Mortality and morbidity among DN patients: Data set up

GbAd / PRo, SDC
http://BendixCarstensen.com/SDC/Nefro/
December 2013
Version 5.0

Compiled Tuesday 24th December, 2013, 12:19
from: c:/Bendix/Steno/GbAd/DNx.tex

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

Contents

Chapter 1

Reading and setting up data

We first use SAS to convert data to SAS-xport format:

```
1                                  "Program: get-nef.sas"       14:02 Monday, December 23, 2013

NOTE: Copyright (c) 2002-2008 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software 9.2 (TS2M3)
      Licensed to NOVO NORDISK - BASIC PACKAGE, Site 50800704.
NOTE: This session is executing on the W32_VSPRO  platform.



NOTE: SAS initialization used:
      real time           2.19 seconds
      cpu time            0.40 seconds


NOTE: AUTOEXEC processing beginning; file is c:\stat\sas\autoexec.sas.

---------------------------------------------------------------
C:\Bendix\Steno\GbAd\sas\get-nef.sas
---------------------------------------------------------------
NOTE: Libref HER was successfully assigned as follows:
      Engine:        V9
      Physical Name: C:\Bendix\Steno\GbAd\sas
NOTE: Libref DATA was successfully assigned as follows:
      Engine:        V9
      Physical Name: C:\Bendix\Steno\GbAd\data

NOTE: AUTOEXEC processing completed.

1          data nef ;
2            set data.bendix_revised4 ;
3          run ;

NOTE: There were 1040 observations read from the data set DATA.BENDIX_REVISED4.
NOTE: The data set WORK.NEF has 1040 observations and 25 variables.
NOTE: DATA statement used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds


4
5          proc contents data = nef ;
6          run ;

NOTE: PROCEDURE CONTENTS used (Total process time):
      real time           0.07 seconds
      cpu time            0.04 seconds

NOTE: The PROCEDURE CONTENTS printed page 1.

7
8          options validvarname=V6 ;
9          libname xptout xport '../data/nefro.xpt';
NOTE: Libref XPTOUT was successfully assigned as follows:
      Engine:        XPORT
      Physical Name: C:\Bendix\Steno\GbAd\data\nefro.xpt
10
11         proc copy in = work
12                  out = xptout
13               memtype = data ;
14                select nef ;
15         run;

NOTE: Copying WORK.NEF to XPTOUT.NEF (memtype=DATA).
```

```
NOTE: The variable name vaerdi_Gfr has been truncated to vaerdi_G.
NOTE: The variable vaerdi_G now has a label set to vaerdi_Gfr.
NOTE: The variable name vaerdi_haemo has been truncated to vaerdi_h.
NOTE: The variable vaerdi_h now has a label set to vaerdi_haemo.
NOTE: The variable name vaerdi_ldl has been truncated to vaerdi_l.
NOTE: The variable vaerdi_l now has a label set to vaerdi_ldl.
NOTE: The variable name vaerdi_tchol has been truncated to vaerdi_t.
NOTE: The variable vaerdi_t now has a label set to vaerdi_tchol.
NOTE: The variable label geometrisk mean af alle dualb eller alb/crea målinger, eller den måling som er tættest på bl
      has been truncated to geometrisk mean af alle dualb eller alb/.
NOTE: The variable name vaerdi_hba1c has been truncated to vaerdi_h.
NOTE: The variable vaerdi_h now has a label set to vaerdi_hba1c.
NOTE: Variable VAERDI_H already exists on file XPTOUT.NEF, using VAERDI_2 instead.
NOTE: There were 1040 observations read from the data set WORK.NEF.
NOTE: The data set XPTOUT.NEF has 1040 observations and 25 variables.
NOTE: PROCEDURE COPY used (Total process time):
      real time           0.22 seconds
      cpu time            0.01 seconds


NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
NOTE: The SAS System used:
      real time           2.66 seconds
      cpu time            0.53 seconds

The SAS System                                                          14:02 Monday, December 23, 2013   1

The CONTENTS Procedure

Data Set Name        WORK.NEF                              Observations          1040
Member Type          DATA                                  Variables             25
Engine               V9                                    Indexes               0
Created              23. december 2013 mandag 14:02:11     Observation Length    200
Last Modified        23. december 2013 mandag 14:02:11     Deleted Observations  0
Protection                                                 Compressed            NO
Data Set Type                                              Sorted                NO
Label
Data Representation  WINDOWS_32
Encoding             wlatin1  Western (Windows)


                              Engine/Host Dependent Information

Data Set Page Size          16384
Number of Data Set Pages    14
First Data Page             1
Max Obs per Page            81
Obs in First Data Page      62
Number of Data Set Repairs  0
Filename                    C:\Users\BXC\AppData\Local\Temp\SAS Temporary Files\_TD8028\nef.sas7bdat
Release Created             9.0202M3
Host Created                W32_VSPRO


                                 Alphabetic List of Variables and Attributes

 #  Variable    Type  Len  Format     Informat  Label

 2  DM_type     Char    5
14  Sys_bt      Num     8
19  alb         Num     8                       geometrisk mean af alle dualb eller alb/crea målinger,
                                                eller den måling som er tættest på bl
13  bmi         Num     8  COMMAX8.1
 9  cauDTH      Char    6                       cause of death
 3  doBIRTH     Num     8  DATE9.     DATE9.    date of birth
24  doCVD       Num     8
 7  doDTH       Num     8  DATE9.     DATE9.    date of death
 6  doESRD      Num     8  DATE9.
 5  doNRA       Num     8  DATE9.               date of inclusion
 8  doX         Num     8  DATE9.               end of follow up
 1  id          Num     8  BEST12.    F12.      patient id
21  ins_kg      Num     8  COMMAX8.2            IE insulin/kg bodyweight
25  priorCVD    Num     8
22  rygning     Num     8                       0=never, 1=ever
23  rygning2    Num     8                       0=nej, 1=<3/tidligere 2=ja/3-20 3=>pakke
 4  sex         Num     8                       1=mand, 2=kvinde, beregnet fra CPR
10  substudy    Num     8                       included in deltaGFR 1=yes
15  vaerdi_Gfr  Num     8
16  vaerdi_     Num     8
    haemo
20  vaerdi_     Num     8
    hba1c
17  vaerdi_ldl  Num     8
18  vaerdi_     Num     8
    tchol
12  var_dm      Num     8                       varighed af diabetes i år
11  var_dn      Num     8                       varighed af nefropati i år
```

