Training day in epidemiology
Stream 1:
Data management for epidemiological analysis in R

SDC / CDC
December 2022
http://BendixCarstensen.com/Epi/Courses/IDEG2022/
Final version

Compiled Tuesday 29th November, 2022, 09:10
from: `c:\Bendix\teach\Epi\IDEG2022\pracs/pracs.tex`

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

Paula Bracco
Assistant Professor
Department of Statistics
Post Graduate Program in Epidemiology
Federal University of Rio Grande do Sul, Brazil
paula.abracco@gmail.com

Points covered in lecture:

- Purpose of the data collection
- Registers:
  - records and variables
  - persons as records
  - events as records
  - time intervals as records
- Merging of data frames:
  - levels of information
  - mis-matches

Chapter 1

Cleaning data

1.1 Reading and cleaning data

The first example is based on a sample that looks like the Danish Diabetes Register.
The backbone is a set of dates, namely date of:

- birth
- diabetes diagnosis
- start of non-insulin anti-diabetic medicine (oad)
- death
- end of follow-up

Further we have a separate data set with dates of start of insulin treatment.
We will need a few functions from the `Epi` package, so attach this and the `tidyverse`:

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

1.1.1 Reading data

We will look at a dataset with follow-up of diabetes patients, it sits at the course website in the data folder: www.bendixcarstensen.com/Epi/IDEG2022/data. You can either download it to your own computer so you can do the exercise off-line, or read it directly from the course website:

```
> folder <- "https://bendixcarstensen.com/Epi/Courses/IDEG2022/data/"
> load(file = url(paste0(folder, "DMreg.Rda")), v = TRUE)
Loading objects:
  DMreg
```

Now take a quick glance at the data:

```
> head(DMreg)
```

```
      id sex      dobth        dodm      dodth      dooad        dox
1  50185   F 1940-04-04 dec/02/1998       <NA>       <NA> 2009/12/31
2 307563   M 1939-03-22 apr/24/2003       <NA> 2007-06-13 2009/12/31
3 294104   F 1918-04-21 jul/21/2004       <NA>       <NA> 2009/12/31
4 336439   F 1965-03-24 apr/06/2009       <NA>       <NA> 2009/12/31
5 245651   M 1932-11-17 aug/27/2008       <NA>       <NA> 2009/12/31
6 216824   F 1927-11-15 nov/21/2007 2009-12-04       <NA> 2009/12/04
```

```
> str(DMreg)
'data.frame':        10000 obs. of  7 variables:
 $ id   : num  50185 307563 294104 336439 245651 ...
 $ sex  : Factor w/ 2 levels "M","F": 2 1 2 2 1 2 1 1 2 1 ...
 $ dobth: chr  "1940-04-04" "1939-03-22" "1918-04-21" "1965-03-24" ...
 $ dodm : chr  "dec/02/1998" "apr/24/2003" "jul/21/2004" "apr/06/2009" ...
 $ dodth: chr  NA NA NA NA ...
 $ dooad: chr  NA "2007-06-13" NA NA ...
 $ dox  : chr  "2009/12/31" "2009/12/31" "2009/12/31" "2009/12/31" ...
```

All the date variables look nice at first glance, but they are character variables, so you must transform them with the relevant format. The default format is `yyyy-mm-dd`, so nothing extra is required for variables in this format; the `dodm` and `dox` are in other formats so we need to specify these—you can find the available format modifiers listed on the help page of `strftime`:

```
> ?strftime
```

There will be some trial and error, so we make a copy of the data frame so that we have a reference, and so that we can start afresh without too much hassle:

```
> org <- DMreg
```

(the first line here is just for starting over again)

```
> DMreg <- org
> DMreg$dobth  <- as.Date(DMreg$dobth)
> DMreg$dodm   <- as.Date(DMreg$dodm, format = "%b/%d/%Y")
> DMreg$dodth  <- as.Date(DMreg$dodth)
> DMreg$dooad  <- as.Date(DMreg$dooad)
> DMreg$dox    <- as.Date(DMreg$dox , format = "%Y/%m/%d")
> head(org)
      id sex      dobth        dodm      dodth      dooad        dox
1  50185   F 1940-04-04 dec/02/1998       <NA>       <NA> 2009/12/31
2 307563   M 1939-03-22 apr/24/2003       <NA> 2007-06-13 2009/12/31
3 294104   F 1918-04-21 jul/21/2004       <NA>       <NA> 2009/12/31
4 336439   F 1965-03-24 apr/06/2009       <NA>       <NA> 2009/12/31
5 245651   M 1932-11-17 aug/27/2008       <NA>       <NA> 2009/12/31
6 216824   F 1927-11-15 nov/21/2007 2009-12-04       <NA> 2009/12/04
```

```
> head(DMreg)
      id sex      dobth       dodm      dodth      dooad        dox
1  50185   F 1940-04-04 1998-12-02       <NA>       <NA> 2009-12-31
2 307563   M 1939-03-22 2003-04-24       <NA> 2007-06-13 2009-12-31
3 294104   F 1918-04-21 2004-07-21       <NA>       <NA> 2009-12-31
4 336439   F 1965-03-24 2009-04-06       <NA>       <NA> 2009-12-31
5 245651   M 1932-11-17 2008-08-27       <NA>       <NA> 2009-12-31
6 216824   F 1927-11-15 2007-11-21 2009-12-04       <NA> 2009-12-04
```

```
> str(DMreg)
```

```
'data.frame':	10000 obs. of  7 variables:
 $ id   : num  50185 307563 294104 336439 245651 ...
 $ sex  : Factor w/ 2 levels "M","F": 2 1 2 2 1 2 1 1 2 1 ...
 $ dobth: Date, format: "1940-04-04" "1939-03-22" ...
 $ dodm : Date, format: "1998-12-02" "2003-04-24" ...
 $ dodth: Date, format: NA NA ...
 $ dooad: Date, format: NA "2007-06-13" ...
 $ dox  : Date, format: "2009-12-31" "2009-12-31" ...
```

So we see that all the variables are now converted to dates.

1.1.2 Miscoded dates

But R does not tell you if some dates have invalid formats, those units will just silently be converted to NAs, so we check if any extra missing values have been introduced:

```
> c(sum(is.na(DMreg$dodm )), sum(is.na(org$dodm)))
[1] 0 0
> c(sum(is.na(DMreg$dobth)), sum(is.na(org$dobth)))
[1] 4 0
> c(sum(is.na(DMreg$dodth)), sum(is.na(org$dodth)))
[1] 7498 7497
> c(sum(is.na(DMreg$dooad)), sum(is.na(org$dooad)))
[1] 4507 4505
> c(sum(is.na(DMreg$dox  )), sum(is.na(org$dox  )))
[1] 0 0
```

We see that there are some variables with extra NAs introduced.

We must find those dates that translated to missing, first for dobth, we derive the lines of the DMreg/org where there is a mismatch of NAs:

```
> (wh <- which(is.na(DMreg$dobth) & !is.na(org$dobth)))
[1]  626 2038 3849 6010
> DMreg[wh,]
         id sex dobth       dodm dodth      dooad        dox
626  235221   M  <NA> 2005-08-24  <NA> 2005-10-12 2009-12-31
2038 406109   M  <NA> 2003-05-13  <NA> 2005-11-08 2009-12-31
3849 435466   M  <NA> 1999-09-29  <NA>       <NA> 2009-12-31
6010 230872   F  <NA> 2007-11-21  <NA>       <NA> 2009-12-31
> org[wh,]
         id sex      dobth        dodm dodth      dooad        dox
626  235221   M 1944-17-02 aug/24/2005  <NA> 2005-10-12 2009/12/31
2038 406109   M 1952-29-03 maj/13/2003  <NA> 2005-11-08 2009/12/31
3849 435466   M 1925/36/14 sep/29/1999  <NA>       <NA> 2009/12/31
6010 230872   F 1956-16-10 nov/21/2007  <NA>       <NA> 2009/12/31
```

Now we see the accidental exchange of day and month so we can change it, either in the original data set or directly in DMreg—but remember that the variables in DMreg are date variables, the default format is the ISO-standard “yyyy-mm-dd”, so if we use that we can omit the format= argument to as.Date.

