

Data for the Diabetes Atlas

SDCC

June 2022

www.biostat.ku.dk/~bxc

Version 2

Compiled Tuesday 28th June, 2022, 10:40
from: C:\Bendix\sdc\DMreg\Atlas\atlas.tex

Bendix Carstensen Steno Diabetes Center Copenhagen, Herlev, Denmark
& Department of Biostatistics, University of Copenhagen
bcar0029@regionh.dk
<http://bendixcarstensen.com/>

1	A diabetes map of Denmark	1
1.1	Paraphernalia	1
1.2	Data for the maps 2020	1
1.3	Convenient with aggergation?	5
2	Simple model for the aggregate data set	6
2.1	Prevalences relative to the population mean	11
2.2	Controlling for age	11
3	General function to estimate relative prevalences	13
3.1	Data preparation in general	14
3.2	Collecting results	15
4	Data for diabetes map of Denmark	18
4.1	Paraphernalia	18
4.2	The estimates	18
4.3	The counts	18
4.4	Data dump	19
5	Data files for the atlas	19

1 A diabetes map of Denmark

This program does not generate the maps that will be on the webpage of the SDC Diabetes Ressource Centre. I generates data for prevalence maps “controlled” for age, age plus education and age plus education plus income, for the years 2010, 2012, . . . , 2020.

1.1 Paraphernalia

```
> options(width = 90)
> library(tidyverse)
> library(Epi)
> # library(mapDK) # findes kun under 4.1.0, ikke 4.1.3
> clear()
> source("E:/workdata/707655/util/elapsed.R")
> start()
```

```
-----
Code: E:/workdata/707655/users/BxC/atlas/r/mkAtlas.rnw
Time: 2022-05-16 at 05:46:56
-----
```

First a small convenience function is defined to remove Lexis attributes from a data frame:

```
> unLexis <-
+ function(Lx)
+ {
+ class(Lx) <- setdiff(class(Lx), "Lexis")
+ attr(Lx, "time.scales") <- NULL
+ attr(Lx, "time.since") <- NULL
+ attr(Lx, "breaks") <- NULL
+ Lx
+ }
```

1.2 Data for the maps 2020

We will make a map of the overall prevalence of diabetes in the municipalities of Denmark as of 2020-1-1, say:

```
> setwd("e:/workdata/707655/DMreg2020")
> system.time(load(file = "data/DMfup.Rda" , v = TRUE))
```

Loading objects:

```
DMfup
  bruger    system forløbet
  20.18     0.49     20.75
```

```
> summary(DMfup)
```

Transitions:

From	To				Records:	Events:	Risk time:	Persons:
	noDM	T2	T1	Dead				
noDM	6893318	431635	28752	1172893	8526598	1633280	141768456.6	8041897
T2	0	298631	0	194962	493593	194962	4183399.1	490953
T1	0	0	33683	24572	58255	24572	737988.8	56969
Sum	6893318	730266	62435	1392427	9078446	1852814	146689844.5	8129005

```
> system.time(load(file = "data/popstat.Rda", v = TRUE))
```

Loading objects:

```
popstat
bruger    system forløbet
264.10    3.08    301.51
```

```
> setwd("E:/workdata/707655/users/BxC/atlas/r")
```

We now have the two databases, the population follow-up of all persons resident in Denmark classified by diabetes status, `DMfup`, and the geographical and social variables for the persons, `popstat`, at each 1 Jan 1996–2020.

We then select from the follow-up data those that are alive at a given date, and then records from the status file corresponding to this date are then joined. So for the date `yy` it will be:

```
> yy <- 2020
> prvyr <-
+ function(yy)
+   {
+     # selects records that contain the date yy:
+     prv <- DMfup[DMfup$per <= yy &
+               yy < (DMfup$per + DMfup$lex.dur),
+               c("lex.id", "sex", "per", "doBth", "lex.Cst")]
+     # population information for these persons
+     var <- filter(popstat, pnr %in% prv$lex.id & yr == yy)
+     # merge it onto the prevalence data
+     unLexis(left_join(prv, var, by = c("lex.id" = "pnr")))
+   }
> system.time(prv20 <- prvyr(2020))
bruger    system forløbet
26.99     1.24    28.21

> str(prv20, v = 0)
'data.frame':      6145021 obs. of  15 variables:
 $ lex.id      : chr    ...
 $ sex        : Factor w/ 2 levels "M","W": NULL ...
 $ per        : num  NULL ...
 $ doBth      : num  NULL ...
 $ lex.Cst    : Factor w/ 4 levels "noDM","T2","T1",...: NULL ...
 $ yr        : num  NULL ...
 $ reg        : Factor w/ 5 levels "Nordjylland",...: NULL ...
 $ kom        : Factor w/ 99 levels "København","Frederiksberg",...: NULL ...
 $ udd        : num  NULL ...
 $ udddk      : Factor w/ 15 levels "Førskoleuddannelser",...: NULL ...
 $ eduen      : Factor w/ 9 levels "Early childhood education",...: NULL ...
 $ find       : num  NULL ...
 $ findec     : Factor w/ 10 levels "1","2","3","4",...: NULL ...
 ..- attr(*, "label")= chr    ...
 $ familie_id : chr    ...
 $ antboernh  : num  NULL ...
```

We now have dataframes with one record per person present in the Danish population (according to `DMfup`) at the dates of interest.

But we still have too many persons in the dataset:

```
> with(prv20, rCp(table(has.kom = !is.na(kom),
+                       has.fin = !is.na(findec),
+                       exclude = NULL)))
```

```

      has.fin
has.kom FALSE      TRUE
  FALSE  452,083    75,353
  TRUE      * 5,617,583

```

We will only use those with a valid geographical code and family income, and we want the NAs in `udddk` as a coded category:

```

> prv20 <- ( prv20
+           %>% filter(!is.na(kom) & !is.na(findec))
+           %>% mutate(udddk = factor(ifelse(is.na(udddk),
+                                           "Uoplyst mv.",
+                                           as.character(udddk))),
+           eduen = factor(ifelse(is.na(eduen),
+                                   "Not elsewhere classified",
+                                   as.character(eduen))),
+           DMst = factor(lex.Cst))
+         )
> str(prv20, v = 0)
'data.frame':      5617583 obs. of  16 variables:
 $ lex.id      : chr   ...
 $ sex        : Factor w/ 2 levels "M","W": NULL ...
 $ per        : num  NULL ...
 $ doBth      : num  NULL ...
 $ lex.Cst    : Factor w/ 4 levels "noDM","T2","T1",...: NULL ...
 $ yr         : num  NULL ...
 $ reg        : Factor w/ 5 levels "Nordjylland",...: NULL ...
 $ kom        : Factor w/ 99 levels "København","Frederiksberg",...: NULL ...
 $ udd        : num  NULL ...
 $ udddk      : Factor w/ 10 levels "Adgangsgivende uddannelsesforløb",...: NULL ...
 $ eduen      : Factor w/ 8 levels "Bachelor or equivalent",...: NULL ...
 $ find       : num  NULL ...
 $ findec     : Factor w/ 10 levels "1","2","3","4",...: NULL ...
 ..- attr(*, "label")= chr   ...
 $ familie_id: chr   ...
 $ antboernh  : num  NULL ...
 $ DMst       : Factor w/ 3 levels "noDM","T2","T1": NULL ...
> save(prv20, file = "prv20.Rda")

