Rates, extrapolations and components of prevalence from the resurrected Danish Diabetes Register

SDC
October 2017
http://bendixcarstensen.com/DMreg
Version 3

Compiled Wednesday 6th December, 2017, 02:21
from: /home/bendix/sdc/DMreg/NewReg/r/RevPred.tex

Bendix Carstensen Senior statistician, Clinical Epidemiology
Steno Diabetes Center Copenhagen, Gentofte, Denmark
& Department of Biostatistics, University of Copenhagen
<bcar0029@regionh.dk> <b@bxc.dk>
http://BendixCarstensen.com

Contents

Chapter 1

Data

The maintenance of the National Diabetes Register (NDR) has been discontinued by the Health Data Authority (Sundhedsdatastyrelsen). It is being replaced by the Register of Selected Chronic Diseases (RUKS — Register for Udvalgte Kroniske Sygdomme) which however does not encompass precisely the same persons.

A replacement of the NDR with greater precision than both RUKS and NDR has therefore beeen constructed; it is documented in the report http://BendixCarstensen.com/DMreg/NewReg.pdf, which also documents the construction of the follow-up and prevalence data used.

The following is based on the reconstructed version of a Danish diabetes register compiled as part of project 705093 at DST.

Chapter 2

Prerequisites and Prevalence

2.1 Prerequisites

We will make models for the rates as observed in small intervals of age and calendar time, so we start by specifying the interval length, and then the points at which we want to predict. The transition rates are labeled by the midpoints of the Lexis squares (of width `int`) where we predict them (`a.pt` and `p.pt`), and the prevalences by the midpoints of the age-classes (`a.pt` and the time-points `t.pt` — note that we shall make predictions of rates all the way to 2030 — this will mainly be with the purpose of predicting the total number of DM persons in the population:

```
> int <- 1/12
> a.pt <- seq(0,100,int)[-1] - int/2
> t.pt <- seq(1996,2040,int)
> p.pt <- t.pt[-1] - int/2
```

We shall model all the rates by age-period-cohort models separately for men and women, both on a multiplicative and an additive scale. As a sensitivity analysis we will also model the rates only by an age-period model.

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 <- 8
> nk.p <- 6
> nk.c <- 6
```

Placing of knots

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, 1 )
 [1] 0.1000000 0.1888889 0.2777778 0.3666667 0.4555556 0.5444444 0.6333333 0.7222222
 [9] 0.8111111 0.9000000
```

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

For data on age and period that only take on a finite number of values (100 for age and 40 for date), we may want to place knots for spline at intermediate points. So here is an algorithm that from a vector `x` creates a vector of value evenly over the intervals around each distinct value of `x`:

```
> spread <-
+ function(x)
+    {
+ nx <- numeric( length(x) )    # where the result goes
+ tx <- table(x)                # how many of each value
+ ux <- as.numeric(names(tx))   # the unique values of x (ordered)
+ cx <- cumsum(c(0,tx))         # the indices of the result
+ dx <- diff(ux)                # difference between successive values
+ nd <- length(dx)              # no of differences
+ lo <- ux - dx[c(1,1:nd   )]/2 # lower endpoints of intervals
+ hi <- ux + dx[c(  1:nd,nd)]/2 # upper endpoints of intervals
+ for( i in 1:length(ux)) nx[(cx[i]+1):cx[i+1]] <- runif(tx[i],lo[i],hi[i])
+ sort( nx )
+    }
```

Nice printing

Finally, we define a few utility functions to write nice numbers with comma separators; `fC` is a utility, `fcP` prints numbers, and `fCtable` formats a `ftable`:

```
> fC <- function( x, d=0, w=8 ) formatC( x,
+                                  format = "f",
+                                big.mark = ",",
+                                  digits = d,
+                                   width = w )
> fCp     <- function( x, d=0, w=8, ... ) noquote( fC( x, d=d, w=w ), ... )
> fCtable <- function( x, d=0, w=8, ... )  ftable( fC( x, d=d, w=w ), ... )
```

We save it all for further use:

```
> save( qn, spread, fC, fCp, fCtable,            # functions
+       int, a.pt, t.pt, p.pt, nk.a, nk.p, nk.c, # numbers and data
+       file="./data/inits.Rda" )
```

2.2 Population and forecasts from DST

```
> library( Epi )
```

2.2.1 Population figures

Here are the population figures for the entire country 1996-2017 extracted from the data bank of Statistics Denmark (DST):

```
> bef <- read.csv2( "../data/bef1995.csv", header=FALSE )
> names( bef )[1:2] <- c("sex","A")
> bef$sex <- Relevel(bef$sex,2:1)
> befp <- reshape( bef, direction = "long",
+                          varying = 3:25,
+                            times = 1995:2017,
+                          timevar = "P",
+                          v.names = "N" )[,1:4]
> rownames( befp ) <- NULL
```

However we will not be concerned with centenarians (yet!), so we truncate the data, and label the data frame:

```
> befp <- subset( befp, A<100 )
> attr( befp, "Label" ) <- "DK population figures 1.1.1995-2017"
> str( befp )
'data.frame':        4600 obs. of  4 variables:
 $ sex: Factor w/ 2 levels "M","F": 1 1 1 1 1 1 1 1 1 1 ...
 $ A  : int  0 1 2 3 4 5 6 7 8 9 ...
 $ P  : int  1995 1995 1995 1995 1995 1995 1995 1995 1995 1995 ...
 $ N  : int  35612 34747 35080 33328 32973 31898 30863 29539 29151 28304 ...
 - attr(*, "Label")= chr "DK population figures 1.1.1995-2017"
```

summary(befp) fCtable(addmargins(xtabs(N P + sex, data = befp), 2), w=10)

2.2.2 Population forecast

One prerequisite for the prediction is that we have some knowledge about the future population size of Denmark, so we have acquired this from the data bank of Statistics Denmark:

```
> bef <- read.csv2( "../data/bef2040.csv", header=FALSE )
> names( bef)[1:3] <- c("sex","reg","A")
> bef$sex <- Relevel(bef$sex,2:1)
> befr <- reshape( bef, direction = "long",
+                          varying = 4:27,
+                            times = 2017:2040,
+                          timevar = "P",
+                          v.names = "N" )[,1:5]
> rownames( befr ) <- NULL
> beff <- aggregate( befr$N, befr[,c("sex","A","P")], FUN=sum )
> names( beff )[4] <- "N"
```

Again, we omit the centenarians:

```
> beff <- subset( beff, A<100 )
> attr( beff, "Label" ) <- "DK population forecasts 1.1.2017-2040"
> str( beff )
'data.frame':        4800 obs. of  4 variables:
 $ sex: Factor w/ 2 levels "M","F": 1 2 1 2 1 2 1 2 1 2 ...
 $ A  : int  0 0 1 1 2 2 3 3 4 4 ...
 $ P  : int  2017 2017 2017 2017 2017 2017 2017 2017 2017 2017 ...
 $ N  : int  31970 30054 30364 28889 30069 28451 29674 28302 31101 29187 ...
 - attr(*, "Label")= chr "DK population forecasts 1.1.2017-2040"
> summary( beff )
```

```
 sex            A               P               N
 M:2400   Min.   : 0.00   Min.   :2017   Min.   :  106
 F:2400   1st Qu.:24.75   1st Qu.:2023   1st Qu.:29379
          Median :49.50   Median :2028   Median :34292
          Mean   :49.50   Mean   :2028   Mean   :30193
          3rd Qu.:74.25   3rd Qu.:2034   3rd Qu.:37461
          Max.   :99.00   Max.   :2040   Max.   :45246
```

```
> fCtable( addmargins( xtabs( N ~ P + sex, data = beff ), 2 ), w=10 )
```

```
     sex           M           F         Sum
P
2017       2,859,967   2,887,581   5,747,548
2018       2,879,674   2,903,785   5,783,459
2019       2,896,601   2,918,257   5,814,858
2020       2,911,734   2,931,826   5,843,560
2021       2,925,660   2,944,824   5,870,484
2022       2,938,924   2,957,649   5,896,573
2023       2,951,933   2,970,556   5,922,489
2024       2,964,768   2,983,488   5,948,256
2025       2,977,487   2,996,356   5,973,843
2026       2,989,922   3,009,043   5,998,965
2027       3,002,017   3,021,447   6,023,464
2028       3,013,728   3,033,430   6,047,158
2029       3,024,967   3,044,924   6,069,891
2030       3,035,725   3,055,949   6,091,674
2031       3,046,018   3,066,384   6,112,402
2032       3,055,782   3,076,292   6,132,074
2033       3,065,093   3,085,635   6,150,728
2034       3,073,984   3,094,437   6,168,421
2035       3,082,439   3,102,691   6,185,130
2036       3,090,527   3,110,390   6,200,917
2037       3,098,191   3,117,511   6,215,702
2038       3,105,499   3,124,114   6,229,613
2039       3,112,513   3,130,243   6,242,756
2040       3,119,202   3,135,975   6,255,177
```

2.2.3 Population prevalence and forecast

In order to check that the projections are reasonably consistent with the actual population numbers we graph these:

```
> matplot( 1995:2017, xtabs( N ~ P + sex, data=befp )/10^6,
+          type="l", lwd=3, col=c("blue","red"), lty=1,
+          xlim=c(1995,2040), ylim=c(2.5,3.15), bty="n", las=1,
+          ylab="Population size", xlab="Date" )
> matlines( 2017:2040, xtabs( N ~ P + sex, data=beff )/10^6,
+          type="l", lwd=3, col=c("blue","red"), lty=1 )
> axis( side=1, at=seq(1995,2040,5), labels=NA, tcl=-0.3)
> text( 1995, 3.05+c(-1,1)/50, levels(befp$sex), col=c("blue","red"), adj=0 )
> abline(v=2017)
> abline(  h=25:31/10, col=gray(0.7), lty="22", lend="butt" )
```

In order to be useful in predictions, we would like a tabular representation of the population figures 1995–2040 as an array:

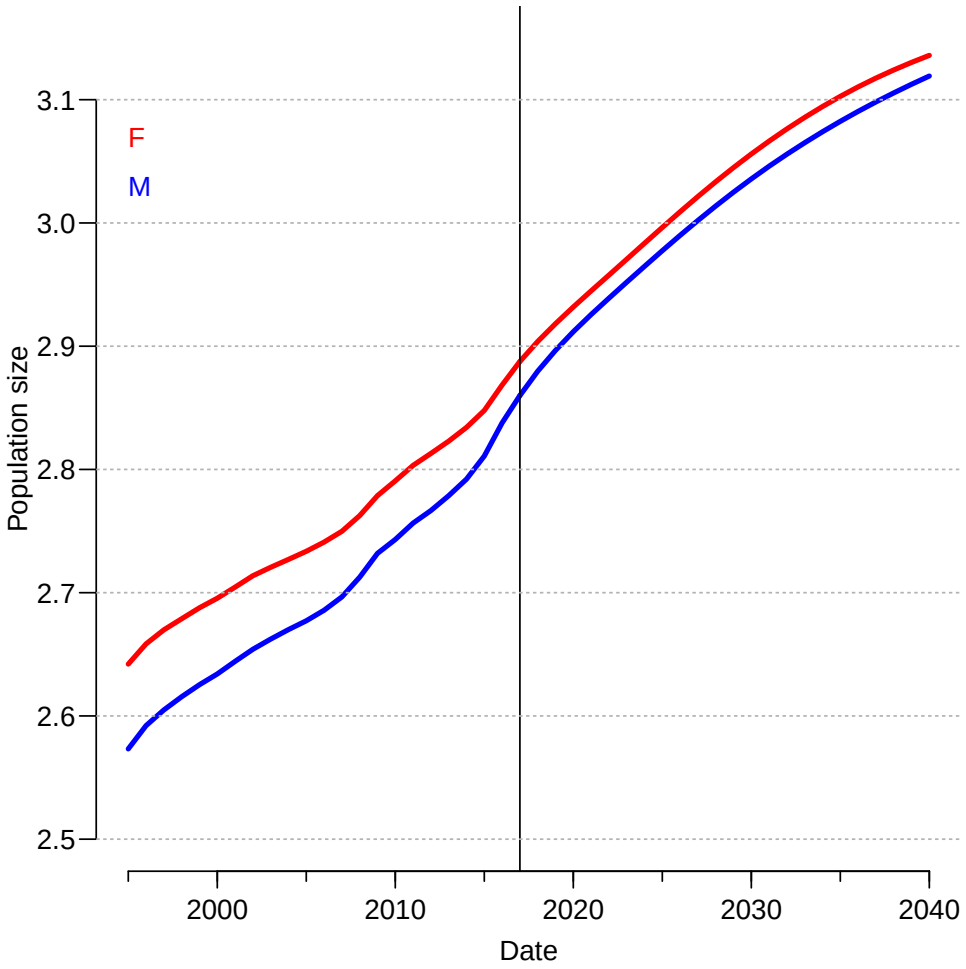

Figure 2.1: *Actual and forecast (after 2017) population size in Denmark, by sex.*
`./init-pop-size`

```
> pop <- xtabs( N ~ A + P + sex,
+               data = rbind( befp,
+                     subset( beff, P > 2017 ) ) )
> attr( pop, "Label" ) <- "DK population size 1995-2040 from DST"
> str( pop )
 'xtabs' int [1:100, 1:46, 1:2] 35612 34747 35080 33328 32973 31898 30863 29539 29151 28304
 - attr(*, "dimnames")=List of 3
  ..$ A  : chr [1:100] "0" "1" "2" "3" ...
  ..$ P  : chr [1:46] "1995" "1996" "1997" "1998" ...
  ..$ sex: chr [1:2] "M" "F"
 - attr(*, "call")= language xtabs(formula = N ~ A + P + sex, data = rbind(befp, subset(bef
 - attr(*, "Label")= chr "DK population size 1995-2040 from DST"
> save( befp, beff, pop, file = "../data/pop.Rda" )
```

2.2.4 Prevalence — data comparison

We retrieve the data needed: We have prevalent no of *persons* in the DK population from the project; her we make a comparison with the numbers from the DST data bank:

`prev.tex`

...now input from prev.tex

```
> fCtable( cbind( xtabs( N ~ P + sex, data=subset(Ptab,A<100 ) ),
+                 xtabs( N ~ P + sex, data=subset(befp,P<2017) ) ) )
```

	M	F	M	F
1995	2,577,348	2,644,348	2,573,246	2,642,061
1996	2,593,933	2,659,003	2,592,144	2,658,455
1997	2,605,838	2,669,688	2,604,849	2,669,840
1998	2,615,989	2,678,244	2,615,579	2,678,822
1999	2,625,377	2,686,747	2,625,336	2,687,736
2000	2,633,776	2,694,076	2,634,044	2,695,492
2001	2,644,359	2,703,324	2,644,241	2,704,486
2002	2,654,451	2,712,332	2,654,070	2,713,732
2003	2,663,262	2,719,286	2,662,337	2,720,597
2004	2,670,638	2,725,513	2,670,045	2,727,001
2005	2,678,122	2,732,331	2,677,199	2,733,578
2006	2,687,283	2,740,335	2,685,747	2,741,032
2007	2,698,927	2,749,680	2,696,567	2,749,802
2008	2,715,951	2,763,406	2,712,563	2,762,497
2009	2,736,892	2,781,097	2,731,906	2,778,735
2010	2,751,941	2,795,857	2,743,150	2,790,699
2011	2,768,093	2,810,606	2,756,443	2,803,285
2012	2,783,377	2,823,975	2,766,623	2,812,969
2013	2,799,152	2,837,119	2,778,686	2,822,933
2014	2,817,303	2,852,137	2,792,123	2,834,116
2015	2,841,794	2,870,495	2,810,857	2,847,836
2016	2,873,087	2,894,371	2,837,735	2,868,458

The relative percentwise difference is:

```
> fCtable( ( xtabs( N ~ P + sex, data=subset(Ptab,A<100 ) ) /
+            xtabs( N ~ P + sex, data=subset(befp,P<2017) ) - 1 )*100, w=8, d=1 )
```

sex	M	F
P		
1995	0.2	0.1
1996	0.1	0.0
1997	0.0	-0.0
1998	0.0	-0.0
1999	0.0	-0.0
2000	-0.0	-0.1
2001	0.0	-0.0
2002	0.0	-0.1
2003	0.0	-0.0
2004	0.0	-0.1
2005	0.0	-0.0
2006	0.1	-0.0
2007	0.1	-0.0
2008	0.1	0.0
2009	0.2	0.1
2010	0.3	0.2
2011	0.4	0.3
2012	0.6	0.4
2013	0.7	0.5
2014	0.9	0.6
2015	1.1	0.8
2016	1.2	0.9

Thus we see the the last few years have a bit to high population figures in `Ptab`, about 1% too high. For consistencty we will use the DST figures in our analyses, so we aggregate the number of DMcases from `Ptab`:

```
> ptab <- subset( Ptab, P>1995, select=c(sex,A,P,DM) )
> str( ptab )
'data.frame':        4580 obs. of  4 variables:
 $ sex: Factor w/ 2 levels "M","F": 1 1 1 1 1 1 1 1 1 1 ...
 $ A  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ P  : num  1996 1997 1998 1999 2000 ...
 $ DM : num  2 0 1 2 2 1 0 0 1 1 ...
> ptab <- merge( ptab, befp, all.y=TRUE )
> ptab$DM <- pmax( 0, ptab$DM, na.rm=TRUE )
> str( ptab )
'data.frame':        4600 obs. of  5 variables:
 $ sex: Factor w/ 2 levels "M","F": 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 : num  0 2 0 1 2 2 1 0 0 1 ...
 $ N  : int  35612 36054 34853 34774 34074 33906 34418 33479 32990 33351 ...
```

2.3 Modeling prevalence of diabetes

We will analyze age-specific prevalence for each sex and each 1st January 1995—2016 separately, even though the datasets used are not independent. We model the prevalences as of 1 January each of the years 1995—2012, as a smooth function of age using a log-link binomial model with a smooth spline with 16 knots.

Using the just defined `qn` function we get:

```
> ( kp.a <- c( 10, with( ptab, quantile( spread(rep(A,DM)), qn(15) ) ) ) )
          3.333333%       10% 16.66667% 23.33333%       30% 36.66667% 43.33333%       50%
 10.00000  27.24371  41.02605  47.35001  51.82561  55.43386  58.53341  61.30921  63.88217
56.66667% 63.33333%       70% 76.66667% 83.33333%       90% 96.66667%
 66.34175  68.77749  71.33921  74.16036  77.30294  81.10825  86.84129
```

We now set up arrays to hold the smoothed prevalences, one coarse array (1×1 year age×period classes) to be multiplied by the population figures, and one with the predefined age-class midpoints `a.pt`

```
> A.pt <- 1:100 - 0.5
> P.pt <- 1996:2016
> pr.fit <- NArray( list( sex = c("M","F"),
+                         A = A.pt,
+                         P = P.pt ) )
> pr.ini <- NArray( list( sex = c("M","F"),
+                         A = a.pt,
+                         P = P.pt ) )
```

So once we have set up the array to hold the smoothed empirical prevalences we can fit the models and fill the smoothed prevalences into `pr.fit` and `pr.ini`:

```
> for( sx in dimnames(pr.fit)[["sex"]] )
+    {
+    for( dt in dimnames(pr.fit)[["P"]] )
+       {
+ pr.mod <- glm( cbind(DM,N-DM) ~ Ns( A, kn=kp.a ),
+                family = binomial(link="log"),
+                  data = subset( ptab, sex==sx & P==as.numeric(dt) ) )
+ pr.fit[sx,,dt] <- predict( pr.mod,
+                            newdata = data.frame( A=A.pt ),
+                               type = "response" )
+ pr.ini[sx,,dt] <- predict( pr.mod,
+                           newdata = data.frame( A=a.pt ),
+                              type = "response" )
+       } # end of loop over dates
+    } # end of sex loop
```

In principle we could have obtained the same by fitting a 3-way interaction model between smooth age, categorical period and sex, but it would be a bit of a hazzle to tease out what we wanted.

We can plot how the age-specific prevalences have evolved over time:

```
> pr.obs <- xtabs( cbind(DM,N) ~ sex + P + A, data=ptab )[,,,c(1,2,2)]
> pr.obs[,,,3] <- pr.obs[,,,"DM"]/pr.obs[,,,"N"]*100
> dimnames(pr.obs)[[4]][3] <- "pct"
> str( pr.obs )
 table [1:2, 1:23, 1:100, 1:3] 0 0 2 3 0 6 1 1 2 0 ...
 - attr(*, "dimnames")=List of 4
  ..$ sex: chr [1:2] "M" "F"
  ..$ P  : chr [1:23] "1995" "1996" "1997" "1998" ...
  ..$ A  : chr [1:100] "0" "1" "2" "3" ...
  ..$    : chr [1:3] "DM" "N" "pct"
> plpr <- function( emp = NULL )
+   {
+ par( mfrow=c(1,2), mar=c(1,0,1,0), mgp=c(3,1,0)/1.6, las=1,
+      oma=c(2,3,0,1), bty="n" )
+ sxpl <- function( sx="M", tx="Men", cl="blue", ax2=TRUE )
+         {
+ matplot( A.pt, pr.fit[sx,,]*100,
+          ylim=c(0,22), xlim=c(20,95), yaxs="i", xaxt="n", yaxt="n", xlab="", ylab="",
+          type="l", lty=1, col=cl, lwd=c(2,1) )
+ abline( h=0:25, v=seq(0,100,5), col=gray(0.9) )
+ matlines( A.pt, pr.fit[sx,,]*100, type="l", lty=1, col=cl, lwd=c(2,1) )
+ if( !is.null(emp) ) for( i in emp ) lines( 0:99, pr.obs[sx,paste(i),,"pct"], type="s", lw
+ text( 25, 22, tx, adj=c(0,1), col=cl, cex=1.2 )
+ text( 89, pr.fit[sx,"89.5","1996"]* 99, "1996", col=cl, adj=c(1,1) )
+ text( 86, pr.fit[sx,"85.5","2016"]*101, "2016", col=cl, adj=c(0,0) )
+ axis( side=1 )
+ axis( side=1, at=seq(5,95,5), labels=NA, tcl=-0.3 )
+ axis( side=1, at=seq(0,90,10), labels=NA, tcl=-0.4 )
+ if( ax2 ) axis( side=2 )
+         }
+ sxpl()
+ sxpl( "F", "Women", "red", FALSE )
+   }
> plpr()
> plpr( c(1996,2016) )
> # plpr( 2010 )
```

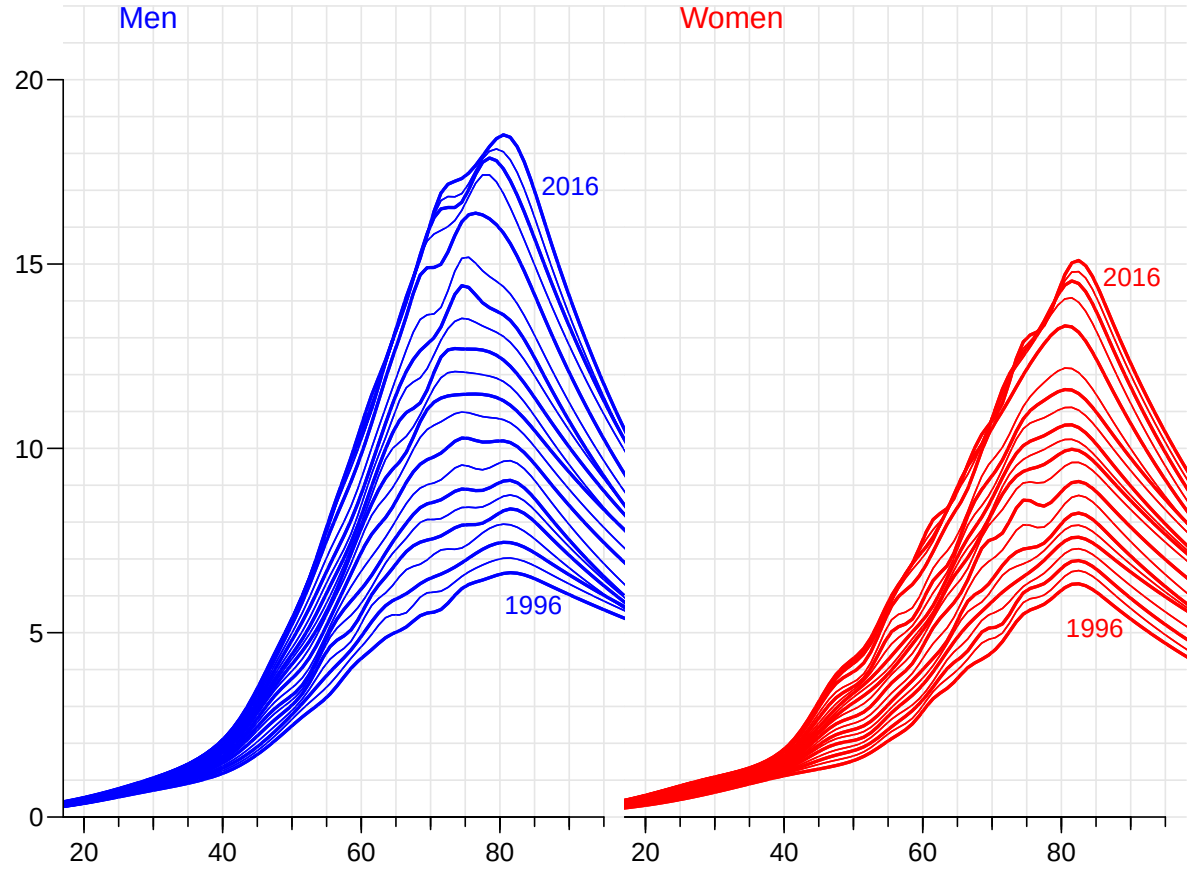

Figure 2.2: *Smoothed age-specific prevalences at 1 January each of the years 1996–2016. Blue is men, red is women; even years plotted with thick lines; thin step lines represents empirical prevalences as of 1 January 1996, resp. 2016* `./prev-obs-sm`

2.3.1 Trends in prevalence

A crude way of summarizing the chanhes in prevalences is to assume that relative change is constant from year to year. So we set up models that does this separately for men and women, and store the predicted values for comparison with those from the model with no assumption about the time evolution:

```
> pr.lfit <- pr.fit
> pr.chg <- NArray( list( dimnames(pr.fit)[["sex"]],
+                         c("% chg/y","lo","hi") ) )
> for( sx in dimnames(pr.fit)[["sex"]] )
+    {
+ lmod <- glm( cbind(DM,N-DM) ~ Ns( A, kn=kp.a ) + P,
+               family = binomial(link="log"),
+                 data = subset( ptab, sex==sx ) )
+ pr.chg[sx,] <- ( ci.exp( lmod, subset="P" ) - 1 ) * 100
+ pr.lfit[sx,,] <- predict( lmod,
+                           newdata = expand.grid( A=A.pt, P=P.pt ),
+                              type = "response" )
+    }
```

This model is of course a simplification of the model above, with an arbitrary age-date interaction, so we can have a peep at how the predicted prevalences looks:

```
> par( mfrow=c(1,2), mar=c(1,0,1,0), mgp=c(3,1,0)/1.6, las=1,
+      oma=c(2,3,0,1), bty="n" )
> lblu <- rgb( 3,3,4,max=4 )
> lred <- rgb( 4,3,3,max=4 )
> matplot( A.pt, pr.fit["M",,]*100,
+          ylim=c(0,22), xlim=c(20,90), yaxs="i", xaxt="n", yaxt="n", xlab="", ylab="",
+          type="n", lty=1, col="blue", lwd=c(1,2) )
> abline( h=0:25, v=seq(0,100,5), col=gray(0.9) )
> matlines( A.pt,  pr.fit["M",,]*100, type="l", lty=1, col=lblu  , lwd=c(2,3) )
> matlines( A.pt, pr.lfit["M",,]*100, type="l", lty=1, col="blue", lwd=c(2,3) )
> text( 25, 22, "Men", adj=c(0,1), col="blue", cex=1.2 )
> text( 89, pr.fit["M","89.5","1996"]* 99, "1996", col="blue", adj=c(1,1) )
> text( 80, pr.fit["M","80.5","2016"]*101, "2016", col="blue", adj=c(0,0) )
> axis( side=1 )
> axis( side=2 )
> matplot( A.pt, pr.fit["F",,]*100,
+          ylim=c(0,22), xlim=c(20,90), yaxs="i", xaxt="n", yaxt="n", xlab="", ylab="",
+          type="n", lty=1, col="red", lwd=c(1,2) )
> abline( h=0:25, v=seq(0,100,5), col=gray(0.9) )
> matlines( A.pt,  pr.fit["F",,]*100, type="l", lty=1, col=lred , lwd=c(2,3) )
> matlines( A.pt, pr.lfit["F",,]*100, type="l", lty=1, col="red", lwd=c(2,3) )
> text( 25, 22, "Women", adj=c(0,1), col="red", cex=1.2 )
> text( 89, pr.fit["F","89.5","1996"]* 99, "1996", col="red", adj=c(1,1) )
> text( 80, pr.fit["F","80.5","2016"]*101, "2016", col="red", adj=c(1,0) )
> axis( side=1 )
> mtext( "Age", side=1, line=1, outer=T )
> mtext( "DM prevalence (%)", side=2, line=2, outer=T, las=0 )
```

From figure 2.3 we see that for men the summary using a constant relative change in prevalence is not a very good summary of the change in prevalences; it does not capture the change in the age of peak prevalence of men from 85 in 1996 to 75 in 2011 and back again in 2016. So the overall estimate of some 4.6% in relative annual increase of prevalences over the 20-year period 1995–2016, is not providing an adequate summary:

```
> round( pr.chg, 2 )
  % chg/y   lo   hi
M    3.41 3.39 3.43
F    3.44 3.42 3.47
```

2.3.2 Prevalence age-period interaction

Hence the relevant description of average changes per year could be a model for the prevalences where we allow the relative change to vary smoothly by age, but require that the relative change is the same across years for any age. This is done by including an interaction between a spline term in age and period, and the subsequently fishing out the relative change using a spline basis with a bit fewer knots to fish out the period multiplier.

Note that this is particular form of interaction — namely one where percentwise changes at any age are the same across the time span, and moreover where it is assumed that size of this changes smoothly by age.

It goes as follows, where we also as before extract the predicted values for comparison with the prevalence curves fitted separately for each year:

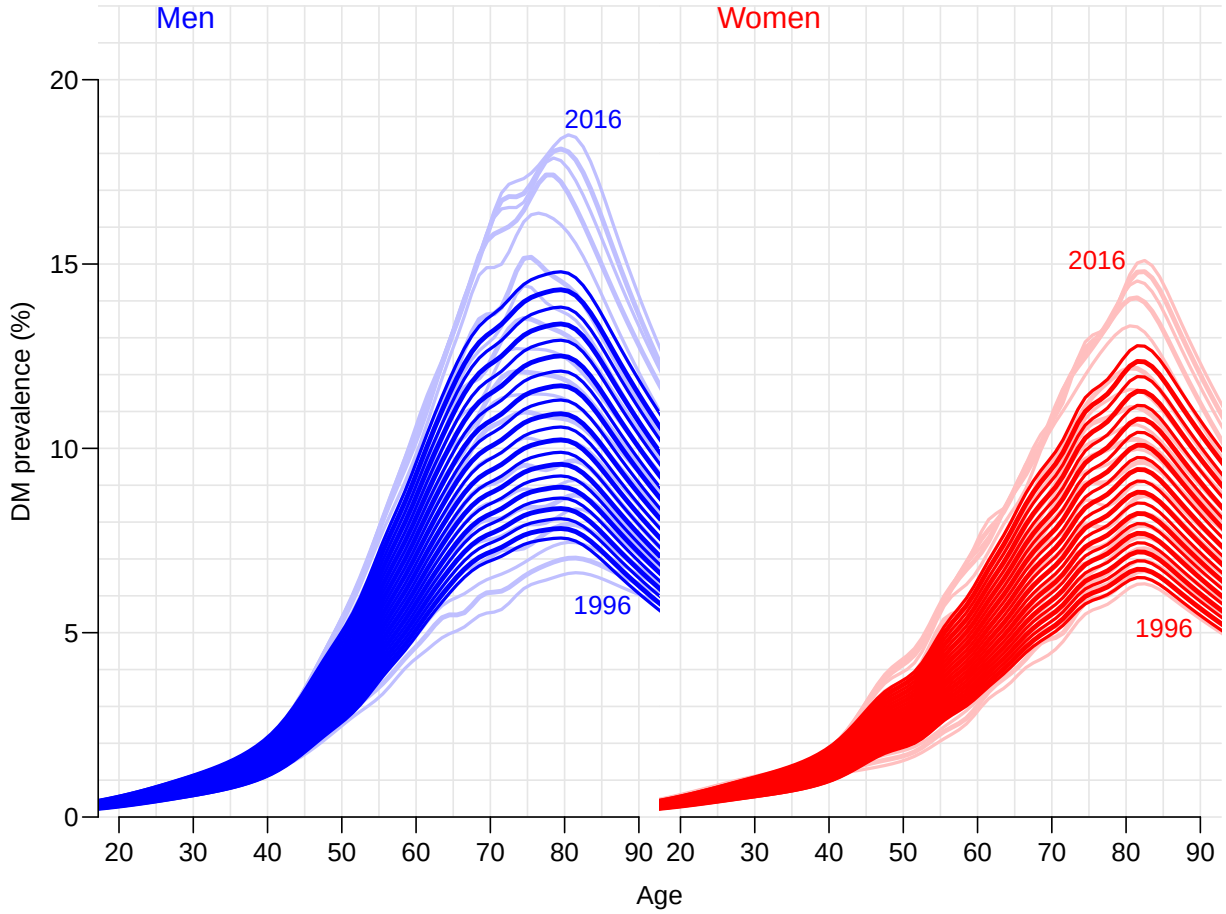

Figure 2.3: *Smoothed age-specific prevalences at 1 January 1996–2016 using a model with constant annual relative change in prevalences (full color) compared to the smoothing of the single years (pale color). Blue is men, red is women.* ./prev-lp-trend

```
> pr.ifit <- pr.fit
> ( kx.a <- c( 10, with( ptab, quantile( rep(A,DM), qn(5) ) ) ) )
    10% 30% 50% 70% 90%
 10  41  55  64  71  81
> CA <- Ns( A.pt, kn=kx.a, intercept=TRUE )
> A.chg <- NArray( list( A.pt, c("Est","lo","hi"), sex=c("M","F") ) )
> for( sx in dimnames(pr.fit)[["sex"]] )
+    {
+ limod <- glm( cbind(DM,N-DM) ~ Ns( A, kn=kp.a ) +
+                       I(P-2000):Ns( A, kn=kx.a, intercept=TRUE ),
+                family = binomial(link="log"),
+                  data = subset( ptab, sex==sx ) )
+ A.chg[,,sx] <- ci.exp( limod, subset="P", ctr.mat=CA )
+ pr.ifit[sx,,] <- predict( limod,
+                           newdata = expand.grid( A=A.pt, P=P.pt ),
+                              type = "response" )
+    }
> str( A.chg )
 num [1:100, 1:3, 1:2] 1.02 1.02 1.02 1.02 1.02 ...
 - attr(*, "dimnames")=List of 3
  ..$    : chr [1:100] "0.5" "1.5" "2.5" "3.5" ...
```

```
  ..$    : chr [1:3] "Est" "lo" "hi"
  ..$ sex: chr [1:2] "M" "F"
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6, bty="n", las=1 )
> matplot( A.pt, ( cbind(A.chg[,,"M"],A.chg[,,"F"]) - 1 )*100,
+          col=rep(c("blue","red"),each=3), lwd=c(3,1,1), lty=1, type="l",
+          ylim=c(0,8), yaxs="i", xlim=c(20,90),
+          ylab="Annual change in DM prevalence (%)", xlab="Age" )
> abline( h=pr.chg[,1], col=c("blue","red") )
```