The second of the deficente dates is intractable, so we have no choice but to enter it as empty, or more brutally, if we trust the calendar year as 2nd July:

```
> DMreg[wh, "dobth"] <- as.Date(c("1944-02-17","1925-7-2","1952-03-29","1956-10-16"))
```

Then for `dodth`, we also find month and day interchanged:

```
> (wh <- which(is.na(DMreg$dodth) & !is.na(org$dodth)))
[1] 5064
> org[wh,]
        id sex      dobth        dodm      dodth dooad        dox
5064 38068   M 1934-06-03 feb/08/1995 2006-18-12  <NA> 2006/12/18
> DMreg[wh,]
        id sex      dobth       dodm dodth dooad        dox
5064 38068   M 1934-06-03 1995-02-08  <NA>  <NA> 2006-12-18
> DMreg[wh, "dodth"] <- as.Date("2006-12-18")
```

And finally for `dooad`:

```
> (wh <- which(is.na(DMreg$dooad) & !is.na(org$dooad)))
[1] 5027 9747
> org[wh,]
         id sex      dobth        dodm dodth      dooad        dox
5027 266467   M 1964-04-19 aug/18/2005  <NA> 2005-25-08 2009/12/31
9747 105517   M 1952-02-08 nov/15/2004  <NA> 2004-15-11 2009/12/31
> DMreg[wh,]
         id sex      dobth       dodm dodth dooad        dox
5027 266467   M 1964-04-19 2005-08-18  <NA>  <NA> 2009-12-31
9747 105517   M 1952-02-08 2004-11-15  <NA>  <NA> 2009-12-31
> DMreg[wh,"dooad"] <- as.Date(c("2005-08-25","2004-11-15"))
```

We now have the `DMreg` cleaned from *some* of the detectable errors; we may encounter others in practice. But also note that there are some typing errors in data that cannot be detected, for example typing `2008-03-09` instead of `2008-09-03`; both are valid dates and there is no way to detect the typo without additional information.

1.1.3 Dates outside range

We must also look out for the time-relation between the data variables; this is most easily done by plotting all pairs of date variables against each other, and adding the identity line to each plot—the latter requires that we use the `panel=` argument to `pairs`. The argument should be a function that adds to a plot, so we define a function that plots points and adds teh identity line $y = x$:

```
> panfun <- function(x, y)
+           {
+           points(x, y, pch = 16, cex=0.7)
+           abline(0, 1, col = "red")
+           }
> (dvar <- fgrep("do", names(DMreg)))
[1] "dobth" "dodm"  "dodth" "dooad" "dox"
> pairs(DMreg[,dvar], panel = panfun)
```

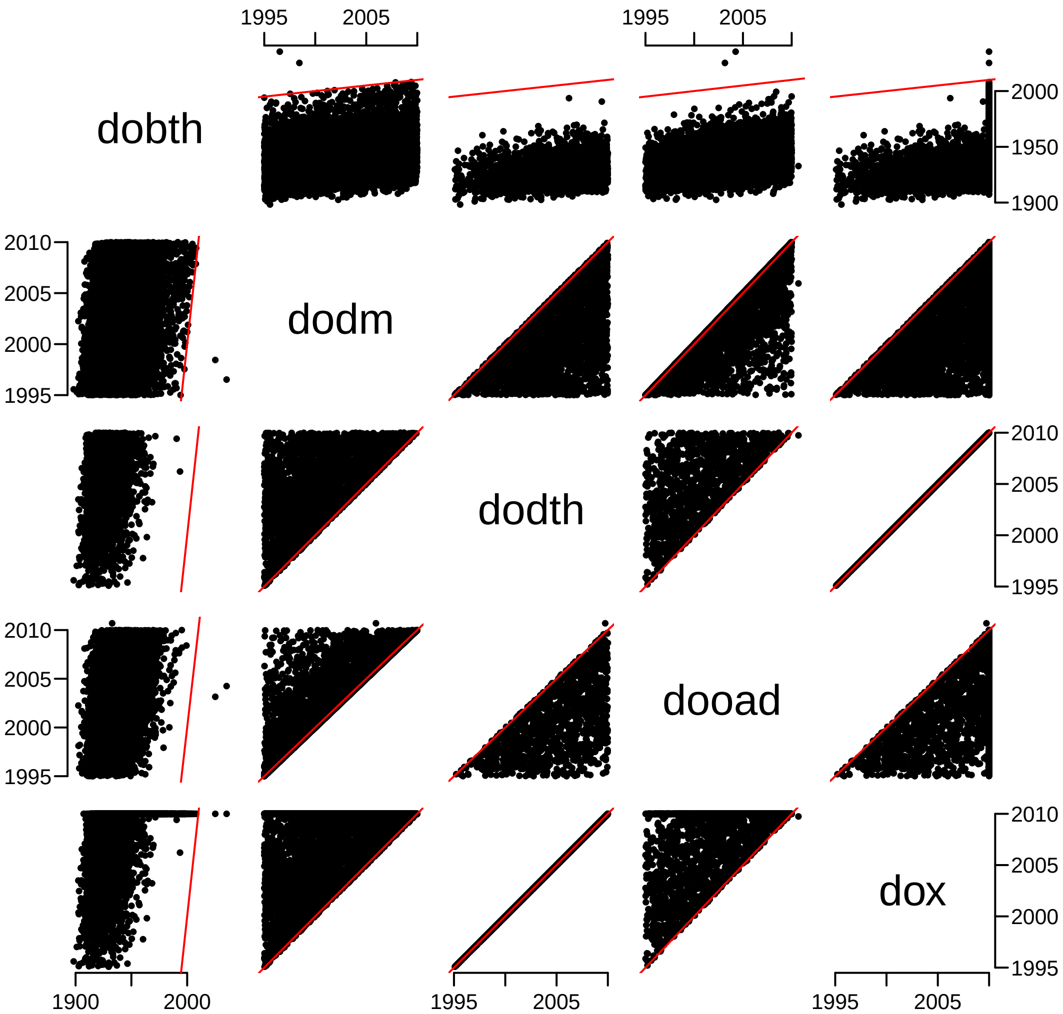

Figure 1.1: *Pairwise scatter plots of the date-variables in* `DMreg`. `../graph/clean-pairs`

Here we see that there are some dates of birth that are miscoded, some births appear long time after any of the other dates:

```
> (wh <- which(DMreg$dobth > as.Date("2010-1-1")))
[1] 6387 7028
> org[wh,]
         id sex      dobth        dodm dodth      dooad        dox
6387 264279   M 2025-03-19 jun/12/1998  <NA> 2003-02-25 2009/12/31
7028  51316   M 2035-05-04 jul/10/1996  <NA> 2004-04-01 2009/12/31
> DMreg[wh,]
         id sex      dobth       dodm dodth      dooad        dox
6387 264279   M 2025-03-19 1998-06-12  <NA> 2003-02-25 2009-12-31
7028  51316   M 2035-05-04 1996-07-10  <NA> 2004-04-01 2009-12-31
```

It appears that the two persons just got the wrong century recorded, so we subtract 100 years from each of the birthdates (`Date` variables are numeric in units of days):

```
> DMreg[wh,"dobth"] <- DMreg[wh,"dobth"] - 36525
> DMreg[wh,]
         id sex      dobth       dodm dodth      dooad        dox
6387 264279   M 1925-03-19 1998-06-12  <NA> 2003-02-25 2009-12-31
7028  51316   M 1935-05-04 1996-07-10  <NA> 2004-04-01 2009-12-31
```

1.1.4 Dates in wrong order

We also see that there are some dates of OAD, `dooad` that are after death:

```
> (wh <- which(DMreg$dooad > DMreg$dodth))
[1] 6370
> org[wh,]
         id sex      dobth        dodm      dodth      dooad        dox
6370 114618   F 1932-10-08 dec/14/2005 2009-09-29 2010-09-11 2009/09/29
> DMreg[wh,]
         id sex      dobth       dodm      dodth      dooad        dox
6370 114618   F 1932-10-08 2005-12-14 2009-09-29 2010-09-11 2009-09-29
```

Unlike other inconsistencies there is no way that we, based on data alone, can find out what is wrong here. One remedy (that will possibly bias the rates of OAD initiation) is to trust the dates of death and just put the `dooad` to `NA` (as if OAD never occurred):

```
> DMreg[wh,"dooad"] <- NA
> DMreg[wh,]
         id sex      dobth       dodm      dodth dooad        dox
6370 114618   F 1932-10-08 2005-12-14 2009-09-29  <NA> 2009-09-29
```

1.1.5 Merging with insulin dates

We also have a separate file with `id` and date of insulin use, located at the same place as the `DMreg` file:

```
> load(file = url(paste0(folder, "DMins.Rda")), v = TRUE)
Loading objects:
  DMins
> str(DMins)
'data.frame':        1814 obs. of  2 variables:
 $ id   : num  38336 132331 161862 109098 258552 ...
 $ doins: chr  "2005-05-10" "2005-12-30" "2009-04-14" "2008-08-22" ...
```

We can merge the insulin dates to the `DMreg` file, but first we would like to see if all `ids` in `DMins` are in `DMreg`; the function `setdiff` is useful for this purpose:

```
> length(setdiff(DMreg$id, DMins$id))
[1] 8209
> length(setdiff(DMins$id, DMreg$id))
[1] 18
> setdiff(DMins$id, DMreg$id)
```

```
 [1] 236386 165310 180036 336806 380315 313226 118609 104926 331627 419936 272411 191857
[13] 200817  74906  83023 199886  76829   8100
```

Of course there are many persons in `DMreg` that do not have a `DMins` record, but also there are some persons in `DMins` that are not in `DMreg`.