```

We can get an overview of the marginal distribution of the variables:

```

> with(prv20, fCp(addmargins(
+               table(DMst , exclude = NULL))))
DMst
  noDM      T2      T1      Sum
5,305,004 281,797  30,782 5,617,583
> with(prv20, fCp(table(sex , exclude = NULL)))
sex
  M      W
2,791,180 2,826,403
> with(prv20, fCp(table(!is.na(doBth), exclude = NULL)))
  TRUE
5,617,583
> with(prv20, fCp(table(reg , exclude = NULL)))

```

```

reg
Nordjylland Midtjylland Syddanmark Hovedstaden Sjælland
    569,784    1,277,687    1,184,941    1,769,862    815,309
> with(prv20, fCp(sort(table(kom , exclude = NULL))))
kom
    Christiansø      Læsø      Fanø      Samsø      Ærø
        82      1,742      3,314      3,549      5,804
    Langeland      Dragør      Vallensbæk      Lemvig      Morsø
    12,103      14,095      16,105      19,318      19,793
    Struer      Odder      Glostrup      Stevns      Ishøj
    20,522      21,997      22,080      22,157      22,328
    Solrød      Kerteminde      Hørsholm      Allerød      Billund
    22,515      23,118      24,414      25,059      25,745
    Albertslund      Lejre      Herlev      Nordfyns      Rebild
    25,995      27,181      28,108      28,835      29,033
    Sorø      Halsnæs      Nyborg      Odsherred      Ringsted
    29,121      30,549      31,245      32,084      33,958
    Brøndby      Brønderslev      Faxe      Vesthimmerlands      Norddjurs
    34,322      35,438      35,656      35,713      36,233
    Tønder      Jammerbugt      Middelfart      Bornholm      Rødovre
    36,432      37,186      37,475      38,712      39,010
    Fredensborg      Furesø      Gribskov      Ikast-Brande      Assens
    39,845      39,889      39,984      40,006      40,189
    Lolland      Mariagerfjord      Syddjurs      Vejen      Tårnby
    40,467      40,694      41,386      41,476      41,979
    Thisted      Egedal      Frederikssund      Vordingborg      Skive
    42,380      42,468      44,067      44,589      44,955
    Hedensted      Favrskov      Kalundborg      Ballerup      Varde
    45,360      47,058      47,407      47,497      48,674
    Høje-Taastrup      Greve      Hillerød      Fredericia      Faaborg-Midtfyn
    48,911      49,317      49,782      50,079      50,376
    Hvidovre      Lyngby-Taarbæk      Haderslev      Rudersdal      Ringkøbing-Skjern
    52,174      54,181      54,369      54,984      55,025
    Holstebro      Svendborg      Aabenraa      Frederikshavn      Guldborgsund
    56,848      57,155      57,325      58,184      59,285
    Køge      Skanderborg      Helsingør      Hjørring      Gladsaxe
    59,459      60,690      61,127      63,024      66,870
    Holbæk      Sønderborg      Gentofte      Slagelse      Næstved
    69,503      72,207      72,433      77,015      80,731
    Roskilde      Herning      Horsens      Kolding      Silkeborg
    84,864      86,164      86,986      89,867      90,493
    Viborg      Randers      Frederiksberg      Vejle      Esbjerg
    94,276      94,975      99,554      111,105      112,228
    Odense      Aalborg      Aarhus      København
    195,820      206,597      331,846      593,258
> with(prv20, fCp(table(findec, exclude = NULL)))
findec
      1      2      3      4      5      6      7      8      9
548,797 561,311 563,369 563,367 563,681 563,518 563,547 563,589 563,347
      10
563,057
> with(prv20, rCp(table(udddk, eduen , exclude = NULL)))
                    eduen
udddk      Bachelor or equivalent Doctoral or equivalent
Adgangsgivende uddannelsesforløb      .

```

Bacheloruddannelser, BACH	96,806	.	.
Erhvervsfaglige uddannelser	.	.	.
Grundskole	.	.	.
Gymnasiale uddannelser	.	.	.
Korte videregående uddannelser, KVV	.	.	.
Lange videregående uddannelser, LVU	.	.	.
Mellemlange videregående uddannelser, MVU	690,785	.	.
Ph.d. og forskeruddannelser	.	37,534	.
Uoplyst mv.	.	.	.
	eduen		
udddk	Lower secondary	Master or equivalent	
Adgangsgivende uddannelsesforløb	.	.	.
Bacheloruddannelser, BACH	.	.	.
Erhvervsfaglige uddannelser	.	.	.
Grundskole	1,335,769	.	.
Gymnasiale uddannelser	.	.	.
Korte videregående uddannelser, KVV	.	.	.
Lange videregående uddannelser, LVU	.	438,018	.
Mellemlange videregående uddannelser, MVU	.	.	.
Ph.d. og forskeruddannelser	.	.	.
Uoplyst mv.	.	*	.
	eduen		
udddk	Not elsewhere classified	Primary	
Adgangsgivende uddannelsesforløb	.	.	.
Bacheloruddannelser, BACH	.	.	.
Erhvervsfaglige uddannelser	.	.	.
Grundskole	12,986	81,682	.
Gymnasiale uddannelser	.	.	.
Korte videregående uddannelser, KVV	.	.	.
Lange videregående uddannelser, LVU	.	.	.
Mellemlange videregående uddannelser, MVU	.	.	.
Ph.d. og forskeruddannelser	.	.	.
Uoplyst mv.	808,957	.	.
	eduen		
udddk	Short cycle tertiary	Upper secondary	
Adgangsgivende uddannelsesforløb	.	3,814	.
Bacheloruddannelser, BACH	.	.	.
Erhvervsfaglige uddannelser	.	1,460,061	.
Grundskole	.	.	.
Gymnasiale uddannelser	.	428,118	.
Korte videregående uddannelser, KVV	218,298	.	.
Lange videregående uddannelser, LVU	.	.	.
Mellemlange videregående uddannelser, MVU	4,754	.	.
Ph.d. og forskeruddannelser	.	.	.
Uoplyst mv.	.	.	.