Then we import the SAS-xport dataset, transform variable names to lower case, and convert "_"s to "."s:

```
> library(foreign)
> library(Epi)
> library(splines)
> print( sessionInfo(), locale=FALSE )
  R version 3.0.2 (2013-09-25)
  Platform: i386-w64-mingw32/i386 (32-bit)

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

  other attached packages:
  [1] Epi_1.1.60     foreign_0.8-55

  loaded via a namespace (and not attached):
  [1] tools_3.0.2
> ga <- read.xport( "./data/nefro.xpt" )
> names( ga ) <- tolower( names(ga) )
> names( ga ) <- gsub( "_", ".", names(ga) )
> head( ga )
        id dm.type dobirth sex donra doesrd dodth   dox caudth substudy var.dn var.dm      bmi
  1 1256414   type2  -10957   1 14793     NA    NA 18627                0      9     13 21.03550
  2 1256416   type2   -9131   1 14984     NA 18084 18084 unknow         1      2     13 30.88889
  3 1256417   type2   -8035   1 14663     NA    NA 18627                1      0     10 22.30061
  4 1256419   type2   -7305   1 14886  15460 16804 16804 kidney         0      5      9       NA
  5 1256420   type2   -7305   1 15027     NA 16769 16769 cancer         1      5      5 28.54545
  6 1256422   type2   -5478   2 14788  17391    NA 18627                1      1     10 18.98693
    sys.bt vaerdi.g vaerdi.h vaerdi.l vaerdi.t  alb vaerdi.2    ins.kg rygning rygning2 docvd
  1    141       78      8.9      2.1      3.9   92      8.8 0.7876231       1        1    NA
  2    159       93      8.8      2.7      5.6  457     10.0 1.4388489       0        0 12329
  3    233      105      9.2      3.1      4.8 1122     14.4        NA       1        1 14627
  4    164       24      7.0      2.3      3.8  965      5.4        NA       1        1 13367
  5    166       99      9.5      2.8      4.7 1104      9.6        NA       1        1 13391
  6    143       58      8.0      3.2      5.3  307     10.2 1.1703959       1        2    NA
    priorcvd
  1        0
  2        1
  3        1
  4        1
  5        1
  6        0
```

We then transform the date-variables to fractional calendar years:

```
> dv <- grep( "do", names(ga) )
> names( ga )[dv]
  [1] "dobirth" "donra"   "doesrd"  "dodth"   "dox"     "docvd"
> ga[,dv] <- ga[,dv]/365.25+1960
```

Finally we prepend a "D" to the labeling of the death-states and save two different subsets of data, one with type 1 data and one with type 2 data:

```
> levels( ga$caudth ) <- paste( "D-", substr(levels( ga$caudth ),1,3), sep="" )
> levels( ga$caudth )[1] <- "DN"
```

We also need to define sex and smoking as factors, the latter appropriately grouped:

```
> ga$sex <- factor( ga$sex, levels=2:1, labels=c("F","M") )
> ga$smoke <- factor( ga$rygning2, labels=c("never","<3","4-20","20+") )
> ga$smoke <- Relevel( ga$smoke, list(1:2,3:4) )
```

We also change the variable names to make the subsequent code more readable:

```
> newnames <- c("gfr","hmgb","ldl","tchol","hba1c")
> oldnames <- names(ga)[c(15:18,20)]
> cbind( oldnames, newnames )
      oldnames   newnames
 [1,] "vaerdi.g" "gfr"
 [2,] "vaerdi.h" "hmgb"
 [3,] "vaerdi.l" "ldl"
 [4,] "vaerdi.t" "tchol"
 [5,] "vaerdi.2" "hba1c"

> names(ga)[match(oldnames,names(ga))] <- newnames
> summary( ga[,c("sex","smoke","dm.type",
+                "bmi","sys.bt","alb","gfr","ins.kg","hba1c","hmgb","tchol")])
   sex          smoke       dm.type         bmi             sys.bt           alb
   F:317   never+<3:713   type1:497   Min.   :13.74   Min.   :104.0   Min.   :    6.0
   M:723   4-20+20+:327   type2:543   1st Qu.:21.03   1st Qu.:134.0   1st Qu.:  227.0
                                      Median :23.77   Median :144.0   Median :  484.0
                                      Mean   :24.61   Mean   :145.7   Mean   :  848.5
                                      3rd Qu.:27.38   3rd Qu.:156.0   3rd Qu.: 1000.0
                                      Max.   :50.35   Max.   :233.0   Max.   :10962.0
                                      NA's   :7       NA's   :10      NA's   :7
      gfr             ins.kg           hba1c            hmgb            tchol
   Min.   :  7.00   Min.   : 0.0000   Min.   : 5.200   Min.   : 4.500   Min.   : 1.800
   1st Qu.: 48.00   1st Qu.: 0.4837   1st Qu.: 7.600   1st Qu.: 7.500   1st Qu.: 4.100
   Median : 70.00   Median : 0.6970   Median : 8.700   Median : 8.200   Median : 4.850
   Mean   : 72.29   Mean   : 1.0495   Mean   : 8.754   Mean   : 8.184   Mean   : 4.949
   3rd Qu.: 94.00   3rd Qu.: 1.0954   3rd Qu.: 9.700   3rd Qu.: 8.900   3rd Qu.: 5.700
   Max.   :178.00   Max.   :28.7751   Max.   :16.300   Max.   :12.300   Max.   :11.100
                    NA's   :200       NA's   :1        NA's   :1        NA's   :2


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

1.1 `Lexis` object with causes of death

First we just take a look at the distribution of the CVD-dates relative to the DN-dates, colored by the variable `priorcvd`:

```
> load( file="./data/ga.Rda" )
> library( Epi )
> with( ga, plot( docvd, donra,
+                 col=c("red","blue")[2-priorcvd], pch=16, cex=0.5) )
> abline(0,1)
> with( ga, table( priorcvd, "has CVD"=!is.na(docvd), exclude=NULL ) )
         has CVD
priorcvd FALSE TRUE <NA>
    0      474  228    0
    1        0  338    0
    <NA>     0    0    0
```

The small table shows that if there is no CVD date then the prior CVD variable is 0. It also has the implication that when we later cut the follow-up at `docvd`, that the 338 persons that have a prior CVD will start in the CVD state.

We store all data in a Lexis object allowing us to see the possible transitions between states. First we set up a object representing time from DN to death, with exit stats equal to cause of death, and time scales age, calendar time, time sine enty to the study (essentially time since mesurement of the clinical variables), time since DN and duration of diabetes.

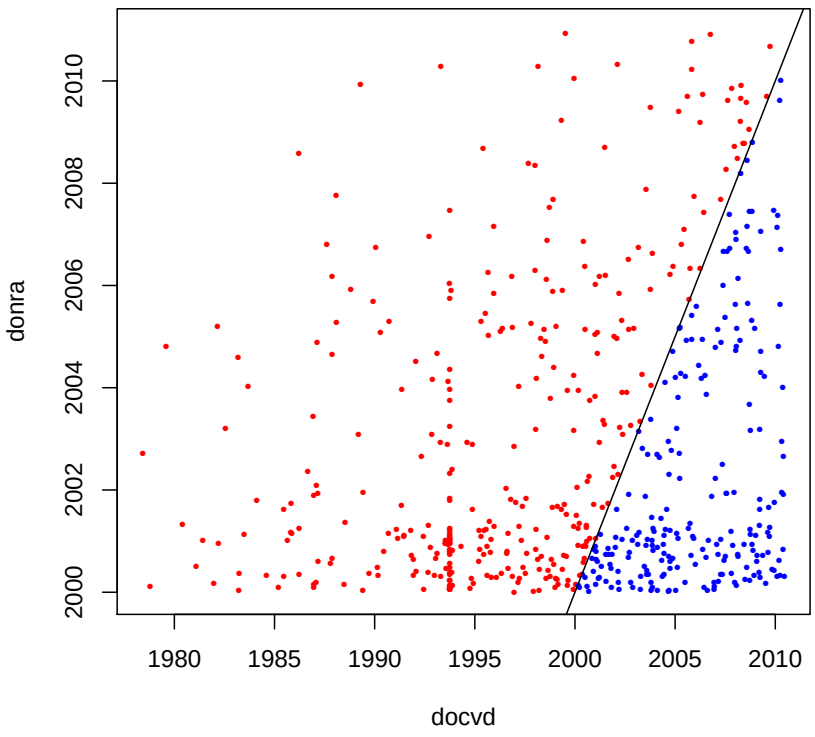

Figure 1.1: *Plotting the date of DN versus the date of CVD shows that the* priorcvd *id coded correctly.*

```
> L1 <- Lexis( data = ga,
+              entry = list( age = donra-dobirth,
+                            per = donra,
+                            tfi = 0,
+                            tfn = var.dn,
+                            dur = var.dm ),
+               exit = list( per=dox ),
+        exit.status = caudth )
  NOTE: entry.status has been set to "DN" for all.