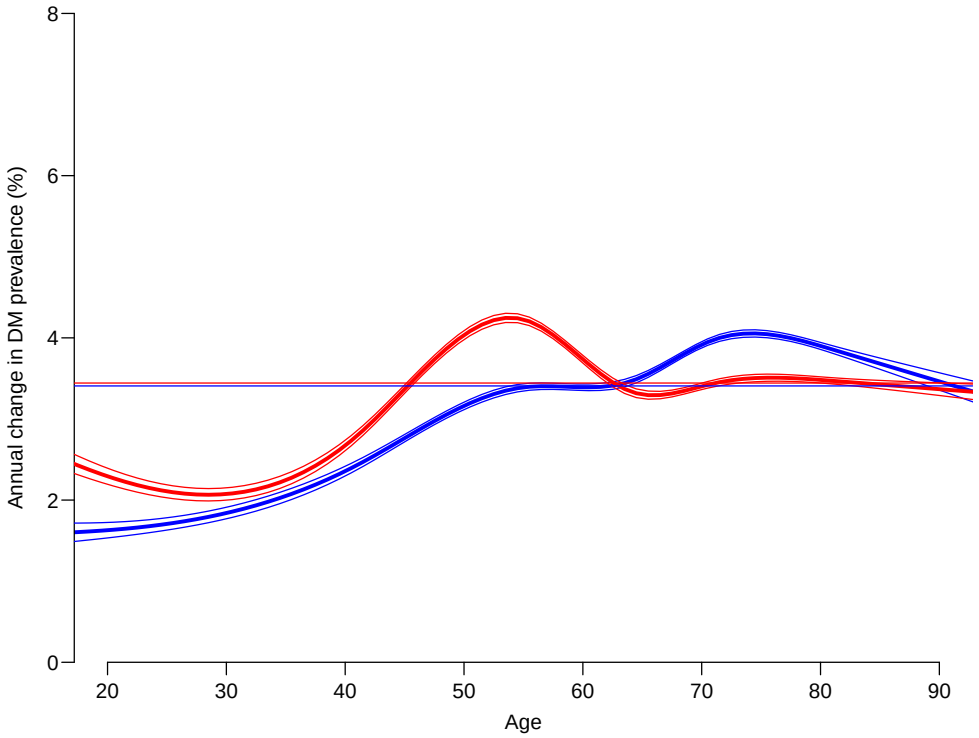

Figure 2.4: *The estimated change in prevalence in different ages, separately for men (blue) and women (red). The horizontal lines indicate the estimate from the naïve model with constant change for all ages.* `./prev-ipl-ch`

We can also, as with the naïve linear change model, show how the fitted values under this interaction model looks relative to the separate analyses by year (or full interaction model). The code is exactly as before, because we put the fitted values into the same structure as before:

```
> par( mfrow=c(1,2), mar=c(1,0,1,0), mgp=c(3,1,0)/1.6, las=1,
+      oma=c(2,3,0,1), bty="n" )
> lblu <- rgb( 3,3,4,max=4 )
> lred <- rgb( 4,3,3,max=4 )
> matplot( A.pt, pr.fit["M",,]*100,
+          ylim=c(0,22), xlim=c(20,90), yaxs="i", xaxt="n", yaxt="n", xlab="", ylab="",
+          type="n", lty=1, col="blue", lwd=c(1,2) )
> abline( h=0:25, v=seq(0,100,5), col=gray(0.9) )
> matlines( A.pt,  pr.fit["M",,]*100, type="l", lty=1, col=lblu  , lwd=c(2,3) )
> matlines( A.pt, pr.ifit["M",,]*100, type="l", lty=1, col="blue", lwd=c(2,3) )
> text( 25, 21.5, "Men", adj=0, col="blue", cex=1.2 )
> text( 89, pr.ifit["M","89.5","1996"]* 99, "1996", col="blue", adj=c(1,1) )
```

```
> text( 80, pr.ifit["M","80.5","2016"]*101, "2016", col="blue", adj=c(0,0) )
> axis( side=1 )
> axis( side=2 )
> matplot( A.pt, pr.fit["F",,]*100,
+          ylim=c(0,22), xlim=c(20,90), yaxs="i", xaxt="n", yaxt="n", xlab="", ylab="",
+          type="n", lty=1, col="red", lwd=c(1,2) )
> abline( h=0:25, v=seq(0,100,5), col=gray(0.9) )
> matlines( A.pt,  pr.fit["F",,]*100, type="l", lty=1, col=lred , lwd=c(2,3) )
> matlines( A.pt, pr.ifit["F",,]*100, type="l", lty=1, col="red", lwd=c(2,3) )
> text( 25, 21.5, "Women", adj=0, col="red", cex=1.2 )
> text( 89, pr.ifit["F","89.5","1996"]* 99, "1996", col="red", adj=c(1,1) )
> text( 80, pr.ifit["F","80.5","2016"]*101, "2016", col="red", adj=c(1,0) )
> axis( side=1 )
> mtext( "Age", side=1, line=1, outer=T )
> mtext( "DM prevalence (%)", side=2, line=2, outer=T, las=0 )
```

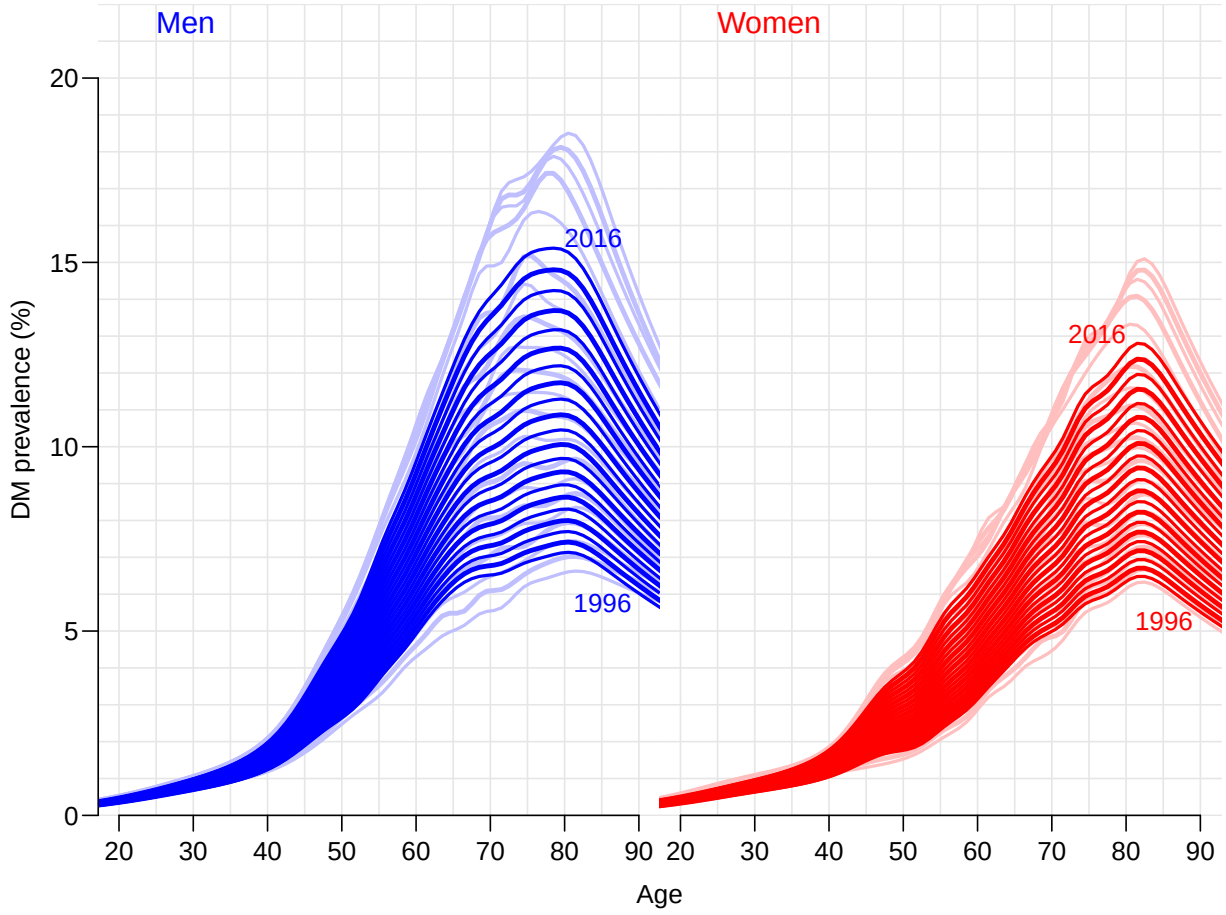

Figure 2.5: *Smoothed age-specific prevalences at 1 January 1996–2016 using a model with age-specific constant annual relative change in prevalences (full color) compared to the smoothing of the single years (pale color). Blue is men, red is women.* `./prev-ipl-cmp`

From figure 2.5 it is seen that the model does not capture the actual pattern a bit better than the simple model with an annual change common across ages — there is a substantial overshoot for men. Hence we shall base the predictions of the fitted prevalennces at 1 Januray 2016.

As a simple check we compute the estimated number of prevalent cases under the smoothed model and compare with the empirical number:

```
> tt <- xtabs( cbind(DM,N) ~ A + sex, data=subset(ptab,P==2016) )[,,c(1,2,2)]
> dimnames(tt)[[3]][1:2] <- c("obs.DM","est.DM")
> str( tt )
 table [1:100, 1:2, 1:3] 0 5 9 19 43 34 44 55 57 90 ...
 - attr(*, "dimnames")=List of 3
  ..$ A  : chr [1:100] "0" "1" "2" "3" ...
  ..$ sex: chr [1:2] "M" "F"
  ..$    : chr [1:3] "obs.DM" "est.DM" "N"
> str( pr.fit[,,"2016"] )
 num [1:2, 1:100] 0.000964 0.000921 0.001053 0.001015 0.001151 ...
 - attr(*, "dimnames")=List of 2
  ..$ sex: chr [1:2] "M" "F"
  ..$ A  : chr [1:100] "0.5" "1.5" "2.5" "3.5" ...
> tt[,,"est.DM"] <- tt[,,"N"] * t(pr.fit[,,"2016"])
> round( (tt[,,"est.DM"]/tt[,,"obs.DM"]-1)*100, 1 )
    sex
A         M      F
  0     Inf    Inf
  1   526.8 1328.6
  2   276.8   96.1
  3   104.1  155.4
  4    -0.8  139.6
  5    48.6  140.0
  6    24.9   58.8
  7    13.8   32.9
  8    17.4    7.2
  9   -17.5   -6.6
  10   15.1  -12.2
  11   -8.3  -12.0
  12  -23.2  -14.6
  13  -28.4  -24.0
  14  -25.2  -26.0
  15  -11.7  -10.3
  16  -24.3  -12.6
  17    8.4  -24.9
  18  -18.0  -13.9
  19  -10.6  -12.1
  20   -3.0  -10.5
  21   -2.5   -7.2
  22    4.2  -15.1
  23   -0.7    9.6
  24   17.6    5.5
  25   14.5   15.6
  26   16.1   21.0
  27   11.2   10.3
  28   12.6   22.7
  29   15.7   18.7
  30   12.4    9.5
  31   12.6   19.1
  32   13.8   20.9
  33    5.0    7.8
  34   12.1   17.7
  35    5.3    8.7
```

```
 36    2.4    4.0
 37    0.4    5.8
 38    2.2    3.8
 39   -2.9   10.8
 40   -1.7   -6.8
 41    1.1   -9.7
 42    3.6   -6.7
 43    8.1    2.5
 44    6.4    1.0
 45    2.2    6.3
 46    4.7    4.6
 47    4.0    6.2
 48    6.5   12.2
 49    7.6    7.1
 50    3.9    3.3
 51    2.1   -0.5
 52    6.5    2.4
 53    0.7   -2.7
 54    1.7    2.1
 55    2.6    4.9
 56    6.3    5.0
 57    5.6    5.5
 58    1.9    2.1
 59    1.4   -1.2
 60    2.3    4.0
 61    1.6    1.9
 62    4.5    2.3
 63    1.4   -0.8
 64    1.4    4.1
 65    1.8    1.2
 66    4.2    1.6
 67    0.1    3.6
 68    1.7    1.9
 69    3.7    1.2
 70    0.7    0.1
 71    1.0    1.9
 72    2.1    3.6
 73   -0.7    2.5
 74    0.2    1.1
 75   -1.2    0.9
 76    3.2    1.6
 77    2.3   -1.7
 78   -1.8    0.7
 79    0.6    0.8
 80   -1.4    5.1
 81    1.8    3.5
 82   -1.7    0.2
 83    2.9    1.5
 84   -1.2    1.8
 85    0.6   -2.1
 86   -2.3   -7.5
 87   -5.0   -5.8
 88   -7.5   -4.8
 89   -9.1   -1.6
 90   -2.7   -2.4
 91   -1.2   -2.7
 92   -1.8   -5.7
```

```
93   -4.9   -0.6
94    7.8    1.9
95   23.7   -2.5
96   17.7   17.0
97  -10.9   17.2
98   34.0    4.7
99  110.2   27.1
```

Finally we save the fitted prevalences as well as the observed numbers:

```
> save( pr.fit, pr.ini, pr.obs, file="./data/prevalences.Rda" )
```

```
> library( Epi )
> library( splines )
> library( haven )
> clear()
> set.seed( 1952 )
```

Chapter 3

Analysis and prediction of rates

3.1 Analysis data

We model the incidence and mortality 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 ($\triangle$ and $\triangle$), classified by age, period and cohort:

```
> load( './data/inits.Rda' )
> FU <- read_sas( "../data/futab.sas7bdat" )
> head( FU )
# A tibble: 6 x 9
    sex state     A     P     C     D    T1    T2           Y
  <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>       <dbl>
1     1    T1     0  1995  1995     0     0     0 0.0015633128
2     1    T1     0  1996  1995     0     0     0 0.0004757016
3     1    T1     0  1997  1996     0     0     0 0.0001704312
4     1    T1     0  1997  1997     0     0     0 0.0004613279
5     1    T1     0  1998  1997     0     0     0 0.0005174538
6     1    T1     0  1998  1998     0     0     0 0.0009267625
```

```
> with( FU, table(state) )
```

```
state
  T1   T2 Well
8499 8458 8795
```

```
> # Compute the correct A and P, make sex a factor, and compute all DM
> FU <- transform( FU, U = P-C-A,
+                      sex = factor(sex,labels=c("M","F") ),
+                       Ag = A,
+                       Pg = P,
+                       DM = T1 + T2,
+                        Y = Y * 1000 )
> FU <- transform( FU, A = Ag + (1+U)/3,
+                       P = Pg + (2-U)/3 )
> FU <- subset( FU, P>1996 )
> # Persons without diabetes
> nD <- subset( FU, state=="Well", select=c(sex,A,P,D,DM,Y) )
> names( nD )[4:6] <- c("D.nD","DM","Y.nD")
> head( nD )
```

```
     sex         A        P D.nD DM     Y.nD
8472   M 0.6666667 1996.333   28  1 17883.86
8473   M 0.3333333 1996.667    0  0 17569.79
```

```
8474   M 0.6666667 1997.333   18  1 17320.57
8475   M 0.3333333 1997.667    1  1 17725.66
8476   M 0.6666667 1998.333   18  2 17093.26
8477   M 0.3333333 1998.667    2  2 17116.36
> # Persons with diabetes
> T1 <- subset( FU, state=="T1"  , select=c(sex,A,P,D,Y) )
> T2 <- subset( FU, state=="T2"  , select=c(sex,A,P,D,Y) )
> DM <- merge( T1, T2, by=1:3, all=TRUE )
> DM <- transform( DM, D = pmax( 0, D.x, na.rm=TRUE ) +
+                          pmax( 0, D.y, na.rm=TRUE ),
+                      Y = pmax( 0, Y.x, na.rm=TRUE ) +
+                          pmax( 0, Y.y, na.rm=TRUE ) )[,c("sex","A","P","D","Y")]
> names( DM )[4:5] <- c("D.DM","Y.DM")
> TT <- merge( nD, DM, all=TRUE )
> TT[is.na(TT)] <- 0
> str( TT )
'data.frame':        8000 obs. of  8 variables:
 $ sex : Factor w/ 2 levels "M","F": 1 1 1 1 1 1 1 1 1 1 ...
 $ A   : num  0.333 0.333 0.333 0.333 0.333 ...
 $ P   : num  1997 1998 1999 2000 2001 ...
 $ D.nD: num  0 1 2 0 0 1 0 1 0 1 ...
 $ DM  : num  0 1 2 2 1 0 0 1 1 0 ...
 $ Y.nD: num  17570 17726 17116 17150 17202 ...
 $ D.DM: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Y.DM: num  0 0.461 0.927 0.309 0.244 ...
> head( TT )
  sex         A        P D.nD DM     Y.nD D.DM      Y.DM
1   M 0.3333333 1996.667    0  0 17569.79    0 0.0000000
2   M 0.3333333 1997.667    1  1 17725.66    0 0.4613279
3   M 0.3333333 1998.667    2  2 17116.36    0 0.9267625
4   M 0.3333333 1999.667    0  2 17150.26    0 0.3093771
5   M 0.3333333 2000.667    0  1 17202.29    0 0.2443532
6   M 0.3333333 2001.667    1  0 16943.84    0 0.0000000
> summary( TT )
 sex            A                P             D.nD             DM
 M:4000   Min.   : 0.3333   Min.   :1996   Min.   :  0.0   Min.   :  0.00
 F:4000   1st Qu.:25.1667   1st Qu.:2001   1st Qu.:  7.0   1st Qu.:  6.00
          Median :50.0000   Median :2006   Median : 51.0   Median : 24.00
          Mean   :50.0000   Mean   :2006   Mean   :117.7   Mean   : 43.31
          3rd Qu.:74.8333   3rd Qu.:2011   3rd Qu.:203.0   3rd Qu.: 73.00
          Max.   :99.6667   Max.   :2016   Max.   :587.0   Max.   :331.00
      Y.nD              D.DM             Y.DM
 Min.   :   22.03   Min.   :  0.00   Min.   :   0.00
 1st Qu.: 8428.40   1st Qu.:  0.00   1st Qu.:  71.51
 Median :16053.32   Median :  5.00   Median : 262.61
 Mean   :13203.66   Mean   : 20.27   Mean   : 447.89
 3rd Qu.:17895.29   3rd Qu.: 36.00   3rd Qu.: 694.13
 Max.   :23341.74   Max.   :121.00   Max.   :2816.01
```

We see from the following tabulation that we truly have data in Lexis-triangles; there are 2 observations in each, one for each sex:

```
> with( subset( TT, A<5 & P<1999 ),
+       print( table( Age=round(A,2),
+                     Per=round(P,2) ),
+              zero.print="." ) )
```

```
      Per
Age    1996.33 1996.67 1997.33 1997.67 1998.33 1998.67
  0.33       .       2       .       2       .       2
  0.67       2       .       2       .       2       .
  1.33       .       2       .       2       .       2
  1.67       2       .       2       .       2       .
  2.33       .       2       .       2       .       2
  2.67       2       .       2       .       2       .
  3.33       .       2       .       2       .       2
  3.67       2       .       2       .       2       .
  4.33       .       2       .       2       .       2
  4.67       2       .       2       .       2       .
```

A brief overview of the number of events and PY (in 1000s) by sex and calendar time:

```
> tt <- xtabs( cbind(DM,D.nD,Y.nD,D.DM,Y.DM,
+                    D.tot=D.nD+D.DM,
+                    Y.tot=Y.nD+Y.DM) ~ sex + floor(P),
+              data = TT )
> tt <- addmargins( tt, 1 )
> ff <- formatC( tt, format="f", digits=1, big.mark=",", width=12 )
>          ff[,,c("DM","D.nD","D.DM","D.tot")] <-
+ formatC( tt[,,c("DM","D.nD","D.DM","D.tot")],
+          format="f", digits=0, big.mark=",", width=8 )
>          ff[,,c("Y.nD","Y.DM","Y.tot")] <-
+ formatC( tt[,,c("Y.nD","Y.DM","Y.tot")]/1000,
+          format="f", digits=1, big.mark=",", width=8 )
> ftable( ff, row.vars=c(1:2) )
                     DM     D.nD     Y.nD     D.DM     Y.DM    D.tot    Y.tot
sex floor(P)
M   1996          6,892   26,692  2,549.8    3,163     50.8   29,855  2,600.6
    1997          6,879   26,019  2,557.5    3,258     54.3   29,277  2,611.8
    1998          7,413   25,391  2,563.2    3,342     58.2   28,733  2,621.4
    1999          7,892   25,006  2,567.8    3,645     62.2   28,651  2,630.0
    2000          7,735   24,251  2,572.8    3,620     66.4   27,871  2,639.3
    2001          7,603   24,453  2,579.3    3,707     70.5   28,160  2,649.8
    2002          8,771   24,136  2,584.8    3,972     74.8   28,108  2,659.6
    2003          9,776   23,875  2,587.6    4,141     79.9   28,016  2,667.5
    2004          9,986   23,053  2,589.1    4,134     85.6   27,187  2,674.8
    2005          9,013   22,334  2,592.1    4,215     91.0   26,549  2,683.1
    2006          8,995   22,644  2,597.9    4,346     95.5   26,990  2,693.4
    2007          9,591   22,455  2,607.2    4,436    100.2   26,891  2,707.4
    2008         10,493   22,028  2,620.3    4,493    105.9   26,521  2,726.3
    2009         11,196   21,749  2,632.3    4,959    111.9   26,708  2,744.3
    2010         12,179   21,526  2,641.2    4,982    118.7   26,508  2,759.8
    2011         15,605   20,653  2,648.0    5,133    127.4   25,786  2,775.4
    2012         12,674   20,200  2,654.3    5,382    136.5   25,582  2,790.8
    2013          9,557   20,336  2,665.1    5,501    142.1   25,837  2,807.2
    2014          9,045   19,737  2,683.0    5,706    145.7   25,443  2,828.6
    2015         10,145   20,116  2,692.2    5,738    148.6   25,854  2,840.8
F   1996          5,848   27,120  2,617.1    2,954     47.4   30,074  2,664.5
    1997          5,778   26,957  2,624.2    3,047     50.1   30,004  2,674.3
    1998          5,989   26,126  2,629.9    3,050     53.0   29,176  2,682.9
    1999          6,522   26,920  2,634.4    3,198     56.0   30,118  2,690.5
    2000          6,413   25,908  2,639.5    3,324     59.3   29,232  2,698.8
    2001          6,240   26,233  2,645.7    3,346     62.3   29,579  2,707.9
    2002          7,737   26,448  2,650.5    3,448     65.9   29,896  2,716.4
    2003          8,352   25,509  2,652.3    3,619     70.2   29,128  2,722.5
```

```
    2004         8,377   24,343  2,653.9    3,501     75.1   27,844  2,729.0
    2005         7,456   24,080  2,656.8    3,702     79.4   27,782  2,736.2
    2006         6,930   24,061  2,662.3    3,796     82.6   27,857  2,744.9
    2007         7,800   24,351  2,670.0    3,897     86.0   28,248  2,756.0
    2008         8,323   23,437  2,681.3    3,760     90.4   27,197  2,771.7
    2009         8,493   23,496  2,693.3    4,007     94.8   27,503  2,788.0
    2010         9,246   22,885  2,703.2    4,175     99.5   27,060  2,802.7
    2011        13,336   22,071  2,710.2    4,083    106.6   26,154  2,816.7
    2012        10,170   21,745  2,714.8    4,123    114.5   25,868  2,829.3
    2013         7,492   21,462  2,724.2    4,298    118.9   25,760  2,843.1
    2014         6,797   20,779  2,738.7    4,381    121.6   25,160  2,860.4
    2015         7,775   20,965  2,741.4    4,601    123.5   25,566  2,864.8
Sum 1996        12,740   53,812  5,166.9    6,117     98.2   59,929  5,265.1
    1997        12,657   52,976  5,181.7    6,305    104.4   59,281  5,286.1
    1998        13,402   51,517  5,193.1    6,392    111.2   57,909  5,304.2
    1999        14,414   51,926  5,202.2    6,843    118.2   58,769  5,320.4
    2000        14,148   50,159  5,212.4    6,944    125.7   57,103  5,338.1
    2001        13,843   50,686  5,225.0    7,053    132.8   57,739  5,357.7
    2002        16,508   50,584  5,235.3    7,420    140.7   58,004  5,376.0
    2003        18,128   49,384  5,239.9    7,760    150.2   57,144  5,390.0
    2004        18,363   47,396  5,243.0    7,635    160.7   55,031  5,403.7
    2005        16,469   46,414  5,248.9    7,917    170.4   54,331  5,419.3
    2006        15,925   46,705  5,260.2    8,142    178.1   54,847  5,438.3
    2007        17,391   46,806  5,277.2    8,333    186.3   55,139  5,463.5
    2008        18,816   45,465  5,301.7    8,253    196.3   53,718  5,498.0
    2009        19,689   45,245  5,325.6    8,966    206.7   54,211  5,532.3
    2010        21,425   44,411  5,344.3    9,157    218.2   53,568  5,562.5
    2011        28,941   42,724  5,358.2    9,216    234.0   51,940  5,592.2
    2012        22,844   41,945  5,369.1    9,505    251.0   51,450  5,620.1
    2013        17,049   41,798  5,389.3    9,799    260.9   51,597  5,650.2
    2014        15,842   40,516  5,421.7   10,087    267.3   50,603  5,689.0
    2015        17,920   41,081  5,433.6   10,339    272.0   51,420  5,705.6
```

From the last column it is plausible that we actually do have the entire Danish population follow-up in the dataset (approx. 5.5 mill. PY/Y).

For the sake of overview in the publications we devise the numbers for the four 5-year periods too, and this time nicely formatted:

```
> tt <- xtabs( cbind(DM,D.nD,Y.nD,D.DM,Y.DM,
+                    D.tot=D.nD+D.DM,
+                    Y.tot=Y.nD+Y.DM) ~ sex + Period,
+              data = transform( subset( TT, P>=1996 ),
+                                Period = factor( (P>=2001)+(P>=2006)+(P>=2011),
+                                                 labels=c("1996-2000",
+                                                          "2001-2005",
+                                                          "2006-2010",
+                                                          "2011-2015") ) ) )
> tt <- addmargins(tt,1:2)
> fCtable( tt[,,1:5], d=0, w=11, row.vars=1:2 )
                        DM        D.nD        Y.nD        D.DM        Y.DM
sex Period
M   1996-2000       36,811     127,359  12,811,103      17,028     291,939
    2001-2005       45,149     117,851  12,933,018      20,169     401,800
    2006-2010       52,454     110,402  13,098,954      23,216     532,305
    2011-2015       57,026     101,042  13,342,590      27,460     700,240
    Sum            191,440     456,654  52,185,664      87,873   1,926,284
F   1996-2000       30,550     133,031  13,145,196      15,573     265,725
```

```
    2001-2005       38,162     126,613  13,259,062      17,616     352,868
    2006-2010       40,792     118,230  13,410,047      19,635     453,299
    2011-2015       45,570     107,022  13,629,291      21,486     584,952
    Sum            155,074     484,896  53,443,596      74,310   1,656,843
Sum 1996-2000       67,361     260,390  25,956,299      32,601     557,664
    2001-2005       83,311     244,464  26,192,080      37,785     754,668
    2006-2010       93,246     228,632  26,509,001      42,851     985,604
    2011-2015      102,596     208,064  26,971,880      48,946   1,285,192
    Sum            346,514     941,550 105,629,260     162,183   3,583,127
```

Also, we devise the incidence and mortality rates per 1000 PY:

```
> rt <- tt[,,c(1,2,4,6)]/tt[,,c(3,3,5,7)]*10^3
> dimnames( rt )[[3]] <- c("inc.DM","mort.nD","mort.DM","mort.Tot")
> fCtable( rt, d=2, w=9, row.vars=1:2 )
                 inc.DM   mort.nD   mort.DM  mort.Tot
sex Period
M   1996-2000      2.87      9.94     58.33     11.02
    2001-2005      3.49      9.11     50.20     10.35
    2006-2010      4.00      8.43     43.61      9.80
    2011-2015      4.27      7.57     39.22      9.15
    Sum            3.67      8.75     45.62     10.06
F   1996-2000      2.32     10.12     58.61     11.08
    2001-2005      2.88      9.55     49.92     10.60
    2006-2010      3.04      8.82     43.32      9.94
    2011-2015      3.34      7.85     36.73      9.04
    Sum            2.90      9.07     44.85     10.15
Sum 1996-2000      2.60     10.03     58.46     11.05
    2001-2005      3.18      9.33     50.07     10.47
    2006-2010      3.52      8.62     43.48      9.87
    2011-2015      3.80      7.71     38.08      9.10
    Sum            3.28      8.91     45.26     10.11
```

3.1.1 Arrays for rate predictions

Ultimately we shall extrapolate incidence and mortality rates into the future (to 2040), so we set up arrays to hold the predicted incidence and mortality rates from the different models, separately for the two sexes; we are using midpoints of age and calendar time categories to identify rates and but boundaries of calendar time categories to identify prevalences:

```
> ht <- function(x) round(c(head(x,3),NA,tail(x,3)),3)
> ht(a.pt)
[1]  0.042  0.125  0.208     NA 99.792 99.875 99.958
> ht(p.pt)
[1] 1996.042 1996.125 1996.208       NA 2039.792 2039.875 2039.958
> ht(t.pt)
[1] 1996.000 1996.083 1996.167       NA 2039.833 2039.917 2040.000
> int
[1] 0.08333333
> fCp( c( mean( diff(a.pt) ), sd( diff(a.pt) ) ), d=5 )
[1]  0.08333  0.00000
```

```
> fCp( c( mean( diff(t.pt) ), sd( diff(t.pt) ) ), d=5 )
[1]  0.08333  0.00000
```

We the set up arrays for incidence rates of diabetes, and mortality rates of persons without, respectively with diabetes, making room for different prediction scenarios. Note that we choose the (age,date) labeling as the midpoints of the chosen intervals:

```
> Lambda <-
+ NArray( list( a = a.pt,
+                p = p.pt,
+              sex = c("M","F"),
+              mod = c("ap","apc","LCa","att","fix","p20","p40","p60") ) )
> str( Lambda )
 logi [1:1200, 1:528, 1:2, 1:8] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.291666666666667
  ..$ p  : chr [1:528] "1996.04166666667" "1996.125" "1996.20833333333" "1996.29166666667"
  ..$ sex: chr [1:2] "M" "F"
  ..$ mod: chr [1:8] "ap" "apc" "LCa" "att" ...
> Mu.nD <- Mu.DM <- Lambda[,,,1:5]
> str( Mu.nD )
 logi [1:1200, 1:528, 1:2, 1:5] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.291666666666667
  ..$ p  : chr [1:528] "1996.04166666667" "1996.125" "1996.20833333333" "1996.29166666667"
  ..$ sex: chr [1:2] "M" "F"
  ..$ mod: chr [1:5] "ap" "apc" "LCa" "att" ...
> fCp( c(length( Lambda ),length( Mu.nD ) ), w=12 )
[1]   10,137,600    6,336,000
```

Note that on dimension 4, we have levels to hold the naïve prediction using the natural splines for the AP, APC and Lee-Carter models, as well for modification by trend attenuation (see below for explanation), `att`. Further, for the incidence rates we have levels for the models with attenuation *plus* an added time-trend, `p20` etc. increasing 2.0, 4.0 and 6.0% per year, respectively.

3.1.2 Data sets for rate modeling

Modeling of rates requires that the person-years are strictly positive (if there are no person-years, there are no observations to model), but there are not person-years for diabetes patients in all Lexis triangles:

```
> fCp( table( 'Y.DM=0'=TT$Y.DM==0,'Y.nD=0'=TT$Y.nD==0, useNA="ifany" ) )
       Y.nD=0
Y.DM=0  FALSE
  FALSE    7,977
  TRUE        23
```

In order to save hassle in the modeling code we construct simple data sets of identical structure for APC and Lee-Carter analysis separately for the three types of transitions we are going to consider. This is mainly a vehicle for simpler code when modeling the three sets of rates.

```
> incdat <- TT[,c("sex","A","P",  "DM","Y.nD")]
> mnDdat <- TT[,c("sex","A","P","D.nD","Y.nD")]
> mDMdat <- TT[,c("sex","A","P","D.DM","Y.DM")]
> names( incdat )[4:5] <-
+ names( mnDdat )[4:5] <-
+ names( mDMdat )[4:5] <- c("D","Y")
> mDMdat <- subset( mDMdat, Y>0 )
> save( incdat, mnDdat, mDMdat, file="./data/rate-dat.Rda" )
> load(                         file="./data/rate-dat.Rda" )
```

3.2 Models for incidence and mortality rates

There are two immediate alternatives to the age-period-cohort (APC) models on the log-scale; age-period-cohort models on the additive scale and Lee-Carter models. The latter may provide a better fit to data (some of the expanded Lee-Carter models necessarily do).

In principle we could explore the entire set of models for men and women separately using the `apc.LCa` function:

```
> system.time(
+ minc <- apc.LCa( subset(incdat,sex=="M"), eps=1e-4 ) )
> finc <- apc.LCa( subset(incdat,sex=="F"), eps=1e-4 )
> mmnD <- apc.LCa( subset(mnDdat,sex=="M"), eps=1e-4 )
> fmnD <- apc.LCa( subset(mnDdat,sex=="F"), eps=1e-4 )
> mmDM <- apc.LCa( subset(mDMdat,sex=="M"), eps=1e-4 )
> fmDM <- apc.LCa( subset(mDMdat,sex=="F"), eps=1e-4 )
> save( incdat, mnDdat, mDMdat,
+       minc, finc, mmnD, fmnD, mmDM, fmDM,
+       file = "./data/imdat.Rda" )
```

However, for now we shall proceed with AP and APC-models for the incidence and mortality rates.

3.3 Incidence rates

Based on this we can now devise the location of the knots for the age, period and cohort terms in the model — recall the function `qn` was defined in the prerequisites (p. 2):

```
> ( ki.a <- with( incdat, quantile( spread(rep(  A,D)), qn(nk.a) ) ) )
   6.25%   18.75%   31.25%   43.75%   56.25%   68.75%   81.25%   93.75%
35.13796 47.85155 54.51978 59.74601 64.37105 69.11418 74.73721 82.77272
> ( ki.p <- with( incdat, quantile( spread(rep(P  ,D)), qn(nk.p,5) ) ) )
3.333333%       22% 40.66667% 59.33333%       78% 96.66667%
 1996.912  2001.669  2005.433  2009.156  2011.917  2015.370
> ( ki.c <- with( incdat, quantile( spread(rep(P-A,D)), qn(nk.c) ) ) )
8.333333%       25% 41.66667% 58.33333%       75% 91.66667%
 1923.786  1934.629  1942.167  1948.102  1956.191  1969.530
```

The models we set up is are age-period and age-period-cohort models, separately for the two sexes. 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. However we want to extract the average trend from the APC-model, so we also fit the model with the parametrization that allows us to extract the trend (and we also check that it actually *is* the same model). We also fit an *ad hoc* model with a separate slope pre- and post 2012 (linear spline) in order to assess changes in incidence trends:

```
> m.inc.ap <- glm( D ~ Ns(A,kn=ki.a) + Ns(P,kn=ki.p),
+                      offset = log( Y ),
+                      family = poisson,
+                        data = subset(incdat,sex=="M") )
> # drift before and after 2012:
> m.inc.a2P <- update( m.inc.ap, . ~ . - Ns(P,kn=ki.p) + pmin(0,P-2012) + pmax(0,P-2012) )
> m.inc.aPC <- update( m.inc.ap, . ~ . - Ns(P  ,kn=ki.p) + I(P) +
+                                detrend( Ns(P  ,kn=ki.p), P  , D ) +
+                                detrend( Ns(P-A,kn=ki.c), P-A, D ) )
> m.inc.apc <- update( m.inc.ap, . ~ . + Ns(P-A,kn=ki.c) )
> c( m.inc.apc$deviance, m.inc.aPC$deviance )
[1] 10006.29 10006.29
> f.inc.ap  <- update( m.inc.ap , data = subset(incdat,sex=="F") )
> f.inc.a2P <- update( m.inc.a2P, data = subset(incdat,sex=="F") )
> f.inc.apc <- update( m.inc.apc, data = subset(incdat,sex=="F") )
> f.inc.aPC <- update( m.inc.aPC, data = subset(incdat,sex=="F") )
```

The average annual trends in incidence from the multiplicative models:

```
> inc.chg <- rbind( ci.exp(m.inc.aPC,subset="I\\(P")-1,
+                   ci.exp(f.inc.aPC,subset="I\\(P")-1,
+                   ci.exp(m.inc.a2P,subset="pm")-1,
+                   ci.exp(f.inc.a2P,subset="pm")-1 )[c(1:3,5,4,6),]*100
> rownames( inc.chg ) <- c("DM incidence change     Men",
+                          "                      Women",
+                          "<2012                   Men",
+                          "                      Women",
+                          ">2012                   Men",
+                          "                      Women")
> round( inc.chg, 1 )
                           exp(Est.)  2.5% 97.5%
DM incidence change     Men       2.0   1.9   2.1
                      Women       2.2   2.1   2.3
<2012                   Men       3.6   3.5   3.7
                      Women       3.5   3.4   3.7
>2012                   Men     -11.1 -11.6 -10.6
                      Women     -12.3 -12.9 -11.8
```

The average increase is similar in women and men, but the period effect is massively non-linear, so these summary estimates are not really informative, see top panels in figure 3.4, so we included average trends before and after 2012, by fitting a linear spline model with a knot at 2012-01-01, yielding the average annual increase of 3.9% before 2012 and a decrease of some 11% after 2012.

Hence in the projections we shall make predictions with future trends in incidence rates of 2, 4 and 6%.