From the cross-tabulation of the Danish (udddk) versus the English (eduen) classifications of educational level it is clearly preferable to use the Danish as it separates KVV and MVU

1.3 Convenient with aggergation?

The dataset `prv20` has about 5.6 mio records, so we might consider tabulating by the variables of interest—those for which we did the marginal tables above. Suppose we cut the ages to 100 classes, then we would (theoretically) get a number of categories like

```
> fCp(3 * 2 * 100 * 99 * 10 * 10)
```

```
[1] 5,940,000
```

—that is a bit more, but not all would be populated, so we check how many records we actually get by aggregation:

```
> ppp <- mutate(subset(prv20, doBth > 1920),
+               en = 1,
+               age = floor(2020 - doBth))
> system.time(
+ agg <- aggregate(ppp$en,
+                 ppp[,c("DMst", "age", "sex", "reg", "kom", "udddk", "findec)],
+                 FUN = sum)
+               bruger    system forløbet
+               48.94     0.42    49.36
> str(agg, v = 0)
'data.frame':      859128 obs. of  8 variables:
 $ DMst  : Factor w/ 3 levels "noDM","T2","T1": NULL ...
 $ age   : num  NULL ...
 $ sex   : Factor w/ 2 levels "M","W": NULL ...
 $ reg   : Factor w/ 5 levels "Nordjylland",...: NULL ...
 $ kom   : Factor w/ 99 levels "København","Frederiksberg",...: NULL ...
 $ udddk : Factor w/ 10 levels "Adgangsgivende uddannelsesforløb",...: NULL ...
 $ findec: Factor w/ 10 levels "1","2","3","4",...: NULL ...
 $ x     : num  NULL ...
> summary(agg)
  DMst      age      sex      reg      kom
noDM:681390  Min.   : 1.00  M:437055  Nordjylland: 87191  København   : 19914
T2  :149673  1st Qu.:37.00  W:422073  Midtjylland:169853  Aarhus      : 17229
T1  : 28065  Median :53.00          Syddanmark :186179  Odense      : 14650
      Mean  :52.65          Hovedstaden:266469  Aalborg     : 14545
      3rd Qu.:69.00          Sjælland   :149436  Frederiksberg: 12957
      Max.  :99.00          Vejle      : 11950
                                   (Other)    :767883
                                udddk      findec      x
Grundskole                :188649  4      : 91880  Min.   : 1.000
Erhvervsfaglige uddannelser :185327  3      : 91279  1st Qu.: 1.000
Mellemlange videregående uddannelser, MVU:127249  5      : 90753  Median : 2.000
Gymnasiale uddannelser      : 84120  6      : 88676  Mean   : 6.537
Korte videregående uddannelser, KVU      : 81409  2      : 88535  3rd Qu.: 6.000
Lange videregående uddannelser, LVU      : 79333  7      : 87303  Max.   :3154.000
(Other)                :113041  (Other):320702
```

So we see that there are much fewer records in the tabulated dataset, somewhat less than 1 mio. records. Hence we shall use the aggregated data for the analyses:

2 Simple model for the aggregate data set

We can fit two models; one with `region` and one with `kommune`, note the latter takes much longer time to fit because of the many parameters, so in the following we will use the

```
> system.time(
+ reg0 <- glm((DMst == "T2") ~ 0 + reg,
+           family = binomial,
+           weight = x,
+           data = agg))
```

```

bruger    system forløbet
  4.52      0.21      4.72

> system.time(
+ kom0 <- glm((DMst == "T2") ~ 0 + kom,
+           family = binomial,
+           weight = x,
+           data = agg))

bruger    system forløbet
  53.24      1.15      54.37

> round(ci.exp(reg0), 3)

                exp(Est.)  2.5% 97.5%
regNordjylland    0.056 0.055 0.057
regMidtjylland    0.048 0.047 0.048
regSyddanmark     0.057 0.056 0.057
regHovedstaden    0.048 0.048 0.048
regSjælland       0.064 0.063 0.064

> round(ci.exp(kom0), 3)

                exp(Est.)  2.5% 97.5%
komKøbenhavn      0.035 0.035 0.036
komFrederiksberg  0.038 0.037 0.040
komBallerup       0.062 0.060 0.064
komBrøndby        0.080 0.077 0.084
komDragør         0.049 0.046 0.053
komGentofte       0.036 0.034 0.037
komGladsaxe       0.049 0.047 0.051
komGlostrup      0.064 0.061 0.068
komHerlev         0.061 0.058 0.064
komAlbertslund   0.074 0.071 0.078
komHvidovre       0.062 0.060 0.064
komHøje-Taastrup 0.069 0.067 0.072
komLyngby-Taarbæk 0.039 0.037 0.041
komRødovre        0.058 0.055 0.060
komIshøj          0.084 0.080 0.088
komTårnby         0.056 0.054 0.059
komVallensbæk    0.054 0.050 0.057
komFuresø         0.045 0.043 0.047
komAllerød        0.044 0.041 0.046
komFredensborg   0.054 0.052 0.057
komHelsingør     0.060 0.058 0.062
komHillerød      0.045 0.043 0.047
komHørsholm      0.049 0.047 0.052
komRudersdal     0.040 0.038 0.041
komEgedal        0.050 0.048 0.052
komFrederikssund 0.065 0.062 0.067
komGreve         0.058 0.055 0.060
komKøge          0.058 0.056 0.060
komHalsnæs       0.078 0.074 0.081
komRoskilde       0.046 0.044 0.047
komSolrød        0.050 0.047 0.053
komGribskov      0.066 0.063 0.068
komOdsherred     0.083 0.080 0.087
komHolbæk        0.061 0.059 0.063
komFaxe          0.063 0.060 0.066
komKalundborg    0.074 0.072 0.077
komRingsted      0.060 0.057 0.062

```

komSlagelse	0.067	0.065	0.069
komStevns	0.065	0.062	0.069
komSorø	0.062	0.059	0.065
komLejre	0.051	0.049	0.054
komLolland	0.088	0.085	0.091
komNæstved	0.062	0.060	0.063
komGuldborgsund	0.072	0.070	0.075
komVordingborg	0.072	0.069	0.075
komBornholm	0.076	0.073	0.079
komMiddelfart	0.058	0.055	0.060
komChristiansø	0.012	0.002	0.083
komAssens	0.060	0.058	0.063
komFaaborg-Midtfyn	0.056	0.053	0.058
komKerteminde	0.058	0.055	0.062
komNyborg	0.064	0.061	0.067
komOdense	0.047	0.046	0.048
komSvendborg	0.053	0.051	0.055
komNordfyns	0.061	0.058	0.064
komLangeland	0.083	0.078	0.089
komÆrø	0.071	0.064	0.079
komHaderslev	0.063	0.061	0.066
komBillund	0.059	0.056	0.062
komSønderborg	0.068	0.066	0.070
komTønder	0.072	0.069	0.075
komEsbjerg	0.055	0.053	0.056
komFanø	0.063	0.054	0.073
komVarde	0.055	0.052	0.057
komVejen	0.060	0.058	0.063
komAabenraa	0.068	0.066	0.071
komFredericia	0.059	0.057	0.062
komHorsens	0.050	0.048	0.052
komKolding	0.052	0.051	0.054
komVejle	0.048	0.046	0.049
komHerning	0.054	0.053	0.056
komHolstebro	0.051	0.049	0.053
komLemvig	0.064	0.061	0.068
komStruer	0.063	0.060	0.067
komSyddjurs	0.053	0.051	0.056
komNorrdjurs	0.063	0.060	0.066
komFavrskov	0.046	0.044	0.048
komOdder	0.052	0.049	0.055
komRanders	0.053	0.052	0.055
komSilkeborg	0.043	0.042	0.045
komSamsø	0.072	0.063	0.082
komSkanderborg	0.041	0.039	0.043
komAarhus	0.038	0.037	0.038
komIkast-Brande	0.053	0.051	0.056
komRingkøbing-Skjern	0.058	0.056	0.061
komHedensted	0.050	0.048	0.053
komMorsø	0.071	0.067	0.075
komSkive	0.056	0.054	0.058
komThisted	0.061	0.058	0.063
komViborg	0.048	0.047	0.050
komBrønderslev	0.058	0.056	0.061
komFrederikshavn	0.066	0.064	0.069
komVesthimmerlands	0.065	0.062	0.068
komLæsø	0.096	0.082	0.114

komRebild	0.048	0.045	0.051
komMariagerfjord	0.061	0.059	0.064
komJammerbugt	0.064	0.062	0.067
komAalborg	0.046	0.045	0.047
komHjørring	0.060	0.058	0.062

