Statistical Analysis in the Lexis Diagram: Age-Period-Cohort models — and cousins

KEA, University of Aarhus
2 – 3 May 2023
http://BendixCarstensen.com/APC/KEA-2023
Version 3

Compiled Monday 1st May, 2023, 09:30
from: `C:\Bendix\teach\APC\courses\KEA2023\pracs/pracs.tex`

Bendix Carstensen Steno Diabetes Center Copenhagen, Herlev, Denmark
& Department of Biostatistics, University of Copenhagen
bendix.carstensen@regionh.dk b@bxc.dk
http://BendixCarstensen.com

Chapter 1

Program and introduction

1.1 Format of the course

The course will be centered around the practicals.

There will be lectures introducing concepts and demonstrating the practicalities of computing. Between lectures you will be asked to do practical computer exercises in R, and there will brief recaps of the practicals after each practical session.

The practicals also include the solution for you to read.

Further, the complete code for each practical is available at the course website, so you can run all the code without thinking. But you also have the opportunity to investigate the strutures you create on the way and do additional things too.

1.2 Topics covered

- Introduction
- Rates and Survival
- Likelihood for rates
- Lifetables
- Who needs the Cox-model anyway?
- Models for tabulated data
- Age-Period and Age-Cohort models
- Age-drift model
- Age-Period-Cohort model
- Age at entry: Age-Duration-Diagnosis
- Tabulation in the Lexis diagram
- APC-model for triangular data
- Non-linear effects
- APC-model: Parametrization
- APC-models for several datasets
- Predicting future rates
- APC-model: Interactions
- Lee-Carter model
- Continuous outcomes

Chapter 2

Practical exercises

2.1 Age-period model

The following exercise is aimed at familiarizing you with the parametrization of the age-period model. It will give you the opportunity explore how to extract and and plot parameter estimates from models. It is based on Danish male lung cancer incidence data in 5-year classes.

First load the Epi package:

```
library(Epi)
library(tidyverse)
```

1. First we read the data in the file lung5-M.txt, and make a table of the events and person-years.

```
lung <- read.table("http://bendixcarstensen.com/APC/KEA-2023/data/lung5-M.txt",
                   header = T)
lung <- read.table("../data/lung5-M.txt", header = T)
head(lung)
```

```
   A    P  D        Y
1 40 1943 80 694046.5
2 40 1948 81 754769.5
3 40 1953 73 769440.7
4 40 1958 99 749264.5
5 40 1963 82 757240.0
6 40 1968 97 709558.5
```

```
with(lung , table(A))
```

```
A
40 45 50 55 60 65 70 75 80 85
11 11 11 11 11 11 11 11 11 11
```

```
with(lung , table(P))
```

```
P
1943 1948 1953 1958 1963 1968 1973 1978 1983 1988 1993
  10   10   10   10   10   10   10   10   10   10   10
```

```
round(ftable(addmargins(xtabs(cbind(D = D,
                                    Y = Y/1000) ~  A + P,
                              data = lung),
                        margin = 1),
             row.vars = c(3, 1)), 0)
```

```
      P   1943  1948  1953  1958  1963  1968  1973  1978  1983  1988  1993
  A
D 40        80    81    73    99    82    97    86    90   116   149    91
  45       135   163   208   226   252   284   263   251   257   265   251
  50       197   292   442   508   560   580   657   608   591   493   446
  55       261   404   596   772  1052  1075  1115  1218  1090   995   696
  60       213   394   577   955  1342  1682  1654  1826  1885  1497  1113
  65       141   273   491   868  1235  1856  2136  2231  2188  2193  1485
  70       110   215   300   596   976  1448  1924  2283  2293  2157  1691
  75        54   126   167   320   514   860  1213  1559  1824  1640  1221
  80        20    57    87   157   220   390   573   753   881   837   716
  85         7    10    23    48    72   110   176   213   307   286   262
  Sum     1218  2015  2964  4549  6305  8382  9797 11032 11432 10512  7972
Y 40       694   755   769   749   757   710   695   756   941  1026   753
  45       622   677   738   754   737   747   698   681   742   924   821
  50       539   601   654   716   734   718   725   675   659   720   701
  55       471   512   571   622   681   699   683   687   641   626   544
  60       403   435   474   528   573   627   644   628   630   591   463
  65       329   358   386   420   463   501   548   564   549   553   421
  70       230   269   295   317   341   374   404   443   459   449   366
  75       140   167   196   215   229   246   268   290   319   336   263
  80        68    81    99   116   126   137   150   163   176   196   168
  85        25    28    34    42    49    56    64    71    78    85    75
  Sum     3521  3882  4217  4480  4691  4814  4880  4959  5194  5508  4575
```

The last table shows that the last period is a bit shorter than the other; it is only 4 years; the person-years are approximately 80% of those in the previous years and previous age.

2. We fit a Poisson model with effects of age (`A`) and period (`P`) as class variables — note that you can use `factor` on the variables in the model formula to get the parametrization with one parameter per level:

```
ap.1 <- glm(D ~ factor(A) + factor(P),
            offset = log(Y / 1000),
            family = poisson,
              data = lung)
summary(ap.1)
```

```
Call:
glm(formula = D ~ factor(A) + factor(P), family = poisson, data = lung,
    offset = log(Y/1000))

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -3.43459    0.04192  -81.93   <2e-16 ***
factor(A)45    0.95258    0.03673   25.93   <2e-16 ***
factor(A)50    1.78237    0.03383   52.69   <2e-16 ***
factor(A)55    2.41412    0.03265   73.94   <2e-16 ***
factor(A)60    2.86259    0.03216   89.01   <2e-16 ***
factor(A)65    3.15159    0.03201   98.47   <2e-16 ***
factor(A)70    3.31784    0.03209  103.40   <2e-16 ***
factor(A)75    3.30980    0.03261  101.50   <2e-16 ***
factor(A)80    3.17640    0.03423   92.81   <2e-16 ***
factor(A)85    2.90983    0.04024   72.32   <2e-16 ***
factor(P)1948  0.39206    0.03629   10.80   <2e-16 ***
factor(P)1953  0.67592    0.03404   19.86   <2e-16 ***
factor(P)1958  1.01434    0.03226   31.44   <2e-16 ***
factor(P)1963  1.26666    0.03130   40.47   <2e-16 ***
factor(P)1968  1.48717    0.03067   48.49   <2e-16 ***
factor(P)1973  1.59239    0.03039   52.40   <2e-16 ***
factor(P)1978  1.67994    0.03020   55.62   <2e-16 ***
factor(P)1983  1.69902    0.03015   56.35   <2e-16 ***
factor(P)1988  1.59958    0.03028   52.83   <2e-16 ***
factor(P)1993  1.52558    0.03078   49.57   <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: 71776.2  on 109  degrees of freedom
Residual deviance:  2723.5  on  90  degrees of freedom
AIC: 3620.5

Number of Fisher Scoring iterations: 5
```

3. As an aside on coding, note that the above code also can be shortened to the following unreadable chunk.

```
ap.1<-glm(D~factor(A)+factor(P),offset=log(Y/1000),family=poisson,data=lung)
```

Never show this kind of code to any one else. . .

4. A more handy way of specifying the model for rates is via the poisreg family from the Epi package, where the response is specified intuitively more logically as a two-column matrix of the bivariate response (event, risk time):

```
ap.1 <- glm(cbind(D, Y/1000) ~ factor(A) + factor(P),
            family = poisreg,
              data = lung)
summary(ap.1)
```

```
Call:
glm(formula = cbind(D, Y/1000) ~ factor(A) + factor(P), family = poisreg,
    data = lung)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -3.43459    0.04192  -81.93   <2e-16 ***
factor(A)45    0.95258    0.03673   25.93   <2e-16 ***
factor(A)50    1.78237    0.03383   52.69   <2e-16 ***
factor(A)55    2.41412    0.03265   73.94   <2e-16 ***
factor(A)60    2.86259    0.03216   89.01   <2e-16 ***
factor(A)65    3.15159    0.03201   98.47   <2e-16 ***
factor(A)70    3.31784    0.03209  103.40   <2e-16 ***
factor(A)75    3.30980    0.03261  101.50   <2e-16 ***
factor(A)80    3.17640    0.03423   92.81   <2e-16 ***
factor(A)85    2.90983    0.04024   72.32   <2e-16 ***
factor(P)1948  0.39206    0.03629   10.80   <2e-16 ***
factor(P)1953  0.67592    0.03404   19.86   <2e-16 ***
factor(P)1958  1.01434    0.03226   31.44   <2e-16 ***
factor(P)1963  1.26666    0.03130   40.47   <2e-16 ***
factor(P)1968  1.48717    0.03067   48.49   <2e-16 ***
factor(P)1973  1.59239    0.03039   52.40   <2e-16 ***
factor(P)1978  1.67994    0.03020   55.62   <2e-16 ***
factor(P)1983  1.69902    0.03015   56.35   <2e-16 ***
factor(P)1988  1.59958    0.03028   52.83   <2e-16 ***
factor(P)1993  1.52558    0.03078   49.57   <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: 71776.2  on 109  degrees of freedom
Residual deviance:  2723.5  on  90  degrees of freedom
AIC: 3620.5

Number of Fisher Scoring iterations: 5
```

The intercept parameter refers to the log-rate (per unit of the offset variable, `Y/1000`, that is per 1000 PY) in the reference age-class (40) and reference period (1943) — note that these do not appear among the `A` resp. `P` parameters.

The `A`-parameters refer to the log-rate-ratio relative to age group 40 — this is assume to be the same in all periods. The `P`-parameters refer to the log-rate-ratio relative to period group 1943 — this is assumed to be the same in all age-classes.

5. We can get the the rates and rate-ratios directly by `ci.exp`:

```
 round(ci.exp(ap.1), 2)

              exp(Est.)  2.5% 97.5%
(Intercept)        0.03  0.03  0.03
factor(A)45        2.59  2.41  2.79
factor(A)50        5.94  5.56  6.35
factor(A)55       11.18 10.49 11.92
factor(A)60       17.51 16.44 18.65
factor(A)65       23.37 21.95 24.89
factor(A)70       27.60 25.92 29.39
factor(A)75       27.38 25.68 29.19
factor(A)80       23.96 22.41 25.62
factor(A)85       18.35 16.96 19.86
factor(P)1948      1.48  1.38  1.59
factor(P)1953      1.97  1.84  2.10
factor(P)1958      2.76  2.59  2.94
factor(P)1963      3.55  3.34  3.77
factor(P)1968      4.42  4.17  4.70
factor(P)1973      4.92  4.63  5.22
factor(P)1978      5.37  5.06  5.69
factor(P)1983      5.47  5.15  5.80
factor(P)1988      4.95  4.67  5.25
factor(P)1993      4.60  4.33  4.88
```

6. When we fit the same model without intercept, the sequence of terms in the model is of importance:

```
ap.0 <- glm(cbind(D, Y / 1000) ~ -1 + factor(A) + factor(P),
             family = poisreg,
               data = lung)
round(ci.exp(ap.0), 3)
```

```
              exp(Est.)  2.5% 97.5%
factor(A)40       0.032 0.030 0.035
factor(A)45       0.084 0.078 0.089
factor(A)50       0.192 0.180 0.204
factor(A)55       0.360 0.340 0.382
factor(A)60       0.564 0.532 0.598
factor(A)65       0.754 0.711 0.798
factor(A)70       0.890 0.839 0.943
factor(A)75       0.883 0.832 0.937
factor(A)80       0.772 0.725 0.823
factor(A)85       0.592 0.549 0.638
factor(P)1948     1.480 1.378 1.589
factor(P)1953     1.966 1.839 2.101
factor(P)1958     2.758 2.589 2.938
factor(P)1963     3.549 3.338 3.774
factor(P)1968     4.425 4.166 4.699
factor(P)1973     4.915 4.631 5.217
factor(P)1978     5.365 5.057 5.692
factor(P)1983     5.469 5.155 5.801
factor(P)1988     4.951 4.666 5.254
factor(P)1993     4.598 4.329 4.884
```

When we put `A` before `P` we get the `A`-parameters as (log) rates in the reference period (1943) and the `P`-parameters as rate-ratios relative to this. We see that the latter are the same as in the previous model.

7. We now fit the same model again, but with the period 1968–72 as the reference period, by using the `relevel` command for factors to make 1968 the first level:

```
ap.3 <- glm(cbind(D, Y / 1000) ~ factor(A) - 1 + relevel(factor(P), "1968"),
            family = poisreg,
            data = lung)
```

We see that 1968 actually *is* the reference level:

```
round(ci.exp(ap.3), 3)
```

```
                              exp(Est.)  2.5% 97.5%
factor(A)40                       0.143 0.134 0.152
factor(A)45                       0.370 0.354 0.386
factor(A)50                       0.848 0.820 0.877
factor(A)55                       1.595 1.550 1.641
factor(A)60                       2.497 2.432 2.564
factor(A)65                       3.334 3.249 3.421
factor(A)70                       3.937 3.835 4.042
factor(A)75                       3.905 3.795 4.019
factor(A)80                       3.418 3.300 3.540
factor(A)85                       2.618 2.479 2.764
```

```
relevel(factor(P), "1968")1943     0.226 0.213 0.240
relevel(factor(P), "1968")1948     0.335 0.319 0.351
relevel(factor(P), "1968")1953     0.444 0.426 0.463
relevel(factor(P), "1968")1958     0.623 0.601 0.646
relevel(factor(P), "1968")1963     0.802 0.776 0.829
relevel(factor(P), "1968")1973     1.111 1.079 1.144
relevel(factor(P), "1968")1978     1.213 1.179 1.248
relevel(factor(P), "1968")1983     1.236 1.202 1.271
relevel(factor(P), "1968")1988     1.119 1.087 1.152
relevel(factor(P), "1968")1993     1.039 1.008 1.072
```

— there is no 1968 parameter here—it is the reference level for the period.

8. We extract the age-parameters from the model, by using the `subset` argument to `ci.exp`:

```
 (ap.cf <- ci.exp(ap.3, subset = "A"))
```

```
             exp(Est.)      2.5%     97.5%
factor(A)40 0.1426419 0.1337940 0.1520748
factor(A)45 0.3697834 0.3539531 0.3863216
factor(A)50 0.8478539 0.8199413 0.8767167
factor(A)55 1.5947318 1.5498928 1.6408681
factor(A)60 2.4971972 2.4323484 2.5637749
factor(A)65 3.3340099 3.2493190 3.4209082
factor(A)70 3.9369963 3.8351257 4.0415728
factor(A)75 3.9054785 3.7951559 4.0190081
factor(A)80 3.4177553 3.2996154 3.5401251
factor(A)85 2.6180013 2.4793893 2.7643626
```

These are the age-specific incidence rates in the reference period; in this case the 1968 period.

9. We can also obtain these as *predicted rates* by devising a *prediction* data frame (`nd` for `new data`):

```
 nd <- data.frame(A = seq(40, 85, 5), P = 1968)
 (ap.rt <- ci.pred(ap.3, nd))
```

```
    Estimate      2.5%     97.5%
1  0.1426419 0.1337940 0.1520748
2  0.3697834 0.3539531 0.3863216
3  0.8478539 0.8199413 0.8767167
4  1.5947318 1.5498928 1.6408681
5  2.4971972 2.4323484 2.5637749
6  3.3340099 3.2493190 3.4209082
7  3.9369963 3.8351257 4.0415728
8  3.9054785 3.7951559 4.0190081
9  3.4177553 3.2996154 3.5401251
10 2.6180013 2.4793893 2.7643626
```

```
 (ap.rt <- ci.pred(ap.0, nd))
```

```
     Estimate      2.5%     97.5%
1  0.1426419 0.1337940 0.1520748
2  0.3697834 0.3539531 0.3863216
3  0.8478539 0.8199413 0.8767167
4  1.5947318 1.5498928 1.6408681
5  2.4971972 2.4323484 2.5637749
6  3.3340099 3.2493190 3.4209082
7  3.9369963 3.8351257 4.0415728
8  3.9054785 3.7951559 4.0190081
9  3.4177553 3.2996154 3.5401251
10 2.6180013 2.4793893 2.7643626
```

We see that we get the same predicted rates from the two models—as long as we can specify what we want as a set of predicted rates it does not matter how we parametrize the model.

10. We then plot the incidence rates as a function of age using shaded c.i. (`matshade` is a function in the `Epi` package):

```
 matshade(seq(40, 85, 5) + 2.5, ci.exp(ap.3, subset = "A"),
          plot = TRUE,
          type = "l", lty = 1, lwd = 1, col = 1,
           log = "y",
          xlab = "Age",
          ylab = "Male lung cancer incidence rate per 1000 PY")
```

We see that we plot the parameter estimates, but it is actually more inituitive to plot the predicted rates, using `ci.pred`.

11. Now for the rate-ratio-parameters, take the rest of the coefficients:

```
 (RR.cf <- ci.exp(ap.3, subset = "P"))

                               exp(Est.)      2.5%     97.5%
relevel(factor(P), "1968")1943 0.2260104 0.2128257 0.2400119
relevel(factor(P), "1968")1948 0.3345003 0.3186216 0.3511705
relevel(factor(P), "1968")1953 0.4443021 0.4260752 0.4633088
relevel(factor(P), "1968")1958 0.6232309 0.6011356 0.6461383
relevel(factor(P), "1968")1963 0.8021069 0.7763218 0.8287485
relevel(factor(P), "1968")1973 1.1109511 1.0790196 1.1438275
relevel(factor(P), "1968")1978 1.2125932 1.1786324 1.2475325
relevel(factor(P), "1968")1983 1.2359544 1.2015891 1.2713025
relevel(factor(P), "1968")1988 1.1189707 1.0872878 1.1515769
relevel(factor(P), "1968")1993 1.0391496 1.0077481 1.0715295
```

Note that the reference group is missing, so we must stick 1s in the correct place. We use the command `rbind` (`row-bind`):

```
 (RR.cf <- rbind(RR.cf[1:5, ], 1, RR.cf[6:10, ]))

                               exp(Est.)      2.5%     97.5%
relevel(factor(P), "1968")1943 0.2260104 0.2128257 0.2400119
relevel(factor(P), "1968")1948 0.3345003 0.3186216 0.3511705
relevel(factor(P), "1968")1953 0.4443021 0.4260752 0.4633088
relevel(factor(P), "1968")1958 0.6232309 0.6011356 0.6461383
```

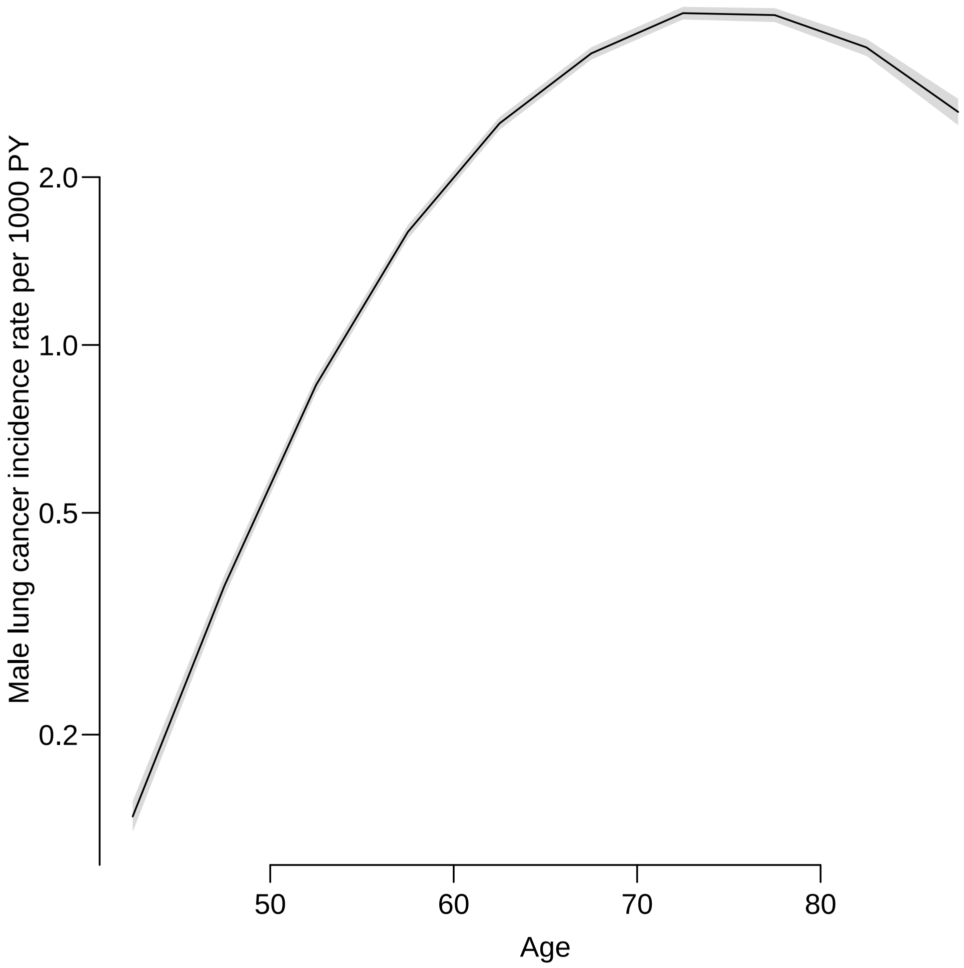

Figure 2.1: *Age-specific incidence rates of male lung cancer in Denmark in 1968. The shaded area is the 95% c.i.* ../graph/AP-AC-agesh

```
relevel(factor(P), "1968")1963 0.8021069 0.7763218 0.8287485
                               1.0000000 1.0000000 1.0000000
relevel(factor(P), "1968")1973 1.1109511 1.0790196 1.1438275
relevel(factor(P), "1968")1978 1.2125932 1.1786324 1.2475325
relevel(factor(P), "1968")1983 1.2359544 1.2015891 1.2713025
relevel(factor(P), "1968")1988 1.1189707 1.0872878 1.1515769
relevel(factor(P), "1968")1993 1.0391496 1.0077481 1.0715295
```

Now we have the same situation as for the age-specific rates, and can plot the relative risks (relative to 1968) in precisely the same way as for the age-specific rates:

```
matshade(seq(1943, 1993, 5) + 2.5, RR.cf,
          lty = 1, lwd = 1, log = "y", col = 1, plot = TRUE,
          xlab = "Calendar time", ylab = "Rate ratio rel. to 1968--72")
abline(h = 1, v = 1970.5, lty = 3)
```

12. What `ci.pred` does is to give a *prediction*, that is a set of *rates*. If we want the *rate-ratios* we are looking for the ratio between two sets of predictions, so not surprisingly we must supply *two* data frames.

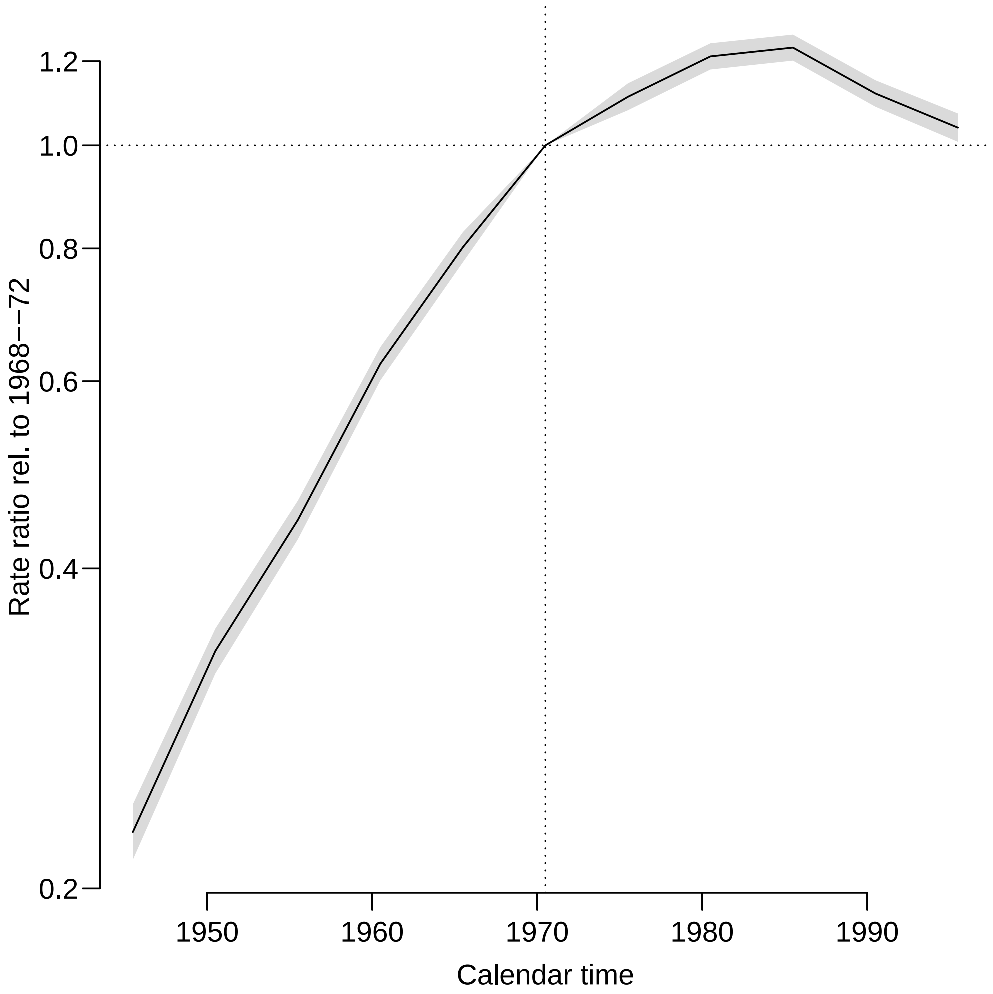

Figure 2.2: *Rate-ratios of male lung cancer in Denmark relative to the period 1968–72.* `../graph/AP-AC-APrrLung`

However this approach does not allow on-the-fly creation of factors in the model formula; this must be done in the `data` argument. In general it is advisable to do the factor definition either in the data separately or as here in the `data=` argument:

```
 ap.x <- glm(cbind(D, Y / 1000) ~ -1 + A + P,
             family = poisreg,
               data = transform(lung, A = factor(A), P = factor(P)))
 summary(ap.x)
```

```
Call:
glm(formula = cbind(D, Y/1000) ~ -1 + A + P, family = poisreg,
    data = transform(lung, A = factor(A), P = factor(P)))

Coefficients:
      Estimate Std. Error z value Pr(>|z|)
A40   -3.43459    0.04192 -81.930  < 2e-16 ***
A45   -2.48201    0.03454 -71.869  < 2e-16 ***
A50   -1.65222    0.03145 -52.534  < 2e-16 ***
A55   -1.02047    0.03020 -33.785  < 2e-16 ***
A60   -0.57201    0.02970 -19.258  < 2e-16 ***
```

```
A65   -0.28300    0.02956  -9.574  < 2e-16 ***
A70   -0.11676    0.02970  -3.931 8.44e-05 ***
A75   -0.12479    0.03031  -4.118 3.83e-05 ***
A80   -0.25819    0.03209  -8.047 8.51e-16 ***
A85   -0.52476    0.03847 -13.642  < 2e-16 ***
P1948  0.39206    0.03629  10.802  < 2e-16 ***
P1953  0.67592    0.03404  19.859  < 2e-16 ***
P1958  1.01434    0.03226  31.439  < 2e-16 ***
P1963  1.26666    0.03130  40.467  < 2e-16 ***
P1968  1.48717    0.03067  48.493  < 2e-16 ***
P1973  1.59239    0.03039  52.403  < 2e-16 ***
P1978  1.67994    0.03020  55.624  < 2e-16 ***
P1983  1.69902    0.03015  56.349  < 2e-16 ***
P1988  1.59958    0.03028  52.826  < 2e-16 ***
P1993  1.52558    0.03078  49.570  < 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: 82830.6  on 110  degrees of freedom
Residual deviance:  2723.5  on  90  degrees of freedom
AIC: 3620.5

Number of Fisher Scoring iterations: 5
```

In order to get the rate-ratios, two data frames are needed, one specifying the target (in this case calendar years), and the other the reference. In principle with all covariates in the model specified, but in some cases we can get away with only specifying the covariates that are different between the two:

```
nd <- data.frame(P = seq(1943, 1993, 5))
nr <- data.frame(P = 1968)
(rrx <- ci.exp(ap.x, list(nd, nr)))
```

```
   exp(Est.)      2.5%     97.5%
1  0.2260104 0.2128257 0.2400119
2  0.3345003 0.3186216 0.3511705
3  0.4443021 0.4260752 0.4633088
4  0.6232309 0.6011356 0.6461383
5  0.8021069 0.7763218 0.8287485
6  1.0000000 1.0000000 1.0000000
7  1.1109511 1.0790196 1.1438275
8  1.2125932 1.1786324 1.2475325
9  1.2359544 1.2015891 1.2713025
10 1.1189707 1.0872878 1.1515769
11 1.0391496 1.0077481 1.0715295
```

The plot of the RR will look exactly as before. Although it seems a bit clumsy to do it this way, its generality will make things much easier along the way—at least as long as we can get away with predictions for deriving results.

2.2 Age-cohort model

13. Data are classified by age and date of follow-up; the difference between date of follow-up and age at follow-up is the date of birth. If we make a table of this difference:

```
with(lung, table(P-A))
```

```
1858 1863 1868 1873 1878 1883 1888 1893 1898 1903 1908 1913 1918 1923 1928 1933 1938 1943
   1    2    3    4    5    6    7    8    9   10   10    9    8    7    6    5    4    3
1948 1953
   2    1
```

we get the number of observations for each level of birth Cohort. We see that the first and last cohort contribute only one observations whereas the 1903 and 1908 cohorts contribute 10 each.

14. Now we fit a Poisson model with effects of age (A) and cohort (C) as factors. We form the factor variable as we did previously:

```
ac.0 <- glm(cbind(D, Y / 1000) ~ A + C,
            family = poisreg,
              data = transform(lung, A = factor(A), C = factor(P-A)))
summary(ac.0)
```

```
Call:
glm(formula = cbind(D, Y/1000) ~ A + C, family = poisreg, data = transform(lung,
    A = factor(A), C = factor(P - A)))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.92725    0.38038 -12.954  < 2e-16 ***
A45          0.96843    0.03800  25.488  < 2e-16 ***
A50          1.83467    0.03591  51.089  < 2e-16 ***
A55          2.51168    0.03508  71.597  < 2e-16 ***
A60          3.02924    0.03476  87.150  < 2e-16 ***
A65          3.40740    0.03471  98.159  < 2e-16 ***
A70          3.67325    0.03487 105.338  < 2e-16 ***
A75          3.78630    0.03545 106.821  < 2e-16 ***
A80          3.78402    0.03704 102.167  < 2e-16 ***
A85          3.66814    0.04280  85.704  < 2e-16 ***
C1863        0.01046    0.42031   0.025 0.980152
C1868        0.51345    0.38845   1.322 0.186240
C1873        0.82684    0.38231   2.163 0.030560 *
C1878        1.05336    0.38054   2.768 0.005639 **
C1883        1.41904    0.37972   3.737 0.000186 ***
C1888        1.91197    0.37927   5.041 4.63e-07 ***
C1893        2.28073    0.37909   6.016 1.78e-09 ***
C1898        2.55794    0.37900   6.749 1.49e-11 ***
C1903        2.76315    0.37895   7.292 3.06e-13 ***
C1908        2.83415    0.37894   7.479 7.48e-14 ***
C1913        2.81410    0.37901   7.425 1.13e-13 ***
C1918        2.86228    0.37902   7.552 4.30e-14 ***
C1923        2.91551    0.37906   7.691 1.45e-14 ***
C1928        2.86546    0.37917   7.557 4.12e-14 ***
C1933        2.86314    0.37936   7.547 4.44e-14 ***
C1938        2.72290    0.37983   7.169 7.57e-13 ***
```

```
C1943        2.68759    0.38066   7.060 1.66e-12 ***
C1948        2.85099    0.38263   7.451 9.27e-14 ***
C1953        2.81411    0.39456   7.132 9.87e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 71776.18  on 109  degrees of freedom
Residual deviance:   829.63  on  81  degrees of freedom
AIC: 1744.7

Number of Fisher Scoring iterations: 4
```

As before the intercept parameter refer to the log-rate in reference age class (40) and reference birth cohort (1858) — rates in a group that is not present in data at all!

15. Also note that since the two extreme cohorts only are represented with one observation each, and since we have one parameter per cohort, these two data points will have perfect fit:

```
 table(near(residuals(ac.0), 0))
```

```
FALSE  TRUE
  108     2
```

```
 subset(lung, near(residuals(ac.0), 0))
```

```
     A    P  D         Y
11  40 1993 91 752950.17
100 85 1943  7  24656.17
```

```
 table(residuals(ac.0) == 0)
```

```
FALSE  TRUE
  108     2
```

Note that `near` gives the correct results—never ask if two quantities on a computer are identical, ask if they are very near each other, using `near`.

16. We fit the same model, without intercept, using the cohort 1908 as the reference cohort. What do the parameters represent now?

```
 ac.r <- glm(cbind(D, Y / 1000) ~ -1 + A + C,
             family = poisreg,
               data = transform(lung,
                                A = factor(A),
                                C = relevel(factor(P - A), "1908")))
 round(ci.exp(ac.r), 3)
```

```
      exp(Est.)  2.5% 97.5%
A40       0.123 0.115 0.132
A45       0.325 0.310 0.340
A50       0.772 0.747 0.799
A55       1.520 1.478 1.563
A60       2.550 2.487 2.615
```

```
A65       3.722 3.634 3.812
A70       4.856 4.740 4.974
A75       5.437 5.295 5.582
A80       5.424 5.248 5.607
A85       4.831 4.580 5.095
C1858     0.059 0.028 0.124
C1863     0.059 0.041 0.085
C1868     0.098 0.083 0.117
C1873     0.134 0.121 0.149
C1878     0.169 0.156 0.181
C1883     0.243 0.230 0.257
C1888     0.398 0.382 0.414
C1893     0.575 0.556 0.595
C1898     0.759 0.736 0.782
C1903     0.931 0.906 0.958
C1913     0.980 0.954 1.007
C1918     1.029 1.000 1.058
C1923     1.085 1.053 1.117
C1928     1.032 0.997 1.068
C1933     1.029 0.987 1.073
C1938     0.895 0.846 0.946
C1943     0.864 0.802 0.930
C1948     1.017 0.914 1.131
C1953     0.980 0.789 1.217
```

The `A` parameters (as output by `ci.exp`) are now the age-specific rates in the 1908 cohort, and the `C` parameters are the rate-ratios relative to the 1908 birth cohort.

17. The 1908 birth cohort is for example represented in the period 1968 and age 60, that is persons at risk in the period 1968-01-01 through 1972-12-31 while between their 60th and 65th birthday. So the earliest born in that range are those that just manage 1 day before their 65th birthday in the period, that is persons born 1903-01-01. The latest born are those that just manage to have their 60th birthday at the last day of the period, that is those born 1912-12-31.

    Thus the persons included in the cohort labeled 1908 are born in the 10-year period from 1903-01-01 to 1912-12-31.

    But also note that the persons in the 1908 cohort are also either represented in the cohort labeled 1903 or the cohort labeled 1913. Hence sometimes these cohorst are called "synthetic cohorts".

