Components of diabetes prevalence in Denmark

SDC
June 2012
Version 1

Compiled Wednesday 6th June, 2012, 01:29
from: `C:/Bendix/Steno/KBJo/prev/DMprev.tex`

Bendix Carstensen
Steno Diabetes Center, Gentofte, Denmark
& Department of Biostatistics, University of Copenhagen
bxc@steno.dk
http://BendixCarstensen.com

Contents

Chapter 1

Introduction

1.1 Problem

This note addresses the problem of partitioning the changes in prevalence of diabetes in the Danish population over the last 15 years to:

1. changes in mortality rates among diabetes patients
2. changes in incidence rates of diabetes in the population

This measure will be sex- and age-specific, and hence independent of the demographic changes in the population.

1.2 Formalization

First we formalize the problem concptually, then statistical, and finally outline the practical implementation based on analysis of rates.

1.2.1 Conceptual

The observed changes in prevalence of DM are a consequence of the changes in mortality and DM-incidence rates in the population and of the changes in the mortality rates in the DM population.

Of these the changes in population mortality presumably have the smaller role, but there is a connection, because they determine the available number of persons susceptible to a DM diagnosis.

Thus the starting point will be the population prevalence of DM as of 1.1.1995. The (age-specific) prevalence at any future point of time is obtained by applying the mortality rates in the two sub-strata of the population (DM / non-DM) and the DM-incidence rates to the non-DM part of the population.

The exercise consists in working out what the prevalence of diabetes would have been if:

1. mortality rates and diabetes rates had remained stable
2. only mortality rates had remained stable, but incidence rates had developed as observed

The difference between observed prevalences and the predicted under scenario

1. is the combined effect of changes in the rates as seen since the starting point chosen.
2. is the effect of changing mortality rates alone, which is what is of interest.

For the sake of completeness we shall compute both.

Chapter 2

Analysis

2.1 Statistical framework

First we consider the setup as outlined in figure 2.1:

```
> library( Epi )
> tm <- matrix(NA,4,4)
> rownames(tm) <- colnames(tm) <- c("Well","DM","Dead (W)","Dead (DM)")
> tm[1,2] <- tm[1,3] <- tm[2,4] <- 1
> boxes.Lexis( tm, boxpos = list( x=c(20,20,80,80),
+                                 y=c(80,20,80,20) ),
+                  wmult=1.2, hmult=3,
+                  txt.arr = c( expression(lambda),
+                               expression(mu[W]),
+                               expression(mu[D][M]) ) )
```

The aim is to provide a precise formula for the age-specific prevalences at calendar time t, $p(a,t)$, given that we know the age-specific prevalence at some reference point t_0, $p(a,t_0)$ (in this case 1995, 2000, or 2005), and the transition rates $\lambda(a,p)$, $\mu_{\mathrm{W}}(a,p)$ and $\mu_{\mathrm{DM}}(a,p)$.

We can derive analytical expressions for this, but the easiest approach is to acquire parametric expressions for the transition rates and then update the age-specific prevalences by applying the transition probability matrix to a $A \times 2$ matrix of number of persons in each of the states Well and DM.

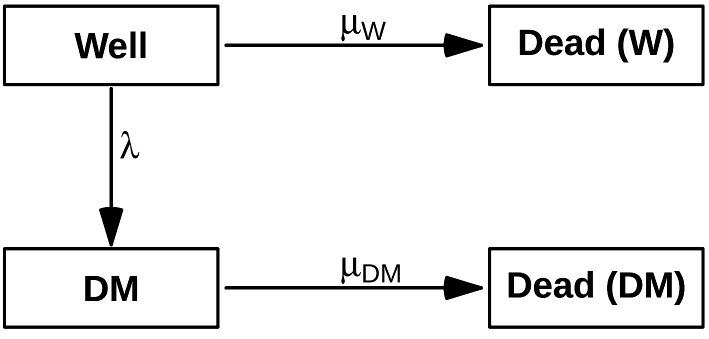

Figure 2.1: *The four states and transitions between them we consider*

For the given transition rates we can compute transition probabilities between states corresponding to a given (small) interval, δ, say, but first deriving the cumulative intensities for intervals of this length

$$\Lambda(a,p) = \lambda(a,p) \times \delta, \qquad M_{\mathrm{W}}(a,p) = \mu_{\mathrm{W}}(a,p) \times \delta, \qquad M_{\mathrm{DM}}(a,p) = \mu_{\mathrm{DM}}(a,p) \times \delta$$

and the the transition matrix $\mathbf{T}(a,p)$:

$$\mathbf{T}(a,p) = \begin{pmatrix} \mathrm{e}^{-\Lambda - M_{\mathrm{W}}} & \lambda \mathrm{e}^{-\Lambda - M_{\mathrm{W}}} \delta & \mu_{\mathrm{W}} \mathrm{e}^{-\Lambda - M_{\mathrm{W}}} \delta \\ 0 & \mathrm{e}^{-M_{\mathrm{DM}}} & \mu_{\mathrm{DM}} \mathrm{e}^{-\Lambda - M_{\mathrm{DM}}} \delta \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathrm{e}^{-\Lambda - M_{\mathrm{W}}} & \Lambda \mathrm{e}^{-\Lambda - M_{\mathrm{W}}} & M_{\mathrm{W}} \mathrm{e}^{-\Lambda - M_{\mathrm{W}}} \\ 0 & \mathrm{e}^{-M_{\mathrm{DM}}} & M_{\mathrm{DM}} \mathrm{e}^{-\Lambda - M_{\mathrm{DM}}} \\ 0 & 0 & 1 \end{pmatrix}$$

So we see that the rates only enter via the cumulative rates over the intervals, so this is what we eventually must compute from models. For simplicity we left out the (a,p) qualification of all the terms in the expressions.