> summary( L1 )

  Transitions:
       To
  From  DN D-can D-CVD D-ext D-kid D-oth D-unk  Records:  Events: Risk time:  Persons:
    DN 704    30    98     5    26    57   120      1040      336    6977.18      1040
```

In order to accommodate the intermediate states CVD and ESRD and how they follow each other, we define subsets of the dataset (transitions) where CVD resp. ESRD is the first (or only) event, and subsets where these are the second.

```
> CVD.1 <- transform( subset(L1, !is.na(docvd) &
+                                pmin(docvd,doesrd,na.rm=TRUE)==docvd ),
+                     cut = docvd,
+                 new.state = "CVD" )[,c("lex.id","cut","new.state")]
> CVD.2 <- transform( subset(L1, !is.na(docvd) & !is.na(doesrd) &
+                                doesrd < docvd ),
+                     cut = docvd,
+                 new.state = "CVD(ESRD)" )[,c("lex.id","cut","new.state")]
> ESRD.1 <- transform( subset(L1, !is.na(doesrd) &
+                                 pmin(docvd,doesrd,na.rm=TRUE)==doesrd ),
+                     cut = doesrd,
+                 new.state = "ESRD" )[,c("lex.id","cut","new.state")]
> ESRD.2 <- transform( subset(L1, !is.na(docvd) & !is.na(doesrd) &
+                                 doesrd > docvd ),
+                     cut = doesrd,
+                 new.state = "ESRD(CVD)" )[,c("lex.id","cut","new.state")]
```

We check that we have not got the same persons in the two subsets for each diagnosis, and also that there is no overlap between those who have one sequence and those with another sequence:

```
> intersect( CVD.1$lex.id,
+            CVD.2$lex.id )
  integer(0)

> intersect( ESRD.1$lex.id,
+            ESRD.2$lex.id )

  integer(0)

> intersect(
+ intersect( CVD.1$lex.id,
+            ESRD.2$lex.id ),
+ intersect( ESRD.1$lex.id,
+            CVD.2$lex.id ) )

  integer(0)
```

We then cut the follow-up at the intermediate events, that is at date of CVD and date of ESRD, defining separate states for CVD alone and CVD following ESRD (and vice versa):

```
> L2 <- cutLexis( data = L1,
+                 cut = CVD.1,
+           timescale = "per",
+           new.scale = "tfCVD",
+    precursor.states = "DN" )
> L3 <- cutLexis( data = L2,
+                 cut = ESRD.1,
+           timescale = "per",
+           new.scale = "tfESRD",
+    precursor.states = "DN" )
> L4 <- cutLexis( data = L3,
+                 cut = ESRD.2,
+           timescale = "per",
+           new.scale = "tfEc",
+    precursor.states = c("DN","CVD") )
> L5 <- cutLexis( data = L4,
+                 cut = CVD.2,
+           timescale = "per",
+           new.scale = "tfCe",
+    precursor.states = c("DN","ESRD") )
> summary( L5 )
```

```
Transitions:
     To
From          DN ESRD CVD(ESRD) CVD ESRD(CVD) D-can D-CVD D-ext D-kid D-oth D-unk  Records:  Events
  DN         324  130         0 177         0    14     9     0     2    19    27       702      37
  ESRD         0   64        51   0         0     1     4     0     5     1     4       130       6
  CVD(ESRD)    0    0        25   0         0     0     6     0     2     2    16        51       2
  CVD          0    0         0 234       119    13    56     5     6    28    54       515      28
  ESRD(CVD)    0    0         0   0        57     2    23     0    11     7    19       119       6
  Sum        324  194        76 411       176    30    98     5    26    57   120      1517      81

Transitions:
     To
From         Risk time:  Persons:
  DN            3912.37       702
  ESRD           346.92       130
  CVD(ESRD)      107.88        51
  CVD           2266.85       515
  ESRD(CVD)      343.15       119
  Sum           6977.18      1040
```

We now have introduced the two auxilary timescales `tfEc` (time from ESRD after CVD) and `tfCe` (time from CVD after ESRD)[1], but we just want these to be time since `CVD+ESRD`, so we fix this:

To see which states are actually represented as ending states we make a colored plot of the transitions:

```
> xx <- c( 10, 20, 30, 20, 30, rep(90,6) )
> yy <- c( 50, 70, 90, 30, 10, seq(5,95,,6) )
> cbind( levels(L5$lex.Xst), xx, yy )
```

```
                    xx   yy
 [1,] "DN"          "10" "50"
 [2,] "ESRD"        "20" "70"
 [3,] "CVD(ESRD)"   "30" "90"
 [4,] "CVD"         "20" "30"
 [5,] "ESRD(CVD)"   "30" "10"
 [6,] "D-can"       "90" "5"
 [7,] "D-CVD"       "90" "23"
 [8,] "D-ext"       "90" "41"
 [9,] "D-kid"       "90" "59"
```

[1]The facility to recognize these by defining them with the same name as an existing time scale is not implented in `cutLexis` (yet).

```
  [10,] "D-oth"     "90" "77"
  [11,] "D-unk"     "90" "95"
```

```
> n5 <- sum( !is.na(t5<-tmat( L5 )) )
> tt5 <- t(t5)
> tt5[!is.na(tt5)] <- 1:n5
> t5 <- t(tt5)
> t5
```

```
             DN ESRD CVD(ESRD) CVD ESRD(CVD) D-can D-CVD D-ext D-kid D-oth D-unk
DN           NA    1        NA   2        NA     3     4    NA     5     6     7
ESRD         NA   NA         8  NA        NA     9    10    NA    11    12    13
CVD(ESRD)    NA   NA        NA  NA        NA    NA    14    NA    15    16    17
CVD          NA   NA        NA  NA        18    19    20    21    22    23    24
ESRD(CVD)    NA   NA        NA  NA        NA    25    26    NA    27    28    29
D-can        NA   NA        NA  NA        NA    NA    NA    NA    NA    NA    NA
D-CVD        NA   NA        NA  NA        NA    NA    NA    NA    NA    NA    NA
D-ext        NA   NA        NA  NA        NA    NA    NA    NA    NA    NA    NA
D-kid        NA   NA        NA  NA        NA    NA    NA    NA    NA    NA    NA
D-oth        NA   NA        NA  NA        NA    NA    NA    NA    NA    NA    NA
D-unk        NA   NA        NA  NA        NA    NA    NA    NA    NA    NA    NA
```