We can also check if there are any `id`-duplicates in `DMins`:

```
> table(table(DMreg$id))
    1
10000
> table(table(DMins$id))

   1    2
1804    5
```

Indeed there is, in `DMins`, and we can see who they are:

```
> tt <- table(DMins$id)
> str(tt)
 'table' int [1:1809(1d)] 1 1 1 1 1 1 1 1 1 1 ...
 - attr(*, "dimnames")=List of 1
  ..$ : chr [1:1809] "375" "625" "743" "767" ...
> tt[tt > 1]

 85582 141923 150246 184075 357993
     2      2      2      2      2
> (nn <- names(tt[tt > 1]))
[1] "85582"  "141923" "150246" "184075" "357993"
> dd <- subset(DMins, id %in% nn)
> dd[order(dd$id),]
         id      doins
651   85582 1998-12-09
1794  85582 1999-01-02
749  141923 1996-10-24
1796 141923 1996-10-20
735  150246 2008-02-28
1792 150246 2008-03-23
106  184075 2005-09-05
1795 184075 2005-09-15
119  357993 2000-04-07
1793 357993 2000-04-10
```

We see that these are obviously registrations from slightly different sources, so in this case we can just pick any of the records from each person. In other circumstances the task of choosing one may not be so simple.

We can check if these persons are represented in `DMreg`:

```
> table(dd$id %in% DMreg$id)
TRUE
  10
```

...they all are.

We first try to merge the `DMins` as it is into the `DMreg`:

```
> xx <- left_join(DMreg, DMins)
> table(table(xx$id))
   1    2
9995    5
```

We see that we got duplicate records now—the contents from `DMreg` is also duplicated:

```
> subset(xx, xx$id %in% dd$id)
         id sex      dobth       dodm      dodth      dooad        dox      doins
548  184075   F 1924-08-13 1998-02-04       <NA> 1998-03-20 2009-12-31 2005-09-05
549  184075   F 1924-08-13 1998-02-04       <NA> 1998-03-20 2009-12-31 2005-09-15
639  357993   M 1961-09-25 2000-03-22       <NA> 2000-03-24 2009-12-31 2000-04-07
640  357993   M 1961-09-25 2000-03-22       <NA> 2000-03-24 2009-12-31 2000-04-10
3765  85582   M 1972-09-06 1998-10-28       <NA>       <NA> 2009-12-31 1998-12-09
3766  85582   M 1972-09-06 1998-10-28       <NA>       <NA> 2009-12-31 1999-01-02
4203 150246   M 1949-03-11 2005-07-13 2009-06-03 2007-09-10 2009-06-03 2008-02-28
4204 150246   M 1949-03-11 2005-07-13 2009-06-03 2007-09-10 2009-06-03 2008-03-23
4275 141923   F 1958-08-10 1996-05-08       <NA>       <NA> 2009-12-31 1996-10-24
4276 141923   F 1958-08-10 1996-05-08       <NA>       <NA> 2009-12-31 1996-10-20
```

This is not what we want. So before we do the merge, we must weed out the duplicates from `DMins`; as noted above, in this case it does not really matter which one we take. To this end `duplicated` is used:

```
> DMins <- subset(DMins, !duplicated(id))
> table(table(DMins$id))
   1
1809
```

Then we can make a proper merge (or "join"), where we only keep records present in the left argument:

```
> nrow(DMreg)
[1] 10000
> DMreg <- left_join(DMreg, DMins)
> nrow(DMreg)
[1] 10000
> table(table(DMreg$id))
    1
10000
> str(DMreg)
'data.frame':        10000 obs. of  8 variables:
 $ id   : num  50185 307563 294104 336439 245651 ...
 $ sex  : Factor w/ 2 levels "M","F": 2 1 2 2 1 2 1 1 2 1 ...
 $ dobth: Date, format: "1940-04-04" "1939-03-22" ...
 $ dodm : Date, format: "1998-12-02" "2003-04-24" ...
 $ dodth: Date, format: NA NA ...
 $ dooad: Date, format: NA "2007-06-13" ...
 $ dox  : Date, format: "2009-12-31" "2009-12-31" ...
 $ doins: chr  NA NA NA NA ...
```

We see that we need to convert `doins` to date format (since `doins` is in standard ISO format, no `format=` argument is need for `as.Date`):

```
> DMreg$doins <- as.Date(DMreg$doins)
> str(DMreg)
'data.frame':        10000 obs. of  8 variables:
 $ id   : num  50185 307563 294104 336439 245651 ...
 $ sex  : Factor w/ 2 levels "M","F": 2 1 2 2 1 2 1 1 2 1 ...
 $ dobth: Date, format: "1940-04-04" "1939-03-22" ...
 $ dodm : Date, format: "1998-12-02" "2003-04-24" ...
 $ dodth: Date, format: NA NA ...
 $ dooad: Date, format: NA "2007-06-13" ...
 $ dox  : Date, format: "2009-12-31" "2009-12-31" ...
 $ doins: Date, format: NA NA ...
```

Finally we save a copy for the mortality analysis:

```
> save(DMreg, file = "DMreg.Rda")
```

Final question: What did we miss to check?

1.1.6 Conclusion

We have shown a few possible complications with date variables; some that are fixable, some that cannot be fixed and some that cannot even be detected.

We did a simple merge, showing the need to explore the matching variables and how many record per person there are, before merging datasets.

Chapter 2

Mortality

2.1 Simple analysis of mortality

On the basis of the partial register we of course cannot assess the size of diabetes incidence rates, because 1) we do not have all incident cases of diabetes and 2) we do not have the risk time for the entire (non-diabetic) population.

But on the basis of this sample we *can* estimate the mortality rates, as a function of age, sex (and, time permitting, insulin exposure).

As before, we again need the `Epi` [1] and the `tidyverse` packages:

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

First we load the groomed data from the previous exercise

```
> setwd("C:/Bendix/teach/Epi/IDEG2022/pracs") # a folder on your computer
> load(file = "DMreg.Rda", v = TRUE)
Loading objects:
  DMreg
> str(DMreg)
'data.frame':        10000 obs. of  8 variables:
 $ id   : num  50185 307563 294104 336439 245651 ...
 $ sex  : Factor w/ 2 levels "M","F": 2 1 2 2 1 2 1 1 2 1 ...
 $ dobth: Date, format: "1940-04-04" "1939-03-22" ...
 $ dodm : Date, format: "1998-12-02" "2003-04-24" ...
 $ dodth: Date, format: NA NA ...
 $ dooad: Date, format: NA "2007-06-13" ...
 $ dox  : Date, format: "2009-12-31" "2009-12-31" ...
 $ doins: Date, format: NA NA ...
```

We working with rates it is more convenient to have dates represented in years; so we convert to years, in the form of `cal.yr`:

```
> head(DMreg)
      id sex      dobth       dodm      dodth      dooad        dox doins
1  50185   F 1940-04-04 1998-12-02       <NA>       <NA> 2009-12-31  <NA>
2 307563   M 1939-03-22 2003-04-24       <NA> 2007-06-13 2009-12-31  <NA>
3 294104   F 1918-04-21 2004-07-21       <NA>       <NA> 2009-12-31  <NA>
4 336439   F 1965-03-24 2009-04-06       <NA>       <NA> 2009-12-31  <NA>
5 245651   M 1932-11-17 2008-08-27       <NA>       <NA> 2009-12-31  <NA>
6 216824   F 1927-11-15 2007-11-21 2009-12-04       <NA> 2009-12-04  <NA>
```

```
> DMreg <- cal.yr(DMreg)
> head(DMreg)
      id sex    dobth     dodm    dodth    dooad      dox doins
1  50185   F 1940.256 1998.917       NA       NA 2009.997    NA
2 307563   M 1939.218 2003.309       NA 2007.446 2009.997    NA
3 294104   F 1918.301 2004.552       NA       NA 2009.997    NA
4 336439   F 1965.225 2009.261       NA       NA 2009.997    NA
5 245651   M 1932.877 2008.653       NA       NA 2009.997    NA
6 216824   F 1927.870 2007.886 2009.923       NA 2009.923    NA
> str(DMreg)
'data.frame':        10000 obs. of  8 variables:
 $ id   : num  50185 307563 294104 336439 245651 ...
 $ sex  : Factor w/ 2 levels "M","F": 2 1 2 2 1 2 1 1 2 1 ...
 $ dobth: 'cal.yr' num  1940 1939 1918 1965 1933 ...
 $ dodm : 'cal.yr' num  1999 2003 2005 2009 2009 ...
 $ dodth: 'cal.yr' num  NA NA NA NA NA ...
 $ dooad: 'cal.yr' num  NA 2007 NA NA NA ...
 $ dox  : 'cal.yr' num  2010 2010 2010 2010 2010 ...
 $ doins: 'cal.yr' num  NA NA NA NA NA NA NA NA NA NA ...
```

Now dates are represented as fractional calendar years. This means that 2010.00 is 1 January 2010, 2014.496 is 1 July 2014, etc.

