Rates and predictions from the (revived) Danish Diabetes Register

SDC
September 2017
http://bendixcarstensen.com/DMreg/NewAna.pdf
Version 1

Compiled Friday 8th September, 2017, 11:23
from: /home/bendix/sdc/DMreg/NewReg/NewPred.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.

```
> library( Epi )
> library( splines )
> clear()
```

Chapter 2

Analysis and prediction of rates

2.1 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 ($\triangledown$ and $\triangle$), classified by age, period and cohort:

```
> load( '../data/inits.Rda' )
> load( '../data/TT.Rda' )
> lls()
   name mode     class      dim             size(Kb)
1  a.pt numeric  numeric    1200                 9.4
2  inc  list     data.frame 113189 10        7,519.8
3  int  numeric  numeric    1                    0.0
4  nk.a numeric  numeric    1                    0.0
5  nk.c numeric  numeric    1                    0.0
6  nk.p numeric  numeric    1                    0.0
7  p.pt numeric  numeric    528                  4.2
8  qn   function function   1                    3.5
9  t.pt numeric  numeric    529                  4.2
10 TT   list     data.frame 40000 12         3,440.4
```

We convert risk time to person years from person-millenia as used in the SAS-programs, and

```
> TT[,c("Y.nD","Y.DM")] <- TT[,c("Y.nD","Y.DM")]*10^3
> str( TT )
'data.frame':        40000 obs. of  12 variables:
 $ sex : Factor w/ 2 levels "M","F": 1 1 1 1 1 1 1 1 1 1 ...
 $ reg : Factor w/ 5 levels "Nord","Midt",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ A   : num  0.333 0.667 0.333 0.667 0.333 ...
 $ P   : num  1997 1996 1998 1997 1999 ...
 $ U   : num  0 1 0 1 0 1 0 1 0 1 ...
 $ T1  : num  0 0 0 0 1 0 0 0 1 0 ...
 $ T2  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ D.nD: num  0 5 0 0 1 3 0 2 0 6 ...
 $ Y.nD: num  1887 1898 1827 1816 1815 ...
 $ DM  : num  0 0 0 0 1 0 0 0 1 0 ...
 $ D.DM: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Y.DM: num  0 0 0 0 0.0185 ...
> head( TT )
```

```
  sex  reg         A        P U T1 T2 D.nD     Y.nD DM D.DM       Y.DM
1   M Nord 0.3333333 1996.667 0  0  0    0 1887.285  0    0 0.00000000
2   M Nord 0.6666667 1996.333 1  0  0    5 1897.674  0    0 0.00000000
3   M Nord 0.3333333 1997.667 0  0  0    0 1826.582  0    0 0.00000000
4   M Nord 0.6666667 1997.333 1  0  0    0 1816.407  0    0 0.00000000
5   M Nord 0.3333333 1998.667 0  1  0    1 1815.320  1    0 0.01848049
6   M Nord 0.6666667 1998.333 1  0  0    3 1798.205  0    0 0.00000000
> summary( TT )
 sex         reg             A                P              U             T1
 M:20000   Nord:8000   Min.   : 0.3333   Min.   :1996   Min.   :0.0   Min.   :0.0000
 F:20000   Midt:8000   1st Qu.:25.1667   1st Qu.:2001   1st Qu.:0.0   1st Qu.:0.0000
           Syd :8000   Median :50.0000   Median :2006   Median :0.5   Median :0.0000
           Hov :8000   Mean   :50.0000   Mean   :2006   Mean   :0.5   Mean   :0.5043
           Sjll:8000   3rd Qu.:74.8333   3rd Qu.:2011   3rd Qu.:1.0   3rd Qu.:1.0000
                       Max.   :99.6667   Max.   :2016   Max.   :1.0   Max.   :9.0000
       T2              D.nD             Y.nD                DM             D.DM
 Min.   : 0.000   Min.   :  0.00   Min.   :   1.521   Min.   : 0.000   Min.   : 0.000
 1st Qu.: 0.000   1st Qu.:  1.00   1st Qu.:1485.592   1st Qu.: 1.000   1st Qu.: 0.000
 Median : 3.000   Median :  9.00   Median :2476.061   Median : 4.000   Median : 1.000
 Mean   : 6.938   Mean   : 18.85   Mean   :2640.284   Mean   : 7.443   Mean   : 3.153
 3rd Qu.:11.000   3rd Qu.: 30.00   3rd Qu.:3891.877   3rd Qu.:11.000   3rd Qu.: 5.000
 Max.   :91.000   Max.   :187.00   Max.   :7959.025   Max.   :91.000   Max.   :30.000
      Y.DM
 Min.   :  0.00
 1st Qu.: 12.36
 Median : 48.80
 Mean   : 88.75
 3rd Qu.:130.13
 Max.   :817.38
> # addmargins(  xtabs( Y.nD+Y.DM ~ floor(P) + sex, data=TT ), 2 )
```

Modeling of rates requires that the person-years are strictly positive, which the person-years for diabetes patients are not in all Lexis triangles:

```
> table( TT$Y.DM==0,TT$Y.nD==0 )
        FALSE
  FALSE 39339
  TRUE    661
```

We see from the tabulation that we truly have data in Lexis-triangles; there are 10 observations in each, one for each cobination of sex and the five geographical regions:

```
> 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       .      10       .      10       .      10
  0.67      10       .      10       .      10       .
  1.33       .      10       .      10       .      10
  1.67      10       .      10       .      10       .
  2.33       .      10       .      10       .      10
  2.67      10       .      10       .      10       .
  3.33       .      10       .      10       .      10
  3.67      10       .      10       .      10       .
  4.33       .      10       .      10       .      10
  4.67      10       .      10       .      10       .
```

A brief overview of the number of events and PY:

```
> 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 )
> round( ftable( addmargins(tt,1:2), row.vars=1:2 ) )
                     DM      D.nD      Y.nD      D.DM      Y.DM     D.tot     Y.tot
sex floor(P)
M   1996           5803     20893   2549628      2405     49813     23298   2599441
    1997           5832     20412   2557382      2490     53342     22902   2610723
    1998           6300     19854   2563031      2569     57252     22423   2620283
    1999           6696     19584   2567653      2817     61256     22401   2628909
    2000           6610     19155   2572711      2826     65491     21981   2638203
    2001           6455     19268   2579212      2838     69538     22106   2648750
    2002           7488     19129   2584718      3088     73836     22217   2658554
    2003           8374     18937   2587506      3226     78972     22163   2666477
    2004           8604     18325   2589030      3245     84671     21570   2673701
    2005           7958     17857   2592038      3314     90000     21171   2682038
    2006           7958     18347   2597829      3403     94568     21750   2692397
    2007           8511     18066   2607102      3495     99287     21561   2706389
    2008           9251     17642   2620265      3488    104985     21130   2725250
    2009           9885     17394   2632274      3791    111007     21185   2743281
    2010          10695     17151   2641090      3833    117745     20984   2758835
    2011          13427     16589   2647979      3970    126508     20559   2774487
    2012          11014     16265   2654244      4113    135638     20378   2789882
    2013           8360     16277   2665032      4250    141211     20527   2806244
    2014           7924     15788   2682918      4352    144848     20140   2827766
    2015           8911     16143   2684676      4433    147382     20576   2832058
    Sum          166056    363076  52176317     67946   1907350    431022  54083667
F   1996           4791     21557   2617015      2289     46687     23846   2663703
    1997           4756     21360   2624116      2352     49359     23712   2673475
    1998           4903     20656   2629819      2341     52268     22997   2682087
    1999           5338     21462   2634358      2489     55319     23951   2689677
    2000           5278     20688   2639480      2588     58590     23276   2698070
    2001           5217     20971   2645599      2568     61566     23539   2707165
    2002           6402     21214   2650437      2683     65172     23897   2715609
    2003           6915     20575   2652208      2828     69536     23403   2721743
    2004           7034     19637   2653858      2758     74384     22395   2728242
    2005           6494     19599   2656729      2907     78711     22506   2735441
    2006           6058     19510   2662228      3025     81897     22535   2744125
    2007           6751     19606   2669969      3047     85325     22653   2755293
    2008           7255     19030   2681305      2941     89692     21971   2770997
    2009           7323     19044   2693224      3144     94069     22188   2787293
    2010           8018     18569   2703131      3297     98841     21866   2801972
    2011          11242     18105   2710123      3184    105893     21289   2816016
    2012           8651     17604   2714779      3247    113800     20851   2828579
    2013           6497     17542   2724176      3386    118237     20928   2842413
    2014           5969     16980   2738722      3430    120983     20410   2859705
    2015           6754     17171   2733760      3677    122505     20848   2856265
    Sum          131646    390880  53435037     58181   1642833    449061  55077870
Sum 1996          10594     42450   5166643      4694     96500     47144   5263144
    1997          10588     41772   5181498      4842    102700     46614   5284198
    1998          11203     40510   5192850      4910    109520     45420   5302369
    1999          12034     41046   5202011      5306    116575     46352   5318586
    2000          11888     39843   5212192      5414    124081     45257   5336273
    2001          11672     40239   5224811      5406    131104     45645   5355915
    2002          13890     40343   5235154      5771    139008     46114   5374163
    2003          15289     39512   5239713      6054    148507     45566   5388220
```

```
     2004      15638    37962   5242888     6003    159055     43965   5401943
     2005      14452    37456   5248768     6221    168711     43677   5417479
     2006      14016    37857   5260058     6428    176464     44285   5436522
     2007      15262    37672   5277070     6542    184612     44214   5461683
     2008      16506    36672   5301570     6429    194677     43101   5496248
     2009      17208    36438   5325498     6935    205076     43373   5530575
     2010      18713    35720   5344221     7130    216586     42850   5560807
     2011      24669    34694   5358102     7154    232400     41848   5590503
     2012      19665    33869   5369023     7360    249438     41229   5618461
     2013      14857    33819   5389208     7636    259449     41455   5648657
     2014      13893    32768   5421640     7782    265831     40550   5687470
     2015      15665    33314   5418436     8110    269887     41424   5688323
     Sum      297702   753956 105611354   126127   3550183    880083 109161537
```

We can devise the incidence and mortality rates — from which we see that there is actually quite a difference between the overall mortality rates and the non-DM mortality rates; mainly because of the different age-distributions.

```
> round( ftable( tt[,,c(1,2,4,6)]/tt[,,c(3,3,5,7)]*10^5, row.vars=1:2 ), 1 )
                  DM   D.nD   D.DM  D.tot
sex floor(P)
M   1996       227.6  819.5 4828.0  896.3
    1997       228.0  798.2 4668.0  877.2
    1998       245.8  774.6 4487.2  855.7
    1999       260.8  762.7 4598.8  852.1
    2000       256.9  744.5 4315.1  833.2
    2001       250.3  747.0 4081.2  834.6
    2002       289.7  740.1 4182.2  835.7
    2003       323.6  731.9 4085.0  831.2
    2004       332.3  707.8 3832.5  806.7
    2005       307.0  688.9 3682.2  789.4
    2006       306.3  706.2 3598.5  807.8
    2007       326.5  693.0 3520.1  796.7
    2008       353.1  673.3 3322.4  775.3
    2009       375.5  660.8 3415.1  772.3
    2010       404.9  649.4 3255.3  760.6
    2011       507.1  626.5 3138.1  741.0
    2012       415.0  612.8 3032.3  730.4
    2013       313.7  610.8 3009.7  731.5
    2014       295.4  588.5 3004.5  712.2
    2015       331.9  601.3 3007.8  726.5
F   1996       183.1  823.7 4902.9  895.2
    1997       181.2  814.0 4765.1  886.9
    1998       186.4  785.5 4478.9  857.4
    1999       202.6  814.7 4499.4  890.5
    2000       200.0  783.8 4417.1  862.7
    2001       197.2  792.7 4171.1  869.5
    2002       241.5  800.4 4116.8  880.0
    2003       260.7  775.8 4067.0  859.9
    2004       265.0  739.9 3707.8  820.9
    2005       244.4  737.7 3693.2  822.8
    2006       227.6  732.8 3693.7  821.2
    2007       252.8  734.3 3571.1  822.2
    2008       270.6  709.7 3279.0  792.9
    2009       271.9  707.1 3342.2  796.0
    2010       296.6  686.9 3335.6  780.4
    2011       414.8  668.1 3006.8  756.0
    2012       318.7  648.5 2853.2  737.2
```

```
    2013       238.5  643.9 2863.7  736.3
    2014       217.9  620.0 2835.1  713.7
    2015       247.1  628.1 3001.5  729.9
```

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 + gP,
+              data = transform( subset( TT, P > 1996 ),
+                                gP = factor( (P>=2001)+(P>=2006)+(P>=2011),
+                                             labels=c("1996-2000",
+                                                      "2001-2005",
+                                                      "2006-2010",
+                                                      "2011-2015") ) ) )
> # round( ftable( tt <- addmargins(tt,1:2), row.vars=1:2 ) )
> ftable( formatC( tt[,,1:5], format="f", digits=0, big.mark=",", width=11 ),
+         row.vars=1:2 )
                         DM        D.nD        Y.nD        D.DM        Y.DM
sex gP
M   1996-2000       31,241      99,898  12,810,404      13,107     287,154
    2001-2005       38,879      93,516  12,932,503      15,711     397,017
    2006-2010       46,300      88,600  13,098,560      18,010     527,593
    2011-2015       49,636      81,062  13,334,849      21,118     695,587
F   1996-2000       25,066     105,723  13,144,789      12,059     262,222
    2001-2005       32,062     101,996  13,258,831      13,744     349,369
    2006-2010       35,405      95,759  13,409,857      15,454     449,824
    2011-2015       39,113      87,402  13,621,560      16,924     581,417
> ftable( formatC( tt[,,c(1,2,4)]/tt[,,c(3,3,5)]*10^5,
+                  format="f", digits=1, big.mark=",", width=10 ),
+         row.vars=1:2 )
                        DM       D.nD       D.DM
sex gP
M   1996-2000       243.9      779.8    4,564.5
    2001-2005       300.6      723.1    3,957.3
    2006-2010       353.5      676.4    3,413.6
    2011-2015       372.2      607.9    3,036.0
F   1996-2000       190.7      804.3    4,598.8
    2001-2005       241.8      769.3    3,933.9
    2006-2010       264.0      714.1    3,435.6
    2011-2015       287.1      641.6    2,910.8
```

2.1.1 Arrays of predictions

We set up arrays to hold the predicted incidence and mortality rates from the different models, separately for the two sexes; we are using points for midpoints of age and calendar time categories to identify rates and for boundaries of calendar time categories to identify prevalences:

```
> ht <- function(x) round(c(head(x,4),NA,tail(x,4)),3)
> ht(a.pt)
[1]  0.042  0.125  0.208  0.292     NA 99.708 99.792 99.875 99.958
> ht(p.pt)
```

```
[1] 1996.042 1996.125 1996.208 1996.292       NA 2039.708 2039.792 2039.875 2039.958
> ht(t.pt)
[1] 1996.000 1996.083 1996.167 1996.250       NA 2039.750 2039.833 2039.917 2040.000
> int
[1] 0.08333333
> mean( diff(a.pt) )
[1] 0.08333333
> mean( diff(p.pt) )
[1] 0.08333333
```

```
> Lambda <- Mu.W <- Mu.DM <-
+ NArray( list( a = a.pt,
+                p = p.pt,
+              sex = c("M","F"),
+              mod = c("ap","apc","LCa",
+                      "att","atx","i20","i25","i30") ) )
> 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" ...
> length( Lambda )
[1] 10137600
```

Note that on dimension 4, we have levels to hold the naive prediction using the natural splines for the AP, APC [2] and Lee-Carter models [1], as well for the model with trend attenuation (see below for explanation), `att`. Further we have levels for the models with stronger attenuation and an added time-trend, `atx`, and finally for the simpler scenarios where the rates at the end of 2015 are predicted to increase 2, 2.5 and 3.0% per year, respectively.

2.1.2 Datasets for rate modeling

First we construct simple datasets 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.