```
> t5[1,-(1:4)]
```

```
  ESRD(CVD)     D-can     D-CVD     D-ext     D-kid     D-oth     D-unk
         NA         3         4        NA         5         6         7
```

```
> a.col <- rep(par("fg"),sum(!is.na(t5)))
> a.col[t5[1,-(1:5)]] <- "red"
> a.col[t5[2,-(1:5)]] <- "blue"
> a.col[t5[3,-(1:5)]] <- "forestgreen"
> a.col[t5[4,-(1:5)]] <- "magenta"
> a.col[t5[5,-(1:5)]] <- "brown"
> TM <- tmat( L5 )
> boxes.matrix( L5,
+               boxpos=list(x=xx,y=yy), hmult=1.4, wmult=1.1,
+               scale.R = 100, show.D=TRUE, show.Y=TRUE, DR.sep=c(" (",")"),
+               col.arr = a.col )
```

We can now make a simplified object by pooling all the dead-states:

```
> L6 <- Relevel( L5, list(Dead=6:11), first=FALSE )
```

```
       type       old       new
1  lex.Cst        DN        DN
2  lex.Cst      ESRD      ESRD
3  lex.Cst CVD(ESRD) CVD(ESRD)
4  lex.Cst       CVD       CVD
5  lex.Cst ESRD(CVD) ESRD(CVD)
6  lex.Cst     D-can
7  lex.Cst     D-CVD
8  lex.Cst     D-ext
9  lex.Cst     D-kid
10 lex.Cst     D-oth
11 lex.Cst     D-unk
12 lex.Xst        DN        DN
13 lex.Xst      ESRD      ESRD
14 lex.Xst CVD(ESRD) CVD(ESRD)
15 lex.Xst       CVD       CVD
16 lex.Xst ESRD(CVD) ESRD(CVD)
17 lex.Xst     D-can      Dead
18 lex.Xst     D-CVD      Dead
19 lex.Xst     D-ext      Dead
20 lex.Xst     D-kid      Dead
21 lex.Xst     D-oth      Dead
22 lex.Xst     D-unk      Dead
```

```
> summary( L6, scale=1000 )
```

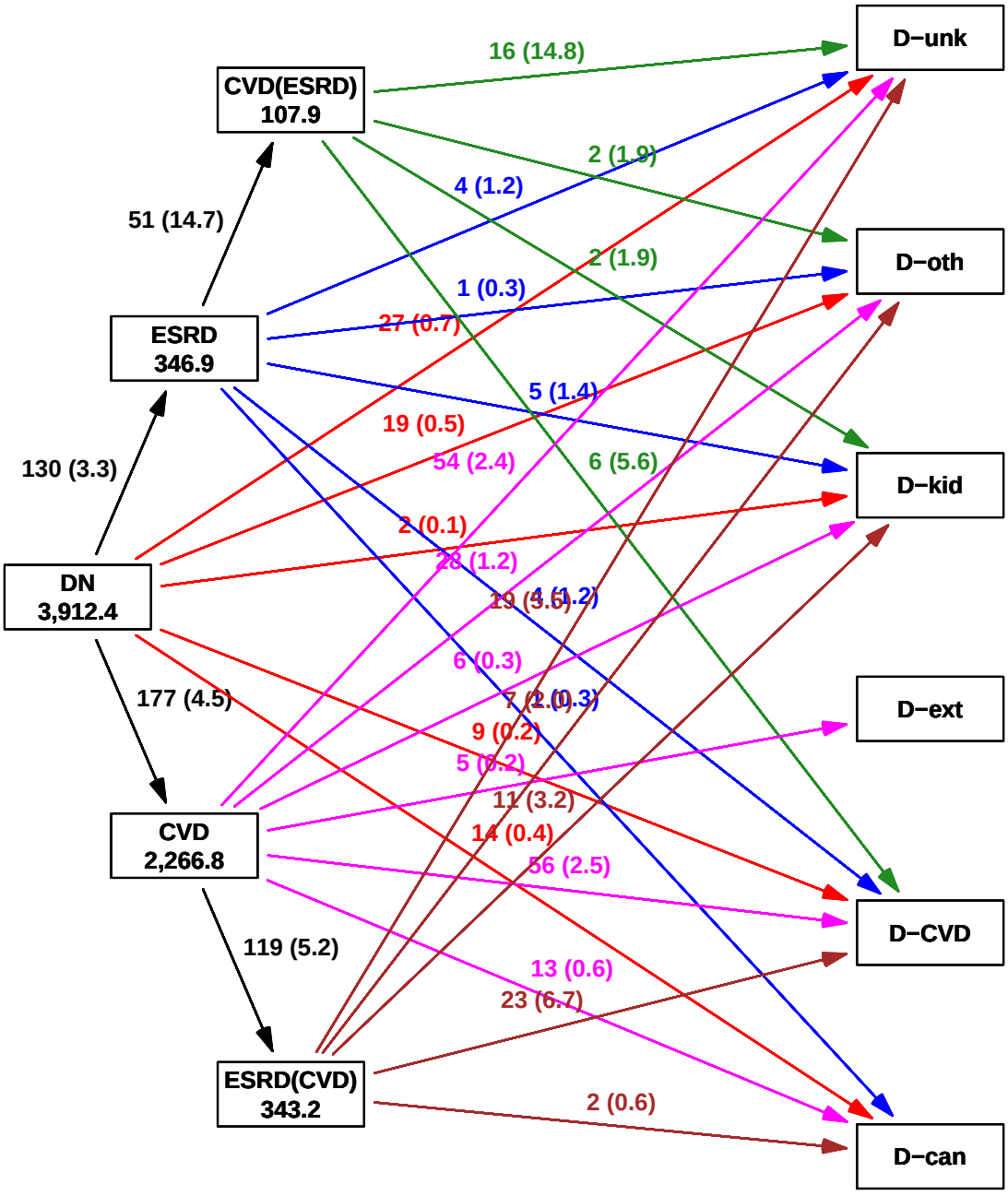

Figure 1.2: *States and transitions to different causes of death.*

```
  Transitions:
       To
  From          DN ESRD CVD(ESRD) CVD ESRD(CVD) Dead  Records:  Events: Risk time:  Persons:
    DN         324  130          0 177         0   71       702      378       3.91       702
    ESRD         0   64         51   0         0   15       130       66       0.35       130
    CVD(ESRD)    0    0         25   0         0   26        51       26       0.11        51
    CVD          0    0          0 234       119  162       515      281       2.27       515
    ESRD(CVD)    0    0          0   0        57   62       119       62       0.34       119
    Sum        324  194         76 411       176  336      1517      813       6.98      1040
> xx <- c( 10, 20, 30, 20, 30, 90 )
> yy <- c( 50, 70, 90, 30, 10, 50 )
> par( mfrow=c(2,1) )
> boxes.Lexis( subset( L6, dm.type=="type1" ),
+              boxpos=list(x=xx,y=yy), hmult=1.4, wmult=1.1,
+              scale.R = 100, show.D=TRUE, show.Y=TRUE,
+              DR.sep=c(" (",")") )
> text( 2, 98, "T1D", adj=c(0,1), font=2, cex=1.5 )
> boxes.Lexis( subset( L6, dm.type=="type2" ),
+              boxpos=list(x=xx,y=yy), hmult=1.4, wmult=1.1,
+              scale.R = 100, show.D=TRUE, show.Y=TRUE,
+              DR.sep=c(" (",")") )
> text( 2, 98, "T2D", adj=c(0,1), font=2, cex=1.5 )


