15

Rates and Poisson regression

Epidemiological studies often involve the calculation of rates, typically rates of death or incidence rates of a chronic or acute disease. This is based upon counts of events occurring within a certain amount of time. The Poisson regression method is often employed for the statistical analysis of such data. However, data that are not actually counts of events but rather measurements of time until an event (or nonevent) can be analyzed by a technique which is formally equivalent.

15.1 Basic ideas

The data that we wish to analyze can be in one of two forms. They can be in *aggregate* form as an observed count x based on a number of person-years T. Often the latter is an approximation based on tables of population size. There may of course be more than one group, and we may wish to formulate various models describing the rates in different groups.

We may also have individual-level data, in which for each subject we have a time under observation T_i and a $0/1$ indicator x_i of whether the subject has had an event. The aggregate data can be thought of as being $x = \sum x_i$ and $T = \sum T_i$, where the sums are over all individuals in the group.

P. Dalgaard, *Introductory Statistics with R*,
DOI: 10.1007/978-0-387-79054-1_15,

15.1.1 The Poisson distribution

The Poisson distribution can be described as the limiting case of the binomial distributions when the size parameter N increases while the expected number of successes $\lambda = Np$ is fixed. This is useful to describe rare event in large populations. The resulting distribution has point probabilities

$$f(x) = \frac{\lambda^x}{x!}e^{-\lambda} \qquad x = 0, 1, \ldots$$

The distribution is theoretically unbounded, although the probabilities for large x will be very small. In R, the Poisson distribution is available via the functions `dpois`, `ppois`, etc.

In the context of epidemiological data, the parameter of interest is usually the expected counts *per unit of observed time*; i.e., the rate at which events occur. This enables the comparison of populations that may be of different size or observed for different lengths of time. Accordingly, we may parameterize the Poisson distribution using

$$\rho = \lambda / T$$

Notice that parts of the literature use λ to denote the rate. The notation used here is chosen so as to stay compatible with the argument name in `dpois`.

The Poisson likelihood

Models for Poisson data can be fitted by the method of maximum likelihood. If we parameterize in terms of ρ, the log-likelihood becomes

$$l(\rho) = \text{constant} + x \log \rho - \rho T$$

which is maximized when $\rho = x/T$. The log-likelihood can be generalized to models involving several counts by summing terms of the same form.

15.1.2 Survival analysis with constant hazard

In this section, for convenience, we use terminology appropriate for mortality studies, although the event may be many things other than the death of the subject.

Individual-level data are essentially survival data as described in Chapter 14, except for changes in notation. One difference, though, is that in the analysis of rates it is often reasonable to assume that the hazard does not change over time, or at least not abruptly so. Rates tend to be obtained over rather short individual time periods, and the origin of the timescale

is not usually keyed to a life-changing event such as disease onset or major surgery.

If the hazard is constant, then the distribution of the lifetime is the *exponential distribution* with density $\rho e^{-\rho t}$ and survival function $e^{-\rho t}$.

Likelihood analysis

Likelihoods for censored data can be constructed using terms that are either the probability density at the time of death or the survival probability in the case of censoring. In the constant-hazard case, the two kinds of terms differ only in the presence of the factor ρ, which we may conveniently encode using the event indicator x_i so that the log-likelihood terms are

$$l(\rho) = x_i \log \rho - \rho T_i$$

Except for the constant, which does not depend on ρ, these terms are formally identical to a Poisson likelihood, where the count is 1 (death) or zero (censoring). This is the crucial "trick" that allows survival data with constant hazard to be analyzed by Poisson regression methods.

The trick can be extended to hazards that are only piecewise constant. Suppose the lifetime of an individual is subdivided as $T_i = T_i^{(1)} + \cdots + T_i^{(k)}$, where the hazard is assumed constant during each section of time. The corresponding log-likelihood term is

$$l(\rho_1, \ldots, \rho_k) = \sum_{j=1}^{k} (x_i^{(j)} \log \rho_j - \rho_j T_i^{(j)})$$

in which the first $k-1$ of the $x_i^{(j)}$ will be 0, and only the last one, $x_i^{(k)}$, can be either 0 or one. The point of writing it in this elaborate form is that it then becomes obvious that the likelihood contribution *might as well* have come from k different individuals where the first $k-1$ had censored observations.

