Contingency tables

Fisman et al. conducted a study on dating behaviour using data from a Speed Dating experiment at Columbia University. Pairs of men and women students interacted for four minutes and then each filled out a form which recorded whether or not they wanted to receive the other's email address, and also various details about themselves including their chosen subject of study, what motivated them to sign up for the speed dating experiment, how often they go out and how often then go on dates. If both individuals wanted each others email addresses, it is considered a match. The original data are available at `http://andrewgelman.com/2008/01/21/the_speeddating_1/`. Download an edited version of the data from the course webpage with

```
> file_path <- "http://www.statslab.cam.ac.uk/~rds37/teaching/statistical_modelling/"
> SD_data <- read.csv(paste0(file_path, "SD_match.csv"))
> SD_data[1:3, ]
  match subject_m   goal_m     date_m go_out_m subject_f   goal_f       date_f go_out_f
1     0      Econ      fun several/yr > 2/week       Law meet ppl almost never > 2/week
2     0      Econ      fun    1/month  2/month       Law meet ppl almost never > 2/week
3     1      Econ     date   > 2/week > 2/week       Law meet ppl almost never > 2/week
```

The first row records the meeting of a male Economics student and a female Law student, which did not result in a match (`match` is 0). The goals of the man and woman were to have fun and to meet people, respectively. We also have the frequencies with which the individuals go out and go out on dates. Let us first focus on the relationship between match and the subjects of the individuals.

```
> SD_subj <- table(SD_data[, c("match", "subject_m", "subject_f")])
```

The `table` function converts the data into contingency table format, though the output from `SD_subj` is perhaps not the easiest to interpret. In order to fit models to the data, we apply `as.data.frame` to the contingency table `SD_subj`. This produces a data frame where each each row gives the number of original observations (`Freq`) that fall into each of the possible categories given by each pair of subject types and match category.

```
> SD_subj
   match       subject_m       subject_f Freq
1      0 Arts+Humanities Arts+Humanities  305
2      1 Arts+Humanities Arts+Humanities   69
3      0            Econ Arts+Humanities  625
```

The `xtabs` function gives a handy way of visualising the data in a more helpful contingency table:

```
> xtabs(Freq ~ subject_m + subject_f + match, data=SD_subj)
```

Since the numbers of Law students are fairly low, we could considering combining them with the Economics students. One way of doing this is as follows (the `vcd` package has a dedicated function to do this, but we will not use this here). Note that typically we wouldn't modify the actual data object `SD_data` but create a copy and then modify the copy.

```
> levels(SD_subj$subject_m)
[1] "Arts+Humanities" "Econ"            "Law"             "Sciences"
> levels(SD_subj$subject_m) <- c("Arts+Humanities", "Econ+Law", "Econ+Law", "Sciences")
> levels(SD_subj$subject_f) <- c("Arts+Humanities", "Econ+Law", "Econ+Law", "Sciences")
```

Now we need to add the frequencies for Economics and Law.

```
> SD_subj <- as.data.frame(xtabs(Freq ~ subject_m + subject_f + match, data=SD_subj))
```

This then makes `SD_subj` have the desired form.
Let us fit a simple independence model using a surrogate Poisson model.

```
> mod1 <- glm(Freq ~., data=SD_subj, family=poisson)
> mod1$dev
[1] 60.74671
```

The final line gives the deviance of the model. What should we compare this to when testing at the 5% level?

It appears that the model doesn't fit too well so we should seek a more complex model. Perhaps the question of interest is whether the the joint distribution of the male and female subject choices is the same in the match or non-match group. The most complex model that allows for this joint distribution to be the same can be fitted as follows.

```
> mod2 <- glm(Freq ~ subject_m*subject_f + match, data=SD_subj, family=poisson)
> anova(mod1, mod2, test="LR")
```

The output indicates we should prefer the second model over the first, but the second model is still not a great fit as can be seen from its high deviance compared to $\chi^2_8(0.05)$.

```
> qchisq(0.05, df=8, lower.tail=FALSE)
```

This suggests then that the joint distribution of the male and female subject choices is different for the match and non-match groups. We can view how this differs from the model that assumes (`subject_m`, `subject_f`) are independent of `match` by comparing the fitted values of the latter model with the observed counts.

```
> xtabs(Freq ~ subject_m + subject_f + match, data=SD_subj)
> xtabs(predict(mod2, newdata=SD_subj, type="response")
+ ~ subject_m + subject_f + match, data=SD_subj)
```

Let us now fit a model with no 3-way interactions, but which includes `mod2` (c.f. H_4 on the final page of you notes).

```
mod3 <- glm(Freq ~ subject_m*subject_f + subject_m*match + subject_f*match,
+ data=SD_subj, family=poisson)
summary(mod3)
```

Although this does appear to fit better than `mod2`, the deviance is still slightly large. What is the p-value for the test against the saturated model? You can view the difference between the fitted values of this model and the observed counts as before. Feel free to now explore other models using `SD_data`—for example you could try to ascertain whether people who tend to go out roughly the same amount tend to match (or perhaps opposites attract?).