```
> 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" )
```

2.2 Models for incidence and mortality rates

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

We can 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 = "imdat.Rda" )
```

Once we have fitted all models for all transitions we can show their relative merits in terms of deviance, and assess to what extent the APC models would suffice or whether substantial improvements could be obtained by using a Lee-Carter model:

```
> load( file = "imdat.Rda" )
> par( mfcol=c(3,2) )
> show.apc.LCa( minc, col.txt="blue", cex=1.2 ); text( 10, 90, "DM\n incidence", cex=3 )
> show.apc.LCa( mmDM, col.txt="blue", cex=1.2 ); text( 10, 90, "DM\n mortality", cex=3 )
> show.apc.LCa( mmnD, col.txt="blue", cex=1.2 ); text( 10, 90, "non-DM\n mortality", cex=3
> show.apc.LCa( finc, col.txt="red" , cex=1.2 )
> show.apc.LCa( fmDM, col.txt="red" , cex=1.2 )
> show.apc.LCa( fmnD, col.txt="red" , cex=1.2 )
```

From the relative model fits shown in figure 2.1 we see that for diabetes incidence and mortality the APC is quite well fitting relative to the classical Lee-Carter model (APa), but that the Lee-Carter type models offer some improvement relative to the APC model for non-DM mortality but only if cohort-age effect is included.

Overall, however, it seems that it is the most elaborate model that offer improvement over the APC-model so we will include the fit of the extended Lee-Carter model in the objects with fitted values too.

Thus, we shall proceed with APC-models for the incidence and mortality rates.

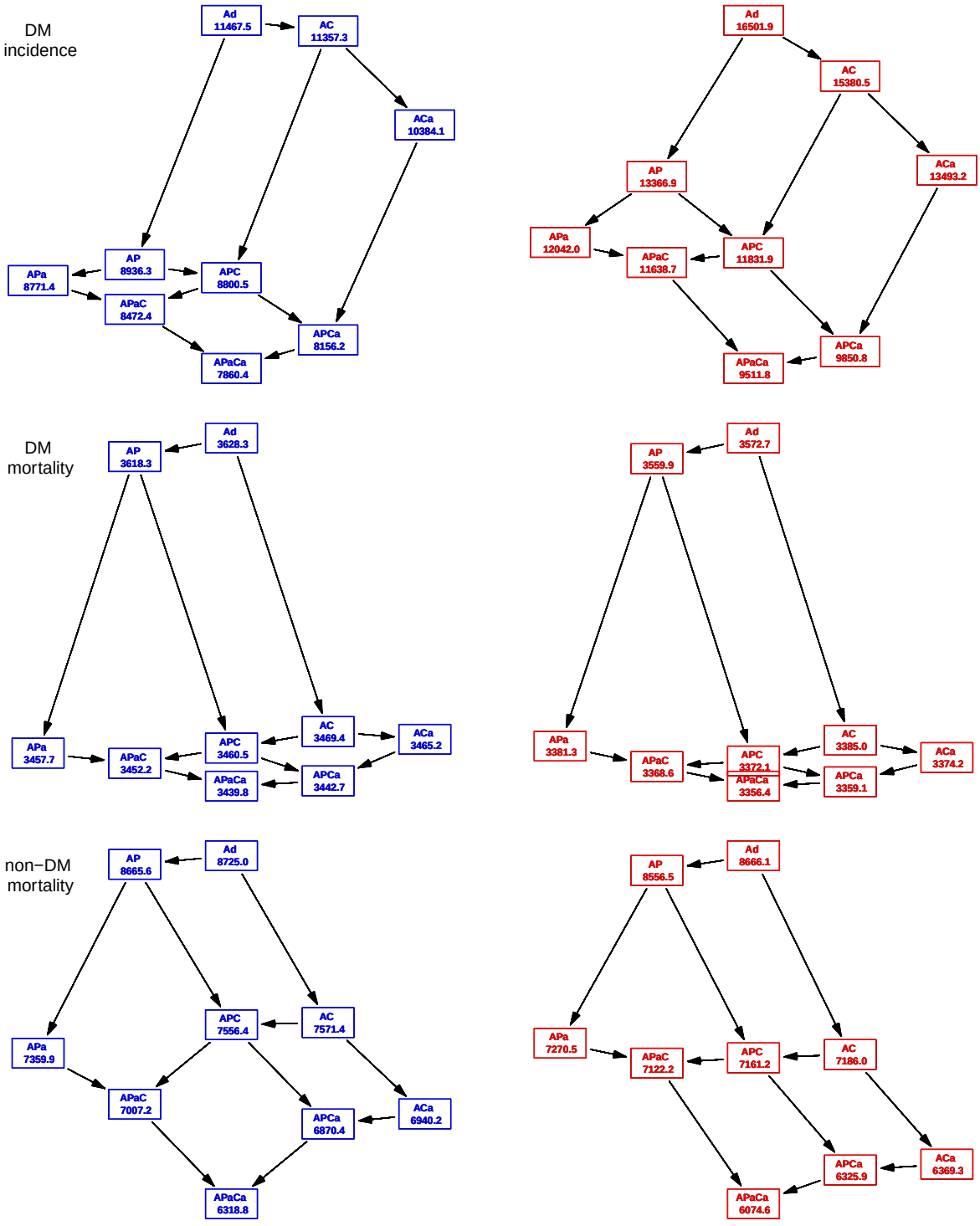

Figure 2.1: *Relative fit of APC and Lee-Carter models for incidence and mortality rates for men (blue) and women (red) respectively.*

2.3 Incidence rates

Based on this we can now derive the location of the knots for the age, period and cohort terms in the model (first loading the prerequisites):

```
> load( file="../data/inits.Rda" )
> ( ki.a <- with( incdat, quantile( rep(  A,D), qn(nk.a) ) ) )
   6.25%   18.75%   31.25%   43.75%   56.25%   68.75%   81.25%   93.75%
30.33333 45.66667 53.33333 58.66667 63.66667 68.33333 74.33333 82.33333

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

8.333333%       25% 41.66667% 58.33333%       75% 91.66667%
 1998.333  2002.667  2006.333  2009.333  2011.667  2014.333

> ( ki.c <- with( incdat, quantile( rep(P-A,D), qn(nk.c) ) ) )

   6.25%   18.75%   31.25%   43.75%   56.25%   68.75%   81.25%   93.75%
1922.333 1931.667 1938.333 1943.667 1948.333 1954.333 1962.333 1977.667
```

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

```
> 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") )
> 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] 8865.076 8865.076

> f.inc.ap  <- update( m.inc.ap , 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 )*100
> rownames( inc.chg ) <- c("DM incidence change     Men",
+                          "                       Women")
> round( inc.chg, 2 )
                          exp(Est.) 2.5% 97.5%
DM incidence change     Men      2.25 2.16  2.34
                      Women      3.50 3.40  3.60
```

The average increase is similar in women and men, but the period effect is massively non-linear, so these summary figures are not really informative, see the comparative figure with the mortality rates below.

Extended Lee-Carter 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=8,
+                                     p=6,
+                                     c=8,
+                                    pi=8,
+                                    ci=8 ),
+                         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=8,
+                                     p=6,
+                                     c=8,
+                                    pi=8,
+                                    ci=8 ),
+                         eps = 1e-4,
+                          VC = FALSE,
+                       quiet = FALSE )
> save( m.inc.LCa,
+       f.inc.LCa, file="../data/incLCa.Rda" )

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

2.3.1 Incidence rate predictions

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

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

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

```
> dim( nd )
[1] 633600      3
```

Note that the prediction data frame was set up with age varying fastest, and the `Lambda` array with age before period, so that the column-major storage of arrays conforms with the predictions form `nd`:

```
> 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 )
> # 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.

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(237,240)] )
[1] "2015.70833333333" "2015.95833333333"
> ( ell <- diff( as.numeric( ell ) ) )
[1] 0.25
> tr.ap  <- (exp(log((Lambda[,240,,"ap" ]/Lambda[,237,,"ap" ]))/ell)-1)*100
> tr.apc <- (exp(log((Lambda[,240,,"apc"]/Lambda[,237,,"apc"]))/ell)-1)*100
> matplot( as.numeric(dimnames(Lambda)[[1]]), cbind( tr.ap, tr.apc ),
+          type="l", lty=rep(c(3,1),each=2), lwd=3, col=c("blue","red"),
+          ylim=range(c(0,tr.ap,tr.apc)), xlab="Age", ylab="Annual change in rates (%)" )
```

2.3.2 A damping extrapolation

Instead of using the naive extrapolation as above where we blindly just prolong the linear trajectories from the natural splines, we may dampen the trend derived from the naive application of the natural splines. I the vein of the recommendation by Sasieni [5] we use an attenuation of the trend beyond 2016 for period and beyond 2000 for cohort. The attenuation is traditionally (*i.e.* in cancer epidemiology [3, 4]) set to 0.92 per year, but we shall be more conservative and use 0.88.

Theory

The arithmetic goes as follows: Suppose the slope of the period or cohort effect is β, but that it would be an exaggeration to continue the period effect indefinitely at a slope of β, so we choose a *damping* factor, d, say, such that the *slope* of the effect at time t is not β, but rather βd^t. Thus for the effect $f(t)$ we have that:

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

Now, if the original linear extrapolation we have from the natural spline parametrization has the form:

$$f(t) = \alpha + \beta t$$

then at the prediction start (for convenience of notation, $t = 0$):

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

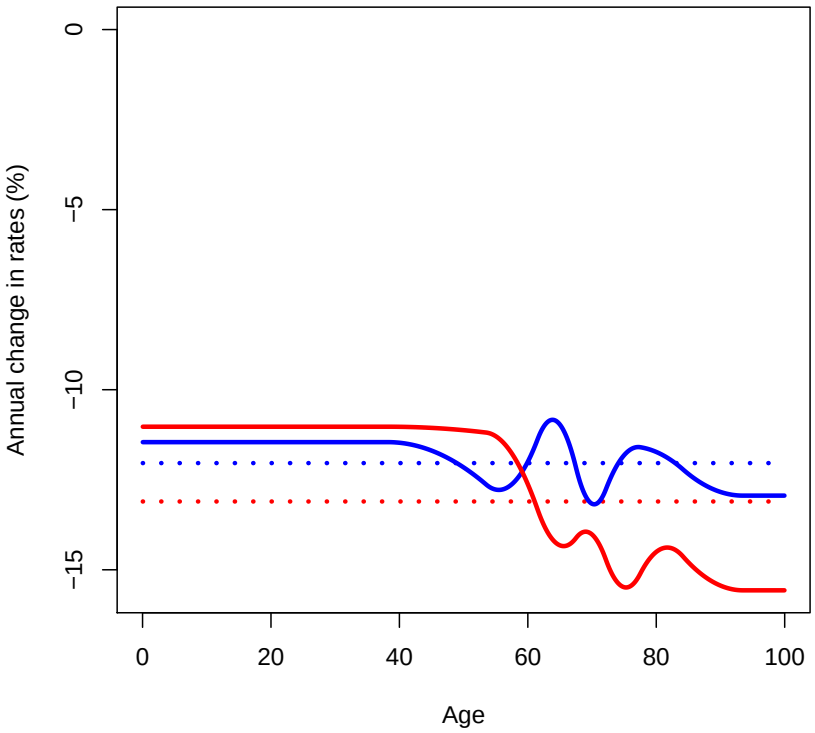

Figure 2.2: *Estimated trend in diabetes incidence rates over the last 3 months of 2015, red curves are women, blue men, the dotted curves are from the AP-model, the full from the APC model.*

Invariance

After the prediction date, p_x, say, the contribution to the linear predictor in the APC-model is at $(a, p), p > p_x$:

$$\eta_{a,p} = f(a)+g(p_x)+\beta(p-p_x)+h(p_x-a)+\gamma(p-a-(p_x-a)) = f(a)+g(p_x)+h(p_x-a)+(\beta+\gamma)(p-p_x)$$

where β and γ are the slopes of the last bit of the persiod resp. cohort term. From the usual theory of APC-model we know that under reparametrization we will always add a given numerical period slope and subtract the same numerical slope fromthe cohort term, so under reparametrization $\beta + \gamma$ will always be constant. Note that this applies to a *given* model structure, that is a *fixed* set of knots for the theree effects.

Now, above we just saw that the attenuation exercise consisted in using the following for prediction of the perido term at p:

$$g(p) = g(p_x) - \beta/\log(d) + \beta d^{p-p_x}/\log(d) = g(p_x) - \beta\frac{1-d^{p-p_x}}{\log(d)}$$

Now by a reparametrization of the APC model by a slope of δ, say, we have:

$$f(a) + g(p) + h(p-a) + \delta(a-p+c) = \big(f(a)+\delta a\big) + \big(f(p)-\delta p\big) + \big(h(p-a)+\delta(p-a)\big)$$

Thus retaining the `model` we only get the same prediction under different parametrizations by using the *same* attenuation factor for period and cohort. In practical terms, it only makes sense to make an attenuation of `both` the perido and the cohort sloes, otherwise the chosen prediction will depend on the chosen parametrization of the given model.