This is the rationale behind time-splitting techniques where the observation time of one subject is divided into observations for multiple pseudo-individuals.

It should be noted that although the models with (piecewise) constant hazard can be fitted and analyzed by likelihood techniques, pretending that the data have come from a Poisson distribution, this does not extend to all aspects of the model. For instance following a cohort to extinction will lead to a fixed total number of events by definition, whereas the corresponding Poisson model implies that the total event count has a Poisson

distribution. Both types of models deal in rates, counts per time, but the difference is to what extent the random variation lies in the counts or in the amount of time. When data are frequently censored (i.e., the event is rare), the survival model becomes well approximated by the Poisson model.

15.2 Fitting Poisson models

The class of generalized linear models (see Section 13.1) also includes the Poisson distribution, which by default uses a log link function. This is the mathematically convenient option and also a quite natural choice since it allows the linear predictor to span the entire real line. We can use this to formulate models for the log rates of the form

$$\log \rho = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_k x_k$$

or, since `glm` needs a model for the expected counts rather than rates,

$$\log \lambda = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_k x_k + \log T$$

A feature of many Poisson models is that the model contains an *offset* in the linear predictor, $\log T$ in this case. Notice that this is not the same as including the term as a regression variable since the regression coefficient is fixed at 1.

The following example was used by Erling B. Andersen in 1977. It involves the rates of lung cancer by age in four Danish cities and may be found as `eba1977` in the `ISwR` package.

```
> names(eba1977)
[1] "city"  "age"   "pop"   "cases"
> attach(eba1977)
```

To fit a model that has multiplicative effects of `age` and `city` on the rate of lung cancer cases, we use the `glm` function in much the same way as in logistic regression. Of course, we need to change the `family` argument to accommodate Poisson-distributed data. We also need to incorporate an offset to account for the different sizes and age structures of the populations in the four cities.

```
> fit <- glm(cases~city+age+offset(log(pop)), family=poisson)
> summary(fit)
Call:
glm(formula = cases ~ city + age + offset(log(pop)), family=poisson)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.63573  -0.67296  -0.03436   0.37258   1.85267
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -5.6321     0.2003 -28.125  < 2e-16 ***
cityHorsens  -0.3301     0.1815  -1.818   0.0690 .
cityKolding  -0.3715     0.1878  -1.978   0.0479 *
cityVejle    -0.2723     0.1879  -1.450   0.1472
age55-59      1.1010     0.2483   4.434 9.23e-06 ***
age60-64      1.5186     0.2316   6.556 5.53e-11 ***
age65-69      1.7677     0.2294   7.704 1.31e-14 ***
age70-74      1.8569     0.2353   7.891 3.00e-15 ***
age75+        1.4197     0.2503   5.672 1.41e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 129.908  on 23  degrees of freedom
Residual deviance:  23.447  on 15  degrees of freedom
AIC: 137.84

Number of Fisher Scoring iterations: 5
```

The offset was included in the model formula in this case. Alternatively, it could have been given as a separate argument as in

```
glm(cases~city+age, offset = log(pop), family=poisson)
```

The table labelled "Coefficients:" contains regression coefficients for the linear predictor along with standard errors and z tests. These can be interpreted in the same way as in ordinary multiple regression or logistic regression. Since both variables are factors and we are using treatment contrasts (see Section 12.3), the coefficients indicate differences in the log rate (i.e., the log of the rate ratio) compared with the city of Fredericia and with the 50–54-year-olds, respectively.

The intercept term refers to the log rate for the group of 50–54-year-olds in Fredericia. Notice that because we used the population size rather than the number of person-years in the offset and the data cover the years 1968–1971, this rate will effectively be per 4 person-years.