Extended Lee-Carter models

We also fit Lee-Carter models as an extension of the APC-models:

```
> system.time(
+ m.inc.LCa <- LCa.fit( data = subset(incdat,sex=="M"),
+                        model = "APaCa",
+                        a.ref = 65,
+                        p.ref = 2010,
+                        c.ref = 1945,
+                         npar = list( a=nk.a,
+                                      p=nk.p,
+                                      c=nk.p,
+                                     pi=5,
+                                     ci=5 ),
+                          eps = 1e-4,
+                           VC = FALSE,
+                        quiet = FALSE ) )
> f.inc.LCa <- LCa.fit( data = subset(incdat,sex=="F"),
+                        model = "APaCa",
+                        a.ref = 65,
+                        p.ref = 2010,
+                        c.ref = 1945,
+                         npar = list( a=nk.a,
+                                      p=nk.p,
+                                      c=nk.p,
+                                     pi=5,
+                                     ci=5 ),
+                          eps = 1e-4,
+                           VC = TRUE,
+                        quiet = FALSE )
> save( m.inc.LCa,
+       f.inc.LCa, file="../data/incLCa.Rda" )


> load( file="../data/incLCa.Rda" )
```

3.3.1 Incidence rate predictions

We want the predicted incidence rates at a grid of points suitable for the calculations of predicted prevalences — essentially for the construction of transition probabilities. We make the predictions for all combinations of `a.pt` and `p.pt`.

All the predictions should be in units of the interval length chosen for calculations, or more precisely we predict cumulative rates over small intervals. Thus we use a prediction data frame with the person-years-variables set to `int`, the length of the step interval we shall use for projections.

Note that the `Lambda` array was set up with age before period, so that the column-major storage of arrays conforms with the predictions obtained using the following `nd` as `newdata`. Note that we are computing the rates at the boundaries of the age-classes (hence the term `+int/2`) but at the midpoints of the periods:

```
> ndn <- function(obj,dno) as.numeric(dimnames(obj)[[dno]])
> nd <- data.frame( expand.grid( A = ndn(Lambda,1),
+                                P = ndn(Lambda,2) ),
+                   Y = int )
> str( nd )
'data.frame':        633600 obs. of  3 variables:
 $ A: num  0.0417 0.125 0.2083 0.2917 0.375 ...
```

```
 $ P: num  1996 1996 1996 1996 1996 ...
 $ Y: num  0.0833 0.0833 0.0833 0.0833 0.0833 ...
```

```
> head( nd )
```

```
           A        P          Y
1 0.04166667 1996.042 0.08333333
2 0.12500000 1996.042 0.08333333
3 0.20833333 1996.042 0.08333333
4 0.29166667 1996.042 0.08333333
5 0.37500000 1996.042 0.08333333
6 0.45833333 1996.042 0.08333333
```

```
> fCp( c( prod( dim(Lambda)[1:2] ), length(Lambda),
+         nrow(nd), nrow(nd)*prod( dim(Lambda)[-(1:2)] ) ) )
```

```
[1]  633,600   10,137,600  633,600   10,137,600
```

```
> 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 )
> # Lambda[,,"M","LCa"] <- predict.LCa( m.inc.LCa, newdata=nd )[,1]
> # Lambda[,,"F","LCa"] <- predict.LCa( f.inc.LCa, newdata=nd )[,1]
```

Thus we have the incidence rates that we need for two sexes, and for two different modeling approaches. Note that we have computed the rates in units of `int`=0.08333.

Now, there is a clear downward trend in the rates at the end; we can show the trend in different ages by extracting the last few dates from the `Lambda` array:

```
> ( ell <- dimnames(Lambda)[[2]][c(220,240)] )
```

```
[1] "2014.29166666667" "2015.95833333333"
```

```
> ( ell <- diff( as.numeric( ell ) ) )
```

```
[1] 1.666667
```

```
> tr.ap  <- (exp(log(Lambda[,240,,"ap" ]/Lambda[,220,,"ap" ])/ell)-1)*100
> tr.apc <- (exp(log(Lambda[,240,,"apc"]/Lambda[,220,,"apc"])/ell)-1)*100
> matplot( as.numeric(dimnames(Lambda)[[1]]), cbind( tr.ap, tr.apc ), las=1,
+          type="l", lty=rep(c("21","solid"),each=2), lend=1, lwd=3, col=c("blue","red"),
+          ylim=range(c(0,tr.ap,tr.apc)), xlab="Age", ylab="Annual change in rates (%)" )
> abline( h=0, col="gray" )
```

3.3.2 A damping extrapolation

Instead of using the naïve extrapolation as above where we blindly just prolong the linear trajectories from the natural splines (for period and cohort), we may dampen the trend derived from the naïve application of the natural splines. The attenuation factor (explained below) is traditionally (*i.e.* in cancer epidemiology [3, 4]) set to 0.92 per year (corresponding to a halving of the drift approximately every 8.3 years).

In the subsequent implementation we shall parametrize the damping factor by the half-time of the slope, thus instead of d use h where: $d^h = 0.5 \Leftrightarrow d = 0.5^{1/h}$.

In order to do this we will need to *have* a drift to halve. To this end we may choose a parametrization of the form:

$$\log(\lambda(a,p)) = f(a) + g(p) + h(p-a) + \beta p$$

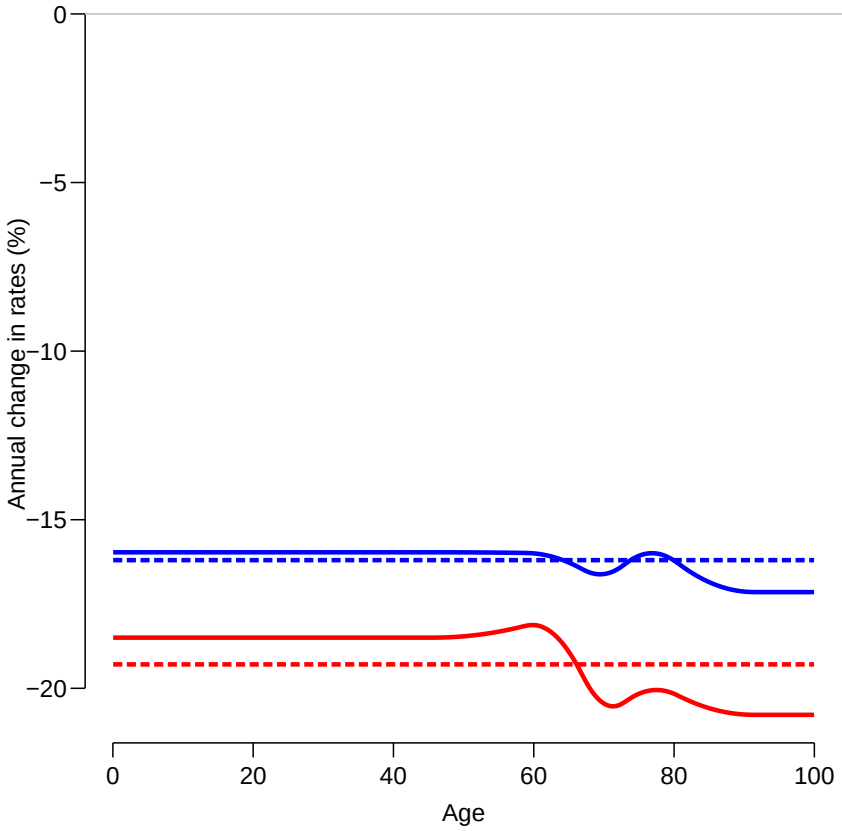

Figure 3.1: *Estimated trend in diabetes incidence rates by age over the last 3 months of 2015, red curves are women, blue men, the dotted curves are from the AP-model, the full lines from the APC model.* `./rates-last-trend`

where g and h are chosen to be flat on average. This is generally in the literature thought to be a unique operation, however it depends on the choice of an inner product in the space of the linear predictor [1][1].

Now suppose we have a parametrization using natural splines [4] with extracted drift for some choice of inner product (that is the model matrix columns defining $g(p)$ are orthogonal to the intercept and p, and similarly for $h(p-a)$[1]). A natural extrapolation beyond the last observation date p_x, say would be to use for $p > p_x$:

$$\log(\lambda(a,p)) = f(a) + g(p_x) + h(p_x - a) + \beta p_x + \beta d^{p-p_x}(p - p_x)$$

where d is a damping factor, such that the drift at date p after p_x is βd^{p-p_x}.

3.3.3 Theory

The following is an explanation of the potential problem with the approach of Sasieni (as I read it), but it is only an explanation of why a different approach is needed in order to obtain a damping of future drifts from APC-models that are invariant under reparametrizations of the APC-model.

[1]Peter Sasieni [5] refer to this model but does not explain how the drift (β) was extracted, that is the parametrization is not unique, though apparently believed so (?).

Exponential damping of a drift

Arithmetically, the attenuation or "damping" machinery works as follows in *continuous* time:

If we model age, period and cohort terms by natural splines, *i.e.* using splines that are linear beyond the outer knots, then these easily lend themselves to prediction beyond data, simply by extending the linear part of the natural spline, see *e.g.* Rutherford *et al.*[4].

Now suppose the slope of the period or cohort effect at the end is β, but that it would be an exaggeration to continue the period effect indefinitely at a slope of β (a prediction of the form $f(t) = k + \beta t$). Therefore we choose a *damping* factor, d, say, such that the *slope* of the effect at time t is not β, but rather βd^t. If $d < 1$ it means that the slope will gradually approach 0 as $t \to \infty$, the faster the smaller d is, so as mentioned above $d = 0.92$ implies that $d^{8.3} = 0.5$ and $d = 0.88$ implies that $d^{5.4} = 0.5$. Subsequently we shall parametrize by the halving time instead of d.

For the damped effect itself, $f(t)$, we have:

$$f'(t) = \beta d^t \quad \Leftrightarrow \quad f(t) = k + \beta d^t / \log(d) \quad \Rightarrow \quad f(0) = k + \beta / \log(d)$$

Solving for k, we get:

$$k = f(0) - \beta / \log(d) \quad \Rightarrow \quad f(t) = f(0) - \frac{\beta}{\log(d)} + \frac{\beta d^t}{\log(d)} = f(0) + \beta \frac{d^t - 1}{\log(d)}$$

where $f(0)$ is the value of the period or cohort term at the beginning of the prediction interval.

Note that the term $(d^t - 1)/\log(d)$ is always non-negative for $0 < d < 1$.

Implementation in APC-models

Suppose the fitted APC model is parametrized as:

$$\log(\lambda(a, p)) = f(a) + g(p) + h(p - a) + \beta p$$

where the last term is brought about by some sort of projection [1], effectively leaving $g(p)$ and $h(p - a)$ as "de-trended", that is with overall average slope 0 in some (in this context, unspecified) sense.

If we assume that g and h are parametrized by natural splines, then $g(p)$ is linear beyond the last knot which we shall assume to be smaller than the end of data (well, really the starting point of the prediction period), p_x, say. Now, any reparametrization involving the period / cohort slopes will give a model of the form (for an arbitrary δ):

$$\begin{aligned} \eta_0(a, p) = \log\big(\lambda(a, p)\big) &= \big(f(a) - \delta a\big) + \big(g(p) + \delta p\big) + \big(h(p - a) - \delta c\big) + \beta p && (3.1) \\ &= \big(f(a) - \delta a\big) + g(p) + \big(h(p - a) - \delta c\big) + (\beta + \delta) p && (3.2) \end{aligned}$$

If a prediction of future rates is going to depend non-linearly on the time since the prediction point, for example by specifying the slope to be the extracted slope multiplied by $d^{p - p_x}$ for some damping factor $0 < d < 1$, the prediction will depend in the parametrization chosen in the following sense:

If we use the parametrization 3.1 to construct predicted rates, the damped version of the log-rates would be (using the notation $(x)_+ = \max(x, 0)$):

$$\eta_1(a,p) = \eta_0(a,p) - \beta(p-p_x)_+ + \frac{\beta(d^{(p-p_x)_+}-1)}{\log(d)}$$

whereas if we use the parametrization 3.2 they would be:

$$\begin{aligned}\eta_2(a,p) &= \eta_0(a,p) - (\beta+\delta)(p-p_x)_+ + (\beta+\delta)\frac{d^{(p-p_x)_+}-1}{\log(d)} \\ &= \eta_1(a,p) - \delta(p-p_x)_+ + \delta\frac{d^{(p-p_x)_+}-1}{\log(d)}\end{aligned}$$

So the prediction with a damping factor (or for that matter any other replacement of βp with a non-linear function of p) will not be invariant under reparametrization of the original model.

Resorting to the postulated uniqueness of the drift as claimed by e.g. Holford [2], will not solve the problem, it will just choose a particular way of extracting the drift [1]. Other ways of extracting the drift will give different results. It all boils down to replacing the predictions based on extrapolation of the period and cohort term by the linear parts of the natural splines by adding the term

$$\beta\big(-(p-p_x)_+ + (d^{p-p_x}-1)/\log(d)\big)$$

for some chosen β.

The good news is that different reasonable ways of extracting the trend do not normally give dramatically different values for the drift β.

The bad news is that the extracted slope may not be anywhere near any slope of rates at the beginning of the prediction interval.

Thus it seems that the reliance on the particular form of parametrization is not a viable path, so here is a more directly data-driven approach, that for a given APC-model is parametrization invariant.

3.3.4 A discrete time generalization

The following is an empirical approach to adjust rates predicted into the future. We use a damping mechanism, taking an approach that does not rely on any particular mathematical form of the predictions, but merely on the predictions being available in suitably small intervals.

Suppose you have prediction of future rates (or log-rates) $\lambda(a,p)$ from an APC-model (well, this goes for any model) — values for occurrence rates in the period-direction.

A slope-attenuation can be numerically implemented by using the empirical gradients of the predictions, so suppose that for a fixed value of a the rates are in the vector `f` and the corresponding dates (p) in the vector `t`. The empirical slopes between successive time points is then simply `diff(f)/diff(t)`, and we could then attenuate these directly by multiplying them by d^t where d is the chosen damping factor and t is the midpoint of the interval. Here is a simple illustration of how it goes:

```
> t <- 0:100/20
> f <- 0.2*t + sin(t) + 0.2*(t)^2
> f <- f/diff(range(f))
```

Then we need the midpoints of the observation intervals:

```
> dd <- 0.8
> dt <- diff(t)
> mt <- t[-1] - dt/2
>  df <- diff(f) / diff(t)
> ddf <- df * dd^mt
> iof <- c( f[1], f[1] + cumsum( df)*dt )
> idf <- c( f[1], f[1] + cumsum(ddf)*dt )
```

Now this is easily implemented in a function which takes the function values `f`, times `t` and damping factor as arguments. What remains to be seen is that this method will produce results that are (at least approximately) invariant under choice of step-size; so we take every 4^{th} and see how it goes:

```
> t <- t[0:25*4+1]
> f <- f[0:25*4+1]
> dt <- diff(t)
> mt <- t[-1] - dt/2
>  df <- diff(f) / diff(t)
> ddf <- df * dd^mt
> iof <- c( f[1], f[1] + cumsum( df)*dt )
> idf <- c( f[1], f[1] + cumsum(ddf)*dt )
```

We can then plot the results from the two approaches together:

```
> # intervals of 0.2
> t <- 0:100/20
> f <- 0.2*t + sin(t) + 0.2*(t)^2
> f <- f/diff(range(f))
> plot( f ~ t, type="l", lwd=8 )
> dd <- 0.8
> dt <- diff(t)
> mt <- t[-1] - dt/2
>  df <- diff(f) / diff(t)
> ddf <- df * dd^mt
> iof <- c( f[1], f[1] + cumsum( df)*dt )
> idf <- c( f[1], f[1] + cumsum(ddf)*dt )
> matlines( t, cbind( iof, idf),
+           type="l", lwd=4, lty=1, col=c("red","forestgreen") )
> # intervals of 1
> t <- t[0:5*20+1]
> f <- f[0:5*20+1]
> dt <- diff(t)
> mt <- t[-1] - dt/2
>  df <- diff(f) / diff(t)
> ddf <- df * dd^mt
> iof <- c( f[1], f[1] + cumsum( df)*dt )
> idf <- c( f[1], f[1] + cumsum(ddf)*dt )
> matlines( t, cbind( iof, idf),
+           type="l", lwd=4, lty=1, col=gray(0.7) )
```

Indeed the resulting curves coincide at the points where the coarser is defined, although a mathematical calculation would show that they are not exactly the same.

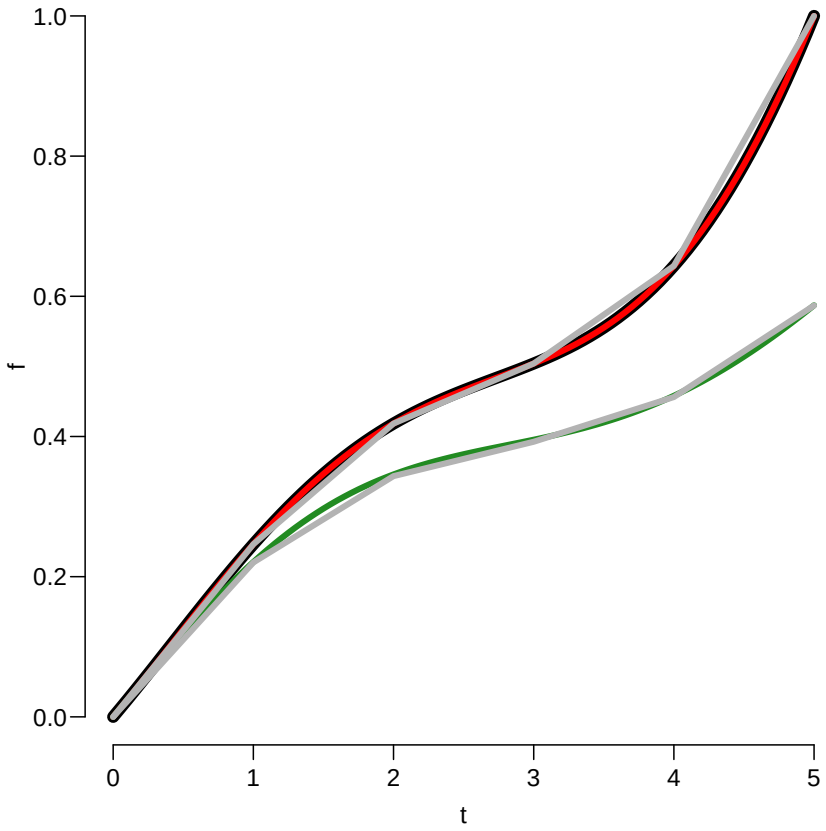

Figure 3.2: *An "arbitrary" curve and the attenuated version, using an attenuation factor of 0.8 (per t unit). The curves are constructed using steps of length 0.2 or 1 (the latter overlaid in gray).* `./rates-att-example`

Adding a drift

For the diabetes incidence we have observed that the incidence rates show a dramatically decreasing tendency over the last few years of observation ($\approx 11\%$/year), hence we may want not only to investigate a scenario where rates are kept or attenuated to constant, but also one where we simply let the rates increase by a certain amount, say 4% per year. This is only going to be used for the incidence rates as a sensitivity analysis.

To this end we update the damping function briefly outlined above by allowing adding a trend (drift) in calendar time on top of the attenuated prediction; we phase it in quadratically over a period of ℓ, by the function q, a parabola with slope 0 at 0 and slope δ at *ell*, and a linear function with slope δ beyond ℓ, that is:

$$q(t) = \begin{cases} 0 < t < \ell & : \quad \big(\delta/(2\ell)\big)t^2 \\ \ell < t < \infty & : \quad -\delta\ell/2 + \delta t \end{cases}$$

or in R-code:

```
> qs <-
+ function( t, ell, delta ) ifelse( t < ell, delta / ell / 2 * t^2,
+                                             delta * t - delta * ell / 2 )
```

which is incorporated in a damping function below.

Implementation

We implement this attenuation and slope addition in a function `damp` which takes 6 arguments:

`f` — a vector of predicted function values (rates or log-rates) to be modified by damping and/or addition of a trend

`t` — an ordered vector of time points where `f` is given. Need not be equidistant. Note that `t-t[1]` is used as exponent to the damping factor, so results are invariant under translation of `t`.

`h` — a scalar, the halving time for the slope. In the function it is converted to a damping factor which will be elevated to the power of `t`, thus dependent on the scaling of `t`: For halving time h we have $d^h = 0.5 \Leftrightarrow d = 0.5^{1/h}$.

`delta` — scalar; the extra slope added to the predictions, works beyond `ell` (`t`$\geq$`ell`), before that the addition is a smooth quadratic fitting with the linear at `ell`. This is an additive factor, so a 10% increase per unit of `t` is obtained by `delta=0.1`.

`ell` — scalar; the run-in interval (on the `t`-scale) for the extra slope.

`logf` — logical indicating whether the supplied `f` represent log-rates or rates. In any case the attenuation is made on the log-rate scale.

With this, a value of 0 for `h` produces an immediately flat (constant) modified curve, corresponding to a fixing of rates at `t` $= 0$. Likewise a choice of 0 for the interval length `ell` corresponds to an immediate start of an added slope of `delta`:

```
> damp <-
+ function( f, t, h, delta = 0,      # added slope (% per t unit),
+                       ell = 0,      # phase-in interval for added slope
+                      logf = FALSE ) # is f a vector of log-rates
+ {
+ # all operations are on log-rates so if we have rates make them log
+ if( !logf ) f <- log( f )
+ # compute the damping factor from half-time
+ d <- 0.5^(1/h)
+ # make sure t start at 0
+ t <- t - t[1]
+ # difference between timepoints of prediction
+ dt <- diff(t)
+ # midpoints of intervals
+ mt <- t[-1] - dt/2
+ # slopes in each interval
+ dfdt <- diff(f) / dt
+ # attenuated slopes
+ atdf <- dfdt * d^mt
+ # function values after attenuating the slope
+ idf <- f[1] + cumsum(c(0,atdf))*dt
+ # remember delta is taken as being in % per t
+ delta <- delta/100
+ # add the extra slope to this
+ idf <- idf + ifelse( t < ell, delta/(2*ell)*t^2,
+                              delta*(t-ell/2) )
```

```
+ if( !logf ) idf <- exp( idf )
+ idf
+ }
```

We can illustrate the damping effect in different ways, firstly, the time it takes to reduce the slope to say, 50, 10 and 1% (ζ, say) of the original one, is illustrated by simply solving:

$$d^t = \zeta \quad \Leftrightarrow \quad t\log(d) = \log(\zeta) \quad \Leftrightarrow \quad t = \log(\zeta)/\log(d)$$

This is the left panel in figure reffig:damp-ex; the other one illustrates the resulting damped / amended curves relative to an arbitrary constant slope:

```
> par( mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6, bty="n", las=1 )
> clr <- rainbow(3)
> d <- seq(0,1,,200)
> zeta <- c(0.5,0.1,0.01)
> matplot( d, outer( d, zeta, function(d,zeta) log(zeta)/log(d) ),
+          type="l", lwd=4, lty=1, col=clr,
+          ylim=c(0,25), xlab="Damping factor",
+          ylab=paste( "Time to reduction to ",
+                      paste( round(zeta*100,1), collapse=", "),
+                      "%, respectively", sep="" ) )
> abline( v=c(0.92, 0.88, 0.7) )
> abline( h=0:10, lty=2, col=gray(0.8) )
>  axis( at=c(0.92, 0.88, 0.7), las=2, side=1 )
> clr <- c("black",rainbow(7))
> tt <- seq( 0,25,0.1)
> ff <- 12.5 - 0.4 * tt
> t0 <- 8
> t  <- (tt-t0)[tt>=t0]
> f  <-      ff[tt>=t0]
> plot( tt, ff, lty=1, lwd=5, type="l", ylim=c(2,12),
+               xlab="Time", ylab="Damped effect")
> matlines( t+t0, cbind( f, damp(f,t,h=5),
+                           damp(f,t,h=Inf),
+                           damp(f,t,h=10,delta=5,ell=5),
+                           damp(f,t,h=2  ,delta=5,ell=5),
+                           damp(f,t,h=2),
+                           damp(f,t,h=2  ,delta=-5,ell=5) ),
+           lty=1, lwd=c(5,rep(3,6)), type="l", col=clr,
+           xlab="Time", ylab="Damped effect")
> text( 5, 5.5-0:6/2, c( "Half-time",
+                        formatC( c(5,Inf,10,2,2,2), format="f", digits=2 ) ),
+                     font=2, col=clr, adj=1 )
> text( 7, 5.5-0:6/2, c( "Added slope / yr",
+                        formatC( c(0,0,1/20,abs(1:-1)/20), format="f", digits=2 ) ),
+                     font=2, col=clr, adj=0 )
> text( 6.9, 5.5-6/2, "-", font=2, col=clr[7], adj=1 )
> segments( c(t0,t0+5), 6,
+           c(t0,t0+5), 12 )
```

So for the incidence array `Lambda` we would do something like:

```
> str( Lambda )
 num [1:1200, 1:528, 1:2, 1:8] 6.19e-06 6.23e-06 6.27e-06 6.31e-06 6.35e-06 ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.291666666666667
```

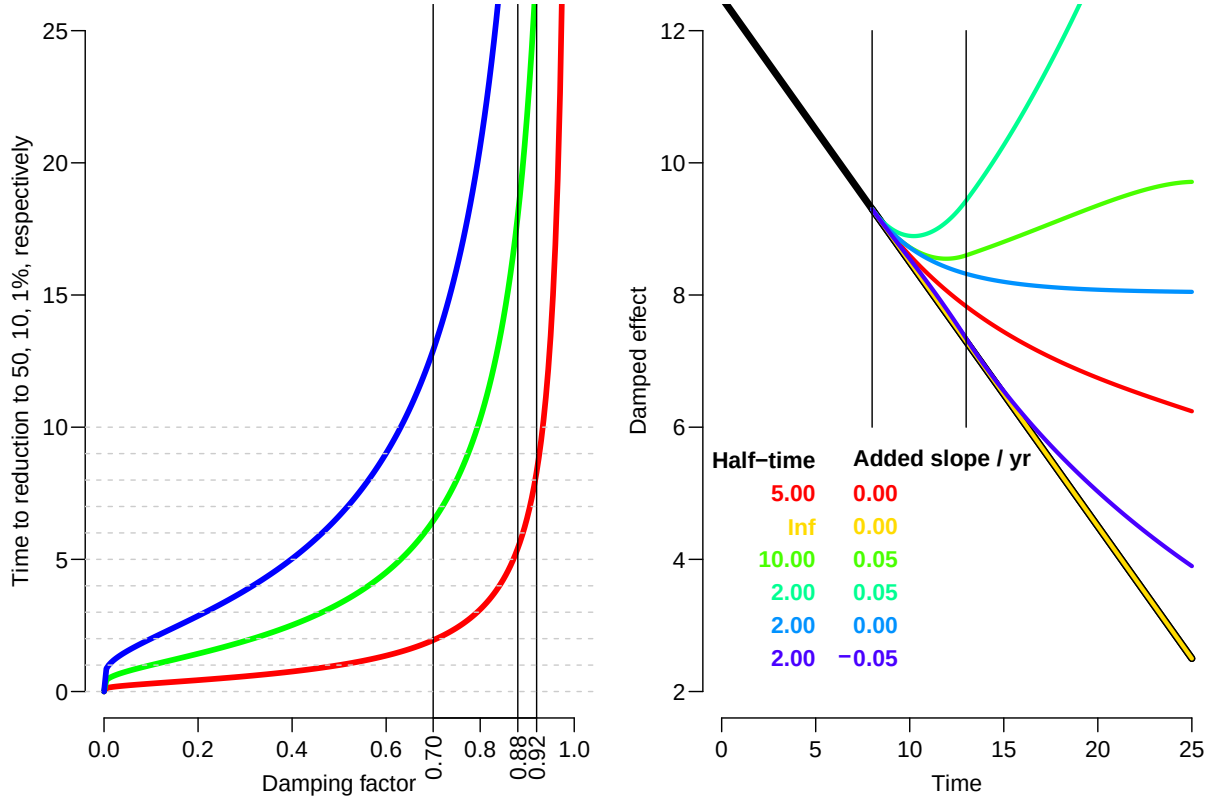

Figure 3.3: *Illustration of the* `damp` *function for attenuation of effects and addition of linear terms for various combinations of the two. The two vertical black lines indicate the starting point of the attenuation and the end of the phase-in of the added slope.* `./rates-damp-ex`

```
  ..$ p  : chr [1:528] "1996.04166666667" "1996.125" "1996.20833333333" "1996.29166666667"
  ..$ sex: chr [1:2] "M" "F"
  ..$ mod: chr [1:8] "ap" "apc" "LCa" "att" ...

> # where is the future
> wh.p <- 241:408
> fp.pt <- p.pt[wh.p]
> fp.pt[1:3]

[1] 2016.042 2016.125 2016.208

> dimnames(Lambda)[[4]]

[1] "ap"  "apc" "LCa" "att" "fix" "p20" "p40" "p60"

> for( ia in dimnames(Lambda)[['a']] )
+ for( is in dimnames(Lambda)[['sex']] )
+    {
+ # Compute the damped values along the period dimension
+           Lambda[ia,wh.p,is,"att"] <-
+ damp( f = Lambda[ia,wh.p,is,"apc"],
+       t = fp.pt,
+       h = 2,
+   delta = 0,
+     ell = 0 )
+     }
```

The outer loop over age could be replaced by a simple `apply`; further we will also make predictions with added slopes of 3, 4 and 5% per year, but first we need models for the mortality rates.

3.4 Mortality rates

3.4.1 Diabetes patients

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

```
> ( kmd.a <- with( mDMdat, quantile( spread(rep(A  ,D)), qn(nk.a) ) ) )
   6.25%   18.75%   31.25%   43.75%   56.25%   68.75%   81.25%   93.75%
56.71605 66.39891 71.95143 76.24136 79.90802 83.43170 87.09844 91.97257
> ( kmd.p <- with( mDMdat, quantile( spread(rep(P  ,D)), qn(nk.p) ) ) )
8.333333%       25% 41.66667% 58.33333%       75% 91.66667%
 1998.182  2002.126  2005.651  2008.941  2011.917  2014.691
> ( kmd.c <- with( mDMdat, quantile( spread(rep(P-A,D)), qn(nk.c) ) ) )
8.333333%       25% 41.66667% 58.33333%       75% 91.66667%
 1913.796  1921.047  1926.330  1931.875  1938.698  1948.921
> m.md.ap <- glm( D ~ Ns(A,kn=kmd.a) + Ns(P,kn=kmd.p),
+                         offset = log(Y),
+                         family = poisson,
+                           data = subset( mDMdat, sex=="M" ) )
> m.md.aPC <- update( m.md.ap, . ~ . - Ns(P  ,kn=kmd.p) + I(P) +
+                             detrend( Ns(P  ,kn=kmd.p), P  , D ) +
+                             detrend( Ns(P-A,kn=kmd.c), P-A, D ) )
> m.md.apc <- update( m.md.ap, . ~ . + Ns(P-A,kn=kmd.c) )
> f.md.ap  <- update( m.md.ap , data = subset( mDMdat, sex=="F" ) )
> f.md.apc <- update( m.md.apc, data = subset( mDMdat, sex=="F" ) )
> f.md.aPC <- update( m.md.aPC, data = subset( mDMdat, sex=="F" ) )
> 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 )
```

3.4.2 Persons without diabetes

The mortality in the population without diabetes is modeled in exactly the same way as the incidence trends, except we also put in knots in early age too:

```
> ( kmw.a <- with( mnDdat, c( 5, 15,
+                         quantile( spread(rep(A  ,D)), qn(nk.a) ) ) ) )
                    6.25%   18.75%   31.25%   43.75%   56.25%   68.75%   81.25%   93.75%
 5.00000 15.00000 50.79327 64.19092 71.63862 77.00912 81.26080 85.02942 88.82574 93.69716
> ( kmw.p <- with( mnDdat, quantile( spread(rep(P  ,D)), qn(nk.p) ) ) )
8.333333%       25% 41.66667% 58.33333%       75% 91.66667%
 1997.467  2000.505  2003.623  2006.951  2010.388  2014.078
> ( kmw.c <- with( mnDdat, quantile( spread(rep(P-A,D)), qn(nk.c) ) ) )
```

```
8.333333%       25% 41.66667% 58.33333%       75% 91.66667%
 1911.125  1918.378  1923.752  1929.660  1938.113  1952.393

> m.mw.ap <- glm( D ~ Ns(A,kn=kmw.a) + Ns(P,kn=kmw.p),
+                     offset = log(Y),
+                     family = poisson,
+                       data = subset( mnDdat, sex=="M" ) )
> m.mw.aPC <- update( m.mw.ap, . ~ . - Ns(P  ,kn=kmw.p) + I(P) +
+                               detrend( Ns(P  ,kn=kmw.p), P  , D ) +
+                               detrend( Ns(P-A,kn=kmw.c), P-A, D ) )
> m.mw.apc <- update( m.mw.ap, . ~ . + Ns(P-A,kn=kmw.c) )
> f.mw.ap  <- update( m.mw.ap , data = subset( mnDdat, sex=="F" ) )
> f.mw.apc <- update( m.mw.apc, data = subset( mnDdat, sex=="F" ) )
> f.mw.aPC <- update( m.mw.aPC, data = subset( mnDdat, sex=="F" ) )
> Mu.nD[,,"M","ap" ] <- predict.glm( m.mw.ap , type="response", newdata=nd )
> Mu.nD[,,"F","ap" ] <- predict.glm( f.mw.ap , type="response", newdata=nd )
> Mu.nD[,,"M","apc"] <- predict.glm( m.mw.apc, type="response", newdata=nd )
> Mu.nD[,,"F","apc"] <- predict.glm( f.mw.apc, type="response", newdata=nd )
```

3.5 Average trends

The average annual trends in all of the rates (in %) can now be summarized:

```
> mort.chg <- rbind( ci.exp(m.md.aPC,subset="I\\(P")-1,
+                    ci.exp(f.md.aPC,subset="I\\(P")-1,
+                    ci.exp(m.mw.aPC,subset="I\\(P")-1,
+                    ci.exp(f.mw.aPC,subset="I\\(P")-1 )*100
> rownames( mort.chg ) <- c("Mortality change,   DM: Men",
+                           "                        Women",
+                           "Mortality change, Well: Men",
+                           "                        Women")
> round( rbind( inc.chg, mort.chg ), 2 )
                           exp(Est.)   2.5%  97.5%
DM incidence change      Men      2.03   1.94   2.12
                       Women      2.17   2.07   2.27
<2012                    Men      3.57   3.47   3.68
                       Women      3.54   3.42   3.66
>2012                    Men    -11.09 -11.57 -10.60
                       Women    -12.32 -12.86 -11.78
Mortality change,   DM: Men     -3.65  -3.77  -3.53
                       Women     -3.30  -3.43  -3.16
Mortality change, Well: Men     -2.85  -2.90  -2.79
                       Women     -2.43  -2.48  -2.37
```

Thus it appears that the incidence rates of diabetes overall are increasing by some 2.5% per year, while mortality rates are decreasing 3.5% per year for persons with diabetes, but only 2.5% per year for persons without — the latter two slightly more for men than for women.

The overall trend in the incidence rates of DM is however somewhat misleading; before 2012 the increase was 4%/y, after 2010, there was a decrease of 11%/y (the latter two estimated from a simple linear spline model with a single knot at 2012).

3.6 Time trends in rates

In order to show how the models predict the time trends in diabetes incidence and mortality, we make a graphical display of the estimated rates in ages 20,. . . ,90 versus calendar time, and of the estimated rates at the beginning of 1996, 2000, . . . , 2016 versus age.