> save( L5, L6, file="./data/CoD-Lexis.Rda" )
```

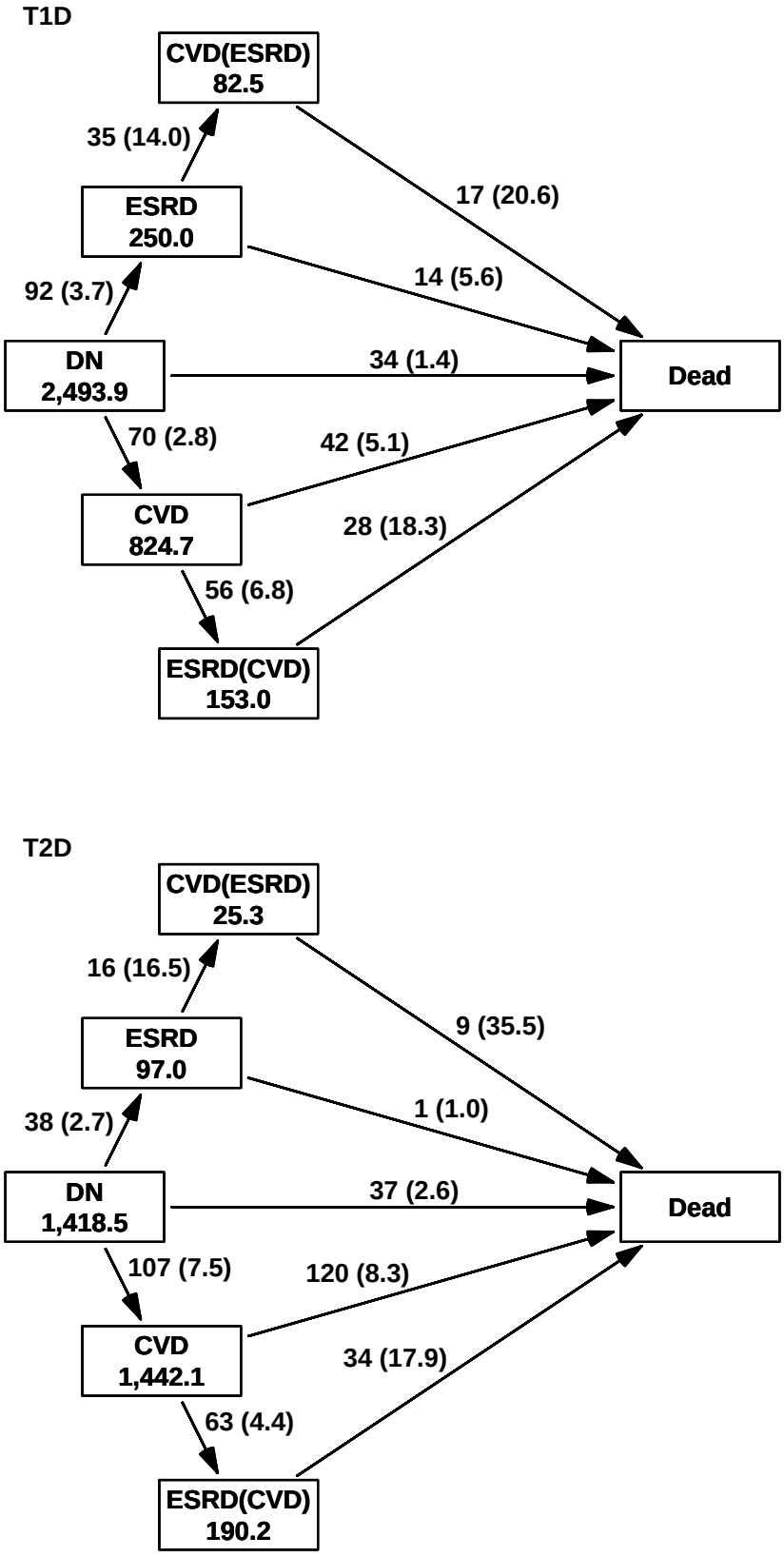

Figure 1.3: *States and transitions between them.*

1.2 `Lexis` object with death subdivided by state

Since we ultimately are going to be interested in how many patients have visited different states we shall instead subdivide the Death state according to the immediately preceding state. However, we shall first merge the states `ESRD(CVD)` and `CVD(ESRD)`, not distinguishing which of the conditons came first:

```
> options( width=90 )
> load( file="./data/CoD-Lexis.Rda" )
> L7 <- transform( L6, tfCE = pmax( tfCe, tfEc, na.rm=TRUE ) )
> L7 <- Relevel( L7, list("ESRD+CVD"=c(3,5)), first=FALSE )
       type       old      new
  1  lex.Cst        DN       DN
  2  lex.Cst      ESRD     ESRD
  3  lex.Cst CVD(ESRD) ESRD+CVD
  4  lex.Cst       CVD      CVD
  5  lex.Cst ESRD(CVD) ESRD+CVD
  6  lex.Cst      Dead
  7  lex.Xst        DN       DN
  8  lex.Xst      ESRD     ESRD
  9  lex.Xst CVD(ESRD) ESRD+CVD
  10 lex.Xst       CVD      CVD
  11 lex.Xst ESRD(CVD) ESRD+CVD
  12 lex.Xst      Dead     Dead

> attr( L7, "time.scales" ) <- c( attr( L7, "time.scales" )[1:7], "tfCE" )
> attr( L7, "time.since"  ) <- c( attr( L7, "time.since"  )[1:7], "ESRD+CVD" )
> attr( L7, "breaks" )      <- c( attr( L7, "breaks" )[1:7], list(tfCE=NULL) )
> L7 <- subset( L7, select = -c(tfEc,tfCe) )
> str( L7 )