A goodness-of-fit statistic is provided by comparing the residual deviance to a χ^2 distribution on the stated degrees of freedom. This statistic is generally considered valid if the expected count in all cells is larger than 5. Accordingly,

```
> min(fitted(fit))
[1] 6.731286
> pchisq(deviance(fit), df.residual(fit), lower=F)
[1] 0.07509017
```

and we see that the model fits the data acceptably. Of course, we could also just have read off the residual deviance and degrees of freedom from the `summary` output:

```
> pchisq(23.45, 15, lower=F)
[1] 0.07504166
```

From the coefficient table, it is obvious that there is an age effect, but it is less clear whether there is a city effect. We can perform χ^2 tests for each term by using `drop1` and looking at the changes in the deviance.

```
> drop1(fit, test="Chisq")
Single term deletions

Model:
cases ~ city + age + offset(log(pop))
       Df Deviance     AIC     LRT Pr(Chi)
<none>      23.447 137.836
city    3   28.307 136.695   4.859  0.1824
age     5  126.515 230.903 103.068  <2e-16 ***
...
```

We see that the `age` term is significant, hardly surprisingly, but the `city` term apparently is not. However, if you can argue a priori that Fredericia could be expected to have a higher cancer rate than the three other cities, then it could be warranted to combine the three other cities into one and perform an analysis as below.

```
> fit2 <- glm(cases~(city=="Fredericia")+age+offset(log(pop)),
+                  family=poisson)
> anova(fit, fit2, test="Chisq")
Analysis of Deviance Table

Model 1: cases ~ city + age + offset(log(pop))
Model 2: cases ~ (city == "Fredericia") + age + offset(log(pop))
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1        15    23.4475
2        17    23.7001 -2  -0.2526    0.8814
> drop1(fit2,  test="Chisq")
Single term deletions

Model:
cases ~ (city == "Fredericia") + age + offset(log(pop))
                     Df Deviance     AIC     LRT Pr(Chi)
<none>                    23.700 134.088
city == "Fredericia"  1   28.307 136.695   4.606 0.03185 *
age                   5  127.117 227.505 103.417 < 2e-16 ***
...
```

According to this, you may combine the three cities other than Fredericia, and, once this is done, Fredericia does indeed appear to be significantly

different from the others. Alternatively, you can look at the coefficients in `fit2` directly

```
> summary(fit2)
...
Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)              -5.9589     0.1809 -32.947  < 2e-16 ***
city == "Fredericia"TRUE  0.3257     0.1481   2.200   0.0278 *
age55-59                  1.1013     0.2483   4.436 9.17e-06 ***
age60-64                  1.5203     0.2316   6.564 5.23e-11 ***
age65-69                  1.7687     0.2294   7.712 1.24e-14 ***
age70-74                  1.8592     0.2352   7.904 2.71e-15 ***
age75+                    1.4212     0.2502   5.680 1.34e-08 ***
...
```

and see the p-value of 0.0278. This agrees with the 0.03185 from `drop1`; you cannot expect the two p-values to be perfectly equal since they rely on different asymptotic approximations. If you really push it, you can argue that a one-sided test with half the p-value is appropriate since you would only expect Fredericia to be more harmful than the others, not less. However, the argumentation becomes tenuous, and in his paper Andersen outlines the possibility of testing Fredericia against the other cities but stops short of providing any p-value, stating that in his opinion "there is no reason to believe a priori that Fredericia is the more dangerous city".

It is sometimes preferred to state the results of Poisson regression analysis in terms of *rate ratios* by taking `exp()` of the estimates (this parallels the presentation of logistic regression analysis in terms of odds ratios in Section 13.4). The intercept term is not really a ratio but a rate, and for nonfactor covariates it should be understood that the coefficient is the relative change *per unit* change in the covariate. Because of the nonlinear transformation, standard errors are not useful; instead one can calculate confidence intervals for the coefficients as follows:

```
> cf <- coefficients(summary(fit2))
> est <- cf[,1]
> s.e. <- cf[,2]
> rr <- exp(cbind(est, est - s.e.*qnorm(.975), est
+                      + s.e.*qnorm(.975) ))
> colnames(rr) <- c("RateRatio", "CI.lo","CI.hi")
> rr
                          RateRatio       CI.lo        CI.hi
(Intercept)             0.002582626 0.001811788  0.003681423
city == "Fredericia"TRUE 1.384992752 1.036131057  1.851314957
age55-59                 3.008134852 1.849135187  4.893571521
age60-64                 4.573665854 2.904833526  7.201245496
age65-69                 5.863391064 3.740395488  9.191368903
age70-74                 6.418715646 4.047748963 10.178474731
age75+                   4.142034525 2.536571645  6.763637070
```