Implementation

The formulae for attenuation are implemented in the function `damp`, which assumes that the predicted effects at times `t` are in `fval`, and that the effects are linear beyond `t0`, but that we attenuate the effect with the (annual) damping factor `dfac`. In cancer epidemiology this is often set to 0.92, but we shall be a bit more conservative and set it to 0.88 (which is also a figure taken out of thin air).

The argument `fval` will in our context be the predicted period resp. cohort term from the model:

```
> damp <-
+ function( fval, t, t0, dfac=0.88 )
+     {
+ # where are the ealiest an latest prediction point after t0
+ wh <- match( range( t[t>t0] ), t )
+ # slope of predicted curve - we assume linearity of teh fitted values
+ beta <- diff( fval[wh] ) / diff( t[wh] )
+ # the attenuated curve - revision beyond first point after t0:
+ ifelse( t>t[wh[1]], fval[wh[1]] - beta * (1-dfac^(t-t[wh[1]])) / log( dfac ), fval )
+     }
```

Note that R will accept the value 0 for the argument `dfac` even if we use the log of it in the function, leading to an immediately flat prediction. This is because `log(0)` is taken to be `-Inf` — a valid value in R, and anything divided by $-\infty$ is taken to be 0. Handy.

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

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

This is the left plot; the other one illustrates the actual damping relative to an arbitrary slope:

```
> par( mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6, bty="n", las=1 )
> d <- seq(0,1,,200)
> fr <- c(0.5,0.1,0.01)
> matplot( d, outer( d, fr, function(d,fr) log(fr)/log(d) ),
+          type="l", lwd=3, lty=1,
+          ylim=c(0,25), xlab="Damping factor",
+          ylab=paste( "Time to reduction to ",
+                      paste( round(fr*100,1), collapse=", "),
+                      "%, respectively", sep="" ) )
> abline( v=c(0.92, 0.88, 0.7) )
>  axis( at=c(0.92, 0.88, 0.7), las=2, side=1 )
> t <- seq( 0,25,,200 )
> f <- 2.5 + 0.4 * t
> t0 <- 16
> matplot( t, cbind( f, damp(f,t,5,dfac=0.88),
+                       damp(f,t,5,dfac=0.70),
+                       damp(f,t,5,dfac=0) ), lty=1, lwd=c(5,2,2,2), type="l",
+          xlab="Time", ylab="Damped effect")
```

```
> text( 6, 12-0:3/2, c( "Damping factor",
+                       formatC( c(0.88,0.70,0), format="f", digits=2 ) ),
+                    font=2, col=1:4, adj=0 )
> abline( v= 5 )
```

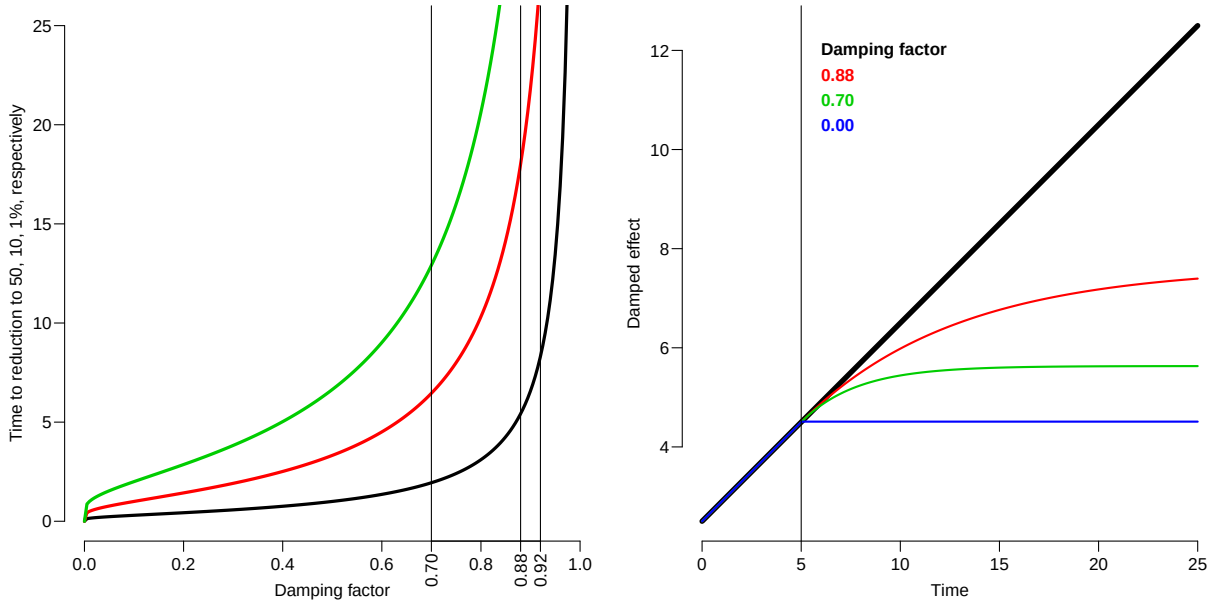

Figure 2.3: *Illustration of the* `damp` *function for attenuation of linear effects. The red curve is with a damping factor of 0.88, the green with 0.70 and the blue with a damping factor of 0; the latter setting any future effect equal to the last observed.*

We can now employ this damping function on the terms-predictions from the apc-models, and show how it pans out in practice for the period and cohort effects. We adjust the predictions for the period and cohort terms and use these to reconstruct the predicted (cumulative) rates, note in particular that the intercept is not in the terms components, it must be retrieved from the attribute `const` in the `predict.glm` object:

```
> predatt.apc <-
+ function( obj, nd, dfac=0.88, p.end=2016, c.end=2000 )
+    {
+ # predicted terms --- linear beyond outer knots
+ pc <- po <- predict.glm( obj, type="terms", newdata=nd )
+ # replace with the attenuated curves
+ pc[,2] <- damp( pc[,2], nd$P     , p.end, dfac=dfac )
+ pc[,3] <- damp( pc[,3], nd$P-nd$A, c.end, dfac=dfac )
+ # return predicted rates as well as original and revised terms predictions
+ list( pr.resp  = exp( apply( pc, 1, sum ) + attr(pc,"const") + log(nd$Y) ),
+       pr.terms = pc,
+       pr.org   = po )
+    }
```

We have observed that the incidence rates show a dramatic decreasing tendency over the last few years of observation (10-15%/year), hence we may want not only to investigate a scenario where rates are kept constant, but also one where we simply increase the rates by 2.5% per year — this is only going to be used for the incidence rates as a sensitivity analysis, and together with the estimated cohort effects.

To this end we update the function `predatt.apc` by allowing adding a trend in calendar time on top of the attenuated prediction; we phase it in quadratically over a period of ℓ, by the function q:

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

or in code:

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

Thus the function `qs` provides a function of `t` which has value 0 at 0, and a slope of `delta` beyond `ell`, so this is a function you would add to the linear predictor of a model with log-link; after `ell` there would be added a linear trend of `delta` og the log-scale, in the case of incidence rates an exponential growth of rates.

So we take the previous attenuation function and install the possibility of adding an extra period trend on top of the attenuated model prediction.

Note that despite the fact we put this extra effect into the period term, this is merely a convenience unrelated to the particular parametrization; we still have the same damping factor for both period and cohort effects (as we should have). The prediction principle here is damping factor *plus* a pure period effect kicking in at the end of the interval

```
> predatt.apc <-
+ function( obj, nd, dfac=0.88, p.end=2016, c.end=2000, ell=1, delta=0 )
+     {
+ # predicted terms --- linear beyond outer knots
+ pc <- po <- predict.glm( obj, type="terms", newdata=nd )
+ # replace with the attenuated curves
+ pc[,3] <- damp( pc[,3], nd$P-nd$A, c.end, dfac=dfac )
+ pc[,2] <- damp( pc[,2], nd$P     , p.end, dfac=dfac ) +
+                 ifelse( nd$P>p.end, qs( nd$P-p.end, ell, delta ), 0 )
+ # return predicted rates as well as original and revised terms predictions
+ list( pr.resp  = exp( apply( pc, 1, sum ) +
+                       attr(pc,"const") +
+                       log(nd$Y) ),
+       pr.terms = pc,
+       pr.org   = po )
+     }
```

We can now illustrate the four variants of predictions, the simple one based on the natural splines from the APC model (which is included in all of ythe obejcts), one using only the 0.88 damping, one adding a long-term trend of 2.5% per year to this and one using exclusively a 2.5% trend

```
> zz <- predatt.apc( m.inc.apc, nd )
> xx <- predatt.apc( m.inc.apc, nd, ell=2, delta=0.025, dfac=0.7 )
> ww <- predatt.apc( m.inc.apc, nd, ell=2, delta=0.025, dfac=0 )
> str( zz )
List of 3
 $ pr.resp : Named num [1:633600] 6.32e-06 6.36e-06 6.40e-06 6.43e-06 6.47e-06 ...
  ..- attr(*, "names")= chr [1:633600] "1" "2" "3" "4" ...
 $ pr.terms: num [1:633600, 1:3] -3.21 -3.2 -3.19 -3.19 -3.18 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:633600] "1" "2" "3" "4" ...
  .. ..$ : chr [1:3] "Ns(A, kn = ki.a)" "Ns(P, kn = ki.p)" "Ns(P - A, kn = ki.c)"
```

```
  ..- attr(*, "constant")= num -6.06
 $ pr.org  : num [1:633600, 1:3] -3.21 -3.2 -3.19 -3.19 -3.18 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:633600] "1" "2" "3" "4" ...
  .. ..$ : chr [1:3] "Ns(A, kn = ki.a)" "Ns(P, kn = ki.p)" "Ns(P - A, kn = ki.c)"
  ..- attr(*, "constant")= num -6.06
> 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 ...
> # Utility to fish out the unique values of A, P and C
> owh <- function(x) match( sort( unique(x) ), x )
> oA <- owh( nd$A )
> oP <- owh( nd$P )
> oC <- owh( nd$P-nd$A )
> # Plotting the curves
> par( mfrow=c(1,3), mar=c(3,1,1,1), oma=c(0,2,0,1), mgp=c(3,1,0)/1.6, las=1, bty="n" )
> matplot( nd$A[oA], exp( cbind( zz$pr.org[oA,1],
+                                zz$pr.terms[oA,1] ) ),
+          type="l", lty=1, lwd=c(5,1), col=gray(c(6,1)/9),
+          xlab="Age", ylab="", ylim=c(0.02,10), log="y" )
> matplot( nd$P[oP], exp( cbind( zz$pr.org[oP,2],
+                                zz$pr.terms[oP,2],
+                                xx$pr.terms[oP,2],
+                                ww$pr.terms[oP,2] ) ),
+          type="l", lty=1, lwd=c(5,2,2,2),
+          col=c(gray(c(6,1)/9),"forestgreen","red"),
+          xlab="Period", ylab="", ylim=c(0.02,10), log="y" )
> abline( v=2016 )
> matplot( (nd$P-nd$A)[oC], exp( cbind( zz$pr.org[oC,3],
+                                       zz$pr.terms[oC,3],
+                                       xx$pr.terms[oC,3],
+                                       ww$pr.terms[oC,3] ) ),
+          type="l", lty=1, lwd=c(5,2,2,2),
+          col=c(gray(c(6,1)/9),"forestgreen","red"),
+          xlab="Cohort", ylab="", ylim=c(0.02,10), log="y" )
> abline( v=2000 )
> mtext( "Relative effect", side=4, line=0, outer=TRUE, las=0 )
```

With this machinery in place we are now able to add the predicted (cumulative) rates to the `Lambda` array:

```
> Lambda[,,"M","att"] <- predatt.apc( m.inc.apc, nd, )$pr.resp
> Lambda[,,"F","att"] <- predatt.apc( f.inc.apc, nd, )$pr.resp
> Lambda[,,"M","atx"] <- predatt.apc( m.inc.apc, nd, delta=0.025, dfac=0.7 )$pr.resp
> Lambda[,,"F","atx"] <- predatt.apc( f.inc.apc, nd, delta=0.025, dfac=0.7 )$pr.resp
> Lambda[,,"M","i20"] <- predatt.apc( m.inc.apc, nd, delta=0.020, dfac=0 )$pr.resp
> Lambda[,,"F","i20"] <- predatt.apc( f.inc.apc, nd, delta=0.020, dfac=0 )$pr.resp
> Lambda[,,"M","i25"] <- predatt.apc( m.inc.apc, nd, delta=0.025, dfac=0 )$pr.resp
> Lambda[,,"F","i25"] <- predatt.apc( f.inc.apc, nd, delta=0.025, dfac=0 )$pr.resp
> Lambda[,,"M","i30"] <- predatt.apc( m.inc.apc, nd, delta=0.030, dfac=0 )$pr.resp
> Lambda[,,"F","i30"] <- predatt.apc( f.inc.apc, nd, delta=0.030, dfac=0 )$pr.resp
```