  Classes 'Lexis' and 'data.frame':        1517 obs. of  38 variables:
   $ age     : num  70.5 66 62.1 60.8 62.3 ...
   $ per     : num  2001 2001 2000 2001 2002 ...
   $ tfi     : num  0 0 0 0 1.57 ...
   $ tfn     : num  9 2 0 5 6.57 ...
   $ dur     : num  13 13 10 9 10.6 ...
   $ tfCVD   : num  NA 7.269 0.0986 4.1588 5.7303 ...
   $ tfESRD  : num  NA NA NA NA NA NA NA 0 NA NA ...
   $ lex.dur : num  10.5 8.49 10.85 1.57 3.68 ...
   $ lex.Cst : Factor w/ 5 levels "DN","ESRD","CVD",..: 1 3 3 3 5 3 1 2 3 3 ...
   $ lex.Xst : Factor w/ 5 levels "DN","ESRD","CVD",..: 1 4 3 5 4 4 2 2 3 4 ...
   $ lex.id  : int  1 2 3 4 4 5 6 6 7 8 ...
   $ id      : num  1256414 1256416 1256417 1256419 1256419 ...
   $ dm.type : Factor w/ 2 levels "type1","type2": 2 2 2 2 2 2 2 2 2 1 ...
   $ dobirth : num  1930 1935 1938 1940 1940 ...
   $ sex     : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 1 1 2 1 ...
   $ donra   : num  2001 2001 2000 2001 2001 ...
   $ doesrd  : num  NA NA NA 2002 2002 ...
   $ dodth   : num  NA 2010 NA 2006 2006 ...
   $ dox     : num  2011 2010 2011 2006 2006 ...
   $ caudth  : Factor w/ 7 levels "DN","D-can","D-CVD",..: 1 7 1 5 5 2 1 1 1 3 ...
   $ substudy: num  0 1 1 0 0 1 1 1 1 0 ...
   $ var.dn  : num  9 2 0 5 5 5 1 1 12 8 ...
   $ var.dm  : num  13 13 10 9 9 5 10 10 12 29 ...
   $ bmi     : num  21 30.9 22.3 NA NA ...
   $ sys.bt  : num  141 159 233 164 164 166 143 143 138 145 ...
   $ gfr     : num  78 93 105 24 24 99 58 58 71 45 ...
   $ hmgb    : num  8.9 8.8 9.2 7 7 9.5 8 8 8.2 9.4 ...
   $ ldl     : num  2.1 2.7 3.1 2.3 2.3 2.8 3.2 3.2 2.4 3.1 ...
   $ tchol   : num  3.9 5.6 4.8 3.8 3.8 4.7 5.3 5.3 4.8 5.6 ...
   $ alb     : num  92 457 1122 965 965 ...
   $ hba1c   : num  8.8 10 14.4 5.4 5.4 9.6 10.2 10.2 9.9 7.4 ...
   $ ins.kg  : num  0.788 1.439 NA NA NA ...
   $ rygning : num  1 0 1 1 1 1 1 1 1 1 ...
```

```
 $ rygning2: num  1 0 1 1 1 1 2 2 1 2 ...
 $ docvd   : num  NA 1994 2000 1997 1997 ...
 $ priorcvd: num  0 1 1 1 1 1 0 0 1 1 ...
 $ smoke   : Factor w/ 2 levels "never+<3","4-20+20+": 1 1 1 1 1 1 2 2 1 2 ...
 $ tfCE    : num  NA NA NA NA 0 NA NA NA NA NA ...
 - attr(*, "breaks")=List of 8
  ..$ age   : NULL
  ..$ per   : NULL
  ..$ tfi   : NULL
  ..$ tfn   : NULL
  ..$ dur   : NULL
  ..$ tfCVD : NULL
  ..$ tfESRD: NULL
  ..$ tfCE  : NULL
 - attr(*, "time.scales")= chr  "age" "per" "tfi" "tfn" ...
 - attr(*, "time.since")= chr  "" "" "" "" ...
> summary( L7 )
  Transitions:
       To
  From         DN ESRD CVD Dead ESRD+CVD  Records:  Events: Risk time:  Persons:
    DN        324  130 177   71        0       702      378    3912.37       702
    ESRD        0   64   0   15       51       130       66     346.92       130
    CVD         0    0 234  162      119       515      281    2266.85       515
    ESRD+CVD    0    0   0   88       82       170       88     451.04       170
    Sum       324  194 411  336      252      1517      813    6977.18      1040
> rbind( timeScales(L7), attr(L7,"time.since") )
       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]    [,7]     [,8]
  [1,] "age" "per" "tfi" "tfn" "dur" "tfCVD" "tfESRD" "tfCE"
  [2,] ""    ""    ""    ""    ""    "CVD"   "ESRD"   "ESRD+CVD"
```

Then we can subdivide the death state according to the immediately preceding state:

```
> L7$lex.Xst <- with( L7,
+                     factor( ifelse( as.character(lex.Xst) == "Dead",
+                                     paste( as.character(lex.Xst),
+                                            "(",
+                                            as.character(lex.Cst),
+                                            ")", sep="" ),
+                                     as.character(lex.Xst) ) ) )
> L7$lex.Cst <- with( L7,
+                     factor( as.character(lex.Cst),
+                             levels=levels(lex.Xst) ) )
> L7 <- Relevel( L7, c(6,1,8,7,3,2,5,4) )
> summary( L7 )
  Transitions:
       To
  From         DN CVD ESRD+CVD ESRD Dead(DN) Dead(CVD) Dead(ESRD+CVD) Dead(ESRD)  Records:
    DN        324 177        0  130       71         0               0          0       702
    CVD         0 234      119    0        0       162               0          0       515
    ESRD+CVD    0   0       82    0        0         0              88          0       170
    ESRD        0   0       51   64        0         0               0         15       130
    Sum       324 411      252  194       71       162              88         15      1517

  Transitions:
       To
  From         Events: Risk time:  Persons:
    DN             378    3912.37       702
    CVD            281    2266.85       515
    ESRD+CVD        88     451.04       170
    ESRD            66     346.92       130
    Sum            813    6977.18      1040
```

The diagram of states would now have to be laid out slightly differently:

```
> bp <- list( x = c( 10, 40, 43, 19, 90, 90, 90, 90 ),
+             y = c( 95, 65, 35,  5, 95, 65, 35,  5 ) )
> boxes( L7, boxpos=bp, cex=1.2, lwd=3, wmult=1.1, hmult=1.3,
+        show.BE="nozero", scale.R=100, digits.R=1, DR.sep=c(" (",")"),
+        pos.arr=c(0.4,0.6)[c(1,2,1,1,1,1,2,1)] )
```

1.2.1 Illustrating the cuts of follow-up time

It is also illustrative to see how the records of a single person is cut up at the times of CVD and ESRD in the two instances, so we select two persons that actually transverse 3 different states:

```
> tt <- with( L7, table(lex.id) )
> ( who <- names( tt[tt>2] ) )
  [1] "19"   "20"   "23"   "51"   "76"   "81"   "89"   "101"  "107"  "113"  "119"  "126"
 [13] "130"  "151"  "155"  "162"  "167"  "170"  "172"  "193"  "199"  "202"  "211"  "220"
 [25] "226"  "227"  "242"  "249"  "267"  "283"  "284"  "298"  "299"  "300"  "308"  "318"
 [37] "331"  "371"  "378"  "379"  "407"  "427"  "429"  "441"  "470"  "501"  "508"  "509"
 [49] "510"  "515"  "517"  "527"  "528"  "542"  "548"  "563"  "567"  "584"  "587"  "606"
 [61] "651"  "654"  "659"  "660"  "673"  "690"  "692"  "702"  "703"  "707"  "729"  "735"
 [73] "740"  "750"  "751"  "753"  "800"  "810"  "839"  "854"  "868"  "871"  "876"  "882"
 [85] "890"  "901"  "905"  "917"  "940"  "949"  "1001" "1009"
```

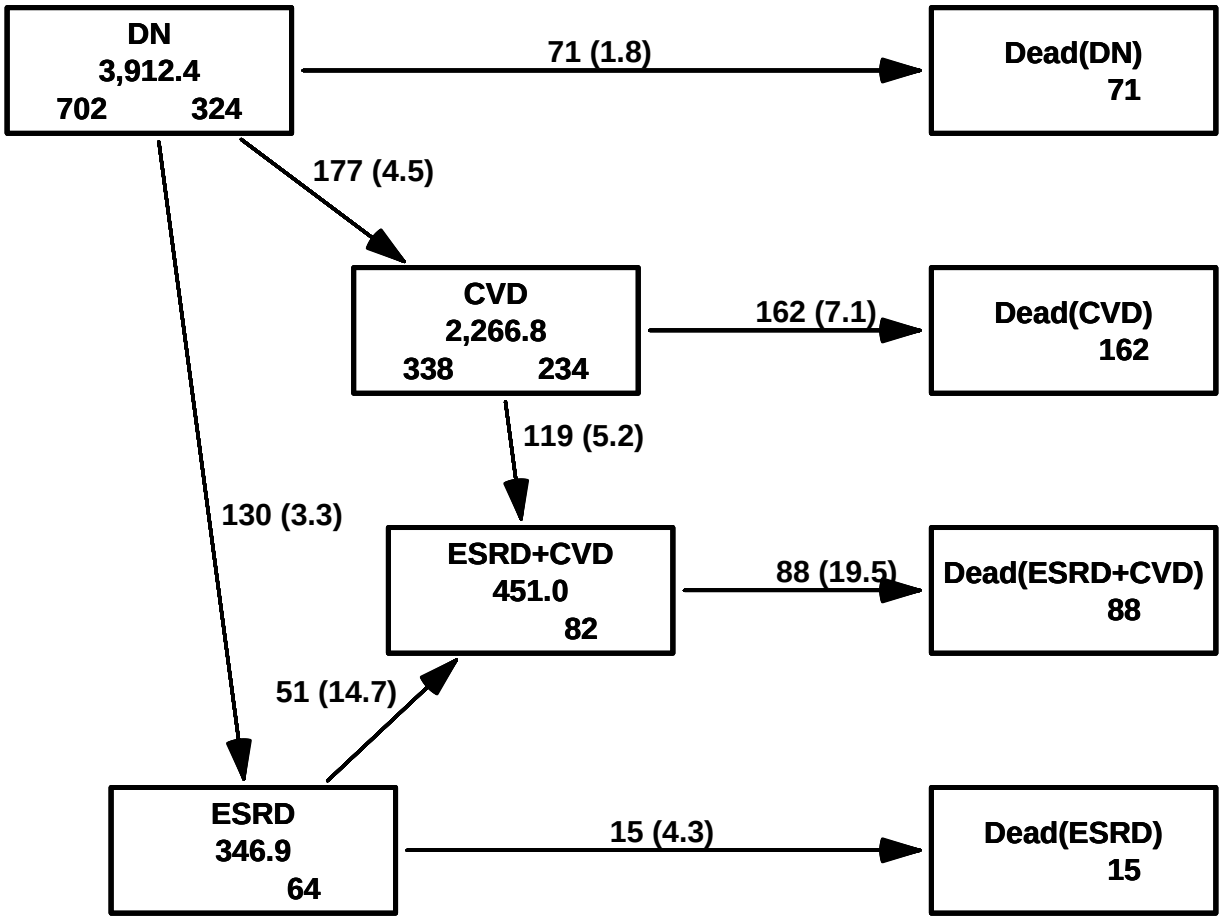

Figure 1.4: *Total number of patients in different states. Numbers in boxes are person-years and persons starting resp. ending their follow-up in each box. Numbers on arrows are number of transitions and crude rates in % per year.*

```
> options( digits=5, width=97 )
> subset( L7, lex.id %in% who[c(1,4)] )[,1:11]
```

```
         age    per     tfi     tfn    dur tfCVD tfESRD lex.dur  lex.Cst        lex.Xst lex.id
19    54.064 2000.3 0.00000  6.0000 23.000    NA     NA 0.99932       DN           ESRD     19
192   55.064 2001.3 0.99932  6.9993 23.999    NA 0.0000 1.97673     ESRD       ESRD+CVD     19
1921 57.040 2003.3 2.97604  8.9760 25.976    NA 1.9767 5.84805 ESRD+CVD Dead(ESRD+CVD)     19
51    49.969 2000.7 0.00000 14.0000 32.000    NA     NA 3.14031       DN           ESRD     51
511   53.109 2003.9 3.14031 17.1403 35.140    NA 0.0000 5.23477     ESRD       ESRD+CVD     51
5111 58.344 2009.1 8.37509 22.3751 40.375    NA 5.2348 1.90554 ESRD+CVD       ESRD+CVD     51
```

```
> options( digits=7, width=90 )
```

1.2.2 Illustrating follow-up in Lexis diagrams

To create an overview of the follow-up time in the various states we make a Lexis-diagram separately for T1 and T2 patients:

```
> ypi <- 7
> yl <- c(15,90)
> xl <- c(1998,2013)
> # Colors for all subsequent plots with the states
> # clx is a version with 4 "paled" version of the colors.
> clr <- c("limegreen","blue","orange","red")
> clx <- c( clr, rgb( t(col2rgb(clr))/3+255/3*2, max=255 ) )
> ### Transprify them a bit, does not work on .emf
> ### clx <- c( clr, rev( rgb(t(col2rgb(clr)),alpha=140,max=255) ) )
>
> pdf( "./graph/DN-Lexis-1-2.pdf",
+      height=1+diff(yl)/ypi,
+      width=2+2*diff(xl)/ypi )
> par( mfrow=c(1,2), mai=c(3,3,1,1)/4, omi=c(0,0,0,0),
+      mgp=c(3,1,0)/1.6, las=1 )
> for( tp in levels(L7$dm.type)[1:2] )
+    {
+ LL <- subset(L7,dm.type==tp)
+ plsymb <- c(NA,16)[1+(substr(LL$lex.Xst,1,4)=="Dead")]
+ plSymb <- c(NA,1 )[1+(substr(LL$lex.Xst,1,4)=="Dead")]
+ plot( LL,
+       time.scale=c("per","age"), xlab="Date of FU", ylab="Age",
+       col=clr[LL$lex.Cst],
+       xaxs="i", yaxs="i", xaxt="n", yaxt="n", xlim=xl, ylim=yl,
+       grid=seq(10,90,5), lty.grid=1 )
+ axis( side=1, at=1990+0:5*5, labels=rep("",6) )
+ axis( side=1, at=1990+0:5*10 )
+ axis( side=2, at=0:20*5, labels=rep("",21) )
+ axis( side=2, at=0:20*10 )
+ points( LL, pch=plsymb, cex=0.7, col=clr[LL$lex.Cst] )
+ points( LL, pch=plSymb, cex=0.7, col="white", lwd=2 )
+     }
> dev.off()
  null device
            1
```

Finally we save the original and the expanded `Lexis` objects `L1` and `L5` to be used for separate analyses by type of diabetes. We shall also throughout need the colors for the states:

```
> save( L1, L7,  clr, clx, file="./data/Base-Lexis.Rda" )
```

The reason for saving the `L1` object is, that in this each person is represented by one record, and it is thus suitable for baseline tabulations of covariates.

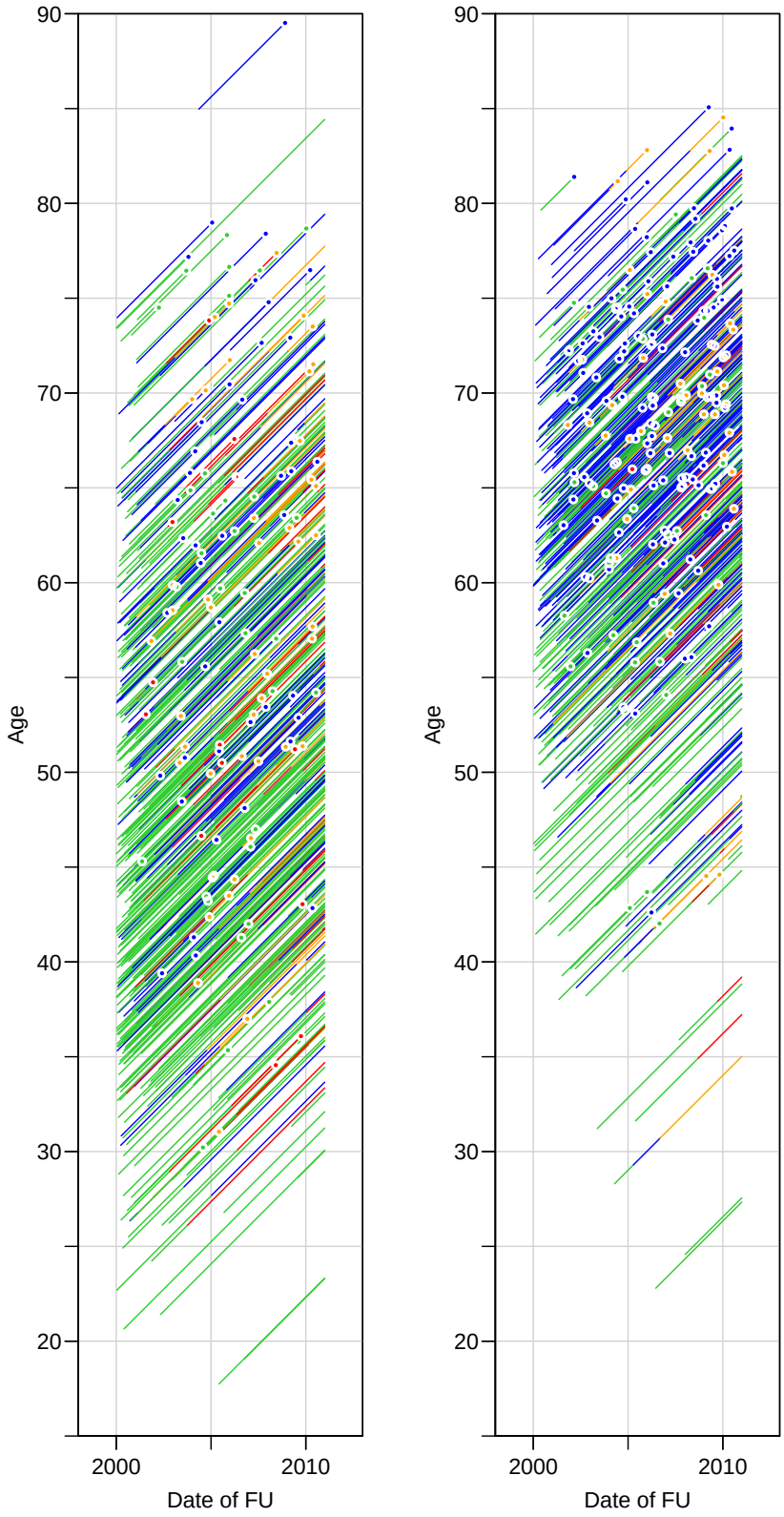

Figure 1.5: *Lexis diagrams of follow-up for T1 (left) and T2 (right) patients. DN state is green, CVD blue, ESRD after CVD purple and ESRD without CVD red. Dots indicates deaths.*

```
 R 3.0.2
 ------------------------------------------
 Program:  DNx.rnw
  Folder:  c:\Bendix\Steno\GbAd
 Started:  tirsdag 24. december 2013, 12:18:51
 ------------------------------------------
Writing to file DNx.tex
Processing code chunks with options ...
 1 : keep.source term hide (DNprep.rnw:12)
 2 : echo keep.source term verbatim (label = intro1, DNprep.rnw:16)
 3 : echo keep.source term verbatim (label = intro2, DNprep.rnw:27)
 4 : echo keep.source term verbatim (label = intro3, DNprep.rnw:35)
 5 : echo keep.source term verbatim (DNprep.rnw:41)
 6 : echo keep.source term verbatim (label = summary, DNprep.rnw:48)
 7 : echo keep.source term verbatim (label = saving, DNprep.rnw:56)
 8 : echo keep.source term verbatim eps pdf  (label = cvd-in, DNprep.rnw:64)
 9 : echo keep.source term verbatim (DNprep.rnw:86)
10 : echo keep.source term verbatim (DNprep.rnw:101)
11 : echo keep.source term verbatim (label = intersects, DNprep.rnw:122)
12 : echo keep.source term verbatim (DNprep.rnw:136)
13 : echo keep.source term verbatim eps pdf  (label = boxes, DNprep.rnw:172)
14 : echo keep.source term verbatim eps pdf  (label = boxes-p, DNprep.rnw:197)
      type       old       new
1  lex.Cst        DN        DN
2  lex.Cst      ESRD      ESRD
3  lex.Cst CVD(ESRD) CVD(ESRD)
4  lex.Cst       CVD       CVD
5  lex.Cst ESRD(CVD) ESRD(CVD)
6  lex.Cst     D-can
7  lex.Cst     D-CVD
8  lex.Cst     D-ext
9  lex.Cst     D-kid
10 lex.Cst     D-oth
11 lex.Cst     D-unk
12 lex.Xst        DN        DN
13 lex.Xst      ESRD      ESRD
14 lex.Xst CVD(ESRD) CVD(ESRD)
15 lex.Xst       CVD       CVD
16 lex.Xst ESRD(CVD) ESRD(CVD)
17 lex.Xst     D-can      Dead
18 lex.Xst     D-CVD      Dead
19 lex.Xst     D-ext      Dead
20 lex.Xst     D-kid      Dead
21 lex.Xst     D-oth      Dead
22 lex.Xst     D-unk      Dead
15 : echo keep.source term verbatim (DNprep.rnw:215)
16 : echo keep.source term verbatim (DNprep.rnw:227)
17 : echo keep.source term verbatim (DNprep.rnw:242)
18 : echo keep.source term verbatim eps pdf  (label = rev-box, DNprep.rnw:257)
19 : echo keep.source term verbatim (DNprep.rnw:274)
20 : echo keep.source term verbatim (label = Lexis, DNprep.rnw:286)
21 : echo keep.source term verbatim (DNprep.rnw:328)

You can now run (pdf)latex on 'DNx.tex'

 ------------------------------------------
 Program: DNx.rnw
  Folder: c:\Bendix\Steno\GbAd
   Ended: tirsdag 24. december 2013, 12:18:55
 Elapsed: 00:00:04
 ------------------------------------------
```