3.6.1 Age-specific rates by date

A brief overview of the mortality and incidence rates over time. Recall that for the sake of future calculations the rates in the arrays are in units of $1/\texttt{int} = 1/12$; hence if we want to have rates not in units of `int` (person-years) but in units of 1000 person-years we must multiply by $1000/\texttt{int}$. We also define some hues of blue and red for use in the graph:

```
> gsc <- c("#11CC11","#66CC66","#99CC99")
> rsc <- function(n) rgb( cbind(       255    , seq(0,200,,n), seq(0,200,,n) ), max=255 )
> bsc <- function(n) rgb( cbind( seq(0,200,,n), seq(0,200,,n),       255     ), max=255 )
> pts <- as.numeric( dimnames(Lambda)[[2]] )[1:240]
> ( dimnames(Lambda)[[1]][agr <- seq(240,1080,120)] )
[1] "19.9583333333333" "29.9583333333333" "39.9583333333333" "49.9583333333333"
[5] "59.9583333333333" "69.9583333333333" "79.9583333333333" "89.9583333333333"
> yticks <- outer( c(1:5,7), 10^(-2:2) )
> rpl <-
+ function( Lambda, sx, yl )
+ {
+ plot( NA, ylim=c(0.3,200), xlim=c(1996,2016), yaxt="n", xaxt="n",
+           ylab=yl, xlab="", log="y" )
+ abline( h=yticks, col=gray(0.8) )
+ matlines( pts, t(Lambda[agr,1:240,sx,"apc"])*1000/int,
+          lty=1, lwd=3, type="l",
+          col=if(sx=="M") bsc(8) else rsc(8) )
+ mtext( side=2, yl, line=2.5, las=0, cex=0.66 )
+ }
> par( mfrow=c(3,2), mar=c(1,1,1,1), oma=c(3,3,0,0), mgp=c(3,1,0)/1.6, las=1, bty="n" )
> rpl( Lambda, "M", "DM incidence per 1000 PY" )
>                          axis( side=2 ) ; axis( side=2, at=yticks, labels=NA )
> text( 1996, 200, "Men", adj=c(0,1), cex=1.5, col="blue" )
> rpl( Lambda, "F", "" )
> text( 1996, 200, "Women", adj=c(0,1), cex=1.5, col="red" )
> rpl(  Mu.nD , "M", "Population mortality per 1000 PY" )
>                          axis( side=2 ) ; axis( side=2, at=yticks, labels=NA )
> rpl(  Mu.nD , "F", "" )
> rpl(  Mu.DM, "M", "DM mortality per 1000 PY" )
>                          axis( side=2 ) ; axis( side=2, at=yticks, labels=NA )
>                          axis( side=1 ) ; axis( side=1, at=1996:2016, labels=NA )
> rpl(  Mu.DM, "F", "" ) ; axis( side=1 ) ; axis( side=1, at=1996:2016, labels=NA )
> mtext( "Date of follow-up", side=1, line=2, outer=TRUE, cex=0.66 )
```

3.6.2 Date specific rates by age

First we take a look at the predicted rates in the arrays and tease out the indices on the date-scale that we shall use for plotting the rate by age every 4 years, except that we also include 2014:

```
> str( Lambda )
 num [1:1200, 1:528, 1:2, 1:8] 6.19e-06 6.23e-06 6.27e-06 6.31e-06 6.35e-06 ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.291666666666667"
  ..$ p  : chr [1:528] "1996.04166666667" "1996.125" "1996.20833333333" "1996.29166666667"
  ..$ sex: chr [1:2] "M" "F"
  ..$ mod: chr [1:8] "ap" "apc" "LCa" "att" ...

> range( as.numeric(dimnames(Lambda)[[2]]) )

[1] 1996.042 2039.958

> wh <- 1+c(0,2,4,6,8,9,10)*24
> as.numeric(dimnames(Lambda)[[2]])[wh]

[1] 1996.042 2000.042 2004.042 2008.042 2012.042 2014.042 2016.042

> as.numeric(dimnames(Mu.DM)[[2]])[wh]

[1] 1996.042 2000.042 2004.042 2008.042 2012.042 2014.042 2016.042

> as.numeric(dimnames(Mu.nD)[[2]])[wh]

[1] 1996.042 2000.042 2004.042 2008.042 2012.042 2014.042 2016.042
```

We now plot the rates by age for each of these calendar years, as fitted by the APC model:

```
> a.pt <- as.numeric(dimnames(Lambda)[[1]])
> a.pt <- ndn(Lambda,1)
> rtpl <-
+ function( Lambda, ylb, clr, sx, md, leg=FALSE )
+ {
+ plot( NA, ylim=c(0.3,200), xlim=c(20,95), yaxt="n", xaxt="n",
+           ylab=ylb, xlab="", log="y" )
+ abline( h=yticks, col=gray(0.8) )
+ matlines( a.pt, Lambda[,wh,sx,md]*(1000/int),
+           type="l", lty=1, lwd=3, col=clr )
+ if(sx=="M") axis(side=2)
+ if(leg)
+ text( 20, 200*0.6^(1:length(wh)), round(ndn(Lambda,2)[wh]), col=clr, adj=0 )
+ }
> for( md in c("ap","apc") )
+ par( mfcol=c(3,2), mar=c(1,1,1,1), oma=c(3,3,0,0), mgp=c(3,1,0)/1.6,
+      las=1, bty="n" )
> for( sx in c("M","F") )
+     {
+ lw <- length(wh)
+ clr <- if(sx=="M") bsc(lw)[lw:1] else rsc(lw)[lw:1]
+ clr[5] <- gray(0.5)
+ rtpl( Lambda, if(sx=="M") "DM incidence per 1000 PY" else "", clr, sx, md, leg=TRUE )
+ rtpl( Mu.nD, if(sx=="M") "Population mortality per 1000 PY" else "", clr, sx, md )
+ rtpl( Mu.DM, if(sx=="M") "DM mortality per 1000 PY" else "", clr, sx, md )
+ axis( side=1 )
+ axis(side=1,at=seq(20,95,10),tcl=-0.5,labels=NA)
+ axis(side=1,at=seq(20,95, 5),tcl=-0.3,labels=NA)
+     }
> mtext( "Age at follow-up", side=1, line=2, outer=TRUE, cex=0.66 )
```

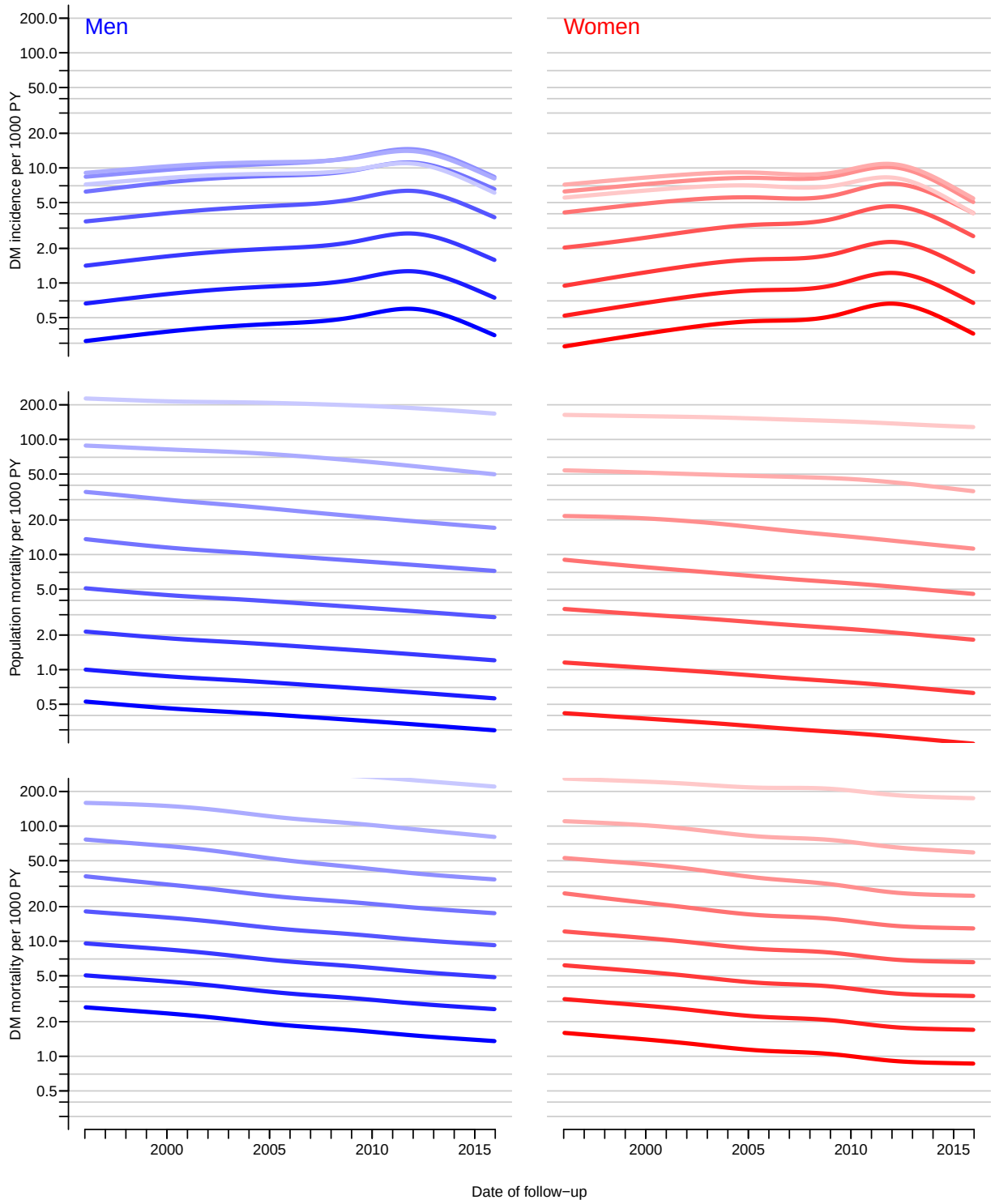

Figure 3.4: *Trends in incidence and mortality rates at ages 20 (darkest), 30, . . . , 90 (lightest), as estimated from separate age-period-cohort models for men and women.* `./rates-ratesbyper`

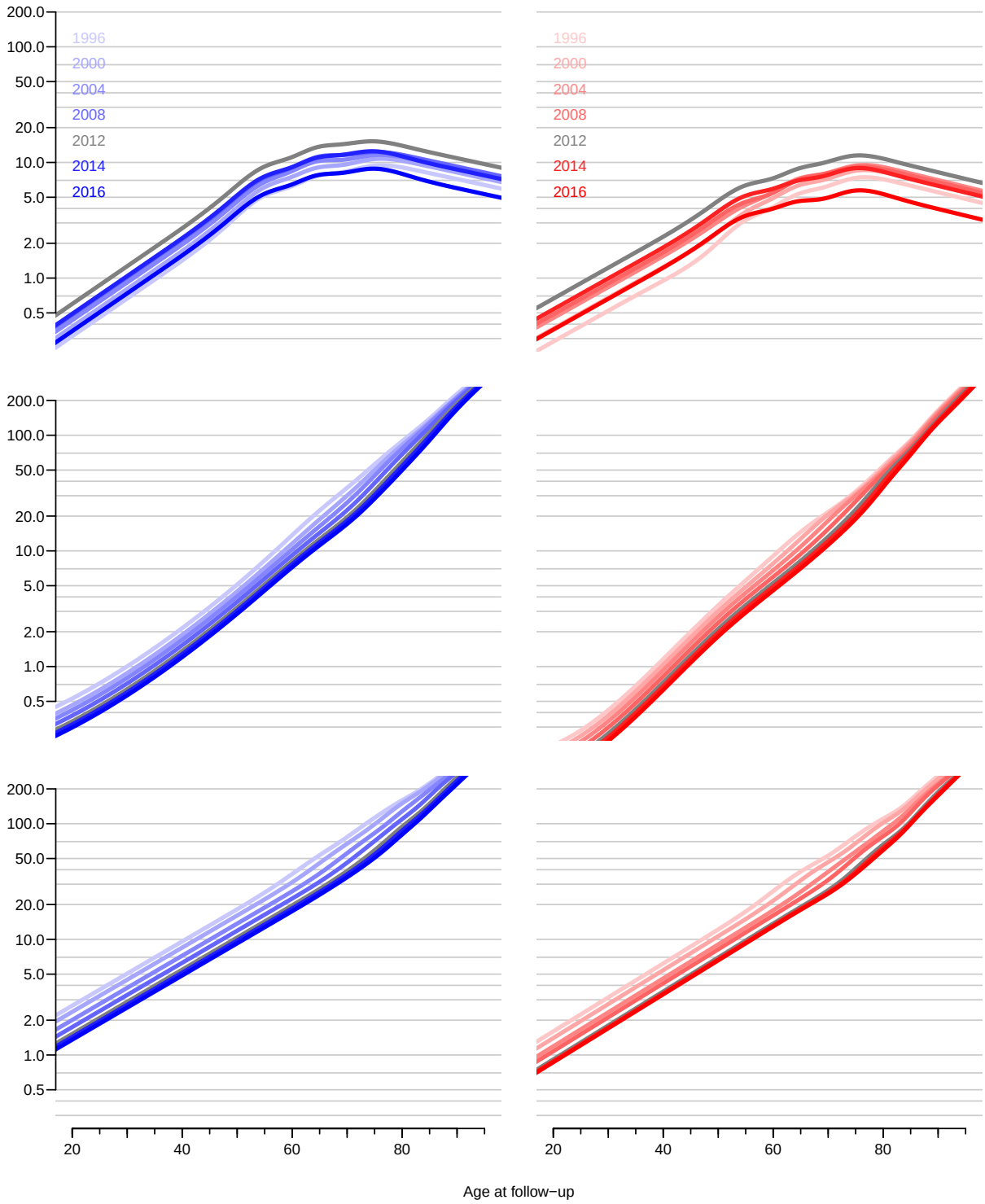

Figure 3.5: *Incidence and mortality rates by age from 1996 through 2016, as estimated from separate age-period-cohort models for men and women.* `./rates-ratesbyage`

3.7 Extrapolation of rates

It is possible to extrapolate the rates beyond the observed dates by simply extending the linear part of the natural splines; in fact this is already done in the rate-objects `Lambda`, `Mu.nD` and `Mu.DM`. However, as seen in figure 3.4 the predicted decline in diabetes rates is presumably way too dramatic.

We therefore repeat the attenuation exercise for mortality rates too, but we shall not add future increases to the rates as we do for sensitivity analyses of the incidence rates.

```
> str( Lambda )
 num [1:1200, 1:528, 1:2, 1:8] 6.19e-06 6.23e-06 6.27e-06 6.31e-06 6.35e-06 ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.291666666666667"
  ..$ p  : chr [1:528] "1996.04166666667" "1996.125" "1996.20833333333" "1996.29166666667"
  ..$ sex: chr [1:2] "M" "F"
  ..$ mod: chr [1:8] "ap" "apc" "LCa" "att" ...
> args( damp )
function (f, t, h, delta = 0, ell = 0, logf = FALSE)
NULL
> # where is the future ?
> wh.p <- (1:length(p.pt))[p.pt>=2016]
> # and what are the dates we are considering ?
> p.pt <- p.pt[p.pt>=2016]
> range( p.pt )
[1] 2016.042 2039.958
> dimnames(Lambda)[[4]]
[1] "ap"  "apc" "LCa" "att" "fix" "p20" "p40" "p60"
```

Digression on `apply`

Note that when we apply a function that returns a vector to an array the returned vector will be the first dimension in the result

```
> ( M <- matrix( 1:6, 3, 2 ) )
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
```

The second argument to `apply` is a vector of dimensions of the object (first argument) to keep, the returned result of the function will be the first dimension of the resulting array, as these two examples show:

```
> apply( M, 1, rev )
     [,1] [,2] [,3]
[1,]    4    5    6
[2,]    1    2    3
> apply( M, 1, function(x) c(x,rev(x)) )
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    4    5    6
[4,]    1    2    3
```

This is what we shall do here, hence the `t()` in the following (and in the code below where predictions are altered):

```
> t( apply( M, 1, rev ) )
     [,1] [,2]
[1,]    4    1
[2,]    5    2
[3,]    6    3
```

. . . end of digression.

3.7.1 Attenuated extrapolations

Here we make the changes to the rates in the period indicated by `wh.p`:

```
> wh.p[1:2]
[1] 241 242
> dimnames( Lambda )[[2]][wh.p[1:2]]
[1] "2016.04166666667" "2016.125"
> system.time(
+ for( is in dimnames(Lambda)[['sex']] )
+    {
+ Lambda[,wh.p,is,"fix"] <- t(apply(Lambda[,wh.p,is,"apc"],1,damp,t=p.pt,h=0.0,delta=0.0))
+ Lambda[,wh.p,is,"att"] <- t(apply(Lambda[,wh.p,is,"apc"],1,damp,t=p.pt,h=3.0,delta=0.0))
+ Lambda[,wh.p,is,"p20"] <- t(apply(Lambda[,wh.p,is,"apc"],1,damp,t=p.pt,h=0.5,delta=2.0,
+                                                                          ell=2))
+ Lambda[,wh.p,is,"p40"] <- t(apply(Lambda[,wh.p,is,"apc"],1,damp,t=p.pt,h=0.5,delta=4.0,
+                                                                          ell=2))
+ Lambda[,wh.p,is,"p60"] <- t(apply(Lambda[,wh.p,is,"apc"],1,damp,t=p.pt,h=0.5,delta=6.0,
+                                                                          ell=2))
+  Mu.DM[,wh.p,is,"fix"] <- t(apply( Mu.DM[,wh.p,is,"apc"],1,damp,t=p.pt,h=0.0,delta=0.0))
+  Mu.DM[,wh.p,is,"att"] <- t(apply( Mu.DM[,wh.p,is,"apc"],1,damp,t=p.pt,h=5.0,delta=0.0))
+  Mu.nD[,wh.p,is,"fix"] <- t(apply( Mu.nD[,wh.p,is,"apc"],1,damp,t=p.pt,h=0.0,delta=0.0))
+  Mu.nD[,wh.p,is,"att"] <- t(apply( Mu.nD[,wh.p,is,"apc"],1,damp,t=p.pt,h=5.0,delta=0.0))
+    } )
   user  system elapsed
  3.650   0.000   3.645
```

In principle we could use the same attenuation along the age-dimension within each cohort, but we are only interested in the calendar time, and even though the cohort-specific curves increase *both* by age and calendar time, we are not interested in attenuation of the *combined effect* of age and calendar time. Or put differently — we are not interested in any attenuation along the age-scale.

For the sake of easier use we fill in the prediction dimenasions for the first years be for we save:

```
> str( Lambda )
 num [1:1200, 1:528, 1:2, 1:8] 6.19e-06 6.23e-06 6.27e-06 6.31e-06 6.35e-06 ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.291666666666667
  ..$ p  : chr [1:528] "1996.04166666667" "1996.125" "1996.20833333333" "1996.29166666667"
  ..$ sex: chr [1:2] "M" "F"
  ..$ mod: chr [1:8] "ap" "apc" "LCa" "att" ...
```

```
> table( is.na(Lambda[700,,1,"att"]))

FALSE  TRUE
  288   240

> ndn(Lambda,2)

  [1] 1996.042 1996.125 1996.208 1996.292 1996.375 1996.458 1996.542 1996.625 1996.708
 [10] 1996.792 1996.875 1996.958 1997.042 1997.125 1997.208 1997.292 1997.375 1997.458
 [19] 1997.542 1997.625 1997.708 1997.792 1997.875 1997.958 1998.042 1998.125 1998.208
 [28] 1998.292 1998.375 1998.458 1998.542 1998.625 1998.708 1998.792 1998.875 1998.958
 [37] 1999.042 1999.125 1999.208 1999.292 1999.375 1999.458 1999.542 1999.625 1999.708
 [46] 1999.792 1999.875 1999.958 2000.042 2000.125 2000.208 2000.292 2000.375 2000.458
 [55] 2000.542 2000.625 2000.708 2000.792 2000.875 2000.958 2001.042 2001.125 2001.208
 [64] 2001.292 2001.375 2001.458 2001.542 2001.625 2001.708 2001.792 2001.875 2001.958
 [73] 2002.042 2002.125 2002.208 2002.292 2002.375 2002.458 2002.542 2002.625 2002.708
 [82] 2002.792 2002.875 2002.958 2003.042 2003.125 2003.208 2003.292 2003.375 2003.458
 [91] 2003.542 2003.625 2003.708 2003.792 2003.875 2003.958 2004.042 2004.125 2004.208
[100] 2004.292 2004.375 2004.458 2004.542 2004.625 2004.708 2004.792 2004.875 2004.958
[109] 2005.042 2005.125 2005.208 2005.292 2005.375 2005.458 2005.542 2005.625 2005.708
[118] 2005.792 2005.875 2005.958 2006.042 2006.125 2006.208 2006.292 2006.375 2006.458
[127] 2006.542 2006.625 2006.708 2006.792 2006.875 2006.958 2007.042 2007.125 2007.208
[136] 2007.292 2007.375 2007.458 2007.542 2007.625 2007.708 2007.792 2007.875 2007.958
[145] 2008.042 2008.125 2008.208 2008.292 2008.375 2008.458 2008.542 2008.625 2008.708
[154] 2008.792 2008.875 2008.958 2009.042 2009.125 2009.208 2009.292 2009.375 2009.458
[163] 2009.542 2009.625 2009.708 2009.792 2009.875 2009.958 2010.042 2010.125 2010.208
[172] 2010.292 2010.375 2010.458 2010.542 2010.625 2010.708 2010.792 2010.875 2010.958
[181] 2011.042 2011.125 2011.208 2011.292 2011.375 2011.458 2011.542 2011.625 2011.708
[190] 2011.792 2011.875 2011.958 2012.042 2012.125 2012.208 2012.292 2012.375 2012.458
[199] 2012.542 2012.625 2012.708 2012.792 2012.875 2012.958 2013.042 2013.125 2013.208
[208] 2013.292 2013.375 2013.458 2013.542 2013.625 2013.708 2013.792 2013.875 2013.958
[217] 2014.042 2014.125 2014.208 2014.292 2014.375 2014.458 2014.542 2014.625 2014.708
[226] 2014.792 2014.875 2014.958 2015.042 2015.125 2015.208 2015.292 2015.375 2015.458
[235] 2015.542 2015.625 2015.708 2015.792 2015.875 2015.958 2016.042 2016.125 2016.208
[244] 2016.292 2016.375 2016.458 2016.542 2016.625 2016.708 2016.792 2016.875 2016.958
[253] 2017.042 2017.125 2017.208 2017.292 2017.375 2017.458 2017.542 2017.625 2017.708
[262] 2017.792 2017.875 2017.958 2018.042 2018.125 2018.208 2018.292 2018.375 2018.458
[271] 2018.542 2018.625 2018.708 2018.792 2018.875 2018.958 2019.042 2019.125 2019.208
[280] 2019.292 2019.375 2019.458 2019.542 2019.625 2019.708 2019.792 2019.875 2019.958
[289] 2020.042 2020.125 2020.208 2020.292 2020.375 2020.458 2020.542 2020.625 2020.708
[298] 2020.792 2020.875 2020.958 2021.042 2021.125 2021.208 2021.292 2021.375 2021.458
[307] 2021.542 2021.625 2021.708 2021.792 2021.875 2021.958 2022.042 2022.125 2022.208
[316] 2022.292 2022.375 2022.458 2022.542 2022.625 2022.708 2022.792 2022.875 2022.958
[325] 2023.042 2023.125 2023.208 2023.292 2023.375 2023.458 2023.542 2023.625 2023.708
[334] 2023.792 2023.875 2023.958 2024.042 2024.125 2024.208 2024.292 2024.375 2024.458
[343] 2024.542 2024.625 2024.708 2024.792 2024.875 2024.958 2025.042 2025.125 2025.208
[352] 2025.292 2025.375 2025.458 2025.542 2025.625 2025.708 2025.792 2025.875 2025.958
[361] 2026.042 2026.125 2026.208 2026.292 2026.375 2026.458 2026.542 2026.625 2026.708
[370] 2026.792 2026.875 2026.958 2027.042 2027.125 2027.208 2027.292 2027.375 2027.458
[379] 2027.542 2027.625 2027.708 2027.792 2027.875 2027.958 2028.042 2028.125 2028.208
[388] 2028.292 2028.375 2028.458 2028.542 2028.625 2028.708 2028.792 2028.875 2028.958
[397] 2029.042 2029.125 2029.208 2029.292 2029.375 2029.458 2029.542 2029.625 2029.708
[406] 2029.792 2029.875 2029.958 2030.042 2030.125 2030.208 2030.292 2030.375 2030.458
[415] 2030.542 2030.625 2030.708 2030.792 2030.875 2030.958 2031.042 2031.125 2031.208
[424] 2031.292 2031.375 2031.458 2031.542 2031.625 2031.708 2031.792 2031.875 2031.958
[433] 2032.042 2032.125 2032.208 2032.292 2032.375 2032.458 2032.542 2032.625 2032.708
[442] 2032.792 2032.875 2032.958 2033.042 2033.125 2033.208 2033.292 2033.375 2033.458
[451] 2033.542 2033.625 2033.708 2033.792 2033.875 2033.958 2034.042 2034.125 2034.208
[460] 2034.292 2034.375 2034.458 2034.542 2034.625 2034.708 2034.792 2034.875 2034.958
```

```
[469] 2035.042 2035.125 2035.208 2035.292 2035.375 2035.458 2035.542 2035.625 2035.708
[478] 2035.792 2035.875 2035.958 2036.042 2036.125 2036.208 2036.292 2036.375 2036.458
[487] 2036.542 2036.625 2036.708 2036.792 2036.875 2036.958 2037.042 2037.125 2037.208
[496] 2037.292 2037.375 2037.458 2037.542 2037.625 2037.708 2037.792 2037.875 2037.958
[505] 2038.042 2038.125 2038.208 2038.292 2038.375 2038.458 2038.542 2038.625 2038.708
[514] 2038.792 2038.875 2038.958 2039.042 2039.125 2039.208 2039.292 2039.375 2039.458
[523] 2039.542 2039.625 2039.708 2039.792 2039.875 2039.958
```

```
> str(Mu.DM)
```

```
 num [1:1200, 1:528, 1:2, 1:5] 5.92e-05 5.95e-05 5.98e-05 6.01e-05 6.05e-05 ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.291666666666667"
  ..$ p  : chr [1:528] "1996.04166666667" "1996.125" "1996.20833333333" "1996.29166666667"
  ..$ sex: chr [1:2] "M" "F"
  ..$ mod: chr [1:5] "ap" "apc" "LCa" "att" ...
```

```
> for( i in 4:8 ) Lambda[,1:240,,i] <- Lambda[,1:240,,"apc"]
> for( i in 4:5 )
+    {
+  Mu.DM[,1:240,,i] <-  Mu.DM[,1:240,,"apc"]
+  Mu.nD[,1:240,,i] <-  Mu.nD[,1:240,,"apc"]
+    }
> save( Lambda, Mu.DM, Mu.nD, file="./data/allrates.Rda" )
```

3.8 The fitted rates 1996–2016

```
> load( file="./data/allrates.Rda" )
> library( Epi )
```

Now we have the three types of rates in the illness-death model in the arrays `Lambda`, `Mu.DM` and `Mu.nD`, so we plot these for select ages as a function of time — again recall they are in units of events per `int`, so we rescale as before to get rates per 1000 PY:

```
> pts <- as.numeric( dimnames(Lambda)[[2]] )
> rpl <-
+ function( Lambda, sx, yl=NULL, inc=NULL, icol=NULL )
+ {
+ clr <- if( sx=="M") bsc(8) else rsc(8)
+ plot( NA, log="y", yaxt="n", ylim=c(0.3,200), xlim=c(1996,2040),
+       ylab="", xaxt="n" )
+ abline( h=yticks, col=gray(0.8) )
+ matlines( pts, t(Lambda[agr,,sx,"apc"])*1000/int,
+           lty=1, lwd=5, type="l", col=clr )
+ matlines( pts, t(Lambda[agr,,sx,"att"])*1000/int,
+           lty=1, lwd=2, type="l", col="forestgreen" )
+ matlines( pts, t(Lambda[agr,,sx,"fix"])*1000/int,
+           lty=1, lwd=2, type="l", col="black" )
+ if( !is.null(inc) )
+ for( ii in 1:length(inc) )
+    matlines( pts, t(Lambda[agr,,sx,inc[ii]])*1000/int,
+              lty=1, lwd=2, type="l", col=icol[ii] )
+ abline( v=2016, lty=3, col=gray(0.6) )
+ mtext( side=2, yl, line=2.5, las=0, cex=0.66 )
+ }
```

```
> par( mfrow=c(3,2), mar=c(1,1,1,1), oma=c(3,3,0,0), mgp=c(3,1,0)/1.6, las=1, bty="n" )
> rpl( Lambda, "M", "DM incidence per 1000 PY", inc=c("p20","p40","p60"), icol=gsc )
> axis( side=2 ) ; axis( side=2, at=yticks, labels=NA, tcl=-0.3 )
> text( 1996, 200, "Men", adj=c(0,1), cex=1.5, col="blue" )
> rpl( Lambda, "F", "", inc=c("p20","p40","p60"), icol=gsc )
> text( 1996, 200, "Women", adj=c(0,1), cex=1.5, col="red" )
> rpl(  Mu.nD, "M", "Population mortality per 1000 PY" )
> axis( side=2 ) ; axis( side=2, at=yticks, labels=NA, tcl=-0.3 )
> rpl(  Mu.nD, "F", "" )
> rpl(  Mu.DM, "M", "DM mortality per 1000 PY" )
> axis( side=2 ) ; axis( side=2, at=yticks, labels=NA, tcl=-0.3 )
> axis( side=1 ) ; axis( side=1, at=seq(1995,2040,5), labels=NA, tcl=-0.4 )
>                  axis( side=1, at=1996:2040, labels=NA, tcl=-0.3 )
> rpl(  Mu.DM, "F", "" )
> axis( side=1 )
> axis( side=1, at=seq(1995,2040,5), labels=NA, tcl=-0.4 )
> axis( side=1, at=seq(1995,2040,1), labels=NA, tcl=-0.3 )
> mtext( "Date of follow-up", side=1, line=2, outer=TRUE )
```

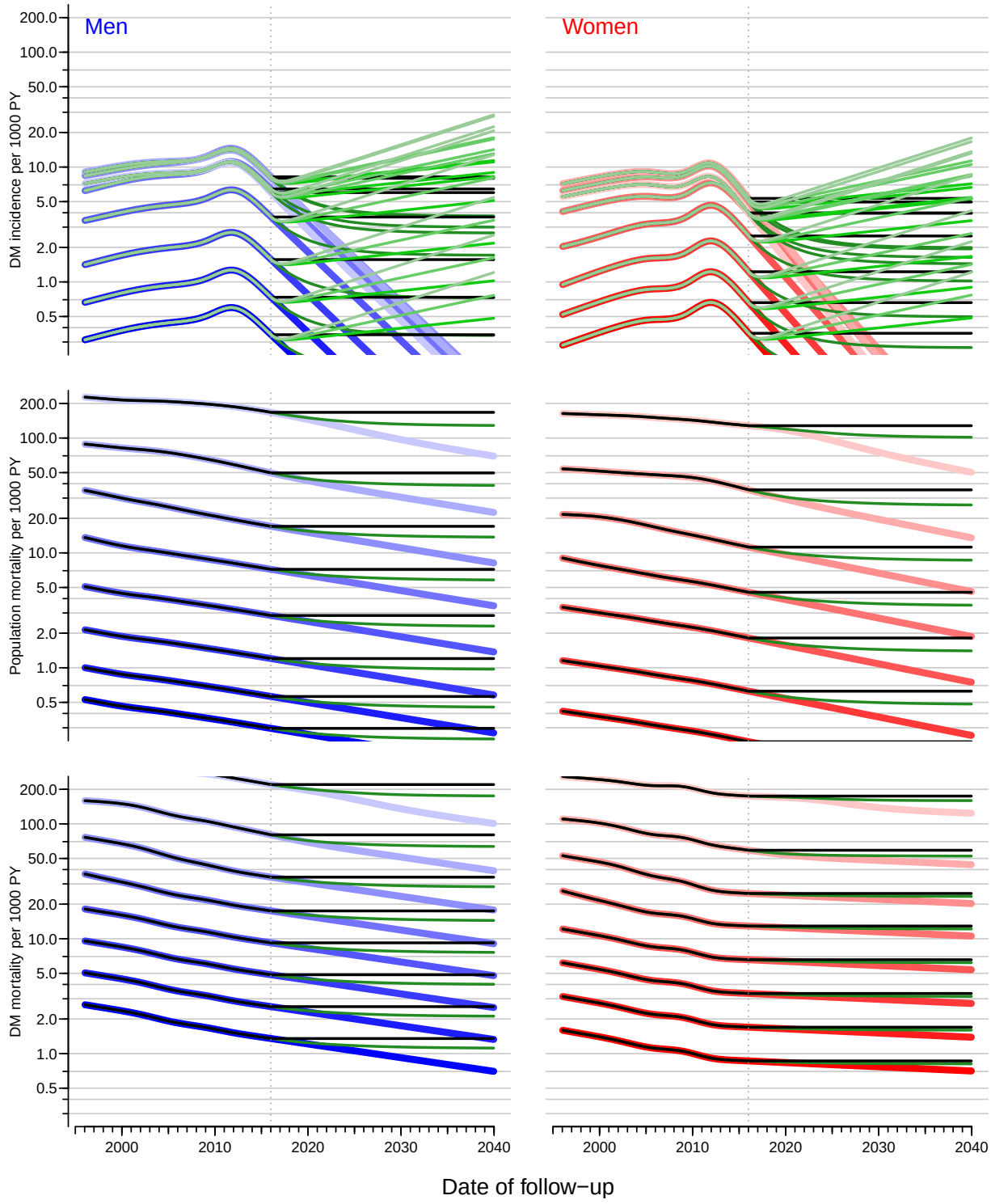

Figure 3.6: *Trends in incidence and mortality rates for ages 20 (darkest), 30, . . . , 90 (lightest), as estimated from the age-period-cohort models and projcted by naive extrapolation of the natural splines.*
The thin black lines indicate predictions fixing all rates to the level at the end of 2015, the thin green lines using an attenuation of the trend halving the slope every 3 years, and the thick green lines for incidence rates represent an annual increase of 2.0, 4.0 and 6.0 % per year (dark to light) added (phased in over 2 years).
The vertical dotted lines indicate the end of available data. `./rates-projrates`

Chapter 4

Predicting prevalence of diabetes

In order to predict the prevalent *number* of DM patients in the future, we use the predicted incidence and mortality rates for the period 2016–2040 to predict the prevalence (fraction with DM) and subsequent multiply this with the population predictions from Statistics Denmark.

We use different scenarios for the `incidence` rates:

- Use the naïvely predicted rates from the APC-model with natural splines — the "`apc`" component of the rate-arrays.

  This will give a prediction of numbers which are presumably the least credible, because of the dramatic decrease seen in the last few years (after 2012).

- Use the attenuated rates — the "`att`" component of the rate-arrays.

- Use the rates fixed at the 2016 level — the "`fix`" component of the rate-arrays.

- Use the rates from 2016 with an annual increase of 2, 4 and 6% respectively — the "`p20`", "`p40`", "`p60`" components of the rate-arrays.

Furthermore, the whole exercise will be repeated with mortality (both non-DM and DM) declining at the predicted pace and with mortality rates attenuated after 1 January 2016 by a half-time of the slope of 3 years and finally with mortality rates fixed at the 2016 level, so in total 18 different scenarios.

4.1 Predicted rates

We will start with the observed (smoothed) age-specific prevalences at 2016-01-01 and then use the different scenarios laid out above to predict the prevalences each year till 2040.