Now if we have the *number* of persons in age-class a and period p in states (W,DM,Dead) in the vector $p(a,p)$ then:

$$p(a+1,p+1) = p(a,p)\mathbf{T}(a,p)$$

so updating the array of the number of persons in each state is merely a matter of matrix multiplication.

If we instead have the *fraction* of persons in states (W,DM) in the vector $q(a,p)$ (which is now just a 2-vector) then:

$$\tilde{p}(a+1,p+1) = q(a,p)\mathbf{T}(a,p)[\mathbf{1}:\mathbf{2},]$$

where we then will get the fraction of the persons in age a at time p who at time $p+1$ (and hence in age $a+1$) who are in states (W,DM,Dead). But since we are only interested in the progression of prevalences, then we instead use:

$$Q(a+1,p+1) = q(a,p)\mathbf{T}(a,p)[\mathbf{1}:\mathbf{2},\mathbf{1}:\mathbf{2}] q(a+1,p+1) = Q(a+1,p+1) / \sum_{\mathrm{W,DM}} Q(a+1,p+1)$$

so we update the prevalences at every step.

2.1.1 Births

Note however that for every step in the updating we will lose estimates in an age-class; in order for this to work we need to feed in the number of births in each age-group with some assumption about the distribution between DM/non-DM; which we will assume is 1/0, that is no new-born diabetics enter.

Chapter 3

Practical implementation in R

3.1 Prerequisites

We are going to make models for the rates in small intervals of age and calendar time, so we start by specifying the interval length, and the the points at which we want to predict the transition rates:

```
> int <- 0.1
> a.pt <- seq(   0, 100-int,int) + int/2
> p.pt <- seq(1995,2010-int,int) + int/2
```

We shall model all the rates by age-period-cohort models separately for men and women. For sensitivity we will also model the rates only by age-period models.

We will use natural splines to model the effects of age, period and cohort, and for all analyses we will use the same `number` of knots for these three effects, but of course place them differently based on the location of information, *i.e.* the events:

```
> nk.a <- 10
> nk.p <- 5
> nk.c <- 8
```

For the practical location of the spline knots we also define a small function which from the number of knots derives reasonable quantiles:

```
> qn <- function( nk, bd=2 ) seq( from =   1/(bd*nk),
+                                   to = 1-1/(bd*nk),
+                               length = nk )
> qn( 10, 2 )

 [1] 0.05 0.15 0.25 0.35 0.45 0.55 0.65 0.75 0.85 0.95

> qn( 10, 5 )

 [1] 0.0200000 0.1266667 0.2333333 0.3400000 0.4466667 0.5533333 0.6600000
 [8] 0.7666667 0.8733333 0.9800000
```

So this function generates `nk`, equidistant points in the interval (0,1) where the outer points are $1/(2 \times \texttt{nk})$ from the end. The second parameter is for modifying the multiplier 2.

Finally, to simplify the calling of the natural spline function we import a wrapper for the `ns` from the `splines` package:

```
> library( splines )
> source( "C:/stat/r/bxc/library.sources/useful/r/Ns.r" )
> Ns

function (x, df = NULL, knots = NULL, intercept = FALSE, Boundary.knots = NULL)
{
    if (is.null(Boundary.knots)) {
        if (!is.null(knots)) {
            knots <- sort(unique(knots))
            ok <- c(1, length(knots))
            Boundary.knots <- knots[ok]
            knots <- knots[-ok]
        }
    }
    ns(x, df = df, knots = knots, intercept = intercept, Boundary.knots = Boundary.knots)
}
```

3.2 Incidence rates

We model the incidence rates from a tabulation of the diabetes register made in conjunction with the annual update of the register. The analytical units are the Lexis triangles, classified by age, period and cohort (◸ and ◿):

```
> load( "C:/Bendix/Steno/DM-register/NDR/2009/data/inc.Rdata" )
> head( inc )

  sex         A        P        C D upper        Y      dm.Y    well.Y
1   K 0.6666667 1995.333 1994.667 2     1 17100.67 0.1300479 17100.54
2   K 0.3333333 1995.667 1995.333 0     0 17025.50 0.0000000 17025.50
3   K 0.6666667 1996.333 1995.667 4     1 17069.17 1.8617385 17067.30
4   K 0.3333333 1996.667 1996.333 4     0 16469.50 1.4455852 16468.05
5   K 0.6666667 1997.333 1996.667 2     1 16501.83 1.9835729 16499.85
6   K 0.3333333 1997.667 1997.333 0     0 16434.00 0.0000000 16434.00

> with( inc, addmargins( xtabs( D ~ floor(P) + sex ) ) )

        sex
floor(P)      K      M    Sum
    1995   7145   7748  14893
    1996   7392   8022  15414
    1997   7533   7927  15460
    1998   8044   8814  16858
    1999   8555   9307  17862
    2000   8883   9631  18514
    2001   9477  10211  19688
    2002  10912  11197  22109
    2003  11506  12477  23983
    2004  11615  12553  24168
    2005  10636  11663  22299
    2006  10930  12092  23022
    2007  11763  12703  24466
    2008  12641  13995  26636
    2009  12297  14242  26539
    Sum  149329 162582 311911
```

Thus the dataset has the variable `D` for the number of DM-cases and the variable `well.Y` for the amount of follow-up among the non-diabetic part of the population, the latter is the correct denominator for analysis of the rates. The covariates of interest are of course `A`, `P` and `C` with values for age, period and cohort at the midpoints of the Lexis-triangles.