Actually, we can do better by using the `confint` function. This calculates confidence intervals by profiling the likelihood function instead of using the approximation with the normal distribution inherent in the use of asymptotic standard errors. This is done like this:

```
> exp(cbind(coef(fit2), confint(fit2)))
Waiting for profiling to be done...
                                          2.5 %       97.5 %
(Intercept)              0.002582626 0.001776461  0.003617228
city == "Fredericia"TRUE 1.384992752 1.029362341  1.841224091
age55-59                 3.008134852 1.843578634  4.902339637
age60-64                 4.573665854 2.912314045  7.248143959
age65-69                 5.863391064 3.752718226  9.256907108
age70-74                 6.418715646 4.053262281 10.234338998
age75+                   4.142034525 2.527117848  6.771833979
```

In the present case, we are well within the regime where the asymptotic normal approximation works well, so there is little difference between the two displays. However, in some cases where some expected cell counts are low and one or several coefficients are poorly determined, the difference can be substantial.

15.3 Computing rates

We return to the Welsh nickel worker data discussed in Chapter 10. In that section, we discussed how to split the individual lifetime data into smaller pieces that could reasonably be merged with the standard mortality table in the `ewrates` data.

The result of this initial data restructuring is in the `nickel.expand` data set. It contains data from a lot of short time intervals like this:

```
> head(nickel.expand)
  agr  ygr  id icd exposure      dob  age1st   agein ageout lung
1  20 1931 325   0        0 1910.500 14.0737 23.7465     25    6
2  20 1931 273   0        0 1909.500 14.6913 24.7465     25    6
3  20 1931 110   0        0 1909.247 14.0302 24.9999     25    6
4  20 1931 574   0        0 1909.729 14.0356 24.5177     25    6
5  20 1931 213   0        0 1910.129 14.2018 24.1177     25    6
6  20 1931 546   0        0 1909.500 14.4945 24.7465     25    6
  nasal other
1     0  3116
2     0  3116
3     0  3116
4     0  3116
5     0  3116
6     0  3116
```

The same individuals reappear later in the data at older ages. For example, all data for the individual with `id` number 325 are

```
> subset(nickel.expand, id==325)
    agr  ygr  id icd exposure    dob  age1st   agein  ageout lung
1    20 1931 325   0        0 1910.5 14.0737 23.7465 25.0000    6
13   25 1931 325   0        0 1910.5 14.0737 25.0000 30.0000   14
172  30 1936 325   0        0 1910.5 14.0737 30.0000 35.0000   30
391  35 1941 325   0        0 1910.5 14.0737 35.0000 40.0000   81
728  40 1946 325 434        0 1910.5 14.0737 40.0000 43.0343  236
    nasal other
1       0  3116
13      0  3024
172     1  3188
391     1  3549
728     3  3643
```

Accordingly, this subject enters the study at age 23.7 and we follow him through five age groups until his death at age 43.

The variable `ygr` reflects the year of entry into the interval, so even though the subject dies in 1953, the last record is coded as belonging to the years 1946–1950.

Subject no. 325 has the `icd` code 434 in his last record. This refers to the International Classification of Diseases (version 7) and indicates "Other and unspecified diseases of the heart" as the cause of death. For the purposes of this chapter, we are primarily interested in lung cancer, which has codes 162 and 163, so we define a variable to indicate whether this is the cause of death. (Expect a warning about masking the `lung` data set upon attaching.)

```
> nickel.expand <- within(nickel.expand,
+      lung.cancer <- as.numeric(icd %in% c(162,163)))
> attach(nickel.expand)
```

The `%in%` operator returns a logical vector that is `TRUE` when the corresponding element of the operand on the left is contained in the vector that is the operand on the right and `FALSE` in all other cases. Use of this operator is slightly dangerous in the case of an `NA` element in `icd`, but in these particular data, there are none. We convert the result to zero or one since we are going to pretend that it is a Poisson count later on (this is not strictly necessary). Notice that by using `lung.cancer` as the endpoint, we treat death from all other causes, including "unknown", as censoring.