2.2 Mortality by sex

The overall mortality by sex is based on the number of deaths and amount of follow-up time (person-years) for each sex:

```
> ms <- xtabs(cbind(D = !is.na(dodth),
+                   Y = dox - dodm) ~ sex,
+             data = DMreg)
> ms
sex       D        Y
  M  1345.00 27614.21
  F  1158.00 26659.05
> round(cbind(ms, ms[,"D"] / ms[,"Y"] * 100), 2)
     D        Y
M 1345 27614.21 4.87
F 1158 26659.05 4.34
```

thus the overall mortality rate is 4.87/100 PY for men and 4.34 for women, a M/W rate ratio of 1.12:

```
> exp(-diff(log(ms[,"D"] / ms[,"Y"])))
      F
1.12131
```

2.3 Mortality by age

If we want mortality by age, we have the problem that (unlike sex) persons' age varies during the follow-up, and the length of the follow-up is non-negligible:

```
> with(DMreg, summary(dox - dodm))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   2.029   4.794   5.427   8.244  14.995
```

2.3.1 Age at diagnosis

If we just categorize *persons* by age, we will be using age *at diagnosis*:

```
> DMreg <- mutate(DMreg, adiag = dodm - dobth,
+                        adx = cut(adiag, seq(0,110,10), right=FALSE))
> table(DMreg$adx)
   [0,10)   [10,20)   [20,30)   [30,40)   [40,50)   [50,60)   [60,70)   [70,80)   [80,90)
       69       131       215       547      1196      2093      2561      2112       954
 [90,100) [100,110)
      121         1
```

We can make a table of mortality as before (this time it is a 3-dimensional table / array):

```
> ms <- xtabs(cbind(D = !is.na(dodth), Y = dox - dodm) ~ adx + sex, data = DMreg)
> str(ms)
 'xtabs' num [1:11, 1:2, 1:2] 0 1 0 11 57 192 340 494 220 30 ...
 - attr(*, "dimnames")=List of 3
  ..$ adx: chr [1:11] "[0,10)" "[10,20)" "[20,30)" "[30,40)" ...
  ..$ sex: chr [1:2] "M" "F"
  ..$    : chr [1:2] "D" "Y"
 - attr(*, "call")= language xtabs(formula = cbind(D = !is.na(dodth), Y = dox - dodm) ~ adx + s
> rate <- ms[,,"D"] / ms[,,"Y"] * 100
> str(rate)
 'table' num [1:11, 1:2] 0 0.203 0 0.612 1.306 ...
 - attr(*, "dimnames")=List of 2
  ..$ adx: chr [1:11] "[0,10)" "[10,20)" "[20,30)" "[30,40)" ...
  ..$ sex: chr [1:2] "M" "F"
> rate
           sex
adx                  M          F
  [0,10)     0.0000000  0.0000000
  [10,20)    0.2029155  0.2859212
  [20,30)    0.0000000  0.0000000
  [30,40)    0.6119964  0.4859219
  [40,50)    1.3064534  0.7376040
  [50,60)    2.6131730  1.9552697
  [60,70)    4.6436173  3.1859847
  [70,80)   11.2331860  7.8069146
  [80,90)   20.5032750 14.3736217
  [90,100)  47.2713546 41.8692684
  [100,110)             0.0000000
```

We can then show the deaths, person-years, rates and the M/F RR side-by-side:

```
> round(cbind(ms[,,"D"], ms[,,"Y"], rate, RR = rate[,"M"] / rate[,"F"]), 2)
            M   F       M       F     M     F   RR
[0,10)      0   0  215.14  167.69  0.00  0.00  NaN
[10,20)     1   1  492.82  349.75  0.20  0.29 0.71
[20,30)     0   0  542.49 1069.27  0.00  0.00  NaN
[30,40)    11  10 1797.40 2057.94  0.61  0.49 1.26
[40,50)    57  25 4362.96 3389.35  1.31  0.74 1.77
[50,60)   192 101 7347.39 5165.53  2.61  1.96 1.34
[60,70)   340 212 7321.88 6654.14  4.64  3.19 1.46
[70,80)   494 424 4397.68 5431.08 11.23  7.81 1.44
[80,90)   220 318 1073.00 2212.39 20.50 14.37 1.43
[90,100)   30  67   63.46  160.02 47.27 41.87 1.13
[100,110)   0   0    0.00    1.89   NaN  0.00  NaN
```

So we see that men have a higher mortality than women for all ages over 30 at diagnosis. Below age 30 there is no information available — only 2 deaths.

2.3.2 Age at follow-up

If we want the mortality by age at follow-up, we must split the follow-up in age-intervals.

This can be done by defining the follow-up as a `Lexis` object, in this case with age as the only time scale:

```
> Lx <- Lexis(entry = list(age = dodm - dobth),
+              exit = list(age = dox - dobth),
+       exit.status = factor(!is.na(dodth), labels = c("A","D")),
+              data = DMreg)
NOTE: entry.status has been set to "A" for all.
NOTE: Dropping  4  rows with duration of follow up < tol
> summary(Lx)

Transitions:
     To
From    A    D  Records:  Events: Risk time:  Persons:
   A 7497 2499      9996     2499   54273.27      9996
```

With this set up, we can subdivide follow-up in, say, 5-year bins:

```
> sL <- splitLexis(Lx, seq(0,110,5), "age")
> summary(sL)
Transitions:
     To
From     A    D  Records:  Events: Risk time:  Persons:
   A 18327 2499     20826     2499   54273.27      9996
```

We see we now have twice as many records, the follow-up of each person is split over several records, and the variable `age` now refers to the age at the beginning of each of these intervals:

```
> sL <- mutate(sL, afu = cut(age, seq(0,110,10), right=FALSE))
```

The code for calculation of the rates by age at follow-up looks very similar to the previous; but this time we are using age *at follow-up* and not age at diagnosis.

```
> mf <- xtabs(cbind(D = lex.Xst == "D",
+                   Y = lex.dur)
+              ~ afu + sex,
+              data = sL)
> str(mf)
 'xtabs' num [1:11, 1:2, 1:2] 0 1 0 5 32 119 275 486 348 76 ...
 - attr(*, "dimnames")=List of 3
  ..$ afu: chr [1:11] "[0,10)" "[10,20)" "[20,30)" "[30,40)" ...
  ..$ sex: chr [1:2] "M" "F"
  ..$    : chr [1:2] "D" "Y"
 - attr(*, "call")= language xtabs(formula = cbind(D = lex.Xst == "D", Y = lex.dur) ~ afu + sex
> rtfu <- mf[,,"D"] / mf[,,"Y"] * 100
> str(rtfu)
```

```
 'table' num [1:11, 1:2] 0 0.28 0 0.426 1.054 ...
 - attr(*, "dimnames")=List of 2
  ..$ afu: chr [1:11] "[0,10)" "[10,20)" "[20,30)" "[30,40)" ...
  ..$ sex: chr [1:2] "M" "F"
> rtfu
           sex
afu                  M          F
  [0,10)      0.0000000  0.0000000
  [10,20)     0.2801813  0.3893986
  [20,30)     0.0000000  0.0000000
  [30,40)     0.4256046  0.2455456
  [40,50)     1.0537937  0.5470184
  [50,60)     1.8965263  1.3803300
  [60,70)     3.4633811  2.5685907
  [70,80)     8.3234104  5.1855799
  [80,90)    16.0654971 11.1704274
  [90,100)   33.7561106 28.4425236
  [100,110)  35.4956268 69.6156290
```