First we load the estimated / predicted rates

```
> library( Epi )
> clear()
> sessionInfo()
R version 3.4.2 (2017-09-28)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 14.04.5 LTS

Matrix products: default
```

```
BLAS: /usr/lib/openblas-base/libopenblas.so.0
LAPACK: /usr/lib/lapack/liblapack.so.3.0

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C                   LC_TIME=en_DK.UTF-8
 [4] LC_COLLATE=en_US.UTF-8     LC_MONETARY=en_US.UTF-8        LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                      LC_ADDRESS=C
[10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] utils     datasets  graphics  grDevices stats     methods   base

other attached packages:
[1] Epi_2.23

loaded via a namespace (and not attached):
 [1] cmprsk_2.2-7      zoo_1.8-0         MASS_7.3-47       compiler_3.4.2
 [5] Matrix_1.2-11     plyr_1.8.4        parallel_3.4.2    survival_2.41-3
 [9] etm_0.6-2         Rcpp_0.12.12      splines_3.4.2     grid_3.4.2
[13] numDeriv_2016.8-1 lattice_0.20-35
> load( file="./data/inits.Rda" )
> load( file="./data/allrates.Rda" )
> lls()
   name    mode     class    dim               size(Kb)
1  a.pt    numeric  numeric  1200                   9.4
2  fC      function function 1                      2.3
3  fCp     function function 1                      2.3
4  fCtable function function 1                      2.3
5  int     numeric  numeric  1                      0.0
6  Lambda  numeric  array    1200 528 2 8       79,324.7
7  Mu.DM   numeric  array    1200 528 2 5       49,624.5
8  Mu.nD   numeric  array    1200 528 2 5       49,624.5
9  nk.a    numeric  numeric  1                      0.0
10 nk.c    numeric  numeric  1                      0.0
11 nk.p    numeric  numeric  1                      0.0
12 p.pt    numeric  numeric  528                    4.2
13 qn      function function 1                      3.5
14 spread  function function 1                     22.2
15 t.pt    numeric  numeric  529                    4.2
```

We shall use the simulation scheme to predict the course of DM prevalence development in the future population under various scenarios of mortality and incidence development. So we set up structures to help calculations and hold results:

`pr.fit` — array of predicted age-specific prevalences at 1995-01-01 to 2016-01-01, smoothed by natural splines. This was derived in the section on prevalence:

```
> load( file="./data/prevalences.Rda" )
> str( pr.fit )
 num [1:2, 1:100, 1:21] 0.000573 0.000562 0.000632 0.000614 0.000697 ...
 - attr(*, "dimnames")=List of 3
  ..$ sex: chr [1:2] "M" "F"
  ..$ A  : chr [1:100] "0.5" "1.5" "2.5" "3.5" ...
  ..$ P  : chr [1:21] "1996" "1997" "1998" "1999" ...

> str( pr.ini )
```

```
 num [1:2, 1:1200, 1:21] 0.000548 0.000539 0.000553 0.000543 0.000557 ...
 - attr(*, "dimnames")=List of 3
  ..$ sex: chr [1:2] "M" "F"
  ..$ A  : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.2916666666
  ..$ P  : chr [1:21] "1996" "1997" "1998" "1999" ...
```

```
> str( pr.obs )
```

```
 table [1:2, 1:23, 1:100, 1:3] 0 0 2 3 0 6 1 1 2 0 ...
 - attr(*, "dimnames")=List of 4
  ..$ sex: chr [1:2] "M" "F"
  ..$ P  : chr [1:23] "1995" "1996" "1997" "1998" ...
  ..$ A  : chr [1:100] "0" "1" "2" "3" ...
  ..$    : chr [1:3] "DM" "N" "pct"
```

```
> dimnames( pr.fit )[[3]]
```

```
 [1] "1996" "1997" "1998" "1999" "2000" "2001" "2002" "2003" "2004" "2005" "2006" "200
[13] "2008" "2009" "2010" "2011" "2012" "2013" "2014" "2015" "2016"
```

`TR` — array of transition probabilities between states no-DM and DM (omitting Death). Transition probabilities are computed under the 18 different extrapolation scenarios. The rates that form the base for these refer to (midpoints of) date intervals of length `int` and are therefore labeled on the period dimension by the midpoint of these, a total of 14/`int`=168. The labeling along the age-dimension is by the border between the age-intervals

`prv` — array of predicted prevalences based on the initial prevalences at 2016-01-01 and the transition probabilities as put in `TR`. Note that we use 2 scenario dimensions, one referring to the 6 scenarios for *incidence* rate prediction:

- `apc` — naïve predictions of rates by extrapolating the linear part of the natural splines.
- `att` — attenuation of the slopes predicted, halving the slope of the predictions every 3 years.
- `fix` — fixing rates at the level of 1 January 2016.
- `p20` — sharp attenuation of the slopes of predicted rates with a slope half time of 6 months, and adding an extra increase of 2% per year, increasing from 0 to full 2% after 2 years.
- `p40` — same as above, but with 4% added.
- `p60` — same as above, but with 6% added.

and the other to the two projection scenarios for *mortality* rates (using the same for mortality rates among person with and without diabetes):

- `apc` — continuing the trend from the spline models, essentially amounting to a continuing decline in mortality.
- `att` — attenuating the decline with a half-time of 3 years; that is the *slope* in mortality is halved every 3 years — over the 14 years prediction period this is a reduction to $0.5^{14/3} = 0.04$ of the original slope.

`fix` — fixing rates at the level of 1 January 2016.

`prn` — array of predicted *number* of DM patients in one-year age classes at the 1 January each year. So the same structure as `prv`, but with substantially fewer entries — namely 1-year age-classes and 1 year spaced dates.

4.2 Transition probabilities

In order to get the predicted prevalences (fractions) of DM 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 an array. But we will only need the rates from 2016 and onward, so we restrict the arrays with the rates to this period, and also to the relevant scenarios:

```
> dimnames( Lambda )[[2]][240:241]
[1] "2015.95833333333" "2016.04166666667"

> dimnames( Lambda )[[4]]

[1] "ap"  "apc" "LCa" "att" "fix" "p20" "p40" "p60"

> dimnames( Mu.nD  )[[4]]

[1] "ap"  "apc" "LCa" "att" "fix"

> rLambda <- Lambda[,-(1:240),,c("apc","att","fix","p20","p40","p60")]
> rMu.nD  <-  Mu.nD[,-(1:240),,c("apc","att","fix")]
> rMu.DM  <-  Mu.DM[,-(1:240),,c("apc","att","fix")]
> states <- c("nD","DM")
> TR <- NArray( c( dimnames( rLambda )[-4],
+                  list( imod = dimnames( rLambda )[[4]],
+                        mmod = dimnames( rMu.nD  )[[4]],
+                        from = states,
+                          to = states ) ) )
> str( TR ) ; fCp( length( TR ) )

 logi [1:1200, 1:288, 1:2, 1:6, 1:3, 1:2, 1:2] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 7
  ..$ a   : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.29166666666666
  ..$ p   : chr [1:288] "2016.04166666667" "2016.125" "2016.20833333333" "2016.29166666667"
  ..$ sex : chr [1:2] "M" "F"
  ..$ imod: chr [1:6] "apc" "att" "fix" "p20" ...
  ..$ mmod: chr [1:3] "apc" "att" "fix"
  ..$ from: chr [1:2] "nD" "DM"
  ..$ to  : chr [1:2] "nD" "DM"

[1] 49,766,400

> str( rLambda ) ; fCp( length( rLambda ) )

 num [1:1200, 1:288, 1:2, 1:6] 6.48e-06 6.52e-06 6.56e-06 6.60e-06 6.64e-06 ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.29166666666666
  ..$ p  : chr [1:288] "2016.04166666667" "2016.125" "2016.20833333333" "2016.29166666667"
  ..$ sex: chr [1:2] "M" "F"
  ..$ mod: chr [1:6] "apc" "att" "fix" "p20" ...

[1] 4,147,200
```

In order to fill `TR`, we need the cumulative incidences over intervals of length `int`. But these were exactly what we predicted in the previous sections by setting the person years equal to `int`.

So we can now compute the one-step transition matrices for every combination of `a.pt` and `p.pt`. Note that we are using the small-interval approximation to the survival: $\exp(-\Lambda - M_{\mathrm{nD}}) \approx 1 - \Lambda - M_{\mathrm{nD}}$ and Λ as the transition probability from nD to DM. This only works because we operate with small intervals where the approximation is good, and in particular where the probability of two transitions in one intervals is negligible

```
> for( ii in dimnames(TR)[["imod"]] )
+ for( im in dimnames(TR)[["mmod"]] )
+    {
+ TR[,,,ii,im,"nD","nD"] <- 1 - (rLambda[,,,ii]+rMu.nD[,,,im])
+ TR[,,,ii,im,"nD","DM"] <-      rLambda[,,,ii]
+ TR[,,,ii,im,"DM","nD"] <- 0
+ TR[,,,ii,im,"DM","DM"] <- 1 -                 rMu.DM[,,,im]
+    }
```

Note that we have not included the "Dead" state in the calculations, because we only bother about the *fraction* of diabetes patients in each age class at each time-point. So the probabilities we compute do not sum to 1 within the "from" states; we only compute how many of the persons alive that end up being alive at the next time point — and in what state they are at that point.

We have now collected the transition probabilities between "nD" and "DM" as well as the probabilities of remaining in each of these two states, all referring to a duration of `int`.

```
> save( TR, file="./data/TRf.Rda" )
```

4.3 Prediction of the observed prevalences

We do not need to predict the population size; we can get away with only predicting the prevalences as fractions. When we multiply the fraction of persons in states (nD,DM) with the transition matrix, we get the fraction of the persons alive at the beginning of the interval (as either nD or DM) that are in states (nD,DM) at the end of the interval. These do not sum to 1 (because the ones dying in the interval are left out), so we rescale to prevalence in each step.

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

```
> dpr <- dimnames(TR)[1:5]
> names( dpr )[2] <- "t"
> dpr[[2]] <- t.pt[t.pt>=2016]
> prv <- NArray( dpr )
> str( prv ) ; fCp( length( prv ) )
 logi [1:1200, 1:289, 1:2, 1:6, 1:3] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 5
  ..$ a   : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.29166666666666
  ..$ t   : chr [1:289] "2016" "2016.08333333333" "2016.16666666667" "2016.25" ...
  ..$ sex : chr [1:2] "M" "F"
  ..$ imod: chr [1:6] "apc" "att" "fix" "p20" ...
  ..$ mmod: chr [1:3] "apc" "att" "fix"
[1] 12,484,800
```

Thus we must make a loop that updates the prevalences at 2016-01-01 to those at subsequent times, (that is next time, next age) but first we must initialize the prevalences as modeled on 2016-01-01, as well as the prevalences at ages 0 (the new-born) that we set to 0; note that we exploit the column major storage of arrays to get `prv` filled with identical values across the two last dimensions (incidence and mortality scenarios).

```
> # Smoothed prevalences at 2016-01-01 - the starting values
> # Repeated by virtue of the column major storage of arrays
> for( sx in c("M","F") ) prv[,"2016",sx,,] <- pr.ini[sx,,"2016"]
> # Prevalences at age 0:
> prv[1,,,,] <- 0
```

From these initial values we can compute the predicted prevalences under the different scenarios. We take the fraction of the population in age class `ia` at time `ip` that end up as diabetes patients at time `ip+1` (and hence in age class `ia+1`), and divide by the fraction of all that remain alive, which is the diabetes patients, *plus* those who survive free of diabetes:

```
> system.time(
+ for( ip in 1:(dim(prv)[2]-1) )
+ for( ia in 1:(dim(prv)[1]-1) )
+ prv[ia+1,ip+1,,,] <- (    prv[ia,ip,,,]  * TR[ia,ip,,,,"DM","DM"]
+                       +(1-prv[ia,ip,,,]) * TR[ia,ip,,,,"nD","DM"] ) /
+                      (    prv[ia,ip,,,]  * TR[ia,ip,,,,"DM","DM"]
+                       +(1-prv[ia,ip,,,]) * TR[ia,ip,,,,"nD","DM"]
+                       +(1-prv[ia,ip,,,]) * TR[ia,ip,,,,"nD","nD"] )
+             )
   user  system elapsed
 17.000   0.007  17.003
```

Note that the code above is particularly simple because we only need to compute the prevalence at the next date and age. If we had had a more elaborate model with, say complications states, the calculations in the loop would have been a matrix-multiplication updating the state-distribution, but this simplification would have been at the expense of another three loop-levels, namely over the three last dimensions of the `prv` array.

We can then show a few of the predicted age-specific prevalences

```
> str( prv )
 num [1:1200, 1:289, 1:2, 1:6, 1:3] 0 0.000933 0.00094 0.000947 0.000954 ...
 - attr(*, "dimnames")=List of 5
  ..$ a   : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.29166666666666
  ..$ t   : chr [1:289] "2016" "2016.08333333333" "2016.16666666667" "2016.25" ...
  ..$ sex : chr [1:2] "M" "F"
  ..$ imod: chr [1:6] "apc" "att" "fix" "p20" ...
  ..$ mmod: chr [1:3] "apc" "att" "fix"
```

```
> pplt <-
+ function( sex, per, isc, msc )
+    {
+ np <- length(per)
+ matplot( a.pt, prv[,per,sex,isc,msc]*100,
+          type="l", lty=1, lwd=c(3:3), col=gray((1:np+1)/(np+5)),
+          xlim=c(20,90), ylim=c(0,30), xaxt="n", yaxt="n", yaxs="i" )
+ abline( h=seq(0,30,5), v=seq(0,90,10), col=gray(0.6), lty="22", lend="butt" )
+ text( 22, 24, paste( sx, ": Inc:", isc, "\nMort:", msc ), adj=c(0,1) )
```

```
+    }
> par( mfcol=c(5,4), mar=c(1,1,0,0), oma=c(3,3,1,1), mgp=c(3,1,0)/1.6, bty="n", las=1 )
> for( sx in c("M","F") )
+ for( ms in c("apc","fix") )
+    {
+ for( is in c("att","fix","p20","p40","p60") )
+    {
+    pplt( sx, paste(seq(2016,2040,3)), is, ms )
+    if(sx=="M" & ms=="apc") axis(side=2)
+    }
+    axis( side=1 )
+    }
> mtext( "Age (years)", side=1, outer=TRUE, line=1 )
> mtext( "Prevalence (%)", side=2, outer=TRUE, line=1, las=0 )

> save( prv, file="./data/prv-pred.Rda" )
> load(      file="./data/prv-pred.Rda" )
```

4.4 The actual numbers of diabetes patients in Denmark

In the previous section we only looked at the age-specific prevalences, because these are the quantities that are driven by the incidence and mortality rates. However, it is also of interest to see how the *number* of diabetes patients would have looked under the different scenarios.

To show the number of patients we set up an array `prn` with *structure* (but not extent) as `prv` to hold the *number* of diabetes patients by category, assuming the age-distribution in the population to be as actually observed (that is as extracted from Statistics Denmark). However `prn` will have 100 age-classes rather than 1200 (100/`int`), and only 15 dates (2016–2040): `prv`. This is because we have the predicted population size in 1-year classes.

```
> dn <- dimnames(prv)
> dn[[1]] <- 0:99
> dn[[2]] <- 2016:2040
> prn <- NArray( dn )
> str( prv ) ; fCp( length(prv) )
 num [1:1200, 1:289, 1:2, 1:6, 1:3] 0 0.000933 0.00094 0.000947 0.000954 ...
 - attr(*, "dimnames")=List of 5
  ..$ a   : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.29166666666666
  ..$ t   : chr [1:289] "2016" "2016.08333333333" "2016.16666666667" "2016.25" ...
  ..$ sex : chr [1:2] "M" "F"
  ..$ imod: chr [1:6] "apc" "att" "fix" "p20" ...
  ..$ mmod: chr [1:3] "apc" "att" "fix"
[1] 12,484,800
> str( prn ) ; fCp( length(prn) )
 logi [1:100, 1:25, 1:2, 1:6, 1:3] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 5
  ..$ a   : chr [1:100] "0" "1" "2" "3" ...
  ..$ t   : chr [1:25] "2016" "2017" "2018" "2019" ...
  ..$ sex : chr [1:2] "M" "F"
  ..$ imod: chr [1:6] "apc" "att" "fix" "p20" ...
  ..$ mmod: chr [1:3] "apc" "att" "fix"
```

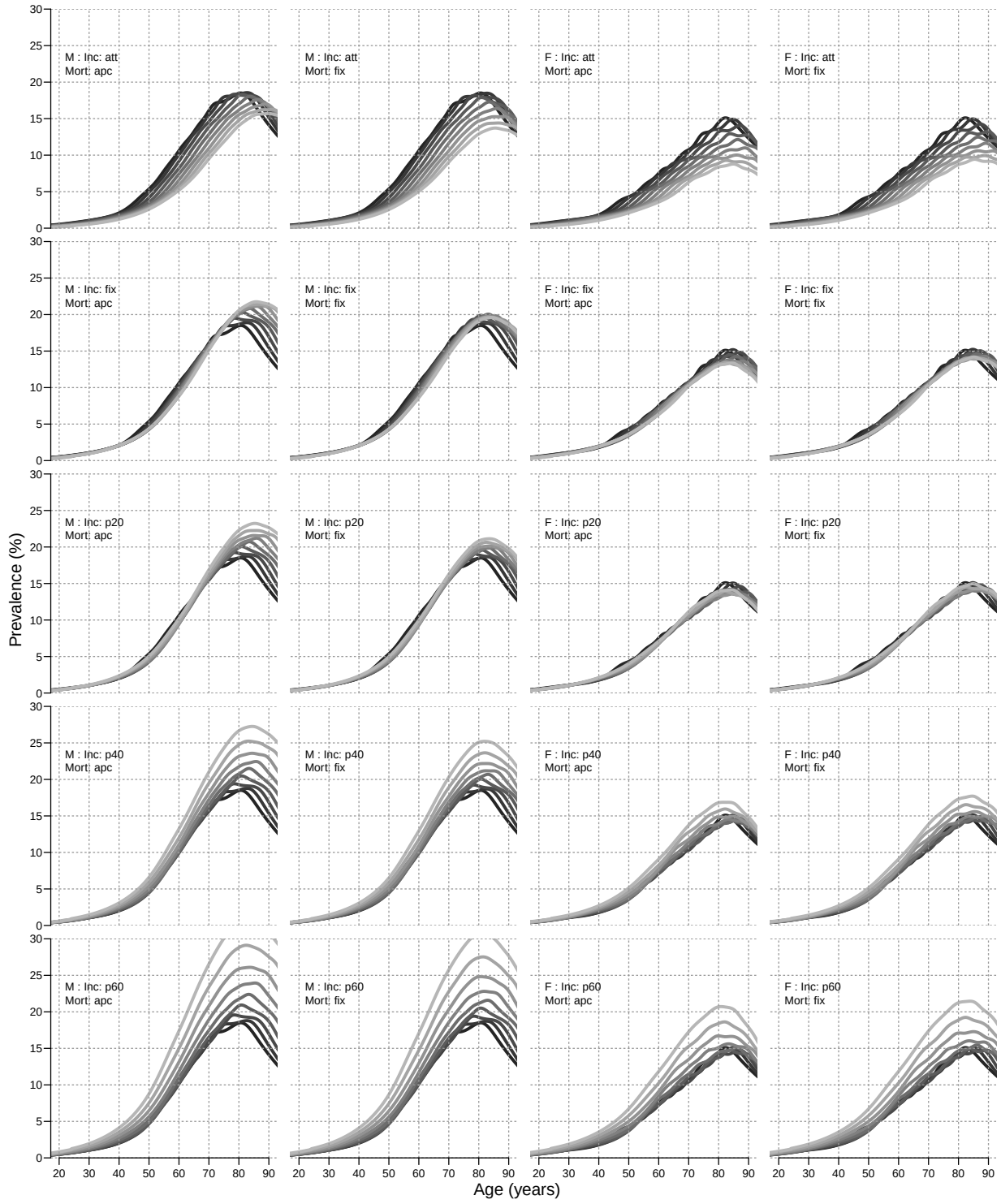

Figure 4.1: *Predicted age-specific prevalences 2016–40 by 3 years for men and women under different scenarios. Colouring is from dark (2016) to light (2040).* `./pred-a-prv`

```
[1]   90,000
```

In order to fill in the numbers we use the estimated age-specific prevalences at 1st January each year, that is at the dates 2016-01-01,...,2040-01-01 in the entries along the `t`-dimension of `prv`. Moreover we want the prevalences for a 1 year age class rather than age-classes of length `int`. So we take the average prevalences from `prv` over each one-year age-interval.

This is really just a simple matrix operation; take a diagonal matrix of 1/12 (well, `int`), and repeat each column 12 (1/`int`) times:

```
> dd <- diag(100)[,rep(1:100,each=1/int)]*int
> round( dd[1:3,1:16], 4 )
      [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]  [,10]  [,11]  [,12]
[1,] 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833
[2,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
[3,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
      [,13]  [,14]  [,15]  [,16]
[1,] 0.0000 0.0000 0.0000 0.0000
[2,] 0.0833 0.0833 0.0833 0.0833
[3,] 0.0000 0.0000 0.0000 0.0000
```

Pre-multiplying this 100×1200 matrix to the 1200 ($= 100/$`int`) age-specific prevalences gives the average prevalences in the 100 1-year age-classes. So we just select the dates at which we want the prevalences:

```
> dimnames(prv)[[2]][wh<-seq(1,289,12)]
 [1] "2016" "2017" "2018" "2019" "2020" "2021" "2022" "2023" "2024" "2025" "2026" "2027"
[13] "2028" "2029" "2030" "2031" "2032" "2033" "2034" "2035" "2036" "2037" "2038" "2039"
[25] "2040"
> for( sx in dimnames(prn)[[3]] )
+ for( im in dimnames(prn)[[4]] )
+ for( mm in dimnames(prn)[[5]] )
+ prn[,,sx,im,mm] <- dd %*% prv[,wh,sx,im,mm]
```

Now `prn` contains the prevalences (as fractions) for 100 age classes and the 25 dates (for each combination of sex and prediction assumptions for incidences and mortalities, respectively). We need to multiply these prevalences by the population figures for each age, date and sex. This is in the array `pop`:

```
> load( "../data/pop.Rda" )
> str( pop )
 'xtabs' int [1:100, 1:46, 1:2] 35612 34747 35080 33328 32973 31898 30863 29539 29151 28304
 - attr(*, "dimnames")=List of 3
  ..$ A  : chr [1:100] "0" "1" "2" "3" ...
  ..$ P  : chr [1:46] "1995" "1996" "1997" "1998" ...
  ..$ sex: chr [1:2] "M" "F"
 - attr(*, "call")= language xtabs(formula = N ~ A + P + sex, data = rbind(befp, subset(bef
 - attr(*, "Label")= chr "DK population size 1995-2040 from DST"


> dmp <- prn
> for( ii in dimnames(dmp)[[4]] )
+ for( im in dimnames(dmp)[[5]] )
+      dmp[,,,ii,im] <- prn[,,,ii,im] * pop[,dimnames(prn)[[2]],]
> save( dmp, file="./data/dmp.Rda" )
> load(      file="./data/dmp.Rda" )
```

4.5 Time trends in prevalent number of DM patients

First we make a table of the total number of DM patients by date, sex and scenario:

```
> str( dmp )
 num [1:100, 1:25, 1:2, 1:6, 1:3] 26.7 31.4 33.9 38.8 42.7 ...
 - attr(*, "dimnames")=List of 5
  ..$ a   : chr [1:100] "0" "1" "2" "3" ...
  ..$ t   : chr [1:25] "2016" "2017" "2018" "2019" ...
  ..$ sex : chr [1:2] "M" "F"
  ..$ imod: chr [1:6] "apc" "att" "fix" "p20" ...
  ..$ mmod: chr [1:3] "apc" "att" "fix"
> dimnames( dmp )[[4]]
[1] "apc" "att" "fix" "p20" "p40" "p60"
> fCtable( addmargins( round( apply( dmp, 2:5, sum ) ), 2 ),
+          col.vars=3, row.vars=c(2:1,4), d=0, w=7 )
              imod     apc     att     fix     p20     p40     p60
sex t    mmod
M   2016 apc       153,841 153,841 153,841 153,841 153,841 153,841
         att       153,841 153,841 153,841 153,841 153,841 153,841
         fix       153,841 153,841 153,841 153,841 153,841 153,841
    2017 apc       155,351 155,391 155,960 155,553 155,564 155,574
         att       155,350 155,390 155,959 155,552 155,563 155,573
         fix       155,325 155,365 155,934 155,528 155,538 155,549
    2018 apc       155,665 155,953 158,062 156,921 157,011 157,102
         att       155,657 155,945 158,053 156,913 157,002 157,093
         fix       155,561 155,849 157,957 156,817 156,907 156,997
    2019 apc       154,961 155,808 160,115 158,289 158,590 158,900
         att       154,932 155,779 160,086 158,260 158,562 158,871
         fix       154,727 155,574 159,880 158,054 158,356 158,665
    2020 apc       153,414 155,161 162,122 159,747 160,415 161,108
         att       153,350 155,096 162,056 159,682 160,349 161,042
         fix       153,005 154,750 161,706 159,334 160,001 160,694
    2021 apc       151,173 154,153 164,077 161,314 162,508 163,766
         att       151,053 154,032 163,953 161,192 162,385 163,643
         fix       150,545 153,522 163,433 160,676 161,868 163,126
    2022 apc       148,367 152,888 165,984 162,999 164,888 166,906
         att       148,170 152,690 165,779 162,796 164,684 166,701
         fix       147,482 151,997 165,068 162,091 163,977 165,993
    2023 apc       145,097 151,432 167,835 164,794 167,553 170,542
         att       144,802 151,133 167,522 164,485 167,242 170,230
         fix       143,922 150,244 166,603 163,575 166,329 169,313
    2024 apc       141,449 149,834 169,622 166,693 170,505 174,691
         att       141,033 149,410 169,175 166,252 170,060 174,243
         fix       139,956 148,317 168,034 165,124 168,926 173,103
    2025 apc       137,503 148,135 171,345 168,698 173,751 179,377
         att       136,946 147,563 170,736 168,096 173,143 178,764
         fix       135,670 146,262 169,364 166,741 171,778 177,388
    2026 apc       133,317 146,359 172,997 170,798 177,287 184,614
         att       132,599 145,617 172,196 170,007 176,487 183,803
         fix       131,126 144,106 170,588 168,418 174,884 182,184
    2027 apc       128,948 144,530 174,574 172,991 181,120 190,423
         att       128,051 143,595 173,554 171,983 180,097 189,382
         fix       126,388 141,877 171,706 170,158 178,250 187,512
    2028 apc       124,443 142,666 176,076 175,276 185,253 196,826
         att       123,353 141,518 174,809 174,023 183,976 195,523
```

	fix	121,509	139,599	172,722	171,961	181,883	193,398
2029	apc	119,846	140,782	177,503	177,651	189,690	203,846
	att	118,549	139,403	175,961	176,126	188,129	202,247
	fix	116,538	137,290	173,638	173,829	185,790	199,863
2030	apc	115,190	138,889	178,853	180,114	194,435	211,504
	att	113,679	137,263	177,012	178,290	192,560	209,574
	fix	111,515	134,967	174,458	175,762	189,977	206,931
2031	apc	110,505	136,996	180,124	182,661	199,491	219,819
	att	108,775	135,109	177,960	180,513	197,272	217,525
	fix	106,476	132,643	175,183	177,760	194,448	214,624
2032	apc	105,823	135,114	181,322	185,296	204,866	228,820
	att	103,871	132,956	178,814	182,800	202,275	226,126
	fix	101,455	130,333	175,823	179,830	199,214	222,969
2033	apc	101,165	133,251	182,446	188,016	210,563	238,527
	att	98,993	130,813	179,576	185,150	207,573	235,401
	fix	96,480	128,047	176,380	181,972	204,281	231,988
2034	apc	96,557	131,418	183,509	190,832	216,599	248,975
	att	94,170	128,695	180,257	187,576	213,182	245,383
	fix	91,578	125,800	176,869	184,196	209,665	241,718
2035	apc	92,017	129,623	184,514	193,746	222,983	260,193
	att	89,423	126,611	180,867	190,080	219,113	256,101
	fix	86,770	123,601	177,296	186,508	215,375	252,186
2036	apc	87,565	127,879	185,474	196,770	229,733	272,220
	att	84,774	124,576	181,419	192,677	225,385	267,595
	fix	82,078	121,464	177,675	188,920	221,431	263,431
2037	apc	83,214	126,191	186,396	199,907	236,860	285,086
	att	80,239	122,596	181,919	195,370	232,008	279,894
	fix	77,516	119,394	178,012	191,434	227,840	275,480
2038	apc	78,981	124,570	187,292	203,171	244,385	298,833
	att	75,834	120,684	182,384	198,173	239,004	293,041
	fix	73,099	117,404	178,322	194,065	234,624	288,377
2039	apc	74,875	123,023	188,172	206,569	252,323	313,496
	att	71,572	118,847	182,823	201,094	246,385	307,069
	fix	68,838	115,498	178,614	196,817	241,795	302,154
2040	apc	70,906	121,555	189,044	210,108	260,689	329,109
	att	67,461	117,091	183,242	204,138	254,167	322,013
	fix	64,741	113,682	178,894	199,697	249,367	316,845
F 2016	apc	127,520	127,520	127,520	127,520	127,520	127,520
	att	127,520	127,520	127,520	127,520	127,520	127,520
	fix	127,520	127,520	127,520	127,520	127,520	127,520
2017	apc	128,173	128,207	128,705	128,346	128,354	128,361
	att	128,173	128,208	128,706	128,347	128,354	128,362
	fix	128,185	128,220	128,718	128,359	128,366	128,374
2018	apc	127,773	128,015	129,844	128,830	128,894	128,959
	att	127,777	128,019	129,848	128,834	128,898	128,963
	fix	127,823	128,065	129,895	128,881	128,945	129,010
2019	apc	126,522	127,219	130,930	129,282	129,497	129,718
	att	126,536	127,232	130,943	129,295	129,511	129,731
	fix	126,633	127,329	131,042	129,393	129,608	129,829
2020	apc	124,598	126,004	131,968	129,783	130,259	130,754
	att	124,627	126,033	131,998	129,812	130,288	130,784
	fix	124,787	126,194	132,161	129,974	130,451	130,947
2021	apc	122,131	124,487	132,950	130,343	131,195	132,094
	att	122,184	124,539	133,004	130,396	131,249	132,148
	fix	122,416	124,773	133,244	130,633	131,487	132,386
2022	apc	119,244	122,756	133,882	130,973	132,323	133,766
	att	119,328	122,840	133,970	131,060	132,410	133,853

	fix	119,638	123,154	134,294	131,381	132,732	134,176
2023	apc	116,028	120,871	134,760	131,674	133,647	135,785
	att	116,151	120,995	134,891	131,803	133,777	135,916
	fix	116,544	121,394	135,307	132,214	134,189	136,331
2024	apc	112,555	118,870	135,577	132,438	135,165	138,162
	att	112,725	119,043	135,761	132,619	135,347	138,347
	fix	113,202	119,529	136,275	133,125	135,857	138,859
2025	apc	108,887	116,784	136,329	133,261	136,878	140,911
	att	109,110	117,014	136,576	133,504	137,124	141,159
	fix	109,673	117,590	137,191	134,109	137,734	141,775
2026	apc	105,079	114,641	137,016	134,144	138,792	144,048
	att	105,362	114,934	137,335	134,459	139,111	144,371
	fix	106,008	115,600	138,054	135,166	139,826	145,095
2027	apc	101,174	112,460	137,636	135,086	140,911	147,591
	att	101,522	112,823	138,038	135,481	141,312	148,000
	fix	102,250	113,578	138,862	136,292	142,135	148,834
2028	apc	97,209	110,256	138,188	136,085	143,235	151,554
	att	97,626	110,695	138,680	136,569	143,729	152,059
	fix	98,432	111,537	139,612	137,485	144,662	153,009
2029	apc	93,214	108,041	138,668	137,135	145,766	155,952
	att	93,702	108,559	139,259	137,716	146,362	156,564
	fix	94,581	109,486	140,298	138,739	147,406	157,632
2030	apc	89,219	105,829	139,083	138,242	148,513	160,806
	att	89,779	106,431	139,780	138,928	149,219	161,536
	fix	90,726	107,440	140,927	140,058	150,378	162,727
2031	apc	85,241	103,624	139,425	139,398	151,470	166,128
	att	85,874	104,313	140,235	140,196	152,297	166,987
	fix	86,884	105,400	141,488	141,432	153,570	168,302
2032	apc	81,304	101,439	139,703	140,609	154,649	171,941
	att	82,010	102,217	140,631	141,526	155,604	172,941
	fix	83,076	103,377	141,990	142,868	156,994	174,385
2033	apc	77,421	99,278	139,915	141,871	158,049	178,263
	att	78,200	100,148	140,968	142,914	159,143	179,415
	fix	79,315	101,376	142,429	144,360	160,649	180,989
2034	apc	73,611	97,153	140,069	143,191	161,683	185,120
	att	74,460	98,114	141,252	144,367	162,924	186,436
	fix	75,617	99,405	142,813	145,915	164,546	188,144
2035	apc	69,885	95,068	140,170	144,572	165,557	192,536
	att	70,801	96,121	141,487	145,886	166,954	194,029
	fix	71,992	97,469	143,144	147,533	168,691	195,872
2036	apc	66,256	93,033	140,228	146,022	169,685	200,544
	att	67,235	94,177	141,683	147,479	171,245	202,226
	fix	68,452	95,576	143,430	149,221	173,097	204,204
2037	apc	62,732	91,054	140,248	147,544	174,076	209,170
	att	63,769	92,285	141,842	149,146	175,805	211,051
	fix	65,004	93,727	143,672	150,979	177,768	213,165
2038	apc	59,324	89,138	140,242	149,148	178,744	218,452
	att	60,412	90,454	141,973	150,897	180,649	220,541
	fix	61,656	91,932	143,881	152,816	182,720	222,790
2039	apc	56,037	87,291	140,217	150,840	183,704	228,420
	att	57,169	88,686	142,084	152,736	185,786	230,726
	fix	58,415	90,194	144,063	154,736	187,963	233,111
2040	apc	52,877	85,517	140,181	152,627	188,966	239,110
	att	54,045	86,986	142,181	154,668	191,230	241,641
	fix	55,287	88,518	144,226	156,746	193,512	244,162
Sum 2016	apc	281,361	281,361	281,361	281,361	281,361	281,361
	att	281,361	281,361	281,361	281,361	281,361	281,361

	fix	281,361	281,361	281,361	281,361	281,361	281,361
2017	apc	283,524	283,598	284,665	283,899	283,918	283,935
	att	283,523	283,598	284,665	283,899	283,917	283,935
	fix	283,510	283,585	284,652	283,887	283,904	283,923
2018	apc	283,438	283,968	287,906	285,751	285,905	286,061
	att	283,434	283,964	287,901	285,747	285,900	286,056
	fix	283,384	283,914	287,852	285,698	285,852	286,007
2019	apc	281,483	283,027	291,045	287,571	288,087	288,618
	att	281,468	283,011	291,029	287,555	288,073	288,602
	fix	281,360	282,903	290,922	287,447	287,964	288,494
2020	apc	278,012	281,165	294,090	289,530	290,674	291,862
	att	277,977	281,129	294,054	289,494	290,637	291,826
	fix	277,792	280,944	293,867	289,308	290,452	291,641
2021	apc	273,304	278,640	297,027	291,657	293,703	295,860
	att	273,237	278,571	296,957	291,588	293,634	295,791
	fix	272,961	278,295	296,677	291,309	293,355	295,512
2022	apc	267,611	275,644	299,866	293,972	297,211	300,672
	att	267,498	275,530	299,749	293,856	297,094	300,554
	fix	267,120	275,151	299,362	293,472	296,709	300,169
2023	apc	261,125	272,303	302,595	296,468	301,200	306,327
	att	260,953	272,128	302,413	296,288	301,019	306,146
	fix	260,466	271,638	301,910	295,789	300,518	305,644
2024	apc	254,004	268,704	305,199	299,131	305,670	312,853
	att	253,758	268,453	304,936	298,871	305,407	312,590
	fix	253,158	267,846	304,309	298,249	304,783	311,962
2025	apc	246,390	264,919	307,674	301,959	310,629	320,288
	att	246,056	264,577	307,312	301,600	310,267	319,923
	fix	245,343	263,852	306,555	300,850	309,512	319,163
2026	apc	238,396	261,000	310,013	304,942	316,079	328,662
	att	237,961	260,551	309,531	304,466	315,598	328,174
	fix	237,134	259,706	308,642	303,584	314,710	327,279
2027	apc	230,122	256,990	312,210	308,077	322,031	338,014
	att	229,573	256,418	311,592	307,464	321,409	337,382
	fix	228,638	255,455	310,568	306,450	320,385	336,346
2028	apc	221,652	252,922	314,264	311,361	328,488	348,380
	att	220,979	252,213	313,489	310,592	327,705	347,582
	fix	219,941	251,136	312,334	309,446	326,545	346,407
2029	apc	213,060	248,823	316,171	314,786	335,456	359,798
	att	212,251	247,962	315,220	313,842	334,491	358,811
	fix	211,119	246,776	313,936	312,568	333,196	357,495
2030	apc	204,409	244,718	317,936	318,356	342,948	372,310
	att	203,458	243,694	316,792	317,218	341,779	371,110
	fix	202,241	242,407	315,385	315,820	340,355	369,658
2031	apc	195,746	240,620	319,549	322,059	350,961	385,947
	att	194,649	239,422	318,195	320,709	349,569	384,512
	fix	193,360	238,043	316,671	319,192	348,018	382,926
2032	apc	187,127	236,553	321,025	325,905	359,515	400,761
	att	185,881	235,173	319,445	324,326	357,879	399,067
	fix	184,531	233,710	317,813	322,698	356,208	397,354
2033	apc	178,586	232,529	322,361	329,887	368,612	416,790
	att	177,193	230,961	320,544	328,064	366,716	414,816
	fix	175,795	229,423	318,809	326,332	364,930	412,977
2034	apc	170,168	228,571	323,578	334,023	378,282	434,095
	att	168,630	226,809	321,509	331,943	376,106	431,819
	fix	167,195	225,205	319,682	330,111	374,211	429,862
2035	apc	161,902	224,691	324,684	338,318	388,540	452,729
	att	160,224	222,732	322,354	335,966	386,067	450,130

```
         fix       158,762 221,070 320,440 334,041 384,066 448,058
    2036 apc       153,821 220,912 325,702 342,792 399,418 472,764
         att       152,009 218,753 323,102 340,156 396,630 469,821
         fix       150,530 217,040 321,105 338,141 394,528 467,635
    2037 apc       145,946 217,245 326,644 347,451 410,936 494,256
         att       144,008 214,881 323,761 344,516 407,813 490,945
         fix       142,520 213,121 321,684 342,413 405,608 488,645
    2038 apc       138,305 213,708 327,534 352,319 423,129 517,285
         att       136,246 211,138 324,357 349,070 419,653 513,582
         fix       134,755 209,336 322,203 346,881 417,344 511,167
    2039 apc       130,912 210,314 328,389 357,409 436,027 541,916
         att       128,741 207,533 324,907 353,830 432,171 537,795
         fix       127,253 205,692 322,677 351,553 429,758 535,265
    2040 apc       123,783 207,072 329,225 362,735 449,655 568,219
         att       121,506 204,077 325,423 358,806 445,397 563,654
         fix       120,028 202,200 323,120 356,443 442,879 561,007
> fCtable( addmargins( apply( dmp[,c(1,5,10,15),,"p40","apc"], 2:3, sum ) ), w=10 )
     sex          M          F        Sum
t
2016        153,841    127,520    281,362
2020        160,415    130,259    290,674
2025        173,751    136,878    310,628
2030        194,435    148,513    342,948
Sum         682,442    543,170  1,225,611
```