The parameters here are however odds not probabilities, if we want probabilities we must transform—and multiply by 100 to get percentages

```
> o2p <- function(o) o / (1 + o)
> round(o2p(ci.exp(reg0)) * 100, 1)
      exp(Est.) 2.5% 97.5%
regNordjylland      5.3 5.2 5.4
regMidtjylland      4.6 4.5 4.6
regSyddanmark       5.3 5.3 5.4
regHovedstaden      4.6 4.6 4.6
regSjælland         6.0 5.9 6.0

> round(o2p(ci.exp(kom0)) * 100, 1)
      exp(Est.) 2.5% 97.5%
komKøbenhavn        3.4 3.4 3.5
komFrederiksberg    3.7 3.6 3.8
komBallerup         5.8 5.6 6.0
komBrøndby          7.4 7.2 7.7
komDragør           4.7 4.4 5.1
komGentofte         3.5 3.3 3.6
komGladsaxe         4.7 4.5 4.8
komGlostrup        6.0 5.7 6.4
komHerlev           5.7 5.5 6.0
komAlbertslund      6.9 6.6 7.2
komHvidovre         5.8 5.6 6.0
komHøje-Taastrup   6.5 6.3 6.7
komLyngby-Taarbæk   3.7 3.6 3.9
komRødovre          5.4 5.2 5.7
komIshøj            7.7 7.4 8.1
komTårnby           5.3 5.1 5.5
komVallensbæk       5.1 4.8 5.4
komFuresø           4.3 4.1 4.5
komAllerød          4.2 3.9 4.4
komFredensborg      5.2 5.0 5.4
komHelsingør        5.6 5.5 5.8
komHillerød         4.3 4.2 4.5
komHørsholm        4.7 4.4 5.0
komRudersdal        3.8 3.7 4.0
komEgedal           4.8 4.6 5.0
komFrederikssund    6.1 5.9 6.3
komGreve            5.5 5.3 5.7
komKøge             5.5 5.3 5.7
komHalsnæs          7.2 6.9 7.5
komRoskilde          4.4 4.2 4.5
komSolrød           4.8 4.5 5.1
komGribskov         6.2 5.9 6.4
komOdsherred        7.7 7.4 8.0
komHolbæk           5.8 5.6 6.0
komFaxe             5.9 5.7 6.2
komKalundborg       6.9 6.7 7.2
komRingsted         5.6 5.4 5.9
```

komSlagelse	6.3	6.1	6.5
komStevns	6.1	5.8	6.5
komSorø	5.9	5.6	6.1
komLejre	4.9	4.6	5.2
komLolland	8.1	7.8	8.4
komNæstved	5.8	5.6	6.0
komGuldborgsund	6.8	6.6	7.0
komVordingborg	6.7	6.5	7.0
komBornholm	7.0	6.8	7.3
komMiddelfart	5.4	5.2	5.7
komChristiansø	1.2	0.2	7.7
komAssens	5.7	5.4	5.9
komFaaborg-Midtfyn	5.3	5.1	5.5
komKerteminde	5.5	5.2	5.8
komNyborg	6.0	5.8	6.3
komOdense	4.5	4.4	4.6
komSvendborg	5.0	4.9	5.2
komNordfyns	5.8	5.5	6.0
komLangeland	7.7	7.2	8.2
komÆrø	6.6	6.0	7.3
komHaderslev	6.0	5.8	6.2
komBillund	5.6	5.3	5.9
komSønderborg	6.4	6.2	6.5
komTønder	6.7	6.4	6.9
komEsbjerg	5.2	5.0	5.3
komFanø	5.9	5.2	6.8
komVarde	5.2	5.0	5.4
komVejen	5.7	5.5	5.9
komAabenraa	6.4	6.2	6.6
komFredericia	5.6	5.4	5.8
komHorsens	4.8	4.6	4.9
komKolding	5.0	4.8	5.1
komVejle	4.5	4.4	4.7
komHerning	5.2	5.0	5.3
komHolstebro	4.8	4.7	5.0
komLemvig	6.1	5.7	6.4
komStruer	6.0	5.6	6.3
komSyddjurs	5.1	4.9	5.3
komNorddjurs	5.9	5.7	6.2
komFavrskov	4.4	4.2	4.6
komOdder	4.9	4.7	5.2
komRanders	5.0	4.9	5.2
komSilkeborg	4.1	4.0	4.3
komSamsø	6.7	5.9	7.6
komSkanderborg	3.9	3.8	4.1
komAarhus	3.6	3.6	3.7
komIkast-Brande	5.1	4.9	5.3
komRingkøbing-Skjern	5.5	5.3	5.7
komHedensted	4.8	4.6	5.0
komMorsø	6.6	6.3	7.0
komSkive	5.3	5.1	5.5
komThisted	5.7	5.5	6.0
komViborg	4.6	4.5	4.8
komBrønderslev	5.5	5.3	5.8
komFrederikshavn	6.2	6.0	6.4
komVesthimmerlands	6.1	5.8	6.3
komLæsø	8.8	7.5	10.2

komRebild	4.6	4.3	4.8
komMariagerfjord	5.8	5.6	6.0
komJammerbugt	6.0	5.8	6.3
komAalborg	4.4	4.3	4.5
komHjørring	5.7	5.5	5.8

2.1 Prevalences relative to the population mean

But we also want the **relative** prevalences—relative to the overall mean prevalence, which is just to take the area-specific prevalences and divide by the overall prevalence:

```
> prpop <- with(agg, sum((DMst == "T2") * x) / sum(x))
> round(cbind(ci.exp(reg0)[,1] / prpop), 2)
      [,1]
regNordjylland 1.12
regMidtjylland 0.95
regSyddanmark  1.13
regHovedstaden 0.96
regSjælland    1.27
```

—but the exercise is not quite so trivial when it comes to *adjusted* prevalence ratios.

2.2 Controlling for age

We can expand and fit a model with control for age—we should then use the binomial with log link. The logit link will not provide *relative* prevalences, and there is no way to derive these from odds ratios.

The binomial model with log link does however not work well, it almost always crashes immediately due to silly string values. The trick to alleviate this behaviour is to get a close approximation to the parameters by fitting a Poisson model first, and then use the parameters from this as starting values for fitting of the log-link binomial model.

The point here is that the Poisson and binomial models are very close (at least for small-ish probabilities), so the parameters from a Poisson model (which has the log as the canonical link and therefor almost always converges) are bound to be close to the parameters from the binomial model with log link.