18. In order to extract the cohort-specific rate-ratio parameters we use the same machinery as for the period-RRs; note that the possibility of supplying two data frames only works for models specified without too many bells and whistles:

```
ndc <- data.frame(C = seq(1858, 1953, 5))
ndr <- data.frame(C = 1908)
try(RR.C <- ci.exp(ac.r, list(ndc, ndr)))
   (RR.C <- ci.exp(ac.0, list(ndc, ndr)))
```

```
    exp(Est.)       2.5%     97.5%
1  0.05876855 0.02796332 0.1235097
2  0.05938629 0.04146988 0.0850432
3  0.09820451 0.08277938 0.1165040
```

```
4  0.13435012 0.12110391 0.1490452
5  0.16850582 0.15647290 0.1814641
6  0.24290000 0.22987080 0.2566677
7  0.39765267 0.38150319 0.4144858
8  0.57498146 0.55558344 0.5950568
9  0.75865134 0.73613440 0.7818570
10 0.93146302 0.90603145 0.9576084
11 1.00000000 1.00000000 1.0000000
12 0.98015018 0.95413844 1.0068711
13 1.02853256 1.00032663 1.0575338
14 1.08476601 1.05335625 1.1171124
15 1.03180855 0.99700215 1.0678301
16 1.02941676 0.98736791 1.0732563
17 0.89472043 0.84629743 0.9459141
18 0.86367228 0.80177930 0.9303431
19 1.01698726 0.91442286 1.1310556
20 0.98016430 0.78931893 1.2171532
```

We can then plot these against the cohort:

```
matshade(ndc$C, RR.C, log = 'y', plot = TRUE,
          xlab = "Date of birth", ylab = "Lung cancer incidence RR")
abline(h = 1, v = 1908, lty = 3)
```

19. The age-specific rates for the 1908 cohort we get from `ci.pred`:

```
ai.coh <- ci.pred(ac.0, data.frame(A = factor(seq(40, 85, 5)),
                                   C = '1908'))
```

We can then plot these, and at the same time include the age-specific rates from the age-period model:

```
matshade(seq(40, 85, 5), ai.coh, log = "y", plot = TRUE)
```

Since the rates of lung cancer are increasing by calendar time it follows that the longitudinal rates have a steeper slope by age than the cross-sectional. If there were a general *de*crease in rates by calendar time, the logituninal curves would be flatter than the cross-sectional.

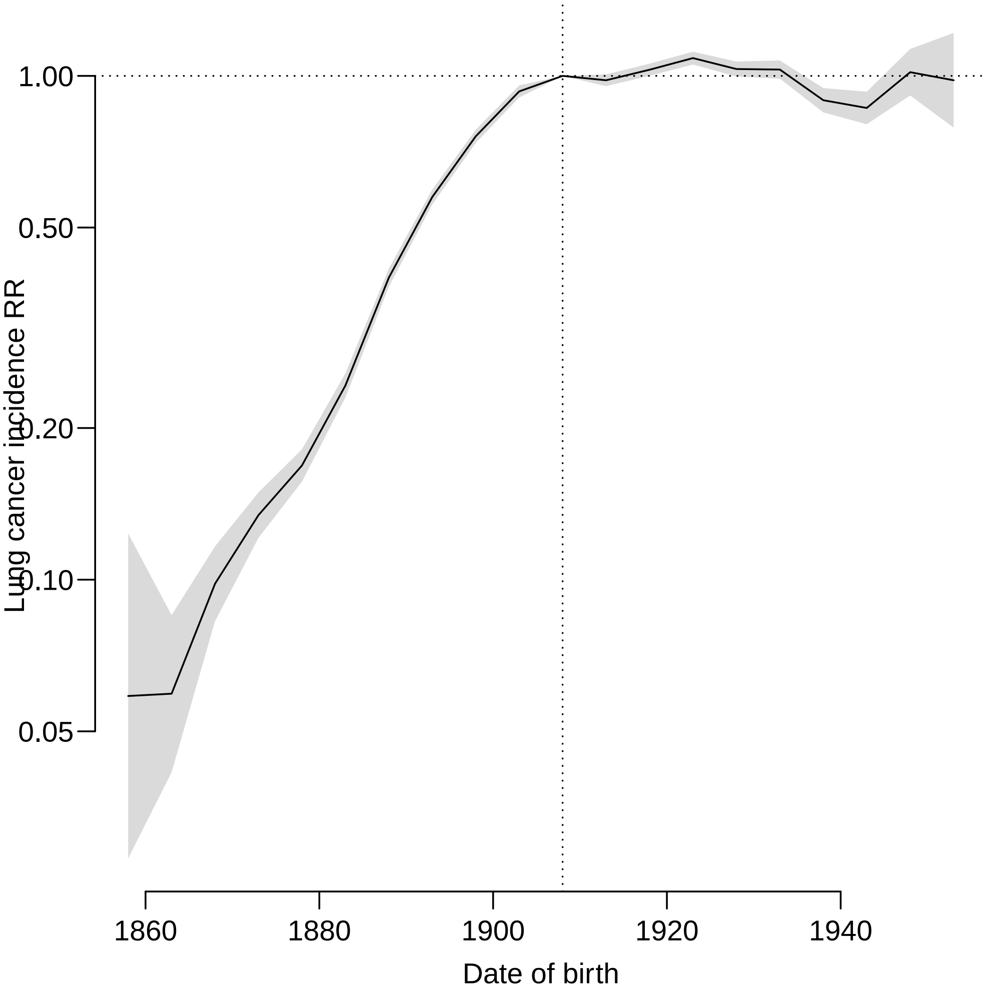

Figure 2.3: *Cohort RR of lung cancer relative to the 1908 cohort.* `../graph/AP-AC-cohRR`

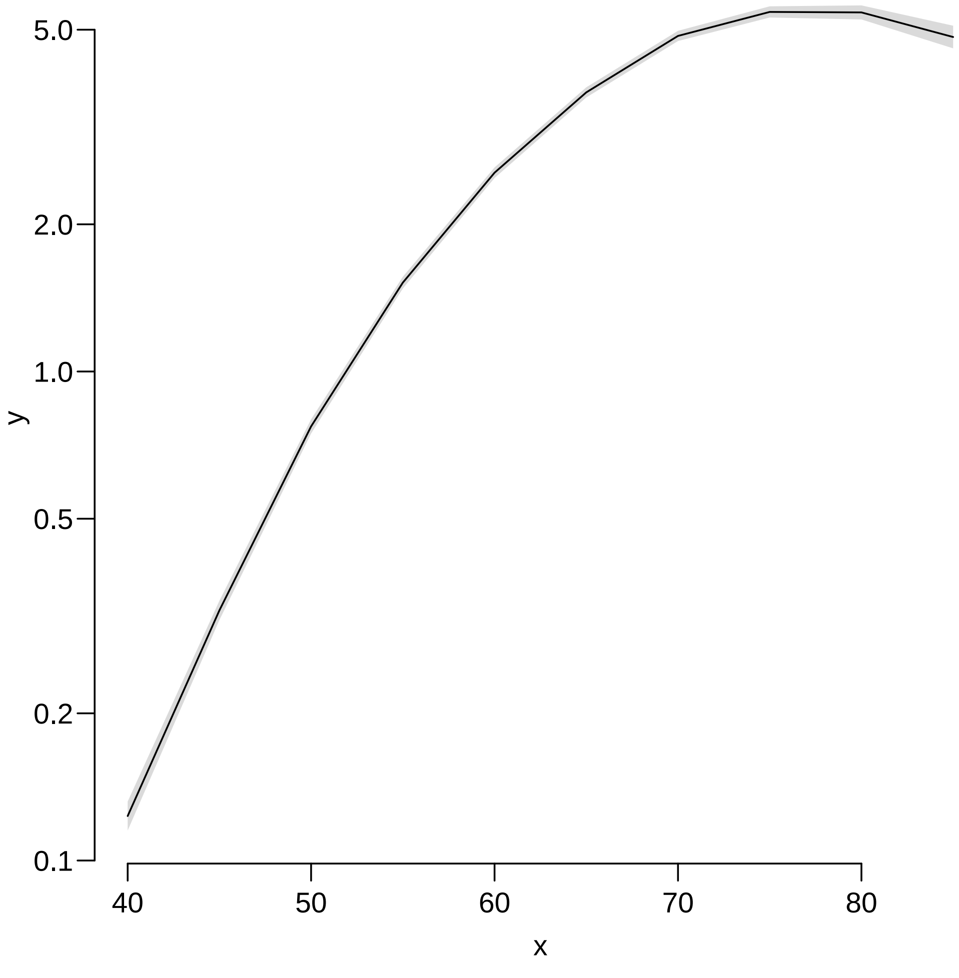

Figure 2.4: *Age-specific rates of male lung cancer in Denmark.* `../graph/AP-AC-Aincmp`

2.3 Age-drift model

This exercise is aimed at introducing the age-drift model and make you familiar with the two different ways of parametrizing this model.

1. First we read the data in the file `lung5-M.txt` and create the cohort variable:

```
lung <- read.table("http://bendixcarstensen.com/APC/KEA-2023/data/lung5-M.txt",
                   header = T)
# lung <- read.table( "../data/lung5-M.txt", header=T )
lung$C <- lung$P - lung$A
table(lung$C)
```

```
1858 1863 1868 1873 1878 1883 1888 1893 1898 1903 1908 1913 1918 1923 1928 1933 1938 1943
   1    2    3    4    5    6    7    8    9   10   10    9    8    7    6    5    4    3
1948 1953
   2    1
```

2. We fit the model to have age-parameters that refer to the period 1968–72. The midpoint of this period is 1970.5, but the periods are coded by their left endpoint, so we need to enter the value which makes the period 1968–72 appear as 0 in the modelling, in this case 1968:

```
mp <- glm(cbind(D, Y / 1000) ~ -1 + factor(A) + I(P - 1968),
          family = poisreg,
            data = lung )
round(ci.lin(mp), 4)
```

```
            Estimate StdErr        z P    2.5%  97.5%
factor(A)40  -2.2015 0.0310 -71.0282 0 -2.2622 -2.1407
factor(A)45  -1.2518 0.0199 -63.0322 0 -1.2907 -1.2129
factor(A)50  -0.4079 0.0137 -29.7038 0 -0.4349 -0.3810
factor(A)55   0.2390 0.0105  22.7735 0  0.2185  0.2596
factor(A)60   0.6932 0.0089  78.1006 0  0.6758  0.7106
factor(A)65   0.9794 0.0083 117.4868 0  0.9631  0.9958
factor(A)70   1.1413 0.0087 131.4254 0  1.1243  1.1584
factor(A)75   1.1300 0.0105 107.7920 0  1.1094  1.1505
factor(A)80   0.9936 0.0148  67.1831 0  0.9647  1.0226
factor(A)85   0.7290 0.0258  28.2214 0  0.6783  0.7796
I(P - 1968)   0.0233 0.0003  90.6987 0  0.0228  0.0238
```

The parameters now represent the log-rates in each of the age-classes in the period 1968–72. The period-parameter is the the average annual change in log-rates, in this case an increase of 2.3% / year.

However it would be more natural to have the coding of the age and period variables by the midpoint of the intervals. And notthere is nothing that requires thatfactor levels must be integers:

```
lung <- transform(lung, A = A + 2.5,
                        P = P + 2.5)
mp <- glm(cbind(D, Y / 1000) ~ -1 + factor(A) + I(P-1970.5),
          family = poisreg,
            data = lung )
ci.lin(mp)[,1:2]
```

```
                  Estimate       StdErr
factor(A)42.5 -2.2014942 0.0309946664
factor(A)47.5 -1.2517777 0.0198593498
factor(A)52.5 -0.4079411 0.0137336208
factor(A)57.5  0.2390327 0.0104960826
factor(A)62.5  0.6931761 0.0088754212
factor(A)67.5  0.9794432 0.0083366224
factor(A)72.5  1.1413394 0.0086843112
factor(A)77.5  1.1299602 0.0104827795
factor(A)82.5  0.9936383 0.0147900075
factor(A)87.5  0.7289606 0.0258301025
I(P - 1970.5)  0.0233067 0.0002569684
```

```
ci.exp(mp)
```

```
               exp(Est.)      2.5%     97.5%
factor(A)42.5 0.1106377 0.1041167 0.1175671
factor(A)47.5 0.2859959 0.2750778 0.2973474
factor(A)52.5 0.6650180 0.6473562 0.6831617
factor(A)57.5 1.2700200 1.2441601 1.2964174
factor(A)62.5 2.0000579 1.9655667 2.0351543
factor(A)67.5 2.6629732 2.6198151 2.7068422
factor(A)72.5 3.1309592 3.0781184 3.1847072
factor(A)77.5 3.0955334 3.0325819 3.1597917
factor(A)82.5 2.7010438 2.6238703 2.7804873
factor(A)87.5 2.0729250 1.9705931 2.1805709
I(P - 1970.5) 1.0235804 1.0230650 1.0240961
```

3. We now fit the same model, but with cohort as the continuous variable, centered around 1908:

```
mc <- glm(cbind(D, Y / 1000) ~ -1 + factor(A) + I(C - 1908),
          family = poisreg,
            data = lung)
round(ci.exp(mc), 6)
```

```
               exp(Est.)     2.5%    97.5%
factor(A)42.5  0.069417 0.065235 0.073866
factor(A)47.5  0.201618 0.193656 0.209908
factor(A)52.5  0.526761 0.512241 0.541693
factor(A)57.5  1.130318 1.106733 1.154406
factor(A)62.5  2.000058 1.965567 2.035154
factor(A)67.5  2.992104 2.944658 3.040314
factor(A)72.5  3.952730 3.887625 4.018925
factor(A)77.5  4.391017 4.302400 4.481459
factor(A)82.5  4.304979 4.181384 4.432228
factor(A)87.5  3.712213 3.527885 3.906172
I(C - 1908)    1.023580 1.023065 1.024096
```

```
round(ci.exp(mp), 6)
```

```
               exp(Est.)     2.5%    97.5%
factor(A)42.5  0.110638 0.104117 0.117567
factor(A)47.5  0.285996 0.275078 0.297347
factor(A)52.5  0.665018 0.647356 0.683162
factor(A)57.5  1.270020 1.244160 1.296417
factor(A)62.5  2.000058 1.965567 2.035154
factor(A)67.5  2.662973 2.619815 2.706842
factor(A)72.5  3.130959 3.078118 3.184707
factor(A)77.5  3.095533 3.032582 3.159792
factor(A)82.5  2.701044 2.623870 2.780487
factor(A)87.5  2.072925 1.970593 2.180571
I(P - 1970.5)  1.023580 1.023065 1.024096
```

4. We see that the estimated slope (the drift!) is exactly the same as in the period-model, but the age-estimates are not.

   Moreover the two model `mp` and `mc` are really the same model just parametrized differently; the residual deviances are the same:

   ```
   oo <- options(digits = 9)
   c(deviance(mp), deviance(mc))
   ```

   ```
   [1] 6417.38106 6417.38106
   ```

   ```
   options(oo)
   ```

5. If we write how the cohort model is parametrized we have:

$$\begin{aligned} log(\lambda_{ap}) &= \alpha_a + \beta(c - 1908) \\ &= \alpha_a + \beta(p - a - 1908) \\ &= [\alpha_a + \beta(62.5 - a)] + \beta(p - 1970.5) \end{aligned}$$

   The expression in the square brackets are the age-parameters in the age-period model. Hence, the age parameters are linked by a simple linear relation, which is easily verified empirically:

   ```
   (ap <- ci.lin(mp)[1:10,1])
   ```

   ```
   factor(A)42.5 factor(A)47.5 factor(A)52.5 factor(A)57.5 factor(A)62.5 factor(A)67.5
      -2.2014942    -1.2517777    -0.4079411     0.2390327     0.6931761     0.9794432
   factor(A)72.5 factor(A)77.5 factor(A)82.5 factor(A)87.5
       1.1413394     1.1299602     0.9936383     0.7289606
   ```

   ```
   (ac <- ci.lin(mc)[1:10,1])
   ```

   ```
   factor(A)42.5 factor(A)47.5 factor(A)52.5 factor(A)57.5 factor(A)62.5 factor(A)67.5
      -2.6676283    -1.6013783    -0.6410082     0.1224992     0.6931761     1.0959767
   factor(A)72.5 factor(A)77.5 factor(A)82.5 factor(A)87.5
       1.3744064     1.4795608     1.4597724     1.3116282
   ```

   ```
   cbind(ap, ac, ap - ac, diff(ap - ac))
   ```

   ```
                         ap         ac
   factor(A)42.5 -2.2014942 -2.6676283  4.661340e-01 -0.1165335
   factor(A)47.5 -1.2517777 -1.6013783  3.496005e-01 -0.1165335
   factor(A)52.5 -0.4079411 -0.6410082  2.330670e-01 -0.1165335
   factor(A)57.5  0.2390327  0.1224992  1.165335e-01 -0.1165335
   factor(A)62.5  0.6931761  0.6931761 -3.330669e-16 -0.1165335
   factor(A)67.5  0.9794432  1.0959767 -1.165335e-01 -0.1165335
   factor(A)72.5  1.1413394  1.3744064 -2.330670e-01 -0.1165335
   factor(A)77.5  1.1299602  1.4795608 -3.496005e-01 -0.1165335
   factor(A)82.5  0.9936383  1.4597724 -4.661340e-01 -0.1165335
   factor(A)87.5  0.7289606  1.3116282 -5.826676e-01 -0.1165335
   ```

```
c.sl <- ci.lin(mc)[11,1]
a.pt <- seq(40, 85, 5) + 2.5
cbind( ap, ac + c.sl*(62.5-a.pt) )
```

```
                     ap
factor(A)42.5 -2.2014942 -2.2014942
factor(A)47.5 -1.2517777 -1.2517777
factor(A)52.5 -0.4079411 -0.4079411
factor(A)57.5  0.2390327  0.2390327
factor(A)62.5  0.6931761  0.6931761
factor(A)67.5  0.9794432  0.9794432
factor(A)72.5  1.1413394  1.1413394
factor(A)77.5  1.1299602  1.1299602
factor(A)82.5  0.9936383  0.9936383
factor(A)87.5  0.7289606  0.7289606
```

6.
```
matshade(a.pt, cbind(ci.exp(mp, subset = "A"),
                     ci.exp(mc, subset = "A")) * 10^5, plot = TRUE,
         log = "y", xlab = "Age", ylab = "Lung cancer incidence rates / 100,000",
         lty = 1, lwd = 1, col = c("black","blue") )
```

7. The relative risks are from the model:

$$\log(\lambda_{ap}) = \alpha_p + \delta(p - 1970.5)$$

Therefore, with an x-variable: (1943,...,1993) + 2.5, the relative risk will be:

$$\text{RR} = \hat{\delta} \times x$$

and the upper and lower confidence bands:

$$\text{RR} = (\hat{\delta} \pm 1.96 \times \text{s.e.}(\delta)) \times x$$

We can find the estimated RRs with confidence intervals using a suitable 1-column contrast matrix. We of course need a separate one for period and cohort since these cover different time-spans:

```
p.pt <- seq(min(lung$P), max(lung$P), , 10) + 2.5
c.pt <- seq(min(lung$C), max(lung$C), , 10)
ctr.p <- cbind(p.pt - 1970.5)
ctr.c <- cbind(c.pt - 1908  )
matshade(c.pt, ci.exp(mc, subset = "C", ctr.mat = ctr.c), plot = TRUE,
         log = "y", xlab = "Calendar time", ylab = "Rate ratio", xlim = c(1850,2000),
         type = "l", lty = 1, lwd = 1, col = "blue" )
matshade(p.pt, ci.exp(mp, subset = "P", ctr.mat = ctr.p),
         type = "l", lty = 1, lwd = 1, col = "black" )
abline(h = 1, lty = 3)
points(c(1908, 1970.5), c(1, 1), pch = 16)
```

The effect of time (the drift) is the same for the two parametrizations, but the age-specific rates refer either to cross-sectional rates (from the model with period drift) or longitudinal rates (from the model with cohort drift).

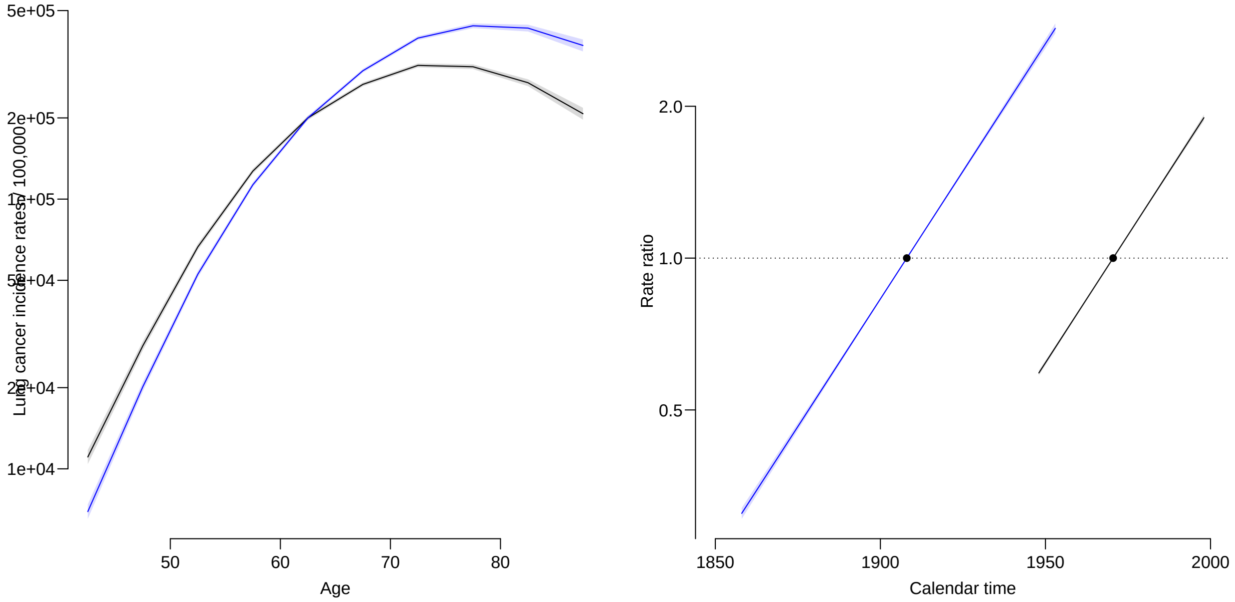

Figure 2.5: *Age-specific rates from the age-drift model (left) and the rate-ratios as estimated under the two different parametrizations.*

2.4 Age-period-cohort model

We will need the results from the age-period, the age-cohort and the age-drift models in this exercise so we briefly fit these models after we have read data.

1. Read the data in the file `lung5-M.txt` as in the previous exercises, recode age and perido to midpoints and fit the three models we discussed so far:

```
lung <- read.table("http://bendixcarstensen.com/APC/KEA-2023/data/lung5-M.txt",
                   header = T)
# lung <- read.table( "../data/lung5-M.txt", header=T )
lung <- transform(lung, A = A + 2.5,
                        P = P + 2.5)
str(lung)
```

```
'data.frame':        110 obs. of  4 variables:
 $ A: num  42.5 42.5 42.5 42.5 42.5 42.5 42.5 42.5 42.5 42.5 ...
 $ P: num  1946 1950 1956 1960 1966 ...
 $ D: int  80 81 73 99 82 97 86 90 116 149 ...
 $ Y: num  694047 754770 769441 749265 757240 ...
```

```
head(lung)
```

```
     A      P  D        Y
1 42.5 1945.5 80 694046.5
2 42.5 1950.5 81 754769.5
3 42.5 1955.5 73 769440.7
4 42.5 1960.5 99 749264.5
5 42.5 1965.5 82 757240.0
6 42.5 1970.5 97 709558.5
```

```
m.AP <- glm(cbind(D, Y / 1000) ~ factor(A) + factor(P),
            family = poisreg,
              data = lung)
m.AC <- glm(cbind(D, Y / 1000) ~ factor(A) + factor(P-A),
            family = poisreg,
              data = lung )
m.Ad <- glm(cbind(D, Y / 1000) ~ factor(A) + P,
            family = poisreg,
              data = lung )
```

2. We then fit the age-period-cohort model. Note that there is no such variable as the cohort in the dataset; we have to compute this as $P - A$. This is best done on the fly instead of cluttering up the data frame with another variable. In the same go we fit the simplest model with age alone:

```
m.APC <- glm(cbind(D, Y / 1000) ~ factor(A) + factor(P) + factor(P-A),
             family = poisreg,
               data = lung )
m.A   <- glm(cbind(D, Y / 1000) ~ factor(A),
             family = poisreg,
               data = lung )
```

3. We can use `anova.glm` to test the different models in a sequence that gives all the valid comparisons:

```
anova(m.A, m.Ad, m.AP, m.APC, m.AC, m.Ad, test = "Chisq")

Analysis of Deviance Table

Model 1: cbind(D, Y/1000) ~ factor(A)
Model 2: cbind(D, Y/1000) ~ factor(A) + P
Model 3: cbind(D, Y/1000) ~ factor(A) + factor(P)
Model 4: cbind(D, Y/1000) ~ factor(A) + factor(P) + factor(P - A)
Model 5: cbind(D, Y/1000) ~ factor(A) + factor(P - A)
Model 6: cbind(D, Y/1000) ~ factor(A) + P
  Resid. Df Resid. Dev  Df Deviance  Pr(>Chi)
1       100    15103.0
2        99     6417.4   1   8685.6 < 2.2e-16 ***
3        90     2723.5   9   3693.9 < 2.2e-16 ***
4        72      208.5  18   2514.9 < 2.2e-16 ***
5        81      829.6  -9   -621.1 < 2.2e-16 ***
6        99     6417.4 -18  -5587.8 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The successive tests refer to:

(a) linear effect of period/cohort

(b) non-linear effect of period

(c) non-linear effect of cohort (in the presence of period)

(d) non-linear effect of period (in the presence of cohort)

(e) non-linear effect of cohort

Clearly, with the large amounts of data that we are dealing with, all of the tests are strongly significant, but comparing the likelihood ratio statistics there is some indication that the period curvature (non-linear component) is stronger than the cohort one.

4. When we look at the parameters from the APC-model we seet that one parameter is aliased; the last one:

```
summary(m.APC)

Call:
glm(formula = cbind(D, Y/1000) ~ factor(A) + factor(P) + factor(P -
    A), family = poisreg, data = lung)

Coefficients: (1 not defined because of singularities)
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       -3.66465    0.20670 -17.729  < 2e-16 ***
factor(A)47.5      0.83902    0.04204  19.959  < 2e-16 ***
factor(A)52.5      1.56343    0.05286  29.575  < 2e-16 ***
factor(A)57.5      2.09418    0.06914  30.287  < 2e-16 ***
factor(A)62.5      2.46656    0.08757  28.166  < 2e-16 ***
factor(A)67.5      2.70325    0.10697  25.271  < 2e-16 ***
factor(A)72.5      2.82959    0.12688  22.302  < 2e-16 ***
factor(A)77.5      2.80263    0.14713  19.048  < 2e-16 ***
factor(A)82.5      2.66268    0.16774  15.874  < 2e-16 ***
factor(A)87.5      2.40553    0.18764  12.820  < 2e-16 ***
factor(P)1950.5    0.25057    0.04115   6.090 1.13e-09 ***
factor(P)1955.5    0.41507    0.05276   7.867 3.63e-15 ***
factor(P)1960.5    0.66570    0.06907   9.638  < 2e-16 ***
factor(P)1965.5    0.86971    0.08757   9.931  < 2e-16 ***
factor(P)1970.5    1.08681    0.10698  10.159  < 2e-16 ***
factor(P)1975.5    1.22682    0.12689   9.669  < 2e-16 ***
factor(P)1980.5    1.38050    0.14705   9.388  < 2e-16 ***
factor(P)1985.5    1.49023    0.16740   8.902  < 2e-16 ***
factor(P)1990.5    1.49948    0.18786   7.982 1.44e-15 ***
factor(P)1995.5    1.55151    0.20810   7.456 8.94e-14 ***
factor(P - A)1863 -0.16176    0.40257  -0.402  0.68781
factor(P - A)1868  0.18003    0.35012   0.514  0.60712
factor(P - A)1873  0.32401    0.32402   1.000  0.31732
factor(P - A)1878  0.37418    0.30269   1.236  0.21639
factor(P - A)1883  0.55427    0.28250   1.962  0.04976 *
factor(P - A)1888  0.85582    0.26291   3.255  0.00113 **
factor(P - A)1893  1.03312    0.24387   4.236 2.27e-05 ***
factor(P - A)1898  1.12663    0.22526   5.001 5.69e-07 ***
factor(P - A)1903  1.16461    0.20711   5.623 1.87e-08 ***
factor(P - A)1908  1.08855    0.18955   5.743 9.31e-09 ***
factor(P - A)1913  0.93886    0.17277   5.434 5.51e-08 ***
factor(P - A)1918  0.86676    0.15692   5.524 3.32e-08 ***
factor(P - A)1923  0.80275    0.14237   5.639 1.72e-08 ***
factor(P - A)1928  0.63395    0.12973   4.887 1.03e-06 ***
factor(P - A)1933  0.51050    0.11962   4.268 1.97e-05 ***
factor(P - A)1938  0.24666    0.11320   2.179  0.02934 *
factor(P - A)1943  0.08854    0.11116   0.796  0.42575
factor(P - A)1948  0.14073    0.11556   1.218  0.22329
factor(P - A)1953       NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 71776.18  on 109  degrees of freedom
Residual deviance:   208.55  on  72  degrees of freedom
AIC: 1141.6

Number of Fisher Scoring iterations: 5
```

```
ci.exp(m.APC)
```

```
                    exp(Est.)        2.5%      97.5%
(Intercept)        0.02561318  0.01708117  0.0384069
factor(A)47.5      2.31409902  2.13108464  2.5128304
factor(A)52.5      4.77515585  4.30516704  5.2964526
factor(A)57.5      8.11876406  7.08979634  9.2970696
factor(A)62.5     11.78179739  9.92363503 13.9878935
factor(A)67.5     14.92814389 12.10461923 18.4102842
factor(A)72.5     16.93853500 13.20922592 21.7207253
factor(A)77.5     16.48797782 12.35744806 21.9991548
factor(A)82.5     14.33459300 10.31818227 19.9144143
factor(A)87.5     11.08431809  7.67348914 16.0112441
factor(P)1950.5    1.28476394  1.18521914  1.3926694
factor(P)1955.5    1.51448060  1.36569791  1.6794721
factor(P)1960.5    1.94585380  1.69947908  2.2279456
factor(P)1965.5    2.38621591  2.00986288  2.8330422
factor(P)1970.5    2.96480996  2.40401569  3.6564229
factor(P)1975.5    3.41035301  2.65945333  4.3732701
factor(P)1980.5    3.97687760  2.98106505  5.3053373
factor(P)1985.5    4.43812674  3.19677614  6.1615103
factor(P)1990.5    4.47936443  3.09961981  6.4732796
factor(P)1995.5    4.71858424  3.13818856  7.0948691
factor(P - A)1863  0.85064170  0.38643341  1.8724864
factor(P - A)1868  1.19725146  0.60278035  2.3779990
factor(P - A)1873  1.38266771  0.73267615  2.6092974
factor(P - A)1878  1.45380465  0.80325598  2.6312259
factor(P - A)1883  1.74067729  1.00057899  3.0282041
factor(P - A)1888  2.35329687  1.40569073  3.9397046
factor(P - A)1893  2.80980871  1.74217725  4.5317002
factor(P - A)1898  3.08524022  1.98400719  4.7977181
factor(P - A)1903  3.20467871  2.13546688  4.8092367
factor(P - A)1908  2.96996935  2.04836292  4.3062281
factor(P - A)1913  2.55705968  1.82253900  3.5876073
factor(P - A)1918  2.37919865  1.74928376  3.2359451
factor(P - A)1923  2.23166844  1.68828253  2.9499470
factor(P - A)1928  1.88503396  1.46180822  2.4307929
factor(P - A)1933  1.66611805  1.31791922  2.1063122
factor(P - A)1938  1.27974254  1.02509758  1.5976440
factor(P - A)1943  1.09257743  0.87867975  1.3585444
factor(P - A)1948  1.15110993  0.91781647  1.4437026
factor(P - A)1953  1.00000000  1.00000000  1.0000000
```

This is because the linear effect (the drift) is included both in `P` and in `C + A`.

It is a bad idea to use the extreme cohorts as reference points; these parameters are based on one point only; better to fix the two extreme periods.

5. When we want to fit models where some of the factor levels are merged or sorted as the first one, we use the `Relevel` function to do this (remember to read the help page for `Relevel`, which is not the same as `relevel`):

```
lung$Pr <- Relevel(factor(lung$P), list("first & last"=c("1945.5", "1995.5")))
lung$Cr <- Relevel(factor(lung$P - lung$A), "1908")
```

We of course check that the results of these operations are as we would like them to be:

```
with(lung, table(P  , Pr))
```

```
        Pr
P        first & last 1950.5 1955.5 1960.5 1965.5 1970.5 1975.5 1980.5 1985.5 1990.5
  1945.5           10      0      0      0      0      0      0      0      0      0
  1950.5            0     10      0      0      0      0      0      0      0      0
  1955.5            0      0     10      0      0      0      0      0      0      0
  1960.5            0      0      0     10      0      0      0      0      0      0
  1965.5            0      0      0      0     10      0      0      0      0      0
  1970.5            0      0      0      0      0     10      0      0      0      0
  1975.5            0      0      0      0      0      0     10      0      0      0
  1980.5            0      0      0      0      0      0      0     10      0      0
  1985.5            0      0      0      0      0      0      0      0     10      0
  1990.5            0      0      0      0      0      0      0      0      0     10
  1995.5           10      0      0      0      0      0      0      0      0      0
```

```
with(lung, table(P-A, Cr))
```

```
    Cr
     1908 1858 1863 1868 1873 1878 1883 1888 1893 1898 1903 1913 1918 1923 1928 1933
1858    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0
1863    0    0    2    0    0    0    0    0    0    0    0    0    0    0    0    0
1868    0    0    0    3    0    0    0    0    0    0    0    0    0    0    0    0
1873    0    0    0    0    4    0    0    0    0    0    0    0    0    0    0    0
1878    0    0    0    0    0    5    0    0    0    0    0    0    0    0    0    0
1883    0    0    0    0    0    0    6    0    0    0    0    0    0    0    0    0
1888    0    0    0    0    0    0    0    7    0    0    0    0    0    0    0    0
1893    0    0    0    0    0    0    0    0    8    0    0    0    0    0    0    0
1898    0    0    0    0    0    0    0    0    0    9    0    0    0    0    0    0
1903    0    0    0    0    0    0    0    0    0    0   10    0    0    0    0    0
1908   10    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
1913    0    0    0    0    0    0    0    0    0    0    0    9    0    0    0    0
1918    0    0    0    0    0    0    0    0    0    0    0    0    8    0    0    0
1923    0    0    0    0    0    0    0    0    0    0    0    0    0    7    0    0
1928    0    0    0    0    0    0    0    0    0    0    0    0    0    0    6    0
1933    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    5
1938    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
1943    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
1948    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
1953    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    Cr
     1938 1943 1948 1953
1858    0    0    0    0
1863    0    0    0    0
1868    0    0    0    0
1873    0    0    0    0
1878    0    0    0    0
1883    0    0    0    0
1888    0    0    0    0
1893    0    0    0    0
1898    0    0    0    0
1903    0    0    0    0
1908    0    0    0    0
1913    0    0    0    0
1918    0    0    0    0
1923    0    0    0    0
1928    0    0    0    0
1933    0    0    0    0
1938    4    0    0    0
1943    0    3    0    0
1948    0    0    2    0
1953    0    0    0    1
```

6. We can now fit the models with the recoded factors:

```
m.APC1 <- glm(cbind(D, Y / 1000) ~ -1 + factor(A) + factor(Pr) + factor(Cr),
              family = poisreg,
                data = lung)
coef( m.APC1 )
```

```
   factor(A)42.5    factor(A)47.5    factor(A)52.5    factor(A)57.5    factor(A)62.5
    -2.420945836     -1.426774537     -0.547217464      0.138684738      0.666213432
   factor(A)67.5    factor(A)72.5    factor(A)77.5    factor(A)82.5    factor(A)87.5
     1.058056849      1.339550651      1.467741826      1.482936915      1.380943413
factor(Pr)1950.5 factor(Pr)1955.5 factor(Pr)1960.5 factor(Pr)1965.5 factor(Pr)1970.5
```

```
     0.095424116      0.104770778      0.200248212      0.249105289      0.311058535
factor(Pr)1975.5 factor(Pr)1980.5 factor(Pr)1985.5 factor(Pr)1990.5   factor(Cr)1858
     0.295910526      0.294440825      0.249025339      0.103123244     -2.640060438
  factor(Cr)1863   factor(Cr)1868   factor(Cr)1873   factor(Cr)1878   factor(Cr)1883
    -2.646673834     -2.149730193     -1.850593043     -1.645272902     -1.310031751
  factor(Cr)1888   factor(Cr)1893   factor(Cr)1898   factor(Cr)1903   factor(Cr)1913
    -0.853337885     -0.520887869     -0.272223872     -0.079090672      0.005457283
  factor(Cr)1918   factor(Cr)1923   factor(Cr)1928   factor(Cr)1933   factor(Cr)1938
     0.088513857      0.179650494      0.165997726      0.197699170      0.089012570
  factor(Cr)1943   factor(Cr)1948   factor(Cr)1953
     0.086044048      0.293382042      0.307806293
```

The age-coefficients are log-rates (where the rates are in units person-year^{-1}), the cohort parameters are log-rate-ratios relative to a trend from the first to the last period.

7. We can use `ci.exp` to extract the parameters with confidence limits from this model:

```
A.eff <- ci.exp(m.APC1, subset = "A")
P.eff <- rbind(c(1,1,1),
               ci.exp( m.APC1, subset="P" ),
               c(1,1,1))
(C.ref <- match("1908", levels(with(lung, factor(P - A)))))
```

```
[1] 11
```

```
(C.nlv <- nlevels(with(lung,factor(P-A))))
```

```
[1] 20
```

```
C.eff <- rbind(ci.exp(m.APC1, subset="C")[1:10,],
               c(1,1,1),
               ci.exp(m.APC1, subset="C")[11:19,] )
```

In order to plot these we need the time points on the respective scales:

```
A.pt <- sort(unique(lung$A) )
P.pt <- sort(unique(lung$P) )
C.pt <- sort(unique(lung$P - lung$A ))
```

Then we can plot the estimated effects

```
par(mfrow = c(1, 3), las = 1)
matshade(A.pt, A.eff, plot = TRUE,
         xlab = "Age", ylab = "Lung cancer rate per 1000 PY", log="y" )
matshade(P.pt, P.eff, plot = TRUE,
         xlab = "Period", ylab = "RR", log="y")
abline(h = 1, lty = 3)
matshade(C.pt, C.eff, plot=TRUE,
         xlab = "Cohort", ylab = "RR", log = "y")
abline(h = 1, v = 1908, lty = 3)
```

This is is not a particularly informative plot, as the scales are all different—the rates are between 10^{-4} and 5×10^{-3}, whereas the cohort RRs are between 0.05 and slightly more than 1. So if we rescale the rate to rates per 1000, and then demand that all display have y-axis from 0.05 to 5, we get comparable displays:

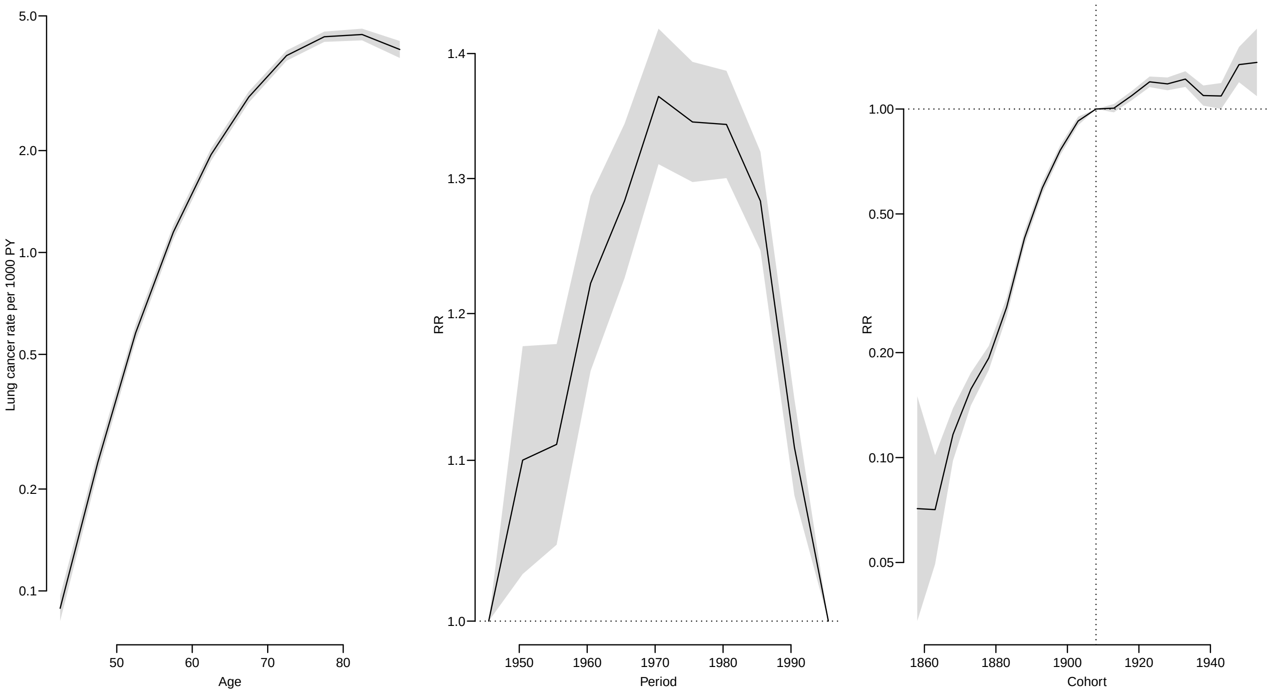

Figure 2.6: *Estimates of the age-period-cohort model effects — with first and last period as reference and cohort 1908 as reference.*

```
par(mfrow = c(1, 3), las = 1)
matshade(A.pt, A.eff, plot=TRUE,
         xlab="Age", ylab="Rates", ylim=c(0.1,4), log="y"  )
matshade(P.pt, P.eff, plot=TRUE,
         xlab="Period", ylab="RR", ylim=c(0.1,4)/2, log="y"  )
abline(h = 1)
matshade(C.pt, C.eff, plot=TRUE,
         xlab="Cohort", ylab="RR", ylim=c(0.1,4)/2, log="y"  )
abline( h=1 )
```

The parameters in this model represent age-specific rates, that approximates the rates in the 1980 cohort (as predicted. . .), cohort RRs relative to this cohort, and finally period "residual" RRs.

But note an explicit decision has been made as to how the period residuals are defined; namely as the deviations from the line between the periods 1943 and 1993.

8. We now fit the model with two cohorts aliased and one period as fixpoint. To decide which of the cohort to alias (and define as the first level of the factor) we tabulate no of observations and no of cases

```
with(lung, table(P - A))
```

```
1858 1863 1868 1873 1878 1883 1888 1893 1898 1903 1908 1913 1918 1923 1928 1933 1938 1943
   1    2    3    4    5    6    7    8    9   10   10    9    8    7    6    5    4    3
1948 1953
   2    1
```

```
with(lung, tapply(D, list(P - A), sum))
```

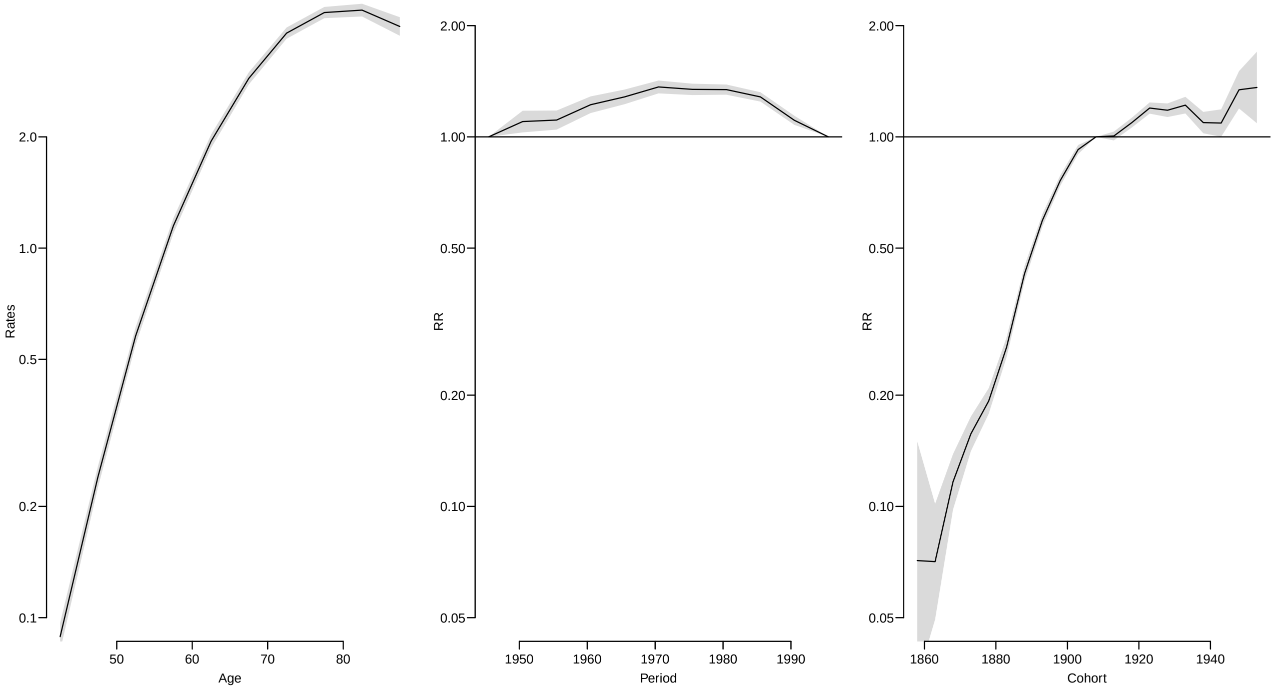

Figure 2.7: *Estimates of the age-period-cohort model estimates, scaled displays.*

```
 1858  1863  1868  1873  1878  1883  1888  1893  1898  1903  1908  1913  1918  1923  1928
    7    30   134   371   752  1436  2822  4668  6934  9305 10873 10468  9438  8010  5040
 1933  1938  1943  1948  1953
 3036  1536   827   400    91
```

Rather arbitrarily we decide on cohorts 1878 and 1933; the numbers of these in the cohort numbers are computed by:

```
 (C.ref.pos <- with(lung, match(c("1878", "1933"), levels(factor(P - A)))))
```

```
[1]  5 16
```

```
 (P.ref.pos <- with(lung, match("1975.5", levels(factor(P)))))
```

```
[1] 7
```

```
 lung$Cx <- Relevel(factor(lung$P - lung$A), list("refCoh"=c("1878", "1933")))
 lung$Px <- Relevel(factor(lung$P), "1975.5" )
```

With these definitions we can now fit the model with the alternative parametrization:

```
 m.APC2 <- glm(cbind(D, Y / 1000) ~ -1 + factor(A) + factor(Px) + factor(Cx),
               family = poisreg,
                 data = lung)
```

We note that it is only the parametrization that differs; the fitted model is the same:

```
c(deviance(m.APC ),
  deviance(m.APC1),
  deviance(m.APC2))
```

```
[1] 208.5476 208.5476 208.5476
```

9. We use the same points for the age, period and cohort as before, but now extract the parameters in a slightly different way:

```
A.Eff <- ci.exp(m.APC2, subset = "A")
P.Eff <- ci.exp(m.APC2, subset = "P")
nP <- nrow(P.Eff)
P.Eff <- rbind(P.Eff[1:(P.ref.pos-1),],
               c(1,1,1),
               P.Eff[P.ref.pos:nP,])
C.Eff <- ci.exp(m.APC2, subset = "C")
nC <- nrow(C.Eff)
C.Eff <- rbind(C.Eff[1:(C.ref.pos[1]-1),],
               c(1,1,1),
               C.Eff[(C.ref.pos[1]):(C.ref.pos[2]-2),],
               c(1,1,1),
               C.Eff[(C.ref.pos[2]-1):nC,] )
```

We can now plot the two sets of parameters in the same plots:

```
par(mfrow = c(1, 3), las = 2, mar = c(4, 3, 0.5, 0.5), mgp = c(3, 1, 0) / 1.6)
matshade(A.pt, cbind(A.eff, A.Eff), plot = TRUE,
         xlab = "Age", ylab = "Rates", ylim = c(0.1, 4),
         log = "y", col = c("black", "blue"))
matshade(P.pt, cbind(P.eff, P.Eff), plot = TRUE,
         xlab = "Period", ylab = "RR", ylim = c(0.1, 4)/2,
         log = "y", col = c("black", "blue"))
abline(h = 1)
points(c(1943, 1993, 1973)+2.5, rep(1, 3), pch = 16, col = c("black", "blue")[c(1, 1, 2)]
matshade(C.pt, cbind(C.eff, C.Eff), plot = TRUE,
          xlab = "Cohort", ylab = "RR", ylim = c(0.1, 4)/2,
          log = "y", col = c("black", "blue"))
points(c(1878, 1933, 1908), rep(1, 3), pch = 16, col = c("black", "blue")[c(2, 2, 1)])
abline(h = 1)
```

It is clear from the estimates that very different displays can be obtained from different parametrizations. So something more interpretable may be needed. . .

10. A more credible parametrization of the APC-model can be obtained using the `apc.fit` function from the `Epi` package. It offers different *parametrizations* of different *models*. One possible model to use is the one we just fitted namely the model with one parameter per level of age, period and cohort (using `model = 'factor'`). Additional to this we must specifiy the *principle* of parametrization:

    - `"ACP"` gives age-specific rates, cohort specific rate ratios relative to cohort `ref.c`, and period specific rate-ratio residuals, constrained to have 0 slope on average and 0 on average.

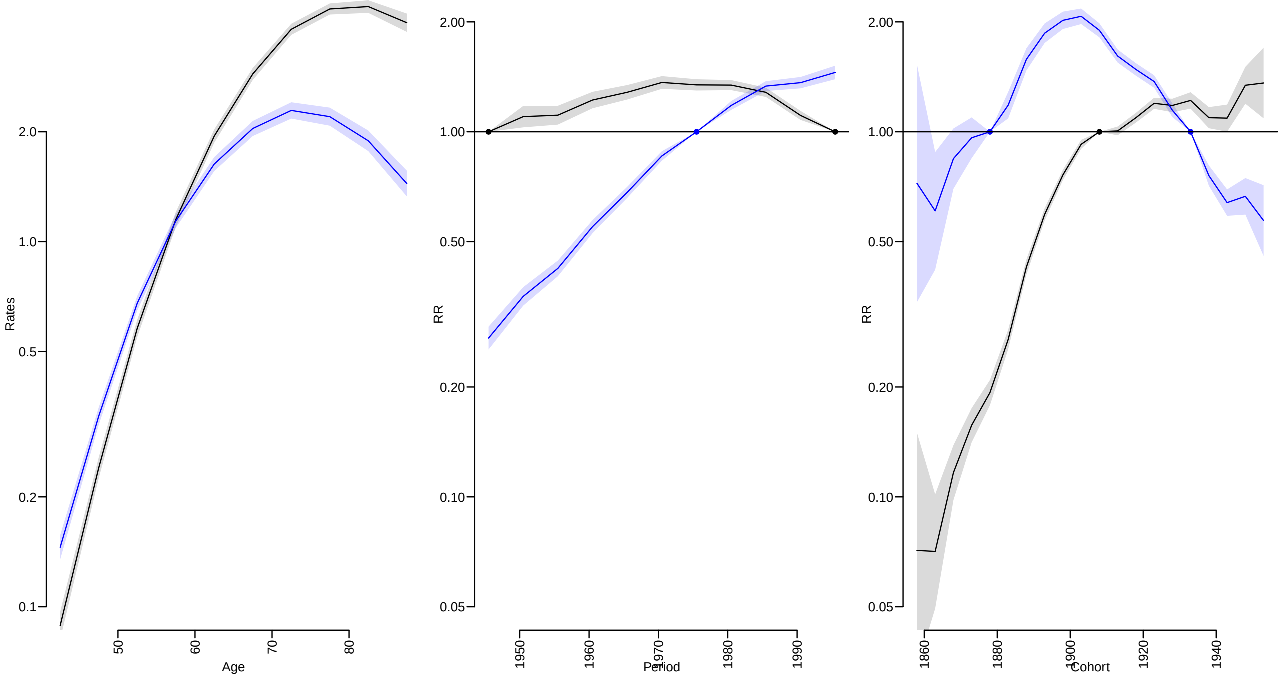

Figure 2.8: *Estimates of the age-period-cohort model estimates, from the two different parametrizations.* ../graph/APC-parm3

- "APC" gives age-specific rates, period specific rate ratios relative to period ref.p, and cohort specific rate-ratio residuals, constrained to have 0 slope on average and 0 on average.

The paramtrization is dependent on what we mean by "0 slope on average and 0 on average". In essence, this boils down to choosing a definition of orthogonality — essentially an inner product in the observation space, as explained in the lectures.

The default is to choose an inner product that weighs observations according to the amount of person-years in each unit of observation, proportional to the observed information about the log-rate in each (minus the 2^{nd} derivative of the log-likelihood w.r.t. the log-rate.)

Now fit the factor model with two different parametrizations:

```
 f.cp <- apc.fit(lung, model = "factor", parm = "ACP", ref.c = 1908, scale = 1000)
[1] "ML of APC-model Poisson with log(Y) offset : ( ACP ):\n"
             Model       AIC Mod. df.  Mod. dev. Test df. Test dev.      Pr(>Chi)
1              Age 15980.060      100 15103.0012       NA        NA            NA
2        Age-drift  7296.440       99  6417.3811        1 8685.6201  0.000000e+00
3       Age-Cohort  1744.688       81   829.6293       18 5587.7517  0.000000e+00
4 Age-Period-Cohort  1141.606      72   208.5476        9  621.0817 6.244585e-128
5       Age-Period  3620.525       90  2723.4660       18 2514.9183  0.000000e+00
6        Age-drift  7296.440       99  6417.3811        9 3693.9151  0.000000e+00
  Test dev/df     H0
1          NA
2  8685.62013 zero drift
3   310.43065 Coh eff|dr.
4    69.00908 Per eff|Coh
```

```
5   139.71769 Coh eff|Per
6   410.43501 Per eff|dr.
```

```
f.pc <- apc.fit(lung, model = "factor", parm = "APC", ref.p = 1968, scale = 1000)
```

```
[1] "ML of APC-model Poisson with log(Y) offset : ( APC ):\n"
              Model       AIC Mod. df.  Mod. dev. Test df. Test dev.      Pr(>Chi)
1               Age 15980.060      100 15103.0012       NA        NA            NA
2         Age-drift  7296.440       99  6417.3811        1 8685.6201  0.000000e+00
3        Age-Cohort  1744.688       81   829.6293       18 5587.7517  0.000000e+00
4 Age-Period-Cohort  1141.606       72   208.5476        9  621.0817 6.244585e-128
5        Age-Period  3620.525       90  2723.4660       18 2514.9183  0.000000e+00
6         Age-drift  7296.440       99  6417.3811        9 3693.9151  0.000000e+00
  Test dev/df     H0
1          NA
2  8685.62013 zero drift
3   310.43065 Coh eff|dr.
4    69.00908 Per eff|Coh
5   139.71769 Coh eff|Per
6   410.43501 Per eff|dr.
```

```
names(f.pc)
```

```
[1] "Type"  "Model" "Age"   "Per"   "Coh"   "Drift" "Ref"   "Anova"
```

One of the components of the result is the `Drift` which is the average secular trend extracted from the model (for the given inner product)

```
f.cp$Drift
```

```
                exp(Est.)     2.5%    97.5%
APC (Y-weights)  1.021348 1.020444 1.022253
A-d              1.023580 1.023065 1.024096
```

```
f.pc$Drift
```

```
                exp(Est.)     2.5%    97.5%
APC (Y-weights)  1.021348 1.020444 1.022253
A-d              1.023580 1.023065 1.024096
```

The drift is independent of the chosen parametrization, but different from the drift parameter in the age-drift model. It also depends on the chosen inner product — of which 4 possible are directly available in `apc.fit`:

```
(drifts <- rbind(
apc.fit(lung, model = "factor", dr = "d", pr = FALSE)$Drift,
apc.fit(lung, model = "factor", dr = "r", pr = FALSE)$Drift,
apc.fit(lung, model = "factor", dr = "y", pr = FALSE)$Drift,
apc.fit(lung, model = "factor", dr = "n", pr = FALSE)$Drift)[c(2, 1, 3, 5, 7), ])
```

```
No reference cohort given;  reference cohort for age-effects is chosen as
 the median date of birth for persons  with event:  1913 .
No reference cohort given;  reference cohort for age-effects is chosen as
 the median date of birth for persons  with event:  1913 .
No reference cohort given;  reference cohort for age-effects is chosen as
 the median date of birth for persons  with event:  1913 .
No reference cohort given;  reference cohort for age-effects is chosen as
```

```
 the median date of birth for persons  with event:  1913 .
                   exp(Est.)     2.5%    97.5%
A-d                 1.023580 1.023065 1.024096
APC (D-weights)     1.019870 1.019272 1.020468
APC (Y^2/D-weights) 1.017361 1.015949 1.018775
APC (Y-weights)     1.021348 1.020444 1.022253
APC (1-weights)     1.032769 1.031537 1.034003
```

It appears that in this case the drift allocated by the naive inner product allocates the largest increase (3.3%/year), whereas the other options are in the vicinity of 2%/year.

11. The default plot method (`plot.apc`) to show the estimates in a single graph for all three allowing comparison of effects because the scaling of both x- and y-axis is the same for all effects. We add confidence intervals in various ways by using `pc.matshade`:

```
 par(mar = c(3, 4, 0, 4), las = 1)
 plot(f.cp, lwd = 1, r.txt = "Male lungcancer incidence in Denmark, per 1000 PY",
      col = "transparent")

cp.offset    RR.fac
     1765         1

 pc.points(1908, 1, lwd = 2)
    matshade(f.cp$Age[, 1], f.cp$Age[, -1], lwd = 2)
 pc.matshade(f.cp$Per[, 1], f.cp$Per[, -1], lwd = 2)
 pc.matshade(f.cp$Coh[, 1], f.cp$Coh[, -1], lty = "21", lwd = 2)
 pc.points(1965.5, 1, col = "blue", lwd = 2)
    matshade(f.pc$Age[, 1], f.pc$Age[, -1], col = "blue")
 pc.matshade(f.pc$Per[, 1], f.pc$Per[, -1], col = "blue")
 pc.matshade(f.pc$Coh[, 1], f.pc$Coh[, -1], col = "blue", lty = "21", lwd = 2)
```

12. Finally, we fit a model with natural splines — this is the default model used by `apc.fit`; the default is to use 5 knots for each of the three effects, placed so that the number of events between each pair of knots is the same. We add the estimates from this to the plots of the previous models:

```
 s.cp <- apc.fit(lung, parm = "ACP", ref.c = 1908, scale = 1000)

[1] "ML of APC-model Poisson with log(Y) offset : ( ACP ):\n"
              Model       AIC Mod. df.  Mod. dev. Test df. Test dev.      Pr(>Chi)
1               Age 16109.089      105 15242.0305       NA        NA            NA
2         Age-drift  7433.044      104  6563.9857        1 8678.0448  0.000000e+00
3        Age-Cohort  1891.432      101  1016.3729        3 5547.6128  0.000000e+00
4 Age-Period-Cohort  1300.314       98   419.2548        3  597.1181 4.247697e-129
5        Age-Period  3785.570      101  2910.5113        3 2491.2565  0.000000e+00
6         Age-drift  7433.044      104  6563.9857        3 3653.4744  0.000000e+00
  Test dev/df     H0
1          NA
2   8678.0448 zero drift
3   1849.2043 Coh eff|dr.
4    199.0394 Per eff|Coh
5    830.4188 Coh eff|Per
6   1217.8248 Per eff|dr.
```

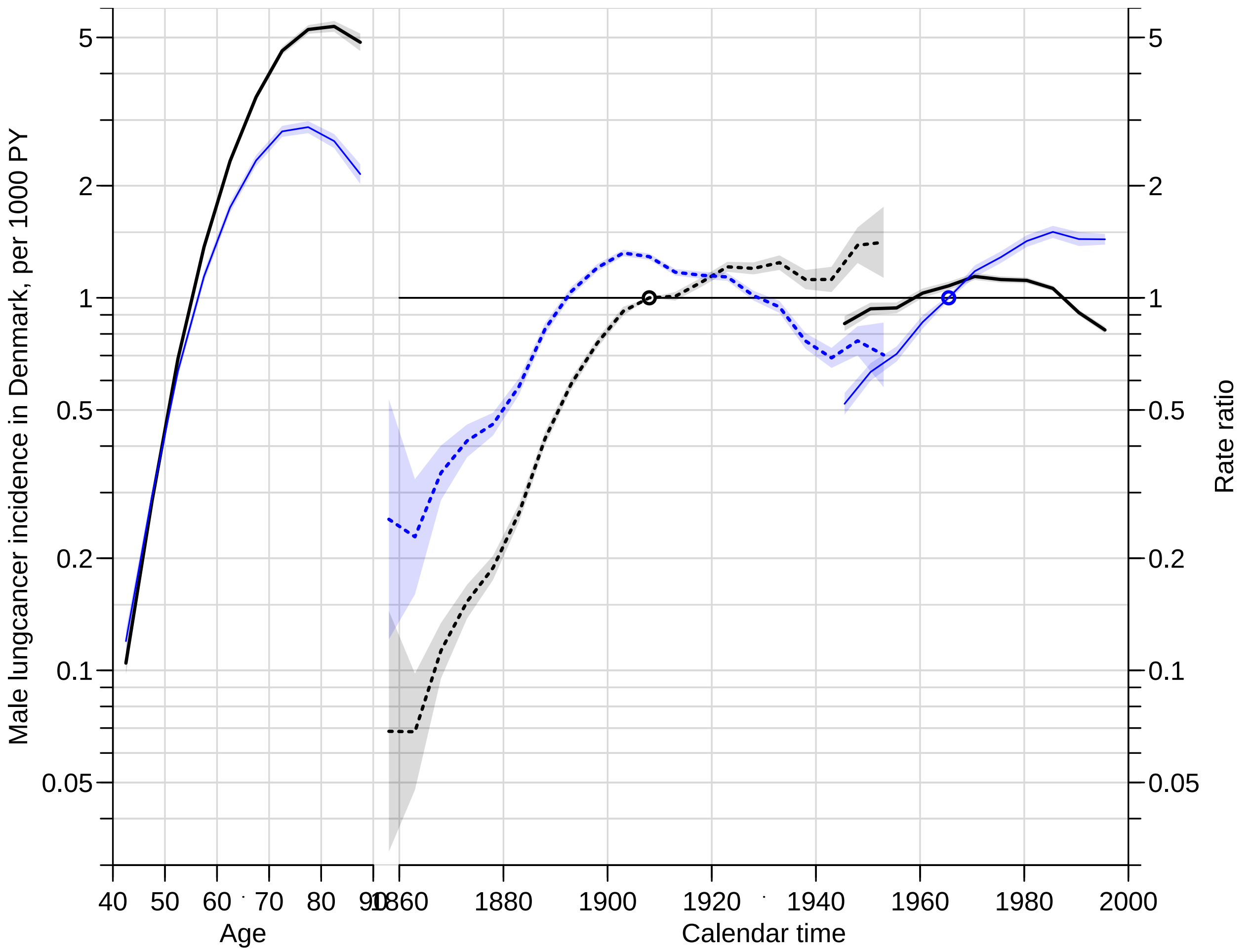

Figure 2.9: *The factor APC-model for male lung cancer in Denmark, using cohort major (black) or period major (blue) parametrization.* ../graph/APC-pc-cp

```
par(mar = c(3, 4, 0, 4), las = 1)
plot(f.cp, lwd = 1,
     r.txt = "Male lungcancer incidence in Denmark, per 1000 PY")
```

```
cp.offset    RR.fac
     1765         1
```

```
   matshade(f.cp$Age[, 1], f.cp$Age[, -1], col = "red")
pc.matshade(f.cp$Per[, 1], f.cp$Per[, -1], col = "red")
pc.matshade(f.cp$Coh[, 1], f.cp$Coh[, -1], col = "red", lty = "21")
   matshade(f.pc$Age[, 1], f.pc$Age[, -1], col = "blue")
pc.matshade(f.pc$Per[, 1], f.pc$Per[, -1], col = "blue")
pc.matshade(f.pc$Coh[, 1], f.pc$Coh[, -1], col = "blue", lty = "21")
   matshade(s.cp$Age[, 1], s.cp$Age[, -1], col = "forestgreen", lwd =2)
pc.matshade(s.cp$Per[, 1], s.cp$Per[, -1], col = "forestgreen", lwd =2)
pc.matshade(s.cp$Coh[, 1], s.cp$Coh[, -1], col = "forestgreen", lwd =2, lty = "21")
```

We see that there are no major differences between the two types of models — the advantage is that the smooth effects are more credible from a substantial point of view. The factor model bases the effects associated with the first and last few cohorts on very

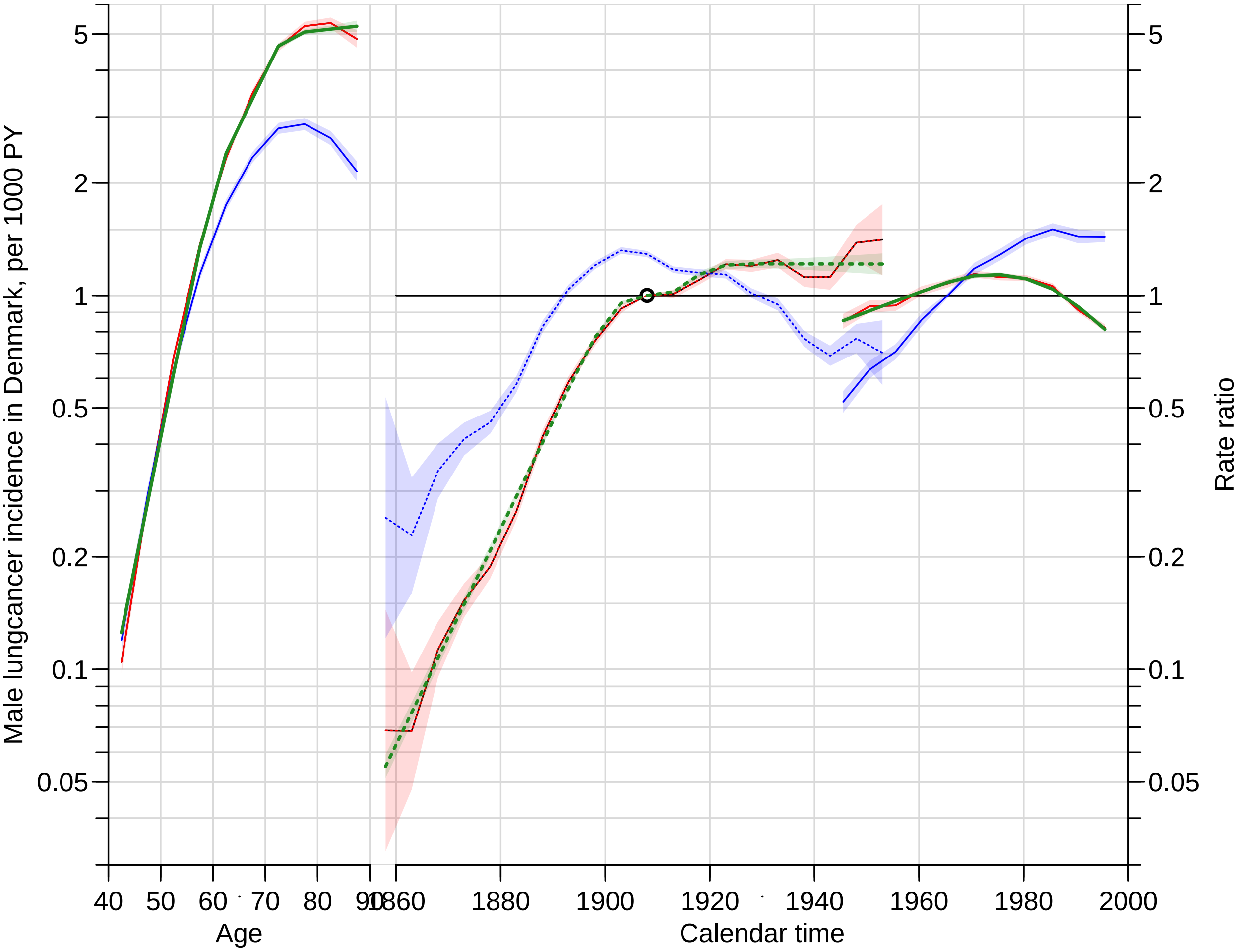

Figure 2.10: *The factor APC-model for male lung cancer in Denmark, using cohort major (black) or period major (blue) parametrization, with the cohort major parametrization of the spline model overlaid in green.* `../graph/APC-pc-cp-sp`

little information; it does not use the quantitative information about the date of birth (cohort).

The curves from the last model suggests that there is not much difference between birth cohorts after 1910, and that seem to be a calendar time decline in rates. However we should keep in mind that the model is also compatible with a decrease in cohort effects and a steep increase in period effects.

Incidentally, the estimated drifts are also different from those from the factor model:

```
 Dr <- cbind(drifts, rbind(
 apc.fit(lung, dr = "d", parm = "APC", pr = FALSE)$Drift,
 apc.fit(lung, dr = "r", parm = "APC", pr = FALSE)$Drift,
 apc.fit(lung, dr = "y", parm = "APC", pr = FALSE)$Drift,
 apc.fit(lung, dr = "n", parm = "APC", pr = FALSE)$Drift)[c(2, 1, 3, 5, 7), ])
```

```
No reference period given;  reference period for age-effects is chosen as
 the median date of event:  1980.5 .
No reference period given;  reference period for age-effects is chosen as
 the median date of event:  1980.5 .
No reference period given;  reference period for age-effects is chosen as
```