Gamma regression

The following section is optional. A soft drink bottler is analysing vending machine service routes in his distribution system, and is interested in predicting the amount of time required by the route driver to service the vending machines in an outlet. The industrial engineer responsible for the study has suggested that the two most important variables affecting the delivery time are the number of cases of product stocked and the distance walked by the route driver. The engineer has collected 25 observations on delivery time (in minutes), number of cases and distance walked (feet). Download the data from the course webpage with

```
> file_path <- "http://www.statslab.cam.ac.uk/~rds37/teaching/statistical_modelling/"
> (Drinks <- read.table(paste(file_path, "drinks.txt", sep =""), header = TRUE))
> attach(Drinks)
```

Note we have used `read.table` rather than `read.csv` as the data is tab delimited rather than comma separated. It could be argued that for this data, the standard deviation of the time should not be constant, but should be proportional to the number of cases and/or the distance walked. We can consider the errors to multiplicative, rather than additive. One option is to transform the responses using a logarithmic transformation, much as we did with the `hills` and `mammals` datasets studied in the example sheets. An alternative, which retains the original scale of measurement, is to observe that if

$$Y = \mu\varepsilon,$$

where, without loss of generality, $\mathbb{E}(\varepsilon) = 1$ and $\mathrm{Var}(\varepsilon) = \sigma^2$, then $\mathrm{Var}(Y) = \sigma^2\mu^2$. This suggests using a gamma model for the data, as it has variance function $V(\mu) = \mu^2$. The canonical link function for the gamma family is $g(\mu) = -1/\mu$. We can fit a gamma model with

```
> GammaMod1 <- glm(Time ~ Distance + Cases, family = Gamma)
> summary(GammaMod1)
```

The style of most of the output should be familiar by now. Let Y_i denote the i^{th} time, let x_i denote the i^{th} number of cases, and let z_i denote the distance. The model is that $Y_1, \ldots, Y_n$ are independent gamma random variables, with Y_i having shape parameter $\nu = 1/\sigma^2$ and mean μ_i, where

$$\frac{1}{\mu_i} = \beta_1 + \beta_2 x_i + \beta_3 z_i,$$

for $i = 1, \ldots, n$. Notice that R uses $1/\mu$, rather than $-1/\mu$ as the link function (though of course the only effect is to multiply the parameter estimates by -1). One new piece of information in the summary is the estimate of the dispersion parameter, which is based on the estimate

$$\tilde{\sigma}^2 = \frac{1}{n-p}\sum_{i=1}^{n}\frac{(Y_i - \hat{\mu}_i)^2}{a_i V(\hat{\mu}_i)}$$

discussed in lectures (here we have $a_i = 1$ for all i). If you compute this for `GammaMod1` you will notice that you get a slightly different estimate to that shown in the summary output. The reason is that R approximates $\tilde{\sigma}^2$ above by

$$\frac{1}{n-p}\sum_{i=1}^{n}\frac{\{\tilde{Y}_i(\hat{\mu}_{m-1})\}^2}{W_{ii}(\hat{\mu}_{m-1})}$$

where m is the final iteration in the IWLS algorithm for computing the m.l.e. Since

$$\tilde{Y}_i(\mu) = g'(\mu_i)(y_i - \mu_i)$$

and

$$W_{ii}(\mu) = \frac{1}{a_i V(\mu_i)\{g'(\mu_i)\}^2},$$

our $\tilde{\sigma}^2$ and the estimate produced by R are essentially the same.

Note that since the dispersion parameter is unknown here, the $p \times p$ block of the Fisher information matrix corresponding to β, which we have written as $i(\beta)$ in fact depends on σ^2. The standard errors in the summary output are given by the square roots of the diagonal entries of $i(\beta)^{-1}$ with the m.l.e. $\hat{\beta}$ plugged in for β and $\tilde{\sigma}^2$ substituted in for σ^2.

From the p-values based on asymptotic normality of the m.l.e. in the summary output, it appears that one of the explanatory variables could be removed from the model. Try fitting a new model with this term removed and store this in `GammaMod0`.

Is the increase in deviance significant? Recall that when testing model M_0 against M_1 , where $M_0 \subset M_1$ with parameters $p_0 < p$, respectively, we can employ the likelihood ratio statistic

$$\frac{D(y; M_0) - D(y; M_1)}{\sigma^2},$$

which approximately follows a $\chi^2_{p-p_0}$ distribution (here $D(y; M_0)$ and $D(y; M_1)$ are the deviances of models M_0 and M_1 respectively). If σ^2 is not known but is instead estimated by $\tilde{\sigma}^2$, then the following approximate result is used

$$\frac{\frac{1}{p-p_0}\{D(y; M_0) - D(y; M_1)\}}{\tilde{\sigma}^2} \sim F_{p-p_0, n-p}.$$

The deviance and dispersion are stored as components of the `glm` object and the output of `summary.glm` respectively. Therefore we can calculate the approximate p-value for the test above with

```
> Ftest <- (GammaMod0$dev - GammaMod1$dev) / summary(GammaMod1)$dispersion
> pf(Ftest, df1 = 1, df2 = 25 - 3, lower.tail = FALSE)
```

`anova` also performs this for you if you supply the `test = ''F''` option.
Are there any disadvantages to the link function that we have used? We can also fit a model with log link using.

```
> GammaMod2 <- glm(Time ~ Distance + Cases, family = Gamma(link = log))
```

How can we interpret the coefficients of this model? How does the fit compare to that of `GammaMod1`?