We would like to see the overall change in the number of diabetes patients, as recorded in the structure `dmp`

```
> DMall <- dmp[,,"M",,] + dmp[,,"F",,]
> str( DMall )
 num [1:100, 1:25, 1:6, 1:3] 50.9 59.9 65.3 74.6 83.4 ...
 - attr(*, "dimnames")=List of 4
  ..$ a   : chr [1:100] "0" "1" "2" "3" ...
  ..$ t   : chr [1:25] "2016" "2017" "2018" "2019" ...
  ..$ imod: chr [1:6] "apc" "att" "fix" "p20" ...
  ..$ mmod: chr [1:3] "apc" "att" "fix"
> DMcum <- apply( DMall, 2:4, cumsum )
> DMcum <- DMcum[c(1,1:100),,,]
> DMcum[1,,,] <- 0
> DMcum <- DMcum/1000
> str( DMcum )
 num [1:101, 1:25, 1:6, 1:3] 0 0.0509 0.1108 0.1761 0.2507 ...
 - attr(*, "dimnames")=List of 4
  ..$ a   : chr [1:101] "0" "0" "1" "2" ...
  ..$ t   : chr [1:25] "2016" "2017" "2018" "2019" ...
  ..$ imod: chr [1:6] "apc" "att" "fix" "p20" ...
  ..$ mmod: chr [1:3] "apc" "att" "fix"
> # Same operation for empirival numbers
> str( pr.obs )
 table [1:2, 1:23, 1:100, 1:3] 0 0 2 3 0 6 1 1 2 0 ...
 - attr(*, "dimnames")=List of 4
  ..$ sex: chr [1:2] "M" "F"
  ..$ P  : chr [1:23] "1995" "1996" "1997" "1998" ...
  ..$ A  : chr [1:100] "0" "1" "2" "3" ...
  ..$    : chr [1:3] "DM" "N" "pct"
```

```
> Ocum <- apply( pr.obs["M",,,"DM"]+pr.obs["F",,,"DM"], 1, cumsum )
> Ocum <- Ocum[c(1,1:100),]
> Ocum[1,] <- 0
> Ocum <- Ocum[,paste(1996:2015)]/1000
> str( Ocum )
 num [1:101, 1:20] 0 0.005 0.02 0.039 0.071 0.094 0.141 0.194 0.266 0.346 ...
 - attr(*, "dimnames")=List of 2
  ..$ A: chr [1:101] "0" "0" "1" "2" ...
  ..$ P: chr [1:20] "1996" "1997" "1998" "1999" ...
```

Finally we can plot the predicted numbers from the different scenarios (stacking across age-classes):

```
> range( DMcum )
[1]   0.0000 568.2196
> range(  Ocum )
[1]   0.000 268.565


> ryr <- c(2016:2040,2040:2016)
> leg <- c("DM Incidence fixed at 2016 level",
+          "Linear projection of DM inc. from 2016",
+          "Attenuated linear projection of DM inc.",
+          "DM incidence increasing 2.0%/y",
+          "DM incidence increasing 4.0%/y",
+          "DM incidence increasing 6.0%/y")
> names( leg ) <- c("fix","apc","att","p20","p40","p60")
> cbind( leg )
    leg
fix "DM Incidence fixed at 2016 level"
apc "Linear projection of DM inc. from 2016"
att "Attenuated linear projection of DM inc."
p20 "DM incidence increasing 2.0%/y"
p40 "DM incidence increasing 4.0%/y"
p60 "DM incidence increasing 6.0%/y"
> pl.num <-
+ function( wh.m, mtxt, prmax=2030, ymax=400, add.old=TRUE )
+ {
+ if( add.old) ryr <- c(1996:2040,2040:1996)
+ par( mfrow=c(3,2), mar=c(2,1,0,3), oma=c(0,0,2,0), mgp=c(3,1,0)/1.6, las=1, bty="n" )
+ xl <- c( 2016-add.old*20, prmax )
+ for( wh.i in c("apc","att","fix","p20","p40","p60") ) # c(2,3,1,4:6) )
+    {
+ plot( NA,
+       xlim=xl, xlab="", xaxt="n", xaxs="i",
+       ylim=c(0,ymax), xaxs="i", yaxs="i", yaxt="n", ylab="" )
+ axis( side=4, at=0:9*100 )
+ axis( side=4, at=seq(0,9,1/4)*100, labels=NA, tcl=-0.3 )
+ axis( side=1, at=seq(1990,2040,10) )
+ axis( side=1, at=seq(1995,2040,1), labels=NA, tcl=-0.3 )
+ for( i in 1:10 ) polygon( ryr,
+                           if( add.old ){ c( Ocum[1+(i-1)*10,],
+                                             DMcum[1+(i-1)*10,,wh.i,wh.m],
+                                          rev(c(Ocum[1+ i   *10,],
+                                             DMcum[1+ i   *10,,wh.i,wh.m])) )
```

```
+                          } else { c( DMcum[1+(i-1)*10,,wh.i,wh.m],
+                                      rev( DMcum[1+ i   *10,,wh.i,wh.m] ) ) },
+                            col=gray( (17-i)/18 ), border=gray(0.8) ) #"transparent" )
+ abline( h=seq(50,400,50), v=seq(2020,2040,5), col=gray(1), lty="14", lend="butt" )
+ if( add.old ) segments( 2016, 0, 2016, 320 )
+ for( i in seq(55,85,10) ) text( prmax-0.2, DMcum[paste(i),paste(prmax-1),wh.i,wh.m],
+                                 paste( i-5,"-",i+4,sep=""), adj=c(1,1) )
+ text( xl[1]+1, ymax*0.9, paste( wh.i, ": ", leg[wh.i], sep="" ), adj=c(0,1) )
+    }
+ mtext( mtxt, side=3, line=1, outer=TRUE, cex=0.66, adj=0 )
+ }
> pl.num( "apc", "Mortality constantly decreasing" )


> pl.num( "apc", "Mortality constantly decreasing", 2040, 550 )


> pl.num( "att", "Mortality decrease attenuated" )


> pl.num( "att", "Mortality decrease attenuated", 2040, 550 )


> pl.num( "fix", "Mortality fixed at 2016" )


> pl.num( "fix", "Mortality fixed at 2016", 2040, 550 )
```

From figure 4.2 it appears that it is the decreasing incidence rates of diabetes that carries the major differences of more than 100,000 patients in 2040. The decrease in the number of incident cases is very recent; during the period 2012–2014 there was a drop and a very slight pick-up during 2015.

Thus the prediction of the number of future patients is crucially dependent on the tiny amount of information available about future diabetes incidence rates in the rather odd behaviour of the rates in the years 2012 through 2015.

However, even the quite brutal assumption of a pick up of increasing DM incidence rates by 5% per year will not bring the predicted number of patients over 400,000 in 2040. So to say that the number of diabetes patients is less than this in 2040 seems to be a fairly safe bet.

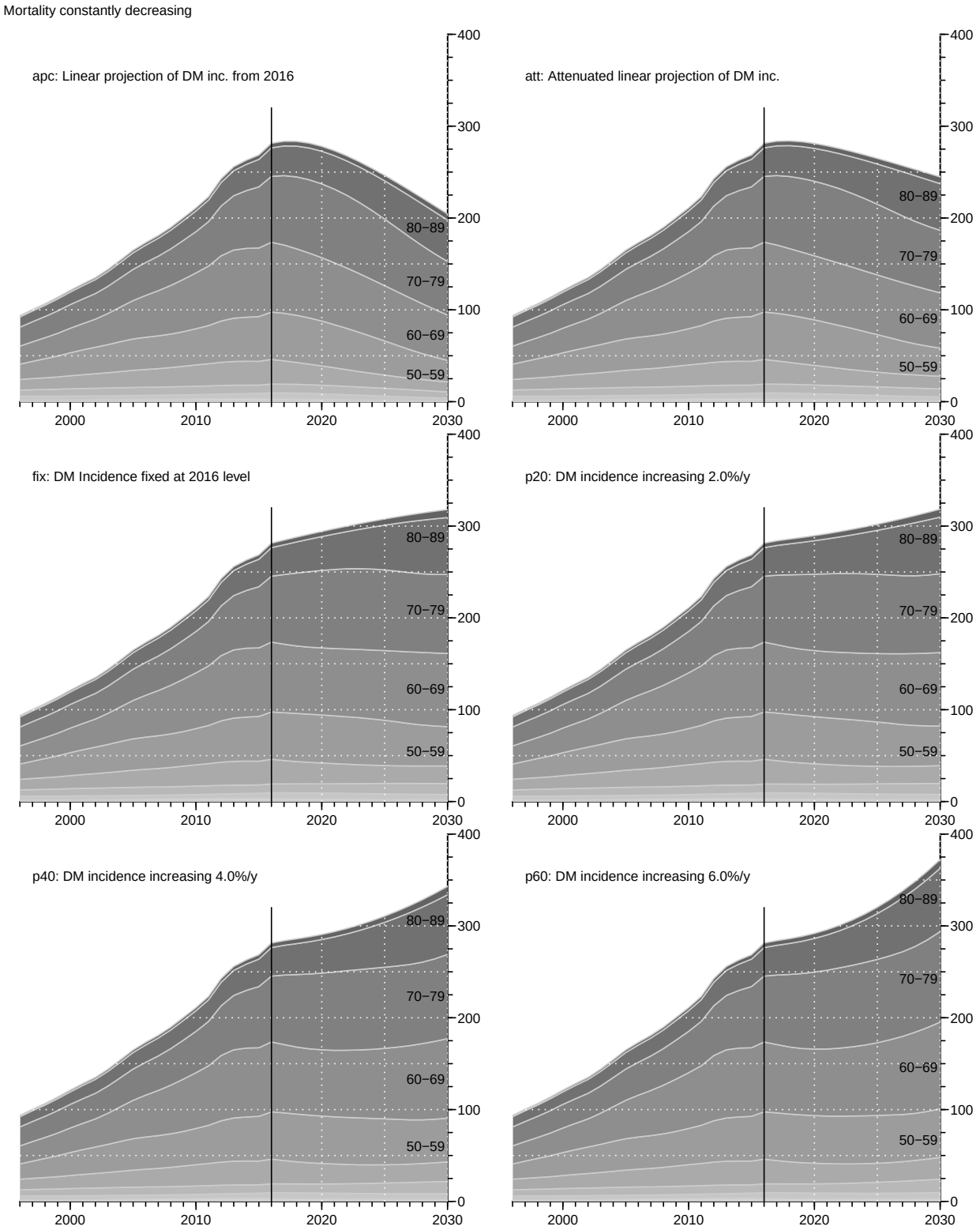

Figure 4.2: *Predicted number of diabetes patients in Denmark under different incidence rate scenarios, using a continuing decrease in mortality (both for non-DM and DM persons).* `./pred-prnum-apc`

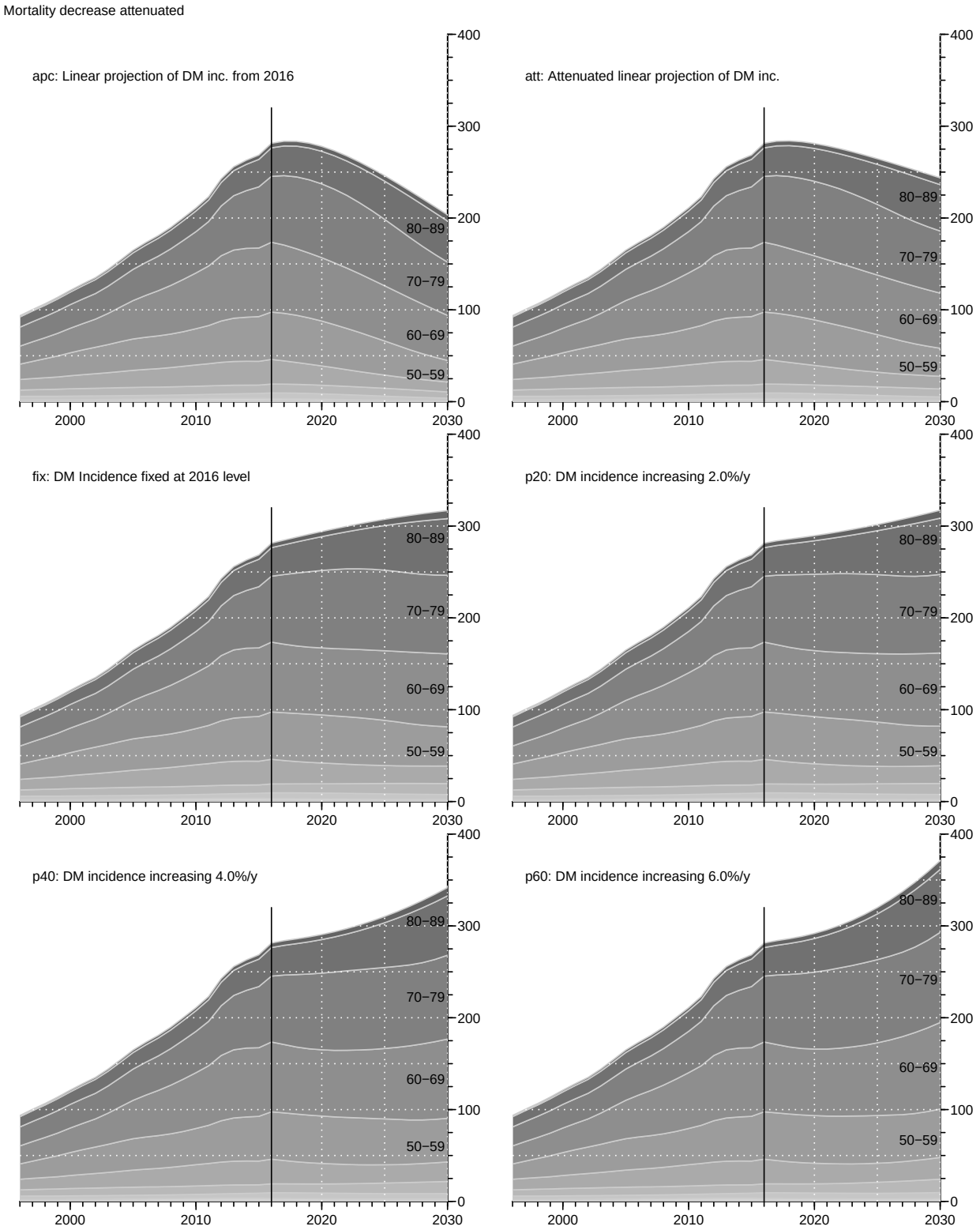

Figure 4.3: *Predicted number of diabetes patients in Denmark under different incidence rate scenarios, using an attenuated decrease in mortality (both for non-DM and DM persons).* `./pred-prnum-att`

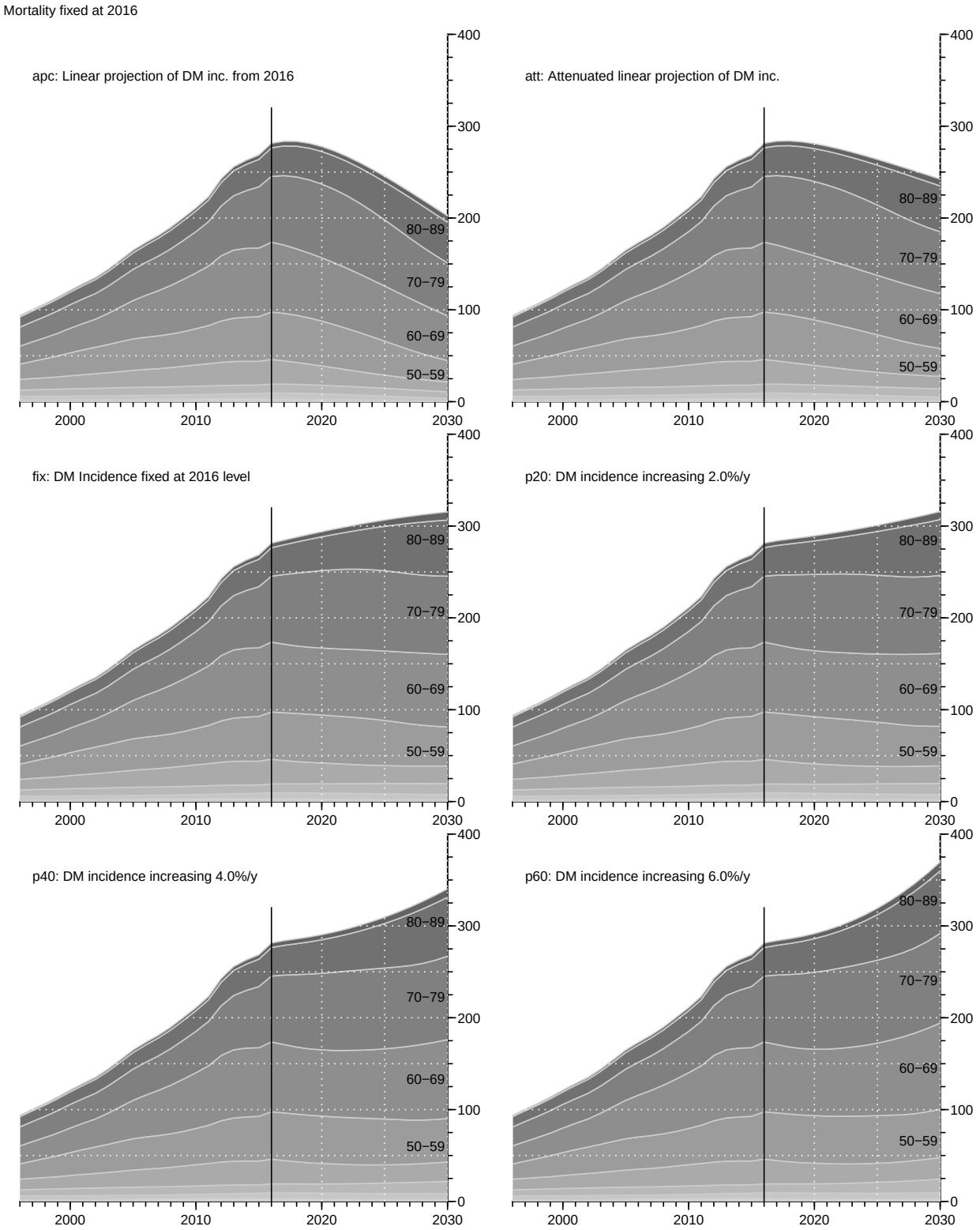

Figure 4.4: *Predicted number of diabetes patients in Denmark under different incidence rate scenarios, using mortality (both for non-DM and DM persons) fixed at the 2016 level.* `./pred-prnum-fix`

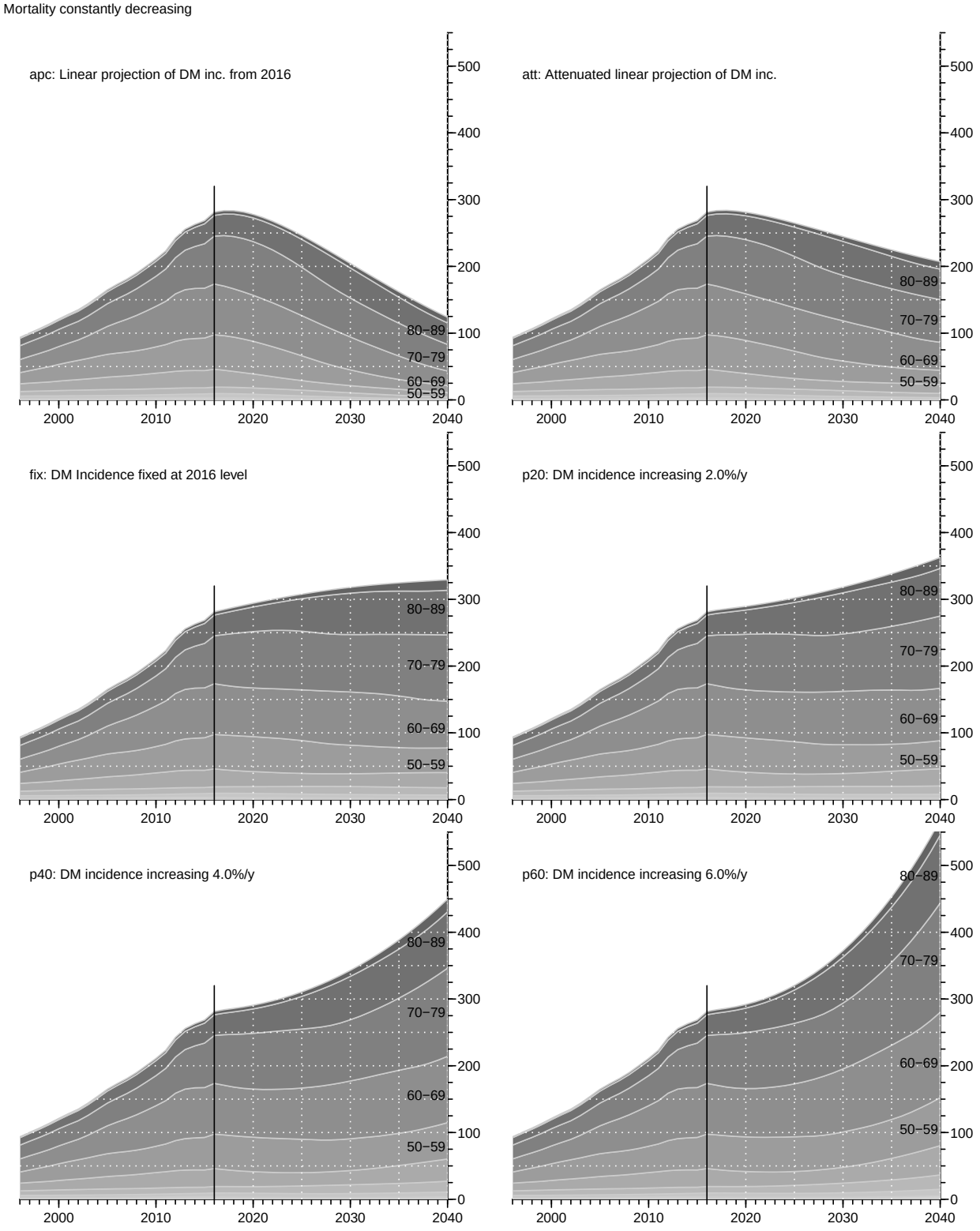

Figure 4.5: *Predicted number of diabetes patients in Denmark under different incidence rate scenarios, using a continuing decrease in mortality (both for non-DM and DM persons).* `./pred-prnum-apc2040`

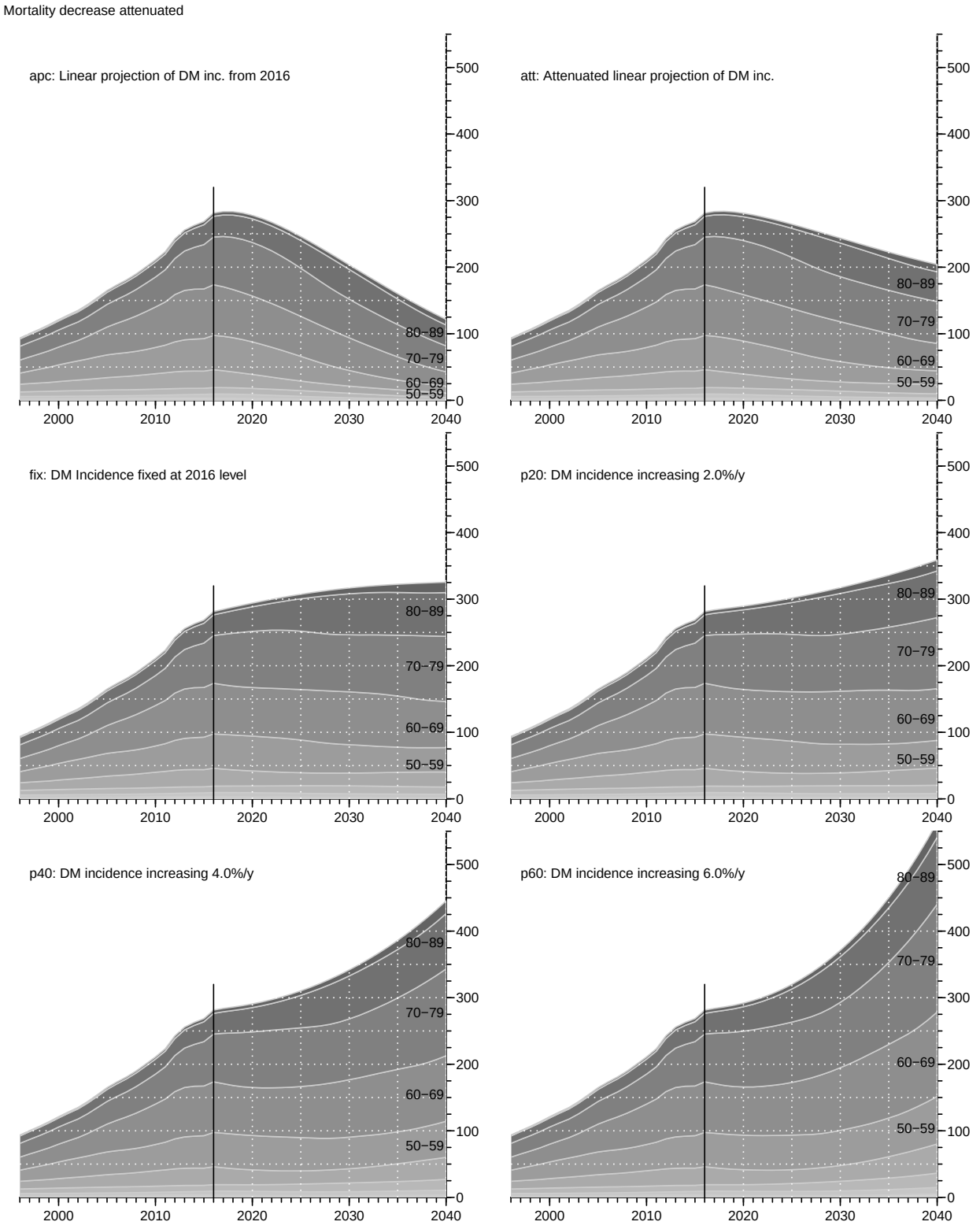

Figure 4.6: *Predicted number of diabetes patients in Denmark under different incidence rate scenarios, using an attenuated decrease in mortality (both for non-DM and DM persons).* `./pred-prnum-att2040`

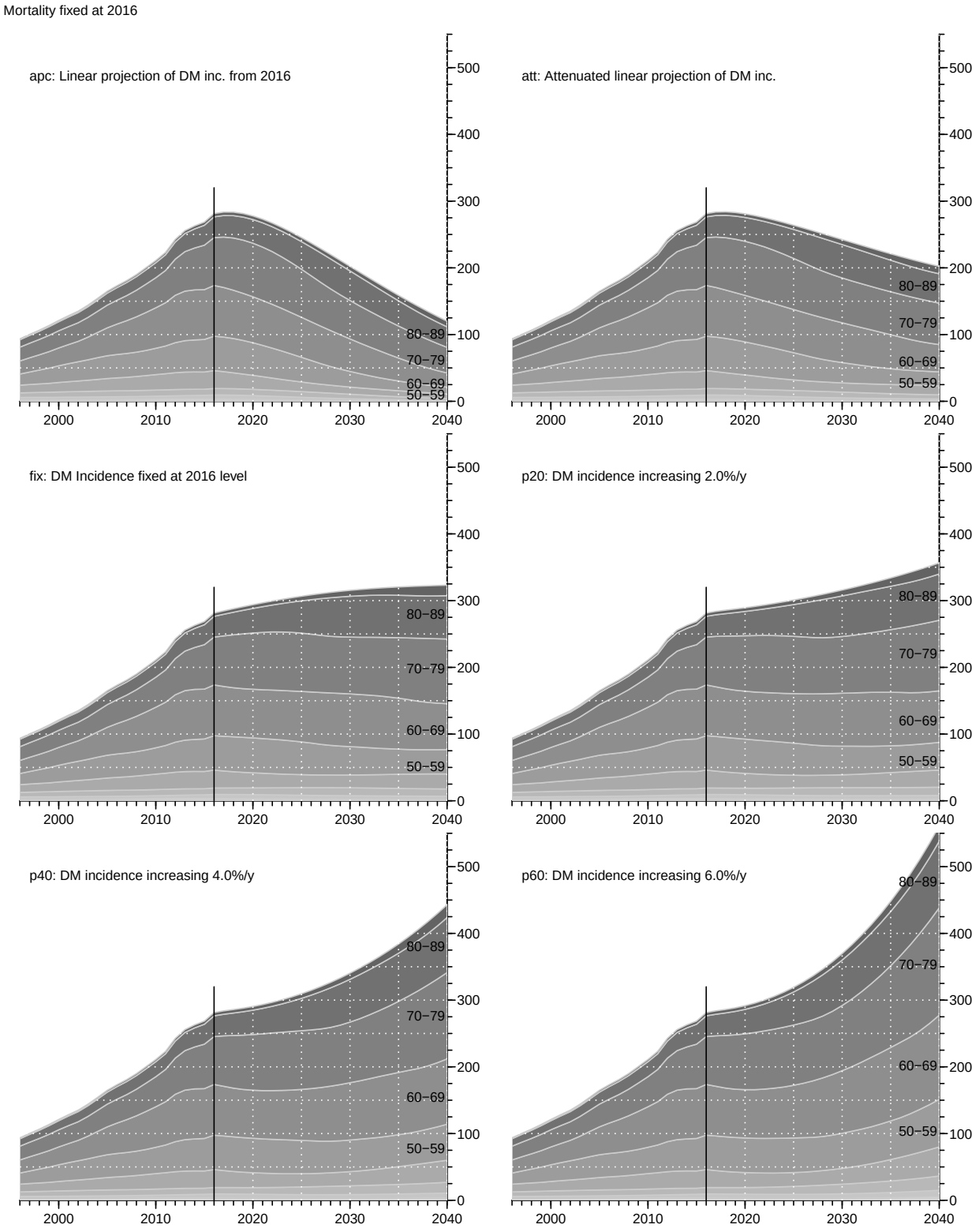

Figure 4.7: *Predicted number of diabetes patients in Denmark under different incidence rate scenarios, using mortality (both for non-DM and DM persons) fixed at the 2016 level.* `./pred-prnum-fix2040`

Chapter 5

Components of prevalence

The purpose of this chapter is to use the estimated transition rates to predict the prevalences during follow-up based on prevalences at 1996 and incidence and mortality rates in the period 1996–2016. This is not *per se* an interesting endeavour, because we have the prevalence data available, but it will serve as an illustration that the rates are adequately modelled and that the degree of approximation is adequate when using an interval length as chosen.

Moreover the methodology used for the forecasting of rates and prevalences in previous chapter where we used different scenarios can also be applied here

```
> library( Epi )
> clear()
> load( file="./data/inits.Rda")
> load( file="./data/allrates.Rda")
> load( file="./data/prevalences.Rda")
> lls()
   name    mode     class    dim                size(Kb)
1  a.pt    numeric  numeric  1200                    9.4
2  fC      function function 1                       2.3
3  fCp     function function 1                       2.3
4  fCtable function function 1                       2.3
5  int     numeric  numeric  1                       0.0
6  Lambda  numeric  array    1200 528 2 8       79,324.7
7  Mu.DM   numeric  array    1200 528 2 5       49,624.5
8  Mu.nD   numeric  array    1200 528 2 5       49,624.5
9  nk.a    numeric  numeric  1                       0.0
10 nk.c    numeric  numeric  1                       0.0
11 nk.p    numeric  numeric  1                       0.0
12 p.pt    numeric  numeric  528                     4.2
13 pr.fit  numeric  array    2 100 21               40.7
14 pr.ini  numeric  array    2 1200 21             480.5
15 pr.obs  numeric  table    2 23 100 3            115.9
16 qn      function function 1                       3.5
17 spread  function function 1                      22.2
18 t.pt    numeric  numeric  529                     4.2
```

We shall use the simulation scheme to predict the course of DM prevalence development in the population under various scenarios of mortality and incidence development. So we set up various structures to hold results and clarify calculations:

`pr.fit` — array of empirical age-specific prevalences at 1.1.1996–1.1.2016, smoothed by natural splines.

`TR` — array of transition probabilities between states Well and DM and Death. Transition probabilities are computed under the 4 different scenarios combining mortality and incidence rates either as they actually developed 1996–2016 or assuming they were constant at the 1996 level. These refer to intervals of length `int` and are therefore labeled on the period dimension by the midpoint of these, a total of 20/`int`.

`prv` — array of predicted prevalences based on the initial prevalences at 1.1.1996 and the transition probabilities as put in `TR`. The scenario dimension refers to the 4 scenarios: "`obs`", "`m-fix`", "`i-fix`" and "`all-f`", but this dimension in the array is expanded by 3 extra levels "`mort`", "`inc`" and "`const`" that are to be filled with the part of the prevalences that are attributable to decrease in mortality, increase in incidence and the disequilibrium between rates and prevalence in 1996. Likewise, the period dimension is expanded by one relative to that in `TR` (replacing `p.pt` (period points) with `t.pt` (time points), since this refer to points in time and not time intervals (periods).

`prn` — array of predicted *number* of DM patients in one-year age classes at the 1 January each year. So the same structure as `prv`, but with substantially fewer entries.

5.1 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.