```
 the median date of event:  1980.5 .
No reference period given;  reference period for age-effects is chosen as
 the median date of event:  1980.5 .
```

```
 colnames(Dr)[c(1, 4)] <- c("Factor", "Spline")
 round((Dr-1)*100, 2)
```

```
                    Factor 2.5% 97.5% Spline 2.5% 97.5%
A-d                   2.36 2.31  2.41   2.36 2.31  2.41
APC (D-weights)       1.99 1.93  2.05   1.98 1.92  2.04
APC (Y^2/D-weights)   1.74 1.59  1.88   1.63 1.53  1.74
APC (Y-weights)       2.13 2.04  2.23   2.09 2.01  2.17
APC (1-weights)       3.28 3.15  3.40   3.26 3.19  3.34
```

Thus, there is no such thing as an “identifiable trend”.

Generally the option `dr="y"` is preferable for extraction of an overall drift.

2.5 Age-period-cohort model for Lexis triangles

1. First we read the Danish male lung cancer data tabulated by age period *and* birth cohort, `lung5-Mc.txt` and list the first few lines of the dataset. We also define the synthetic cohorts as `P5-A5`:

   ```
   library( Epi)
   ltri <- read.table( "../data/lung5-Mc.txt", header = T)
   # ltri <- read.table( "http://bendixcarstensen.com/APC/KEA-2023/data/lung5-Mc.txt", heade
   head(ltri)
   ```

   ```
     A5   P5   C5  D        Y up       Ax       Px       Cx
   1 40 1943 1898 52 336233.8  1 43.33333 1944.667 1901.333
   2 40 1943 1903 28 357812.7  0 41.66667 1946.333 1904.667
   3 40 1948 1903 51 363783.7  1 43.33333 1949.667 1906.333
   4 40 1948 1908 30 390985.8  0 41.66667 1951.333 1909.667
   5 40 1953 1908 50 391925.3  1 43.33333 1954.667 1911.333
   6 40 1953 1913 23 377515.3  0 41.66667 1956.333 1914.667
   ```

   ```
   with(ltri, table(P5 - A5 - C5))
   ```

   ```
     0   5
   110 110
   ```

   The table shows that the `C5` are the “correct” cohorts, so we also form the synthetic cohorts:

   ```
   ltri$S5 <- ltri$P5 - ltri$A5
   ```

2. Make a Lexis diagram showing the subdivision of the follow-data. You will explore the function `Lexis.diagram`.

   As an esoteric exercise we can plot the number of cases in each of the triangles:

   ```
   par( mar = c(3,3,1,1), mgp = c(3,1,0)/1.6)
   Lexis.diagram(age = 30 + c(0,65),
                date = 1938 + c(0,65),
            coh.grid = TRUE)
   with(ltri, text(Px, Ax, paste(D), cex = 0.55))
   box()
   ```

3. Use the variables `A5` and `P5` to fit a traditional age-period-cohort model with synthetic cohort defined by `S5 = P5-A5`:

   ```
   ms <- glm(cbind(D, Y) ~ -1 + factor(A5) + factor(P5) + factor(S5),
             family = poisreg, data = ltri)
   summary(ms)
   ```

   ```
   Call:
   glm(formula = cbind(D, Y) ~ -1 + factor(A5) + factor(P5) + factor(S5),
       family = poisreg, data = ltri)

   Coefficients: (1 not defined because of singularities)
                   Estimate Std. Error z value Pr(>|z|)
   factor(A5)40   -10.57240    0.20670 -51.148  < 2e-16 ***
   ```

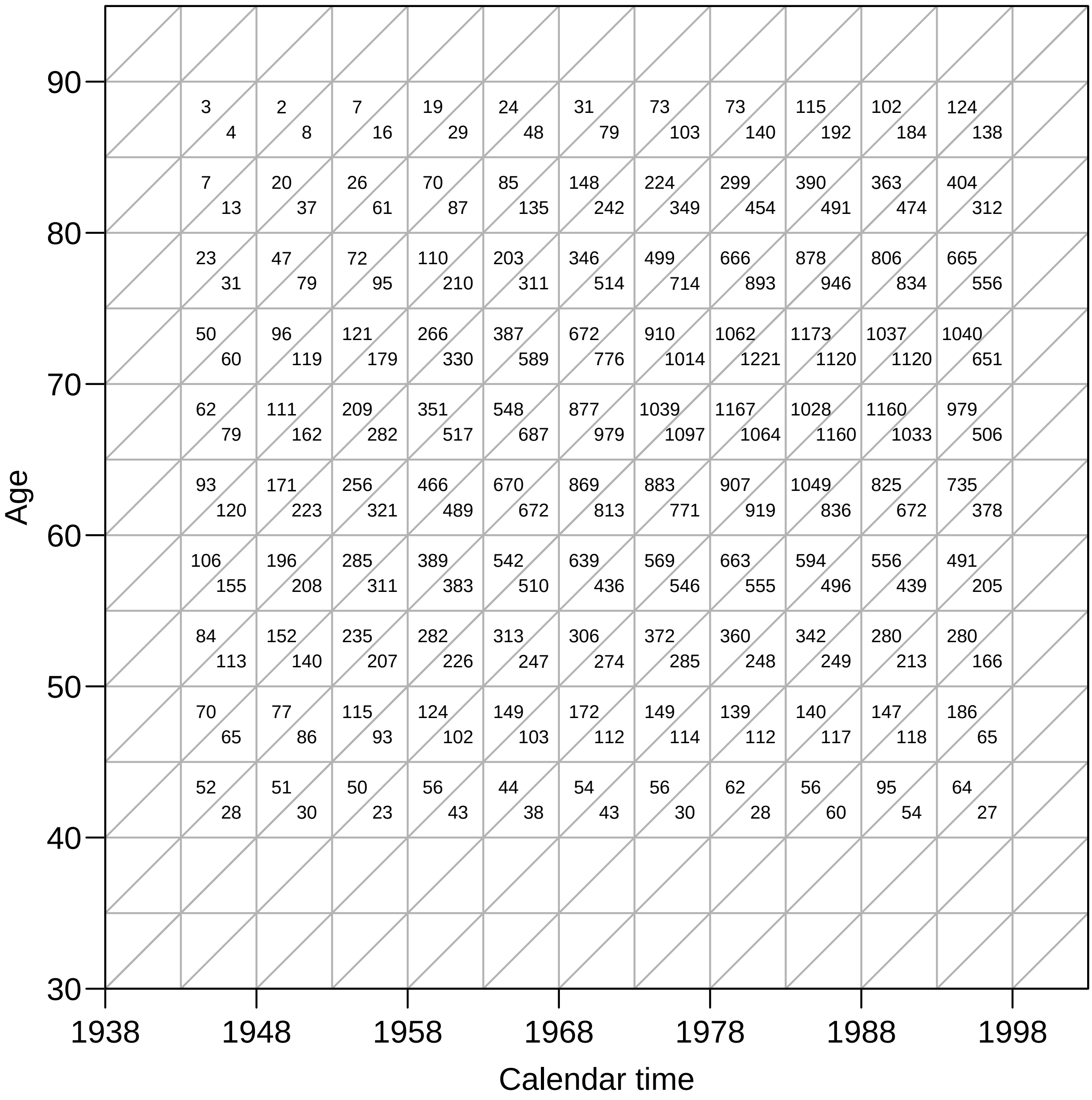

Figure 2.11: *Lexis diagram showing the extent of the lung cancer data. Numbers are incident cases.*

```
factor(A5)45    -9.73338    0.22540 -43.182  < 2e-16 ***
factor(A5)50    -9.00898    0.24334 -37.022  < 2e-16 ***
factor(A5)55    -8.47823    0.26194 -32.368  < 2e-16 ***
factor(A5)60    -8.10585    0.28092 -28.854  < 2e-16 ***
factor(A5)65    -7.86916    0.30018 -26.215  < 2e-16 ***
factor(A5)70    -7.74281    0.31966 -24.222  < 2e-16 ***
factor(A5)75    -7.76977    0.33934 -22.897  < 2e-16 ***
factor(A5)80    -7.90973    0.35925 -22.017  < 2e-16 ***
factor(A5)85    -8.16687    0.37796 -21.608  < 2e-16 ***
factor(P5)1948   0.25057    0.04115   6.090 1.13e-09 ***
factor(P5)1953   0.41507    0.05276   7.867 3.63e-15 ***
factor(P5)1958   0.66570    0.06907   9.638  < 2e-16 ***
factor(P5)1963   0.86971    0.08757   9.931  < 2e-16 ***
factor(P5)1968   1.08681    0.10698  10.159  < 2e-16 ***
factor(P5)1973   1.22682    0.12689   9.669  < 2e-16 ***
factor(P5)1978   1.38050    0.14705   9.388  < 2e-16 ***
factor(P5)1983   1.49023    0.16740   8.902  < 2e-16 ***
factor(P5)1988   1.49948    0.18786   7.982 1.44e-15 ***
factor(P5)1993   1.55151    0.20810   7.456 8.94e-14 ***
factor(S5)1863  -0.16176    0.40257  -0.402  0.68781
factor(S5)1868   0.18003    0.35012   0.514  0.60712
factor(S5)1873   0.32401    0.32402   1.000  0.31732
factor(S5)1878   0.37418    0.30269   1.236  0.21639
factor(S5)1883   0.55427    0.28250   1.962  0.04976 *
factor(S5)1888   0.85582    0.26291   3.255  0.00113 **
factor(S5)1893   1.03312    0.24387   4.236 2.27e-05 ***
factor(S5)1898   1.12663    0.22526   5.001 5.69e-07 ***
factor(S5)1903   1.16461    0.20711   5.623 1.87e-08 ***
factor(S5)1908   1.08855    0.18955   5.743 9.31e-09 ***
factor(S5)1913   0.93886    0.17277   5.434 5.51e-08 ***
factor(S5)1918   0.86676    0.15692   5.524 3.32e-08 ***
factor(S5)1923   0.80275    0.14237   5.639 1.72e-08 ***
factor(S5)1928   0.63395    0.12973   4.887 1.03e-06 ***
factor(S5)1933   0.51050    0.11962   4.268 1.97e-05 ***
factor(S5)1938   0.24666    0.11320   2.179  0.02934 *
factor(S5)1943   0.08854    0.11116   0.796  0.42575
factor(S5)1948   0.14073    0.11556   1.218  0.22329
factor(S5)1953        NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1.0037e+08  on 220  degrees of freedom
Residual deviance: 8.8866e+02  on 182  degrees of freedom
AIC: 2521.9

Number of Fisher Scoring iterations: 11
```

How many parameters does this model have?

4. Now we fit the model with the “real” cohort:

```
mc <- glm(cbind(D, Y) ~ -1 + factor(A5) + factor(P5) + factor(C5),
           family = poisreg, data = ltri)
summary(mc)$df
```

```
[1]  40 180  40
```

```
summary(ms)$df
```

```
[1]  38 182  39
```

We see that the number of parameters is now as you would expect with three factors with numbers of levels 10 (A5), 11 (P5) and 21 (C5), namely $1 + 10 + 11 + 21 - 3 = 40$, as you see from the output.

5. Plot the parameter estimates from the two models on top of each other, with confidence intervals. Remember to put the right scales on the plots.

```
 par(mfrow = c(1,3))
 (a.pt <- as.numeric(levels(factor(ltri$A5))))
```

```
 [1] 40 45 50 55 60 65 70 75 80 85
```

```
 (p.pt <- as.numeric(levels(factor(ltri$P5))))
```

```
 [1] 1943 1948 1953 1958 1963 1968 1973 1978 1983 1988 1993
```

```
 (s.pt <- as.numeric(levels(factor(ltri$S5))))
```

```
 [1] 1858 1863 1868 1873 1878 1883 1888 1893 1898 1903 1908 1913 1918 1923 1928 1933 1938
[18] 1943 1948 1953
```

```
 (c.pt <- as.numeric(levels(factor(ltri$C5))))
```

```
 [1] 1853 1858 1863 1868 1873 1878 1883 1888 1893 1898 1903 1908 1913 1918 1923 1928 1933
[18] 1938 1943 1948 1953
```

```
 matshade(a.pt, cbind(ci.exp(ms, subset = "A5"),
                      ci.exp(mc, subset = "A5")), plot = TRUE,
          lty = 1, lwd = 2, col = c("black", "blue"),
          xlab = "Age", ylab = "Rates", log = "y")
 matshade(p.pt, rbind(1,
                cbind(ci.exp(ms, subset = "P5"),
                      ci.exp(mc, subset = "P5"))), plot = TRUE,
          lty = 1, lwd = 2, col = c("black", "blue"),
          xlab = "Period", ylab = "Rates", log = "y")
 matshade(s.pt, rbind(1, ci.exp(ms, subset = "S5")), plot = TRUE,
          lty = 1, lwd = 2, col = "black",
          xlab = "cohort", ylab = "Rates", log = "y")
 matshade(c.pt, rbind(1, ci.exp(mc, subset = "C5")),
          lty = 1, lwd = 2, col = "blue",
          xlab = "cohort", ylab = "Rates", log = "y")
```

It is seen that the confidence bands are much wider for the age and cohort effects but narrower for the period effects.

6. Now fit the model using the proper midpoints of the triangles as factor levels. How many parameters does this model have?

```
 mt <- glm(cbind(D, Y) ~ -1 + factor(Ax) + factor(Px) + factor(Cx),
             family = poisreg, data = ltri)
 summary(mt)$df
```

```
[1]  76 144  80
```

```
 sum(!is.na(coef(mt)))
```

```
[1] 76
```

```
 length(coef(mt))
```

```
[1] 80
```

```
 nrow(ltri)
```

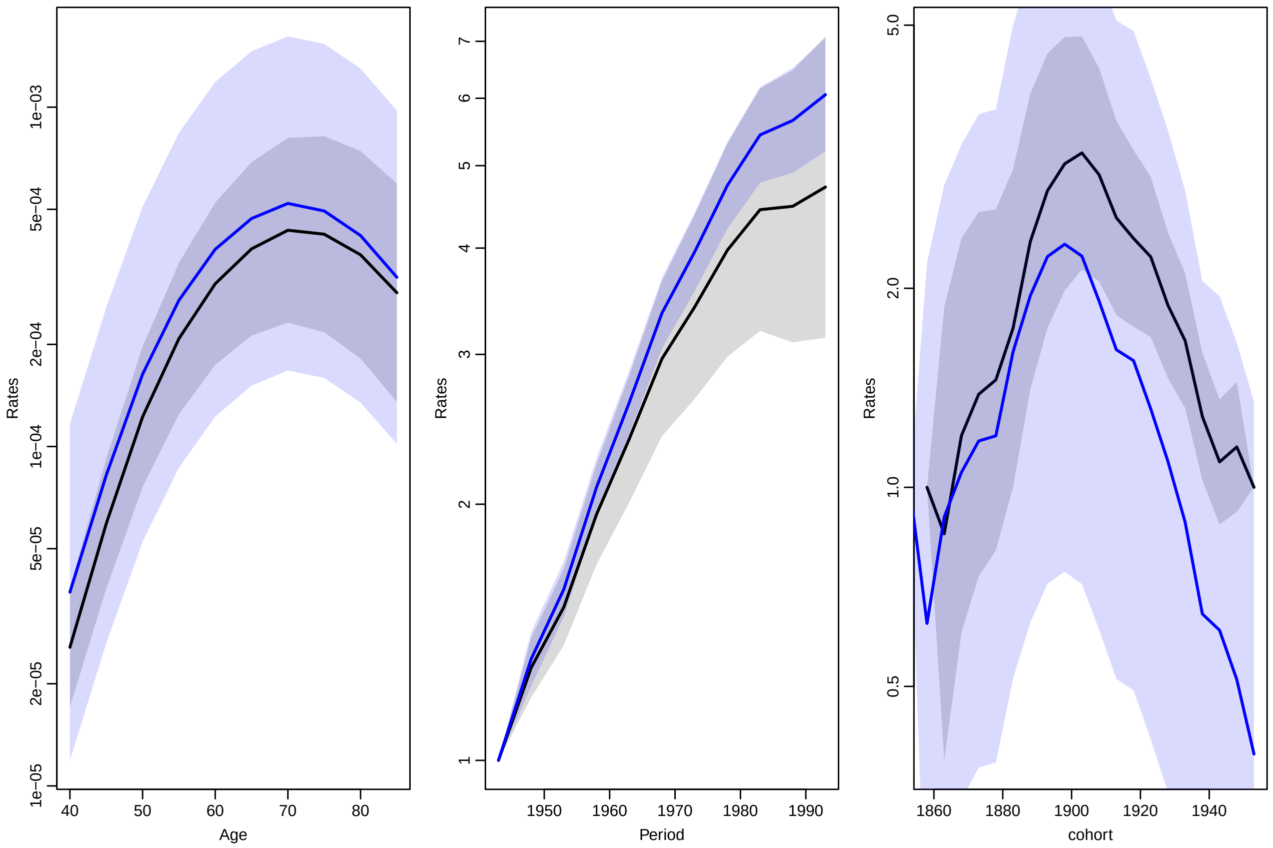

Figure 2.12: *Estimates from models with synthetic cohorts (black) and actual cohorts (blue). Model with actual cohort is wrong.*

```
[1] 220
```

```
 nrow(ltri) - sum(!is.na(coef(mt)))
```

```
[1] 144
```

7. Plot the parameters from this model in three panels as for the previous two models.

```
 par(mfrow = c(1,3))
 (a.pt <- as.numeric(levels(factor(ltri$Ax))))
```

```
 [1] 41.66667 43.33333 46.66667 48.33333 51.66667 53.33333 56.66667 58.33333 61.66667
[10] 63.33333 66.66667 68.33333 71.66667 73.33333 76.66667 78.33333 81.66667 83.33333
[19] 86.66667 88.33333
```

```
 (p.pt <- as.numeric(levels(factor(ltri$Px))))
```

```
 [1] 1944.667 1946.333 1949.667 1951.333 1954.667 1956.333 1959.667 1961.333 1964.667
[10] 1966.333 1969.667 1971.333 1974.667 1976.333 1979.667 1981.333 1984.667 1986.333
[19] 1989.667 1991.333 1994.667 1996.333
```

```
 (c.pt <- as.numeric(levels(factor(ltri$Cx))))
```

```
 [1] 1856.333 1859.667 1861.333 1864.667 1866.333 1869.667 1871.333 1874.667 1876.333
[10] 1879.667 1881.333 1884.667 1886.333 1889.667 1891.333 1894.667 1896.333 1899.667
[19] 1901.333 1904.667 1906.333 1909.667 1911.333 1914.667 1916.333 1919.667 1921.333
[28] 1924.667 1926.333 1929.667 1931.333 1934.667 1936.333 1939.667 1941.333 1944.667
[37] 1946.333 1949.667 1951.333 1954.667
```

```
matshade(a.pt, ci.exp(mt, subset = "Ax") * 10^5,
         lty = 1, lwd = 2, plot = TRUE,
         xlab = "Age", ylab = "Rates per 10^5", log = "y")
matshade(p.pt, rbind(c(1,1,1), ci.exp(mt, subset = "Px")),
         lty = 1, lwd = 2, plot = TRUE,
        xlab = "Period", ylab = "RR", log = "y")
matshade(c.pt, rbind(c(1,1,1), ci.exp(mt, subset = "Cx")),
        lty = 1, lwd = 2, plot = TRUE,
        xlab = "Cohort", ylab = "RR", log = "y")
```

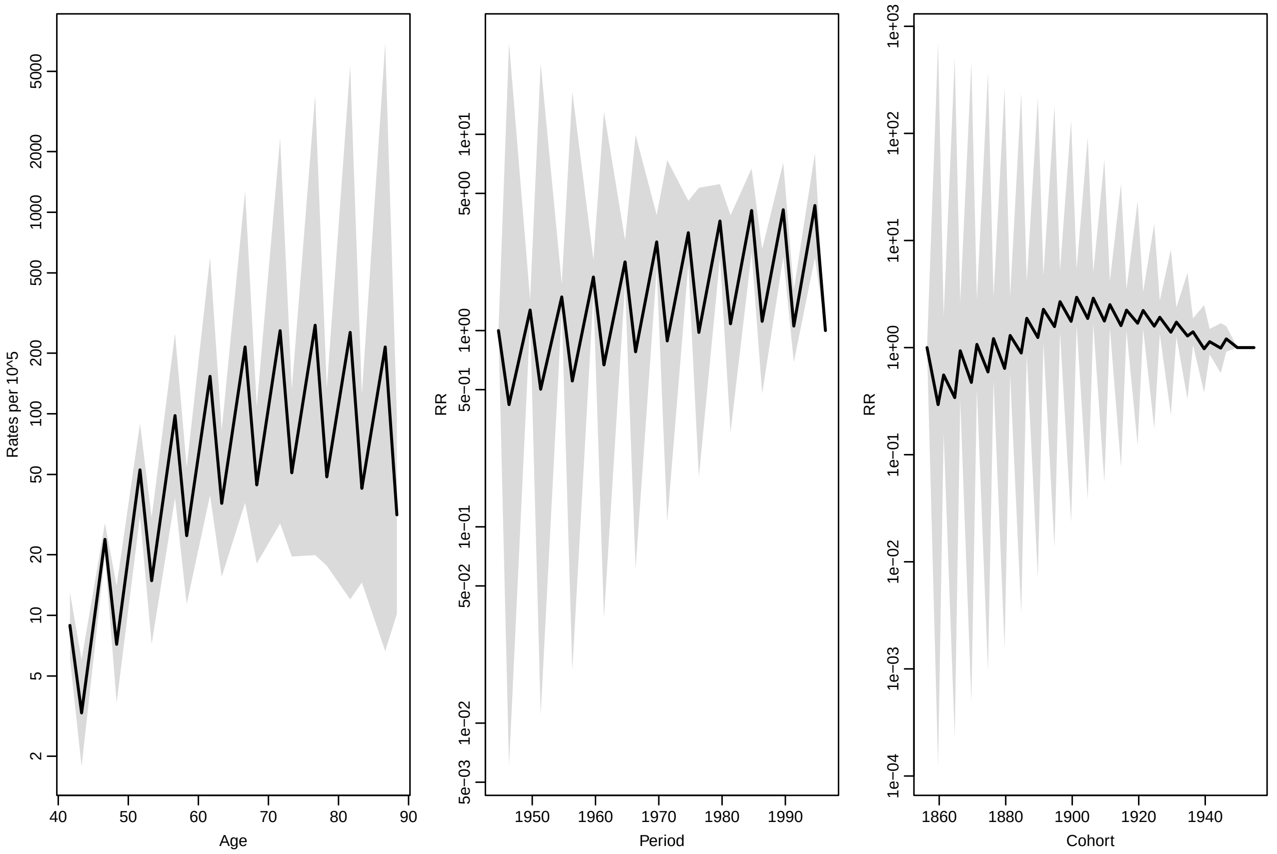

Figure 2.13: *Estimates from the model with factor levels equal to the correct midpoints of the Lexis triangles.*

We see that the parameters clearly do not convey a reasonable picture of the effects; som severe indeterminacy has crept in.

8. What is the residual deviance of this model?

```
summary(mt)$deviance
```

```
[1] 284.7269
```

9. The dataset also has a variable `up`, which indicates whether the observation comes from an upper or lower triangle. Try to tabulate it against `P5 - A5 - C5` and `P5 - A5 - S5`.

```
with(ltri, table(up, P5 - A5 - C5))
```

```
up    0   5
  0 110   0
  1   0 110
```

```
with(ltri, table(up, P5 - A5 - S5))
```

```
up    0
  0 110
  1 110
```

10. Now, fit an age-period cohort model separately for the subset of the dataset from the upper triangles and from the lower triangles. What is the residual deviance from each of these models and what is the sum of these. Compare to the model using the proper midpoints as factor levels.

```
m.up <- glm(cbind(D, Y) ~ -1 + factor(A5) + factor(P5) + factor(S5),
             family = poisreg, data = subset(ltri,up == 1))
summary(m.up)$deviance
```

```
[1] 150.2703
```

```
m.lo <- glm(cbind(D, Y) ~ -1 + factor(A5) + factor(P5) + factor(S5),
             family = poisreg, data = subset(ltri,up == 0))
summary(m.lo)$deviance
```

```
[1] 134.4566
```

```
summary(m.lo)$deviance + summary(m.up)$deviance
```

```
[1] 284.7269
```

```
summary(mt)$deviance
```

```
[1] 284.7269
```

What do you conclude from this?

11. Next, repeat the plots of the parameters from the model using the proper midpoints as factor levels, but now super-posing the estimates (in different color) from each of the two models just fitted. What goes on?

```
par(mfrow = c(1,3))
a.pt <- as.numeric(levels(factor(ltri$Ax)))
p.pt <- as.numeric(levels(factor(ltri$Px)))
c.pt <- as.numeric(levels(factor(ltri$Cx)))
a5.pt <- as.numeric(levels(factor(ltri$A5)))
p5.pt <- as.numeric(levels(factor(ltri$P5)))
s5.pt <- as.numeric(levels(factor(ltri$S5)))
matplot(a.pt, ci.lin(mt, subset = "Ax", Exp = TRUE)[,5:7]/10^5,
         type = "l", lty = 1, lwd = c(2,1,1), col = gray(0.7),
         xlab = "Age", ylab = "Rates", log = "y")
matpoints(a5.pt, ci.lin(m.up, subset = "A5", Exp = TRUE)[,5:7]/10^5,
           pch = c(16,3,3), col = "blue")
matpoints(a5.pt, ci.lin(m.lo, subset = "A5", Exp = TRUE)[,5:7]/10^5,
           pch = c(16,3,3), col = "red")
```

```
matplot(p.pt, rbind(c(1,1,1), ci.lin(mt, subset = "Px",Exp = TRUE)[,5:7]),
         type = "l", lty = 1, lwd = c(2,1,1), col = gray(0.7),
         xlab = "Period", ylab = "RR", log = "y")
matpoints(p5.pt[-1], ci.lin(m.up, subset = "P5", Exp = TRUE)[,5:7],
           pch = c(16,3,3), col = "blue")
matpoints(p5.pt[-1], ci.lin(m.lo, subset = "P5", Exp = TRUE)[,5:7],
           pch = c(16,3,3), col = "red")
matplot(c.pt, rbind(c(1,1,1),ci.lin(mt, subset = "Cx", Exp = TRUE)[,5:7]),
         type = "l", lty = 1, lwd = c(2,1,1), col = gray(0.7),
         xlab = "Cohort", ylab = "RR", log = "y")
matpoints(s5.pt[-1], ci.lin(m.up, subset = "S5", Exp = TRUE)[,5:7],
           pch = c(16,3,3), col = "blue")
matpoints(s5.pt[-1], ci.lin(m.lo, subset = "S5", Exp = TRUE)[,5:7],
           pch = c(16,3,3), col = "red")
```

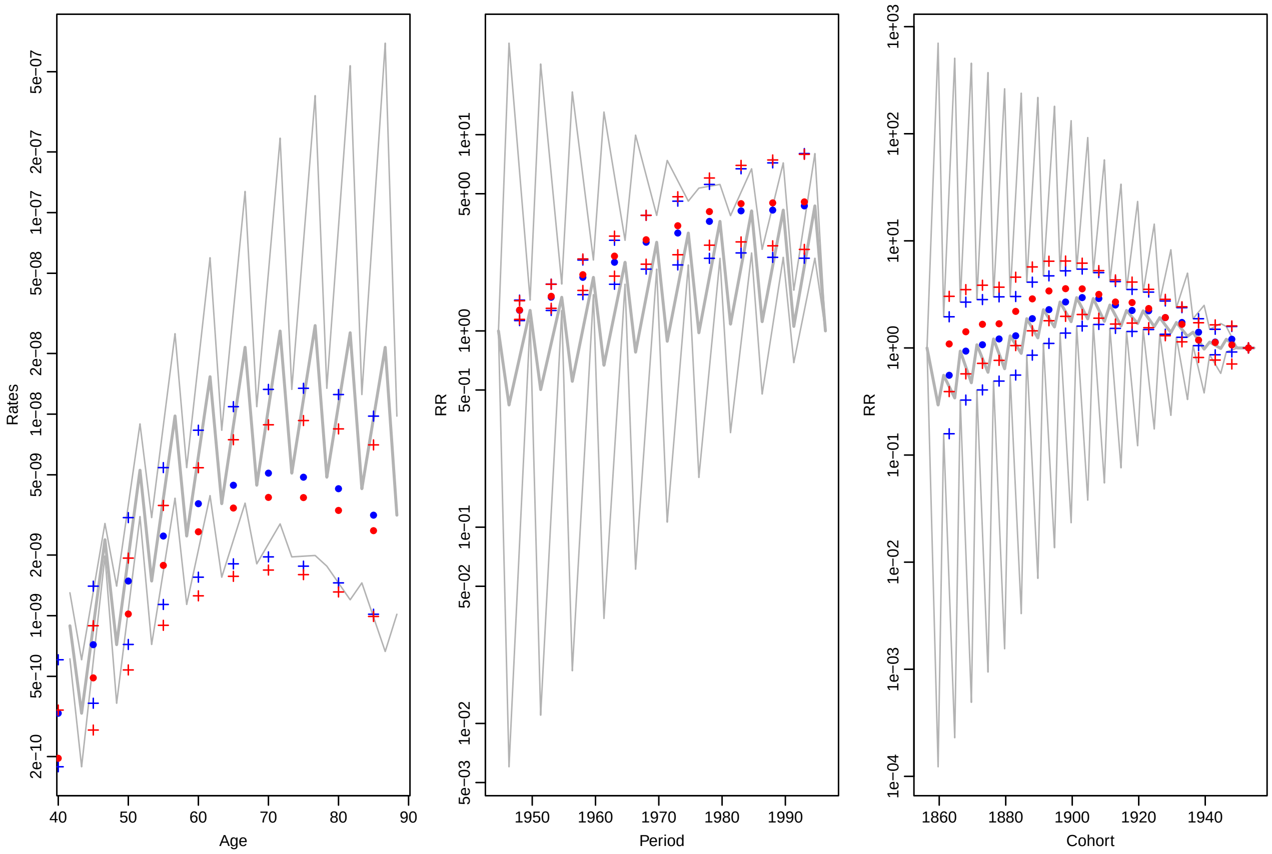

Figure 2.14: *Estimates from the model with factor levels equal to the correct midpoints of the Lexis triangles (gray curves) and the same model fitted separately for the upper (blue) and lower (red) Lexis triangles.*

The model fitted with the “correct” factor levels is actually two different models. This is because observations in upper triangles are modelled by one set of the parameters, and those in lower triangel by another set of parameters.

Because of the ordering of the levels, the parametrization is different, but that is all.

There is no way out of the squeeze, except by resorting to parametric models for the actual underlying scales, abandoning the factor modelling, and by that also the ridiculous inherent assumption of echangeability of factor levels.

12. We now load the splines package and fit a model using the correct midpoints of the triangles as quantitative variables in restricted cubic splines, using the function `ns`:

```
 library(splines)
 mspl <- glm(cbind(D, Y) ~ -1 + ns(Ax,df = 7,intercept = T)
                              + ns(Px,df = 6,intercept = F)
                              + ns(Cx,df = 6,intercept = F),
              family = poisreg, data = ltri)
 summary(mspl)
```

```
Call:
glm(formula = cbind(D, Y) ~ -1 + ns(Ax, df = 7, intercept = T) +
    ns(Px, df = 6, intercept = F) + ns(Cx, df = 6, intercept = F),
    family = poisreg, data = ltri)

Coefficients: (1 not defined because of singularities)
                                Estimate Std. Error z value Pr(>|z|)
ns(Ax, df = 7, intercept = T)1  -8.08248    0.09584 -84.329  < 2e-16 ***
ns(Ax, df = 7, intercept = T)2  -8.81421    0.11261 -78.271  < 2e-16 ***
ns(Ax, df = 7, intercept = T)3  -8.20301    0.11520 -71.209  < 2e-16 ***
ns(Ax, df = 7, intercept = T)4  -7.90599    0.11814 -66.921  < 2e-16 ***
ns(Ax, df = 7, intercept = T)5  -3.98298    0.08558 -46.540  < 2e-16 ***
ns(Ax, df = 7, intercept = T)6 -21.35542    0.24841 -85.967  < 2e-16 ***
ns(Ax, df = 7, intercept = T)7   0.70588    0.05540  12.741  < 2e-16 ***
ns(Px, df = 6, intercept = F)1   0.59989    0.03777  15.883  < 2e-16 ***
ns(Px, df = 6, intercept = F)2   0.94029    0.04319  21.771  < 2e-16 ***
ns(Px, df = 6, intercept = F)3   1.18582    0.04354  27.237  < 2e-16 ***
ns(Px, df = 6, intercept = F)4   1.22421    0.04204  29.122  < 2e-16 ***
ns(Px, df = 6, intercept = F)5   1.46929    0.08247  17.816  < 2e-16 ***
ns(Px, df = 6, intercept = F)6   1.07376    0.04202  25.555  < 2e-16 ***
ns(Cx, df = 6, intercept = F)1   1.57834    0.10334  15.273  < 2e-16 ***
ns(Cx, df = 6, intercept = F)2   1.60219    0.11202  14.303  < 2e-16 ***
ns(Cx, df = 6, intercept = F)3   1.37407    0.10178  13.500  < 2e-16 ***
ns(Cx, df = 6, intercept = F)4   1.03167    0.07211  14.306  < 2e-16 ***
ns(Cx, df = 6, intercept = F)5   1.19310    0.21716   5.494 3.93e-08 ***
ns(Cx, df = 6, intercept = F)6        NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1.0037e+08  on 220  degrees of freedom
Residual deviance: 4.3344e+02  on 202  degrees of freedom
AIC: 2026.7

Number of Fisher Scoring iterations: 11
```

```
 summary(mt)$deviance
```

```
[1] 284.7269
```

```
 summary(mspl)$deviance
```

```
[1] 433.4351
```

```
 summary(mt)$deviance - summary(mspl)$deviance
```

```
[1] -148.7082
```

```
summary(mt)$df       - summary(mspl)$df
```

```
[1]  58 -58  61
```

13. How do the deviances compare?

14. Make a prediction of the terms, using `predict.glm` using the argument `type = "terms"` and `se.fit = TRUE`. Remember to look up the help page for `predict.glm`.