So here is the trick controlling for age using a natural spline, and using the `update` function to make sure that the fitted model is actually the same:

```
> Xa20 <- glm(DMst %in% "T2" ~ reg +
+           Ns(age, knots = 1:9*10),
+           weight = x,
+           family = poisson,
+           data = agg)
> cfX <- coef(Xa20)
> Ra20 <- update(Xa20,
+               start = cfX,
+               family = binomial(link = log))
```

This model now have parameters that are log-prevalences for each area. But we would like the *relative* prevalences for each kommune using the entire country as a reference.

Now, we have area parameters, θ_k , the first of which is the reference area. But we want to scale them relative to an overall average, θ_{DK} , say.

The population average parameter, θ_{DK} , is an area parameter defined so that when multiplied to the estimated age-effects (which are those for the reference area), and these adjusted age-effects are used on each person in the population and added over all persons we get the total number of diabetes cases in the population.

So first we need the overall no. cases, persons and prevalence:

```
> (Ndia <- sum((agg$DMst == "T2") * agg$x))
[1] 281695
> (Npop <- sum(agg$x))
[1] 5616449
> (Praw <- Ndia / Npop)
[1] 0.05015536
```

To find the right scaling we make a prediction for all persons in the population, but assuming they all were from the reference area. We can get the predicted values of the prevalences assuming everyone is in the reference area:

```
> (aref <- levels(agg$reg)[1])
[1] "Nordjylland"
> prvPr <- predict(Ra20,
+               newdata = mutate(agg, reg = aref),
+               type = "response")
> round(summary(prvPr) * 100, 4)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.0007  1.1650  5.6769  7.1406 13.0013 17.2279
```

What we get in `prvPr` are the predicted probabilities (prevalences), for each row in the dataset. But each row represents x persons, so the total expected number of T2 patients under the assumption that they all reside in the reference area, but otherwise have their own age etc. is:

```
> sum(prvPr * agg$x)
[1] 280516.1
```

If this were equal to the total no. of observed cases, then the reference area would be spot on the DK average. But if it is smaller, say, then the reference area has a prevalence smaller than the DK average, and hence it (and all other area parameters) should be adjusted downwards by the ratio of the expected numbers to the observed, to produce prevalence ratios relative to the DK mean:

```
> # area parameters scaled to make the reference have expected
> # number equal to observed:
> RPscale <- sum(prvPr * agg$x) / Npop
> # Relative Prevalence referring to DK total
> RP <- c(1, ci.exp(Ra20, subset = "reg")[,1]) * RPscale
> names(RP) <- levels(agg$reg)
> cbind(RP)
      RP
Nordjylland 0.04994546
Midtjylland 0.04677780
Syddanmark  0.05030162
Hovedstaden 0.05077926
Sjælland    0.05362587
```

If we would rather express the variation in terms of prevalence (as a proportion) we can just multiply the RP with the overall raw prevalence:

```
> cbind(RPr <- RP * Praw)
           [,1]
Nordjylland 0.002505032
Midtjylland 0.002346157
Syddanmark  0.002522895
Hovedstaden 0.002546852
Sjælland    0.002689624
```

Note that this derivation of the relative prevalences adjusted for age only involved age via the model object, so the same pieces of code will work for model controlling for further variables:

3 General function to estimate relative prevalences

We now piece together the code above into functions that produces the relative prevalences controlled at various levels and for different subsets

```
> # convenience values to check function during development
> type <- "T2"
> model <- ~ reg + Ns(age, knots = 1:9 * 10)
> geo <- NULL #"reg"
> adat <- subset(agg, sex = "M")
> # here is the function that fits the model and derives Rprv and Aprv
> fitprev <-
+ function(type, # character vector of outcome (T1, T2 or c(T1,T2) - quoted though)
+             model, # what model to use, a one-sided formula, with the geo (kom/reg)
+             # as the first term
+             adat) # the dataset
+ {
+ # We need the raw prevalence in the data set analysed
+ Ndia <- sum((adat$DMst %in% type) * adat$x)
+ Npop <- sum(adat$x)
+ Praw <- Ndia / Npop
+
+ # manipulate the formula
+ formul <- (DMst %in% type) ~ 1
+ formul[3] <- model[2]
+
+ # Poisson model
+ Pmod <- glm(formul,
+             weight = x,
+             family = poisson,
+             data = adat)
+ # Binomial log-link model based on Poisson model parameters
+ cfP <- coef(Pmod)
+ Bmod <- update(Pmod,
+               start = cfP,
+               family = binomial(link = log))
+
+ # find the reference level for the geo and put in the nd data frame
+ geo <- attr(terms(Pmod), "term.labels")[1]
+ georef <- levels(adat[, geo])[1]
+ nd <- adat
```

```

+ nd[,geo] <- georef
+
+ # predicted prevalences as if all were in "georef"
+ prvPr <- predict(Bmod,
+                 newdata = nd,
+                 type = "response")
+
+ # adjust to Rprv relative to general mean
+ RPscale <- sum(prvPr * adat$x) / Ndia
+
+ # Relative Prevalence referring to DK total
+ RP <- c(1, ci.exp(Bmod, subset = geo)[,1]) * RPscale
+ names(RP) <- levels(agg[,geo])
+
+ # add the Adjusted prevalence
+ cbind(Rel = RP, Adj = RP * Praw)
+ }

```

With the fitting function designed we can now test it for men and women from 2020:

```

> system.time(
+ Wa <- fitprev(type = "T2",
+              model = ~ reg + Ns(age, knots = 1:9 * 10),
+              adat = subset(agg, sex == "W"))
  bruger    system forløbet
    7.48     0.53     8.01
> system.time(
+ Ma <- fitprev(type = "T2",
+              model = ~ reg + Ns(age, knots = 1:9 * 10),
+              adat = subset(agg, sex == "M"))
  bruger    system forløbet
    9.11     0.59     9.70
> cbind(Ma, Wa)
              Rel      Adj      Rel      Adj
Nordjylland 0.9822290 0.05456699 1.0068963 0.04513140
Midtjylland 0.9233265 0.05129470 0.9405992 0.04215981
Syddanmark  1.0028846 0.05571449 0.9991770 0.04478540
Hovedstaden 1.0237465 0.05687346 1.0084253 0.04519993
Sjælland    1.0733253 0.05962777 1.0624808 0.04762282

```

3.1 Data preparation in general

Now, the function `fitprev` relies on the input dataset (argument `adat`) which in this example above was the 2020 dataset, but we need to have a function that generates the dataset for any date (`yy`) we choose:

```

> mkdat <-
+ function(yy)
+ {
+ # selects records that contain the date yy:
+ prv <- DMfup[DMfup$per <= yy &
+             yy < (DMfup$per + DMfup$lex.dur),
+             c("lex.id", "sex", "per", "doBth", "lex.Cst")]
+ }