Note that the structures `Lambda`, `Mu.nD` and `Mu.DM`, contain predicted rates up to 2040 under different scenarios, that we do not need:

```
> str( Lambda )
 num [1:1200, 1:528, 1:2, 1:8] 6.19e-06 6.23e-06 6.27e-06 6.31e-06 6.35e-06 ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.291666666666667"
  ..$ p  : chr [1:528] "1996.04166666667" "1996.125" "1996.20833333333" "1996.29166666667"
  ..$ sex: chr [1:2] "M" "F"
  ..$ mod: chr [1:8] "ap" "apc" "LCa" "att" ...
```

. . . we only need the dates till 2016, and the models `ap` and `apc`:

```
> # 2nd dimension of rates is all the way to 2040, only need to the end
> # of 2015
> dimnames(Lambda)[[2]][239+0:2]
[1] "2015.875"         "2015.95833333333" "2016.04166666667"

> Lambda <- Lambda[,1:240,,1:2]
> Mu.nD  <- Mu.nD [,1:240,,1:2]
> Mu.DM  <- Mu.DM [,1:240,,1:2]
> states <- c("nD","DM")
> TR <- NArray( c( dimnames(Lambda),
+                  list( from = states,
+                          to = states,
+                       scene = c("obs","m-fix","i-fix","all-f" ) ) ) )
> str( TR )
```

```
 logi [1:1200, 1:240, 1:2, 1:2, 1:2, 1:2, 1:4] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 7
  ..$ a    : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.2916666666666
  ..$ p    : chr [1:240] "1996.04166666667" "1996.125" "1996.20833333333" "1996.29166666667
  ..$ sex  : chr [1:2] "M" "F"
  ..$ mod  : chr [1:2] "ap" "apc"
  ..$ from : chr [1:2] "nD" "DM"
  ..$ to   : chr [1:2] "nD" "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 1996 level is trivial, because transition probabilities in that case only depend on age and not on period.

In order to fill `TR`, 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` and `p.pt`, both in steps of `int` (in this case 0.083 year):

```
> int
[1] 0.08333333
```

```
> TR[,,,,"nD","nD","obs"] <- 1 - Lambda - Mu.nD
> TR[,,,,"nD","DM","obs"] <-     Lambda
> TR[,,,,"DM","nD","obs"] <- 0
> TR[,,,,"DM","DM","obs"] <- 1 - Mu.DM
```

Note that we have not included the "Dead" state in the calculations, because we only bother about the *fraction* of diabetes patients in each age class at each time-point. So the probabilities we compute do not sum to 1 within the "from" states; we only compute the fraction of the persons alive that end up being alive at the next time point.

When we fix the mortality or incidence at the 1996 level we just replace the expressions above with expressions where we replace the date dimension by `rep(1,np)`, (where `np` is the number of periods) for either incidence, mortality or both:

```
> ( np <- dim(Lambda)[2] )
  p
240
```

```
> #
> TR[,,,,"nD","nD","m-fix"] <- 1 - Lambda - Mu.nD[,rep(1,np),,]
> TR[,,,,"nD","DM","m-fix"] <-     Lambda
> TR[,,,,"DM","nD","m-fix"] <- 0
> TR[,,,,"DM","DM","m-fix"] <- 1 -          Mu.DM[,rep(1,np),,]
> #
> TR[,,,,"nD","nD","i-fix"] <- 1 - Lambda[,rep(1,np),,] - Mu.nD
> TR[,,,,"nD","DM","i-fix"] <-     Lambda[,rep(1,np),,]
> TR[,,,,"DM","nD","i-fix"] <- 0
> TR[,,,,"DM","DM","i-fix"] <- 1 -                         Mu.DM
> #
> TR[,,,,"nD","nD","all-f"] <- 1 - Lambda[,rep(1,np),,] - Mu.nD[,rep(1,np),,]
> TR[,,,,"nD","DM","all-f"] <-     Lambda[,rep(1,np),,]
> TR[,,,,"DM","nD","all-f"] <- 0
> TR[,,,,"DM","DM","all-f"] <- 1 -                         Mu.DM[,rep(1,np),,]
```

We have now collected the transition probabilities between "Well" and "DM" as well as the probabilities of remaining in each of these, all referring to a duration of `int`:

```
> attr( TR, "label" ) <- "Transition probabilities 1996-2016"
> str( TR )
 num [1:1200, 1:240, 1:2, 1:2, 1:2, 1:2, 1:4] 1 1 1 1 1 ...
 - attr(*, "dimnames")=List of 7
  ..$ a    : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.2916666666666
  ..$ p    : chr [1:240] "1996.04166666667" "1996.125" "1996.20833333333" "1996.29166666667
  ..$ sex  : chr [1:2] "M" "F"
  ..$ mod  : chr [1:2] "ap" "apc"
  ..$ from : chr [1:2] "nD" "DM"
  ..$ to   : chr [1:2] "nD" "DM"
  ..$ scene: chr [1:4] "obs" "m-fix" "i-fix" "all-f"
 - attr(*, "label")= chr "Transition probabilities 1996-2016"
> fCp( length( TR ) )
[1] 18,432,000
> save( TR, file="../data/TRc.Rda" )
```

5.2 Prediction of the observed prevalences

Note that we do not need to predict the population size; we can get away with only predicting 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 (Well,DM), which does not sum to 1 (because of the ones dying), so we must rescale to prevalence age in each step.

First we set up an array to hold the predicted prevalences under different scenarios. Later we shall also compute the fraction of the prevalences that are attributable to trends in mortality and incidence as well as to the non-stationarity of the rates/prevalences as of 1996, so we put in three extra levels of the last dimension, and one extra levels of the period dimension because we want to predict to the end of the last period too (or, to put it differently, we need an extra first level to hold the starting prevalences as of 1.1.1996).

```
> dpr <- c( dimnames(Lambda)[1:4],
+           list( c(dimnames(TR)[["scene"]],"mort","inc","const","org") ) )
> names( dpr )[c(2,5)] <- c("t","what")
> t.pt[240+0:2]
[1] 2015.917 2016.000 2016.083
> dpr[["t"]] <- t.pt[1:241]
> prv <- NArray(dpr)
> str( prv )
 logi [1:1200, 1:241, 1:2, 1:2, 1:8] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 5
  ..$ a   : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.2916666666666
  ..$ t   : chr [1:241] "1996" "1996.08333333333" "1996.16666666667" "1996.25" ...
  ..$ sex : chr [1:2] "M" "F"
  ..$ mod : chr [1:2] "ap" "apc"
  ..$ what: chr [1:8] "obs" "m-fix" "i-fix" "all-f" ...
> fCp( length(prv) )
[1] 9,254,400
```

In order to update the prevalences at 1996-01-01 to subsequent dates, we must first initialize the prevalences as modeled at 1996-01-01, as well as the prevalences at ages 0 (the new-born — set to 0); note that we exploit the column major storage of arrays to get `prv` filled with identical values across the two last dimensions (model and scenario).

```
> # Smoothed prevalences at 1.1.1996 - the starting values
> for( sx in c("M","F") ) prv[,1,sx,,] <- pr.ini[sx,,"1996"]
> # Prevalences at age 0
> prv[1,,"M",,] <- 0
> prv[1,,"F",,] <- 0
> # check we got values in the right spots
> round( ftable( prv[1:3,1:3,,,], row.vars=4:1 )*100, 1 )
```

```
                                          what obs m-fix i-fix all-f mort inc const org
mod sex t                 a
ap  M   1996              0.0416666666666667      0.0   0.0   0.0   0.0  0.0 0.0   0.0 0.0
                          0.125                   0.1   0.1   0.1   0.1  0.1 0.1   0.1 0.1
                          0.208333333333333       0.1   0.1   0.1   0.1  0.1 0.1   0.1 0.1
          1996.08333333333 0.0416666666666667     0.0   0.0   0.0   0.0  0.0 0.0   0.0 0.0
                          0.125                    NA    NA    NA    NA   NA  NA    NA  NA
                          0.208333333333333        NA    NA    NA    NA   NA  NA    NA  NA
          1996.16666666667 0.0416666666666667     0.0   0.0   0.0   0.0  0.0 0.0   0.0 0.0
                          0.125                    NA    NA    NA    NA   NA  NA    NA  NA
                          0.208333333333333        NA    NA    NA    NA   NA  NA    NA  NA
    F   1996              0.0416666666666667      0.0   0.0   0.0   0.0  0.0 0.0   0.0 0.0
                          0.125                   0.1   0.1   0.1   0.1  0.1 0.1   0.1 0.1
                          0.208333333333333       0.1   0.1   0.1   0.1  0.1 0.1   0.1 0.1
          1996.08333333333 0.0416666666666667     0.0   0.0   0.0   0.0  0.0 0.0   0.0 0.0
                          0.125                    NA    NA    NA    NA   NA  NA    NA  NA
                          0.208333333333333        NA    NA    NA    NA   NA  NA    NA  NA
          1996.16666666667 0.0416666666666667     0.0   0.0   0.0   0.0  0.0 0.0   0.0 0.0
                          0.125                    NA    NA    NA    NA   NA  NA    NA  NA
                          0.208333333333333        NA    NA    NA    NA   NA  NA    NA  NA
apc M   1996              0.0416666666666667      0.0   0.0   0.0   0.0  0.0 0.0   0.0 0.0
                          0.125                   0.1   0.1   0.1   0.1  0.1 0.1   0.1 0.1
                          0.208333333333333       0.1   0.1   0.1   0.1  0.1 0.1   0.1 0.1
          1996.08333333333 0.0416666666666667     0.0   0.0   0.0   0.0  0.0 0.0   0.0 0.0
                          0.125                    NA    NA    NA    NA   NA  NA    NA  NA
                          0.208333333333333        NA    NA    NA    NA   NA  NA    NA  NA
          1996.16666666667 0.0416666666666667     0.0   0.0   0.0   0.0  0.0 0.0   0.0 0.0
                          0.125                    NA    NA    NA    NA   NA  NA    NA  NA
                          0.208333333333333        NA    NA    NA    NA   NA  NA    NA  NA
    F   1996              0.0416666666666667      0.0   0.0   0.0   0.0  0.0 0.0   0.0 0.0
                          0.125                   0.1   0.1   0.1   0.1  0.1 0.1   0.1 0.1
                          0.208333333333333       0.1   0.1   0.1   0.1  0.1 0.1   0.1 0.1
          1996.08333333333 0.0416666666666667     0.0   0.0   0.0   0.0  0.0 0.0   0.0 0.0
                          0.125                    NA    NA    NA    NA   NA  NA    NA  NA
                          0.208333333333333        NA    NA    NA    NA   NA  NA    NA  NA
          1996.16666666667 0.0416666666666667     0.0   0.0   0.0   0.0  0.0 0.0   0.0 0.0
                          0.125                    NA    NA    NA    NA   NA  NA    NA  NA
                          0.208333333333333        NA    NA    NA    NA   NA  NA    NA  NA
```

So now we have checked that we have put initial values correctly into `prv`, basically at the period edge (at 1996) and the age edge (at 0). Then we can compute the predicted prevalences under the different scenarios. We take the fraction of the population in age class `ia` at time `ip` that end up as diabetes patients at time `ip+1` (and hence in age class

`ia+1`), and divide by the fraction of all that remain alive, which is the diabetes patients (survivors and new), *plus* those who survive free of diabetes:

```
> system.time(
+ for( ip in 1:(dim(prv)[2]-1) )
+ for( ia in 1:(dim(prv)[1]-1) )
+ prv[ia+1,ip+1,,,1:4] <-
+   (    prv[ia,ip,,,1:4]  * TR[ia,ip,,,"DM","DM",]
+    +(1-prv[ia,ip,,,1:4]) * TR[ia,ip,,,"nD","DM",] ) /
+   (    prv[ia,ip,,,1:4]  * TR[ia,ip,,,"DM","DM",]
+    +(1-prv[ia,ip,,,1:4]) * TR[ia,ip,,,"nD","DM",]
+    +(1-prv[ia,ip,,,1:4]) * TR[ia,ip,,,"nD","nD",] )
+             )
   user  system elapsed
 10.040   0.001  10.039
```

Note that the reason that the last dimension, `scene`, is explicitly mentioned in the array `prv` is because the length of this dimension in `prv` is 7, but the corresponding in `TR` only 4 — recall that `prv` also has thee extra levels to provide for the estimated part of the prevalences attributable to mortality change, incidence changes, and non-equilibrium at 1996.

Moreover, note that the code above is particularly simple because we only need to compute the prevalence at the next date and age. If we had had a more elaborate model with, say complications states, the calculations in the loop would have been a matrix-multiplication updating the state-distribution, but this simplification would have been at the expense of another three loop-levels, namely over the the three last dimensions of the `prv` array — the matrix machinery only operates on 2-dimensional structures, matrices. (Well, maybe some `apply` construction could be set up).

We can then show a few of the predicted prevalences in (%):

```
> round( prv[900+1:4,1+(0:3)*24,1,1,1,drop=F]*100, 3 )
, , sex = M, mod = ap, what = obs

                  t
a                   1996  1998  2000  2002
  75.0416666666667 6.263 7.041 7.969 9.203
  75.125           6.272 7.056 7.972 9.205
  75.2083333333333 6.281 7.071 7.977 9.206
  75.2916666666667 6.289 7.085 7.981 9.207


> save( a.pt, prv, file="../data/prv.Rda" )
> load(            file="../data/prv.Rda" )
```

5.2.1 Checking the prediction

With this initial prediction in place we can now check whether we have made a reasonable approximation to the observed prevalences at 1.1.2016.

In the array `prv` are all the prevalences as predicted from the prevalence in 1996 using the estimated incidences and mortalities; predicted at intervals of `inc` whereas we have the smoothed empirical prevalences at 1 January 1996,...,2016 in the array `pr.fit`:

```
> str( pr.fit )
 num [1:2, 1:100, 1:21] 0.000573 0.000562 0.000632 0.000614 0.000697 ...
 - attr(*, "dimnames")=List of 3
  ..$ sex: chr [1:2] "M" "F"
  ..$ A  : chr [1:100] "0.5" "1.5" "2.5" "3.5" ...
  ..$ P  : chr [1:21] "1996" "1997" "1998" "1999" ...
```

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

```
> ( wh <- paste(seq(1996,2016,5)) )
[1] "1996" "2001" "2006" "2011" "2016"
> par( mfrow=c(1,2), mar=c(0,0,0,0), oma=c(3,4,1,1), mgp=c(3,1,0)/1.6,
+      las=1, bty="n", lend="butt" )
> matplot( a.pt, pr.ini["M",,wh]*100,
+          xlim=c(10,95), ylim=c(0,20), yaxs="i",
+          xlab="Age", ylab="Prevalence (%)",
+          type="l", col="blue", lty=1, lwd=2 )
> axis( side=1, at=1:9*10, labels=NA )
> axis( side=1, at=seq(10,95,5), labels=NA, tcl=-0.3 )
> matlines( a.pt, prv[,wh,"M","apc","obs"]*100,
+          type="l", col="blue", lty="32", lwd=4 )
> matplot( a.pt, pr.ini["F",,wh]*100,
+          xlim=c(10,95), ylim=c(0,20), yaxs="i",
+          xlab="Age", ylab="", yaxt="n",
+          type="l", col="red", lty=1, lwd=2 )
> matlines( a.pt, prv[,wh,"F","apc","obs"]*100,
+           type="l", col="red", lty="32", lwd=3 )
> axis( side=1, at=1:9*10, labels=NA )
> mtext( "Prevalence of DM (%)", side=2, line=2, las=0, outer=TRUE )
> mtext( "Age", side=1, line=2, las=0, outer=TRUE )

> par( mfrow=c(1,2), mar=c(0,0,0,0), oma=c(3,4,1,1), mgp=c(3,1,0)/1.6,
+      las=1, bty="n" )
> matplot( a.pt, cbind(prv[,wh,"M","apc","obs"],
+                      prv[,wh,"M","ap" ,"obs"],
+                      pr.ini["M",,wh])*100,
+          xlim=c(10,95), ylim=c(0,20), yaxs="i",
+          xlab="Age", ylab="Prevalence (%)",
+          type="l", col="black", lty=rep(c(0,1),c(10,5)), lwd=2 )
> matlines( a.pt, cbind(prv[,wh,"M","ap" ,"obs"],
+                       prv[,wh,"M","apc","obs"])*100,
+           type="l", col="blue", lty=rep(c("12","42"),c(5,5)), lwd=4 )
> axis( side=1, at=1:9*10, labels=NA )
> matplot( a.pt, cbind(prv[,wh,"F","apc","obs"],
+                      prv[,wh,"F","ap" ,"obs"],
+                      pr.ini["F",,wh])*100,
+          xlim=c(10,95), ylim=c(0,20), yaxs="i",
+          xlab="Age", ylab="", yaxt="n",
+          type="l", col="black", lty=rep(c(0,1),c(10,5)), lwd=2 )
> matlines( a.pt, cbind(prv[,wh,"F","ap" ,"obs"],
+                       prv[,wh,"F","apc","obs"])*100,
+           type="l", col="red", lty=rep(c("12","42"),c(5,5)), lwd=4 )
> axis( side=1, at=1:9*10, labels=NA )
> mtext( "Prevalence of DM (%)", side=2, line=2, las=0, outer=TRUE )
> mtext( "Age", side=1, line=2, las=0, outer=TRUE )
```

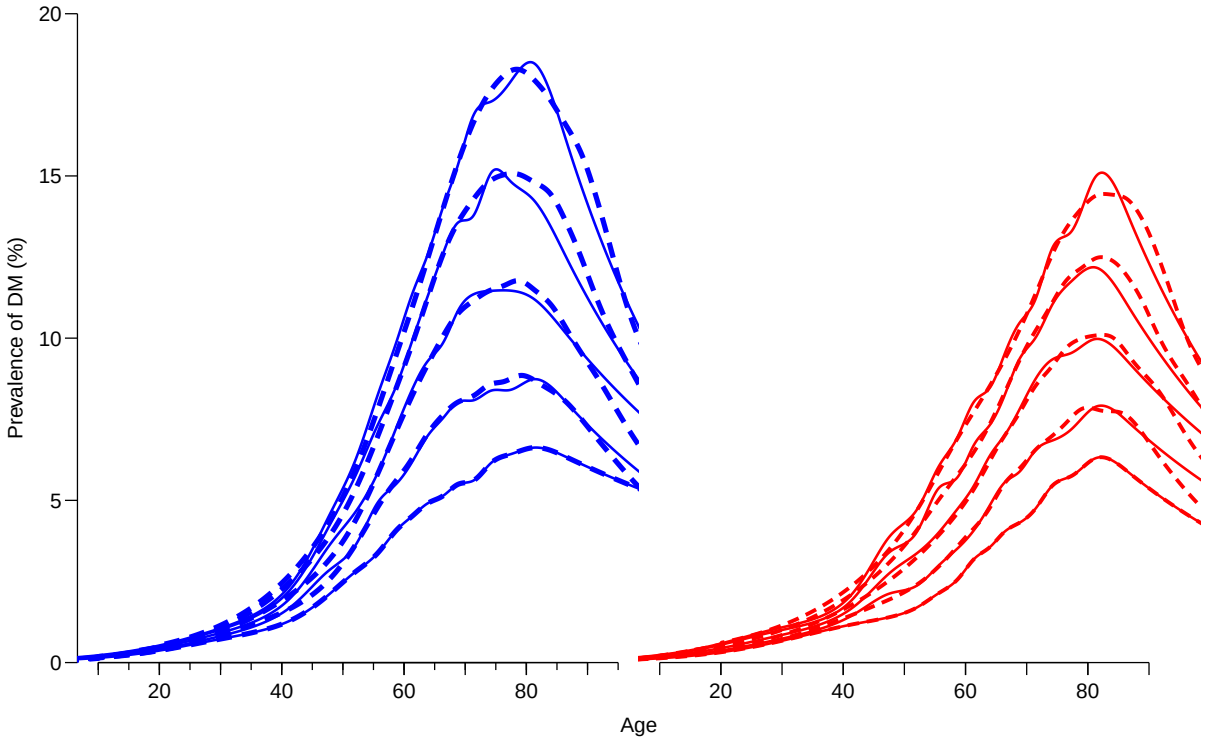

Figure 5.1: *The empirical (smoothed) prevalences at 1 January 1996, 2001,...,2016 (full lines) and the predicted prevalences from using the estimated incidence and mortality rates from an APC-model, computed at 1 month intervals by age and calendar time.* `./comp-check`

Since the APC-models for rates provide a better fit (figure **??**), we shall use these in the reporting of the different scenarios.

We now compare the predicted prevalences under the four scenarios at 2016-01-01:

```
> str( prv )
 num [1:1200, 1:241, 1:2, 1:2, 1:8] 0 0.000553 0.000557 0.000562 0.000566 ...
 - attr(*, "dimnames")=List of 5
  ..$ a   : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.29166666666666
  ..$ t   : chr [1:241] "1996" "1996.08333333333" "1996.16666666667" "1996.25" ...
  ..$ sex : chr [1:2] "M" "F"
  ..$ mod : chr [1:2] "ap" "apc"
  ..$ what: chr [1:8] "obs" "m-fix" "i-fix" "all-f" ...
```

```
> dimnames(prv)[["t"]][np <- 241]
```

```
[1] "2016"
```

```
> prv[floor(dim(prv)[1]/1.5)+1:5,np,"M","apc",]*100
```

```
                  what
a                      obs    m-fix    i-fix    all-f mort inc const org
  66.7083333333333 14.19621 13.01884 10.70683 9.757011   NA  NA    NA  NA
  66.7916666666667 14.24620 13.05811 10.74700 9.788210   NA  NA    NA  NA
  66.875           14.29606 13.09718 10.78704 9.819227   NA  NA    NA  NA
  66.9583333333333 14.34577 13.13605 10.82695 9.850060   NA  NA    NA  NA
  67.0416666666667 14.39533 13.17470 10.86673 9.880705   NA  NA    NA  NA
```

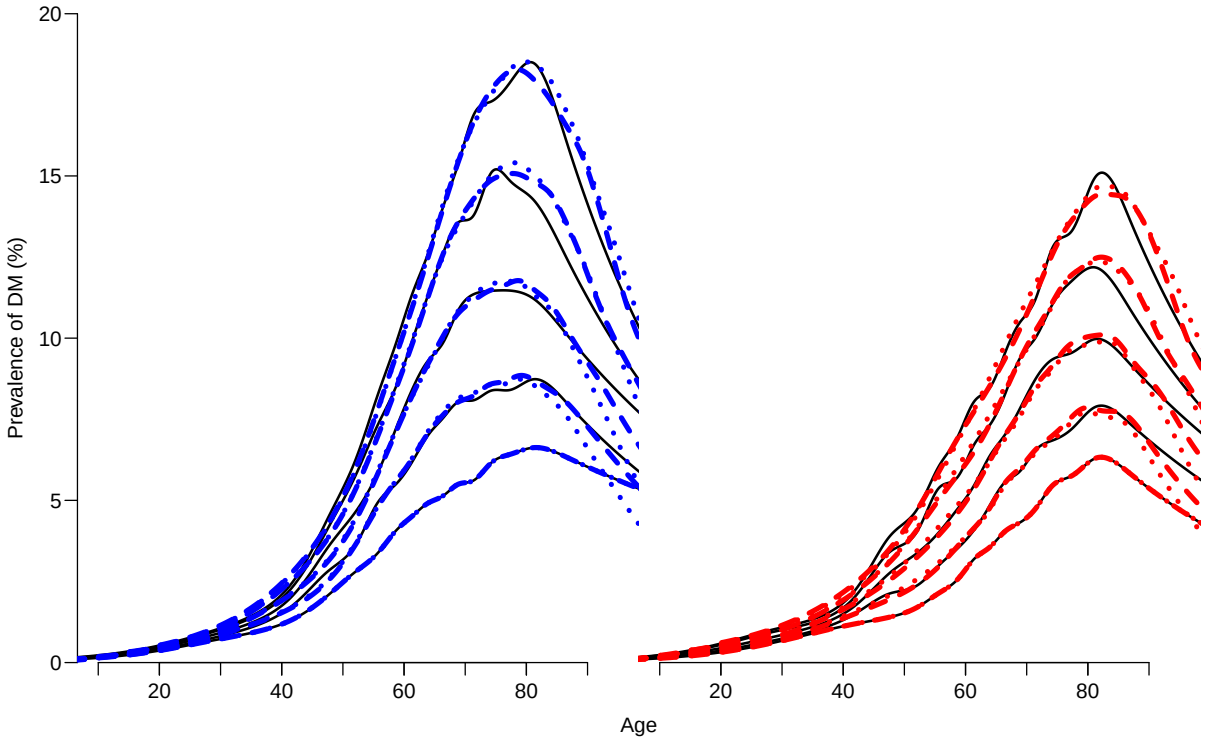

Figure 5.2: *Plot of observed (full lines) and predicted prevalences in 2001, using simple age-period-models (dotted lines) or age-period-cohort models (broken lines). The broken lines give a slightly better approximation to the smoothed empirical rates (black lines), but the differences are generally quite small.* `./comp-check-x`

```
> par( mfrow=c(1,2), mar=c(0,0,0,0), oma=c(3,4,1,1),
+      mgp=c(3,1,0)/1.6, las=1, bty="n" )
> lpl <- function( sx, cl )
+ matplot( a.pt, cbind(prv[,np,sx,"apc",c("obs","m-fix","i-fix","all-f")],
+                       prv[, 1,sx,"apc",1  ])*100,
+          xlim=c(20,90), ylim=c(0,20), yaxs="i", yaxt="n",
+          xlab="Age", ylab="Prevalence (%)",
+          type="l", lty=c("solid","22","33")[c(1,2,1,3,1)],
+          col=cl,   lwd=c(5,5,3,3,1) )
> lpl( "M", "blue" )
> axis( side=2 )
> axis( side=2, at=0:18, labels=NA, tcl=-0.3)
> lpl( "F", "red" )
> dimnames(prv)[[5]]
[1] "obs"   "m-fix" "i-fix" "all-f" "mort"  "inc"   "const" "org"
```

Thus, in figure **??**, the difference between the thick fill and the thick broken is the effect of declining mortality — the difference between the thick and the thin full lines is the effect of increasing incidence rates. Finally, the difference between the thin broken line and the very thin full line (lowest) is the effect of the imbalance between incidence and mortality rates in 1995 (disequilibrium).

```
> scen <- c("Mort changes, Inc changes",
+           "Mort 1996, Inc changes",
+           "Mort changes, Inc 1996",
```

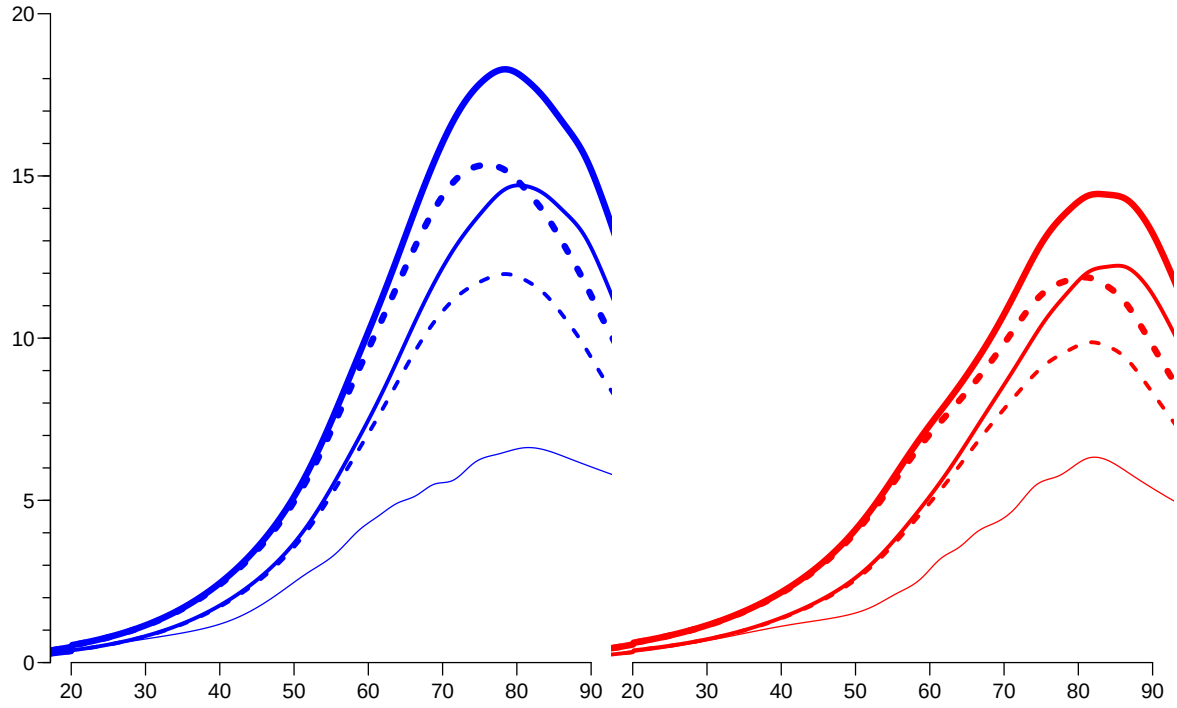

Figure 5.3: *The predicted prevalences under different scenarios:*
Full lines: Mortality rates evolve as observed.
Broken lines: Mortality rates remain as 1996.
Thick lines: Incidence rates evolve as observed.
Thin lines: Incidence rates remain as in 1996.
The very thin lines lowest in the two displays are the observed age-specific prevalences in 1996. ./comp-causes

```
+           "Mort 1996, Inc 1996",
+           "Prevalence 1996")
> c.a <- dimnames(prv)[[1]][floor(dim(prv)[1]/1.34)]
> n.a <- as.numeric( c.a )
> np <- 241
> hts <- c(prv[c.a,np,"M","apc",1:4],
+          prv[c.a, 1,"M","apc",1])*100
> cau.exp <-
+ function( wh=1:5, fill=FALSE )
+ {
+ pdf( paste( "comp-DMpr-", paste(wh,collapse=""), if( fill ) "F",
+             ".pdf", sep="" ), height=11*9/16, width=11 )
+ par( mfrow=c(1,2), mar=c(0,0,0,0), oma=c(3,4,1,1),
+      mgp=c(3,1,0)/1.6, las=1, bty="n" )
+ matplot( a.pt, cbind(prv[,np,"M","apc",],prv[,1,"M","apc",1])*100,
+          xlim=c(20,90), ylim=c(0,20), xlab="Age",
+          ylab="Prevalence (%)", yaxs="i",
+          type="l", lty=rep(c(1,0),2), lwd=c(4,4,2,2,0)+1, col="blue" )
+ axis( side=2, at=0:18, labels=NA, tcl=-0.3 )
+ mtext( "Prevalence (%)", side=2, line=2.5, outer=T, las=0 )
+ matlines( a.pt, prv[,np,"M","apc",]*100,
+          type="l", lty=rep(c("11","22"),2), lwd=c(4,4,2,2)+1, col="blue" )
```

comp.tex

```
+ matlines( a.pt, prv[,1,"M","apc",]*100,
+           type="l", lty=1, lwd=1, col="blue" )
+ text( rep(20,5)[wh], hts[wh], scen[wh], adj=0, col=gray(0.3), cex=1.4 )
+ for( i in 1:15 )
+ arrows( (20.20+strwidth(scen,cex=1.4))[wh], hts[wh],
+                              rep(n.a,5)[wh], hts[wh],
+         col=gray(0.3), angle=i, lwd=2 )
+ if( fill ) polygon( c(a.pt,rev(a.pt)),
+                     c(prv[,np,"M","apc",wh[1]],
+                   rev(prv[,if(wh[2]==5) 1 else np,"M","apc",wh[2]]))*100,
+                     col=rgb(0,0,1,0.3), border="transparent" )
+ matplot( a.pt, cbind(prv[,np,"F","apc",],prv[,1,"F","apc",1])*100,
+          xlim=c(20,90), ylim=c(0,20), xlab="Age", yaxt="n", yaxs="i",
+          type="l", lty=rep(c(1,0),2), lwd=c(4,4,2,2,0)+1, col="red" )
+ matlines( a.pt, prv[,np,"F","apc",]*100,
+          type="l", lty=rep(c("11","22"),2), lwd=c(4,4,2,2)+1, col="red" )
+ matlines( a.pt, prv[,1,"F","apc",]*100, type="l", lty=1, lwd=1, col="red" )
+ if( fill ) polygon( c(a.pt,rev(a.pt)),
+                     c(prv[,np,"F","apc",wh[1]],
+                   rev(prv[,if(wh[2]==5) 1 else np,"F","apc",wh[2]]))*100,
+                     col=rgb(1,0,0,0.3), border="transparent" )
+ dev.off()
+ }
> cau.exp(1:5)
null device
          1


> for( ff in c(FALSE,TRUE) )
+    {
+ cau.exp(1:2,fill=ff)
+ cau.exp(3:4,fill=ff)
+ cau.exp(4:5,fill=ff)
+ cau.exp(c(1,3),fill=ff)
+ cau.exp(c(2,4),fill=ff)
+    }
```

Figure 5.3 shows the predicted prevalences under 4 different scenarios compared to the observed prevalences as of 1.1.1996.

5.3 How much is attributable to what?

We can compute how much of the age-specific prevalences that are attributable to mortality changes and how much to changes in incidence rates.

The effect of mortality decline can be computed either as the difference between "obs" and "m-fix" or as the difference between "i-fix" and "all-f"; the latter is the mortality effect in a scenario with stable (fixed) incidence rates. There is of course no guarantee that these two quantities are the same.

Similarly the effect of incidence increase can be computed either as the difference between "obs" and "i-fix" or as the difference between "m-fix" and "all-f"; the latter the incidence effect as it would have been if in a scenario with stable (fixed) incidence rates.

Hence we explore how different these quantities are:

```
> par( mfrow=c(1,2), mar=c(3,0,0,0), oma=c(0,4,1,1),
+      mgp=c(3,1,0)/1.6, las=1, bty="n" )
> matplot( a.pt, cbind( prv[,np,"M","apc","obs"  ]-
+                       prv[,np,"M","apc","m-fix"],
+                       prv[,np,"M","apc","i-fix"]-
+                       prv[,np,"M","apc","all-f"] )*100,
+          xlim=c(20,90), ylim=c(0,4.5), xlab="Age", ylab="Prevalence (%)",
+          type="l", lty=1, lwd=c(4,2)+1, col="blue", yaxs="i" )
> axis( side=2, at=0:14/2, labels=NA, tcl=-0.3 )
> mtext( "Prevalence difference (%)", side=2, line=2.5, outer=T, las=0 )
> matlines(a.pt, cbind( prv[,np,"M","apc","obs"  ]-
+                       prv[,np,"M","apc","i-fix"],
+                       prv[,np,"M","apc","m-fix"]-
+                       prv[,np,"M","apc","all-f"] )*100,
+          type="l", lty="22", lwd=c(4,2)+1, col="blue" )
> matplot( a.pt, cbind( prv[,np,"F","apc","obs"  ]-
+                       prv[,np,"F","apc","m-fix"],
+                       prv[,np,"F","apc","i-fix"]-
+                       prv[,np,"F","apc","all-f"] )*100,
+          xlim=c(20,90), ylim=c(0,4.5), xlab="Age", yaxt="n", yaxs="i",
+          type="l", lty=1, lwd=c(4,2)+1, col="red" )
> matlines(a.pt, cbind( prv[,np,"F","apc","obs"  ]-
+                       prv[,np,"F","apc","i-fix"],
+                       prv[,np,"F","apc","m-fix"]-
+                       prv[,np,"F","apc","all-f"] )*100,
+          type="l", lty="22", lwd=c(4,2)+1, col="red" )
```

From figure 5.4 we see that the two different ways of computing the contribution give pretty much the same results — the differences never exceed 0.5%. Therefore, if we want to attribute fractions of the prevalence in 2016 to decreasing mortality and increasing incidence respectively, we would want two measures that had a sum equal the the difference between the scenario with observed mortality and incidence rates ("`obs`"), and the scenario with rates fixed to those from 1996 ("`all-f`"). This is obtained by taking the average of the two curves in each scenario.

The thin lines at the bottom of figure 5.3 represent the prevalence at 1.1.1996, so it is pretty clear that the incidence an mortality rates as observed by 1996 did not provide for at steady state.

So basically we can subdivide the prevalence at any point in time into 4 components:

1. the "inherited" prevalences from 1996.
2. the prevalence attributable to rates of mortality and incidence as of 1996 — the "epidemiological disequilibrium" as of 1996.
3. the prevalence attributable to the *change* in the incidence rates.
4. the prevalence attributable to the *change* in the mortality rates.

So we now fill out the remaining 3 dimensions of `prv` according to this:

```
> prv[,,,,"mort" ] <- ( prv[,,,,"obs"  ]-prv[,,,,"m-fix"] +
+                       prv[,,,,"i-fix"]-prv[,,,,"all-f"] ) / 2
> prv[,,,,"inc"  ] <- ( prv[,,,,"obs"  ]-prv[,,,,"i-fix"] +
+                       prv[,,,,"m-fix"]-prv[,,,,"all-f"] ) / 2
> prv[,,,,"const"] <-   prv[,,,,"all-f"]-prv[,rep(1,dim(prv)[2]),,,"obs"]
```

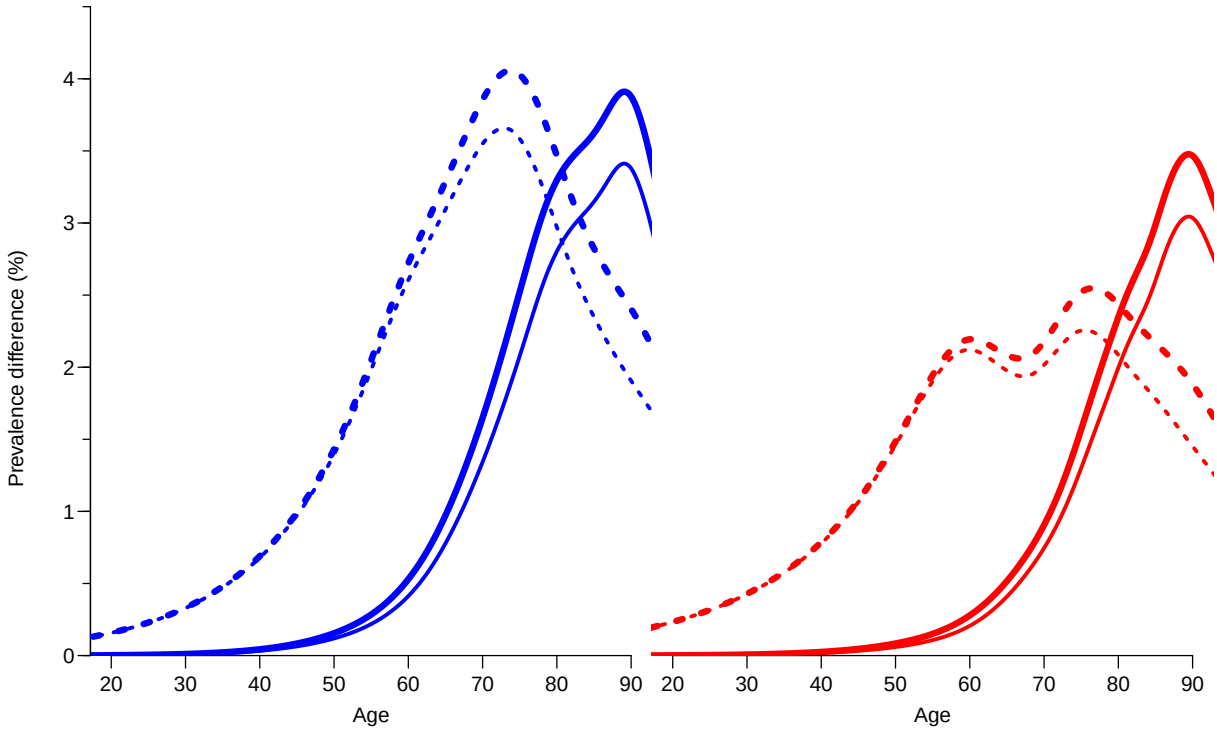

Figure 5.4: *Suggested contributions to age-specific prevalences from increasing mortalities over the period 1996–2016; the thick lines are obtained by subtracting the prediction based on fixing one rate from the one using the observed rates; thin lines based on subtracting the prediction based on fixing both rates from that where only one is fixed. Full lines are for differences attributable to changes in mortality rates, broken lines are for changes attributable to changes in incidence rates.* `./comp-attrib`

The component `obs` of `prv` is the total prevalence (from the modeling), whereas `const`, `inc` and `mort` are the three components. It would be more logical if the `obs` were the last component:

```
> prv[,,,,"org"] <-
+ prv[,,,,"obs"] - ( prv[,,,,"mort"] +
+                    prv[,,,,"inc"]  +
+                    prv[,,,,"const"] )
```

The components `obs`, `const`, `inc` and `mort` now together make up the total prevalence of diabetes (as proportions) for a given combinations of sex, age and date. Thus we can show these for each of the 21 dates 1 January 1996,. . . ,2016.

First we define a function to make the component plots, and then we can plot the resulting development for men and women, for convenience we also put the latter in a function.