```
pspl <- predict(mspl, type = "terms", se.fit = TRUE)
str(pspl)
```

```
List of 3
 $ fit           : num [1:220, 1:3] -10.8 -11.1 -10.8 -11.1 -10.8 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:220] "1" "2" "3" "4" ...
  .. ..$ : chr [1:3] "ns(Ax, df = 7, intercept = T)" "ns(Px, df = 6, intercept = F)" "ns(C
  ..- attr(*, "constant")= num 0
 $ se.fit        : num [1:220, 1:3] 0.107 0.109 0.107 0.109 0.107 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:220] "1" "2" "3" "4" ...
  .. ..$ : chr [1:3] "ns(Ax, df = 7, intercept = T)" "ns(Px, df = 6, intercept = F)" "ns(C
 $ residual.scale: num 1
```

```
a.ord <- order(ltri$Ax)
p.ord <- order(ltri$Px)
c.ord <- order(ltri$Cx)
par(mfrow = c(1,3))
matplot(ltri$Ax[a.ord], exp(cbind(pspl$fit[,1],
                                  pspl$se.fit[,1])[a.ord,]
                            %*% ci.mat())*10^5,
        type = "l", lty = 1, lwd = c(2,1,1), col = gray(0.2),
        xlab = "Age", ylab = "Rates", log = "y")
matplot(ltri$Px[p.ord], exp(cbind(pspl$fit[,2],
                                  pspl$se.fit[,2])[p.ord,]
                            %*% ci.mat()),
        type = "l", lty = 1, lwd = c(2,1,1), col = gray(0.2),
        xlab = "Period", ylab = "RR", log = "y")
matplot(ltri$Cx[c.ord], exp(cbind(pspl$fit[,3],
                                  pspl$se.fit[,3])[c.ord,]
                            %*% ci.mat()),
        type = "l", lty = 1, lwd = c(2,1,1), col = gray(0.2),
        xlab = "Cohort", ylab = "RR", log = "y" )
```

15. The terms shown in the plot are not useful, they are assumin that the last of the spline parameters are 0, essentially constraining the last value of the cohort effect to be 0 (the point is not in the plot because the value is rendered as `NA`).

    The function `apc.fit` fits a model with natural splines as the above, but also reparametrizes the age, period and cohort effects explicitly according to well defined principles.

```
lACP <- apc.fit(A = ltri$Ax,
                P = ltri$Px,
                D = ltri$D,
                Y = ltri$Y,
            ref.c = 1900)
```

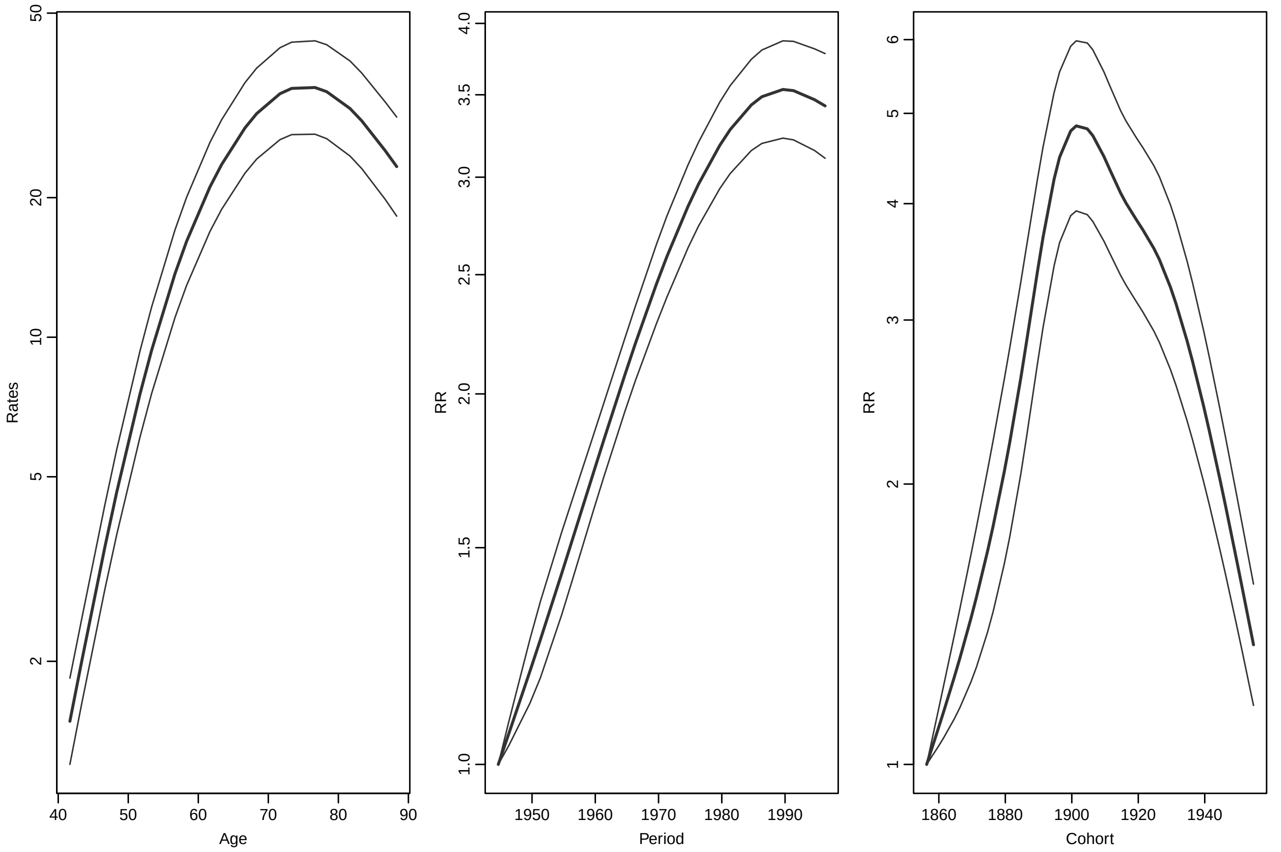

Figure 2.15: *Estimates of terms from the spline model for the Lexis triangeled data.* `../graph/apc-tri-parmest4`

```
[1] "ML of APC-model Poisson with log(Y) offset : ( ACP ):\n"
              Model      AIC Mod. df.  Mod. dev. Test df. Test dev.      Pr(>Chi)
1               Age 17135.504      215 15568.2387       NA        NA            NA
2         Age-drift  8534.339      214  6965.0731        1 8603.1656  0.000000e+00
3        Age-Cohort  2727.795      211  1152.5297        3 5812.5433  0.000000e+00
4 Age-Period-Cohort  2130.641      208   549.3752        3  603.1545 2.087058e-130
5        Age-Period  4786.694      211  3211.4282        3 2662.0530  0.000000e+00
6         Age-drift  8534.339      214  6965.0731        3 3753.6448  0.000000e+00
  Test dev/df    H0
1          NA
2   8603.1656 zero drift
3   1937.5144 Coh eff|dr.
4    201.0515 Per eff|Coh
5    887.3510 Coh eff|Per
6   1251.2149 Per eff|dr.
```

```
 class(lACP) ; names(lACP)
```

```
[1] "apc"
```

```
[1] "Type"  "Model" "Age"   "Per"   "Coh"   "Drift" "Ref"   "Anova" "Knots"
```

```
 plot(lACP)
```

```
cp.offset    RR.fac
 1765.000     0.001
```

Inspect the components of `lACP` to find out what they are. Or read the manual page for `apc.fit` to find out.

16. Now try the default plot method for `apc` objects.

```
 plot(lACP)
```

```
cp.offset    RR.fac
 1765.000     0.001
```

```
 lAPC <- apc.fit(A = ltri$Ax,
                 P = ltri$Px,
                 D = ltri$D,
                 Y = ltri$Y / 1000,
              parm = "APC",
             ref.p = 1950)
```

```
[1] "ML of APC-model Poisson with log(Y) offset : ( APC ):\n"
             Model       AIC Mod. df.  Mod. dev. Test df. Test dev.      Pr(>Chi)
1              Age 17135.504      215 15568.2387       NA        NA            NA
2        Age-drift  8534.339      214  6965.0731        1 8603.1656  0.000000e+00
3       Age-Cohort  2727.795      211  1152.5297        3 5812.5433  0.000000e+00
4 Age-Period-Cohort  2130.641      208   549.3752        3  603.1545 2.087058e-130
5       Age-Period  4786.694      211  3211.4282        3 2662.0530  0.000000e+00
6        Age-drift  8534.339      214  6965.0731        3 3753.6448  0.000000e+00
  Test dev/df     H0
1          NA
2   8603.1656 zero drift
3   1937.5144 Coh eff|dr.
4    201.0515 Per eff|Coh
5    887.3510 Coh eff|Per
6   1251.2149 Per eff|dr.
```

```
 plot(lAPC)
```

```
cp.offset    RR.fac
   1765.0       0.1
```

17. Make slicker plot of the effects using `apc.frame` and `matshade` and `pc.matshade`

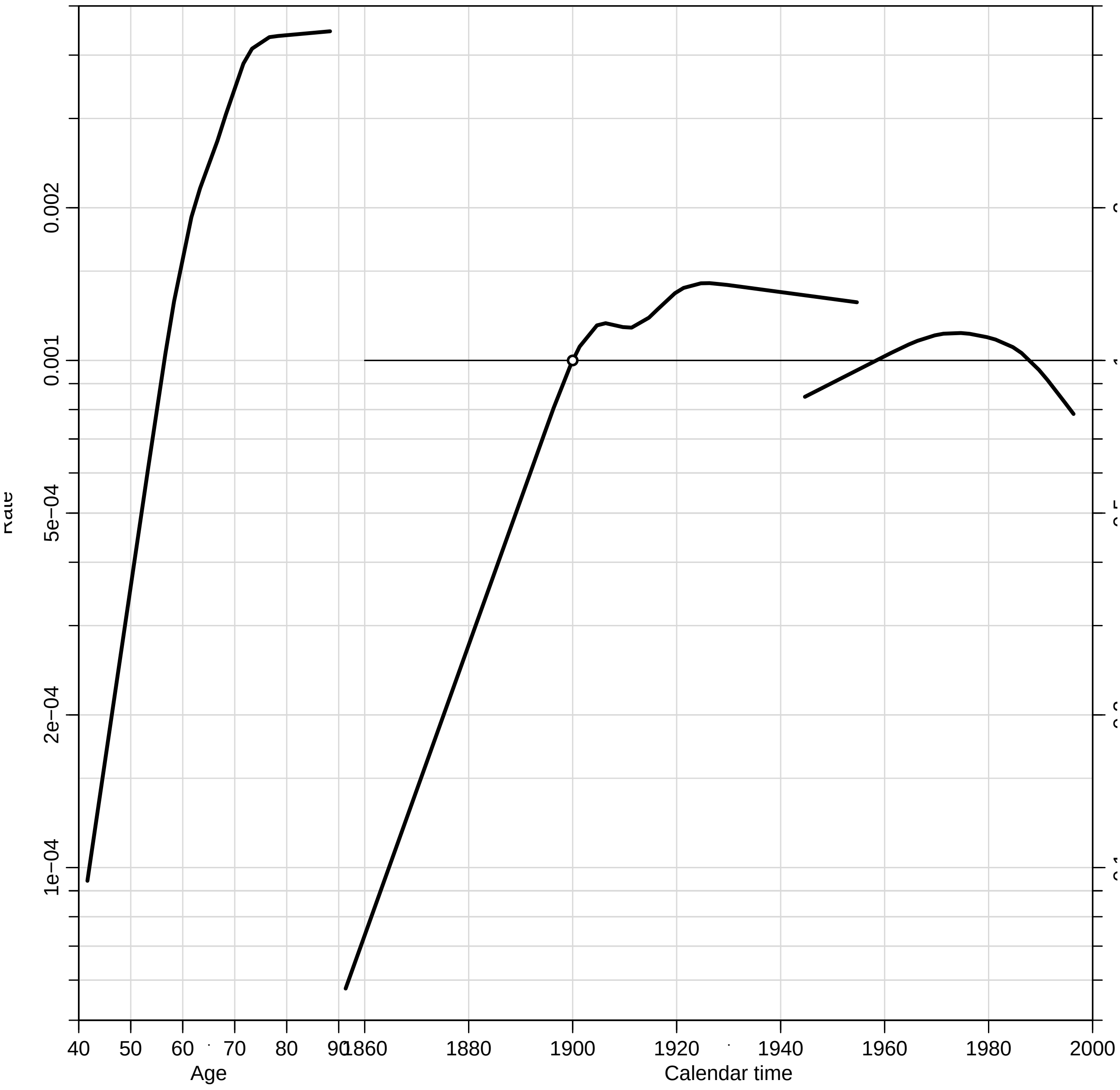

Figure 2.16: *Default APC-parameter plot for the model with period effect as residual.* `../graph/apc-tri-lungACP`

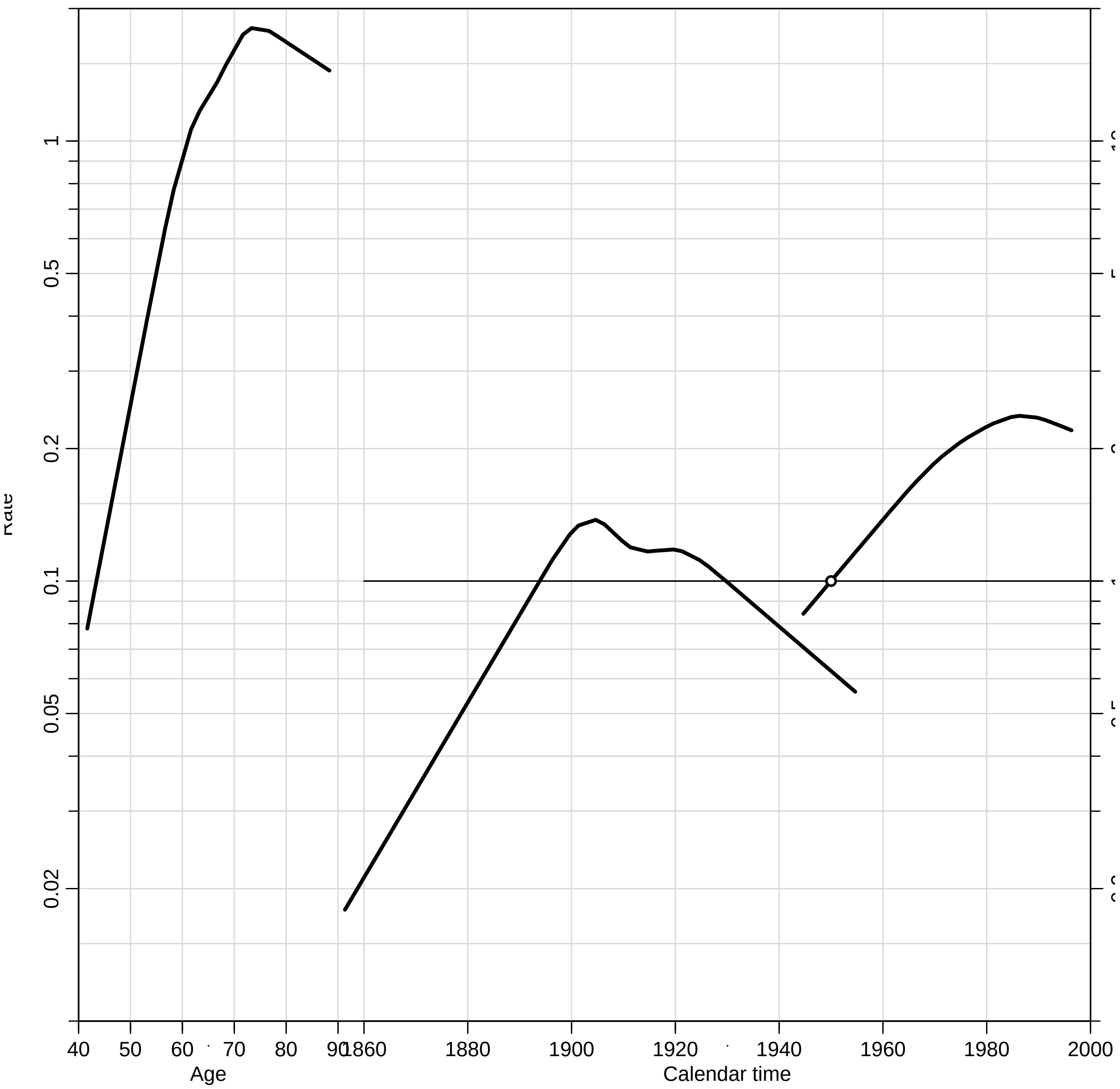

Figure 2.17: *Default APC-parameter plot for the model with cohort effect as residual.* `../graph/apc-tri-lungAPC`

```
library(Epi)
library(tidyverse)
```

2.6 Lung cancer: the sex difference

The following exercise is aimed at investigating the effect of age, period and cohort on the lung cancer incidence for both sexes using one complex age-period-cohort model. First, we will use 5-year triangular data to xxxx and build separate models for males and females. Further the complex model will be built for 1-year triangular data.

1. First we read 1-year triangular data from data set `apc-Lung.txt`

   ```
   library( Epi )
   library( splines )
   lung <- read.table( "../data/apc-Lung.txt", header=T )
   head( lung)
   ```

2. The variables `A`, `P` and `C` are the left endpoints of the tabulation intervals, so the value of the variable `P-A-C` is 0 for lower triangles and 1 for upper triangles in the Lexis diagram. This can the be used to compute the correct values of the mean age and period (and cohort) in the dataset.

   ```
   lung <- transform( lung, up = P-A-C, At = A, Pt = P, Ct = C )
   lung <- transform( lung, A = At + 1/3 + up/3,
                            P = Pt + 2/3 - up/3 )
   lung <- transform( lung, C = P - A )
   head( lung )
   ```

   A bit of care is required with the `transform` function; each of the assignments is made in the original data frame given as the first argument, hence it is not possible compute the correct `C` using the computed values of `A` and `P`, so it has to be done in two steps as above. Or by explicitly defining as: `C = Pt+2/3-up/3 - (At+1/3+up/3)`

3. We can make an overview of the rates if we can produce a table of the rates in a suitable form. This can be done by grouping on the fly and tabulating by sex too:

   ```
   lrate <- with( subset( lung, A>40 & A<90 ),
                  tapply( D, list(sex,
                                  floor(A/5)*5+2.5,
                                  floor((P-1943)/5)*5+1943+2.5),
                          sum ) /
                  tapply( Y, list(sex,
                                  floor(A/5)*5+2.5,
                                  floor((P-1943)/5)*5+1943+2.5),
                          sum ) * 10^5 )
   ```

   With this three-way table we can plot the rates for males and females in one go, using the same scale for the axes among men and women; as seen in the figure **??**:

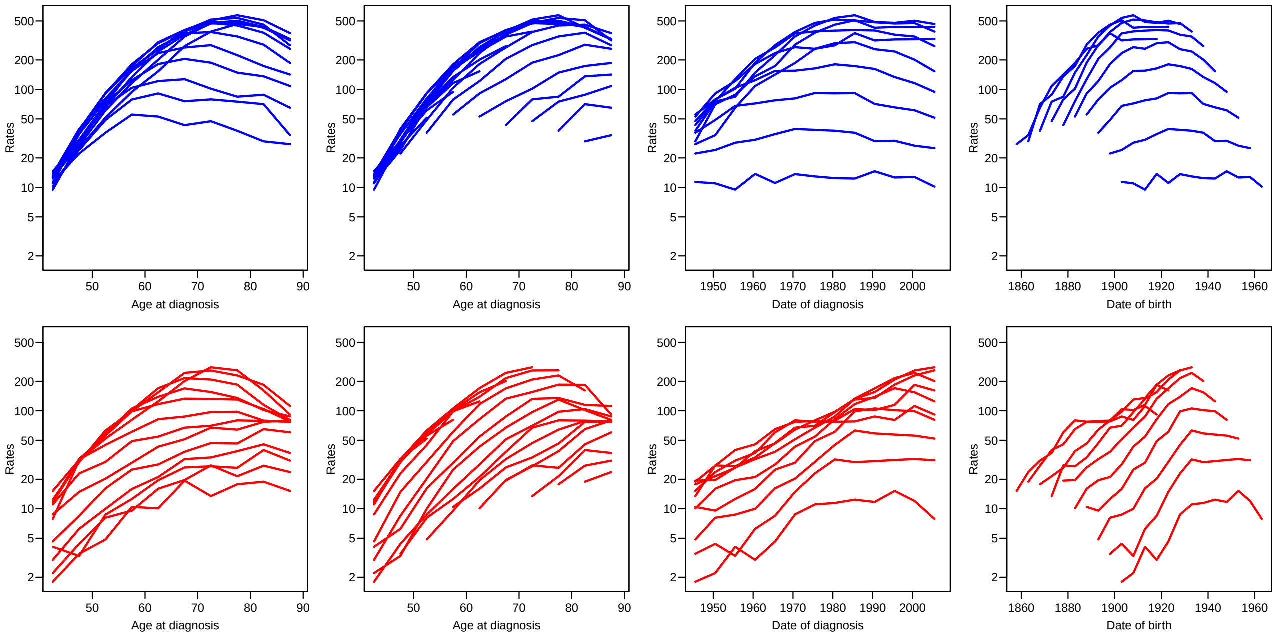

Figure 2.18: *Empirical rates of lung cancer in 5×5 age-period squares of the Lexis diagram for men (blue) and women (red).*

```
# x11(h=18,w=27,p=24)
par( mfrow=c(2,4), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
rateplot( lrate[1,,], col="blue", ylim=range(lrate,na.rm=T) )
rateplot( lrate[2,,], col="red" , ylim=range(lrate,na.rm=T) )
```

4. The models are easily fitted separately using the `subset` function on the data frame:

```
apc.m <- apc.fit( subset(lung,sex==1 & A>40), npar=c(8,8,15), ref.c=1930, scale=10^5 )
apc.f <- apc.fit( subset(lung,sex==2 & A>40), npar=c(8,8,15), ref.c=1930, scale=10^5 )
```

The default is to allocate the drift with the cohort and leave the period effect flat with an average of 0 (on the log-scale).

We can plot the the results separately and then judging from the displays find out what display is required for a sensible common plot

```
par( mfrow=c(1,1) )
apc.plot( apc.m, col="blue" )


apc.plot( apc.f, col="red" )
```

Now we can set up a plotting frame for the apc-plot of both set of estimated effects in one frame:

```
    r.lab <- c(6,c(1,2,5)*10,c(1,2,5)*100)
   rr.ref <- 200
    r.tic <- c(5:9,1:9*10,1:7*100)
par( las=1, mar=c(4,3,1,4), mgp=c(3,1,0)/1.6 )
```

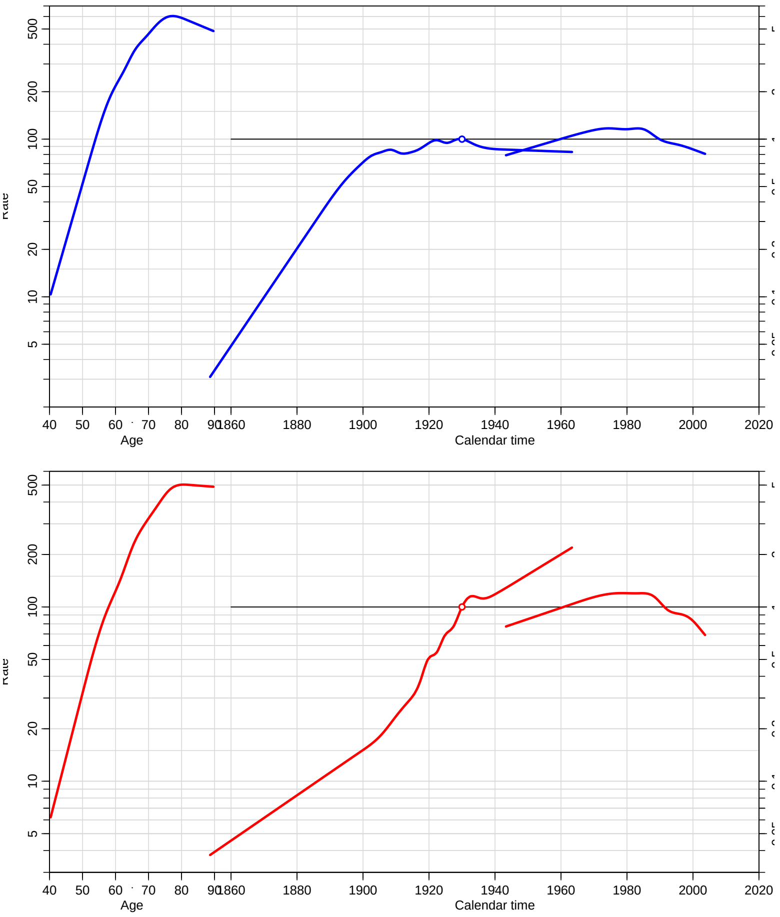

Figure 2.19: *Initial sketch plots for the male and the female rates of lung cancer incidence in Denmark.*

```
apc.frame( a.lab = seq(40,90,20),
          cp.lab = seq(1880,2000,20),
           r.lab = c(6,c(1,2,5)*10,c(1,2,5)*100),
          rr.lab = r.lab / rr.ref,
          rr.ref = rr.ref,
           a.tic = seq(35,90,5),
          cp.tic = seq(1855,2005,5),
           r.tic = r.tic,
          rr.tic = r.tic / rr.ref,
         tic.fac = 1.3,
           a.txt = "Age",
          cp.txt = "Calendar time",
           r.txt = "Lung cancer rate per 100,000 person-years",
          rr.txt = "Rate ratio",
        ref.line = TRUE,
             gap = 7,
        col.grid = gray(0.85),
           sides = c(1,2,4) )
apc.lines( apc.m, col="blue", ci=T )
apc.lines( apc.f, col="red" , ci=T )
```

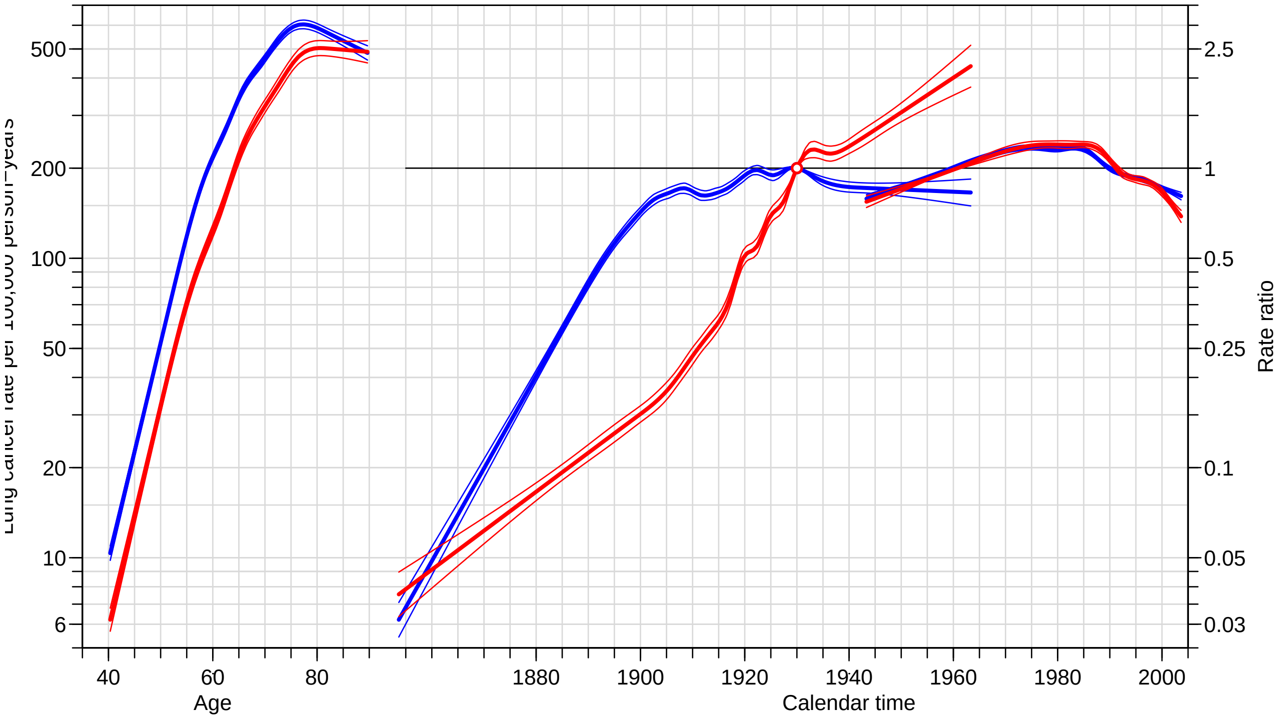

Figure 2.20: *APC estimates for male and female lung cancer incidence rates in Denmark.* `../graph/lung-sex-apc-2`

5. The ratios of the rates also follows an age-period-cohort model:

$$\begin{aligned}\log\bigl(\lambda_M(a.p)/\lambda_F(a,p)\bigr) &= \log\bigl(\lambda_M(a.p)\bigr) - \log\bigl(\lambda_F(a,p)\bigr)\\ &= \bigl(f_M(a) - f_F(a)\bigr)+\\ &\ \bigl(g_M(p) - g_F(p)\bigr)+\\ &\ \bigl(h_M(c) - h_F(c)\bigr)\end{aligned}$$

so for the rate-ratios we have exactly the same identification problems, but we can for a start just compute the ratios of the effects with confidence intervals.

Note that since we constrained the cohort effects to be 0 for the 1930 cohort (`ref.c=1930`), the difference between cohort effects for men and women will also be 0 in 1930. And moreover, since the mean and slope of the period effects are 0 for both sexes too, this will also be the case for the difference; so the APC-model induced for the sex-ratio will have the same constraints as the ones for the two sexes.

To derive the RRs from the estimated effects from the two independent sets of data it is easier to devise a small function that takes two sets of estimated rates/RRs with c.i.s and returns the ratio with c.i.s:

```
rr <- function(one, two) cbind(one[,1], ci.ratio(one[,-1], two[,-1]))
rr.Age <- rr( apc.m$Age, apc.f$Age )
rr.Per <- rr( apc.m$Per, apc.f$Per )
rr.Coh <- rr( apc.m$Coh, apc.f$Coh )
```

In order to plot these in an apc-frame, we can just fake an apc-object, and

In order to get a reasonable apc-frame we compute the ranges of the RRs:

```
( RRr <- range( rbind(rr.Age[,-1],
                      rr.Per[,-1],
                      rr.Coh[,-1]) ) )
```

So we can now use these to devise a frame which stretches from 0.2 to 5. But we will also need an `apc` object with the rate-ratios in, in order to use `apc.lines` to plot them simply. This is most easily done by copying one of the other objects and replacing the estimates with the RR estimates:

```
apc.mf <- apc.m
apc.mf$Age <- rr.Age
apc.mf$Per <- rr.Per
apc.mf$Coh <- rr.Coh
```

So now we can plot first the fame and then put in the RRs:

```
par( las=1, mar=c(4,3,1,2), mgp=c(3,1,0)/1.6 )
apc.frame( a.lab = seq(40,90,20),
          cp.lab = seq(1880,2000,20),
           r.lab = c(0.2,0.5,1,2,5),
          rr.ref = 1,
           a.tic = seq(35,90,5),
          cp.tic = seq(1855,2005,5),
           r.tic = c(2:9/10,1:5),
         tic.fac = 1.3,
           a.txt = "Age",
          cp.txt = "Calendar time",
           r.txt = "M/F Rate ratio of lung cancer",
          rr.txt = "",
        ref.line = TRUE,
             gap = 13,
        col.grid = gray(0.85),
           sides = c(1,2,4) )
abline( h=1 )
apc.lines( apc.mf, col="black", ci=T )
```

Note that we put in a reference line using `abline(h=1)`, because the `ref.line=TRUE` argument to `apc.frame` only produces a reference line on the calendar time part of the plot, and we want one at the age-range too, since we are plotting RRs for all three effects.

Alternatively we could add the RR to the joint plot:

```
    r.lab <- c(6,c(1,2,5)*10,c(1,2,5)*100)
   rr.ref <- 100
    r.tic <- c(5:9,1:9*10,1:6*100)
par( las=1, mar=c(4,3,1,4), mgp=c(3,1,0)/1.6 )
apc.frame( a.lab = seq(40,90,20),
          cp.lab = seq(1880,2000,20),
           r.lab = c(6,c(1,2,5)*10,c(1,2,5)*100),
          rr.lab = r.lab / rr.ref,
          rr.ref = rr.ref,
           a.tic = seq(35,90,5),
```

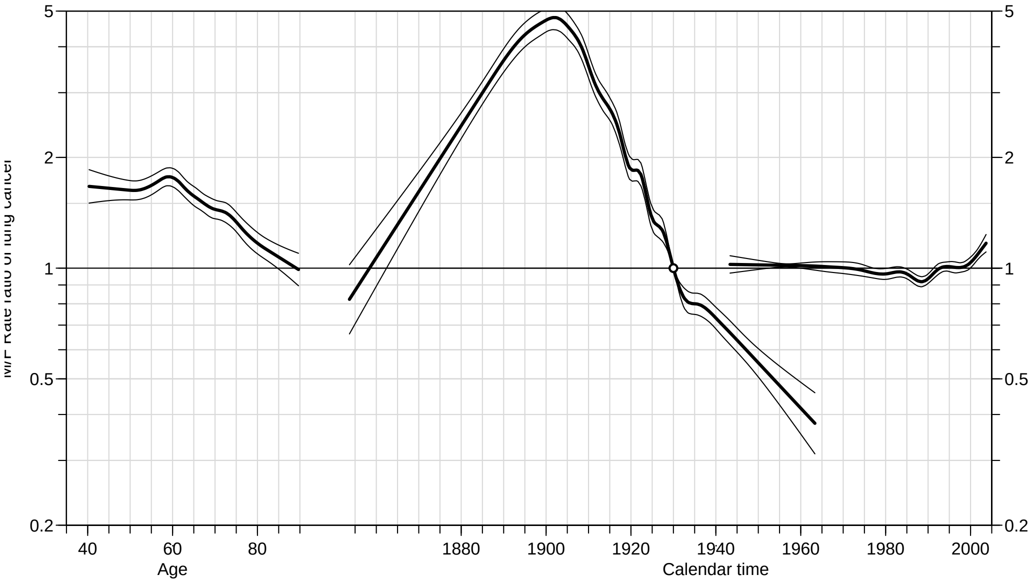

Figure 2.21: *M/F rate-ratio of lung cancer in Denmark.*

```
          cp.tic = seq(1855,2005,5),
           r.tic = r.tic,
          rr.tic = r.tic / rr.ref,
         tic.fac = 1.3,
           a.txt = "Age",
          cp.txt = "Calendar time",
           r.txt = "Lung cancer rate per 100,000 person-years",
          rr.txt = "Rate ratio",
        ref.line = TRUE,
             gap = 13,
        col.grid = gray(0.85),
           sides = c(1,2,4) )
apc.lines( apc.m, col="blue", ci=T )
apc.lines( apc.f, col="red" , ci=T )
   matlines( rr.Age[,1], rr.Age[,-1]*rr.ref, lwd=c(3,1,1), col="black", lty=1 )
pc.matlines( rr.Per[,1], rr.Per[,-1]       , lwd=c(3,1,1), col="black", lty=1 )
pc.matlines( rr.Coh[,1], rr.Coh[,-1]       , lwd=c(3,1,1), col="black", lty=1 )
```

6. In order to explicitly fix the knots we just use those from the male `apc` object, then we can construct the design matrices for the effects by first constructing the full ranks and then de-trending them using the `detrend` function:

```
A.kn <- apc.m$Knots$Age ; nk.A <- length(A.kn)
P.kn <- apc.m$Knots$Per ; nk.P <- length(P.kn)
C.kn <- apc.m$Knots$Coh ; nk.C <- length(C.kn)
MA   <- Ns( lung$A, knots=A.kn, intercept=TRUE )
MP   <- Ns( lung$P, knots=P.kn, intercept=TRUE )
MP   <- detrend( MP, lung$P )
MC   <- Ns( lung$C, knots=C.kn, intercept=TRUE )
MC   <- detrend( MC, lung$C )
lung$sex <- factor(lung$sex,labels=c("M","F"))
```

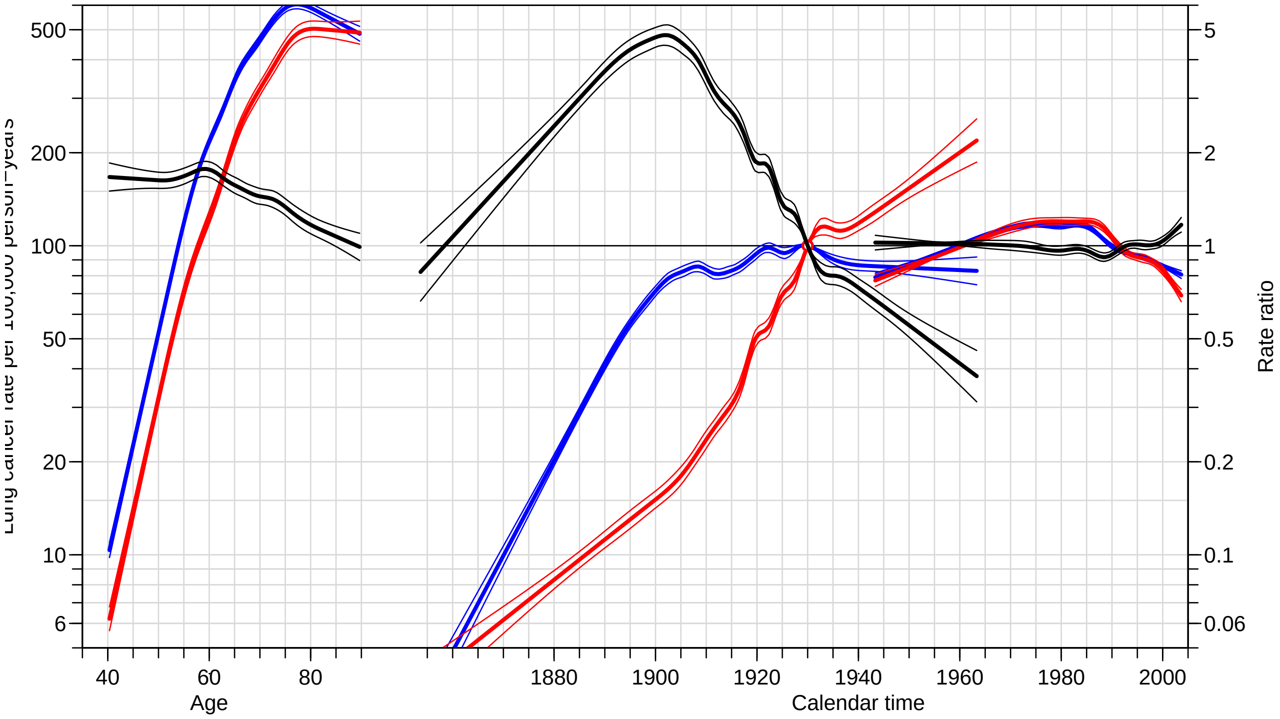

Figure 2.22: *APC estimates for male (blue) and female (red) lung cancer incidence rates in Denmark. The ratio of the APC-effects are in black.* `../graph/lung-sex-apc-3`

With these matrices we can now fit the models we want; the model with sex-interaction on all three variables and the one where we assume identical 2nd order period-effects:

```
mc.int <- glm( D ~ -1 + MA:sex + MP:sex + MC:sex + I(C-1930):sex,
                    offset=log(Y), family=poisson, data=lung )
mp.int <- glm( D ~ -1 + MA:sex + MP:sex + MC:sex + I(P-1980):sex,
                    offset=log(Y), family=poisson, data=lung )
rbind( ci.exp( mc.int, subset="I" ),
       ci.exp( mp.int, subset="I" ) )
```

For the sake of completeness we check if the drift terms also come out identical in the reduced models:

```
mcc.int <- update( mc.int, . ~ . - MC:sex + MC )
mpc.int <- update( mp.int, . ~ . - MC:sex + MC )
rbind( ci.exp( mcc.int, subset="I" ),
       ci.exp( mpc.int, subset="I" ) )
mcp.int <- update( mc.int, . ~ . - MP:sex + MP )
mpp.int <- update( mp.int, . ~ . - MP:sex + MP )
rbind( ci.exp( mcp.int, subset="I" ),
       ci.exp( mpp.int, subset="I" ) )
```

7. We can check if any of the second-order terms are identical between males and females by removing the interaction with sex. This will however only work for the period and the cohort effect, because the intercept and linear effect of age is included with the

age-effect and removing the interaction there would be tantamount to testing whether the absolute levels and the (first order) shape were the same.

So we start by checking whether the period and age-effects have the same second-order properties (i.e. same shape):

```
m.per <- update( mc.int, . ~ . - MP:sex + MP )
m.coh <- update( mc.int, . ~ . - MC:sex + MC )
anova( m.coh, mc.int, m.per, test="Chisq" )
```

Although both effects are significant, there is a much smaller deviance for the period effect, so we might assume that the period-effects have the same shape — this is also apparent from the plot of the RRs seen in figure 2.22.

As goes for the age-effect we can test the same hypothesis, but we want to test a slightly stronger hypothesis, namely that the actual slope with age is the same too, so when we update the model we include the main effect of sex, but *not* the interaction with sex and age; or rather we make successive tests for this:

```
m.age <- update( mc.int, . ~ . - MA:sex + MA + sex + sex:A )
m.aln <- update( m.age, . ~ . - sex:A )
anova( mc.int, m.age, m.aln, test="Chisq" )
```

We see that there quite strong evidence against the hypothesis that the age-effects have the same shape and even stronger that they should have the same "slopes", i.e. first-order shapes too.

8. Thus it seems that a relevant description of the relationship of lung cancer rates between males and females in Denmark is that they follow an age-cohort model. This model is already fitted, but in order to facilitate extraction of the parameters we refit it with a parametrization of the linear cohort effect that gives the difference of these, so it is easier to use a contrast matrix to get it out. Note that we for the convenience of extraction of the interaction effects we have included the intercept in the model — otherwise the parametrization of the `MA:sex` intercept goes wrong:

```
m.RR <- glm( D ~ -1 + MA     + MP + cbind(MC,C-1930) +
                      MA:sex +      cbind(MC,C-1930):sex,
                 offset = log(Y), family=poisson, data=lung )
pr.RR <- predict( m.RR, type="terms", se.fit=TRUE )
str( pr.RR )
dimnames( pr.RR$fit )[[2]]
```

The last two terms are those that we are interested in, so we can just extract the predicted values. But these will have the length (and order!) of the dataset, so we start by finding a set of units, `au`, that correspond to the age-range, and a set of units, `cu`, that correspond to the cohort-range:

```
# Unique ages and cohort
au <- match( sort(unique(lung$A)), lung$A)
cu <- match( sort(unique(lung$C)), lung$C)
```

For these units we derive the the log-RR between males and females. But note the parametrization of the model:

```
ci.lin( m.RR )[,1:2]
```

This indicates that we need to extract not any old unique set of units with cohort values; they must be among the units corresponding to males for the age-effect and to females for the cohort effect::

```
au <- match( sort(unique(lung$A)), lung$A[lung$sex=="M"])
cu <- match( sort(unique(lung$C)), lung$C[lung$sex=="F"])
```

but then we must remember to take this into account when we extract the estimated terms. Note that once we select the columns, we only have a vector left, from which we select the units `au` resp. `cu`:

```
A.term <- exp( cbind(pr.RR$fit   [lung$sex=="M","MA:sex"][au],
                     pr.RR$se.fit[lung$sex=="M","MA:sex"][au]) %*% ci.mat() )
C.term <- exp(-cbind(pr.RR$fit   [lung$sex=="F","cbind(MC, C - 1930):sex"][cu],
                     pr.RR$se.fit[lung$sex=="F","cbind(MC, C - 1930):sex"][cu]) %*% ci.ma
```

Another way is directly to reconstruct the age and the period effects by taking the unique rows of the cohort and age-design matrices and multiply on the parameters of the interaction terms in order to get the log-RRs:

```
# Unique ages and cohort
au <- match( sort(unique(lung$A)), lung$A)
cu <- match( sort(unique(lung$C)), lung$C)
# Corresponding subsets of the design matrices
A.ctr <- MA[au,]
C.ctr <- cbind( MC[cu,], (lung$C-1930)[cu] )
# Parameter names
parnam <- names( coef(m.RR) )
# Have we found the age-parameters we want?
a.par <- intersect( grep("MA",parnam), grep("sexM",parnam) )
parnam[a.par]
# Have we found the cohort-parameters we want?
c.par <- c( grep("MC",parnam), grep("I",parnam) )
c.par <- intersect( c.par, grep("sex",parnam) )
parnam[c.par]
# Then we can extract effects, the parametrization for the cohort
# effect is for F/M, hence we use -C.ctr
A.eff <- ci.lin( m.RR, subset=a.par, ctr.mat= A.ctr, Exp=TRUE )[,5:7]
C.eff <- ci.lin( m.RR, subset=c.par, ctr.mat=-C.ctr, Exp=TRUE )[,5:7]
```

These effects can now be plotted side by side, with the results of the two different approaches on top of each other:

```
par( mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
matplot( lung$A[au], A.eff,
         log="y", ylim=c(0.5,5),
         type="l", lty=1, col="black", lwd=c(3,1,1) )
matlines( lung$A[au], A.term, lty=2, col="red", lwd=c(3,1,1) )
```

```
abline(h=1)
matplot( lung$C[cu], C.eff,
         log="y", ylim=c(0.5,5),
         type="l", lty=1, col="black", lwd=c(3,1,1) )
matlines( lung$C[cu], C.term, lty=2, col="red", lwd=c(3,1,1) )
abline(h=1)
```

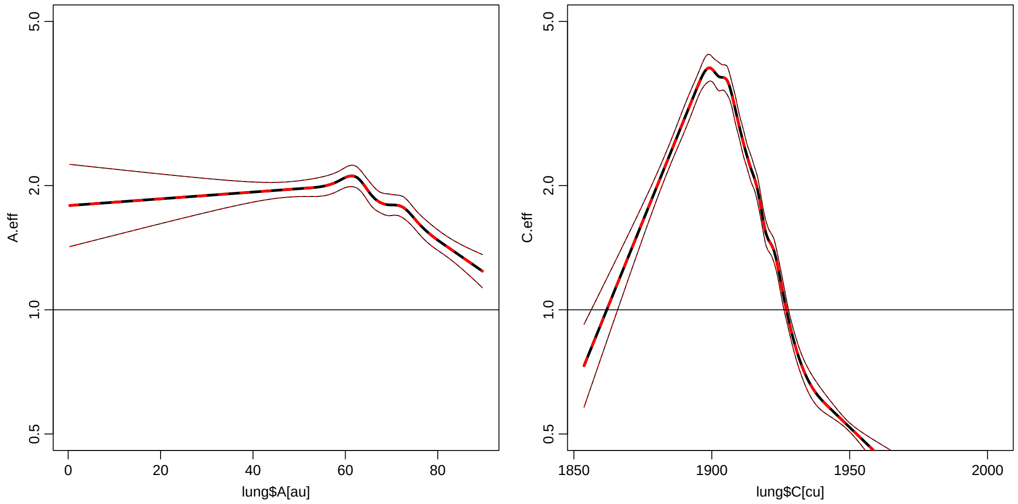

Figure 2.23: *Comparing the M/F rate-ratio between the approach using* `predict.glm` *and the approach using explicit extraction of parameters.*

Now these effects could also be superposed on those from the separate APC-models:

```
par( las=1, mar=c(4,3,1,2), mgp=c(3,1,0)/1.6 )
apc.frame( a.lab = seq(40,90,20),
          cp.lab = seq(1880,2000,20),
           r.lab = c(0.5,1,2,5),
          rr.ref = 1,
           a.tic = seq(35,90,5),
          cp.tic = seq(1855,2005,5),
           r.tic = c(4:9/10,1:6),
         tic.fac = 1.3,
           a.txt = "Age",
          cp.txt = "Calendar time",
           r.txt = "M/F Rate ratio of lung cancer",
          rr.txt = "",
        ref.line = TRUE,
             gap = 13,
        col.grid = gray(0.85),
           sides = c(1,2,4) )
abline( h=1 )
apc.lines( apc.mf, col="black", ci=F, lwd=2 )
   matlines( lung$A[au], A.eff, lwd=c(1,1,1), lty=1, col="blue" )
pc.matlines( lung$C[cu], C.eff, lwd=c(1,1,1), lty=1, col="blue" )
```

A note on the reference point A short glance at figure 2.24 shows that we have not got what we wanted; the cohort RR is not centered at 1930. We have not done anything

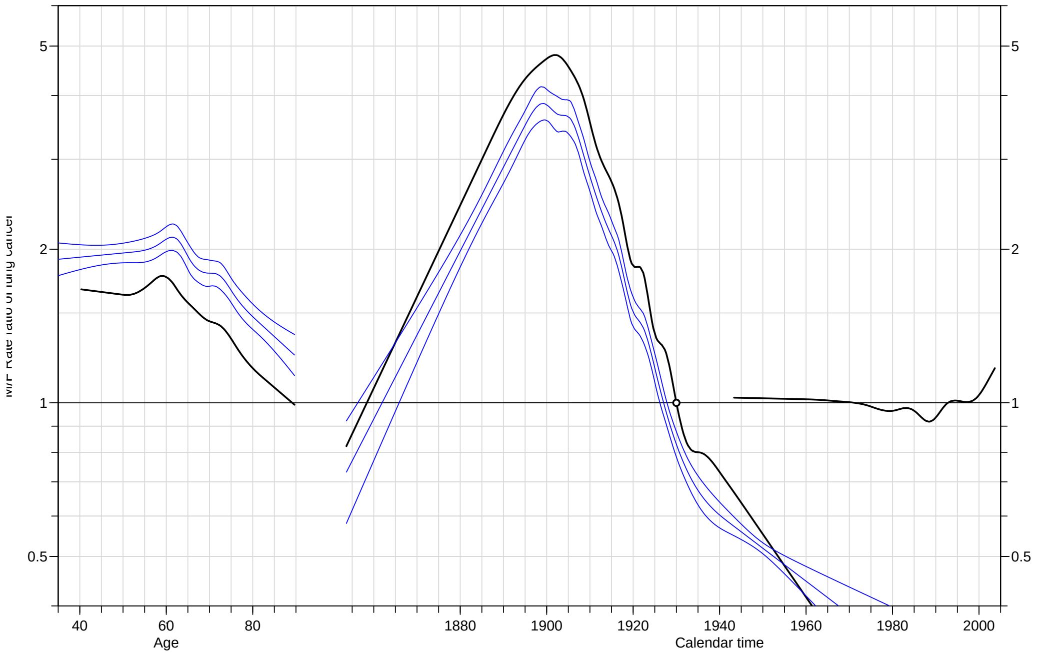

Figure 2.24: *Comparing the M/F rate-ratio between the simple approach and the approach using an explicit model.*

to achieve this; the choice of the reference point requires a bit extra work when we have splines in the model, because splines do not provide an explicit reference we can extract.

The trick is to take the cohort design matrix (as generated by `ns()`) and subtract a matrix where all rows are identical, corresponding to `ns(1930,...)`. In this case it is quite straightforward, because we fit an APC-model to females and then add RRs for males which are just an age-effect and a cohort effect centered at 1930. So we just reparametrize the model with two new matrices for the RRs. We define the interaction matrices as matrices for the age and cohort effects, but where all rows corresponding to females are 0. The trick is to use the column-major storage of elements in matrices. When we use the `*` operator on matrices they are treated as vectors, and since the vector `(lung$sex=="M")` is shorter this is recycled, so that precisely all rows in `MA` and `MC` corresponding to women are set to 0:

```
maleA <- ns( lung$A, knots=A.kn[-c(1,nk.A)], Bo=A.kn[c(1,nk.A)], intercept=TRUE ) *
          (lung$sex=="M")
maleC <- ( ns(             lung$C, knots=C.kn[-c(1,nk.C)], Bo=C.kn[c(1,nk.C)] ) -
            ns( rep(1930,nrow(lung)), knots=C.kn[-c(1,nk.C)], Bo=C.kn[c(1,nk.C)] ) ) *
          (lung$sex=="M")
```

To get the estimated RRs we define the contrast matrices similarly:

```
A.pt <- 40:90
C.pt <- 1860:1960
ctr.A <- ns(                 A.pt  , knots=A.kn[-c(1,nk.A)], Bo=A.kn[c(1,nk.A)],
                                     intercept=TRUE )
ctr.C <- ns(                 C.pt  , knots=C.kn[-c(1,nk.C)], Bo=C.kn[c(1,nk.C)] ) -
         ns( rep(1930,length(C.pt)), knots=C.kn[-c(1,nk.C)], Bo=C.kn[c(1,nk.C)] )
```

Hence we can now just use these two matrices in the specification of the model and then extract the parameters corresponding to them, to get the desired effects:

```
M.RR <- glm( D ~ -1 + MA      + MP + cbind(MC,C-1930) +
                      maleA + maleC,
                 offset = log(Y), family=poisson, data=lung )
A.eff <- ci.lin( M.RR, subset="maleA", ctr.mat=ctr.A, E=T )[,5:7]
C.eff <- ci.lin( M.RR, subset="maleC", ctr.mat=ctr.C, E=T )[,5:7]
```

```
par( las=1, mar=c(4,3,1,2), mgp=c(3,1,0)/1.6 )
apc.frame( a.lab = seq(40,90,20),
          cp.lab = seq(1880,2000,20),
           r.lab = c(0.5,1,2,5),
          rr.ref = 1,
           a.tic = seq(35,90,5),
          cp.tic = seq(1855,2005,5),
           r.tic = c(4:9/10,1:6),
         tic.fac = 1.3,
           a.txt = "Age",
          cp.txt = "Calendar time",
           r.txt = "M/F Rate ratio of lung cancer",
          rr.txt = "",
        ref.line = TRUE,
             gap = 13,
        col.grid = gray(0.85),
           sides = c(1,2,4) )
abline( h=1 )
apc.lines( apc.mf, col="black", ci=TRUE, lwd=c(2,1,1) )
   matlines( A.pt, A.eff, lwd=c(3,1,1), lty=1, col="blue" )
pc.matlines( C.pt, C.eff, lwd=c(3,1,1), lty=1, col="blue" )
```

In figure 2.25 we now have the estimated M/F RRs in blue from a model where we assume that the calendar time effect is identical for men and women. Is is clear that men have higher incidence rates than women, particularly in ages around 50, but also that major generational effects is at stake — men were increasing rates of lung cancer relative to women until birth cohorts around 1900, then a major catch-up has been made by women. The cohorts in the 1950s have a M/F RR of 0.6 relative to the 1930 cohort, which is the one used for the age-specific RRs. The age-specific RRs are all below 1.75; and so since $1.75 \times 0.6 = 1.05$, we can conclude that with the exception of ages just around 50, women in the generations born after 1950 have higher lung cancer rates than men from the same generations.

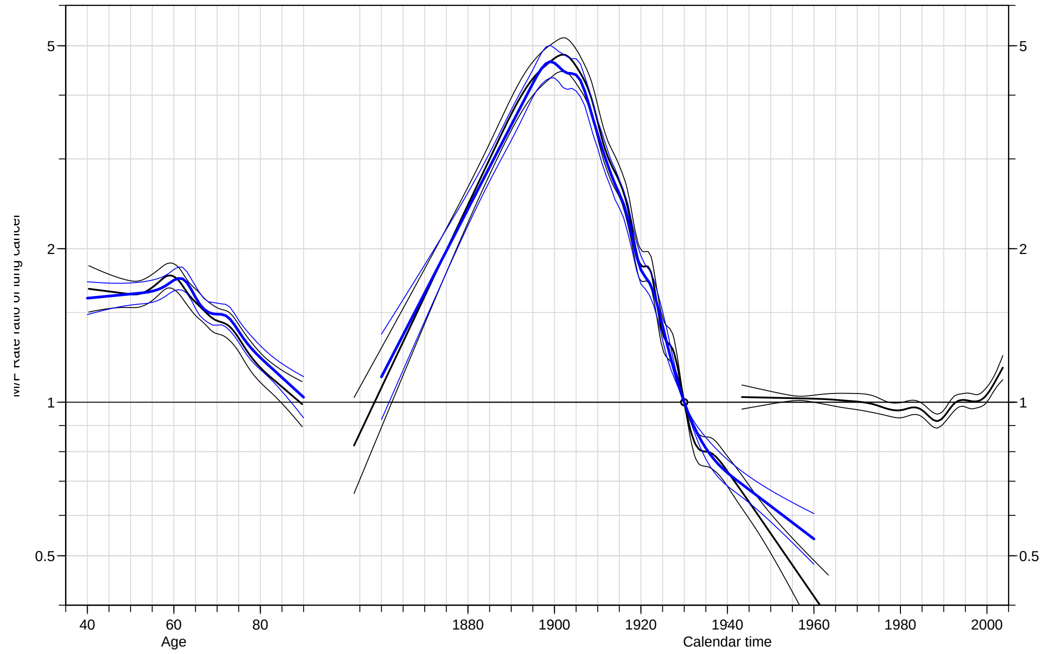

Figure 2.25: *Comparing the M/F rate-ratio between the simple approach and the approach using an explicit model.*

2.7 Histological subtypes of testis cancer

1. First we load the data, restrict to two main types, and to the relevant age-range, and for convenience also rename the variables:

```
th <- read.table("../data/testis-hist.txt", header = TRUE)
th <- rename(th, "A" = "age",
                 "P" = "diag",
                 "D" = "d",
                 "Y" = "y")
names(th)
```

```
[1] "a"     "p"     "c"     "Y"     "A"     "P"     "birth" "hist"  "D"
```

2. We restrict the data set to the two main histological types and the relevant age-range .

```
th <- subset(th, hist != 3 & A > 15 & A < 65)
table(th$hist)
```

```
   1    2
5400 5400
```

```
th <- transform(th, hist = factor(hist,
                                  labels = c("Seminoma", "non-Semi")))
str(th)
```

```
'data.frame':        10800 obs. of  9 variables:
 $ a    : int  15 15 15 15 16 16 16 16 17 17 ...
 $ p    : int  1943 1943 1943 1943 1943 1943 1943 1943 1943 1943 ...
 $ c    : int  1927 1927 1928 1928 1926 1926 1927 1927 1925 1925 ...
 $ Y    : num  15684 15684 15504 15504 16017 ...
 $ A    : num  15.7 15.7 15.3 15.3 16.7 ...
 $ P    : num  1943 1943 1944 1944 1943 ...
 $ birth: num  1928 1928 1928 1928 1927 ...
 $ hist : Factor w/ 2 levels "Seminoma","non-Semi": 1 2 1 2 1 2 1 2 1 2 ...
 $ D    : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
head(th)
```

```
    a    p    c        Y        A        P    birth     hist D
91 15 1943 1927 15683.67 15.66667 1943.333 1927.667 Seminoma 0
92 15 1943 1927 15683.67 15.66667 1943.333 1927.667 non-Semi 0
94 15 1943 1928 15504.33 15.33333 1943.667 1928.333 Seminoma 0
95 15 1943 1928 15504.33 15.33333 1943.667 1928.333 non-Semi 0
97 16 1943 1926 16017.00 16.66667 1943.333 1926.667 Seminoma 0
98 16 1943 1926 16017.00 16.66667 1943.333 1926.667 non-Semi 0
```

Finally we also make a quick overview over the number of cases and person-years for each histological subtype. Note that the person-years are identical between the different histological types:

```
ftable(xtabs(cbind(D, Y) ~ Agr + hist,
             data = transform(th, Agr = cut(A, seq(15, 65, 5), right = FALSE))),
       row.vars = 1)
```

```
        hist Seminoma         non-Semi
                D       Y         D       Y
Agr
[15,20)        28 9866173      268 9866173
[20,25)       194 9782823      727 9782823
[25,30)       572 9561920      848 9561920
[30,35)       902 9263680      634 9263680
[35,40)       908 8954294      401 8954294
[40,45)       692 8606038      266 8606038
[45,50)       475 8139267      161 8139267
[50,55)       343 7443401       85 7443401
[55,60)       215 6740090       72 6740090
[60,65)       132 5997263       32 5997263
```

2.7.1 The age-incidence crossover

This is a little extra, paraphrasing the age-incidence cross-over that has been discussed in the article: "Age-Related Crossover in Breast Cancer Incidence Rates Between Black and White Ethnic Groups" by William F. Anderson , Philip S. Rosenberg , Idan Menashe , Aya Mitani & Ruth M. Pfeiffer, JNCI, 100, 24, December 17, 2008.

To see what it is all about, we fit APC-models separately for seminoma and non-seminoma, using different parametrizations. We also compute the age-specific rate-ratio between seminoma and non-seminoma and see when they cross. To this end we first define a small function that takes effects from two apc objects as input, and return the rate-ratios in the shape of a similar object.

```
rr <- function(one, two) cbind(one[,1], ci.ratio(one[,-1], two[,-1]))
```

Then we fit APC-models separately for the seminomas and non-seminomas, using two different parametrizations for each — the only difference being the reference point for the cohort; either 1945 or 1920.

```
sem.1945 <- apc.fit( subset(th, hist=="Seminoma"),
                     ref.c=1945,
                     npar=c(8,5,15), scale=10^5 )

[1] "ML of APC-model Poisson with log(Y) offset : ( ACP ):\n"
              Model      AIC Mod. df. Mod. dev. Test df.  Test dev.      Pr(>Chi)
1               Age 11673.66     5392  5677.477       NA         NA            NA
2         Age-drift 11072.33     5391  5074.144        1 603.333150 3.153666e-133
3        Age-Cohort 11051.06     5378  5026.880       13  47.263479  8.720862e-06
4 Age-Period-Cohort 11043.02     5375  5012.836        3  14.043830  2.846095e-03
5        Age-Period 11074.96     5388  5070.776       13  57.939233  1.223573e-07
6         Age-drift 11072.33     5391  5074.144        3   3.368075  3.382796e-01
  Test dev/df    H0
1          NA
2  603.333150 zero drift
3    3.635652 Coh eff|dr.
4    4.681277 Per eff|Coh
5    4.456864 Coh eff|Per
6    1.122692 Per eff|dr.
```

```
n.s.1945 <- apc.fit( subset(th, hist=="non-Semi"),
                     ref.c=1945,
                     npar=c(8,5,15), scale=10^5 )
```

```
[1] "ML of APC-model Poisson with log(Y) offset : ( ACP ):\n"
              Model       AIC Mod. df. Mod. dev. Test df. Test dev.      Pr(>Chi)
1               Age 10015.477     5392  5202.544       NA        NA            NA
2         Age-drift  9316.399     5391  4501.466        1 701.07777 1.743153e-154
3        Age-Cohort  9287.616     5378  4446.683       13  54.78360  4.410070e-07
4 Age-Period-Cohort  9211.341     5375  4364.408        3  82.27482  9.977226e-18
5        Age-Period  9250.661     5388  4429.728       13  65.32035  5.765780e-09
6         Age-drift  9316.399     5391  4501.466        3  71.73807  1.811437e-15
  Test dev/df     H0
1          NA
2  701.077773 zero drift
3    4.214123 Coh eff|dr.
4   27.424939 Per eff|Coh
5    5.024642 Coh eff|Per
6   23.912690 Per eff|dr.
```

```
sem.1920 <- apc.fit( subset(th, hist=="Seminoma"),
                     ref.c=1920,
                     npar=c(8,5,15), scale=10^5 )
```

```
[1] "ML of APC-model Poisson with log(Y) offset : ( ACP ):\n"
              Model      AIC Mod. df. Mod. dev. Test df.  Test dev.      Pr(>Chi)
1               Age 11673.66     5392  5677.477       NA         NA            NA
2         Age-drift 11072.33     5391  5074.144        1 603.333150 3.153666e-133
3        Age-Cohort 11051.06     5378  5026.880       13  47.263479  8.720862e-06
4 Age-Period-Cohort 11043.02     5375  5012.836        3  14.043830  2.846095e-03
5        Age-Period 11074.96     5388  5070.776       13  57.939233  1.223573e-07
6         Age-drift 11072.33     5391  5074.144        3   3.368075  3.382796e-01
  Test dev/df     H0
1          NA
2  603.333150 zero drift
3    3.635652 Coh eff|dr.
4    4.681277 Per eff|Coh
5    4.456864 Coh eff|Per
6    1.122692 Per eff|dr.
```

```
n.s.1920 <- apc.fit( subset(th, hist=="non-Semi"),
                     ref.c=1920,
                     npar=c(8,5,15), scale=10^5 )
```

```
[1] "ML of APC-model Poisson with log(Y) offset : ( ACP ):\n"
              Model       AIC Mod. df. Mod. dev. Test df. Test dev.      Pr(>Chi)
1               Age 10015.477     5392  5202.544       NA        NA            NA
2         Age-drift  9316.399     5391  4501.466        1 701.07777 1.743153e-154
3        Age-Cohort  9287.616     5378  4446.683       13  54.78360  4.410070e-07
4 Age-Period-Cohort  9211.341     5375  4364.408        3  82.27482  9.977226e-18
5        Age-Period  9250.661     5388  4429.728       13  65.32035  5.765780e-09
6         Age-drift  9316.399     5391  4501.466        3  71.73807  1.811437e-15
  Test dev/df     H0
1          NA
2  701.077773 zero drift
3    4.214123 Coh eff|dr.
4   27.424939 Per eff|Coh
5    5.024642 Coh eff|Per
6   23.912690 Per eff|dr.
```

We can now use these objects to compute the RR of the estimated age- period- and cohort-effects:

```
rrA.1945 <- rr( sem.1945$Age, n.s.1945$Age )
rrA.1920 <- rr( sem.1920$Age, n.s.1920$Age )
rrP.1945 <- rr( sem.1945$Per, n.s.1945$Per )
rrP.1920 <- rr( sem.1920$Per, n.s.1920$Per )
rrC.1945 <- rr( sem.1945$Coh, n.s.1945$Coh )
rrC.1920 <- rr( sem.1920$Coh, n.s.1920$Coh )
```

We can now make a plot with the two subtypes plotted in different colors and and the two parametrizations plotted by different line types. We note that since we have chosen the period effects to be 0 on average with 0 slope, they are identical for the two parametrizations.

```
par(mar=c(3, 4, 0.5, 2))
apc.frame(r.lab = c(c(      10)/100,
                    c(2, 5, 10)/10,
                    c(2, 5, 10, 15)),
          r.tic = c(c(1:10)/10,
                    c(2:10)),
          rr.ref = 1,
          a.lab = seq(10, 70, 20),
          a.tic = 1:7*10,
          cp.lab = seq(1880, 2000, 20),
          cp.tic = 188:200*10,
          gap = 5 )
apc.lines(sem.1945, col = "blue", lwd = 2)
apc.lines(n.s.1945, col = "red" , lwd = 2)
apc.lines(sem.1920, col = "blue", lty = "21", lend = "butt", lwd = 3)
apc.lines(n.s.1920, col = "red" , lty = "21", lend = "butt", lwd = 3)
```

```
par(mar=c(3, 4, 0.5, 2))
apc.frame(r.lab = c(c(      10)/100,
                    c(2, 5, 10)/10,
                    c(2, 5, 10, 15)),
          r.tic = c(c(1:10)/10,
                    c(2:10)),
          rr.ref = 1,
          a.lab = seq(10, 70, 20),
          a.tic = 1:7*10,
          cp.lab = seq(1880, 2000, 20),
          cp.tic = 188:200*10,
          gap = 5 )
   lines( rrA.1945[,1], rrA.1945[,2], lwd=2 )
   lines( rrA.1920[,1], rrA.1920[,2], lwd=2, lty="22" )
pc.lines( rrP.1945[,1], rrP.1945[,2], lwd=2, col=gray(0.5) )
pc.lines( rrP.1920[,1], rrP.1920[,2], lwd=2, col=gray(0.5), lty="22" )
pc.lines( rrC.1945[,1], rrC.1945[,2], lwd=2 )
pc.lines( rrC.1920[,1], rrC.1920[,2], lwd=2, lty="22" )
abline(h=1)
```

It is seen that the two age-specific rate-ratios are 1 at different ages, although they are derived from the same model(s). The differance (on the log scale) of the age-specific

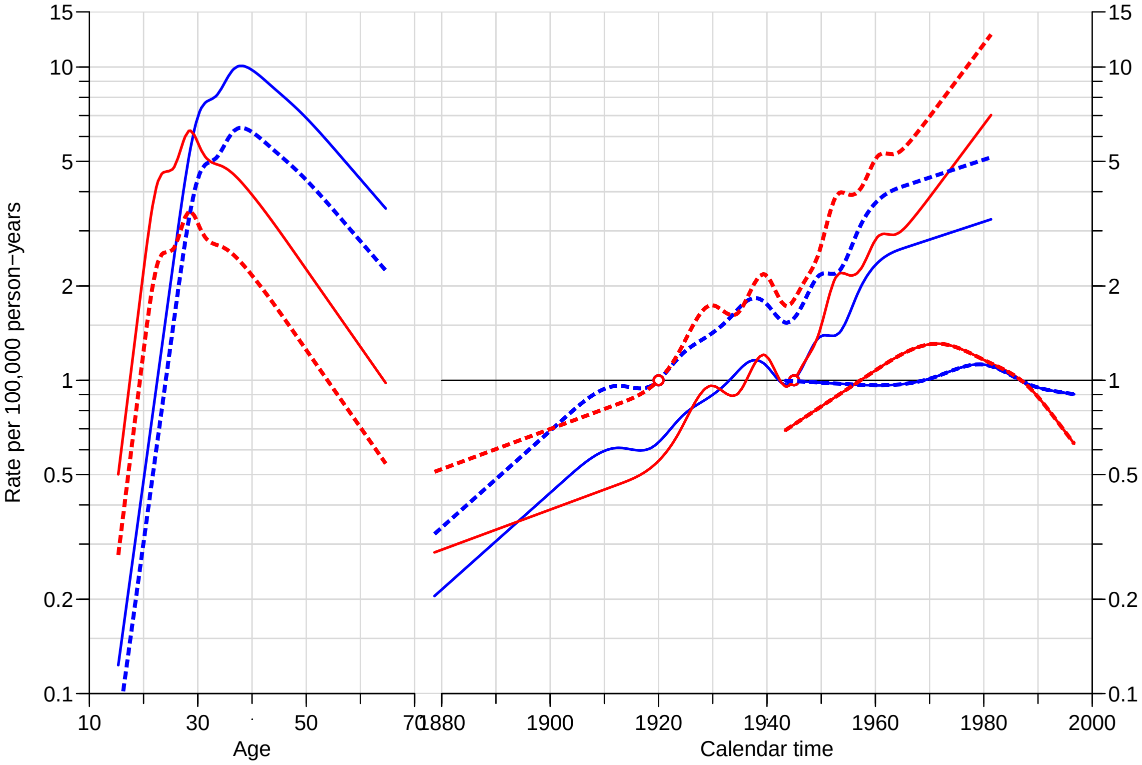

Figure 2.26: *Estimated age-, period- and cohort-effects for Seminoma (blue) and non-Seminoma (red), using either 1920 or 1945 as the reference cohort.*

RRs is the opposite of the difference of the cohort RRs. So statements about one histological subtype being more frequent than the other are not meaningful in the context of an age-perido-cohort model.

The reason is that if the rates of seminoma and non-seminoma both follow an APC-model (different parameters, of course), then the RR between the two will also follow an APC-model. And you will have to make exactly the same decisions for the rate-ratios as for any of the two separate models. The example here illustrates that the restriction on the period-effect to be 0 on average with 0 slope carries over to the RR. Hence, it might be more productive to constrain *both* the cohort and the period effects to be 0 on average, and take out the drift as a separate parameter for each subtype.