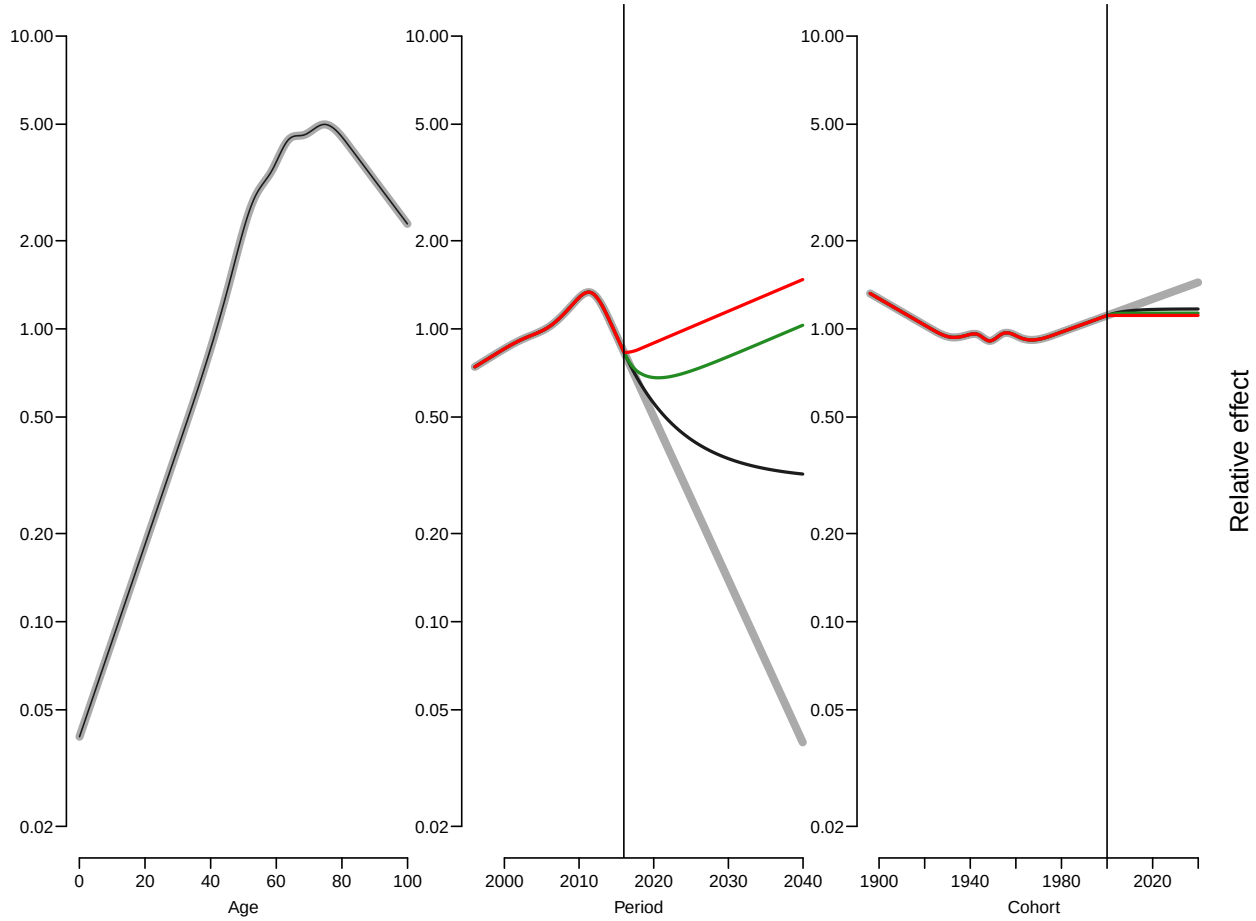

Figure 2.4: *Demonstration of the attenuation of the period and cohort terms from an APC-model. The attenuation effect used is the magic number 0.88 per year; that is, the slope of the effects decreases by a factor 0.88 per year, corresponding to a 47% decrease at 5 years and 72% decrease at 10 years in the slope of the effects.*
The green line is a prediction with a damping factor of 0.7 and an added increase of 2.5% phased in over 2 years.
The red line is an annual increase in period effect of 2.5% phased in over 2 years using the predicted rates at the end of 2015

2.4 Mortality rates

2.4.1 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( rep(A  ,D), qn(nk.a) ) ) )
   6.25%   18.75%   31.25%   43.75%   56.25%   68.75%   81.25%   93.75%
56.33333 66.33333 71.66667 75.66667 79.66667 83.33333 86.66667 91.66667
> ( kmd.p <- with( mDMdat, quantile( rep(P  ,D), qn(nk.p) ) ) )
8.333333%       25% 41.66667% 58.33333%       75% 91.66667%
 1998.333  2002.333  2005.667  2009.333  2011.667  2014.667
> ( kmd.c <- with( mDMdat, quantile( rep(P-A,D), qn(nk.c) ) ) )
   6.25%   18.75%   31.25%   43.75%   56.25%   68.75%   81.25%   93.75%
1912.667 1919.333 1923.333 1927.333 1931.667 1936.333 1942.667 1951.667
> m.md.ap <- glm( D ~ Ns(A,knots=kmd.a) + Ns(P,knots=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 )
> Mu.DM[,,"M","att"] <- predatt.apc( m.md.apc, nd, )$pr.resp
> Mu.DM[,,"F","att"] <- predatt.apc( f.md.apc, nd, )$pr.resp
> Mu.DM[,,"M","atx"] <- predatt.apc( m.md.apc, nd, delta=0.025, dfac=0.7 )$pr.resp
> Mu.DM[,,"F","atx"] <- predatt.apc( f.md.apc, nd, delta=0.025, dfac=0.7 )$pr.resp
> Mu.DM[,,"M","i20"] <- predatt.apc( m.md.apc, nd, delta=0.020, dfac=0 )$pr.resp
> Mu.DM[,,"F","i20"] <- predatt.apc( f.md.apc, nd, delta=0.020, dfac=0 )$pr.resp
> Mu.DM[,,"M","i25"] <- predatt.apc( m.md.apc, nd, delta=0.025, dfac=0 )$pr.resp
> Mu.DM[,,"F","i25"] <- predatt.apc( f.md.apc, nd, delta=0.025, dfac=0 )$pr.resp
> Mu.DM[,,"M","i30"] <- predatt.apc( m.md.apc, nd, delta=0.030, dfac=0 )$pr.resp
> Mu.DM[,,"F","i30"] <- predatt.apc( f.md.apc, nd, delta=0.030, dfac=0 )$pr.resp
```

2.4.2 Persons without diabetes

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

```
> ( kmw.a <- with( mnDdat, c( 5, 15,
+                         quantile( 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 50.66667 64.33333 71.66667 77.33333 81.33333 85.33333 88.66667 93.66667
> ( kmw.p <- with( mnDdat, quantile( rep(P  ,D), qn(nk.p) ) ) )
```

```
8.333333%       25% 41.66667% 58.33333%       75% 91.66667%
 1997.333  2000.667  2003.667  2006.667  2010.333  2014.333
> ( kmw.c <- with( mnDdat, quantile( rep(P-A,D), qn(nk.c) ) ) )
   6.25%   18.75%   31.25%   43.75%   56.25%   68.75%   81.25%   93.75%
1909.667 1915.667 1920.333 1924.333 1928.667 1934.333 1942.667 1955.667
> m.mw.ap <- glm( D ~ Ns(A,knots=kmw.a) + Ns(P,knots=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.W[,,"M","ap" ] <- predict.glm( m.mw.ap , type="response", newdata=nd )
> Mu.W[,,"F","ap" ] <- predict.glm( f.mw.ap , type="response", newdata=nd )
> Mu.W[,,"M","apc"] <- predict.glm( m.mw.apc, type="response", newdata=nd )
> Mu.W[,,"F","apc"] <- predict.glm( f.mw.apc, type="response", newdata=nd )
> Mu.W[,,"M","att"] <- predatt.apc( m.mw.apc, nd, )$pr.resp
> Mu.W[,,"F","att"] <- predatt.apc( f.mw.apc, nd, )$pr.resp
> Mu.W[,,"M","atx"] <- predatt.apc( m.mw.apc, nd, delta=0.025, dfac=0.7 )$pr.resp
> Mu.W[,,"F","atx"] <- predatt.apc( f.mw.apc, nd, delta=0.025, dfac=0.7 )$pr.resp
> Mu.W[,,"M","i20"] <- predatt.apc( m.mw.apc, nd, delta=0.020, dfac=0 )$pr.resp
> Mu.W[,,"F","i20"] <- predatt.apc( f.mw.apc, nd, delta=0.020, dfac=0 )$pr.resp
> Mu.W[,,"M","i25"] <- predatt.apc( m.mw.apc, nd, delta=0.025, dfac=0 )$pr.resp
> Mu.W[,,"F","i25"] <- predatt.apc( f.mw.apc, nd, delta=0.025, dfac=0 )$pr.resp
> Mu.W[,,"M","i30"] <- predatt.apc( m.mw.apc, nd, delta=0.030, dfac=0 )$pr.resp
> Mu.W[,,"F","i30"] <- predatt.apc( f.mw.apc, nd, delta=0.030, dfac=0 )$pr.resp
```

2.5 Average trends

The average annual trends in all of the rates 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 ), 1 )
                           exp(Est.) 2.5% 97.5%
DM incidence change     Men       2.2  2.2   2.3
                        Women     3.5  3.4   3.6
Mortality change,   DM: Men       0.0  0.0   0.0
                        Women     0.0  0.0   0.0
Mortality change, Well: Men       0.0  0.0   0.0
                        Women     0.0  0.0   0.0
```

Thus it appears that the incidence rates of diabetes 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 perons without.

For convenience of calculations and for subsequent use, we save the estimated rates and other quantities of interest:

```
> save( Lambda, Mu.W, Mu.DM, a.pt, p.pt, t.pt, int, qn, file="../data/rateEsts.Rda" )
> load( file="../data/rateEsts.Rda" )
```

2.6 Time trends of estimated 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:

A brief overview of the mortality and incidence rates over time:

```
> 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"
> 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" )
> yticks <- outer( 1:9, 10^(-2:1) )[1:30]
> rpl <- function( Lambda, sx, yl ){
+ matplot( pts, t(Lambda[agr,1:240,sx,"apc"])*1000,
+          log="y", yaxt="n", ylim=c(0.01,30), ylab="", xaxt="n",
+          lty=1, lwd=5, type="l",
+          col=gray(4:11/13) )#if(sx=="M") "blue" else "red" )
+ mtext( side=2, yl, line=2.5, las=0 )
+ }
> rpl( Lambda, "M", "DM incidence per 1000 PY" )
>                           axis( side=2 ) ; axis( side=2, at=yticks, labels=NA )
> text( 1996, 30, "Men", adj=c(0,1), cex=1.5 )
> rpl( Lambda, "F", "" )
> text( 1996, 30, "Women", adj=c(0,1), cex=1.5 )
> rpl(  Mu.W , "M", "Population mortality per 1000 PY" )
>                           axis( side=2 ) ; axis( side=2, at=yticks, labels=NA )
> rpl(  Mu.W , "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 )
```

2.7 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.W` and `Mu.DM`. However, as seen in figure 2.6 the predicted decline in diabetes rates is presumably way too dramatic.

```
> pts <- as.numeric( dimnames(Lambda)[[2]] )
> ( 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"
```

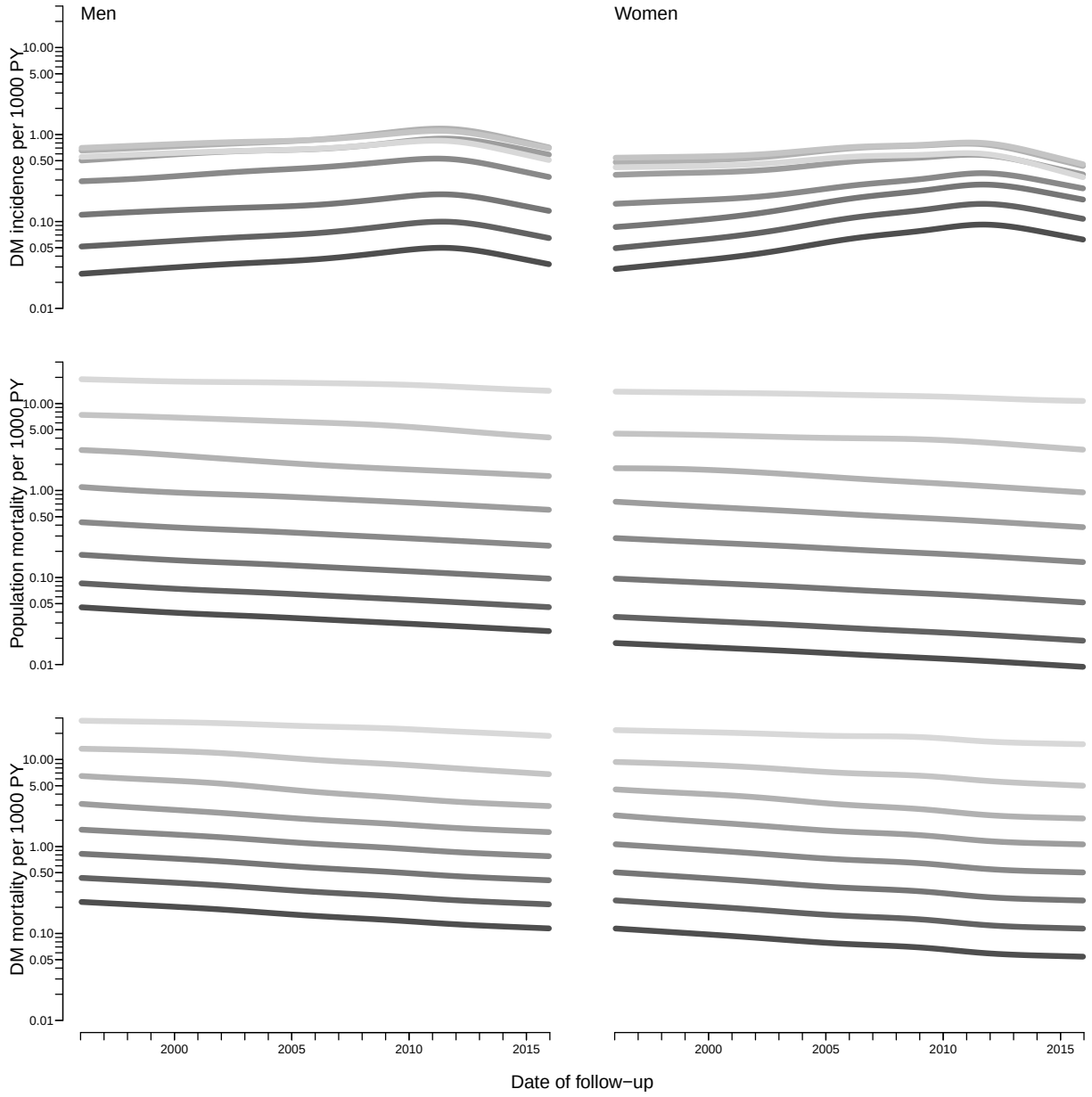

Figure 2.5: *Trends in incidence and mortality rates for ages 20 (darkest), 30, . . . , 90 (lightest), as estimated from the age-period-cohort models.*

```
> 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" )
> yticks <- outer( 1:9, 10^(-2:1) )[1:30]
> rpl <-
+ function( Lambda, sx, yl, inc=FALSE )
+ {
+ matplot( pts, t(Lambda[agr,,sx,"apc"])*1000,
+          log="y", yaxt="n", ylim=c(0.01,30), xlim=c(1996,2030), ylab="", xaxt="n",
+          lty=1, lwd=5, type="l", col=gray(4:11/13) )
+ matlines( pts, t(Lambda[agr,,sx,"att"])*1000,
+           lty=1, lwd=1, type="l", col="black" )
+ matlines( pts, t(Lambda[agr,,sx,"atx"])*1000,
+           lty=1, lwd=1, type="l", col="forestgreen" )
+ if( inc )
```

```
+ matlines( pts, t(Lambda[agr,,sx,"i25"])*1000,
+           lty=1, lwd=1, type="l", col="red" )
+ abline( v=2016, lty=3, col=gray(0.6) )
+ mtext( side=2, yl, line=2.5, las=0 )
+ }
> rpl( Lambda, "M", "DM incidence per 1000 PY", inc=TRUE )
>                          axis( side=2 ) ; axis( side=2, at=yticks, labels=NA, tcl=-0.3 )
> text( 1996, 30, "Men", adj=c(0,1), cex=1.5 )
> rpl( Lambda, "F", "", inc=TRUE )
> text( 1996, 30, "Women", adj=c(0,1), cex=1.5 )
> rpl(  Mu.W , "M", "Population mortality per 1000 PY" )
>                          axis( side=2 ) ; axis( side=2, at=yticks, labels=NA, tcl=-0.3 )
> rpl(  Mu.W , "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=1996:2030, labels=NA, tcl=-0.3
> rpl(  Mu.DM, "F", "" ) ; axis( side=1 ) ; axis( side=1, at=1996:2030, labels=NA, tcl=-0.3
> mtext( "Date of follow-up", side=1, line=2, outer=TRUE )
```