```
> poly.parts <-
+ function( x, crv, col, xlim, ylim, txt="" )
+ {
+ crv <- t(apply(cbind(0,crv),1,cumsum))
+ matplot( x, crv, type="n", xaxt="n", yaxt="n", xlab="", ylab="",
+          xlim=xlim, ylim=ylim, yaxs="i", bty="n" )
+ for( i in 2:ncol(crv) )
```

`comp.tex`

```
+ polygon( c(x,rev(x)), c(crv[,i],rev(crv[,i-1])),
+          col=col[i-1], border=col[i-1])
+ text( par("usr")[1:2]%*%c(0.1,0.9),
+       par("usr")[3:4]%*%c(0.9,0.1), txt, adj=c(1,0), font=2 )
+ }
> one.comp <-
+ function( sex, clr )
+ {
+ par( mfrow=c(3,6), mar=c(0,0,0,0), oma=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
+ yn <- round(seq(1,dim(prv)[2],,21))
+ for(j in 1:21)
+ {
+ poly.parts( a.pt,
+             prv[,yn[j],sex,"apc",c("org","const","inc","mort")]*100,
+             col=clr, xlim=c(20,90), ylim=c(0,25),
+             txt=dimnames(prv)[[2]][yn[j]] )
+ abline(h=0)
+ if( j==1 ) text( rep(25,3), c(13,15,17)+0.5,
+                  c("Imbalance","Incidence","Mortality"),
+                  col=clr[2:4], font=2, adj=0, cex=1.2 )
+ if( j %in% c(1,7,13) ) axis( side=2 )
+ if( j %in% 13:18 ) axis( side=1 )
+ mtext( "Age", side=1, outer=TRUE, line=1.5, font=1, las=0 )
+ mtext( "Prevalence of DM", side=2, outer=TRUE, line=1.5, font=1, las=0 )
+ }
+ }
```

With these functions in place we make a graph of the prevalence of DM by 2016 subdivided by component of change since 1996:

```
> par( mfrow=c(1,2), mar=c(1,0,1,0), mgp=c(3,1,0)/1.6, las=1,
+      oma=c(2,3,0,1), bty="n" )
> clr <- rgb(c(3,2,1.5,0)/3,c(3,2,1.5,0)/3,1)
> poly.parts( a.pt, prv[,"2016","M","apc",c("org","const","inc","mort")]*100,
+             col=clr, xlim=c(20,90), ylim=c(0,20) )
> abline(h=0:22,v=2:9*10,col=gray(0.9))
> axis( side=1 )
> axis( side=2 )
> axis( side=2, at=1:18, labels=NA, tcl=-0.3 )
> text( rep(25,3), 17:19-1.5,
+       c("Imbalance","Incidence","Mortality"),
+       col=clr[2:4], font=2, adj=0, cex=1.0 )
> clr <- rgb(1,c(3,2,1.5,0)/3,c(3,2,1.5,0)/3)
> poly.parts( a.pt, prv[,"2016","F","apc",c("org","const","inc","mort")]*100,
+             col=clr, xlim=c(20,90), ylim=c(0,20) )
> abline(h=0:22,v=2:9*10,col=gray(0.9))
> axis( side=1 )
> text( rep(25,3), 17:19-1.5,
+       c("Imbalance","Incidence","Mortality"),
+       col=clr[2:4], font=2, adj=0, cex=1.0 )
> mtext( "Age", side=1, outer=TRUE, line=0.8, font=1, las=0 )
> mtext( "Prevalence of DM (%)", side=2, outer=TRUE, line=2, font=1, las=0 )
```

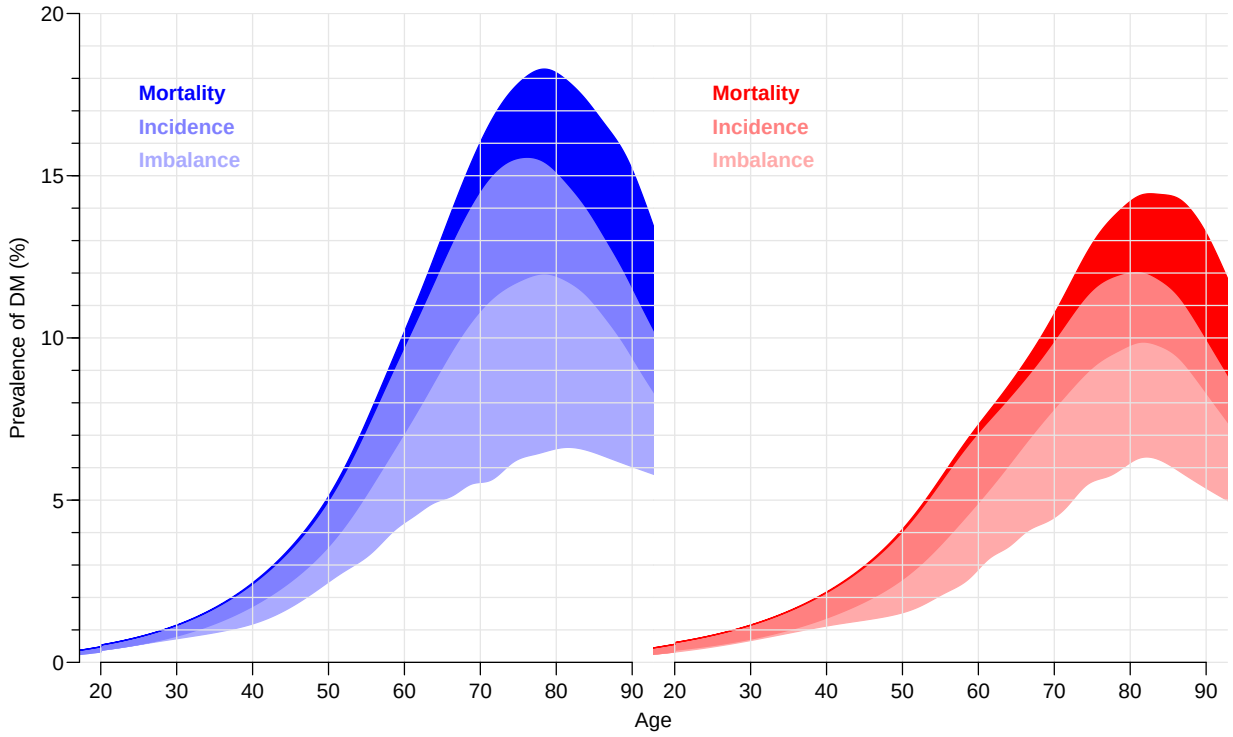

Figure 5.5: *Changes in predicted age-specific prevalences of DM in Denmark 2016 among men (blue) and women (red), partitioned by the contribution from incidence and mortality decrease, in since 1996 and from disequilibrium between incidence and mortality at 1996 ("Imbalance").* `./comp-prev-comp-2016`

5.4 Number of diabetes patients in Denmark

In the previous section we only looked at the age-specific prevalences, because these are the quantities that are driven by the incidence and mortality rates. However, it is also of interest to see how the actual number of diabetes patients would have looked under the different scenarios, specifically how the *number* of the current patients that can be attributed to the various components.

Also note that since the previous calculations were for age-specific prevalences we have a constant reference as the prevalences at 1996, but when we multiply by the population figures we would of course see differences in numbers and age-distribution of the diabetes population even if the age-specific prevalences were unchanged.

To show these effects we set up an array `prn` with `structure` (but not extent) as `prv` to hold the number of diabetes patients by category, assuming the age-distribution in the population to be as actually observed (that is as extracted from Statistics Denmark). However `prn` will have 100 age-classes rather than 1200 (100/`int`), and only 21 dates: `prv`.

```
> dn <- dimnames(prv)
> dn[[1]] <- 0:99
> dn[[2]] <- 1996:2016
> dn[[5]] <- dn[[5]][5:8]
> prn <- NArray( dn )
> str(prv)
 num [1:1200, 1:241, 1:2, 1:2, 1:8] 0 0.000553 0.000557 0.000562 0.000566 ...
 - attr(*, "dimnames")=List of 5
  ..$ a   : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.29166666666666
  ..$ t   : chr [1:241] "1996" "1996.08333333333" "1996.16666666667" "1996.25" ...
  ..$ sex : chr [1:2] "M" "F"
  ..$ mod : chr [1:2] "ap" "apc"
  ..$ what: chr [1:8] "obs" "m-fix" "i-fix" "all-f" ...
> str(prn)
 logi [1:100, 1:21, 1:2, 1:2, 1:4] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 5
  ..$ a   : chr [1:100] "0" "1" "2" "3" ...
  ..$ t   : chr [1:21] "1996" "1997" "1998" "1999" ...
  ..$ sex : chr [1:2] "M" "F"
  ..$ mod : chr [1:2] "ap" "apc"
  ..$ what: chr [1:4] "mort" "inc" "const" "org"
```

In order to fill in the numbers we use the estimated age-specific prevalences at 1st January each year, that is at the dates 1996-01-01,...,2016-01-01 in the entries along the `t`-dimension of `prv`. Moreover, we want the prevalences for 1 year age class rather than age-classes of length `int`. So we take the average prevalences from `prv` over each one-year age-interval. The vectors `wh.a` and `wh.p` will hold the number of the age and period classes from `prv` which have the desired prevalences (as proportions) that we will use for multiplication with the population figures:

```
> prv.p <- as.numeric( dimnames(prv)[["t"]] )
> prn.p <- as.numeric( dimnames(prn)[["t"]] )
> ( wh.p <- match( prn.p, prv.p ) )
 [1]   1  13  25  37  49  61  73  85  97 109 121 133 145 157 169 181 193 205 217 229 241
> prv <- pmax( prv, 0 )
```

Now `wh.p` contains the numbers on the date dimension in `prv.p` that we shall use. But we want the prevalences for 1 year age classes rather than for age-classes of length `int`. So we take the average prevalences from `prv.p` over each one-year age-interval.

This is really just a simple matrix operation; take a diagonal matrix of 1/12 (well, `int`), and repeat each column 12 (1/`int`) times:

```
> dd <- diag(100)[,rep(1:100,each=1/int)]*int
> dim(dd)
[1]  100 1200
```

Pre-multiplying this 100×1200 matrix to the 1200 (= 100/`int`) age-specific prevalences gives the average prevalences in the 100 1-year age-classes. So we just select the dates at which we want the prevalences:

```
> for( sx in dimnames(prn)[[3]] )
+ for( im in dimnames(prn)[[4]] )
+ for( mm in dimnames(prn)[[5]] )
+ prn[,,sx,im,mm] <- dd %*% prv[,wh.p,sx,im,mm]
```

Now `prn` contains the estimated prevalences (as fractions) for 100 age classes and the 21 dates. We need to multiply these prevalences by the population figures at these times. The population figures are in `pop`

```
> load( "../data/pop.Rda" )
> dimnames( pop )[[2]]
 [1] "1995" "1996" "1997" "1998" "1999" "2000" "2001" "2002" "2003" "2004" "2005" "2006"
[13] "2007" "2008" "2009" "2010" "2011" "2012" "2013" "2014" "2015" "2016" "2017" "2018"
[25] "2019" "2020" "2021" "2022" "2023" "2024" "2025" "2026" "2027" "2028" "2029" "2030"
[37] "2031" "2032" "2033" "2034" "2035" "2036" "2037" "2038" "2039" "2040"
> str( pop[,2:22,] )
 'table' int [1:100, 1:21, 1:2] 36054 36007 34960 35352 33643 33281 32202 31123 29823 29405
 - attr(*, "dimnames")=List of 3
  ..$ A  : chr [1:100] "0" "1" "2" "3" ...
  ..$ P  : chr [1:21] "1996" "1997" "1998" "1999" ...
  ..$ sex: chr [1:2] "M" "F"
> str( prn[,,,1,1] )
 num [1:100, 1:21, 1:2] 0 0 0 0 0 0 0 0 0 0 ...
 - attr(*, "dimnames")=List of 3
  ..$ a  : chr [1:100] "0" "1" "2" "3" ...
  ..$ t  : chr [1:21] "1996" "1997" "1998" "1999" ...
  ..$ sex: chr [1:2] "M" "F"
```

Hence we multiply the population figures by the prevalences to get the total number of persons in the different groups:

```
> for( i in dimnames(prn)[[4]] )
+ for( j in dimnames(prn)[[5]] )
+ prn[,,,i,j] <- prn[,,,i,j] * pop[,2:22,]
> dmp <- apply( prn[,,,"apc",], 1:3, sum )
> str( dmp )
 num [1:100, 1:21, 1:2] 19 22.8 24.4 27.2 28.5 ...
 - attr(*, "dimnames")=List of 3
  ..$ a  : chr [1:100] "0" "1" "2" "3" ...
  ..$ t  : chr [1:21] "1996" "1997" "1998" "1999" ...
  ..$ sex: chr [1:2] "M" "F"
```

First we draw a simple pyramid of the age-distribution of diabetes patients in Denmark:

```
> # Note: This uses the undocumented feature that if the first
> # number in a column is negative this is taken as the left endpoint of
> # the bar. So c(-m,m,f) is a bar starting at -m, and a division at
> # -m+m(=0) and an upper end at -m+m-f. Coloring is from the top, that is
> # the part stretching from -m+m to -m+m+f get the first color
> pp <- "2016"
> oo <- c("mort","inc","const","org")
> lim <- 6
> clr <- c("red","blue")
> draw.dmp <-
+ function(pp)
+ {
+ par( mar=c(3,3,3,0), mgp=c(3,1,0)/1.6, las=1 )
+ barplot( height=t( cbind( -dmp[,pp,"M"],
+                           dmp[,pp,"M"],
+                           dmp[,pp,"F"] ) )/ 1000,
+          horiz=TRUE, col=clr,
+          border="transparent",space=0,axes=FALSE,
+          names.arg=rep("",dim(prn)[1]),
+          xlim=c(-1,1)*lim*1.05,
+          xlab="Persons in 1 year class (1000s)",ylab="Age")
+ abline(h=seq(0,100,5),
+        v=seq(-lim,lim,0.5),
+        col="white")
+ axis( side=1, at=seq(-lim,lim,1), labels=abs(seq(-lim,lim,1)) )
+ axis( side=2, at=seq(0,100,20) )
+ axis( side=2, at=seq(0,100,5), labels=NA, tcl=-0.3 )
+ mtext( pp, at=-lim, adj=1.4, cex=1.3, font=1 )
+ mtext( formatC(sum(dmp[,pp,"M"]),0,format="f",big.mark=","),
+        at=-1, col="blue", line=0, cex=0.99 )
+ mtext( formatC(sum(dmp[,pp,"F"]),0,format="f",big.mark=","),
+        at= 1, col="red" , line=0, cex=0.99 )
+ mtext( "N", at=0, line=0, cex=0.99 )
+ }
> pdf( "comp-obs-film.pdf", width=8, height=6 )
> for( pp in paste(1996:2016) ) draw.dmp(pp)
> dev.off()
null device
          1

> for( pp in paste(1996:2016) )
+ {
+ pdf( paste("comp-obs-", pp, ".pdf", sep=""), width=8, height=6 )
+ draw.dmp(pp)
+ dev.off()
+ }
```

Now we can also draw a population pyramid of the *predicted* number of DM patients using colors that range from very light to full:

```
> shd <- c(0.0, 1.1, 2.0, 2.8) / 3
> een <- rep(1,4)
> clr <- rgb( c(een,rev(shd)),
+             c(shd,rev(shd)),
+             c(shd,    een ) )
> clr
```

```
[1] "#FF0000" "#FF5E5E" "#FFAAAA" "#FFEEEE" "#EEEEFF" "#AAAAFF" "#5E5EFF" "#0000FF"
> oo <- c("mort","inc","const","org")
> lim <- 6
> draw.pyr <-
+ function(pp)
+ {
+ par( mar=c(3,3,3,0), mgp=c(3,1,0)/1.6, las=1 )
+ barplot( height=t( cbind( -apply(prn[,pp,"M","apc",       ], 1, sum),
+                                  prn[,pp,"M","apc",    oo ],
+                                  prn[,pp,"F","apc",rev(oo)] ) )/ 1000,
+          horiz=TRUE, col=clr[c(1,8:2)], border=rep("transparent",8),
+          space=0, axes=FALSE, names.arg=rep("",dim(prn)[1]),
+          xlim=c(-1,1)*lim*1.05,
+          xlab="Persons in 1 year class (1000s)",ylab="Age")
+ abline(h=seq(0,100,5),
+        v=seq(-lim,lim,0.5),
+        col="white")
+ axis( side=1, at=seq(-lim,lim,1), labels=abs(seq(-lim,lim,1)) )
+ axis( side=2, at=seq(0,100,20) )
+ axis( side=2, at=seq(0,100,5), labels=NA, tcl=-0.3 )
+ tt <- addmargins( apply( prn[,pp,,"apc",],2:3, sum ), 2 )
+ nn <- tt / tt[,5] * 100
+ ppos <- seq(1,5.9,,5)-0.1
+ npos <- -rev(ppos)
+ mtext( pp, at=-lim, adj=1.8, line=2, cex=1.2, font=1 )
+ mtext( c(lg<- c("Mort","Inc","Imbal","Org","All"),rev(lg)),
+        at=c(npos,ppos), col="black", cex=0.99, line=2 )
+ mtext( formatC(tt["M",1:5],0,,"f",,,","),
+        at=npos, col="blue", line=1, cex=0.99 )
+ mtext( formatC(tt["F",5:1],0,,"f",,,","),
+        at=ppos, col="red" , line=1, cex=0.99 )
+ mtext( formatC(nn["M",1:4],1,4,"f"),
+        at=npos[1:4], col="blue", line=0, cex=0.99 )
+ mtext( formatC(nn["F",4:1],1,4,"f"),
+        at=ppos[2:5], col="red" , line=0, cex=0.99 )
+ mtext( "N", at=0, line=1, cex=0.99 )
+ mtext( "%", at=0, line=0, cex=0.99 )
+ }
> pdf( "comp-DMpr-film.pdf", width=9, height=6 )
> for( pp in paste(1996:2016) ) draw.pyr(pp)
> dev.off()
null device
          1

> for( pp in paste(1996:2016) )
+ {
+ pdf( paste("comp-DMpr-", pp, ".pdf", sep=""), width=8, height=6 )
+ draw.pyr(pp)
+ dev.off()
+ }
```

5.5 Time trends in the components

It is of course also of interest to see how large a fraction of the DM-patients in various ages that can be attributed to the different components at different times.

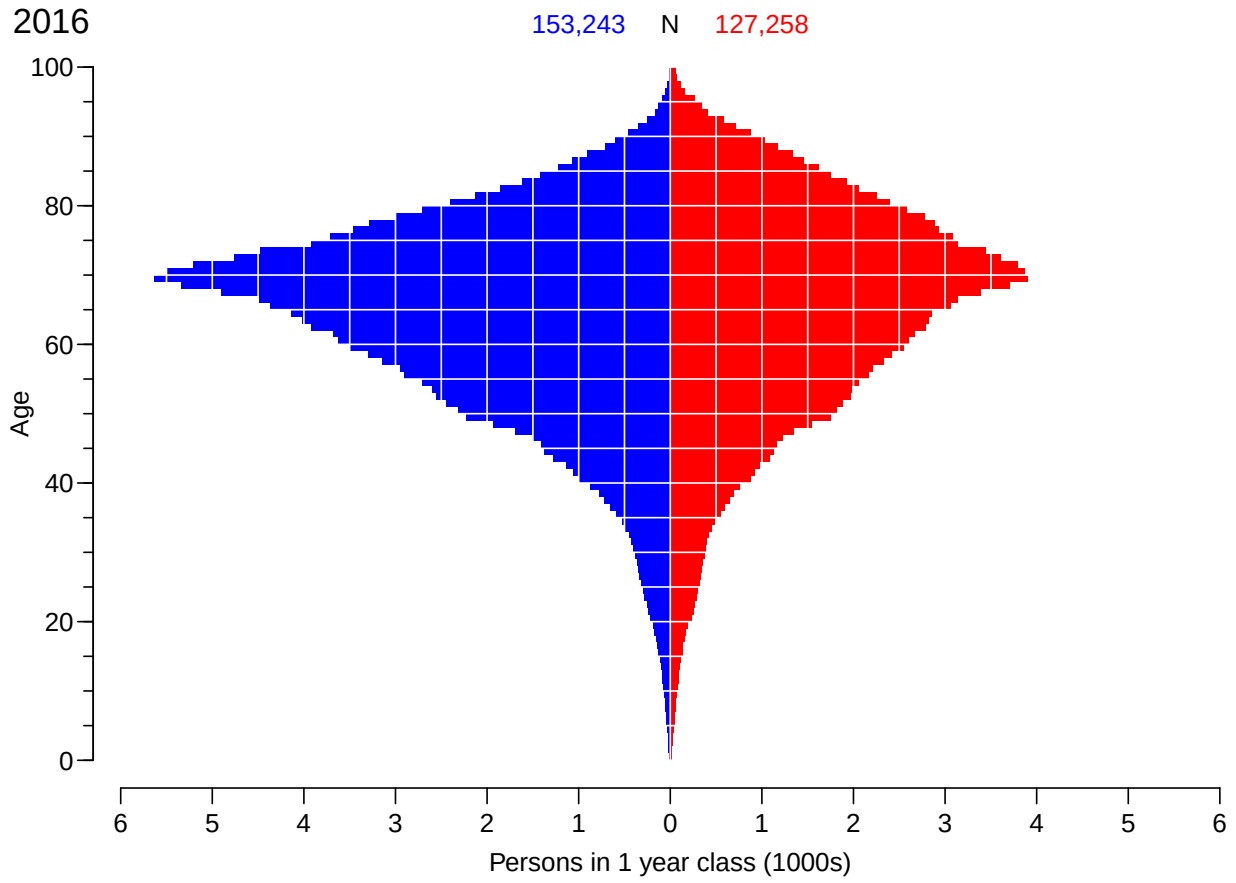

Figure 5.6: *Empirical age-distribution of the diabetes cases in Denmark as of 1.1.2016.*
`./comp-obs-2016`

```
> str( prv )
 num [1:1200, 1:241, 1:2, 1:2, 1:8] 0 0.000553 0.000557 0.000562 0.000566 ...
 - attr(*, "dimnames")=List of 5
  ..$ a   : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.29166666666666
  ..$ t   : chr [1:241] "1996" "1996.08333333333" "1996.16666666667" "1996.25" ...
  ..$ sex : chr [1:2] "M" "F"
  ..$ mod : chr [1:2] "ap" "apc"
  ..$ what: chr [1:8] "obs" "m-fix" "i-fix" "all-f" ...
```

```
> dimnames( prv )[[5]]
[1] "obs"   "m-fix" "i-fix" "all-f" "mort"  "inc"   "const" "org"
```

We extract the prevalences at ages 60, 70 and 80, and compute the fractions of all DM patients attributable to each component:

```
> aloc <- match( 6:8*10, floor(as.numeric(dimnames(prv)[[1]])) )
> ptrend <- ( prv[aloc,,,"apc",-(2:4)] + prv[aloc-1,,,"apc",-(2:4)] )/2
> str( ptrend )
 num [1:3, 1:241, 1:2, 1:5] 0.043 0.0555 0.0658 0.0433 0.0559 ...
 - attr(*, "dimnames")=List of 4
  ..$ a   : chr [1:3] "60.0416666666667" "70.0416666666667" "80.0416666666667"
```

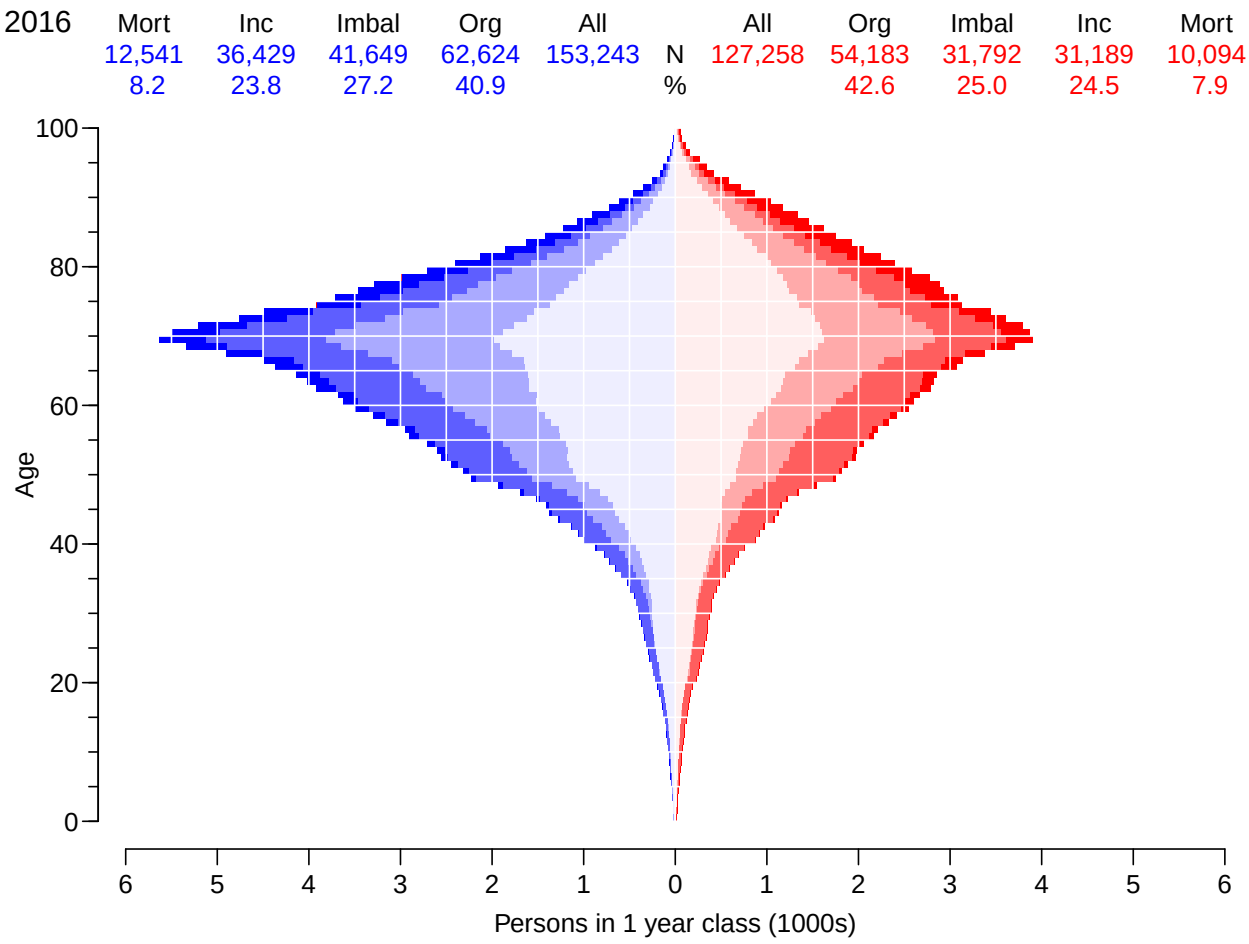

Figure 5.7: *Age-distribution of the predicted no. of diabetes cases in Denmark as of 1.1.2016, subdivided by the components of disease prevalence:*
Mort: decrease in mortality,
Inc: increase in incidence,
Imbal: constant rates from 1996 (non-steady-state imbalance between incidence and mortality in 1996
Org: age-specific prevalences at 1996-01-01. `./comp-DMpr-2016`

```
..$ t   : chr [1:241] "1996" "1996.08333333333" "1996.16666666667" "1996.25" ...
..$ sex : chr [1:2] "M" "F"
..$ what: chr [1:5] "obs" "mort" "inc" "const" ...
```

```
> # Fraction of all DM at each age
> ptrend[,,,-1] <- ptrend[,,,-1]/ptrend[,,,rep("obs",4)]
```

We can now plot the stacked fractions of the components for ages 60, 70 and 80 separately for men and women:

```
> p.pt <- as.numeric(dimnames(ptrend)[[2]])
> par( mfcol=c(3,2), mar=c(0,0,0,2), oma=c(3,4,1,0), mgp=c(3,1,0)/1.6, las=1, bty="n" )
> for( sx in c("M","F") )
+ for( ag in 1:3 )
+ {
+ plot( NA, xlim=c(1996,2016), ylim=0:1*100,
```

```
+           xaxs="i", yaxs="i", xaxt="n", yaxt="n",xlab="",ylab="" )
+ #if( sx=="M" ){
+ #axis( side=2 )
+ #axis( side=2, at=1:9*10, labels=NA ) }
+ if( ag==3 ){
+ axis( side=1 )
+ axis( side=1, at=1:20+1996, labels=NA, tcl=-0.3 ) }
+ polygon( c(p.pt,rev(p.pt)), c( ptrend[ag,,sx,"mort"] *100,
+                              rev(ptrend[ag,,sx,"mort"])*  0),
+          col=clr[if(sx=="M") 8 else 1], border="transparent" )
+ polygon( c(p.pt,rev(p.pt)), c(ptrend[ag,,sx,"mort"],
+                             rev(ptrend[ag,,sx,"mort"]+
+                                 ptrend[ag,,sx,"inc" ]))*100,
+          col=clr[if(sx=="M") 7 else 2], border="transparent" )
+ polygon( c(p.pt,rev(p.pt)), c(ptrend[ag,,sx,"mort"]+
+                                 ptrend[ag,,sx,"inc" ],
+                             rev(ptrend[ag,,sx,"mort"]+
+                                 ptrend[ag,,sx,"inc" ]+
+                                 ptrend[ag,,sx,"const"]))*100,
+          col=clr[if(sx=="M") 6 else 3], border="transparent" )
+ abline( v=seq(2000,2015,5), h=1:9*10, col="white" )
+ axis( side=4, at=1:7*10, tcl=-0.5 )
+ axis( side=4, at=0:14*5, tcl=-0.4, labels=NA )
+ axis( side=4, at=0:70  , tcl=-0.2, labels=NA )
+ text( 1997, 75, paste( if(sx=="F") "Women\n" else "Men\n","age",(6:8*10)[ag]),
+       cex=1.2, font=2, adj=0 )
+ box(col=gray(0.7),bty="o")
+ }
> mtext( side=1, "Date", line=2, outer=TRUE, cex=0.67 )
> mtext( side=2, "Prevalence component (%)", line=2, outer=TRUE, cex=0.67, las=0 )
```

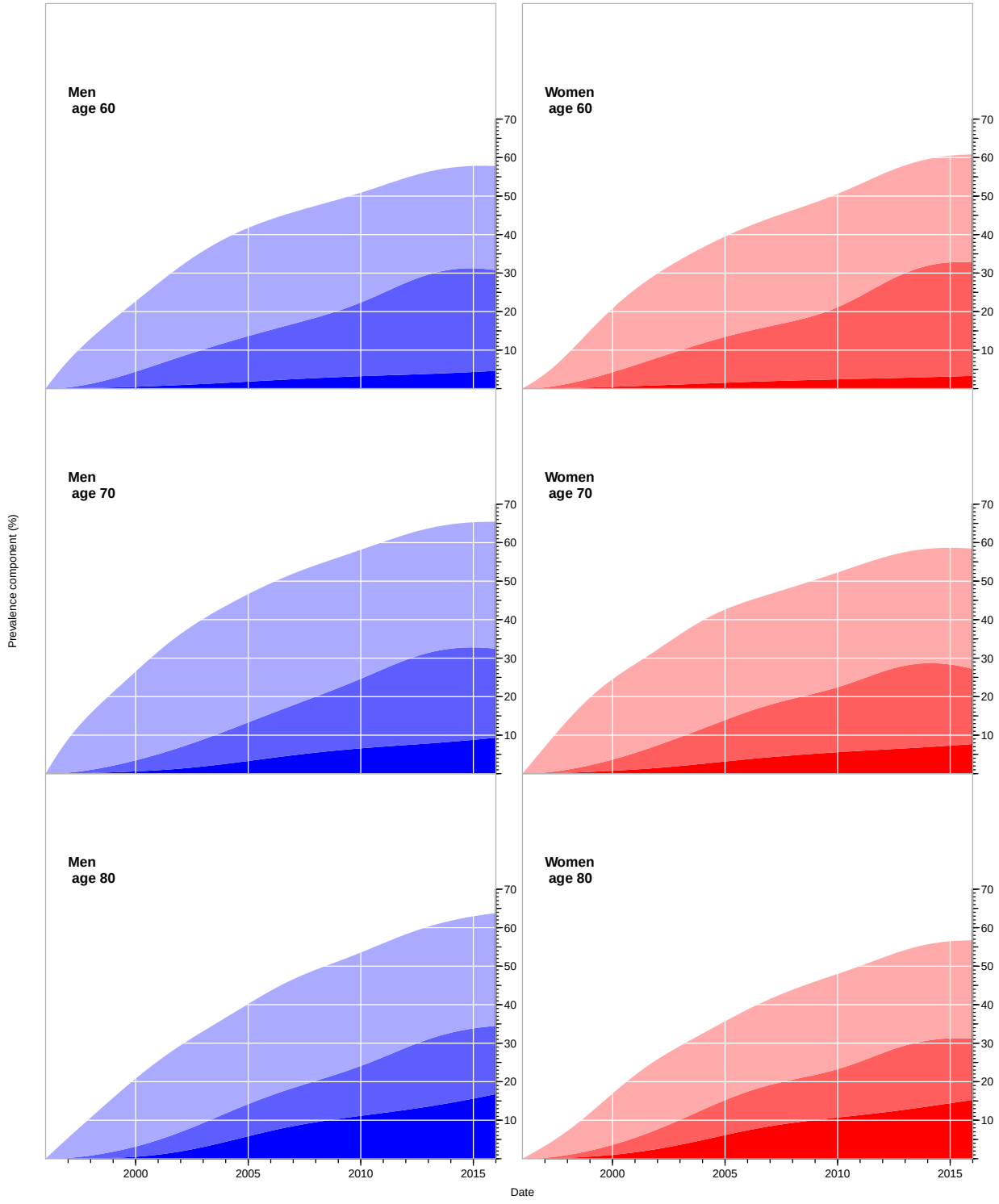

Figure 5.8: *Fraction of the prevalent cases at different times attributable to a) declining mortality (bottom, full color), b) increasing incidence (middle, pale color) and c) prevalence/mortality imbalance at 1996 (top, weak color). The white area above the curves corresponds to the fraction of the cases that would have been around if incidence and mortality rates had remained as in 1996.* `./comp-frac`

References

[1] B Carstensen. Age-Period-Cohort models for the Lexis diagram. *Statistics in Medicine*, 26(15):3018–3045, July 2007.

[2] TR Holford. The estimation of age, period and cohort effects for vital rates. *Biometrics*, 39:311–324, 1983.

[3] B Møller, H Fekjær, T Hakulinen, H Sigvaldason, HH Storm, M Talbäck, and T Haldorsen. Prediction of cancer incidence in the Nordic contries: empirical comparison of different approaches. *Statistics in Medicine*, 22:2751–2766, 2003.

[4] M. J. Rutherford, J. R. Thompson, and P. C. Lambert. Projecting cancer incidence using age-period-cohort models incorporating restricted cubic splines. *Int J Biostat*, 8(1):33, Nov 2012.

[5] Peter Sasieni. Age-period-cohort models in Stata. *Stata Journal*, 12:46–60, 2012.