We can now compare the rates by age at follow-up with those for age at diagnosis:

```
> round(cbind(ms[,,"D"], ms[,,"Y"], rate, RR = rate[,"M"] / rate[,"F"]), 2)
            M   F       M       F     M     F   RR
[0,10)      0   0  215.14  167.69  0.00  0.00  NaN
[10,20)     1   1  492.82  349.75  0.20  0.29 0.71
[20,30)     0   0  542.49 1069.27  0.00  0.00  NaN
[30,40)    11  10 1797.40 2057.94  0.61  0.49 1.26
[40,50)    57  25 4362.96 3389.35  1.31  0.74 1.77
[50,60)   192 101 7347.39 5165.53  2.61  1.96 1.34
[60,70)   340 212 7321.88 6654.14  4.64  3.19 1.46
[70,80)   494 424 4397.68 5431.08 11.23  7.81 1.44
[80,90)   220 318 1073.00 2212.39 20.50 14.37 1.43
[90,100)   30  67   63.46  160.02 47.27 41.87 1.13
[100,110)   0   0    0.00    1.89   NaN  0.00  NaN
> round(cbind(mf[,,"D"], mf[,,"Y"], rtfu, RR = rtfu[,"M"] / rtfu[,"F"]), 2)
            M   F       M       F     M     F   RR
[0,10)      0   0  115.99   80.77  0.00  0.00  NaN
[10,20)     1   1  356.91  256.81  0.28  0.39 0.72
[20,30)     0   0  481.97  609.62  0.00  0.00  NaN
[30,40)     5   4 1174.80 1629.03  0.43  0.25 1.73
[40,50)    32  15 3036.65 2742.14  1.05  0.55 1.93
[50,60)   119  62 6274.63 4491.68  1.90  1.38 1.37
[60,70)   275 157 7940.22 6112.30  3.46  2.57 1.35
[70,80)   486 331 5838.95 6383.09  8.32  5.19 1.61
[80,90)   348 423 2166.13 3786.78 16.07 11.17 1.44
[90,100)   76 160  225.14  562.54 33.76 28.44 1.19
[100,110)   1   3    2.82    4.31 35.50 69.62 0.51
```

We see that the size of the mortality rates are pushed up in age by using the age at follow-up, and also that the M/F RR is larger in all age-classes when using age at follow-up.

2.3.3 Model for smooth age effects

The tabular analysis really belongs in the last century, we would like to see mortality rates as a smooth function of age for men and women.

To this end we split the data in 1-year groups and use the resulting `age` as a *quantitative* variable in modeling of the age effect:

```
> sL <- splitLexis(Lx, 0:100, "age")
> summary(sL)
Transitions:
     To
From     A    D  Records:  Events: Risk time:  Persons:
   A 61621 2499     64120     2499   54273.27      9996
```

We can use a Poisson model to estimate the rates:

```
> mi <- glm.Lexis(sL, ~ Ns(age, knots = 2:9*10) * sex)
stats::glm Poisson analysis of Lexis object sL with log link:
Rates for the transition:
A->D
```

Digression

The function `glm.Lexis` exploits the structure of the Lexis object `sL` to simplify the code; it is really a wrapper for:

```
> glm(cbind(lex.Xst == "D" & lex.Xst != lex.Cst,
+           lex.dur)
+      ~ Ns(age, knots = 2:9*10) * sex,
+      family = poisreg,
+       data = subset(sL, lex.Cst == "A"))
```

which in turn will give the same results as:

```
> glm(lex.Xst == "D" & lex.Xst != lex.Cst
+      ~ Ns(age, knots = 2:9*10) * sex,
+      offset = log(lex.dur),
+      family = poisson,
+        data = subset(sL, lex.Cst == "A"))
```

—note the differences between `poisson` (from the default package `stats`) and `poisreg` (from the `Epi` package).

End of digression

To compute the rates we need a prediction data frame, and we use `matshade` to show the estimated rates for men and women:

```
> prm = data.frame(age = 30:95, sex = "M")
> prf = data.frame(age = 30:95, sex = "F")
> matshade(prm$age, cbind(ci.pred(mi, prm),
+                         ci.pred(mi, prf)) * 100,
+          plot = TRUE,
+          log = "y", col =c("blue","red"),
+          xlab = "Age at FU",
+          ylab = "Mortality per 100 PY")
```

From figure 2.1 we see that mortality among diabetes patients is higher among men than women, almost by a common factor across all ages, converging in ages over 85.

We can estimate the M/F rate ratio by fitting a proportional hazards model, that is one where the age-effect is the same for men and women:

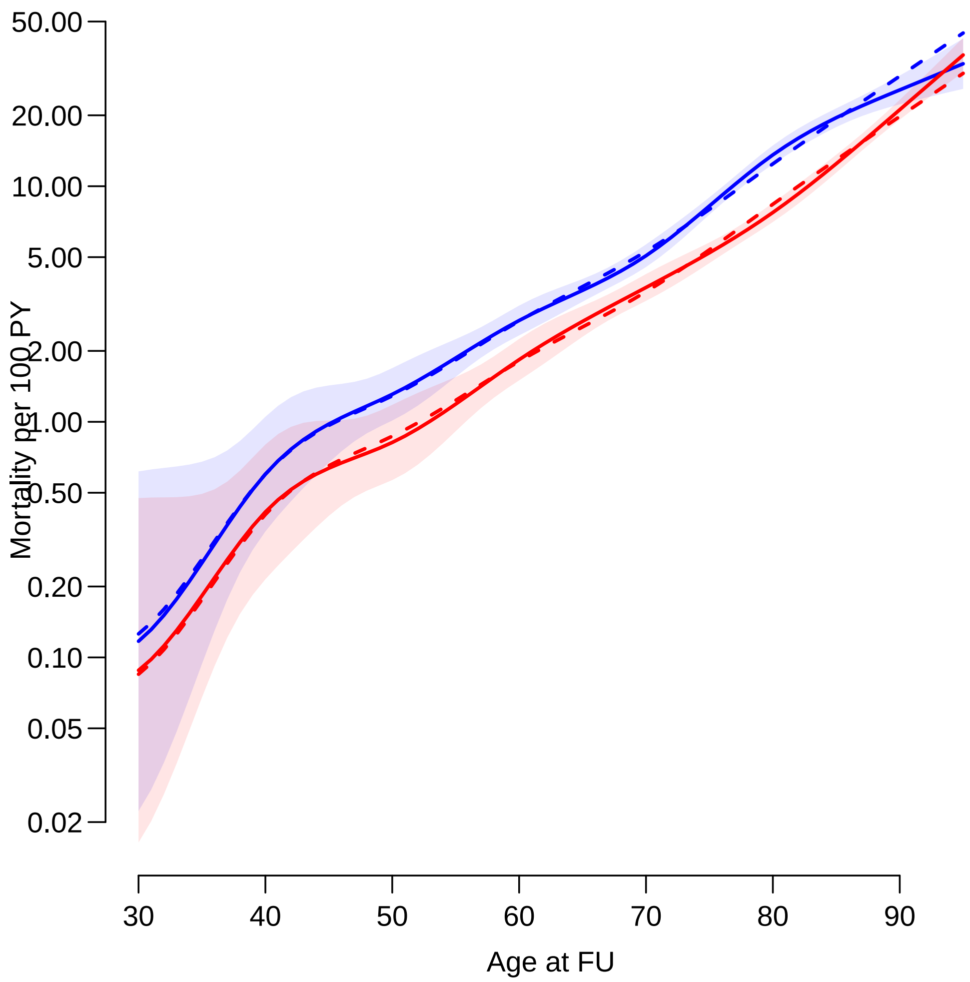

Figure 2.1: *Age-specific mortality rates for Danish diabetes patients 1995–2010. Blue is men, red is women, broken lines are from model with proportional hazards (no interaction between age and sex), full lines from a model with interaction.* `../graph/mort-mf`

```
> ma <- glm.Lexis(sL, ~ Ns(age, knots = 2:9*10) + sex)
stats::glm Poisson analysis of Lexis object sL with log link:
Rates for the transition:
A->D
> round(ci.exp(ma), 3)
                           exp(Est.)  2.5%    97.5%
(Intercept)                    0.001  0.000    0.004
Ns(age, knots = 2:9 * 10)1     6.666  1.378   32.241
Ns(age, knots = 2:9 * 10)2     9.318  3.040   28.559
Ns(age, knots = 2:9 * 10)3    22.266  6.963   71.209
Ns(age, knots = 2:9 * 10)4    39.375 12.919  120.003
Ns(age, knots = 2:9 * 10)5   114.284 35.861  364.212
Ns(age, knots = 2:9 * 10)6   116.542 22.779  596.259
Ns(age, knots = 2:9 * 10)7   304.150 82.707 1118.489
sexF                           0.674  0.622    0.730
> 1 / ci.exp(ma, subset ="sex")
```

```
     exp(Est.)    2.5%    97.5%
sexF  1.483316 1.60716 1.369016
```

so we see that men have 48% higher mortality than women. We can add the rates estimated in the proportional hazards model as dotted lines; we see that the deviation between the two sets of estimated rates is quite small.

