 3) & (pathologist$B ==
+     3))
> pathologist$D4 <- as.numeric((pathologist$A == 4) & (pathologist$B ==
+     4))
```

Then she fits the model:

```
> fit.qi <- glm(y ~ A + B + D1 + D2 + D3 + D4, family = poisson(log),
+     data = pathologist, subset=(A!=5) & (B!=5))
```

The coefficients for the D_i parameters are used to calculate the odds ratio of agreement.

$$\begin{aligned} \tau_{ab} = \quad & \frac{\pi_{aa}\pi_{bb}}{\pi_{ab}\pi_{ba}} \quad && = \frac{\mu_{aa}\mu_{bb}}{\mu_{ab}\mu_{ba}} \\ = \quad & exp(\delta_a + \delta_b) && = exp(D_a + D_b) \end{aligned}$$

For example, Agresti (2007) calculates that the odds that one observer's rating is category 2 rather than 3 is $\hat{\tau}_{23}$=12.3 times as high when the other observer's rating is also 2 than when it's 3.

Though the model fits better than the independence model it still lacks some fit; this is because the quasi-independence model assumes independence for all the off-diagonal elements, something that not always occur. An alternative is to fit a quasi-symmetry or ordinal quasi-symmetry model. The former would account for any association among the off-diagonal elements.

As shown before, to fit a quasi-symmetry model, we need to create an extra vector/column that would represent the diagonal and off-diagonal elements.

```
> pathologist2 <- pathologist[(pathologist$A!=5) & (pathologist$B!=5), ]
> pathologist2$A <- pathologist2$A[drop = TRUE]
> pathologist2$B <- pathologist2$B[drop = TRUE]
> pathologist2$symm <- paste(pmin(as.numeric(pathologist2$A),
+  as.numeric(pathologist2$B)), pmax(as.numeric(pathologist2$A),
+  as.numeric(pathologist2$B)), sep = ",")
> pathologist2$symm <- factor(pathologist2$symm,
+                             levels = (unique(pathologist2$symm)))
```

From what it's said on Thompson (2007, p.194) it seems that "because of the two symmetric cell with a count of 0 that appear in table 8.6 (Agresti 2007), the maximum likelihood fitted values must also be 0". Now it's interesting that in order to do that (*i.e.* condition those cells to be zero), Thompson (2007) sets these two cell (m_{14} and m_{41}) to zero on the fitted values of the quasi-independent model instead of the observed values/counts. In other words she uses the fitted values of the quasi-independent model as a starting point instead of the observed values.

```
> starter <- fitted(fit.qi)
> starter[c(4,13)] <- 0 # the fixed zeroes
```

The index `c(4,13)` represents the mentioned cell counted row-wise. The variable `starter` is then used in the fitting process.

```
> fit.qsymm <- glm(y~symm+A, data=pathologist2,
+                  family = poisson(log), control=glm.control(maxit=100),
+                  etastart = starter)
```

Another curious thing is that because we have set some cells to zero we need to modify the degrees of freedom from 3 to 2. The odds ratio of agreement are calculated with the coefficients of the `symms` levels. For example the odds of one observer's classification being's 2 instead of 3 when the other observer's classification is 2 rather than 3 is.

```
>  coefs <- fit.qsymm$coefficients
> as.numeric(exp(coefs["symm2,2"] + coefs["symm3,3"]- 2*coefs["symm2,3"]))
```

```
[1] 10.72846
```

6.1 Kappa measure of agreement

Finally, Agresti (2007) describes another way to evaluate agreement but unlike what it's been showed, it isn't based on model but a single value that it's denominated *kappa measure of agreement* or *Cohen's kappa*. It is defined by:

$$\kappa = \frac{\Sigma\pi_{ii} - \Sigma\pi_{i+}\pi_{+j}}{1 - \Sigma\pi_{i+}\pi_{+j}} \tag{14}$$

6.1.1 Cohen's kappa in R

In R it's easy to calculate this index, as well as, a variant of it that is used with ordered categories, and therefore, weights the differences among categories (*weighted kappa*). This is done with the package `vcd`. Remember that it works with tables instead of data frames.