```

```

+ # population information at this date
+ var <- filter(popstat, yr == yy)
+
+ # merge it with the prevalence data, only using common persons
+ prv <- inner_join(prv, var, by = c("lex.id" = "pnr"))
+
+ # restrict to those with known geography (those from popstat)
+ prv <- ( unLexis(prv)
+   %>% filter(!is.na(kom) & !is.na(findec))
+   %>% mutate(udddk = factor(ifelse(is.na(udddk),
+                                   "Uoplyst mv.",
+                                   as.character(udddk))),
+             DMst = factor(lex.Cst),
+             age = floor(yy - doBth),
+             en = 1)
+   %>% filter(age < 100)
+   )
+
+ # aggregate
+ system.time(
+ agg <- aggregate(prv$en,
+                  prv[,c("DMst", "age", "sex", "reg", "kom", "udddk", "findec")],
+                  FUN = sum)
+ agg
+ }
> system.time(agg20 <- mkdat(2020))
  bruger    system forløbet
  66.90     1.52     68.42

```

3.2 Collecting results

The model fitting and data generation functions are now used to fill an array of relative and adjusted prevalences:

```

> atlas <- NArray(list(date = 1996:2020,
+                       type = c("T1", "T2"),
+                       sex = c("M", "W", "M+W"),
+                       adj = c("None", "+age", "+edu", "+inc"),
+                       kom = levels(popstat$kom),
+                       mod = c("Rel", "Adj")))

```

But we may have to run this programme many times, resuming after previous runs, so we load the current status of the atlas file and show what parts of it that has been filled:

```

> load(file = "atlas.Rnw", v = T)
Loading objects:
  atlas
> str(atlas)
 num [1:25, 1:2, 1:3, 1:4, 1:99, 1:2] NA NA NA NA NA ...
- attr(*, "dimnames")=List of 6
 ..$ date: chr [1:25] "1996" "1997" "1998" "1999" ...
 ..$ type: chr [1:2] "T1" "T2"
 ..$ sex : chr [1:3] "M" "W" "M+W"
 ..$ adj : chr [1:4] "None" "+age" "+edu" "+inc"
 ..$ kom : chr [1:99] "København" "Frederiksberg" "Ballerup" "Brøndby" ...
 ..$ mod : chr [1:2] "Rel" "Adj"

```

```
> round(ftable(atlas[,"T2",1,2,1,], row.vars = 1), 8)
```

date	mod	Rel	Adj
1996		NA	NA
1997		NA	NA
1998		NA	NA
1999		NA	NA
2000		NA	NA
2001		NA	NA
2002	1.42529937	0.02657899	
2003	1.41583215	0.02889575	
2004	1.39523577	0.03157563	
2005	1.37252871	0.03409185	
2006	1.35913168	0.03619281	
2007	1.33861773	0.03796496	
2008	1.32071209	0.03992256	
2009	1.29023557	0.04156095	
2010	1.25982422	0.04362982	
2011	1.25566431	0.04707895	
2012	1.23775337	0.05129125	
2013	1.22554191	0.05415171	
2014	1.21549862	0.05556999	
2015	1.19043517	0.05632513	
2016	1.18350817	0.05788777	
2017	1.13686628	0.05783962	
2018	1.11387080	0.05820670	
2019	1.09711479	0.06168053	
2020	1.06395543	0.05910724	

```
> for(yy in 2001:1996) # the dates 2020:2002 were run earlier
+ {
+   agg <- mkdat(yy)
+   for(sx in c("M+W","M","W"))
+   for(tp in c("T1","T2"))
+   {
+     cat(yy, sx, tp, format(Sys.time(), "%F, %T, "))
+     if (sx == "M+W")
+     {
+       adt <- agg
+       atlas[paste(yy), tp, sx, "None",,] <-
+         fitprev(tp, ~ kom + sex, adat = adt)
+       cat(" None")
+       atlas[paste(yy), tp, sx, "+age",,] <-
+         fitprev(tp, ~ kom + sex + Ns(age, knots = 1:9*10), adat = adt)
+       cat("+age")
+       atlas[paste(yy), tp, sx, "+edu",,] <-
+         fitprev(tp, ~ kom + sex + Ns(age, knots = 1:9*10) + udddk, adat = adt)
+       cat("+edu")
+       atlas[paste(yy), tp, sx, "+inc",,] <-
+         fitprev(tp, ~ kom + sex + Ns(age, knots = 1:9*10) + udddk + findec, adat = adt)
+       cat("+inc")
+     }
+   }
+   else
+   {
+     adt <- subset(agg, sex == sx)
+     atlas[paste(yy), tp, sx, "None",,] <-
```

```

+     fitprev(tp, ~ kom, adat = adt)
+     cat(" None")
+ atlas[paste(yy), tp, sx, "+age",,] <-
+     fitprev(tp, ~ kom + Ns(age, knots = 1:9*10), adat = adt)
+     cat("+age")
+ atlas[paste(yy), tp, sx, "+edu",,] <-
+     fitprev(tp, ~ kom + Ns(age, knots = 1:9*10) + udddk, adat = adt)
+     cat("+edu")
+ atlas[paste(yy), tp, sx, "+inc",,] <-
+     fitprev(tp, ~ kom + Ns(age, knots = 1:9*10) + udddk + findec, adat = adt)
+     cat("+inc")
+   }
+ save(atlas, file = "atlas.Rnw")
+ cat(" - saved\n")
+ flush.console() # print buffered output to the results file
+ }
+ }

```

```

2001 M+W T1 2022-05-16, 05:59:37, None+age+edu+inc - saved
2001 M+W T2 2022-05-16, 06:11:27, None+age+edu+inc - saved
2001 M T1 2022-05-16, 06:23:47, None+age+edu+inc - saved
2001 M T2 2022-05-16, 06:29:46, None+age+edu+inc - saved
2001 W T1 2022-05-16, 06:35:49, None+age+edu+inc - saved
2001 W T2 2022-05-16, 06:41:58, None+age+edu+inc - saved
2000 M+W T1 2022-05-16, 06:49:08, None+age+edu+inc - saved
2000 M+W T2 2022-05-16, 07:01:17, None+age+edu+inc - saved
2000 M T1 2022-05-16, 07:14:52, None+age+edu+inc - saved
2000 M T2 2022-05-16, 07:22:47, None+age+edu+inc - saved
2000 W T1 2022-05-16, 07:29:50, None+age+edu+inc - saved
2000 W T2 2022-05-16, 07:38:27, None+age+edu+inc - saved
1999 M+W T1 2022-05-16, 07:46:33, None+age+edu+inc - saved
1999 M+W T2 2022-05-16, 08:00:07, None+age+edu+inc - saved
1999 M T1 2022-05-16, 08:13:34, None+age+edu+inc - saved
1999 M T2 2022-05-16, 08:19:35, None+age+edu+inc - saved
1999 W T1 2022-05-16, 08:26:39, None+age+edu+inc - saved
1999 W T2 2022-05-16, 08:33:16, None+age+edu+inc - saved
1998 M+W T1 2022-05-16, 08:40:13, None+age+edu+inc - saved
1998 M+W T2 2022-05-16, 08:52:02, None+age+edu+inc - saved
1998 M T1 2022-05-16, 09:04:50, None+age+edu+inc - saved
1998 M T2 2022-05-16, 09:11:03, None+age+edu+inc - saved
1998 W T1 2022-05-16, 09:19:01, None+age+edu+inc - saved
1998 W T2 2022-05-16, 09:29:25, None+age+edu+inc - saved
1997 M+W T1 2022-05-16, 09:38:59, None+age+edu+inc - saved
1997 M+W T2 2022-05-16, 09:50:02, None+age+edu+inc - saved
1997 M T1 2022-05-16, 10:03:27, None+age+edu+inc - saved
1997 M T2 2022-05-16, 10:09:17, None+age+edu+inc - saved
1997 W T1 2022-05-16, 10:16:17, None+age+edu+inc - saved
1997 W T2 2022-05-16, 10:22:53, None+age+edu+inc - saved
1996 M+W T1 2022-05-16, 10:30:48, None+age+edu+inc - saved
1996 M+W T2 2022-05-16, 10:42:53, None+age+edu+inc - saved
1996 M T1 2022-05-16, 10:56:12, None+age+edu+inc - saved
1996 M T2 2022-05-16, 11:01:57, None+age+edu+inc - saved
1996 W T1 2022-05-16, 11:08:37, None+age+edu+inc - saved
1996 W T2 2022-05-16, 11:14:55, None+age+edu+inc - saved