A formal likelihood ratio test of the proportional hazards assumption is:

```
> anova(mi, ma, test = "Chisq")
Analysis of Deviance Table

Model 1: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ Ns(age,
    knots = 2:9 * 10) * sex
Model 2: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ Ns(age,
    knots = 2:9 * 10) + sex
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1     64104      18896
2     64111      18909 -7  -13.531  0.06017 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

. . . so formally there is no evidence of interaction ("non proportionality").

We can show the fitted rates from the two models to quantify this visually (figure 2.1):

```
> matshade(prm$age, cbind(ci.pred(mi, prm),
+                         ci.pred(mi, prf),
+                         ci.pred(ma, prm),
+                         ci.pred(ma, prf)) * 100,
+          plot = TRUE,
+          log = "y", lty = c(1,1,2,2), lwd = 2,
+          col =c("blue","red"), alpha = c(1,1,0,0) / 10,
+          xlab = "Age at FU",
+          ylab = "Mortality per 100 PY")
```

Chapter 3

Prevalence

The following is a brief overview of the basic concepts, amended with exercises in derivation of the measures from the National Danish Diabetes Register. The exercises are given first in general terms, and then in more technical terms for those who wish to pursue the calculations in practice.

3.1 Prevalence

Some use the word prevalence for the *number* of affected people, and specifically refer to the prevalence *proportion* when talking about the fraction affected. Here we shall use the term "prevalence" for the fraction affected.

Prevalence always refers to a specified *point* in time:

empirical prevalence of a disease in a population is the fraction of the population that suffers from the disease

theoretical prevalence of a disease in a population is the *probability* that a randomly chosen person from the population suffers from the disease

At first glance these two look pretty much the same, but when we qualify the concepts by, say, age, differences emerge.

The *empirical* prevalence necessarily requires that the population be divided in age-*classes* to enable the calculation of fractions.

The *theoretical* prevalence lends itself to statistical modeling; it is possible to specify mathematically how the probability of being diseased depends on age, so that we have a probability (that is the prevalence) for any age, say 63.7 years.

3.1.1 Practical

The dataset `dr.dta` is a Stata dataset with a modified version of the Danish National Diabetes Register (all dates are randomly moved ± 7 days, so no persons exist in reality). It is also available as an R-dataset, `dr.Rda`. Both are available in the folder http://bendixcarstensen.com/Epi/Courses/IDEG2022/data/.

Dates are coded in years, so that 1 January 2006 is coded as 2006.0, 1 July 2006 is coded 2006.496 and 31 December 2006 as 2006.997.

1. How would you go about estimating the number of prevalent cases in Denmark as of 1 January 2005 if you had access to this dataset?

   You will need all persons that both have a date of diagnosis before 1.1.2005 and who is not dead at that date.

2. We read the dataset with Rusing:

```
> library(Epi)
> library(tidyverse)
> load(url("http://bendixcarstensen.com/Epi/Courses/IDEG2022/data/dr.Rda"), v = T)

Loading objects:
  dr

> # The local version on your computer would be something like:
> # load(file = "../data/dr.Rda")
> str(dr)

'data.frame':        497232 obs. of  5 variables:
 $ sex  : Factor w/ 2 levels "M","F": 2 2 2 2 1 1 1 1 1 2 ...
 $ doBth: 'cal.yr' num  1900 2000 2000 1901 2001 ...
 $ doDM : 'cal.yr' num  1990 2006 2009 1993 2001 ...
 $ doIns: 'cal.yr' num  NA 2006 2009 NA NA ...
 $ doDth: 'cal.yr' num  1991 NA NA 1994 NA ...

> head(dr)

  sex    doBth     doDM    doIns    doDth
1   F 1899.984 1990.052       NA 1991.475
2   F 2000.006 2005.738 2005.773       NA
3   F 2000.002 2008.628 2008.679       NA
4   F 1900.985 1993.489       NA 1994.130
5   M 2001.011 2001.019       NA       NA
6   M 2001.990 2005.763 2005.865       NA

> summary(dr)

 sex            doBth           doDM           doIns              doDth
 M:257840   Min.   :1889   Min.   :1942   Min.   :1994     Min.   :1990
 F:239392   1st Qu.:1927   1st Qu.:1995   1st Qu.:1995     1st Qu.:1998
            Median :1939   Median :2002   Median :2002     Median :2003
            Mean   :1940   Mean   :2001   Mean   :2002     Mean   :2003
            3rd Qu.:1951   3rd Qu.:2008   3rd Qu.:2007     3rd Qu.:2008
            Max.   :2011   Max.   :2012   Max.   :2012     Max.   :2012
                                          NA's   :375954   NA's   :310870
```

3. The prevalent cases at 1 January 2005 are those diagnosed before 2005, and who died later than 2005 (or did not die).

```
> with(dr, table( doDM < 2005 & (doDth > 2005 | is.na(doDth)), exclude=NULL))

 FALSE   TRUE
292757 204475
```

4. How many men and women?

   The further calculations is best made by selecting only those persons that were alive with diabetes at the 1 January 2005, (the data frame `pr2005`):

```
> pr2005 <- subset(dr, doDM < 2005 & (doDth > 2005 | is.na(doDth)))
> (ptt <- with(pr2005, table(sex)))
```

```
sex
     M      F
104171 100304
```

5. How many in each age-class and sex?

   Here we use the function `floor` that throws away decimals — when we divide the age at 2005 (`2005-doBth`) by 5 and remove the decimals and subsequently multiply by 5 we get numbers 0, 5, 10, ... indicating the lower end of each age category—alternatively we can use `cut`:

```
> with(pr2005, table(cut(2005 - doBth,
+                        seq(0,120,5),
+                        right = FALSE),
+                    sex))
```

```
           sex
                M     F
  [0,5)        48    60
  [5,10)      231   232
  [10,15)     503   480
  [15,20)     675   596
  [20,25)     760   817
  [25,30)    1291  1652
  [30,35)    1914  2813
  [35,40)    3055  3954
  [40,45)    4706  4567
  [45,50)    6725  5452
  [50,55)    9263  6807
  [55,60)   14363  9903
  [60,65)   15521 11054
  [65,70)   14007 11274
  [70,75)   11923 11596
  [75,80)    9446 11032
  [80,85)    6155  9697
  [85,90)    2675  5489
  [90,95)     779  2320
  [95,100)    119   477
  [100,105)    12    31
  [105,110)     0     1
  [110,115)     0     0
  [115,120)     0     0
```

6. In the Epi package is the dataset `N.dk` with the size of the Danish population as of 1 January 1971–2013 by sex and 1-year age-classes. The coding of sex is numeric, so we change it to factor as in the register dataset:

```
> data(N.dk)
> head(N.dk)

  sex A    P     N
1   1 0 1971 35839
2   2 0 1971 34108
3   1 1 1971 36302
4   2 1 1971 34153
5   1 2 1971 37855
6   2 2 1971 35609

> str(N.dk)

'data.frame':        8600 obs. of  4 variables:
 $ sex: num  1 2 1 2 1 2 1 2 1 2 ...
 $ A  : num  0 0 1 1 2 2 3 3 4 4 ...
 $ P  : num  1971 1971 1971 1971 1971 ...
 $ N  : num  35839 34108 36302 34153 37855 ...
 - attr(*, "Contents")= chr "Population size as of 1 January in Denmark"

> N.dk <- transform(N.dk,
+                     sex = factor(sex, labels=c("M","F")))
> xtabs(N ~ sex, data=subset(N.dk, P==2005))

sex
      M       F
2677292 2734113
```

so there are 2,677,292 men in Denmark as of 1 January 2005.

The overall prevalence of diabetes among men and women is computed by taking the number of men and women with diabetes and dividing it by the total number of persons in the population.

```
> (pop <- xtabs(N ~ sex, data = subset(N.dk, P == 2005)))

sex
      M       F
2677292 2734113

> ptt

sex
     M      F
104171 100304

> round(ptt / pop * 100, 1)

sex
  M   F
3.9 3.7
```

so the prevalence of diabetes overall was 3.9 and 3.7 percent respectively in men and and women.

7. What are the age-specific prevalences in, say, 10-year classes?

   We make a tabulation of the number of persons by age and sex, and do the same with the number of DM patients from the register, but we only take the first 20 age-classes (0–4,5–9,. . . ,95–99) as these are the ones that are represented in the population figures.

   Note that we compute the persons' ages at the 1 January 2005 (which is coded as 2005.0).