```
> ## library(vcd)
> path.tab <- xtabs(y ~ A + B, data = pathologist2)
> Kappa(path.tab, weights = "Fleiss-Cohen")
```

```
            value     ASE      z  Pr(>|z|)
Unweighted 0.4930 0.05674  8.688 3.677e-18
Weighted   0.7838 0.03867 20.269 2.399e-91
```

7 Preferences between pairs of outcome categories

The last section of the chapter shows another form of paired data, one that compairs a set of elements in order to establish some sort of ranking of preferences. As it's showed on table 8.9 the same elements are placed in rows and columns and the resulting cells represent the preference for the row over the column. As one may deduce the main diagonal is of no use.

7.1 Bradley–Terry model

For this type of table, the Bradley-Terry model uses a logistic model to calculate the probability that row i is preferred (in the broader sense of the word) over column j. It has the form:

$$logit(\pi_{ij}) = log(\pi_{ij}/\pi_{jj}) = \beta_i - \beta_j \tag{15}$$

Agresti (2007) notes that this model is similar to the quasi-symmetry model and in fact it can be fitted in a similar way. Thus the probability of preference is:

$$\pi_{ij} = exp(\hat{\beta}_i - \hat{\beta}_j)/1 - exp(\hat{\beta}_i - \hat{\beta}_j) \tag{16}$$

7.1.1 Bradley-Terry model in R

Thompson (2007, p. 196–198) describes two ways of fitting the Bradley-Terry model, the first through the logistic model described above and the other through the equivalent loglinear model. The latter also explains the from of the logistic model:

$$\begin{aligned} log\frac{\mu_{ab}}{\mu_{ba}} &= (\lambda + \lambda_a^X + \lambda_b^Y + \lambda_{ab}) - (\lambda + \lambda_b^X + \lambda_a^Y + \lambda_{ab}) \\ &= (\lambda_a^X - \lambda_a^Y) - (\lambda_b^X - \lambda_b^Y) = \beta_a - \beta_b \end{aligned}$$

7.1.2 Logistic approximation to the Bradley-Terry model in R

Thompson (2007) illustrates the calculation in R with the baseball example (showed in Agresti 2002). In order to use the logistic approximation, the author creates a data frame of positive and negative responses that correspond to the off-diagonal cells n_{ij}, n_{ji}, where $i \neq j$. Then she creates for each team a column with codes, representing victories (1), looses (-1) or not played/not applicable (0).

```
          Milwaukee Detroit Toronto NY Boston Cleveland Baltimore
 [1,] 6  7        -1       1       0  0      0         0         0
 [2,] 4  9        -1       0       1  0      0         0         0
 [3,] 6  7        -1       0       0  1      0         0         0
 [4,] 6  7        -1       0       0  0      1         0         0
 [5,] 4  9        -1       0       0  0      0         1         0
 [6,] 2 11        -1       0       0  0      0         0         1
 [7,] 6  7         0      -1       1  0      0         0         0
 [8,] 8  5         0      -1       0  1      0         0         0
 [9,] 2 11         0      -1       0  0      1         0         0
[10,] 4  9         0      -1       0  0      0         1         0
[11,] 4  9         0      -1       0  0      0         0         1
[12,] 6  7         0       0      -1  1      0         0         0
[13,] 6  7         0       0      -1  0      1         0         0
[14,] 5  8         0       0      -1  0      0         1         0
[15,] 1 12         0       0      -1  0      0         0         1
[16,] 7  6         0       0       0 -1      1         0         0
[17,] 6  7         0       0       0 -1      0         1         0
[18,] 3 10         0       0       0 -1      0         0         1
[19,] 6  7         0       0       0  0     -1         1         0
[20,] 1 12         0       0       0  0     -1         0         1
[21,] 7  6         0       0       0  0      0        -1         0
```

This matrix corresponds to the following table:

	Milwaukee	Detroit	Toronto	New York	Boston	Cleveland	Baltimore
Milwaukee	0	7	9	7	7	9	11
Detroit	6	0	7	5	11	9	9
Toronto	4	6	0	7	7	8	12
New York	6	8	6	0	6	7	10
Boston	6	2	6	7	0	7	12
Cleveland	4	4	5	6	6	0	6
Baltimore	2	4	1	3	1	7	0

Table 3: Results of 1987 Season for American League Baseball Teams

Note that because the positive outcomes are based on the cell below the main diagonal, the `Milwaukee` team has only looses, 6 in total, corresponding to column 1 in the table, the rest of values are all 0s because the team no longer appears int he pairs. The next team, `Detroit` has 1 win corresponding to the first cell, then follows five 0s, then five -1, the latter correspond to column number 2 in the table, and the rest are all 0s. For `Boston`, the first three cell are 0s, then one 1, four 0s, 1, three 0s, 1, three 0s, 1, two 0s, 1, two 0s, two -1, and finally 0. This is:

```
Boston = 0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,1,0,0,-1,-1,0
```

Then the author creates a data frame of positive and negative answers.