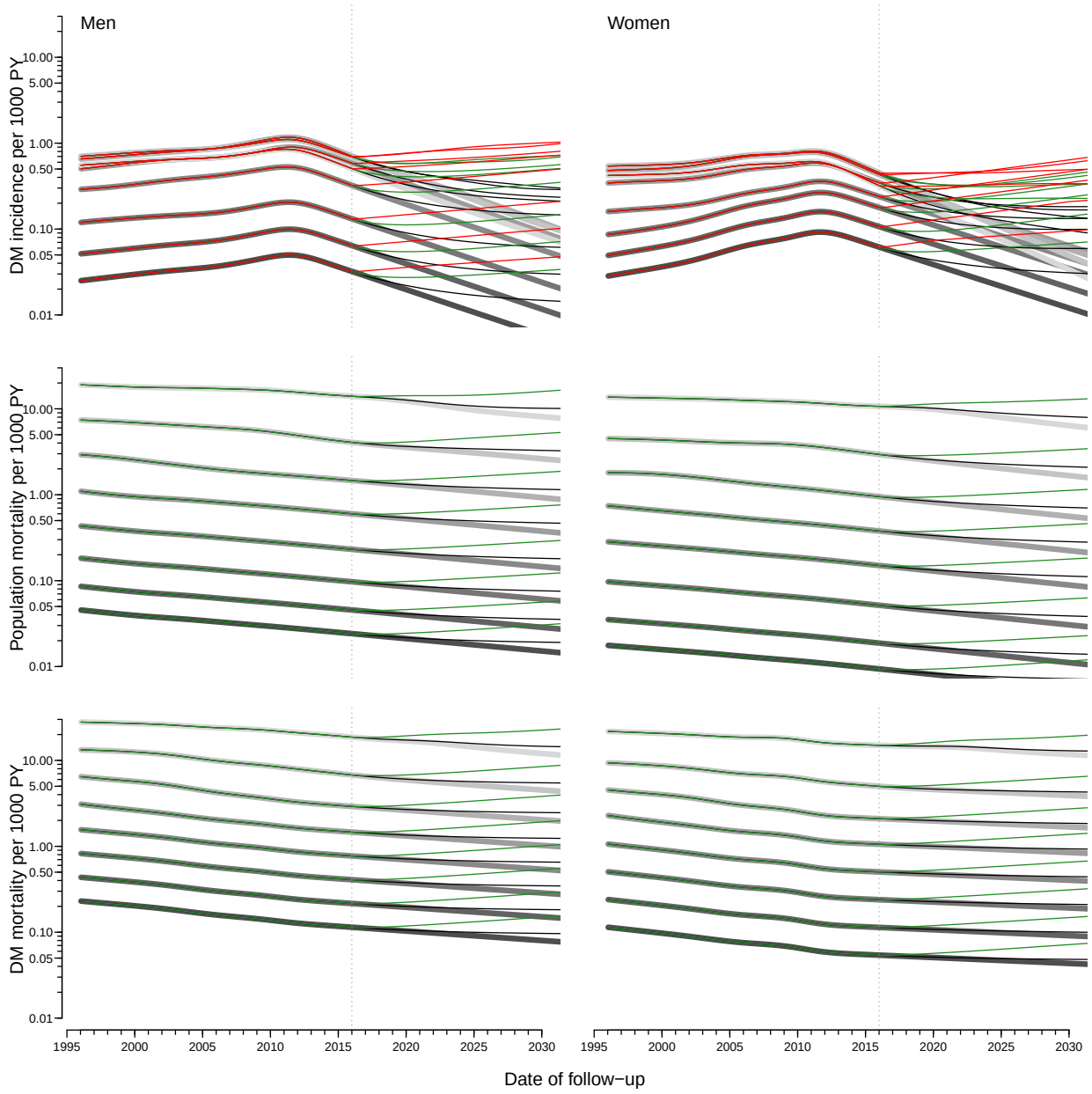

Figure 2.6: *Trends in incidence and mortality rates for ages 20 (darkest), 30, . . . , 90 (lightest), as estimated from the age-period-cohort models and predicted by naive extrapolation of the natural splines. The vertical dotted lines indicate the end of available data, and the thin overlaid lines represent the rate predictions based on attenuated period- and cohort-effects. The red lines are predictions assuming an annual increase in period effects of 2.5%/year — only used for the incidence rates.*

Chapter 3

Predicting prevalence of diabetes

In this chapter we use the predicted rates for the period 2016–2030, under two different scenarios:

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

  This will give a prediction of the numbers which is the least credible.

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

3.1 Predicted rates

We will start with the observed (smoothed) age-specific prevalences at 2016-01-01 and then use the three different scenarios laid out above to predict the prevalences each year till 2030.

First we load the estimated / predicted rates

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

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 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/prFit.Rda" )
> str( pr.fit )
 num [1:2, 1:1200, 1:21] 0.000631 0.000578 0.000635 0.000582 0.00064 ...
 - 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" ...
```

`TR` — array of transition probabilities between states Well and DM and Death. Transition probabilities are computed under the 3 different scenarios combining mortality and incidence rates either as they actually developed 1996–2015. These refer to intervals of length `int` and are therefore labeled on the period dimension by the midpoint of these, a total of 15/`int`.

`prv` — array of predicted prevalences based on the initial prevalences at 2016-01-01 and the transition probabilities as put in `TR`. The scenario dimension refers to the 3 scenarios: "lin", "att" and "fix". Moreover, the period dimension is expanded by one relative to that in `TR`, since this refer to points in time and not time intervals.

`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.

3.2 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. But we will only need the rates from 2016 and onward, so we restrict the arrays with the rates to this period:

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

```
> rLambda <- Lambda[,-(1:240),,]
> rMu.W   <-  Mu.W [,-(1:240),,]
> rMu.DM  <-  Mu.DM[,-(1:240),,]
> states <- c("Well","DM")
> TR <- NArray( c( dimnames( rLambda ),
+                  list( from = states,
+                        to = states ) ) )
> dimnames( TR )[[4]][1] <- "fix"
> str( TR )
 logi [1:1200, 1:288, 1:2, 1:6, 1:2, 1:2] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 6
  ..$ 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] "fix" "apc" "att" "i20" ...
  ..$ from: chr [1:2] "Well" "DM"
  ..$ to  : chr [1:2] "Well" "DM"
```

The situation where both the mortality rates and incidence rates are fixed at the 2016 level ("`fix`") 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`, in steps of `int` (in this case 0.0833 year).

```
> TR[,,,,"Well","Well"] <-     exp(-rLambda-rMu.W)
> TR[,,,,"Well","DM"  ] <- 1 - exp(-rLambda)
> TR[,,,,"DM"  ,"Well"] <- 0
> TR[,,,,"DM"  ,"DM"  ] <-     exp(-rMu.DM)
```

Note that we here fill in the transition probabilities from the age-period model in the `fix` slot if the 4th dimension of `TR`, but we overwrite this with the rates as of 2016:

```
> TR[,,,"fix","Well","Well"] <-     exp(-rLambda[,rep(1,dim(TR)[2]),,"apc"]
+                                        -rMu.W[,rep(1,dim(TR)[2]),,"apc"])
> TR[,,,"fix","Well","DM"  ] <- 1 - exp(-rLambda[,rep(1,dim(TR)[2]),,"apc"])
> TR[,,,"fix","DM"  ,"Well"] <- 0
> TR[,,,"fix","DM"  ,"DM"  ] <-     exp(- rMu.DM[,rep(1,dim(TR)[2]),,"apc"])
```

Likewise we fill in the entries referring to the three scenarios of increasing incidence rates — note that it is only for the incidence rates we impose an increase in rates:

```
> TR[,,,"i20","Well","Well"] <-     exp(-rLambda[,,,"i20"]-rMu.W[,,,"att"])
> TR[,,,"i20","Well","DM"  ] <- 1 - exp(-rLambda[,,,"i20"])
> TR[,,,"i20","DM"  ,"Well"] <- 0
> TR[,,,"i20","DM"  ,"DM"  ] <-     exp(- rMu.DM[,,,"att"])
> TR[,,,"i25","Well","Well"] <-     exp(-rLambda[,,,"i25"]-rMu.W[,,,"att"])
> TR[,,,"i25","Well","DM"  ] <- 1 - exp(-rLambda[,,,"i25"])
> TR[,,,"i25","DM"  ,"Well"] <- 0
> TR[,,,"i25","DM"  ,"DM"  ] <-     exp(- rMu.DM[,,,"att"])
> TR[,,,"i30","Well","Well"] <-     exp(-rLambda[,,,"i30"]-rMu.W[,,,"att"])
> TR[,,,"i30","Well","DM"  ] <- 1 - exp(-rLambda[,,,"i30"])
> TR[,,,"i30","DM"  ,"Well"] <- 0
> TR[,,,"i30","DM"  ,"DM"  ] <-     exp(- rMu.DM[,,,"att"])
```

Finally, 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

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`, a total of about 16 mil. numbers:

```
> str( TR )
 num [1:1200, 1:288, 1:2, 1:6, 1:2, 1:2] 1 1 1 1 1 ...
 - attr(*, "dimnames")=List of 6
  ..$ 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] "fix" "apc" "att" "i20" ...
  ..$ from: chr [1:2] "Well" "DM"
  ..$ to  : chr [1:2] "Well" "DM"
> prod( dim(TR) )
[1] 16588800
> save( TR, file="../data/TRf.Rda" )
```

3.3 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 1995, 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 2016-01-01).

```
> dpr <- dimnames(TR)[1:4]
> names( dpr )[2] <- "t"
> dpr[["t"]] <- t.pt[-(1:240)]
> prv <- NArray( dpr )
> str( prv )
 logi [1:1200, 1:289, 1:2, 1:6] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.291666666666667
  ..$ t  : chr [1:289] "2016" "2016.08333333333" "2016.16666666667" "2016.25" ...
  ..$ sex: chr [1:2] "M" "F"
  ..$ mod: chr [1:6] "fix" "apc" "att" "i20" ...
> prod( dim(prv) )
[1] 4161600
```

Thus we must make a loop that updates the prevalences at 2016-01-01 to those at subsequent times, 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 (model and scenario).

```
> load( file="../data/prFit.Rda" )
> str( pr.fit )
 num [1:2, 1:1200, 1:21] 0.000631 0.000578 0.000635 0.000582 0.00064 ...
 - attr(*, "dimnames")=List of 3
  ..$ sex: chr [1:2] "M" "F"
  ..$ A  : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.291666666666667
  ..$ P  : chr [1:21] "1996" "1997" "1998" "1999" ...
> # Smoothed prevalences at 2016-01-01 - the starting values
> for( sx in c("M","F") ) prv[,"2016",sx,] <- pr.fit[sx,,"2016"]
> # Prevalences at age 0
> prv[1,,,] <- 0
> tt <- prv[c(1:2,floor(dim(prv)[1]/1.5)+1:3),1:3,,]
> dimnames( tt )[[1]] <- round( as.numeric(dimnames( tt )[[1]]), 2 )
> dimnames( tt )[[2]] <- round( as.numeric(dimnames( tt )[[2]]), 2 )
> round( ftable( tt, col.vars=3:4 )*100, 1 )
             sex    M                            F
             mod  fix  apc  att  i20  i25  i30  fix  apc  att  i20  i25  i30
a     t
0.04  2016         0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
      2016.08      0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
      2016.17      0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.12  2016         0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1
      2016.08       NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
      2016.17       NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
66.71 2016        14.2 14.2 14.2 14.2 14.2 14.2  9.6  9.6  9.6  9.6  9.6  9.6
      2016.08       NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
      2016.17       NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
66.79 2016        14.2 14.2 14.2 14.2 14.2 14.2  9.6  9.6  9.6  9.6  9.6  9.6
      2016.08       NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
      2016.17       NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
66.88 2016        14.2 14.2 14.2 14.2 14.2 14.2  9.6  9.6  9.6  9.6  9.6  9.6
      2016.08       NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
      2016.17       NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
```

So now it is checked that we have put the initial values correctly into `prv`. Then we can 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,,,"Well","DM"  ] ) /
+   (    prv[ia,ip,,]  * TR[ia,ip,,,"DM"  ,"DM"  ]
+    +(1-prv[ia,ip,,]) * TR[ia,ip,,,"Well","DM"  ]
+    +(1-prv[ia,ip,,]) * TR[ia,ip,,,"Well","Well"] )
+             )
   user  system elapsed
  12.69    0.00   12.69
```

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.