3.2.1 Models for incidence rates

Based on this we can now derive the location of the knots for this model:

```
> ( ki.a <- with( inc, quantile( rep(A,D), qn(nk.a) ) ) )
```

```
      5%      15%      25%      35%      45%      55%      65%      75%
31.66667 45.33333 52.33333 56.66667 60.66667 64.66667 68.66667 72.66667
     85%      95%
77.66667 84.66667
```

```
> ( ki.p <- with( inc, quantile( rep(P,D), qn(nk.p) ) ) )
```

```
     10%      30%      50%      70%      90%
1997.333 2000.667 2003.667 2006.333 2008.667
```

```
> ( ki.c <- with( inc, quantile( rep(C,D), qn(nk.c) ) ) )
```

```
   6.25%   18.75%   31.25%   43.75%   56.25%   68.75%   81.25%   93.75%
1918.667 1926.667 1933.333 1938.333 1943.667 1948.333 1955.667 1968.667
```

The model we set up is an model age-period-cohort model with these three terms in it. As we are only going to use the model for predictions we need not bother about parametrization issues, so it is not an issue that the model we fit is formally over-parametrized:

```
> m.inc.ap <- glm( D ~ Ns(A,knots=ki.a) + Ns(P,knots=ki.p),
+                      offset = log(well.Y), family=poisson,
+                      data = subset(inc,sex=="M") )
> f.inc.ap  <- update( m.inc.ap, data = subset(inc,sex=="K") )
> m.inc.apc <- update( m.inc.ap, . ~ . + detrend( Ns(P-A,kn=ki.c), P-A ) )
> f.inc.apc <- update( m.inc.apc, data = subset(inc,sex=="K") )
```

3.2.2 Incidence rate predictions

Finally we need the predicted incidence rates at a grid of points suitable for the calculations of predicted prevalences. We make the predictions for all combinations of `a.pt` and `p.pt`.

However, all the predictions should be in units of the interval length chosen for calculations. We note from the calculations above that the quantities that enter the expressions for the transition probabilities are all cumulative rates over the intervals. Thus we use a prediction data frame with the person-years-variables set to `int`:

```
> nd <- data.frame( A = rep(a.pt,length(p.pt)),
+                   P = rep(p.pt,each=length(a.pt)),
+              well.Y = int, dm.Y = int, Y = int )
> head( nd )
```

```
     A       P well.Y dm.Y   Y
1 0.05 1995.05    0.1  0.1 0.1
2 0.15 1995.05    0.1  0.1 0.1
3 0.25 1995.05    0.1  0.1 0.1
4 0.35 1995.05    0.1  0.1 0.1
5 0.45 1995.05    0.1  0.1 0.1
6 0.55 1995.05    0.1  0.1 0.1
```

Then we set up an array to hold the predicted incidence rates from the different models, separtely for the two sexes

```
> rnam <- list( a = a.pt,
+               p = p.pt,
+             sex = c("M","F"),
+              tp = c("apc","ap") )
> head( nd )
```

```
     A       P well.Y dm.Y   Y
1 0.05 1995.05    0.1  0.1 0.1
2 0.15 1995.05    0.1  0.1 0.1
3 0.25 1995.05    0.1  0.1 0.1
4 0.35 1995.05    0.1  0.1 0.1
5 0.45 1995.05    0.1  0.1 0.1
6 0.55 1995.05    0.1  0.1 0.1
```

```
> Lambda <- array( NA, dimnames=rnam, dim=sapply(rnam,length) )
> str( Lambda )
```

```
 logi [1:1000, 1:150, 1:2, 1:2] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:1000] "0.05" "0.15" "0.25" "0.35" ...
  ..$ p  : chr [1:150] "1995.05" "1995.15" "1995.25" "1995.35" ...
  ..$ sex: chr [1:2] "M" "F"
  ..$ tp : chr [1:2] "apc" "ap"
```

```
> Lambda[,,"M","ap" ] <- predict.glm( m.inc.ap , type="response", newdata=nd )
> Lambda[,,"F","ap" ] <- predict.glm( f.inc.ap , type="response", newdata=nd )
> Lambda[,,"M","apc"] <- predict.glm( m.inc.apc, type="response", newdata=nd )
> Lambda[,,"F","apc"] <- predict.glm( f.inc.apc, type="response", newdata=nd )
```

Thus we have the rates that we need for two sexes, and for two different modeling approaches.

3.3 Mortality rates

We have the mortality-rates in 1×1 age×period A-sets (□)

```
> load( "C:/Bendix/Steno/DM-register/NDR/2009/data/pop.mort.Rdata" )
> str( pop.mort )
```

```
'data.frame':        3000 obs. of  9 variables:
 $ sex  : Factor w/ 2 levels "K","M": 1 1 1 1 1 1 1 1 1 1 ...
 $ A    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ P    : num  1995 1996 1997 1998 1999 ...
 $ dm.D : num  0 0 0 0 0 0 0 0 0 0 ...
 $ dm.Y : num  0.13 3.307 1.984 0 0.877 ...
 $ pop.D: num  153 157 163 148 112 149 156 125 121 130 ...
 $ pop.Y: num  34126 33539 32936 32534 32303 ...
 $ D    : num  153 157 163 148 112 149 156 125 121 130 ...
 $ Y    : num  34126 33535 32934 32534 32302 ...
```

```
> head( pop.mort )
```

```
  sex A    P dm.D      dm.Y pop.D    pop.Y   D        Y
1   K 0 1995    0 0.1300479   153 34126.17 153 34126.04
2   K 0 1996    0 3.3073238   157 33538.67 157 33535.36
3   K 0 1997    0 1.9835729   163 32935.83 163 32933.85
4   K 0 1998    0 0.0000000   148 32533.67 148 32533.67
5   K 0 1999    0 0.8767967   112 32303.33 112 32302.46
6   K 0 2000    0 0.7214237   149 32575.17 149 32574.45
```

thus we have the number of deaths and person-time among diabetes patients in `dm.D` and `dm.Y` respectively, and the same quantities for the non-diabetic part of the population in `D` and `Y`.