Each record provides `ageout - agein` person-years of risk time, so to tabulate the risk times, we can just do as follows:

```
> pyr <- tapply(ageout-agein,list(ygr,agr), sum)
> print(round(pyr), na.print="-")
```

```
     20 25  30  35  40  45  50  55  60  65  70  75 80
1931  3 86 268 446 446 431 455 323 159  23   4   -  -
1936  -  - 100 327 504 512 503 472 314 130  20   5  -
1941  -  -   0 105 336 481 482 445 368 235  80  14  3
1946  -  -   -   - 102 335 461 404 369 263 157  43 10
1951  -  -   -   -   - 95 299 415 334 277 181  92 31
1956  -  -   -   -   -  - 89 252 364 257 181 101 52
1961  -  -   -   -   -  -  - 71 221 284 150 104 44
1966  -  -   -   -   -  -  -  - 66 168 208  93 51
1971  -  -   -   -   -  -  -  -  - 57 133 131 54
1976  -  -   -   -   -  -  -  -  -  - 31  68 53
```

Notice that there are many `NA` entries in cells that no subject ever entered. The subjects in the study were born between 1864 and 1910, so there is a large block missing in the lower left and a smaller block in the upper right. The `na.print` option to `print` allows you to represent these missing values by a string that is less visually imposing than the default `"NA"`.

The corresponding counts of lung cancer cases are obtained as

```
> count <- tapply(lung.cancer, list(ygr, agr), sum)
> print(count, na.print="-")
     20 25 30 35 40 45 50 55 60 65 70 75 80
1931  0  0  0  0  0  4  2  2  2  0  0  -  -
1936  -  -  0  0  2  3  4  6  5  1  0  0  -
1941  -  -  0  0  0  3  7  5  6  3  2  0  0
1946  -  -  -  -  0  0  8  7  6  2  2  0  0
1951  -  -  -  -  -  0  3  3  9  6  1  0  0
1956  -  -  -  -  -  -  0  4  3  6  1  2  0
1961  -  -  -  -  -  -  -  0  1  1  3  2  1
1966  -  -  -  -  -  -  -  -  2  0  0  1  0
1971  -  -  -  -  -  -  -  -  -  0  0  2  2
1976  -  -  -  -  -  -  -  -  -  -  0  1  1
```

and the cancer rates can be obtained as the ratio of the counts to the risk time. These are small, so we multiply by 1000 to get rates per 1000 person-years.

```
> print(round(count/pyr*1000, 1), na.print="-")
     20 25 30 35 40   45   50   55   60   65   70   75   80
1931  0  0  0  0  0 9.3  4.4  6.2 12.6  0.0  0.0    -    -
1936  -  -  0  0  4 5.9  7.9 12.7 15.9  7.7  0.0  0.0    -
1941  -  -  0  0  0 6.2 14.5 11.2 16.3 12.8 25.0  0.0  0.0
1946  -  -  -  -  0 0.0 17.4 17.3 16.3  7.6 12.8  0.0  0.0
1951  -  -  -  -  - 0.0 10.0  7.2 27.0 21.7  5.5  0.0  0.0
1956  -  -  -  -  -   -  0.0 15.9  8.2 23.4  5.5 19.8  0.0
1961  -  -  -  -  -   -    -  0.0  4.5  3.5 19.9 19.3 22.8
1966  -  -  -  -  -   -    -    - 30.1  0.0  0.0 10.7  0.0
1971  -  -  -  -  -   -    -    -    -  0.0  0.0 15.2 36.8
1976  -  -  -  -  -   -    -    -    -    -  0.0 14.6 19.0
```

Comparison of these rates with those in `ewrates` suggests that they are very high. However, this kind of display has the disadvantage that it hides the actual counts on which the rates are based. For instance, the lower part of the column for 80–84-year-olds jumps by roughly 20 units for each additional case since there are only about 50 person-years per cell.