```
sem.dr <- apc.fit(subset(th, hist == "Seminoma"),
                  parm = "AdCP", ref.c = 1930,
                  npar = c(8, 5, 15), scale = 10^5 )
```

```
[1] "ML of APC-model Poisson with log(Y) offset : ( ADCP ):\n"
              Model      AIC Mod. df. Mod. dev. Test df.  Test dev.      Pr(>Chi)
1               Age 11673.66     5392  5677.477       NA         NA            NA
2         Age-drift 11072.33     5391  5074.144        1 603.333150 3.153666e-133
3        Age-Cohort 11051.06     5378  5026.880       13  47.263479  8.720862e-06
4 Age-Period-Cohort 11043.02     5375  5012.836        3  14.043830  2.846095e-03
5        Age-Period 11074.96     5388  5070.776       13  57.939233  1.223573e-07
6         Age-drift 11072.33     5391  5074.144        3   3.368075  3.382796e-01
  Test dev/df    H0
1          NA
2  603.333150 zero drift
```

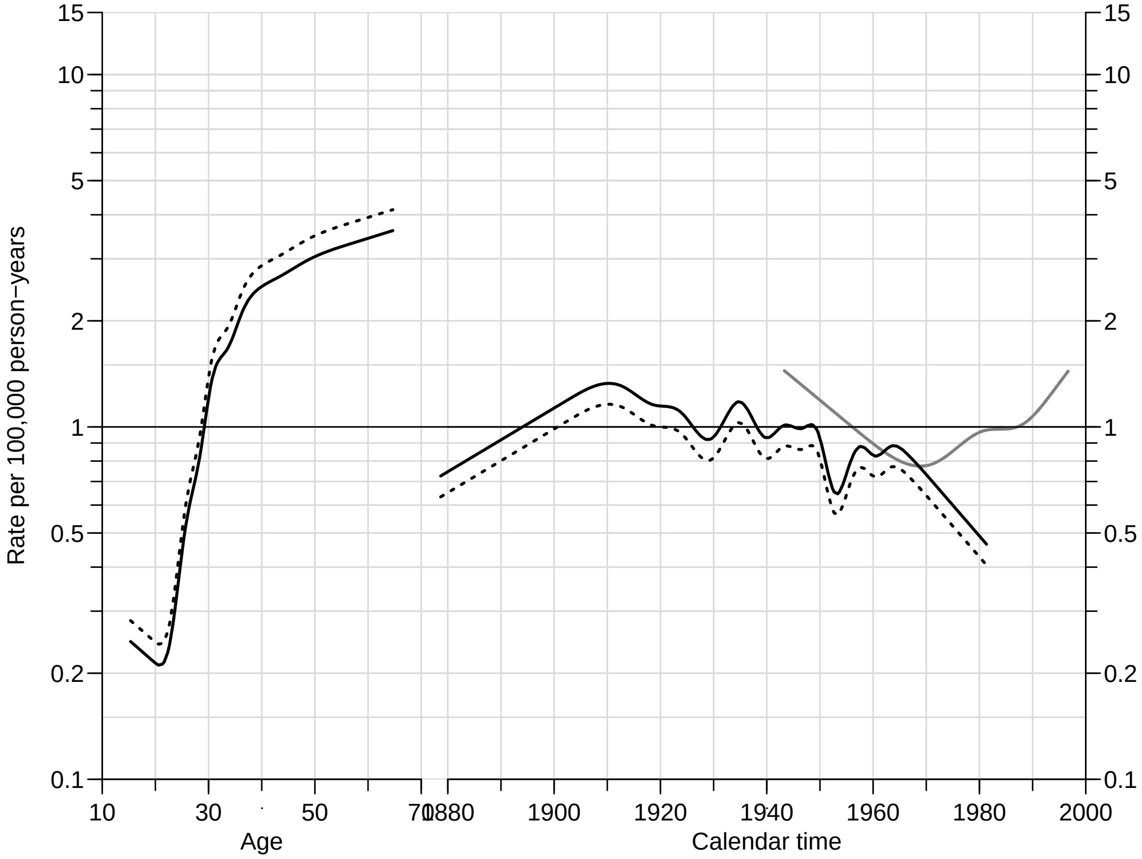

Figure 2.27: *Estimated RRs between the effects for Seminoma versus non-seminoma.*

```
3    3.635652 Coh eff|dr.
4    4.681277 Per eff|Coh
5    4.456864 Coh eff|Per
6    1.122692 Per eff|dr.
```

```
 n.s.dr <- apc.fit(subset(th, hist == "non-Semi"),
                   parm = "AdCP", ref.c = 1930,
                   npar = c(8, 5, 15), scale = 10^5 )
```

```
[1] "ML of APC-model Poisson with log(Y) offset : ( ADCP ):\n"
              Model       AIC Mod. df. Mod. dev. Test df. Test dev.      Pr(>Chi)
1               Age 10015.477     5392  5202.544       NA        NA            NA
2         Age-drift  9316.399     5391  4501.466        1 701.07777 1.743153e-154
3        Age-Cohort  9287.616     5378  4446.683       13  54.78360  4.410070e-07
4 Age-Period-Cohort  9211.341     5375  4364.408        3  82.27482  9.977226e-18
5        Age-Period  9250.661     5388  4429.728       13  65.32035  5.765780e-09
6         Age-drift  9316.399     5391  4501.466        3  71.73807  1.811437e-15
  Test dev/df    H0
1          NA
2  701.077773 zero drift
3    4.214123 Coh eff|dr.
4   27.424939 Per eff|Coh
5    5.024642 Coh eff|Per
6   23.912690 Per eff|dr.
```

Using `parm="AdCP"` gives estimates of cohort and period effects that are constrained this way, and of age-effects referring to a cohort as given by the `ref.c`. Note that it is necessary to fix a reference cohort (or period) if we want age-specific rates estimated.

We can then formally test whether the drift parameter is the same for the two histological subtypes by computing the ratio of the drifts with a c.i. If we look at the drift component of the `apc.fit` object:

```
str(sem.dr$Drift)
```

```
num [1:2, 1:3] 1.03 1.02 1.02 1.02 1.03 ...
- attr(*, "dimnames")=List of 2
  ..$ : chr [1:2] "APC (Y-weights)" "A-d"
  ..$ : chr [1:3] "exp(Est.)" "2.5%" "97.5%"
```

we see that it is a 2×3 matrix. The function `rr` we defined takes two 4-column matrices as input, so this is what se will supply:

```
round( rbind(sem.dr$Drift, n.s.dr$Drift), 4)
```

```
                exp(Est.)   2.5%  97.5%
APC (Y-weights)    1.0260 1.0235 1.0285
A-d                1.0247 1.0226 1.0267
APC (Y-weights)    1.0326 1.0290 1.0362
A-d                1.0309 1.0284 1.0333
```

```
round((rbind(sem.dr$Drift, n.s.dr$Drift) - 1) * 100, 2)
```

```
                exp(Est.) 2.5% 97.5%
APC (Y-weights)      2.60 2.35  2.85
A-d                  2.47 2.26  2.67
APC (Y-weights)      3.26 2.90  3.62
A-d                  3.09 2.84  3.33
```

We see that the drift for seminoma is an increase of 2.6% per year, but for non-seminoma about 3.3% per year. Thus we see that there are indeed different drifts between the two subtypes.

We can then separately look at whether the *shapes* of the RRs by cohort and period are the same. By looking at the confidence interval for the ratios of the cohort and period effects we can assess whether they are the same. A formal test can be made by fitting a joint model.

```
rrA <- rr(sem.dr$Age, n.s.dr$Age)
rrP <- rr(sem.dr$Per, n.s.dr$Per)
rrC <- rr(sem.dr$Coh, n.s.dr$Coh)
apc.frame(r.lab=c(c(  5,10)/100,
                  c(2,5,10)/10,
                  c(2,5,10,15)),
          r.tic=c(c(5:10)/100,
                  c(2:10)/10,
                  c(2:10)),
          rr.ref=1,
          a.lab=seq(20,60,20),
          a.tic=1:7*10,
          cp.lab=seq(1880,2000,20),
          cp.tic=188:200*10,
       r.txt="Seminoma vs. non-Seminoma RR",
       gap=5 )
  matshade(rrA[,1], rrA[,-1], lwd=3, lty=1)
pc.matshade(rrP[,1], rrP[,-1], lwd=3, lty=1)
pc.matshade(rrC[,1], rrC[,-1], lwd=3, lty="21", lend = "butt")
abline(h=1)
pc.points(1930, 1, pch = 16, cex = 1.5, col = "white")
pc.points(1930, 1, pch = 1, lwd = 3, cex = 1.5)
```

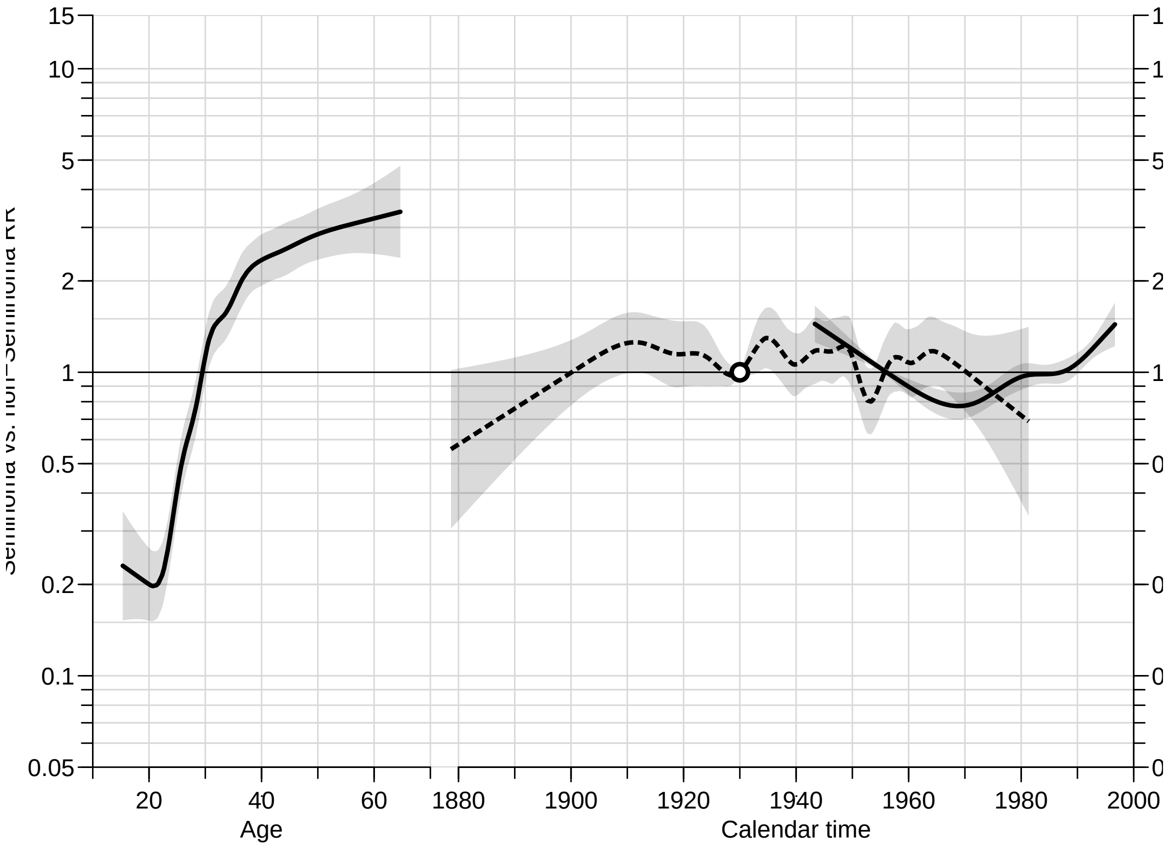

Figure 2.28: *Estimated ratios of the non-linear age-, period- and cohort-effects for Seminoma versus non-Seminoma, using 1930 as the reference cohort.*

Hence the concept of the age-incidence cross-over is only well defined if you are prepared to make assumptions about identity of cohort and period affects at certain timepoints (such as for example *all* timepoints).

2.8 Prediction of breast cancer rates

1. First we read the data and take an overview:

```
library( Epi )
breast <- read.table("../data/breast.txt", header=T )
str( breast )
```

```
'data.frame':        10980 obs. of  5 variables:
 $ A: int  0 0 0 0 0 0 0 0 0 0 ...
 $ P: int  1943 1943 1944 1944 1945 1945 1946 1946 1947 1947 ...
 $ C: int  1942 1943 1943 1944 1944 1945 1945 1946 1946 1947 ...
 $ D: int  0 0 0 0 0 0 0 0 0 0 ...
 $ Y: num  18649 19947 19854 21265 21236 ...
```

```
summary( breast )
```

```
       A              P              C              D                Y          
 Min.   : 0.0   Min.   :1943   Min.   :1853   Min.   : 0.00   Min.   :  385.2  
 1st Qu.:22.0   1st Qu.:1958   1st Qu.:1905   1st Qu.: 0.00   1st Qu.:11059.5  
 Median :44.5   Median :1973   Median :1928   Median : 9.00   Median :14538.3  
 Mean   :44.5   Mean   :1973   Mean   :1928   Mean   :12.11   Mean   :13555.2  
 3rd Qu.:67.0   3rd Qu.:1988   3rd Qu.:1951   3rd Qu.:21.00   3rd Qu.:17767.2  
 Max.   :89.0   Max.   :2003   Max.   :2003   Max.   :69.00   Max.   :22549.0  
```

2. The variables A, P and C are just the left end points of the 1-year classes forming the Lexis triangles, so we must replace these with the correct triangle means. Recall that the upper triangles are characterized by the cohort being from the previous year, i.e. that $p - a - c = 1$.

```
breast <- transform(breast, up = P - A - C )
breast <- transform(breast, A = A + (1 + up) / 3,
                            P = P + (2 - up) / 3,
                            C = C + (1 + up) / 3 )
with(breast, summary(P - A - C))
```

```
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max. 
-2.274e-13 -2.274e-13  0.000e+00  0.000e+00  2.274e-13  2.274e-13 
```

```
head(breast)
```

```
          A        P        C D        Y up
1 0.6666667 1943.333 1942.667 0 18648.83  1
2 0.3333333 1943.667 1943.333 0 19946.50  0
3 0.6666667 1944.333 1943.667 0 19853.67  1
4 0.3333333 1944.667 1944.333 0 21265.00  0
5 0.6666667 1945.333 1944.667 0 21235.67  1
6 0.3333333 1945.667 1945.333 0 22407.00  0
```

3. In order to use `ratetab` we must produce a matrix classified by age and period in suitable intervals. This can be done choosing a tabulation interval length and then using this in producing the tables. This approach enables a simple way of experimenting with the length. Figure **??** shows the results.

```
ti <- 4
rt <- with(subset(breast, A>30 ),
           tapply(D, list(floor( A      /ti)*ti+ti/2,
                          floor((P-1943)/ti)*ti+ti/2+1943), sum ) /
           tapply(Y, list(floor( A      /ti)*ti+ti/2,
                          floor((P-1943)/ti)*ti+ti/2+1943), sum ) * 10^5 )
par( mfrow=c(2,2), mar=c(3,3,0,0), oma=c(0,0,1,1), mgp=c(3,1,0)/1.6 )
rateplot( rt, which= c( "ap", "ac", "pa", "ca"),
          col=heat.colors(22), ann=TRUE )
```

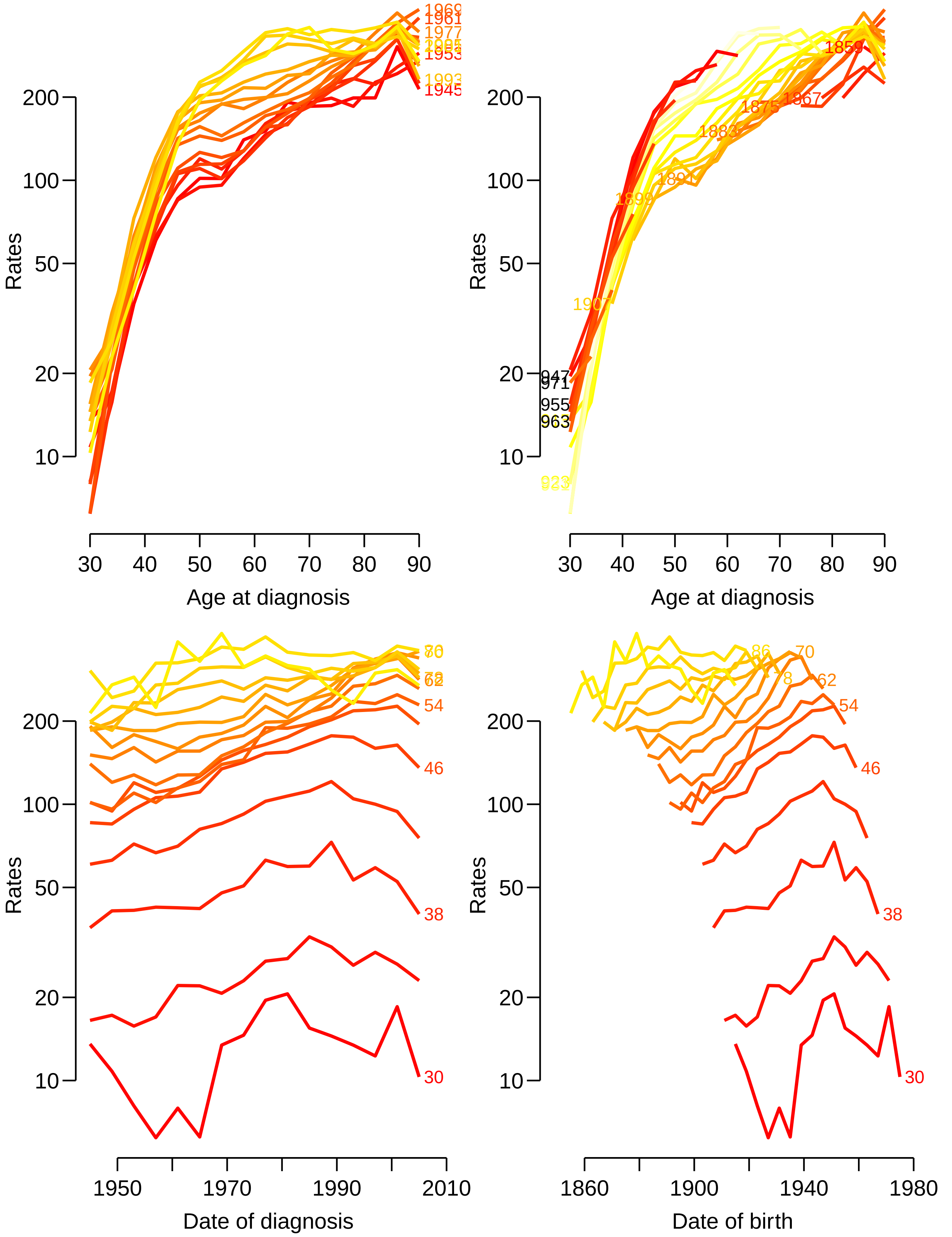

Figure 2.29: *Danish breast cancer rates in 4-year age and period intervals.* `../graph/brcapr-ratetab`

4. We use `apc.fit` to fit a model with age, period and cohort effects as natural splines (the default), and the `plot` method for apc objects to plot the estimated effects:

```
par( mfrow=c(1,1), mar=c(3,4,1,3) )
m1 <- apc.fit(subset(breast, A > 30),
              npar = c(8, 6, 10),
             scale = 10^5,
             ref.c = 1920)
```

```
[1] "ML of APC-model Poisson with log(Y) offset : ( ACP ):\n"
             Model      AIC Mod. df. Mod. dev. Test df. Test dev.      Pr(>Chi)
1              Age 48805.35     7312 16427.665       NA        NA            NA
2        Age-drift 42743.95     7311 10364.264        1 6063.4011  0.000000e+00
3       Age-Cohort 41693.09     7303  9297.401        8 1066.8631 5.484678e-225
4 Age-Period-Cohort 41611.85     7299  9208.167        4   89.2344  1.914805e-18
5       Age-Period 42655.51     7307 10267.825        8 1059.6585 1.971502e-223
6        Age-drift 42743.95     7311 10364.264        4   96.4390  5.632164e-20
  Test dev/df     H0
1          NA
2  6063.40113 zero drift
3   133.35789 Coh eff|dr.
4    22.30860 Per eff|Coh
5   132.45731 Coh eff|Per
6    24.10975 Per eff|dr.
```

```
names(m1)
```

```
[1] "Type"   "Model" "Age"    "Per"    "Coh"    "Drift" "Ref"    "Anova" "Knots"
```

```
m1$Knots
```

```
$Age
[1] 41.33333 48.33333 53.66667 59.33333 64.33333 69.66667 75.33333 82.66667

$Per
[1] 1953.333 1968.333 1978.667 1987.333 1994.333 2000.667

$Coh
 [1] 1883.333 1896.333 1904.333 1910.667 1916.333 1921.667 1927.667 1934.333 1941.667
[10] 1950.667
```

```
plot(m1)
```

```
cp.offset    RR.fac
     1764       100
```

The plot (figure 2.30) is not impressive, so we fine-tune the details by defining them explicit in `apc.frame`. This piece of code is made by copying the definition of all parameters from the help page and successively filling them in with suitable values:

```
par( las=1, mar=c(3,4,1,4), mgp=c(3,1,0)/1.5 )
fp <-  apc.frame( a.lab = seq(30,90,10),
                 cp.lab = seq(1860,2005,20),
                  r.lab = c(c(1,2,5)*10,c(1,2,5)*100),
#                 rr.lab = r.lab / rr.ref,
                 rr.ref = 100,
                  a.tic = seq(30,90,5),
                 cp.tic = seq(1855,2005,5),
```

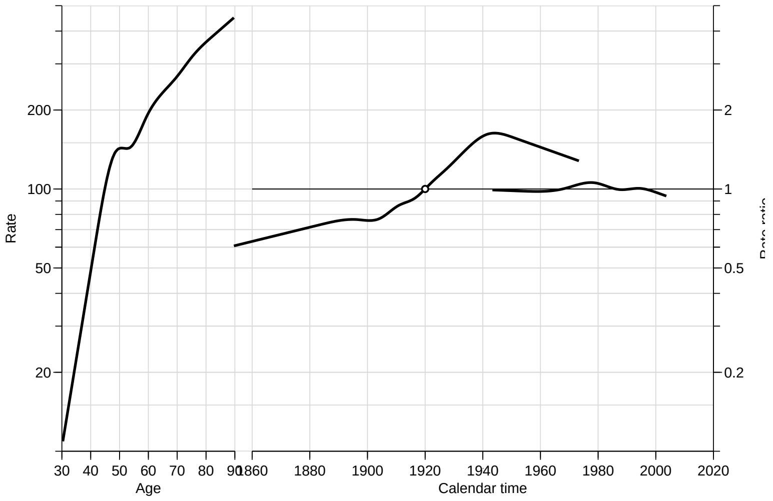

Figure 2.30: *Estimates of age- period- and cohort effects plotted the default way. Note that Clemmesen's hook shows up very clearly in the age-effect.* `../graph/brcapr-apcfit-1`

```
                  r.tic = c(9,1:9*10,1:5*100),
#                 rr.tic = r.tic / rr.ref,
                tic.fac = 1.3,
                  a.txt = "Age",
                 cp.txt = "Calendar time",
                  r.txt = "Rate per 100,000 person-years",
                 rr.txt = "Rate ratio",
                    gap = 8,
               col.grid = gray(0.85),
                  sides = c(1,2,4) )
# lines( m1, ci=T, col="red" )
   matshade(m1$Age[,1], m1$Age[,-1], col="red", lwd = 2 )
pc.matshade(m1$Per[,1], m1$Per[,-1], col="red", lwd = 2 )
pc.matshade(m1$Coh[,1], m1$Coh[,-1], col="red", lwd = 2, lty = "21")
pc.points( 1920, 1, pch=16, col="red" )
```

5. In order to extend the period and cohort effects beyond the range where we have data support (that is the range available in the elements `Age`, `Per` and `Coh` of the `apc` object `m1`), we first define the prediction points and the anchor points on the period scale. We could use arbitrary anchor points, or we could use the last knot and the highest observed period/cohort, and use the property that the natural splines are linear beyond the last knot.

   This is simply using the fitted model beyond the observed data, so predicting rates becomes very simple this way.

   We illustrate the parameter extrapolations used we must find the last knot and the last point (well, any point beyond the last knot), use these as anchor points and then draw a straight line through the predictions at these two points. We compute the predicted values at the end and at 2020:

```
# Last knot and last point on period scale
( P.rf <- c( max(m1$Knots$Per), max(m1$Per[,1]) ) )
```

```
[1] 2000.667 2003.667
```

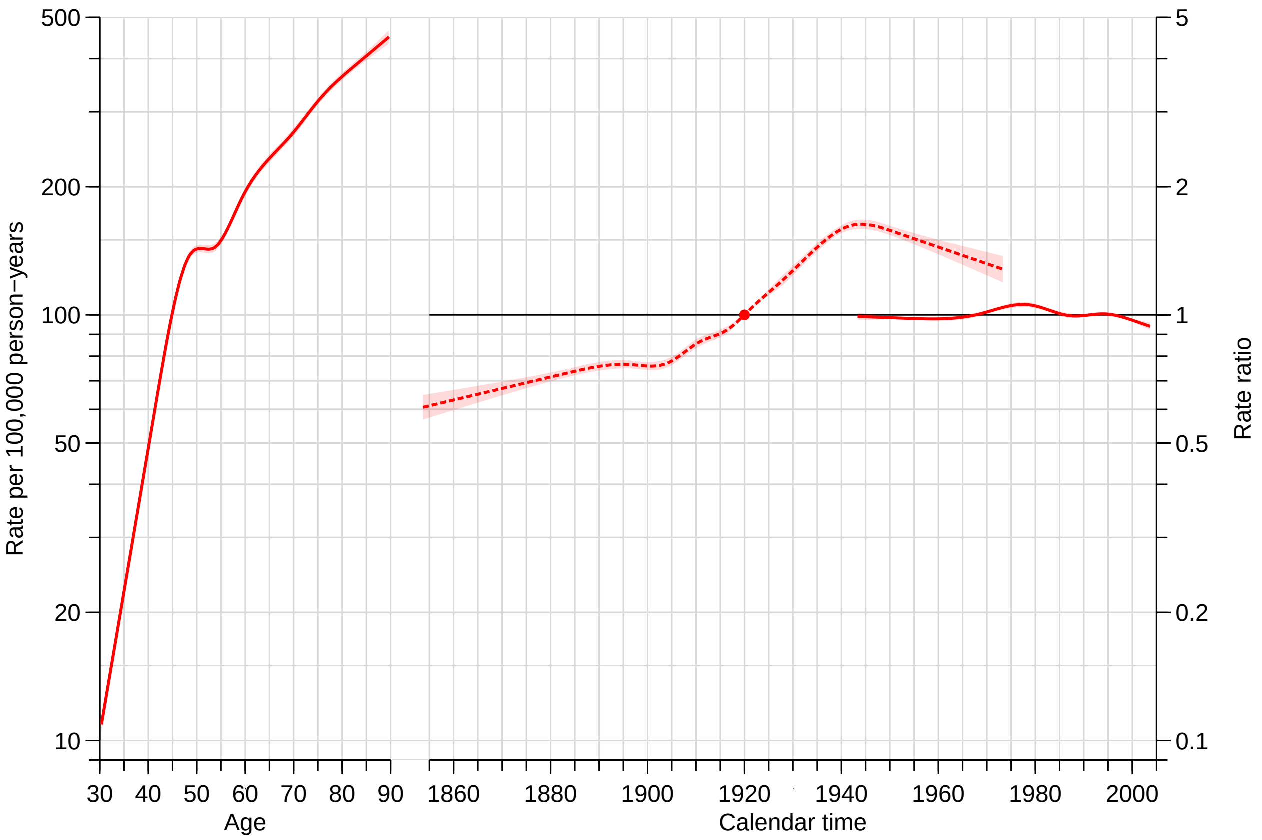

Figure 2.31: *Estimates of age- period- and cohort effects plotted after fine tuning the display using* `apc.frame`. The broken line is the cohort effect. ../graph/brcapr-apcfit-2

```
 # Last point plus one 20 years later
 ( P.pt <- P.rf[2] + 0:1*20 )

[1] 2003.667 2023.667

 # Linear interpolation of log-rates at the two reference points
 ( Pp <- approx( m1$Per[,1], log(m1$Per[,2]), P.rf )$y )

[1] -0.03478862 -0.06184521

 # Liner extrapoltion through these two points to the future points
 ( P.eff <- Pp[2] + (Pp[2]-Pp[1])/diff(P.rf)*(P.pt-P.rf[2]) )

[1] -0.06184521 -0.24222248
```

The same thing done on the cohort scale:

```
 ( C.rf <- c( max( m1$Knots$Coh ), max( m1$Coh[,1] ) ) )

[1] 1950.667 1973.333

 ( C.pt <- C.rf[2] + 0:1*20 )

[1] 1973.333 1993.333

 ( Cp <- approx( m1$Coh[,1], log(m1$Coh[,2]), C.rf )$y )
```

```
[1] 0.4510767 0.2468646
```

```
( C.eff <- Cp[2] + (Cp[2]-Cp[1])/diff(C.rf)*(C.pt-C.rf[2]) )
```

```
[1] 0.2468646 0.0666774
```

Finally, these are added to the plot of the effects, after we have re-drawn the frame with a calendar-time axis extending to 2020 (remember that the `P.eff` and the `C.eff` are log-RRs, and hence we need to take the exp before plotting):

```
par( las=1, mar=c(3,4,1,4), mgp=c(3,1,0)/1.5 )
fp <-  apc.frame( a.lab = seq(30,90,10),
                 cp.lab = seq(1860,2020,20),
                  r.lab = c(c(1,2,5)*10,c(1,2,5)*100),
#                 rr.lab = r.lab / rr.ref,
                 rr.ref = 100,
                  a.tic = seq(30,90,5),
                 cp.tic = seq(1855,2025,5),
                  r.tic = c(9,1:9*10,1:5*100),
#                 rr.tic = r.tic / rr.ref,
                tic.fac = 1.3,
                  a.txt = "Age",
                 cp.txt = "Calendar time",
                  r.txt = "Rate per 100,000 person-years",
                 rr.txt = "Rate ratio",
                    gap = 8,
               col.grid = gray(0.85),
                  sides = c(1,2,4) )
   matshade(m1$Age[,1], m1$Age[,-1], col="red", lwd = 2 )
pc.matshade(m1$Per[,1], m1$Per[,-1], col="red", lwd = 2 )
pc.matshade(m1$Coh[,1], m1$Coh[,-1], col="red", lwd = 2, lty = "21")
pc.points( 1920, 1, pch=16, col="red" )
lines( P.pt-fp[1], exp(P.eff)*fp[2], col=gray(0.0), lty="11", lwd=2 )
lines( C.pt-fp[1], exp(C.eff)*fp[2], col=gray(0.0), lty="11", lwd=2 )
```

6. The fitted model gives an age-effect, a period effect and a cohort effect; the `apc` object contains representations of these three effects as matrices with the age-values and the estimated effects (with c.i.s) at these values and similarly for the period and cohort effects.

   Since the model fitted is using natural splines with linear effects for the part beyond the last knot, we will automatically get a prediction based on a linear extension of these if we just use the `ci.pred` on the model.

   However, the fitted model object is based on the design matrices derived from the parametrization, so it does not lend itself easily to predictions. Hence we fit the model with an arbitrary parametrization using the knots used.