3.3.1 Diabetes patients

First we fix the position of knots by age, period and cohort as we did for the incidence rates, and the we fit the same set of models, and make the same set of predictions, and put in a similarly defined array:

```
> ( kmd.a <- with( pop.mort, quantile( rep(A  ,dm.D), qn(nk.a) ) ) )

 5% 15% 25% 35% 45% 55% 65% 75% 85% 95%
 54  64  69  73  77  79  82  85  88  92

> ( kmd.p <- with( pop.mort, quantile( rep(P  ,dm.D), qn(nk.p) ) ) )

 10%  30%  50%  70%  90%
1996 2000 2003 2006 2008

> ( kmd.c <- with( pop.mort, quantile( rep(P-A,dm.D), qn(nk.c) ) ) )

 6.25% 18.75% 31.25% 43.75% 56.25% 68.75% 81.25% 93.75%
  1910   1916   1920   1923   1927   1931   1937   1947

> m.md.ap <- glm( dm.D ~ Ns(A,knots=kmd.a) + Ns(P,knots=kmd.p),
+                          offset = log(dm.Y), family = poisson,
+                          data = subset( pop.mort, sex=="M" & dm.Y>0 ) )
> m.md.apc <- update( m.md.ap, . ~ . + detrend( Ns(P-A,kn=kmd.c), P-A ) )
> f.md.ap  <- update( m.md.ap , data = subset(pop.mort,sex=="K" & dm.Y>0 ) )
> f.md.apc <- update( m.md.apc, data = subset(pop.mort,sex=="K" & dm.Y>0 ) )
> Mu.DM <- array( NA, dimnames=rnam, dim=sapply(rnam,length) )
> Mu.DM[,,"M","ap" ] <- predict.glm( m.md.ap , type="response", newdata=nd )
> Mu.DM[,,"F","ap" ] <- predict.glm( f.md.ap , type="response", newdata=nd )
> Mu.DM[,,"M","apc"] <- predict.glm( m.md.apc, type="response", newdata=nd )
> Mu.DM[,,"F","apc"] <- predict.glm( f.md.apc, type="response", newdata=nd )
```

3.3.2 The non-diabetic population

The mortalities in the non-diabetic population are modeled in exactly the same way

```
> ( kmw.a <- with( pop.mort, quantile( rep(A  ,D), qn(nk.a) ) ) )
```

```
 5% 15% 25% 35% 45% 55% 65% 75% 85% 95%
 45  60  67  72  77  80  83  86  89  94
```

```
> ( kmw.p <- with( pop.mort, quantile( rep(P  ,D), qn(nk.p) ) ) )
```

```
 10%  30%  50%  70%  90%
1996 1999 2002 2005 2008
```

```
> ( kmw.c <- with( pop.mort, quantile( rep(P-A,D), qn(nk.c) ) ) )
```

```
 6.25% 18.75% 31.25% 43.75% 56.25% 68.75% 81.25% 93.75%
  1908   1913   1918   1922   1926   1931   1940   1954
```

```
> m.mw.ap <- glm( D ~ Ns(A,knots=kmw.a) + Ns(P,knots=kmw.p),
+                       offset = log(Y), family = poisson,
+                       data = subset(pop.mort,sex=="M" & Y>0 ) )
> m.mw.apc <- update( m.mw.ap, . ~ . + detrend( Ns(P-A,kn=kmw.c), P-A ) )
> f.mw.ap  <- update( m.mw.ap , data = subset(pop.mort,sex=="K" & Y>0 ) )
> f.mw.apc <- update( m.mw.apc, data = subset(pop.mort,sex=="K" & Y>0 ) )
> Mu.W <- array( NA, dimnames=rnam, dim=sapply(rnam,length) )
> Mu.W[,,"M","ap" ] <- predict.glm( m.mw.ap , type="response", newdata=nd )
> Mu.W[,,"F","ap" ] <- predict.glm( f.mw.ap , type="response", newdata=nd )
> Mu.W[,,"M","apc"] <- predict.glm( m.mw.apc, type="response", newdata=nd )
> Mu.W[,,"F","apc"] <- predict.glm( f.mw.apc, type="response", newdata=nd )
```

For convenience of calculations we save the estimated rates and other quantities of interest:

```
> save( Lambda, Mu.W, Mu.DM, a.pt, p.pt, int, qn, file="estimates.etc.Rdata" )
```

Chapter 4

Prevalences

```
> load("estimates.etc.Rdata")
```

4.1 Transition probabilities

In order to get the predicted *number* of persons by age, period and prediction type, we need the (1-step) transition matrices at all combinations of age (a) and date (p), this is put in array:

```
> states <- c("Well","DM")
> TR <- NArray( c( dimnames(Lambda),
+                  list( from = states,
+                          to = states,
+                       scene = c("obs","m-fix","i-fix","all-f" ) ) ) )
> str( TR )
```

```
 logi [1:1000, 1:150, 1:2, 1:2, 1:2, 1:2, 1:4] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 7
  ..$ a    : chr [1:1000] "0.05" "0.15" "0.25" "0.35" ...
  ..$ p    : chr [1:150] "1995.05" "1995.15" "1995.25" "1995.35" ...
  ..$ sex  : chr [1:2] "M" "F"
  ..$ tp   : chr [1:2] "apc" "ap"
  ..$ from : chr [1:2] "Well" "DM"
  ..$ to   : chr [1:2] "Well" "DM"
  ..$ scene: chr [1:4] "obs" "m-fix" "i-fix" "all-f"
```

The situation where both the mortality rates and incidence rates are fixed at the 1995 level is trivial, because transition probabilities in that case only depend on age and not on period.