It may be better to compute the expected counts in each cell based on the standard mortality table and then compare that to the actual counts. Since we have already merged in the `ewrates` data, this is just a matter of multiplying each piece of risk time by the rate. We need to divide by `1e6` (i.e., $10^6 = 1000000$) since the standard rates are given per million person-years.

```
> expect.count <- tapply(lung/1e6*(ageout-agein),
+                        list(ygr,agr), sum)
> print(round(expect.count, 1), na.print="-")
     20 25 30 35  40  45  50  55  60  65  70  75  80
1931  0  0  0  0 0.1 0.1 0.2 0.2 0.1 0.0 0.0   -   -
1936  -  -  0  0 0.1 0.1 0.2 0.3 0.2 0.1 0.0 0.0   -
1941  -  -  0  0 0.1 0.2 0.3 0.4 0.4 0.2 0.1 0.0 0.0
1946  -  -  -  - 0.0 0.2 0.4 0.5 0.6 0.5 0.2 0.0 0.0
1951  -  -  -  -   - 0.1 0.4 0.8 0.9 0.8 0.5 0.2 0.0
1956  -  -  -  -   -   - 0.1 0.6 1.2 1.0 0.7 0.3 0.1
1961  -  -  -  -   -   -   - 0.2 0.8 1.4 0.7 0.5 0.1
1966  -  -  -  -   -   -   -   - 0.2 0.9 1.3 0.6 0.2
1971  -  -  -  -   -   -   -   -   - 0.3 0.9 1.0 0.3
1976  -  -  -  -   -   -   -   -   -   - 0.2 0.6 0.4
```

The observed counts are clearly much larger than expected. We can summarize them by calculating the overall SMR (standardized mortality rate), which is simply the ratio of the total number of cases to the total expected number of cases.

```
> expect.tot <- sum(lung/1e6*(ageout-agein))
> expect.tot
[1] 24.19893
> count.tot <- sum(lung.cancer)
> count.tot
[1] 137
> count.tot/expect.tot
[1] 5.661408
```

That is, this data set has almost six times as many cancer deaths as you would expect from the mortality of the general population.

15.4 Models with piecewise constant intensities

We can formulate the SMR analysis as a "Poisson" regression model in the sense of Section 15.1.2. The assumption behind the SMR is that there is a constant rate ratio to the standard mortality, so we can fit a model with only an intercept while having an offset, which is the log of the expected count. This is not really different from modelling rates — the population mortality ρ_i is just absorbed into the offset, $\log \rho_i + \log T_i = \log \rho_i T_i$.

```
> fit <- glm(lung.cancer ~ 1, poisson,
+            offset = log((ageout-agein)*lung/1e6))
> summary(fit)
...
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.73367    0.08544   20.29   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1175.6  on 3723  degrees of freedom
Residual deviance: 1175.6  on 3723  degrees of freedom
AIC: 1451.6

Number of Fisher Scoring iterations: 7
```

Notice that this is based on individual data; the dependent variable `lung.cancer` is zero or one. We could have aggregated the data according to the cross-classification of `agr` and `ygr` and analyzed the number of cases in each cell. This would have allowed `glm` to run much faster, but on the other hand it would then not be possible to add individual covariates such as age at first exposure.

In this case, we cannot use the deviances for model checking both because the expected counts per cell are very small and because we do not actually have Poisson-distributed data. However, the standard error and the p-value should be reliable if the assumptions hold.

The connection between this analysis and the SMR can be seen immediately from

```
> exp(coef(fit))
(Intercept)
   5.661408
```

This value is exactly the SMR value from the previous section.

We can analyze the data more thoroughly using regression methods. As a first approach, we investigate whether the SMR is constant over year and age groups using a multiplicative Poisson model.