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

```
> round( prv[1:10,1:2,1,1,drop=F]*100, 3 )
, , sex = M, mod = fix

                    t
a                    2016 2016.08333333333
  0.0416666666666667 0.000             0.000
  0.125              0.111             0.001
  0.208333333333333  0.111             0.112
  0.291666666666667  0.112             0.112
  0.375              0.113             0.113
  0.458333333333333  0.113             0.114
  0.541666666666667  0.114             0.114
  0.625              0.115             0.115
  0.708333333333333  0.116             0.116
  0.791666666666667  0.116             0.117
```

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

3.4 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 actual 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 (2016–2030): `prv`. This is because we have the predicted population size in 1-year classes.

```
> dn <- dimnames(prv)
> dn[[1]] <- 0:99
> dn[[2]] <- 2016:2030
> prn <- NArray( dn )
> table( (prv>0) + (prv>=0) )
      1       2
   3468 4158132
> str( prv )
 num [1:1200, 1:289, 1:2, 1:6] 0 0.00111 0.00111 0.00112 0.00113 ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:1200] "0.0416666666666667" "0.125" "0.208333333333333" "0.291666666666667
  ..$ t  : chr [1:289] "2016" "2016.08333333333" "2016.16666666667" "2016.25" ...
  ..$ sex: chr [1:2] "M" "F"
  ..$ mod: chr [1:6] "fix" "apc" "att" "i20" ...
> str( prn )
 logi [1:100, 1:15, 1:2, 1:6] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:100] "0" "1" "2" "3" ...
  ..$ t  : chr [1:15] "2016" "2017" "2018" "2019" ...
  ..$ sex: chr [1:2] "M" "F"
  ..$ mod: chr [1:6] "fix" "apc" "att" "i20" ...
```

In order to fill in the numbers we use the estimates age-specific prevalences at 1st January each year, that is at the dates 2016-01-01,...,2030-01-01 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. 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:

```
> # Find the dates in the predicted prevalences prv that matches the
> # dates in prn where empirical rates are available.
> prv.p <- as.numeric( dimnames(prv)[["t"]] )
> prn.p <- as.numeric( dimnames(prn)[["t"]] )
>  wh.p <- match( prn.p, prv.p )
> if( any(is.na(wh.p)) ) # Need to find approximate dates if they do not match
+ for( ip in 1:length(prn.p) )
+    {
+    dd <- abs( prn.p[ip]-prv.p )
+    wh.p[ip] <- (1:length(dd))[dd==min(dd)]
+    }
> wh.p
 [1]   1  13  25  37  49  61  73  85  97 109 121 133 145 157 169
> prv <- pmax( prv, 0 )
> # Ages in the two arrays
> prv.a <- as.numeric( dimnames(prv)[["a"]] )
> prn.a <- as.numeric( dimnames(prn)[["a"]] )
> for( ip in 1:length(wh.p) )
```

```
+ for( ia in 1:length(prn.a) )
+    {
+ wh.a <- which( prn.a[ia]==floor(prv.a) )
+ prn[ia,ip,,] <- apply( prv[wh.a,wh.p[ip],,], 2:3, mean )
+    }
> range( prv )
[1] 0.0000000 0.2293312
> range( prn )
[1] 6.458973e-06 2.097677e-01
```

Now `prn` contains the prevalences (as fractions) for 100 age classes and the 15 dates. We need to multiply these prevalences by the population figures at these times.

3.5 Population forecast from DST

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=TRUE )
> bef$sex <- ifelse( bef$sex==" ", NA, bef$sex )
> for( j in 1:3 )
+ for( i in 2:nrow(bef) )
+     if( is.na(bef[i,j]) ) bef[i,j] <- bef[i-1,j]
> table( bef$sex )
   2    3
3277 3277
> bef$sex <- factor( bef$sex, levels=3:2, labels=c("M","F") )
> bef <- subset( bef, !is.na(N) )
> addmargins( xtabs( N ~ P + sex, data = bef ), 2 )
      sex
P            M       F     Sum
  2016 2837887 2869364 5707251
  2017 2863749 2891251 5755000
  2018 2887236 2911306 5798542
  2019 2906801 2928091 5834892
  2020 2923339 2942471 5865810
  2021 2938021 2955444 5893465
  2022 2951822 2967966 5919788
  2023 2965270 2980437 5945707
  2024 2978491 2992896 5971387
  2025 2991402 3005238 5996640
  2026 3003925 3017277 6021202
  2027 3016021 3028944 6044965
  2028 3027617 3040154 6067771
  2029 3038690 3050838 6089528
  2030 3049243 3060972 6110215
  2031 3059248 3070527 6129775
  2032 3068731 3079506 6148237
  2033 3077703 3087895 6165598
  2034 3086217 3095728 6181945
  2035 3094266 3102972 6197238
  2036 3101872 3109674 6211546
  2037 3109101 3115831 6224932
```

```
  2038 3115961 3121489 6237450
  2039 3122496 3126666 6249162
  2040 3128698 3131429 6260127
```

```
> str( bef )
```

```
'data.frame':        6300 obs. of  4 variables:
 $ sex: Factor w/ 2 levels "M","F": 1 1 1 1 1 1 1 1 1 1 ...
 $ A  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ P  : int  2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 ...
 $ N  : int  30049 30897 31660 32444 33259 34197 35259 36232 36895 37240 ...
```

We need the population figures in an array of the same shape as (some of the dimensions of) `prv`

```
> pop <- xtabs( N ~ A + P + sex,
+               data = subset( bef, A<100 & P<2031 ) )
> str( pop )
 int [1:100, 1:15, 1:2] 30049 29750 29459 30834 31038 33643 33495 34918 34180 34688 ...
 - attr(*, "dimnames")=List of 3
  ..$ A  : chr [1:100] "0" "1" "2" "3" ...
  ..$ P  : chr [1:15] "2016" "2017" "2018" "2019" ...
  ..$ sex: chr [1:2] "M" "F"
 - attr(*, "class")= chr [1:2] "xtabs" "table"
 - attr(*, "call")= language xtabs(formula = N ~ A + P + sex, data = subset(bef, A < 100 &
```

```
> str( prn )
```

```
 num [1:100, 1:15, 1:2, 1:6] 0.00105 0.00123 0.00132 0.00142 0.00154 ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:100] "0" "1" "2" "3" ...
  ..$ t  : chr [1:15] "2016" "2017" "2018" "2019" ...
  ..$ sex: chr [1:2] "M" "F"
  ..$ mod: chr [1:6] "fix" "apc" "att" "i20" ...
```

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

First we draw simple population pyramids of the age-distribution of the diabetes patients in Denmark, as predicted under different scenarios:

```
> # 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"
> lim <- 6
> clr <- c("red","blue")
> draw.dmp <-
+ function(pp,wh)
+ {
+ par( mar=c(3,3,3,0), mgp=c(3,1,0)/1.6, las=1 )
+ barplot( height=t( cbind( -dmp[,pp,"M",wh],
+                           dmp[,pp,"M",wh],
+                           dmp[,pp,"F",wh] ) )/ 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=1, at=seq(-lim,lim,0.5), labels=NA, tcl=-0.3 )
+ 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=1.2, cex=1.0, font=2 )
+ mtext( formatC(sum(dmp[,pp,"M",wh]),0,format="f",big.mark=","),
+        at=-1, col="blue", line=0, cex=0.99, adj=1 )
+ mtext( formatC(sum(dmp[,pp,"F",wh]),0,format="f",big.mark=","),
+        at= 1, col="red" , line=0, cex=0.99, adj=0 )
+ mtext( "N", at=0, line=0, cex=0.99 )
+ }
> pdf( "pred-inc-film.pdf", width=8, height=6 )
> for( pp in paste(2016:2030) ) draw.dmp(pp,"fix")
> dev.off()
null device
          1

> pdf( "pred-apc-film.pdf", width=8, height=6 )
> for( pp in paste(2016:2030) ) draw.dmp(pp,"apc")
> dev.off()

null device
          1

> pdf( "pred-att-film.pdf", width=8, height=6 )
> for( pp in paste(2016:2030) ) draw.dmp(pp,"att")
> dev.off()

null device
          1

> for( pp in paste(2016:2030) )
+ {
+ pdf( paste("pred-att-", pp, ".pdf", sep=""), width=8, height=6 )
+ draw.dmp(pp,"att")
+ dev.off()
+ }



> par( mfrow=c(2,2), mar=c(3,3,0,0), oma=c(0,0,2,0), mgp=c(3,1,0)/1.6, las=1 )
> draw.dmp("2016","att")
> draw.dmp("2020","att")
> draw.dmp("2025","att")
> draw.dmp("2030","att")
> mtext( "Incidence rate decrease attennuates from 2016", side=3, line=0, outer=TRUE )



> par( mfrow=c(2,2), mar=c(3,3,0,0), oma=c(0,0,2,0), mgp=c(3,1,0)/1.6, las=1 )
> draw.dmp("2016","fix")
> draw.dmp("2020","fix")
> draw.dmp("2025","fix")
> draw.dmp("2030","fix")
> mtext( "Incidence and mortality rates constant from 2016", side=3, line=0, outer=TRUE )
```

```
> par( mfrow=c(2,2), mar=c(3,3,0,0), oma=c(0,0,2,0), mgp=c(3,1,0)/1.6, las=1 )
> draw.dmp("2016","apc")
> draw.dmp("2020","apc")
> draw.dmp("2025","apc")
> draw.dmp("2030","apc")
> mtext( "Naive linear prediction from 2016", side=3, line=0, outer=TRUE )


> par( mfrow=c(2,2), mar=c(3,3,0,0), oma=c(0,0,2,0), mgp=c(3,1,0)/1.6, las=1 )
> draw.dmp("2016","i20")
> draw.dmp("2020","i20")
> draw.dmp("2025","i20")
> draw.dmp("2030","i20")
> mtext( "Incidence rates increase 2.0%/y from 2016, mortality decrease", side=3, line=0, o


> par( mfrow=c(2,2), mar=c(3,3,0,0), oma=c(0,0,2,0), mgp=c(3,1,0)/1.6, las=1 )
> draw.dmp("2016","i25")
> draw.dmp("2020","i25")
> draw.dmp("2025","i25")
> draw.dmp("2030","i25")
> mtext( "Incidence rates increase 2.5%/y from 2016, mortality decrease", side=3, line=0, o


> par( mfrow=c(2,2), mar=c(3,3,0,0), oma=c(0,0,2,0), mgp=c(3,1,0)/1.6, las=1 )
> draw.dmp("2016","i30")
> draw.dmp("2020","i30")
> draw.dmp("2025","i30")
> draw.dmp("2030","i30")
> mtext( "Incidence rates increase 3.0%/y from 2016, mortality decrease", side=3, line=0, o
```

3.6 Timetrend in prevalent number of DM patients

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

```
> str( dmp )
 num [1:100, 1:15, 1:2, 1:6] 31.4 36.5 38.9 43.9 47.7 ...
 - attr(*, "dimnames")=List of 4
  ..$ a  : chr [1:100] "0" "1" "2" "3" ...
  ..$ t  : chr [1:15] "2016" "2017" "2018" "2019" ...
  ..$ sex: chr [1:2] "M" "F"
  ..$ mod: chr [1:6] "fix" "apc" "att" "i20" ...
> dimnames( dmp )[[4]]
[1] "fix" "apc" "att" "i20" "i25" "i30"
> ftable( addmargins( round( apply( dmp[,,,4:6], 2:4, sum ) ), 2 ),
+         col.vars=3:2 )
     mod    i20                  i25                  i30
     sex      M      F    Sum      M      F    Sum      M      F    Sum
t
2016     152952 127134 280086 152952 127134 280086 152952 127134 280086
2017     154975 128303 283278 154993 128316 283309 155011 128330 283341
2018     157131 129582 286713 157204 129637 286841 157279 129692 286971
2019     159392 130954 290346 159560 131079 290639 159730 131206 290936
2020     161762 132428 294190 162066 132655 294721 162373 132884 295257
2021     164246 134009 298255 164728 134369 299097 165217 134735 299952
```

```
2022     166850 135715 302565 167553 136242 303795 168271 136780 305051
2023     169581 137554 307135 170552 138284 308836 171547 139032 310579
2024     172443 139525 311968 173729 140495 314224 175051 141492 316543
2025     175442 141637 317079 177094 142887 319981 178796 144176 322972
2026     178578 143893 322471 180645 145463 326108 182784 147088 329872
2027     181854 146297 328151 184390 148229 332619 187023 150236 337259
2028     185267 148851 334118 188327 151190 339517 191513 153628 345141
2029     188820 151556 340376 192458 154349 346807 196261 157269 353530
2030     192516 154417 346933 196792 157710 354502 201275 161166 362441
```

```
> addmargins( round( apply( dmp[,c(1,5,10,15),,"i25"], 2:3, sum ) ), 2 )
```

```
     sex
t           M      F    Sum
  2016 152952 127134 280086
  2020 162066 132655 294721
  2025 177094 142887 319981
  2030 196792 157710 354502
```

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",]
> DMcum <- apply( DMall, 2:3, cumsum )
> DMcum <- DMcum[c(1,1:100),,]
> DMcum[1,,] <- 0
> DMcum <- DMcum/1000
> str( DMcum )
 num [1:101, 1:15, 1:6] 0 0.0635 0.137 0.2157 0.3039 ...
 - attr(*, "dimnames")=List of 3
  ..$ a  : chr [1:101] "0" "0" "1" "2" ...
  ..$ t  : chr [1:15] "2016" "2017" "2018" "2019" ...
  ..$ mod: chr [1:6] "fix" "apc" "att" "i20" ...
```

Finally we can plot the predicted numbers from the different scenarios:

```
> range( DMcum )
[1]   0.0000 362.4401
```

```
> ryr <- c(2016:2030,2030:2016)
> leg <- c("All rates fixed at 2016 level",
+          "Linear projection from 2016",
+          "Attenuated linear projection",
+          "DM incidence increasing 2.0%/y",
+          "DM incidence increasing 2.5%/y",
+          "DM incidence increasing 3.0%/y")
> par( mfrow=c(3,2), mar=c(2,1,0,3), mgp=c(3,1,0)/1.6, las=1, bty="n" )
> for( j in c(2,3,1,4:6) )
+    {
+ plot( NA,
+       xlim=c(2015.5,2030), xlab="", xaxt="n", xaxs="i",
+       ylim=c(0,420), yaxs="i", yaxt="n", ylab="" )
+ axis( side=4, at=0:4*100 )
+ axis( side=4, at=seq(0,4,1/4)*100, labels=NA, tcl=-0.3 )
+ axis( side=1, at=2015+1:3*5 )
+ axis( side=1, at=2016:2030, labels=NA, tcl=-0.3 )
+ for( i in 1:10 ) polygon( ryr, c( DMcum[1+(i-1)*10,,j],
+                                   rev( DMcum[1+ i   *10,,j] ) ),
+                           col=gray( (17-i)/18 ), border=gray(0.8) ) #"transparent" )
+ abline( h=seq(50,400,50), v=c(2020,2025), col=gray(1), lty="16" )
```

```
+ for( i in seq(55,85,10) ) text( 2029, DMcum[paste(i),"2029",j],
+                                 paste( i-5,"-",i+4,sep="") )
+ text( 2016, 400, paste( dimnames(DMcum)[[3]][j], ": ", leg[j], sep="" ), adj=c(0,1) )
+    }
```

From figure 3.7 it appears that it is the decreasing incidence rates of diabetes that carries the major differences of more than 100,000 patients in 2030. 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. Tabulations in the SAS-programs `08-mkFU` (p. **??**) and `09-mkPr` (p. **??**), show that there is no particular break in data where the decrease occur.