In order to fill this in, we need the cumulative incidences over intervals of length `int`. But these were exactly the ones we predicted in the previous sections by setting the person years equal to `int` in the data frame supplied to the `newdata` argument.

So we can now compute the one-`int`-step transition matrices for every combination of `a.pt` (in this case 0–100) and `p.pt` (in this case 1995–2010), both in steps of `int` (in this case 0.1 year).

```
> TR[,,,,"Well","Well","obs"] <- exp(-Lambda-Mu.W)
> TR[,,,,"Well","DM"  ,"obs"] <- exp(-Lambda-Mu.W)*Lambda
> TR[,,,,"DM"  ,"Well","obs"] <- 0
> TR[,,,,"DM"  ,"DM"  ,"obs"] <- exp(-Mu.DM)
```

When we fix the mortalities at the 1995 level we just replace the expressions above with expressions where we replace the date dimension by `rep(1,150)`:

```
> TR[,,,,"Well","Well","m-fix"] <- exp(-Lambda-Mu.W[,rep(1,150),,])
> TR[,,,,"Well","DM"  ,"m-fix"] <- exp(-Lambda-Mu.W[,rep(1,150),,])*Lambda
> TR[,,,,"DM"  ,"Well","m-fix"] <- 0
> TR[,,,,"DM"  ,"DM"  ,"m-fix"] <- exp(-Mu.DM[,rep(1,150),,])
```

```
> TR[,,,,"Well","Well","i-fix"] <- exp(-Lambda[,rep(1,150),,]-Mu.W)
> TR[,,,,"Well","DM"  ,"i-fix"] <- exp(-Lambda[,rep(1,150),,]-Mu.W)*Lambda[,rep(1,150),,]
> TR[,,,,"DM"  ,"Well","i-fix"] <- 0
> TR[,,,,"DM"  ,"DM"  ,"i-fix"] <- exp(-Mu.DM)
```

```
> TR[,,,,"Well","Well","all-f"] <- exp(-Lambda[,rep(1,150),,]-Mu.W[,rep(1,150),,])
> TR[,,,,"Well","DM"  ,"all-f"] <- exp(-Lambda[,rep(1,150),,]-Mu.W[,rep(1,150),,])*Lambda[,rep(1,150
> TR[,,,,"DM"  ,"Well","all-f"] <- 0
> TR[,,,,"DM"  ,"DM"  ,"all-f"] <- exp(-Mu.DM[,rep(1,150),,])
```

4.2 Population prevalences

Finally, in order to get the machinery working, we need the observed prevalences and population size at the starting point, that is at 1.1.1995. These are available form the same tabulation of the diabetes register as before:

```
> load( "C:/Bendix/Steno/DM-register/NDR/2009/data/pr.Rdata" )
> levels( pr$sex ) <- c("F","M")
> str( pr )
```

```
'data.frame':        3200 obs. of  5 variables:
 $ sex: Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
 $ A  : num  0.5 1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5 ...
 $ P  : num  1995 1995 1995 1995 1995 ...
 $ X  : num  0 4 6 5 13 13 20 18 29 27 ...
 $ N  : num  34094 32967 33198 31738 31361 ...
```

```
> head( pr )
```

```
   sex   A    P  X     N
1    F 0.5 1995  0 34094
17   F 1.5 1995  4 32967
33   F 2.5 1995  6 33198
49   F 3.5 1995  5 31738
65   F 4.5 1995 13 31361
81   F 5.5 1995 13 30440
```

These are prevalences for each of the 16 dates 1.1.1995 – 1.1.2010 in 1-year intervals, but to get the machinery running we will need the number of diabetes cases in intervals of length `int`.

So we model the prevalence as of 1.1.1995 as a smooth function of age, and use the predicted prevalences to produce the number of diabetes patients in each age-class.

We model the population prevalences at 1st January 1995, 2000, 2005 and 2010, using a log-link binomial model with a smooth spline with 20 knots:

```
> ( kp.a <- c( 10, with( pr, quantile( rep(A,X), qn(15) ) ) ) )
```

```
          3.333333%       10% 16.66667% 23.33333%       30% 36.66667% 43.33333% 
     10.0      28.5      40.5      47.5      52.5      55.5      58.5      61.5 
      50% 56.66667% 63.33333%       70% 76.66667% 83.33333%       90% 96.66667% 
     64.5      67.5      69.5      72.5      75.5      78.5      82.5      87.5 
```

```
> pr.fit <- NArray( c( dimnames(Lambda)[c(1,3)],
+                       list( date = 1995+0:3*5 ) ) )
> str( pr.fit )
```

```
 logi [1:1000, 1:2, 1:4] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 3
  ..$ a   : chr [1:1000] "0.05" "0.15" "0.25" "0.35" ...
  ..$ sex : chr [1:2] "M" "F"
  ..$ date: chr [1:4] "1995" "2000" "2005" "2010"
```