We need to simplify the groupings because some of the groups contain very few cases. By calculating the marginal tables of counts, we get some idea of what to do.

```
> tapply(lung.cancer, agr, sum)
20 25 30 35 40 45 50 55 60 65 70 75 80
 0  0  0  0  2 10 24 27 34 19  9  8  4
> tapply(lung.cancer, ygr, sum)
1931 1936 1941 1946 1951 1956 1961 1966 1971 1976
  10   21   26   25   22   16    8    3    4    2
```

To get at least 10 cases per level, we combine all values of `agr` up to 45 (i.e., ages less than 50) and also those from 70 and up. Similarly, we combine all values of `ygr` for the periods from 1961 onwards.

```
> detach()
> nickel.expand <- within(nickel.expand,{
+     A <- factor(agr)
+     Y <- factor(ygr)
+     lv <- levels(A)
+     lv[1:6] <- "< 50"
+     lv[11:13] <- "70+"
+     levels(A) <- lv
+     lv <- levels(Y)
+     lv[7:10] <- "1961ff"
+     levels(Y) <- lv
+     rm(lv)
+ })
> attach(nickel.expand)
```

Notice that this is a case where the `within` function (see Section 2.1.8) works better than `transform` because it allows more flexibility, including the creation of temporary variables such as `lv`.

We can analyze the effect of `A` and `Y` on the mortality ratio by building a log-additive model in the usual way. Notice that we still use the original grouping in the calculation of the offset; it is only the SMR that is assumed to be the same for everyone below 50, etc. We use `drop1` to test the significance of the two factors.

```
> fit <- glm(lung.cancer ~ A + Y, poisson,
+            offset=log((ageout-agein)*lung/1e6))
> drop1(fit, test="Chisq")
Single term deletions

Model:
lung.cancer ~ A + Y
```

```
       Df Deviance     AIC     LRT  Pr(Chi)
<none>     1069.73 1367.73
A       5  1073.81 1361.81    4.08   0.5376
Y       6  1118.50 1404.50   48.77 8.29e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So it seems that we do not need the age grouping in the model, but the year grouping is needed. Accordingly, we fit a model with `Y` alone, and by dropping the intercept, we get a parameterization with a separate intercept for each level of `Y`.

```
> fit <- glm(lung.cancer ~ Y - 1, poisson,
+            offset=log((ageout-agein)*lung/1e6))
> summary(fit)
...
Coefficients:
        Estimate Std. Error z value Pr(>|z|)
Y1931     2.6178     0.3162   8.279  < 2e-16 ***
Y1936     3.0126     0.2182  13.805  < 2e-16 ***
Y1941     2.7814     0.1961  14.182  < 2e-16 ***
Y1946     2.2787     0.2000  11.394  < 2e-16 ***
Y1951     1.8038     0.2132   8.461  < 2e-16 ***
Y1956     1.3698     0.2500   5.479 4.27e-08 ***
Y1961ff   0.4746     0.2425   1.957   0.0504 .
....
```

The regression coefficients may again be recognized as log-SMR values, as the following demonstrates:

```
> round(exp(coef(fit)), 1)
  Y1931   Y1936   Y1941   Y1946   Y1951   Y1956 Y1961ff
   13.7    20.3    16.1     9.8     6.1     3.9     1.6
> expect.count <-  tapply(lung/1e6*(ageout-agein), Y, sum)
> count <- tapply(lung.cancer, Y, sum)
> cbind(count=count, expect=round(expect.count,1),
+       SMR= round(count/expect.count, 1))
       count expect  SMR
1931      10    0.7 13.7
1936      21    1.0 20.3
1941      26    1.6 16.1
1946      25    2.6  9.8
1951      22    3.6  6.1
1956      16    4.1  3.9
1961ff    17   10.6  1.6
```

The advantage of using the regression approach is that it provides a framework in which you can formulate statistical tests and investigate the effect of multiple regression variables simultaneously.

Breslow and Day analyzed the nickel data in their seminal book (Breslow and Day, 1987) on the analysis of cohort studies. In their analysis, they split the individual risk times according to three criteria, two of them being age and period, to match the standard mortality table, but they also treat time from employment as a time-dependent covariate with a piecewise constant effect, which requires that the person-year be split further according to the interval boundaries. They then represent time effects using three variables: time since, age at, and year of first employment, `TFE`, `AFE`, and `YFE`, respectively. In addition, they include a measure of exposure level.

The following analysis roughly reproduces the Breslow and Day analysis. It is not completely similar because we settle for splitting time according to `agr` only and use the age at entry into each interval to define the `TFE` variable as well as for choosing the relevant standard mortality rates. However, to enable some comparison of results, we define `cut` groups in a manner that is similar to that of Breslow and Day.