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 2012–2015.

However, even the quite brutal assumption of a pick up of increasing DM incidence rates by 2.5% per year will not bring the predicted number of patients over 400,000 in 2030. So to say that the number of diabetes patients is less than this in 2030 seems to be a fairly safe bet.

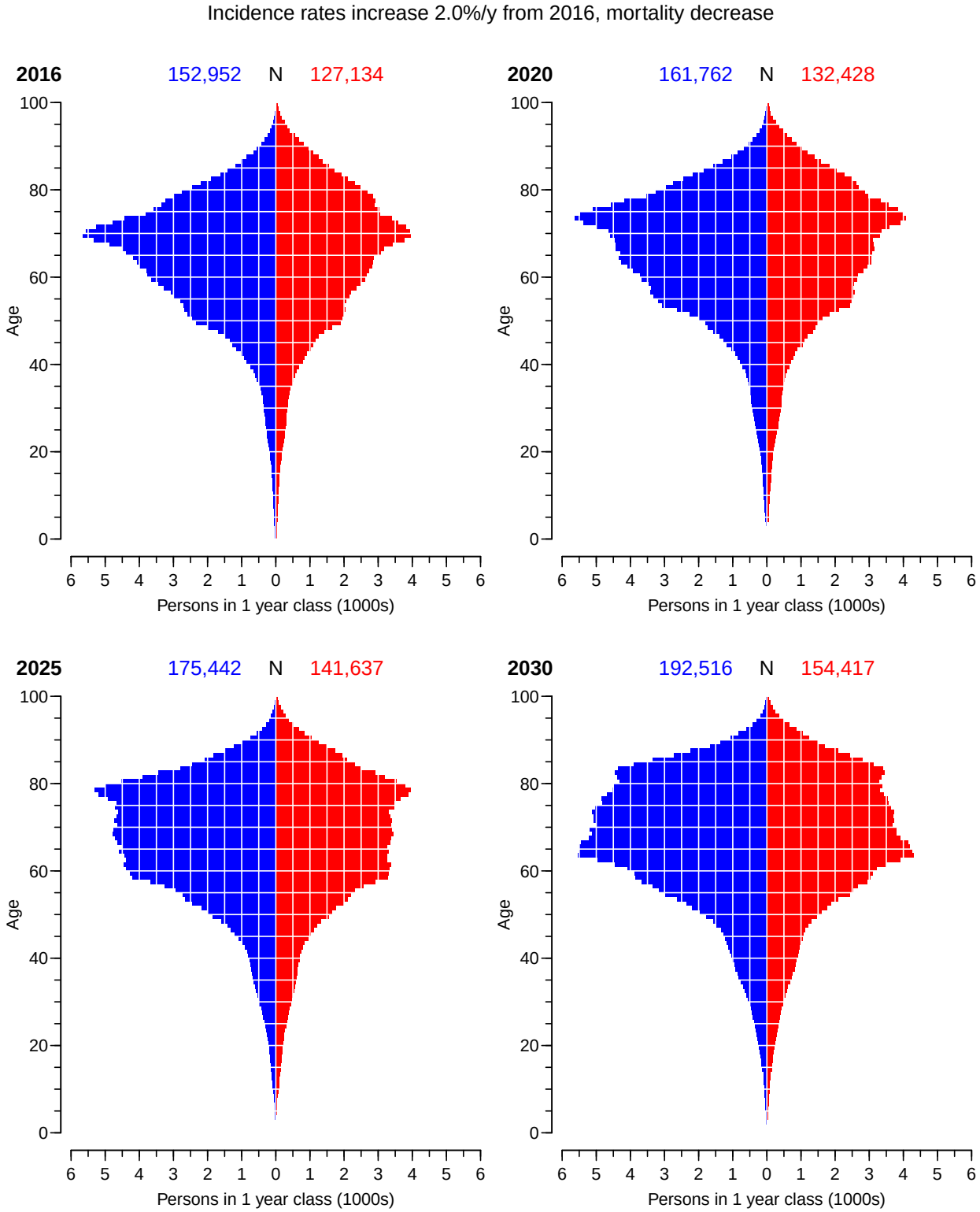

Figure 3.1: *Empirical age-distribution of the diabetes cases in Denmark based on a unchanged mortality and increasing incidence rates (2.0%/y) from 2016.*

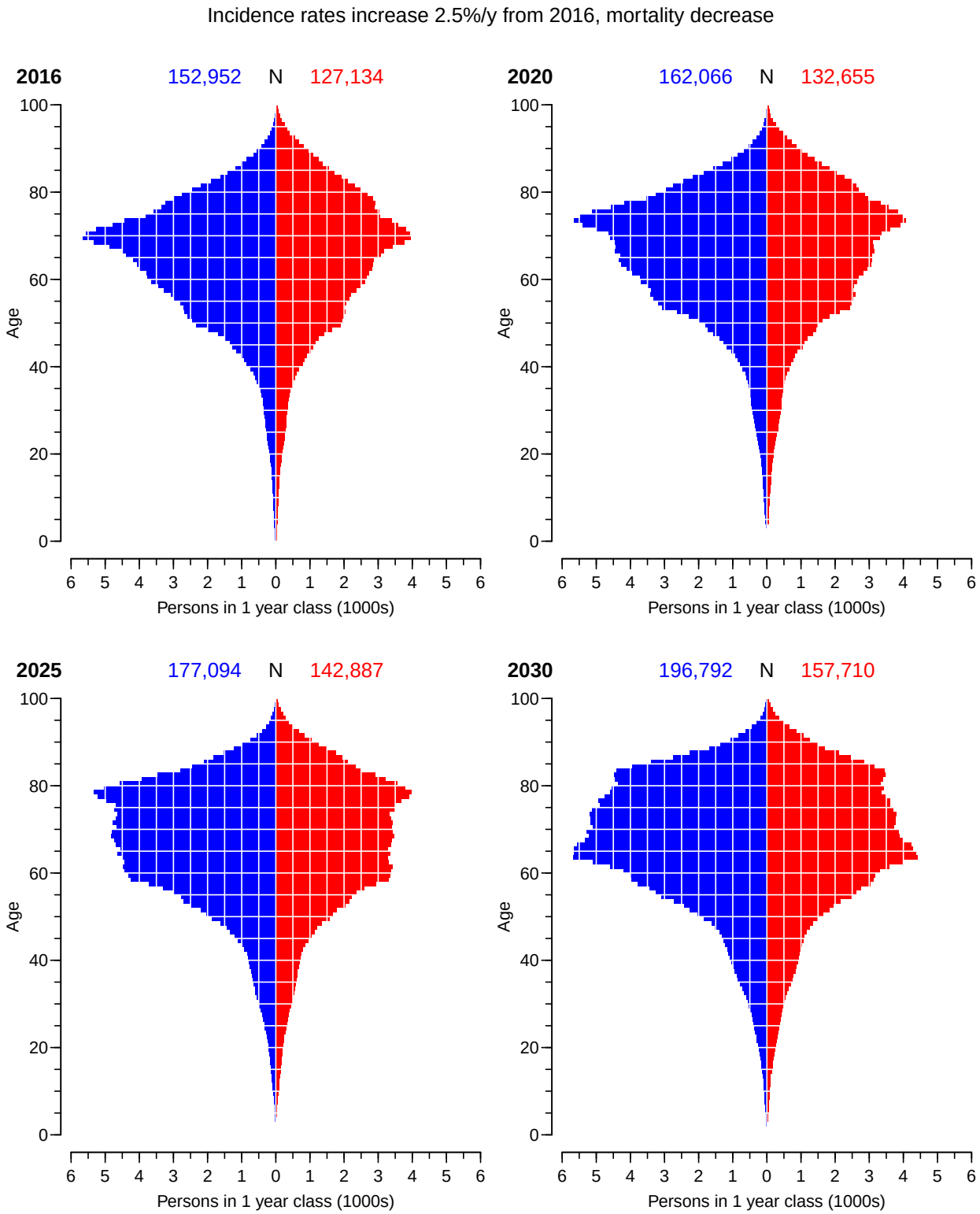

Figure 3.2: *Empirical age-distribution of the diabetes cases in Denmark based on a unchanged mortality and increasing incidence rates (2.5%/y) from 2016.*

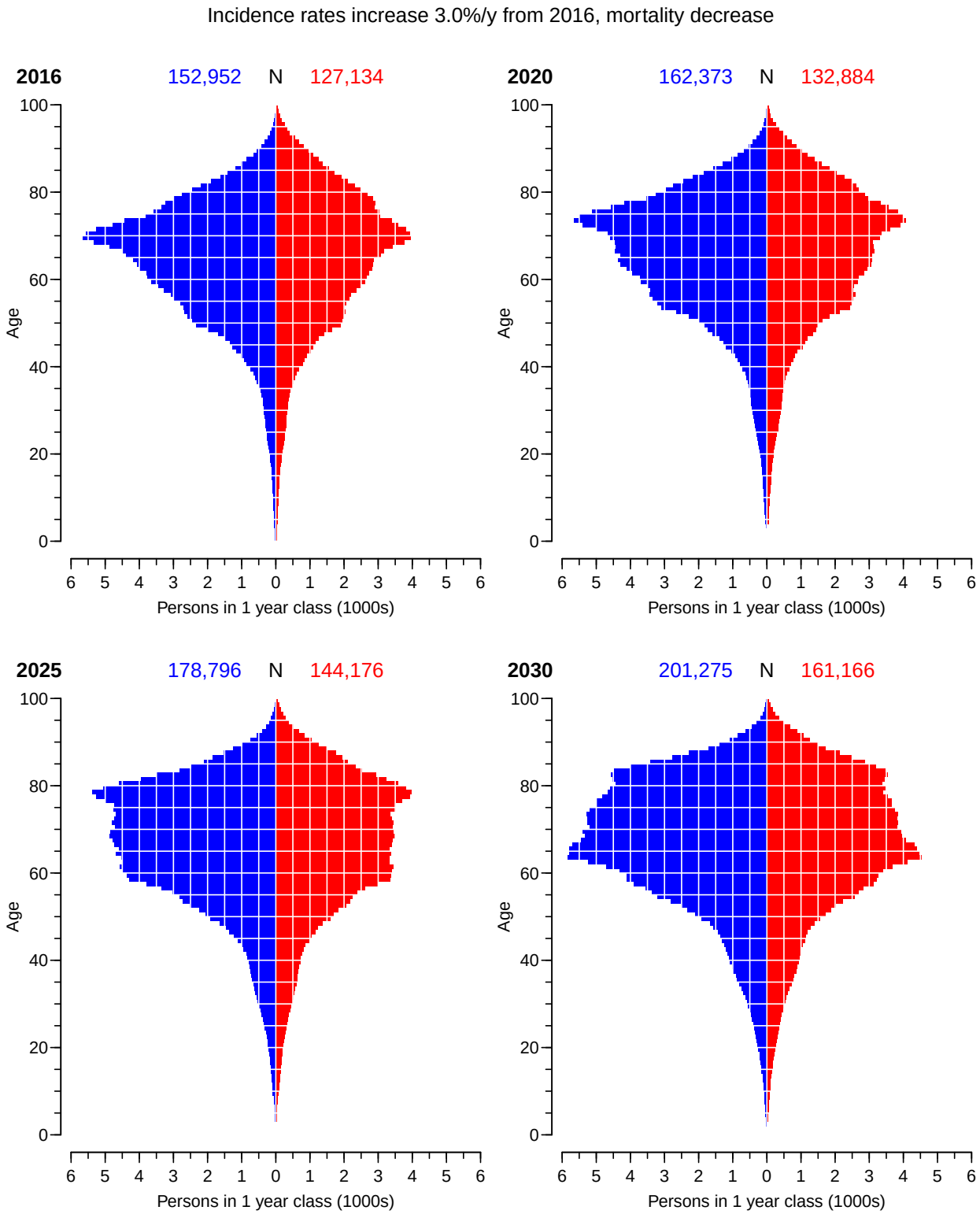

Figure 3.3: *Empirical age-distribution of the diabetes cases in Denmark based on a unchanged mortality and increasing incidence rates (3.0%/y) from 2016.*