So once we have set up the array to hold the predicted prevalences we can fill them in:

```
> for( sx in dimnames(pr.fit)[["sex"]] )
+ for( dt in dimnames(pr.fit)[["date"]] )
+ pr.fit[,sx,dt] <- predict( glm( cbind(X,N-X) ~Ns( A, kn=kp.a ),
+                                  family = binomial(link="log"),
+                                  data = subset( pr, sex==sx & P==as.numeric(dt) ) ),
+                             newdata = data.frame( A=as.numeric(dimnames(pr.fit)[[1]]) ),
+                                type = "response" )
> round( ftable( pr.fit[700+1:10,,], row.vars=1 )*100, 2 )
```

```
      sex      M                       F                  
      date  1995  2000  2005  2010  1995  2000  2005  2010
a                                                         
70.05       6.05  8.88 13.31 16.42  4.97  7.14 10.53 12.63
70.15       6.06  8.89 13.34 16.46  5.00  7.17 10.56 12.67
70.25       6.08  8.90 13.36 16.50  5.02  7.19 10.59 12.72
70.35       6.09  8.91 13.39 16.55  5.04  7.21 10.63 12.77
70.45       6.11  8.93 13.41 16.59  5.07  7.23 10.65 12.82
70.55       6.12  8.94 13.43 16.63  5.09  7.26 10.68 12.87
70.65       6.14  8.96 13.45 16.68  5.12  7.28 10.71 12.92
70.75       6.15  8.97 13.47 16.73  5.14  7.29 10.73 12.97
70.85       6.16  8.99 13.49 16.77  5.17  7.31 10.76 13.02
70.95       6.18  9.01 13.50 16.82  5.20  7.33 10.78 13.07
```

4.3 Prediction of the actually observed prevalences

Note that we do not need to predict the population size; we only predict the prevalences as fractions. When we multiply the fraction of persons in states (Well,DM) with the transition matrix, we get fraction of the persons in the previous state that are in states (W,DM), which does not sum to 1 (because of the dead ones), so we just rescale to prevalence age in each step.

First we set up an array to hold the predicted prevalences under different scenarios:

```
> prv <- NArray( c( dimnames(Lambda)[1:4], dimnames(TR)["scene"] ) )
> str( prv )
```

```
 logi [1:1000, 1:150, 1:2, 1:2, 1:4] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 5
  ..$ a    : chr [1:1000] "0.05" "0.15" "0.25" "0.35" ...
  ..$ p    : chr [1:150] "1995.05" "1995.15" "1995.25" "1995.35" ...
  ..$ sex  : chr [1:2] "M" "F"
  ..$ tp   : chr [1:2] "apc" "ap"
  ..$ scene: chr [1:4] "obs" "m-fix" "i-fix" "all-f"
```

```
> prod( dim(prv) )
```

```
[1] 2400000
```

Thus we must make a loop that updates the prevalences at 1.1.1995 to those 1 year later, but first we must initialize the prevalences as modeled on 1.1.1995:

```
> prv[,1,,,] <- pr.fit[,,1]
> prv[1,,"M",,] <- pr.fit[1,"M",1]
> prv[1,,"F",,] <- pr.fit[1,"F",1]
> round( ftable( prv[c(1:2,700+1:3),1:2,,,], col.vars=4:5 )*100, 2 )
```

```
                 tp     apc                          ap                     
                 scene   obs m-fix i-fix all-f   obs m-fix i-fix all-f
a     p       sex                                                         
0.05  1995.05 M         0.04  0.04  0.04  0.04 0.04  0.04  0.04  0.04
              F         0.03  0.03  0.03  0.03 0.03  0.03  0.03  0.03
      1995.15 M         0.04  0.04  0.04  0.04 0.04  0.04  0.04  0.04
              F         0.03  0.03  0.03  0.03 0.03  0.03  0.03  0.03
0.15  1995.05 M         0.04  0.04  0.04  0.04 0.04  0.04  0.04  0.04
              F         0.03  0.03  0.03  0.03 0.03  0.03  0.03  0.03
      1995.15 M           NA    NA    NA    NA   NA    NA    NA    NA
              F           NA    NA    NA    NA   NA    NA    NA    NA
70.05 1995.05 M         6.05  6.05  6.05  6.05 6.05  6.05  6.05  6.05
              F         4.97  4.97  4.97  4.97 4.97  4.97  4.97  4.97
      1995.15 M           NA    NA    NA    NA   NA    NA    NA    NA
              F           NA    NA    NA    NA   NA    NA    NA    NA
70.15 1995.05 M         6.06  6.06  6.06  6.06 6.06  6.06  6.06  6.06
              F         5.00  5.00  5.00  5.00 5.00  5.00  5.00  5.00
      1995.15 M           NA    NA    NA    NA   NA    NA    NA    NA
              F           NA    NA    NA    NA   NA    NA    NA    NA
70.25 1995.05 M         6.08  6.08  6.08  6.08 6.08  6.08  6.08  6.08
              F         5.02  5.02  5.02  5.02 5.02  5.02  5.02  5.02
      1995.15 M           NA    NA    NA    NA   NA    NA    NA    NA
              F           NA    NA    NA    NA   NA    NA    NA    NA
```