```
> response <- cbind(c(6, 4, 6, 6, 4, 2, 6, 8, 2, 4, 4, 6, 6, 5, 1, 7, 6, 3, 6, 1, 7),
+  13 - c(6, 4, 6, 6, 4, 2, 6, 8, 2, 4, 4, 6, 6, 5, 1, 7, 6, 3, 6, 1, 7))
```

Note that all the teams played 13 matches among them, therefore, the negative outcomes are just 13 minus the positive results.
Now we can fit the model.

```
> fit.BT <- glm(response ~ -1 + Milwaukee + Detroit + Toronto + NY + Boston +
+                   Cleveland, family = binomial)
> summary(fit.BT, cor=FALSE)
```

```
Call:
glm(formula = response ~ -1 + Milwaukee + Detroit + Toronto +
    NY + Boston + Cleveland, family = binomial)

Coefficients:
          Estimate Std. Error z value Pr(>|z|)
Milwaukee   1.5814     0.3433   4.607 4.09e-06 ***
Detroit     1.4364     0.3396   4.230 2.34e-05 ***
Toronto     1.2945     0.3367   3.845 0.000121 ***
NY          1.2476     0.3359   3.715 0.000203 ***
Boston      1.1077     0.3339   3.318 0.000908 ***
Cleveland   0.6839     0.3319   2.061 0.039345 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 49.699  on 21  degrees of freedom
Residual deviance: 15.737  on 15  degrees of freedom
AIC: 87.324

Number of Fisher Scoring iterations: 4
```

The `-1` element, in the formula, indicates that the model has no intercept. Also note that the model doesn't include the last team, `Baltimore`.
The fitted counts are calculated in steps; first the cells above (or below, the order is not important) the main diagonal and then the other half. For this task the function `lower.tri` is very useful.

```
> losing.team <- c("Milwaukee", "Detroit", "Toronto", "NY", "Boston",
+                  "Cleveland",  "Baltimore")
> win.team <- losing.team
> fitted.counts <- matrix(0, nc = 7, nr = 7, dimnames = list(win.team, win.team))
> fitted.counts[lower.tri(fitted.counts)] <-  round(13 * fitted(fit.BT), 1)
> fitted.counts[!lower.tri(fitted.counts, diag=T)] <- 13 - round(13 * fitted(fit.BT), 1)
```

In the same way we can calculate the probabilities of a positive result

```
> fitted.probs <- matrix(0, nc = 7, nr = 7, dimnames = list(win.team, win.team))
> fitted.probs[lower.tri(fitted.probs)] <- round(fitted(fit.BT), 2)
> fitted.probs[!lower.tri(fitted.probs, diag=T)] <- 1 - round(fitted(fit.BT), 2)
```

The `lower.tri` function, used as an index, returns the cell below the main diagonal, while the negation of it (`!lower.tri`) calculates the other half.

7.1.3 Loglinear approximation to the Bradley-Terry model in R

Fitting a loglinear model is somehow easier. First the data construction

```
> table.10.10 <- expand.grid(losing = factor(losing.team, levels = rev(losing.team)),
+                            winning = factor(win.team,  levels = rev(win.team)))
> table.10.10$counts <- c(0, 7, 9, 7, 7, 9, 11, 6, 0, 7, 5, 11, 9, 9, 4, 6,  0, 7,
+                         7, 8, 12, 6, 8, 6, 0, 6, 7, 10, 6, 2, 6, 7, 0, 7, 12, 4, 4, 5, 6,
+                         6, 0, 6, 2, 4, 1, 3, 1, 7, 0)
```

Note that the counts are entered row-wise for the whole table (not just the below-diagonal elements) and that the main diagonal is filled with zeroes. Then as in the previous sections a column is created to indicate the appropriate cells.

```
> table.10.10$symm <- paste(pmin(as.character(table.10.10$winning),
+                                as.character(table.10.10$losing)),
+                           pmax(as.character(table.10.10$winning),
+                                as.character(table.10.10$losing)), sep=",")
```

The above command creates a character vector of pairs of teams order alphabetically.
Then we can fit the loglinear model.