```
M1 <- glm(D ~ Ns(  A, knots = m1$Knots$Age) +
              Ns(P  , knots = m1$Knots$Per) +
              Ns(P-A, knots = m1$Knots$Coh)[,-1],
          offset = log(Y),
          family = poisson,
            data = subset(breast, A > 30))
```

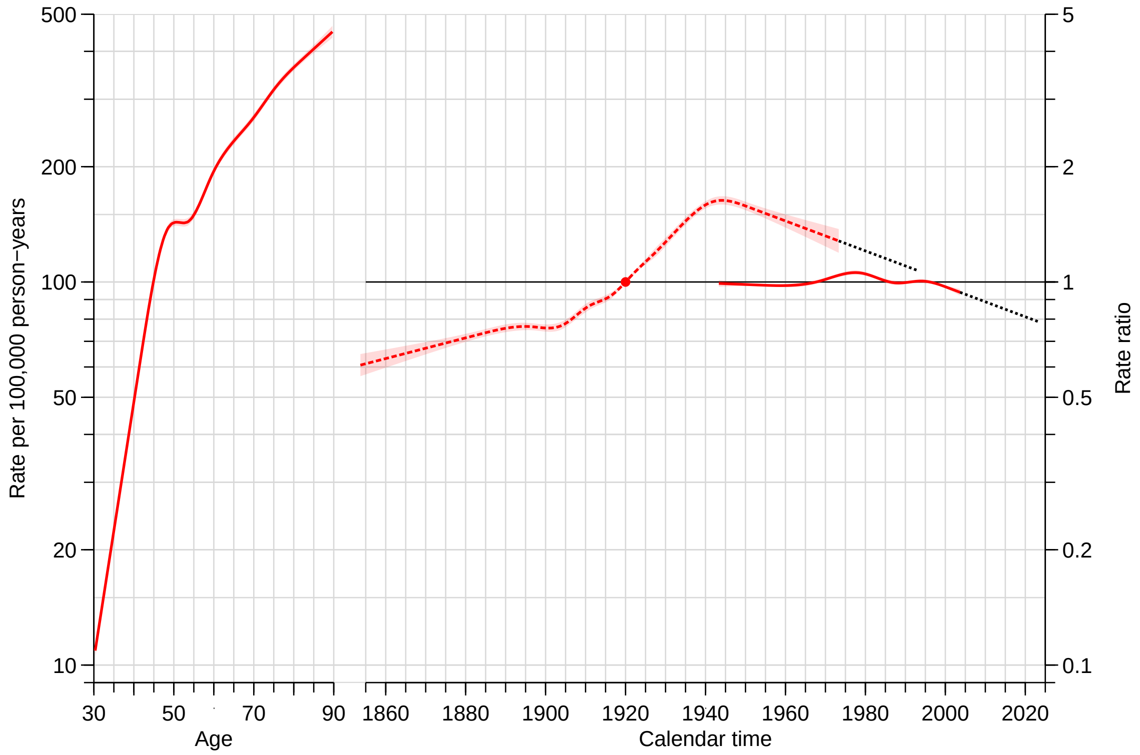

Figure 2.32: *Estimates of age- period- and cohort effects with the linear extension of the period and cohort effects used for prediction of future rates.* ../graph/brcapr-apcfit-3

Note that we have omitted the first column of the cohort term in order to get a model matrix of full rank. Formally there is no need for this, but we will be spared warnings from R that prediction from rank-deficient models may be misleading.

We can check that we actually *did* fit the same model as `apc.fit`:

```
c(M1$deviance, m1$Model$deviance)
```

```
[1] 9208.167 9208.167
```

```
summary(predict(M1) - predict(m1$Model))
```

```
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-1.599e-14  0.000e+00  2.665e-15  2.664e-15  5.329e-15  1.599e-14
```

Digression:
Note that we fitted the model `M1` using the `poisson` family in order to be able to compare the fitted values — the function `apc.fit` internally uses the `poisson` family.

For models fitted with the familiy `poisson`, `predict` will produce (log) fitted `number` of events for the covariates in the original dataset, *including* `Y`, the observed person-years. Logical, as the `log(Y)` enters as a covariate in the model, albeit with a fixed regression coefficient.

For models fitted with `poisreg`, `predict` will produce (log) fitted `rates` for the covariates in the original dataset, *excluding* `Y`, on the scale corresponding to the units in

which the person-time was supplied in the response. Logical, as the `Y` is part of the response, and not of the predictor.
End of digression.

So if we want to predict age-specific rates in 2020–30 and in the 1960–70 cohorts respectively we just set up prediction data frames and use them with the `ci.pred` function. This is where the convenience of the natural splines come in:

```
a.pt <- seq(30, 90, 1/10)
Pfr <- rbind(data.frame(A = a.pt, P = 2020, Y = 1000), NA,
             data.frame(A = a.pt, P = 2030, Y = 1000) )
Cfr <- rbind(data.frame(A = a.pt, P = a.pt + 1960, Y = 1000), NA,
             data.frame(A = a.pt, P = a.pt + 1970, Y = 1000) )
prP <- ci.pred(M1, Pfr)
prC <- ci.pred(M1, Cfr)
```

These predicted rates are easily plotted together:

```
(ct <- c(0, which(is.na(prP[,1])), nrow(prP) + 1))

     602
  0  602 1204

for( i in 1:2 )
   {
wh <- (ct[i]+1):(ct[i+1]-1)
matshade(Pfr$A[wh], cbind(prP, prC)[wh,], plot = (i == 1),
          log = "y", las = 1, xlim = c(27, 90), xlab = "",
          ylab = "Predicted breast cancer incidence per 1000 PY",
          type = "l", lwd = 2, lty = 1, col = c("red", "forestgreen"))
   }
text(rep(29.5,2), prP[c(1,603),1], paste(c(2020,2030)),
      col = "red", adj = 1, cex = 0.8 )
text(rep(29.5,2), prC[c(1,603),1], paste(c(1960,1970)),
      col = "forestgreen", adj = 1, cex = 0.8 )
mtext(side = 1, at = 50, "Age at 2020/2030",    col = "red", line = 2)
mtext(side = 1, at = 70, "Current age", col = "forestgreen", line = 2)
```

7. In order to explore the robustness of the prediction machinery we fit a model where we omitted the last knot of the period effect and subsequently the the last knot of the cohort effect too. First we would like to see the parameters in the same plot as before, so we use `apc.fit` to derive the parametrization:

```
mp <- apc.fit(subset(breast, A > 30),
                npar = list(A = m1$Knots$Age,
                             P = m1$Knots$Per[-length(m1$Knots$Per)],
                             C = m1$Knots$Coh),
                ref.c = 1920, scale = 10^5)

[1] "ML of APC-model Poisson with log(Y) offset : ( ACP ):\n"
              Model      AIC Mod. df. Mod. dev. Test df.  Test dev.      Pr(>Chi)
1               Age 48805.35     7312 16427.665       NA         NA            NA
2         Age-drift 42743.95     7311 10364.264        1 6063.40113  0.000000e+00
3        Age-Cohort 41693.09     7303  9297.401        8 1066.86310 5.484678e-225
4 Age-Period-Cohort 41624.19     7300  9222.505        3   74.89559  3.814905e-16
```

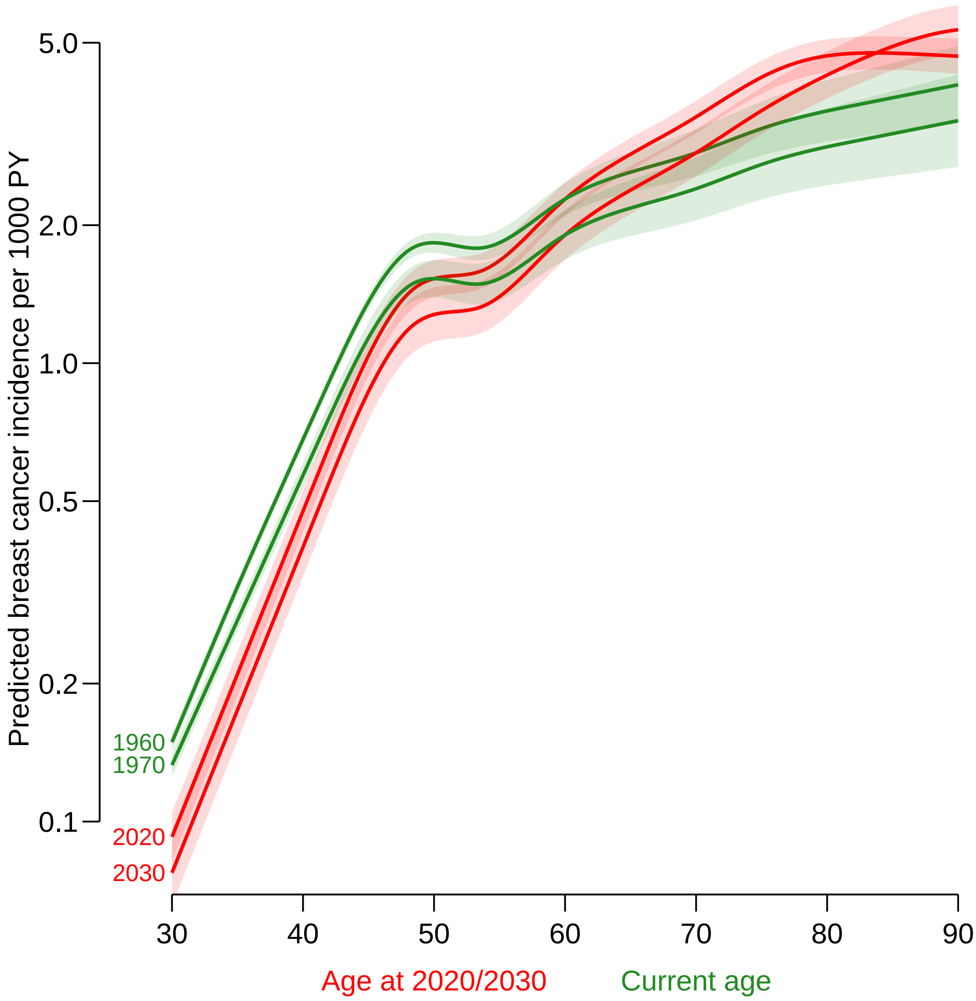

Figure 2.33: *Predicted age-specific breast cancer incidence rates for the dates (1. January) 2020 and 2030 (red), and for the birth cohorts (1. January) 1960 and 1970 (green). Note that for the predictions for the dates 2020 and 2030, age refers to the age at these dates; whereas for the predictions for the birth cohorts 1960 and 1970, age refers to current age.*../graph/brcapr-pred1

```
5        Age-Period 42678.27     7308 10292.581        8 1070.07600 1.110170e-225
6         Age-drift 42743.95     7311 10364.264        3   71.68269  1.861592e-15
  Test dev/df     H0
1          NA
2 6063.40113 zero drift
3  133.35789 Coh eff|dr.
4   24.96520 Per eff|Coh
5  133.75950 Coh eff|Per
6   23.89423 Per eff|dr.
```

```
mpc <- apc.fit(subset(breast, A>30),
               npar = list(A = m1$Knots$Age,
                           P = m1$Knots$Per[-length(m1$Knots$Per)],
                           C = m1$Knots$Coh[-length(m1$Knots$Coh)]),
               ref.c = 1920, scale = 10^5)
```

```
[1] "ML of APC-model Poisson with log(Y) offset : ( ACP ):\n"
              Model      AIC Mod. df. Mod. dev. Test df. Test dev.       Pr(>Chi)
```

```
1               Age 48805.35     7312 16427.665       NA          NA            NA
2         Age-drift 42743.95     7311 10364.264        1 6063.40113  0.000000e+00
3        Age-Cohort 41745.14     7304  9351.454        7 1012.81001 2.055485e-214
4 Age-Period-Cohort 41675.03     7301  9275.346        3   76.10812  2.096935e-16
5        Age-Period 42678.27     7308 10292.581        7 1017.23544 2.273366e-215
6         Age-drift 42743.95     7311 10364.264        3   71.68269  1.861592e-15
  Test dev/df    H0
1          NA
2  6063.40113 zero drift
3   144.68714 Coh eff|dr.
4    25.36937 Per eff|Coh
5   145.31935 Coh eff|Per
6    23.89423 Per eff|dr.
```

We then plot the estimates from these models together with the estimates from the first one — recall that the two latter models have one, resp. two parameters less that the first one we fitted.

```
par( las=1, mar=c(3,4,1,4), mgp=c(3,1,0)/1.5 )
fp <-  apc.frame( a.lab = seq(30,90,10),
                 cp.lab = seq(1860,2020,20),
                  r.lab = c(c(1,2,5)*10,c(1,2,5)*100),
#                 rr.lab = r.lab / rr.ref,
                 rr.ref = 100,
                  a.tic = seq(30,90,5),
                 cp.tic = seq(1855,2025,5),
                  r.tic = c(9,1:9*10,1:5*100),
#                 rr.tic = r.tic / rr.ref,
                tic.fac = 1.3,
                  a.txt = "Age",
                 cp.txt = "Calendar time",
                  r.txt = "Rate per 100,000 person-years",
                 rr.txt = "Rate ratio",
                    gap = 8,
               col.grid = gray(0.85),
                  sides = c(1,2,4) )
lines( m1 , frame.par=fp, ci=T, col="black", lwd=c(3,1,1), knots=TRUE )
lines( mp , frame.par=fp, ci=T, col="red"      , lty=1, lwd=c(3,1,1) )
lines( mpc, frame.par=fp, ci=T, col="limegreen", lty="21", lwd=c(3,1,1) )
```

We see that the difference in the parameter components between the three models is minimal, but this does not necessarily not necessarily the predictions; so in line with the previous set-up, we compute the slope of the period and cohort effects from the two models and compare them with the previous one:

```
pr.slopes <- matrix( NA, 3, 3 )
rownames( pr.slopes ) <- c("Org","-lastP","-lastPC")
colnames( pr.slopes ) <- c("P-slope","C-slope","P-C-slope")
pr.slopes["Org","P-slope"] <- diff(Pp)/diff(P.rf)
pr.slopes["Org","C-slope"] <- diff(Cp)/diff(C.rf)
```

Here are then the calculations from the models where the last knots have been removed for the period, respectively both period and cohort effects:

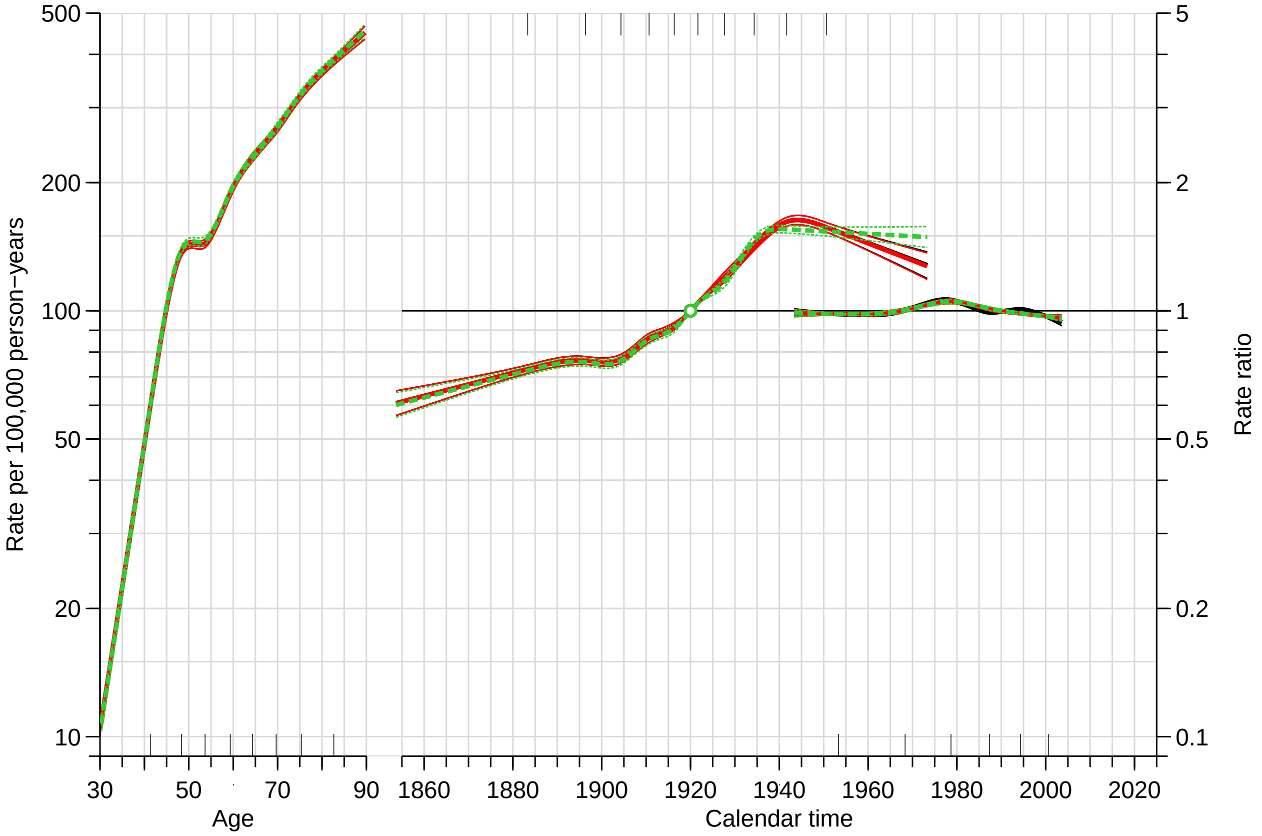

Figure 2.34: *Estimated APC-effects from the three different models. The dotted lines are the models where successively the last period (in red) and cohort (in green) knot were removed.* `../graph/brcapr-apcfit-4`

```
 ( P.rf <- c( max( mp$Knots$Per ), max( mp$Per[,1] ) ) )

[1] 1994.333 2003.667

 P.pt <- P.rf[2] + 0:20
 Pp <- approx( mp$Per[,1], log(mp$Per[,2]), P.rf )$y
 P.eff <- Pp[2] + (Pp[2]-Pp[1])/diff(P.rf)*(P.pt-P.rf[2])
 ( C.rf <- c( max( mp$Knots$Coh ), max( mp$Coh[,1] ) ) )

[1] 1950.667 1973.333

 C.pt <- C.rf[2] + 0:20
 Cp <- approx( mp$Coh[,1], log(mp$Coh[,2]), C.rf )$y
 C.eff <- Cp[2] + (Cp[2]-Cp[1])/diff(C.rf)*(C.pt-C.rf[2])
 pr.slopes["-lastP","P-slope"] <- diff(Pp)/diff(P.rf)
 pr.slopes["-lastP","C-slope"] <- diff(Cp)/diff(C.rf)
 ( P.rf <- c( max( mpc$Knots$Per ), max( mpc$Per[,1] ) ) )

[1] 1994.333 2003.667

 P.pt <- P.rf[2] + 0:20
 Pp <- approx( mpc$Per[,1], log(mpc$Per[,2]), P.rf )$y
 P.eff <- Pp[2] + (Pp[2]-Pp[1])/diff(P.rf)*(P.pt-P.rf[2])
 ( C.rf <- c( max( mpc$Knots$Coh ), max( mpc$Coh[,1] ) ) )

[1] 1941.667 1973.333
```

```
 C.pt <- C.rf[2] + 0:20
 Cp <- approx( mpc$Coh[,1], log(mpc$Coh[,2]), C.rf )$y
 C.eff <- Cp[2] + (Cp[2]-Cp[1])/diff(C.rf)*(C.pt-C.rf[2])
 pr.slopes["-lastPC","P-slope"] <- diff(Pp)/diff(P.rf)
 pr.slopes["-lastPC","C-slope"] <- diff(Cp)/diff(C.rf)
 pr.slopes[,3] <- pr.slopes[,1] + pr.slopes[,2]
 round( pr.slopes, 4 )

        P-slope C-slope P-C-slope
Org     -0.0090 -0.0090   -0.0180
-lastP  -0.0025 -0.0093   -0.0117
-lastPC -0.0029 -0.0013   -0.0043

 round( 100*(exp(pr.slopes)-1), 4 )

        P-slope C-slope P-C-slope
Org     -0.8978 -0.8969   -1.7867
-lastP  -0.2485 -0.9218   -1.1680
-lastPC -0.2945 -0.1322   -0.4262
```

We see that overall period/cohort drift that will be used in the predictions will be annual decreases of 2.2% and 1.1% depending on the models chosen.

8. In order to make the predictions based on the models we fit them in the guise of classical `glm` models (again leaving out a non-identifiable column of the predictor to avoid warnings when predicting):

```
 Mp <- glm( D ~ Ns(   A, knots=mp$Knots$Age ) +
                Ns( P  , knots=mp$Knots$Per ) +
                Ns( P-A, knots=mp$Knots$Coh )[,-1],
                family = poisson,
                offset = log(Y),
                  data =  subset( breast, A>30 ) )
 Mpc <- glm( D ~ Ns(   A, knots=mpc$Knots$Age ) +
                 Ns( P  , knots=mpc$Knots$Per ) +
                 Ns( P-A, knots=mpc$Knots$Coh )[,-1],
                 family = poisson,
                 offset = log(Y),
                   data =  subset( breast, A>30 ) )
```

With these models fitted we can compute the predictions and compare with those based on the first fitted model (which does not have any sacred status relative to the others). We already devised the prediction frames so it's quite simple:

```
 prPp <- ci.pred( Mp, Pfr )
 prCp <- ci.pred( Mp, Cfr )
 prPpc <- ci.pred( Mpc, Pfr )
 prCpc <- ci.pred( Mpc, Cfr )
```

But due to the excess number of curves we plot the different period and cohort predictions separately (and without c.i.s):

```
par( mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
matplot( Pfr$A, cbind( prP[,1], prPp[,1], prPpc[,1] ),
         log="y", las=1, xlim=c(25,90), xlab="Age", ylim=c(0.1,6),
         ylab="Predicted breast cancer incidence per 100,000 PY",
         type="l", lwd=3, lty=1, col=c("gray","limegreen","red") )
matplot( Pfr$A, cbind( prC[,1], prCp[,1], prCpc[,1] ),
         log="y", las=1, xlim=c(25,90), xlab="Age", ylim=c(0.1,6),
         ylab="Predicted breast cancer incidence per 100,000 PY",
         type="l", lwd=3, lty=1, col=c("gray","limegreen","red") )
```

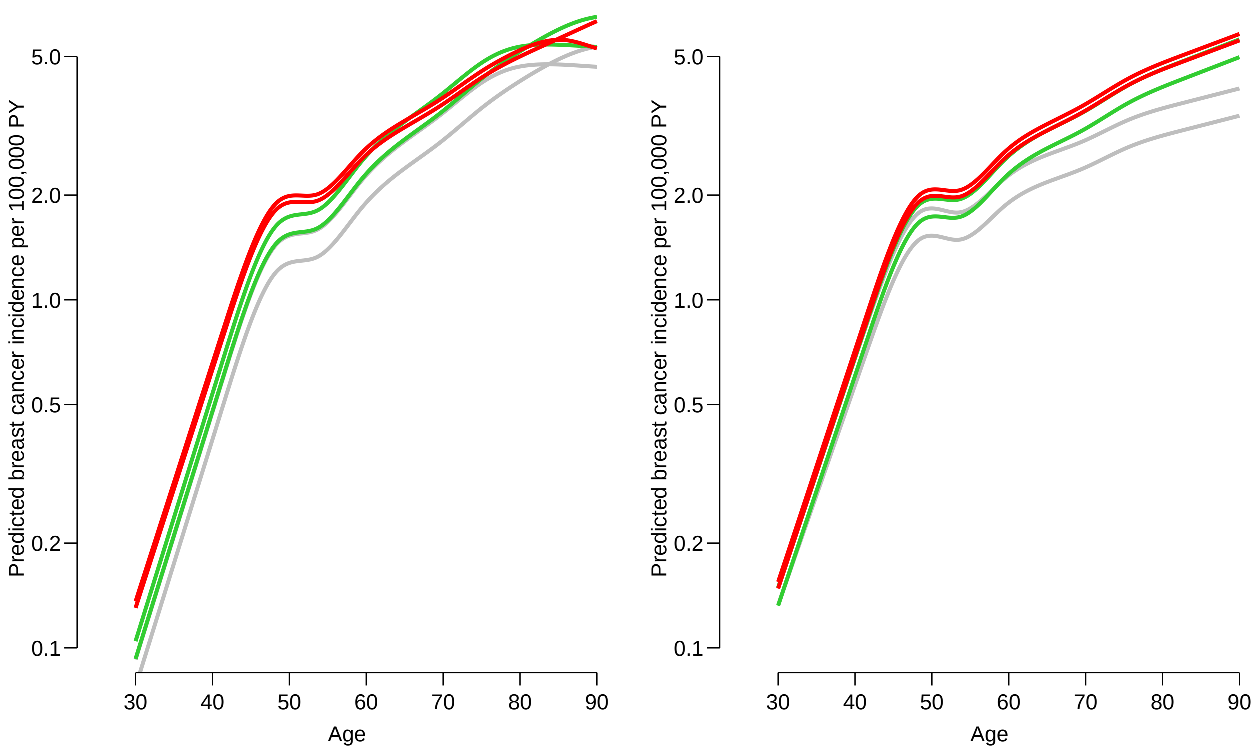

Figure 2.35: *Prediction of cross-sectional rates in 2020, 2025 and 2030 (top down in left panel) and cohorts 1960, 1965 and 1970 (top down in right panel) with the standard knots (gray), and (green) last period knot omitted resp. (red) both last period and cohort knot omitted.* ../graph/brcapr-predx

From figure 2.35 it is seen what could be expected from the parameter estimates, namely that the predictions from the later models are higher because the overall *de*crease in rates is deemed smaller by the later models. Thus again a confirmation that prediction of future rates is a risky business.

Chapter 3

Basic concepts of rates and survival

The following is a summary of relations between various quantities used in analysis of follow-up studies. They are ubiquitous in the analysis and reporting of results. Hence it is important to be familiar with all of them and the relation between them.

3.1 Probability

Survival function:

$$\begin{aligned} S(t) &= \mathrm{P}\{\text{survival at least till } t\} \\ &= \mathrm{P}\{T > t\} = 1 - \mathrm{P}\{T \leq t\} = 1 - F(t) \end{aligned}$$

where T is the variable "time of death"

Conditional survival function:

$$\begin{aligned} S(t|t_{\text{entry}}) &= \mathrm{P}\{\text{survival at least till } t | \text{ alive at } t_{\text{entry}}\} \\ &= S(t)/S(t_{\text{entry}}) \end{aligned}$$

Cumulative distribution function of death times (cumulative risk):

$$\begin{aligned} F(t) &= \mathrm{P}\{\text{death before } t\} \\ &= \mathrm{P}\{T \leq t\} = 1 - S(t) \end{aligned}$$

Density function of death times:

$$f(t) = \lim_{h \to 0} \mathrm{P}\{\text{death in } (t, t+h)\}/h = \lim_{h \to 0} \frac{F(t+h) - F(t)}{h} = F'(t)$$

Intensity:

$$\begin{aligned} \lambda(t) &= \lim_{h \to 0} \mathrm{P}\{\text{event in } (t, t+h] \mid \text{alive at } t\}/h \\ &= \lim_{h \to 0} \frac{F(t+h) - F(t)}{S(t)h} = \frac{f(t)}{S(t)} \\ &= \lim_{h \to 0} -\frac{S(t+h) - S(t)}{S(t)h} = -\frac{\mathrm{d} \log S(t)}{\mathrm{d}t} \end{aligned}$$

The intensity is also known as the hazard function, hazard rate, mortality/morbidity rate or simply “rate”.

Note that f and λ are *scaled* quantities, they have dimension time^{-1}.

Relationships between terms:

$$\begin{aligned} -\frac{\mathrm{d}\log S(t)}{\mathrm{d}t} &= \lambda(t) \\ &\Updownarrow \\ S(t) &= \exp\left(-\int_0^t \lambda(u)\,\mathrm{d}u\right) = \exp\bigl(-\Lambda(t)\bigr) \end{aligned}$$

The quantity $\Lambda(t) = \int_0^t \lambda(s)\,\mathrm{d}s$ is called the *integrated intensity* or the **cumulative rate**. It is *not* an intensity (rate), it is dimensionless, despite its name.

$$\lambda(t) = -\frac{\mathrm{d}\log(S(t))}{\mathrm{d}t} = -\frac{S'(t)}{S(t)} = \frac{F'(t)}{1-F(t)} = \frac{f(t)}{S(t)}$$

The cumulative *risk* of an event (to time t) is:

$$F(t) = \mathrm{P}\{\text{Event before time } t\} = \int_0^t \lambda(u)S(u)\,\mathrm{d}u = 1 - S(t) = 1 - \mathrm{e}^{-\Lambda(t)}$$

For small $|x|$ (< 0.05), we have that $1 - \mathrm{e}^{-x} \approx x$, so for small values of the integrated intensity:

$$\text{Cumulative risk to time } t \approx \Lambda(t) = \text{Cumulative rate}$$

3.2 Statistics

Likelihood contribution from follow up of one person:

The likelihood from a number of small pieces of follow-up from one individual is a product of conditional probabilities:

$$\begin{aligned} \mathrm{P}\{\text{event at } t_4|\text{entry at } t_0\} &= \mathrm{P}\{\text{survive } (t_0,t_1)|\text{ alive at } t_0\} \times \\ & \quad \mathrm{P}\{\text{survive } (t_1,t_2)|\text{ alive at } t_1\} \times \\ & \quad \mathrm{P}\{\text{survive } (t_2,t_3)|\text{ alive at } t_2\} \times \\ & \quad \mathrm{P}\{\text{event at } t_4|\text{ alive at } t_3\} \end{aligned}$$

Each term in this expression corresponds to one *empirical rate*[1] $(d, y) = (\#\text{deaths}, \#\text{risk time})$, i.e. the data obtained from the follow-up of one person in the interval of length y. Each person can contribute many empirical rates, most with $d = 0$; d can only be 1 for the *last* empirical rate for a person.

Log-likelihood for one empirical rate (d, y):

$$\ell(\lambda) = \log\bigl(\mathrm{P}\{d \text{ events in } y \text{ follow-up time}\}\bigr) = d\log(\lambda) - \lambda y$$

This is under the assumption that the rate (λ) is constant over the interval that the empirical rate refers to.

[1] This is a concept coined by BxC, and so is not necessarily generally recognized.

Log-likelihood for several persons. Adding log-likelihoods from a group of persons (only contributions with identical rates) gives:

$$D\log(\lambda) - \lambda Y,$$

where Y is the total follow-up time ($Y = \sum_i y_i$), and D is the total number of failures ($D = \sum_i d_i$), where the sums are over individuals' contributions with the *same* rate, λ, for example from the same age-class fro all individuals.

Note: The Poisson log-likelihood for an observation D with mean λY is:

$$D\log(\lambda Y) - \lambda Y = D\log(\lambda) + D\log(Y) - \lambda Y$$

The term $D\log(Y)$ does not involve the parameter λ, so the likelihood for an observed rate (D, Y) can be maximized by pretending that the no. of cases D is Poisson with mean λY. But this does *not* imply that D follows a Poisson-distribution. It is entirely a likelihood based computational convenience. Anything that is not likelihood based is not justified.

A linear model for the log-rate, $\log(\lambda) = X\beta$ implies that

$$\lambda Y = \exp\bigl(\log(\lambda) + \log(Y)\bigr) = \exp\bigl(X\beta + \log(Y)\bigr)$$

Therefore, in order to get a linear model for $\log(\lambda)$ we must require that $\log(Y)$ appear as a variable in the model for $D \ \sim (\lambda Y)$ with the regression coefficient fixed to 1, a so-called *offset*-term in the linear predictor.

3.3 Competing risks

Competing risks: If there are more than one, say 3, causes of death, occurring with (cause-specific) rates λ_1, λ_2, λ_3, that is:

$$\lambda_c(a) = \lim_{h\to 0} \mathrm{P}\{\text{death from cause } c \text{ in } (a, a+h] \mid \text{alive at } a\}/h, \quad c = 1, 2, 3$$

The survival function is then:

$$S(a) = \exp\left(-\int_0^a \lambda_1(u) + \lambda_2(u) + \lambda_3(u)\,\mathrm{d}u\right)$$

because you have to escape all 3 causes of death. The probability of dying from cause 1 before age a (the cause-specific cumulative risk) is:

$$F_1(a) = \mathrm{P}\{\text{dead from cause 1 at } a\} = \int_0^a \lambda_1(u)S(u)\,\mathrm{d}u \neq 1 - \exp\left(-\int_0^a \lambda_1(u)\,\mathrm{d}u\right)$$

The term $\exp(-\int_0^a \lambda_1(u)\,\mathrm{d}u)$ is sometimes referred to as the "cause-specific survival", but it does not have any probabilistic interpretation in the real world. It is the survival under the assumption that only cause 1 existed and that the mortality rate from this cause was the same as when the other causes were present too.

Together with the survival function, the cause-specific cumulative risks represent a classification of the population at any time in those alive and those dead from causes 1, 2 and 3 respectively:

$$1 = S(a) + \int_0^a \lambda_1(u)S(u)\,\mathrm{d}u + \int_0^a \lambda_2(u)S(u)\,\mathrm{d}u + \int_0^a \lambda_3(u)S(u)\,\mathrm{d}u, \quad \forall a$$

Subdistribution hazard Fine and Gray defined models for the so-called subdistribution hazard, $\tilde{\lambda}_i(a)$. Recall the relationship between between the hazard (λ) and the cumulative risk (F):

$$\lambda(a) = -\frac{\mathrm{d}\log\big(S(a)\big)}{\mathrm{d}a} = -\frac{\mathrm{d}\log\big(1-F(a)\big)}{\mathrm{d}a}$$

When more competing causes of death are present the Fine and Gray idea is to use this transformation to the cause-specific cumulative risk for cause 1, say:

$$\tilde{\lambda}_1(a) = -\frac{\mathrm{d}\log\big(1-F_1(a)\big)}{\mathrm{d}a}$$

Here, $\tilde{\lambda}_1$ is called the subdistribution hazard; as a function of $F_1(a)$ it depends on the survival function S, which depends on *all* the cause-specific hazards:

$$F_1(a) = \mathrm{P}\{\text{dead from cause 1 at } a\} = \int_0^a \lambda_1(u)S(u)\,\mathrm{d}u$$

The subdistribution hazard is merely a transformation of the cause-specific cumulative risk. Namely the same transformation which in the single-cause case transforms the cumulative risk to the hazard. It is a mathematical construct that is not interpretable as a hazard despite its name.

3.4 Demography

Expected residual lifetime: The expected lifetime (at birth) is simply the variable age (a) integrated with respect to the distribution of age at death:

$$\mathrm{EL} = \int_0^\infty af(a)\,\mathrm{d}a$$

where f is the density of the distribution of lifetime (age at death).

The relation between the density f and the survival function S is $f(a) = -S'(a)$, so integration by parts gives:

$$\mathrm{EL} = \int_0^\infty a\big(-S'(a)\big)\,\mathrm{d}a = -\Big[aS(a)\Big]_0^\infty + \int_0^\infty S(a)\,\mathrm{d}a$$

The first of the resulting terms is 0 because $S(a)$ is 0 at the upper limit and a by definition is 0 at the lower limit.

Hence the expected lifetime can be computed as the integral of the survival function.

The expected *residual* lifetime at age a is calculated as the integral of the *conditional* survival function for a person aged a:

$$\mathrm{EL}(a) = \int_a^\infty S(u)/S(a)\,\mathrm{d}u$$

Lifetime lost due to a disease is the difference between the expected residual lifetime for a diseased person and a non-diseased (well) person at the same age. So all that is needed is a(n estimate of the) survival function in each of the two groups.

$$\mathrm{LL}(a) = \int_a^\infty S_{\text{Well}}(u)/S_{\text{Well}}(a) - S_{\text{Diseased}}(u)/S_{\text{Diseased}}(a)\,\mathrm{d}u$$

Note that the definition of the survival function for a non-diseased person requires a decision as to whether one will consider non-diseased persons immune to the disease in question or not. That is whether we will include the possibility of a well person getting ill and subsequently die. This does not show up in the formulae, but is a decision required in order to devise an estimate of S_{Well}.

Lifetime lost by cause of death is using the fact that the difference between the survival probabilities is the same as the difference between the death probabilities. If several causes of death (3, say) are considered then:

$$\begin{aligned} S(a) = 1 &- \mathrm{P}\{\text{dead from cause 1 at } a\} \\ &- \mathrm{P}\{\text{dead from cause 2 at } a\} \\ &- \mathrm{P}\{\text{dead from cause 3 at } a\} \end{aligned}$$

and hence:

$$\begin{aligned} S_{\text{Well}}(a) - S_{\text{Diseased}}(a) = &\ \mathrm{P}\{\text{dead from cause 1 at } a|\text{Diseased}\} \\ &+ \mathrm{P}\{\text{dead from cause 2 at } a|\text{Diseased}\} \\ &+ \mathrm{P}\{\text{dead from cause 3 at } a|\text{Diseased}\} \\ &- \mathrm{P}\{\text{dead from cause 1 at } a|\text{Well}\} \\ &- \mathrm{P}\{\text{dead from cause 2 at } a|\text{Well}\} \\ &- \mathrm{P}\{\text{dead from cause 3 at } a|\text{Well}\} \end{aligned}$$

So we can conveniently define the lifetime lost due to cause 2, say, by:

$$\begin{aligned} \mathrm{LL}_2(a) = \int_a^\infty &\mathrm{P}\{\text{dead from cause 2 at } u|\text{Diseased \& alive at } a\} \\ &- \mathrm{P}\{\text{dead from cause 2 at } u|\text{Well \& alive at } a\}\,\mathrm{d}u \end{aligned}$$

These quantities have the property that their sum is the total years of life lost due to the disease:

$$\mathrm{LL}(a) = \mathrm{LL}_1(a) + \mathrm{LL}_2(a) + \mathrm{LL}_3(a)$$

The terms in the integral are computed as (see the section on competing risks):

$$\begin{aligned} \mathrm{P}\{\text{dead from cause 2 at } x|\text{Diseased \& alive at } a\} &= \int_a^x \lambda_{2,\text{Dis}}(u) S_{\text{Dis}}(u)/S_{\text{Dis}}(a)\,\mathrm{d}u \\ \mathrm{P}\{\text{dead from cause 2 at } x|\text{Well \& alive at } a\} &= \int_a^x \lambda_{2,\text{Well}}(u) S_{\text{Well}}(u)/S_{\text{Well}}(a)\,\mathrm{d}u \end{aligned}$$