```
> detach()
> nickel.expand <- within(nickel.expand,{
+     TFE <- cut(agein-age1st, c(0,20,30,40,50,100), right=F)
+     AFE <- cut(age1st, c(0, 20, 27.5, 35, 100), right=F)
+     YFE <- cut(dob + age1st, c(0, 1910, 1915, 1920, 1925),right=F)
+     EXP <- cut(exposure, c(0, 0.5, 4.5, 8.5, 12.5, 25), right=F)
+ })
> attach(nickel.expand)
```

Some relabelling of group levels might be called for — e.g., the levels for `EXP` are really 0, 0.5–4, 4.5–8, 8.5–12, 12.5+ — but let us not make more of it than necessary.

We fit a multiplicative model and test the significance of the individual terms as follows:

```
> fit <- glm(lung.cancer ~ TFE + AFE + YFE + EXP, poisson,
+            offset=log((ageout-agein)*lung/1e6))
> drop1(fit, test="Chisq")
Single term deletions

Model:
lung.cancer ~ TFE + AFE + YFE + EXP
       Df Deviance     AIC     LRT   Pr(Chi)
<none>     1052.91 1356.91
TFE     4  1107.33 1403.33   54.43 4.287e-11 ***
AFE     3  1054.99 1352.99    2.08 0.5560839
YFE     3  1058.06 1356.06    5.15 0.1608219
EXP     4  1071.98 1367.98   19.07 0.0007606 ***
```

This suggests that the two major terms are `TFE` and `EXP`, whereas `AFE` and `YFE` could be taken out of the model. Notice, though, that it cannot be

concluded from the above that both can be removed. In principle, one of them could become significant when the other is removed. This does not happen in this case, though.

The table of coefficients looks like this:

```
> summary(fit)
...
Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     2.36836    0.55716   4.251 2.13e-05 ***
TFE[20,30)     -0.21788    0.36022  -0.605 0.545284
TFE[30,40)     -0.77184    0.36529  -2.113 0.034605 *
TFE[40,50)     -1.87583    0.41707  -4.498 6.87e-06 ***
TFE[50,100)    -2.22142    0.55068  -4.034 5.48e-05 ***
AFE[20,27.5)    0.28506    0.31524   0.904 0.365868
AFE[27.5,35)    0.21961    0.34011   0.646 0.518462
AFE[35,100)    -0.10818    0.44412  -0.244 0.807556
YFE[1910,1915)  0.04826    0.27193   0.177 0.859137
YFE[1915,1920) -0.56397    0.37585  -1.501 0.133483
YFE[1920,1925) -0.42520    0.30017  -1.417 0.156614
EXP[0.5,4.5)    0.58373    0.21200   2.753 0.005897 **
EXP[4.5,8.5)    1.03175    0.28364   3.638 0.000275 ***
EXP[8.5,12.5)   1.18345    0.37406   3.164 0.001557 **
EXP[12.5,25)    1.28601    0.48236   2.666 0.007674 **
...
```

A dose-response pattern and a declining effect of time since first employment seem to be present.

The results may be more readily interpreted if they are given in terms of ratios and confidence intervals. These can be obtained in exactly the same way as in the analysis of the `eba1977` data.

15.5 Exercises

15.1 In the `bcmort` data set, we defined the `period` and `area` factors in Exercise 10.2. Fit a Poisson regression model to the data with `age`, `period`, and `area` as descriptors, as well as the three two-factor interaction terms. The interaction between `period` and `area` can be interpreted as the effect of screening.

15.2 With the split `stroke` data from Exercise 10.4, fit a Poisson regression model corresponding to a constant hazard in each interval and with multiplicative effects of `age` and `sex`.