So now it is checked that we have put the initial values correctly into `prv`. Then we can just compute the predicted prevalences under the different scenarios:

```
> for( sc in dimnames(TR)[["scene"]] )
+ for( ip in 1:(length(p.pt)-1) )
+ for( ia in 1:(length(a.pt)-1) )
+ prv[ia+1,ip+1,,,sc] <-
+   (    prv[ia,ip,,,sc]  * TR[ia,ip,,,"DM"  ,"DM",sc]
+    +(1-prv[ia,ip,,,sc]) * TR[ia,ip,,,"Well","DM",sc] ) /
+   (    prv[ia,ip,,,sc]  * TR[ia,ip,,,"DM"  ,"DM",sc]
+    +(1-prv[ia,ip,,,sc]) * TR[ia,ip,,,"Well","DM",sc]
+    +(1-prv[ia,ip,,,sc]) * TR[ia,ip,,,"Well","Well",sc] )
> prv[1:10,1:5,1,1,1]
```

```
      p
a          1995.05      1995.15      1995.25      1995.35      1995.45
  0.05 0.0003904028 0.0003904028 0.0003904028 0.0003904028 0.0003904028
  0.15 0.0003947364 0.0003984537 0.0003984744 0.0003984950 0.0003985158
  0.25 0.0003991182 0.0004028463 0.0004065840 0.0004066255 0.0004066670
  0.35 0.0004035487 0.0004072873 0.0004110360 0.0004147944 0.0004148568
  0.45 0.0004080283 0.0004117775 0.0004155369 0.0004193063 0.0004230856
  0.55 0.0004125576 0.0004163172 0.0004200873 0.0004238677 0.0004276581
  0.65 0.0004171372 0.0004209071 0.0004246878 0.0004284790 0.0004322805
  0.75 0.0004217676 0.0004255477 0.0004293388 0.0004331408 0.0004369533
  0.85 0.0004264495 0.0004302396 0.0004340411 0.0004378536 0.0004416770
  0.95 0.0004311833 0.0004349834 0.0004387950 0.0004426180 0.0004464522
```

```
> save( a.pt, prv, file="prv.Rdata" )
```

With this initial prediction in place we can now check whether we have made a reasonable approximation to the observed prevalences.

In the array `prv` are all the prevalences as predicted form the prevalence in 1995 using the estimated incidences and mortalities, whereas we have the smoothed empirical prevalences in 1995, 2000, 2005 and 2010 in the array `pr.fit`:

Thus we have the predicted age-specific prevalences for men in say 2000 in `prv[,51,"M","apc","obs"]`, and the smoothed empirical in `pr.fit[,"M","2000"]`. We now plot these in the same plot:

```
> par( mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6, las=1 )
> matplot( a.pt, cbind(prv[,1:3*50,"M","ap" ,"obs"],
+                      prv[,1:3*50,"M","apc","obs"],
+                      pr.fit[,"M",])*100,
+          xlim=c(10,90), ylim=c(0,20), xlab="Age", ylab="Prevalence (%)",
+          type="l", col="blue", lty=rep(c(3:1),c(3,3,4)), lwd=3 )
> matplot( a.pt, cbind(prv[,1:3*50,"F","ap" ,"obs"],
+                      prv[,1:3*50,"F","apc","obs"],
+                      pr.fit[,"F",])*100,
+          xlim=c(10,90), ylim=c(0,20), xlab="Age", ylab="Prevalence (%)",
+          type="l", col="red", lty=rep(c(3:1),c(3,3,4)), lwd=3 )
```

Since the APC-models for rates provide a better fit we will use these in the reporting of the different scenarios.

We now compare the scenarios at year 2010:

```
> dimnames( prv )[["scene"]]
```

```
[1] "obs"   "m-fix" "i-fix" "all-f"
```

```
> par( mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6, las=1 )
> matplot( a.pt, prv[,150,"M","apc",]*100,
+          xlim=c(20,90), ylim=c(0,20), xlab="Age", ylab="Prevalence (%)",
+          type="l", lty=rep(c(1,0),2), lwd=c(4,4,2,2)+1, col="blue" )
> matlines( a.pt, prv[,150,"M","apc",]*100,
+          type="l", lty=rep(c("11","22"),2), lwd=c(4,4,2,2)+1, col="blue" )
> matplot( a.pt, prv[,150,"F","apc",]*100,
+          xlim=c(20,90), ylim=c(0,20), xlab="Age", ylab="Prevalence (%)",
+          type="l", lty=rep(c(1,0),2), lwd=c(4,4,2,2)+1, col="red" )
> matlines( a.pt, prv[,150,"F","apc",]*100,
+          type="l", lty=rep(c("11","22"),2), lwd=c(4,4,2,2)+1, col="red" )
```

```
null device
          1
```