```

```

-----
Code: E:/workdata/707655/users/BxC/atlas/r/mkAtlas.rnw

```

```
Ends: 2022-05-16 at 11:21:23
Time elapsed:      05:34:27
```

4 Data for diabetes map of Denmark

4.1 Paraphernalia

```
> options(width = 90)
> library(tidyverse)
> library(Epi)
> clear()
> source("E:/workdata/707655/util/elapsed.R")
> start()
```

```
-----
Code: E:/workdata/707655/users/BxC/atlas/r/getAtlas.rnw
Time: 2022-06-21 at 14:59:46
-----
```

4.2 The estimates

```
> setwd("E:/workdata/707655/users/BxC/atlas/R")
> load("atlas.rda", v = TRUE)
Loading objects:
  atlas
> str(atlas)
 num [1:25, 1:2, 1:3, 1:4, 1:99, 1:2] 1.04 1.01 1.02 1 1.01 ...
- attr(*, "dimnames")=List of 6
 ..$ date: chr [1:25] "1996" "1997" "1998" "1999" ...
 ..$ type: chr [1:2] "T1" "T2"
 ..$ sex : chr [1:3] "M" "W" "M+W"
 ..$ adj : chr [1:4] "None" "+age" "+edu" "+inc"
 ..$ kom : chr [1:99] "København" "Frederiksberg" "Ballerup" "Brøndby" ...
 ..$ mod : chr [1:2] "Rel" "Adj"
```

4.3 The counts

We forgot the counts, there are a few that are very small:

```
> system.time(load("prv20.Rda"))
 bruger    system forløbet
 28.16      0.36    29.05
> Np <- with(prv20, addmargins(table(yr, DMst, sex, kom), 2:3))
> str(Np)
'table' num [1, 1:4, 1:3, 1:99] 279731 11038 1502 292271 290674 ...
- attr(*, "dimnames")=List of 4
 ..$ yr : chr "2020"
 ..$ DMst: chr [1:4] "noDM" "T2" "T1" "Sum"
 ..$ sex : chr [1:3] "M" "W" "Sum"
 ..$ kom : chr [1:99] "København" "Frederiksberg" "Ballerup" "Brøndby" ...
```

```
> range(Np)
[1]      0 593258
> Np[Np < 4] <- NA
```

4.4 Data dump

```
> dump(c("atlas","Np"), file = "atlas.R")
```

5 Data files for the atlas

First we read the datafile of estimates exported from DST:

```
> source("atlas.R", encoding = "UTF-8")
> library(Epi)
> lls()
  name mode  class      dim      size(Kb)
1 atlas numeric array    25 2 3 4 99 2      938.8
2 Np      numeric table array 1 4 3 99      17.7
> str(atlas)
num [1:25, 1:2, 1:3, 1:4, 1:99, 1:2] 1.04 1.01 1.02 1 1.01 ...
- attr(*, "dimnames")=List of 6
..$ date: chr [1:25] "1996" "1997" "1998" "1999" ...
..$ type: chr [1:2] "T1" "T2"
..$ sex : chr [1:3] "M" "W" "M+W"
..$ adj : chr [1:4] "None" "+age" "+edu" "+inc"
..$ kom : chr [1:99] "Koebenhavn" "Frederiksberg" "Ballerup" "Broendby" ...
..$ mod : chr [1:2] "Rel" "Adj"
> str(Np)
'table' num [1, 1:4, 1:3, 1:99] 279731 11038 1502 292271 290674 ...
- attr(*, "dimnames")=List of 4
..$ yr : chr "2020"
..$ DMst: chr [1:4] "noDM" "T2" "T1" "Sum"
..$ sex : chr [1:3] "M" "W" "Sum"
..$ kom : chr [1:99] "Koebenhavn" "Frederiksberg" "Ballerup" "Broendby" ...
```

Then we make a separate file for each sex, type of diabetes, type of measure (R for relative prevalence, A for adjuste prevalence) and the 4 levels of adjustment:

```
> M1R <- cbind(N = Np["2020", "T1", "M", ],
+             t(atlas["2020", "T1", "M", , , "Rel"]))
> M1A <- cbind(N = Np["2020", "T1", "M", ],
+             t(atlas["2020", "T1", "M", , , "Adj"] * 100))
> M2R <- cbind(N = Np["2020", "T2", "M", ],
+             t(atlas["2020", "T2", "M", , , "Rel"]))
> M2A <- cbind(N = Np["2020", "T2", "M", ],
+             t(atlas["2020", "T2", "M", , , "Adj"] * 100))
> W1R <- cbind(N = Np["2020", "T1", "W", ],
+             t(atlas["2020", "T1", "W", , , "Rel"]))
> W1A <- cbind(N = Np["2020", "T1", "W", ],
+             t(atlas["2020", "T1", "W", , , "Adj"] * 100))
```

```

> W2R <- cbind(N = Np["2020", "T2", "W", ],
+             t(atlas["2020", "T2", "W", , , "Rel"]))
> W2A <- cbind(N = Np["2020", "T2", "W", ],
+             t(atlas["2020", "T2", "W", , , "Adj"] * 100))
> MW1R <- cbind(N = Np["2020", "T1", "Sum", ],
+             t(atlas["2020", "T1", "M+W", , , "Rel"]))
> MW1A <- cbind(N = Np["2020", "T1", "Sum", ],
+             t(atlas["2020", "T1", "M+W", , , "Adj"] * 100))
> MW2R <- cbind(N = Np["2020", "T2", "Sum", ],
+             t(atlas["2020", "T2", "M+W", , , "Rel"]))
> MW2A <- cbind(N = Np["2020", "T2", "Sum", ],
+             t(atlas["2020", "T2", "M+W", , , "Adj"] * 100))
> str(MW2R)
 num [1:99, 1:5] 20205 3670 2768 2552 663 ...
- attr(*, "dimnames")=List of 2
 ..$ : chr [1:99] "Koebenhavn" "Frederiksberg" "Ballerup" "Broendby" ...
 ..$ : chr [1:5] "N" "None" "+age" "+edu" ...
> round(MW2A, 2)