```
> pop <- xtabs(N ~ cut(A, seq(0, 100, 5), right = FALSE) +
+                  sex,
+                  data = subset(N.dk, near(P, 2005) & A < 100))
> ptt <- with(pr2005, table(cut(2005 - doBth,
+                               seq(0, 100, 5),
+                               right = FALSE),
+                           sex))
> cbind(ptt, pop)
```

```
              M     F      M      F
[0,5)        48    60 167882 160174
[5,10)      231   232 176410 167652
[10,15)     503   480 177531 168497
[15,20)     675   596 156371 148211
[20,25)     760   817 147943 144598
[25,30)    1291  1652 173681 172033
[30,35)    1914  2813 193537 190643
[35,40)    3055  3954 210636 203290
[40,45)    4706  4567 204212 197524
[45,50)    6725  5452 187173 182720
[50,55)    9263  6807 180774 179027
[55,60)   14363  9903 195417 193559
[60,65)   15521 11054 158478 160929
[65,70)   14007 11274 116440 124845
[70,75)   11923 11596  88207 103568
[75,80)    9446 11032  68065  90507
[80,85)    6155  9697  45263  75487
[85,90)    2675  5489  20839  44530
[90,95)     779  2320   7147  20756
[95,100)    119   477   1286   5563
```

```
> round((ptt / pop) * 100, 1)
```

```
          sex
              M    F
  [0,5)     0.0  0.0
  [5,10)    0.1  0.1
  [10,15)   0.3  0.3
  [15,20)   0.4  0.4
  [20,25)   0.5  0.6
  [25,30)   0.7  1.0
  [30,35)   1.0  1.5
  [35,40)   1.5  1.9
  [40,45)   2.3  2.3
  [45,50)   3.6  3.0
  [50,55)   5.1  3.8
  [55,60)   7.3  5.1
  [60,65)   9.8  6.9
  [65,70)  12.0  9.0
```

```
[70,75)  13.5 11.2
[75,80)  13.9 12.2
[80,85)  13.6 12.8
[85,90)  12.8 12.3
[90,95)  10.9 11.2
[95,100)  9.3  8.6
```

8. How does the prevalence look as a function of age?

   We have the two column matrices `ptt` and `pop` with diabetes cases and population size as of 1 January 2005, so we can plot the ratio of these against the mid-point of the age-intervals. But formally what is assumed is that age-specific prevalences are constant in 5-year age-classes:

```
> par(mfrow=c(1,2), bty="n", las=1)
> matplot(seq(2.5,97.5,5),  (ptt/pop)*100,
+          type="l", lty=1, lwd=3, col=c("blue","red"),
+          xlab="Age (years)", ylab="Prevalence (%)", las=1, yaxs="i", ylim=c(0,15))
> matplot(seq(0,100,5),  ((ptt/pop)*100)[c(1:20,20),],
+          type="s", lty=1, lwd=3, col=c("blue","red"),
+          xlab="Age (years)", ylab="Prevalence (%)", las=1, yaxs="i", ylim=c(0,15))
```

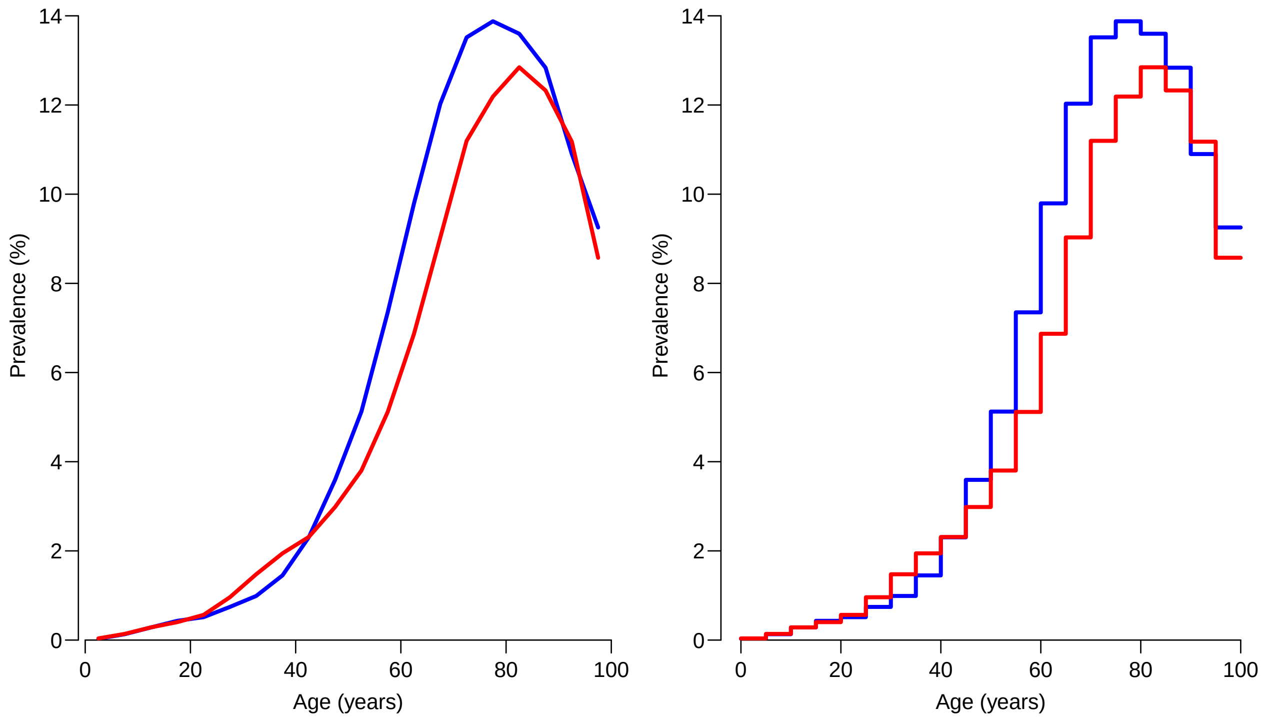

Figure 3.1: *Age-specific prevalence of diabetes at 1 January 2005 in 5-year age-classes in Denmark. The left plot is just connecting the midpoints of the age-classes; the right hand plot shows the formally assumed model with constant prevalence in each 5-year class.* `../graph/prev-prv-5`

9. How does the prevalences look if we use 1-year age-classes?

   This is just the same calculations, replacing 5 by 1 (leaving it a bit superfluous, though) and almost the same code for the plot:

```
> pop <- xtabs(N ~ cut(A, seq(0, 100, 1), right = FALSE) +
+                  sex,
+                  data = subset(N.dk, near(P, 2005) & A < 100))
> ptt <- with(pr2005, table(cut(2005 - doBth,
+                               seq(0, 100, 1),
+                               right = FALSE),
+                           sex))
> par(mfrow=c(1,2), bty="n", las=1)
> matplot(seq(0.5,99.5,1),  (ptt/pop)*100,
+          type="l", lty=1, lwd=3, col=c("red","blue"),
+          xlab="Age (years)", ylab="Prevalence (%)", las=1, yaxs="i", ylim=c(0,15))
> matplot(seq(0,100,1), ((ptt/pop)*100)[c(1:100,100),],
+          type="s", lty=1, lwd=3, col=c("red","blue"),
+          xlab="Age (years)", ylab="Prevalence (%)", las=1, yaxs="i", ylim=c(0,15))
```

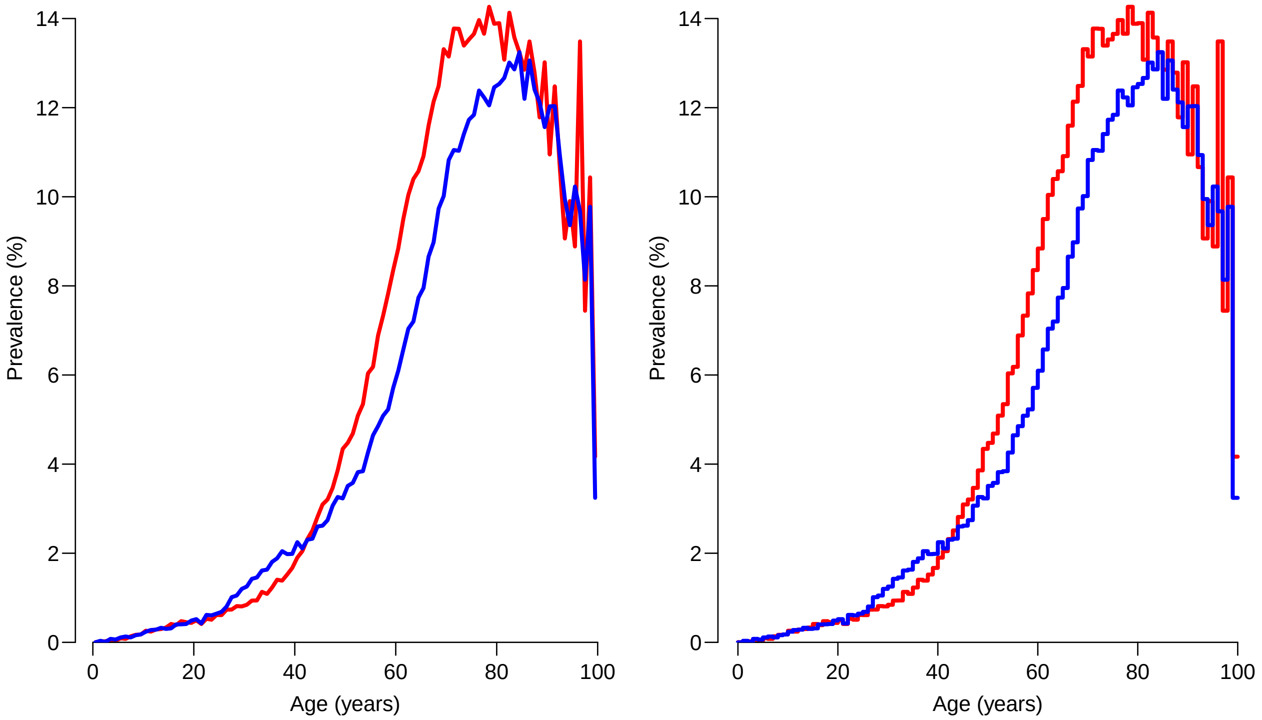

Figure 3.2: *Age-specific prevalence of diabetes at 1 January 2005 in 1-year age-classes in Denmark.* `../graph/prev-prv-1`

From figure 3.2 we get broadly the same picture as from 3.1, but the curves are not "credible".

This is illustrates the differences between the *empirical* prevalences and the `theoretical` prevalences. From a biological/clinical point of view we would of course expect that the prevalence were a smooth function of time, pretty much as approximated by the left hand curve in figure 3.1.

10. How would you go about showing prevalence as a smooth function of age?

    It would be more logical to describe the original data by a smooth curve. Formally, this would require that we knew the exact ages for every person in the Danish population as of 1 January 2005 as well as the diabetes status; we could then model the 2.5 mill. 0/1 variables for men by a binomial model with some smooth age-effect. But we do not have access to these data, so we use the 1-year age classified data for the register and the population. We are then formally making an assumption that prevalences are constant in 1-year age-classes, but we impose restrictions on relationship between the prevalences in the age-classes.

The advantage of this is that we get a more credible relationship between (estimated theoretical) prevalence and age, and in particular one that we can reasonably use for *any* age, not only the midpoints of the intervals.

In practice this is done by fitting a binomial model with a smooth effect of age to the table of prevalent cases and total population using the age-midpoints. In R we need two-column matrix of affected and unaffected as response variable, so the second column must be computed as the population size *minus* the number of patients:

```
> A <- 0:99+0.5
> prM <- cbind(ptt[,"M"], pop[,"M"] - ptt[,"M"])
> prF <- cbind(ptt[,"F"], pop[,"F"] - ptt[,"F"])
> m.pr <- glm(prM ~ Ns(A, knots = seq(10, 95,, 9)), family = binomial)
> f.pr <- glm(prF ~ Ns(A, knots = seq(10, 95,, 9)), family = binomial)
```

`Ns` is a so called natural spline (restricted cubic spline) that specifies a smooth function of `A`.

From this model we can make predictions; in principle for *any* point on the age-scale, but in this case it suffices to do it at the midpoint of the age-categories in order to get a smoothly looking curve.

```
> nd <- data.frame(A=0:99+0.5)
> par(mfrow=c(1,2), bty="n", las=1)
> matplot(nd$A, cbind(ci.pred(m.pr,nd)[,1],
+                      ci.pred(f.pr,nd)[,1])*100,
+          type="l", lty=1, lwd=3, col=c("blue","red"),
+          xlab="Age (years)", ylab="Prevalence (%)", las=1, yaxs="i", ylim=c(0,15))
> matplot(nd$A, cbind(ci.pred(m.pr,nd)[,1],
+                      ci.pred(f.pr,nd)[,1])*100,
+          type="s", lty=1, lwd=3, col=c("blue","red"),
+          xlab="Age (years)", ylab="Prevalence (%)", las=1, yaxs="i", ylim=c(0,15))
```

The *modeling* of prevalences also illustrates the contrast between the *empirical* and *theoretical* prevalences; the former are necessarily tied to a particular grouping of the population; for example by sex and/or age, whereas the latter refer to *any* combination of sex and age; we can in principle refer to the prevalence of DM in women aged 68.3 years:

```
> ci.pred(f.pr, data.frame(A=68.3))
    Estimate       2.5%      97.5%
1 0.09386903 0.09283319 0.09491521
```

This number cannot be derived as an empirical fraction from data; it is a *prediction* from a statistical model. It is our best guess at the probability that a woman aged 68.3 evaluated on 1 January 2005 has diabetes. The model is biologically plausible because the prediction for ages 68.2 and 68.4 are quite similar:

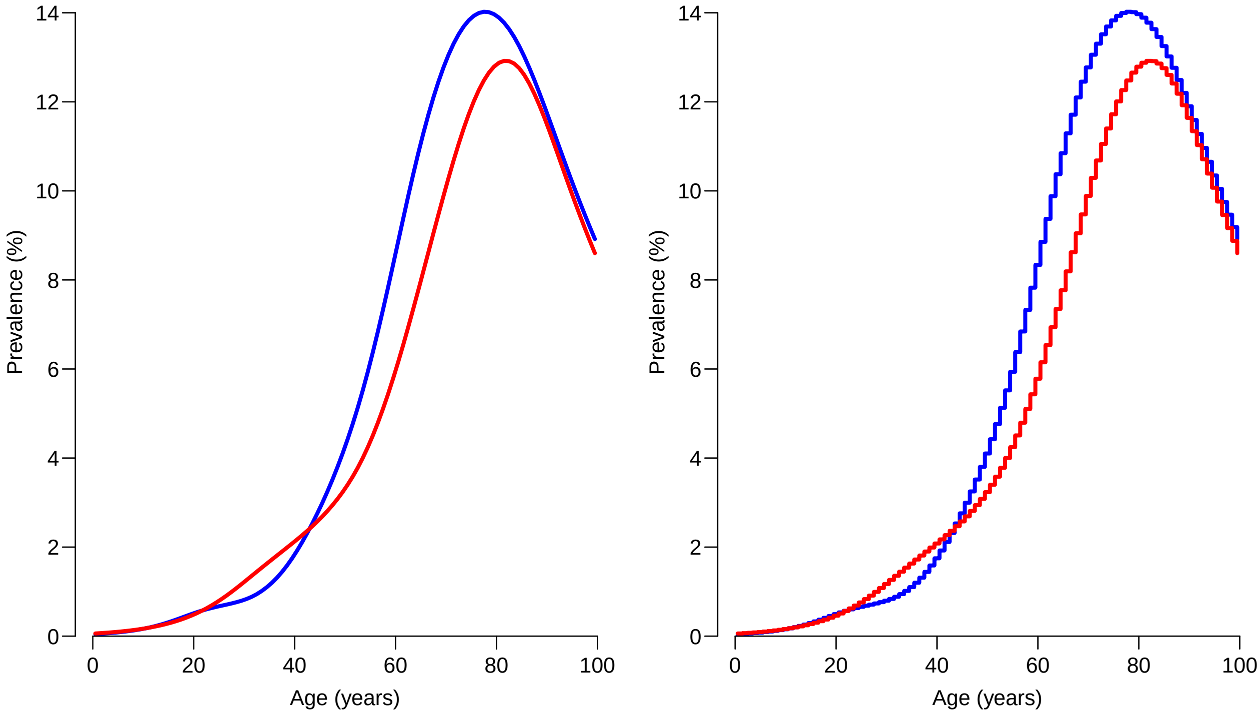

Figure 3.3: *Fitted age-specific prevalences from a binomial model with restricted cubic splines. The left panel is the predicted theoretical prevalence, the right hand plot is the formally fitted model with constant prevalence in each 1-year category and restrictions on the relationship between these.* `../graph/prev-prv-fit`

```
> ci.pred(f.pr, data.frame(A=c(68.2,68.3,68.4)))
    Estimate       2.5%      97.5%
1 0.09344671 0.09241412 0.09448963
2 0.09386903 0.09283319 0.09491521
3 0.09429069 0.09325122 0.09534053
```

We see that we expect that women slightly older has a prevalence (*i.e.* probability of being affected) that is slightly higher too.

References

[1] Bendix Carstensen, Martyn Plummer, Esa Laara, and Michael Hills. *Epi: A Package for Statistical Analysis in Epidemiology*, 2022. R package version 2.47.

[2] Bendix Carstensen. *Epidemiology with R.* Number ISBN: 978-0-19-884133-3. Oxford University Press, 2020.