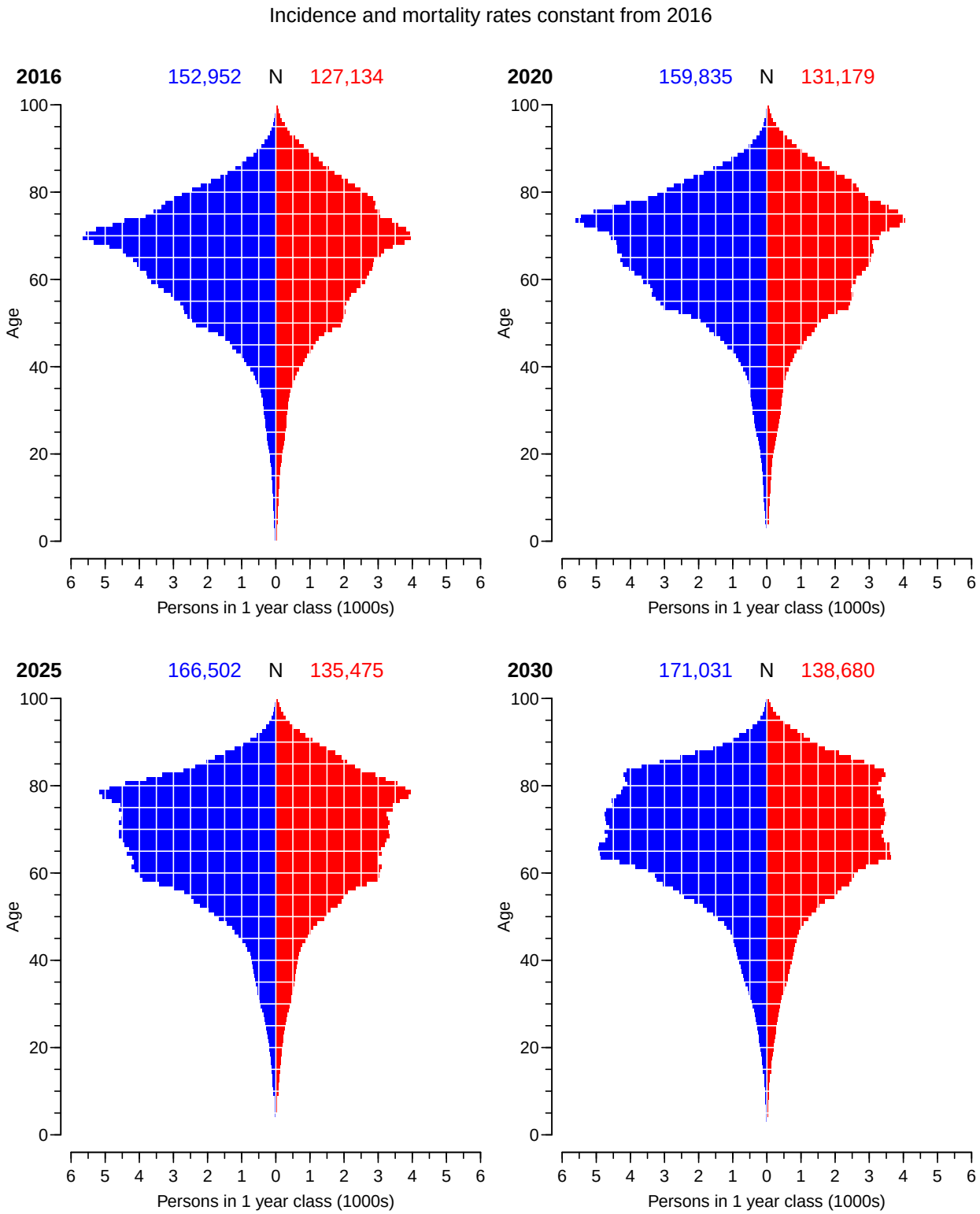

Figure 3.4: *Empirical age-distribution of the diabetes cases in Denmark based on a unchanged mortality and incidence rates from 2016.*

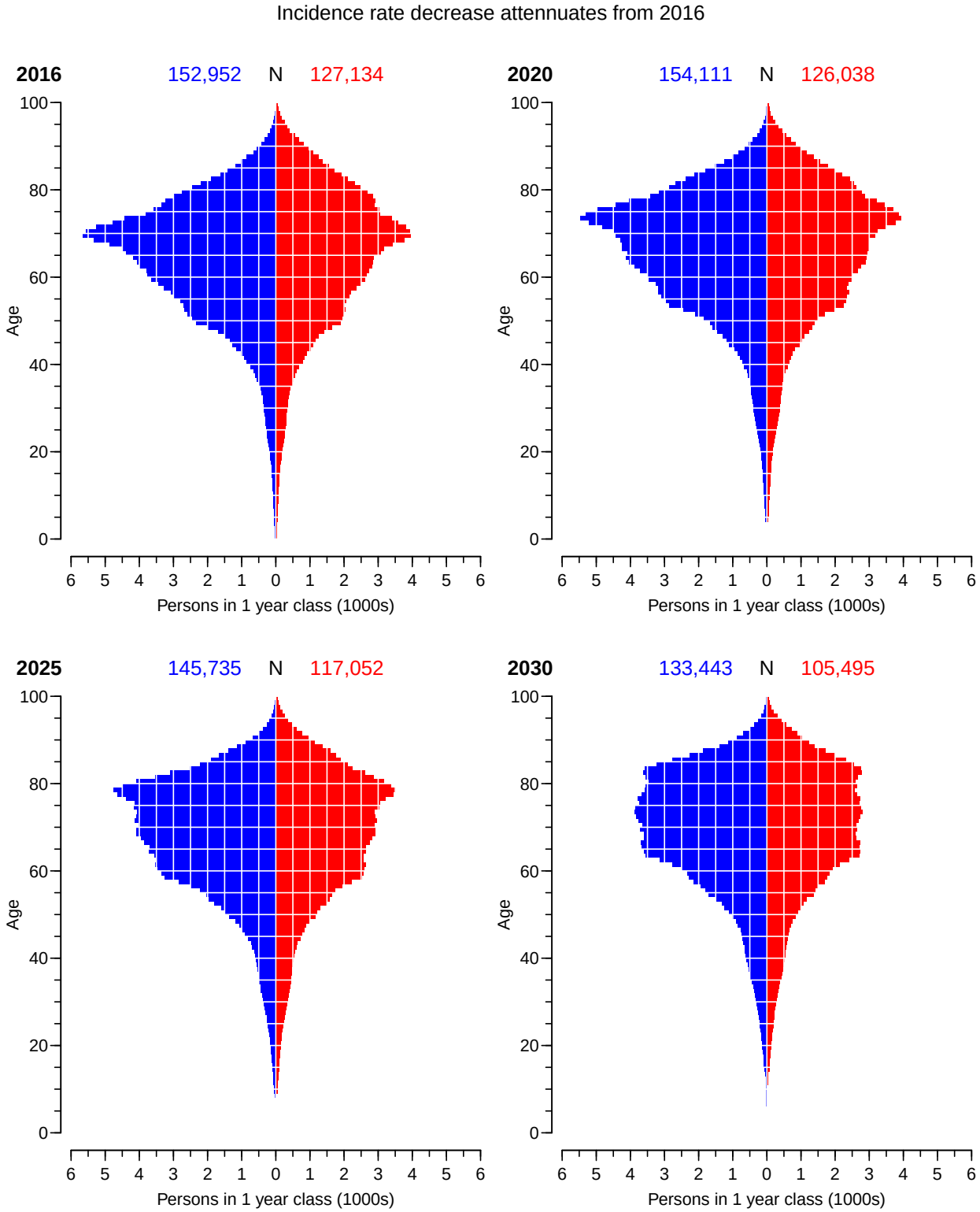

Figure 3.5: *Empirical age-distribution of the diabetes cases in Denmark based on an* attenuated *linear prediction of mortality and incidence rates.*

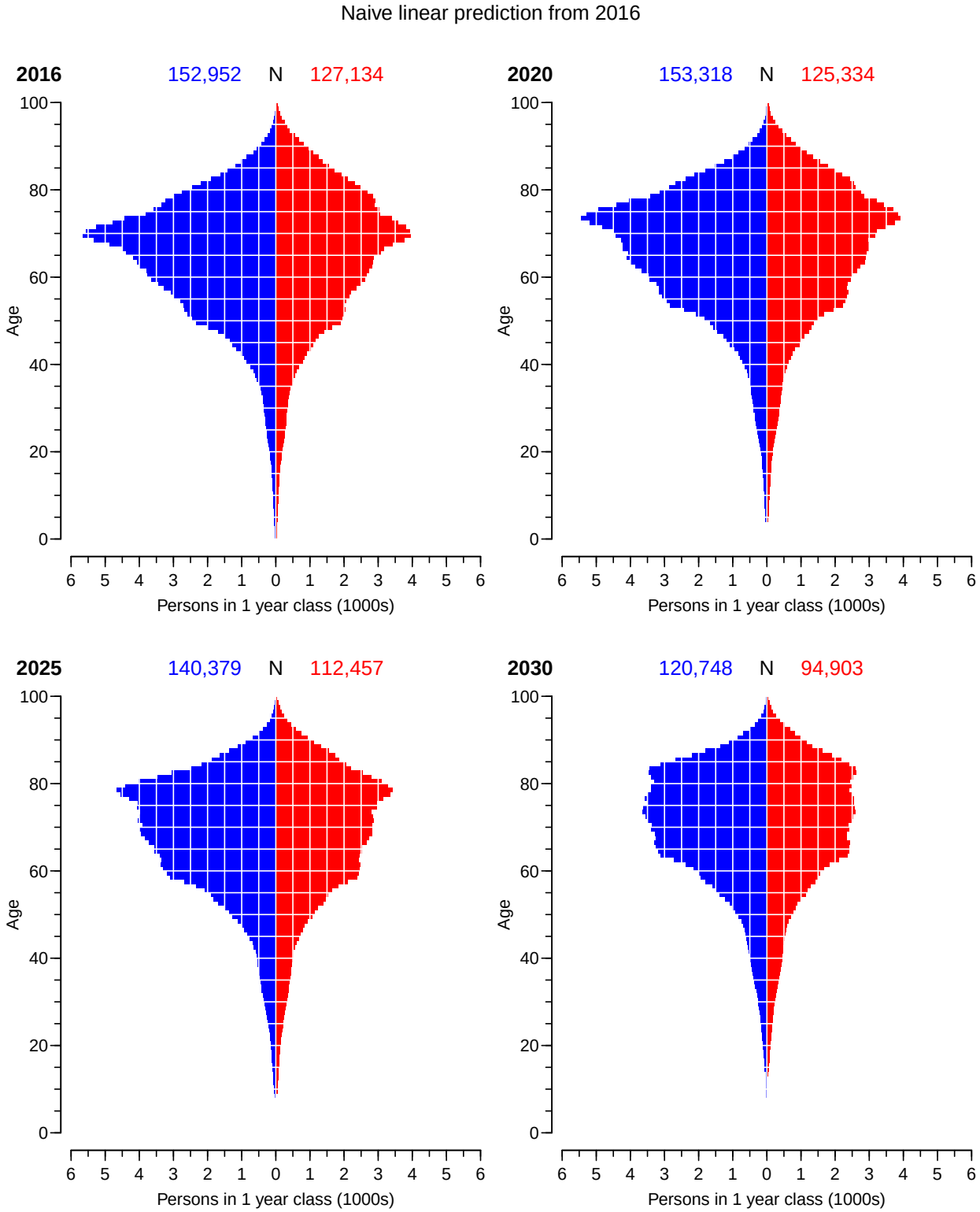

Figure 3.6: *Empirical age-distribution of the diabetes cases in Denmark based on a linear prediction of mortality and incidence rates — mainly decreasing.*

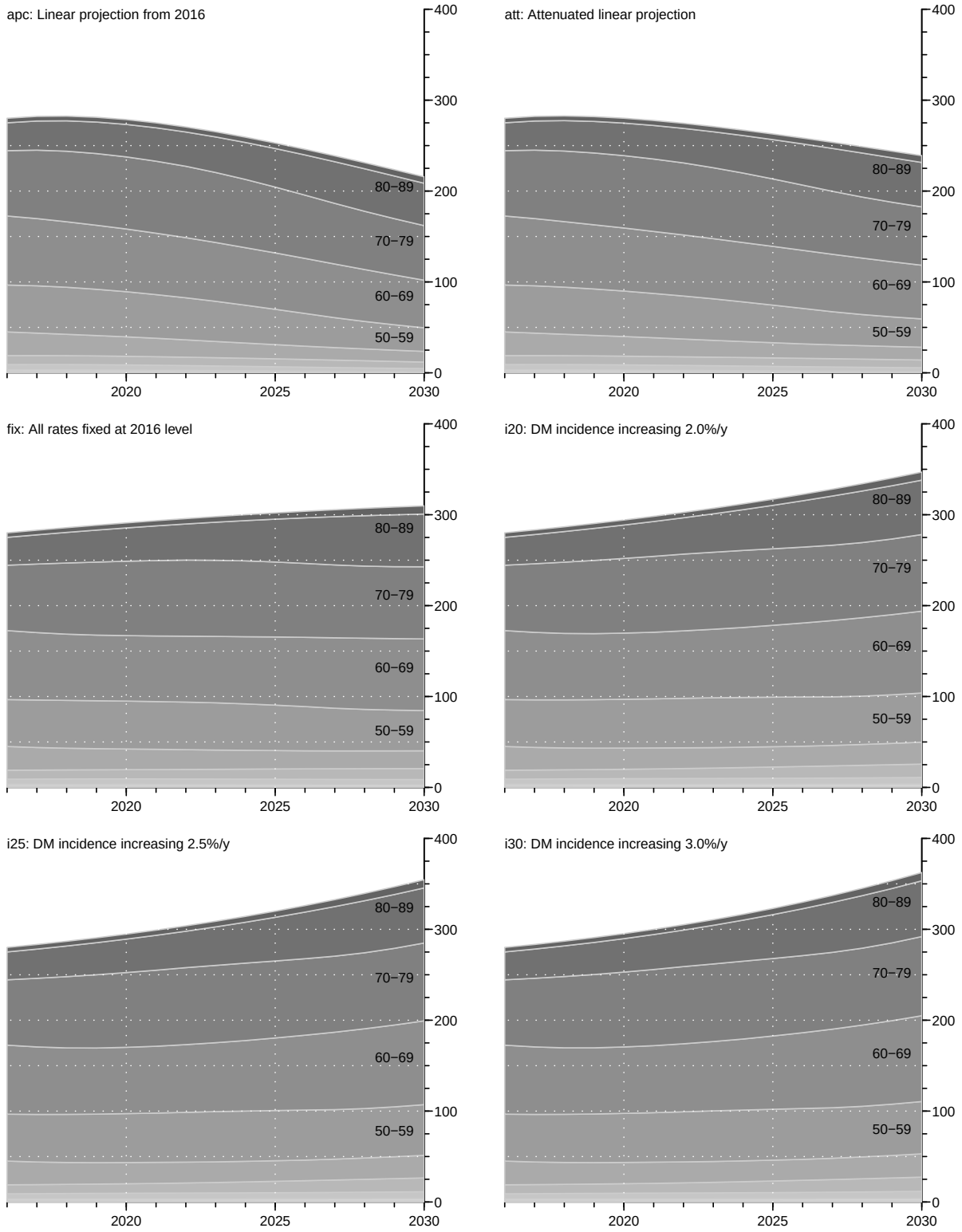

Figure 3.7: *Predicted number of diabetes patients in Denmark under different scenarios.*

References

[1] Renshaw AR and Haberman S. A cohort-based extension to the lee-carter model for mortality reduction factors. *Insurance: Mathematics and Economics*, 2006.

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

[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.