```
> fit.BTQS <- glm(counts ~ symm + winning, data = table.10.10, family = poisson(log))
> summary(fit.BTQS, cor = F)
```

```
Call:
glm(formula = counts ~ symm + winning, family = poisson(log),
    data = table.10.10)

Coefficients:
                         Estimate Std. Error z value Pr(>|z|)
(Intercept)              -19.3026  9426.6168  -0.002 0.998366
symmBaltimore,Boston      20.4744  9426.6168   0.002 0.998267
symmBaltimore,Cleveland   20.7751  9426.6168   0.002 0.998242
symmBaltimore,Detroit     20.2178  9426.6168   0.002 0.998289
symmBaltimore,Milwaukee   20.0991  9426.6168   0.002 0.998299
symmBaltimore,NY          20.3675  9426.6168   0.002 0.998276
symmBaltimore,Toronto     20.3309  9426.6168   0.002 0.998279
symmBoston,Boston         -1.1077 13331.2494   0.000 0.999934
symmBoston,Cleveland      20.2563  9426.6168   0.002 0.998285
symmBoston,Detroit        19.8889  9426.6168   0.002 0.998317
symmBoston,Milwaukee      19.8021  9426.6168   0.002 0.998324
symmBoston,NY             19.9943  9426.6168   0.002 0.998308
symmBoston,Toronto        19.9689  9426.6168   0.002 0.998310
```

```
symmCleveland,Cleveland    -0.6839 13331.2494   0.000 0.999959
symmCleveland,Detroit      20.0451  9426.6168   0.002 0.998303
symmCleveland,Milwaukee    19.9443  9426.6168   0.002 0.998312
symmCleveland,NY           20.1694  9426.6168   0.002 0.998293
symmCleveland,Toronto      20.1393  9426.6168   0.002 0.998295
symmDetroit,Detroit        -1.4364 13331.2494   0.000 0.999914
symmDetroit,Milwaukee      19.6629  9426.6168   0.002 0.998336
symmDetroit,NY             19.8279  9426.6168   0.002 0.998322
symmDetroit,Toronto        19.8064  9426.6168   0.002 0.998324
symmMilwaukee,Milwaukee    -1.5814 13331.2494   0.000 0.999905
symmMilwaukee,NY           19.7460  9426.6168   0.002 0.998329
symmMilwaukee,Toronto      19.7262  9426.6168   0.002 0.998330
symmNY,NY                  -1.2476 13331.2494   0.000 0.999925
symmNY,Toronto             19.9031  9426.6168   0.002 0.998315
symmToronto,Toronto        -1.2945 13331.2494   0.000 0.999923
winningCleveland            0.6839     0.3319   2.061 0.039345 *
winningBoston               1.1077     0.3339   3.318 0.000908 ***
winningNY                   1.2476     0.3359   3.715 0.000203 ***
winningToronto              1.2945     0.3367   3.845 0.000121 ***
winningDetroit              1.4364     0.3396   4.230 2.34e-05 ***
winningMilwaukee            1.5814     0.3433   4.607 4.09e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 133.865  on 48  degrees of freedom
Residual deviance:  15.737  on 15  degrees of freedom
AIC: 236.05

Number of Fisher Scoring iterations: 17
```

The coefficients for the `winning`*team* should be equal to those *team* coefficients from the logistic model.

Finally the fitted counts.

```
> matrix(round(fitted(fit.BTQS), 1), nr = 7, nc = 7, byrow = T,
+        dimnames = list(win.team, losing.team))
```

```
          Milwaukee Detroit Toronto  NY Boston Cleveland Baltimore
Milwaukee       0.0     7.0     7.4 7.6    8.0       9.2      10.8
Detroit         6.0     0.0     7.0 7.1    7.6       8.8      10.5
Toronto         5.6     6.0     0.0 6.7    7.1       8.4      10.2
NY              5.4     5.9     6.3 0.0    7.0       8.3      10.1
Boston          5.0     5.4     5.9 6.0    0.0       7.9       9.8
Cleveland       3.8     4.2     4.6 4.7    5.1       0.0       8.6
Baltimore       2.2     2.5     2.8 2.9    3.2       4.4       0.0
```

References

Agresti, Alan (2002). *Categorical Data Analysis*. 2nd ed. New York: Wiley-Interscience. ISBN: 978-0-471-36093-3.

— (2007). *An Introduction to Categorical Data Analysis*. 2nd ed. Hoboken NJ: Wiley-Interscience. ISBN: 978-0-471-22618-5.

Thompson, Laura A. (2007). *S-plus (and R) Manual to Accompany Agresti's "Categorical Data Analysis" (2002)*.