```

	N	None	+age	+edu	+inc
Koebenhavn	20205	3.41	5.30	5.65	5.29
Frederiksberg	3670	3.70	4.25	4.86	4.89
Ballerup	2768	5.83	5.65	5.67	5.73
Broendby	2552	7.44	7.47	7.10	6.91
Dragoer	663	4.71	4.12	4.36	5.11
Gentofte	2501	3.47	3.36	4.13	4.78
Gladsaxe	3110	4.65	5.21	5.51	5.61
Glostrup	1335	6.05	6.26	6.15	6.19
Herlev	1615	5.76	5.91	5.94	6.00
Albertslund	1797	6.91	7.18	7.21	7.06
Hvidovre	3040	5.83	6.44	6.30	6.34
Hoeje-Taastrup	3174	6.48	6.95	6.78	6.94
Lyngby-Taarbaek	2024	3.74	3.79	4.48	4.88
Roedovre	2124	5.46	5.77	5.75	5.74
Ishoej	1725	7.72	8.75	8.20	8.06
Taarnby	2229	5.31	5.45	5.26	5.56
Vallensbaek	819	5.08	5.47	5.61	6.22
Furesoe	1706	4.28	4.04	4.69	5.29
Alleroed	1048	4.19	3.99	4.50	5.23
Fredensborg	2059	5.18	4.63	5.08	5.45
Helsingoer	3452	5.66	4.89	5.13	5.30
Hilleroed	2157	4.34	4.39	4.72	5.08
Hoersholm	1149	4.73	3.78	4.43	5.17
Rudersdal	2100	3.83	3.44	4.18	4.98
Egedal	2018	4.75	4.83	5.01	5.63
Frederikssund	2674	6.07	5.26	5.26	5.42
Greve	2689	5.45	5.17	5.19	5.66
Koege	3267	5.49	5.40	5.30	5.44
Halsnaes	2202	7.21	5.89	5.80	5.77
Roskilde	3718	4.38	4.37	4.66	4.97
Solroed	1078	4.79	4.81	4.91	5.55
Gribskov	2467	6.16	4.90	4.99	5.29
Odsherred	2473	7.70	5.61	5.41	5.27
Holbaek	4015	5.78	5.44	5.35	5.38
Faxe	2117	5.93	5.28	5.07	5.07
Kalundborg	3283	6.92	5.90	5.60	5.56
Ringsted	1909	5.62	5.76	5.57	5.63
Slagelse	4859	6.31	5.89	5.64	5.60

Stevns	1360	6.12	5.08	4.92	5.05
Soroe	1707	5.86	5.37	5.28	5.30
Lejre	1332	4.90	4.45	4.57	4.88
Lolland	3278	8.09	5.98	5.54	5.10
Naestved	4678	5.80	5.39	5.22	5.23
Guldborgsund	4007	6.76	5.40	5.16	4.92
Vordingborg	2996	6.72	5.29	5.15	4.99
Bornholm	2720	7.03	5.25	5.01	4.71
Middelfart	2039	5.44	4.84	4.79	4.81
Christiansoe	NA	1.20	1.00	1.00	1.11
Assens	2279	5.66	4.98	4.78	4.61
Faaborg-Midtfyn	2652	5.26	4.52	4.39	4.27
Kerteminde	1271	5.50	4.57	4.43	4.32
Nyborg	1886	6.04	5.19	5.03	4.88
Odense	8845	4.52	5.09	5.12	4.98
Svendborg	2879	5.03	4.43	4.46	4.36
Nordfyns	1658	5.74	5.03	4.75	4.60
Langeland	932	7.69	5.14	4.83	4.46
Aeroe	385	6.61	4.24	4.16	3.92
Haderslev	3237	5.95	5.31	5.09	4.91
Billund	1439	5.58	5.23	4.85	4.91
Soenderborg	4588	6.35	5.45	5.33	5.09
Toender	2433	6.67	5.57	5.20	4.98
Esbjerg	5800	5.16	4.96	4.81	4.73
Fanoe	196	5.93	4.07	4.43	4.53
Varde	2516	5.16	4.65	4.39	4.39
Vejen	2353	5.66	5.33	4.98	4.87
Aabenraa	3663	6.39	5.59	5.31	5.12
Fredericia	2804	5.59	5.37	5.19	5.13
Horsens	4138	4.75	5.00	4.88	4.88
Kolding	4483	4.99	5.09	4.99	5.00
Vejle	5039	4.53	4.59	4.52	4.58
Herning	4439	5.14	5.17	4.92	4.97
Holstebro	2755	4.84	4.74	4.55	4.58
Lemvig	1171	6.04	4.84	4.54	4.53
Struer	1222	5.95	4.94	4.69	4.60
Syddjurs	2102	5.07	4.24	4.23	4.28
Norddjurs	2144	5.90	4.96	4.66	4.51
Favrskov	2081	4.42	4.54	4.48	4.64
Odder	1089	4.95	4.39	4.45	4.63
Randers	4790	5.04	4.91	4.74	4.61
Silkeborg	3754	4.15	4.12	4.12	4.19
Samsøe	238	6.72	4.34	4.27	4.08
Skanderborg	2389	3.93	4.16	4.28	4.52
Aarhus	12034	3.63	4.62	4.99	5.07
Ikast-Brande	2027	5.05	4.91	4.56	4.53
Ringkoebing-Skjern	3035	5.50	4.98	4.67	4.67
Hedensted	2173	4.78	4.60	4.34	4.42
Morsoe	1314	6.63	5.40	4.99	4.74
Skive	2386	5.30	4.61	4.36	4.27
Thisted	2426	5.71	4.99	4.68	4.56
Viborg	4353	4.61	4.51	4.40	4.43
Broenderslev	1958	5.51	5.06	4.79	4.63
Frederikshavn	3617	6.21	4.97	4.69	4.59
Vesthimmerlands	2169	6.06	5.26	4.88	4.72
Laesoe	153	8.78	5.21	4.82	4.57
Rebild	1329	4.57	4.53	4.40	4.42

Mariagerfjord	2349	5.76	5.08	4.82	4.70
Jammerbugt	2244	6.02	5.21	4.91	4.80
Aalborg	9102	4.40	4.85	4.84	4.78
Hjoerring	3569	5.66	4.96	4.72	4.61

So we now have 12 arrays of dimension 99×5 , with rows equal to municipality, and columns equal to counts and 4 different levels of adjustment:

None / Age / Age + Education / Age + Education + Income-decile.

Files are for either of the two sexes or both and the measure either being prevalences relative to the country average (R) or adjusted prevalences (A).

```
> write.csv2(M1R , file = "./data/M1R.csv")
> write.csv2(M1A , file = "./data/M1A.csv")
> write.csv2(M2R , file = "./data/M2R.csv")
> write.csv2(M2A , file = "./data/M2A.csv")
> write.csv2(W1R , file = "./data/W1R.csv")
> write.csv2(W1A , file = "./data/W1A.csv")
> write.csv2(W2R , file = "./data/W2R.csv")
> write.csv2(W2A , file = "./data/W2A.csv")
> write.csv2(MW1R, file = "./data/MW1R.csv")
> write.csv2(MW1A, file = "./data/MW1A.csv")
> write.csv2(MW2R, file = "./data/MW2R.csv")
> write.csv2(MW2A, file = "./data/MW2A.csv")
```