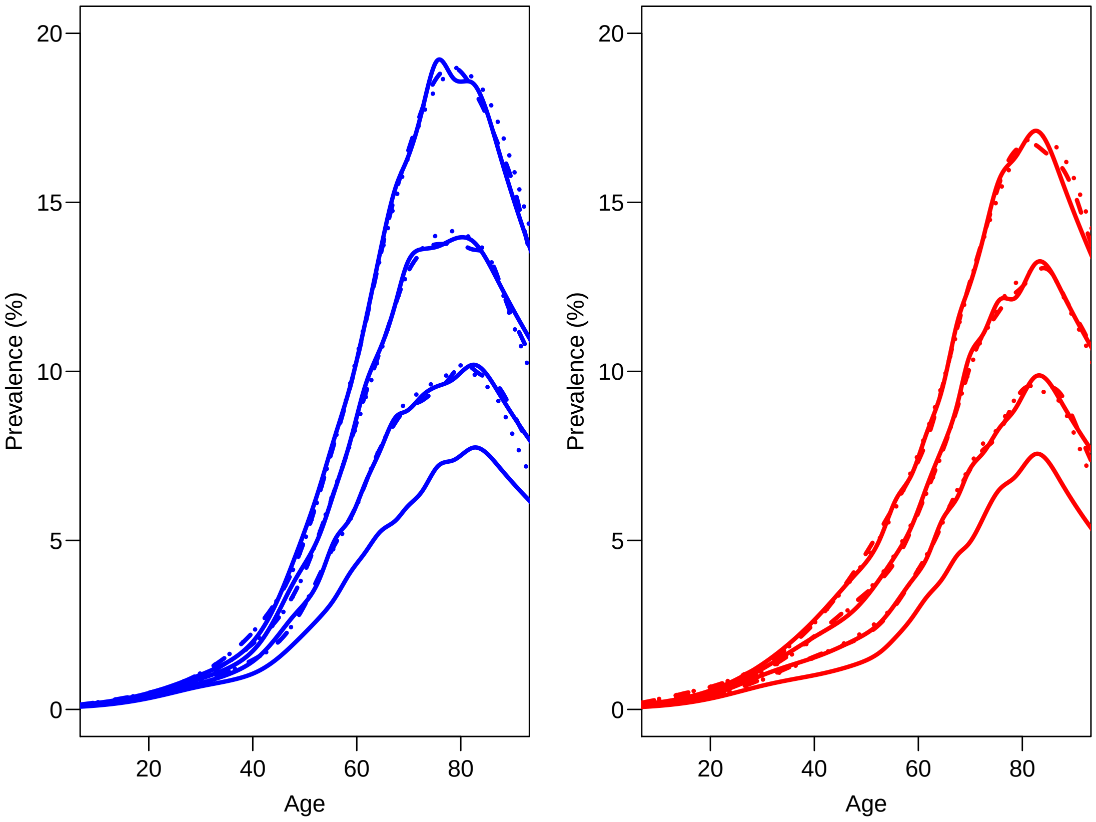

Figure 4.1: *Plot of observed (full lines) and predicted prevalences using simple age-period-models (dotted lines) or age-period-cohort models (broken lines). Clearly the broken lines gives a better approximation to the smoothed empirical rates.*

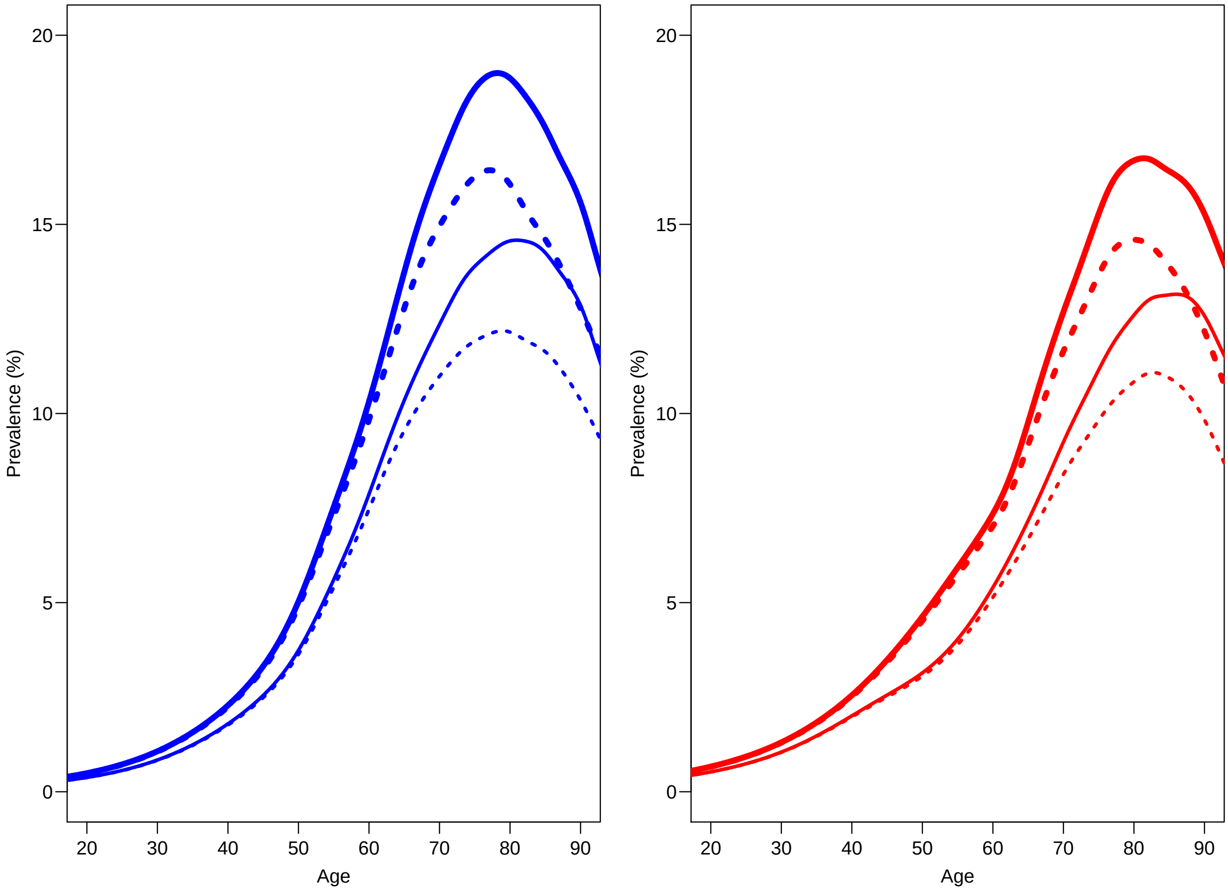

Figure 4.2: *The predicted prevalences under different scenarios:*
Full lines: Mortality rates evolve as observed, Broken lines: Mortality rates remain as 1995. Thick lines: Incidence rates evolve as observed, Thin lines: Incidence rates remain as in 1995.