

Clinical nephropathy in Hong Kong (PWH) & Denmark (SDC)

SDC

April 2015

<http://bendixcarstensen.com/>

Version 3

Compiled Wednesday 22nd April, 2015, 11:07
from: /home/bendix/sdc/proj/HKPWH/Joint.tex

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

Guozhi Jiang Prince of Wales Hospital, CUHK, Hong Kong
b122920@mailserv.cuhk.edu.hk
Dorte Vistisen Steno Diabetes Center, Gentofte, Denmark
dtvs@steno.dk

Contents

1	Analysis 3 classes - HK	1
1.1	Description of data	1
1.1.1	Data overview	1
1.1.2	Outcomes	3
1.2	Analysis	7
2	Analysis 4 classes - HK	12
2.1	Description of data	12
2.1.1	Data overview	12
2.1.2	Outcomes	14
2.2	Analysis	17
3	Reading data - DK	23
3.1	Reading the SDC clinical data	23
3.1.1	Measurement unit variables	30
3.1.2	Renal endpoints	33
3.2	Saving data	35
4	Descriptives - DK	36
4.1	Date variables	36
4.2	Data overview	36
5	Analysis - DK	42
5.1	Outcome data	42
5.2	Trajectory analyses with latent classes	43
5.2.1	2 do next	48

Chapter 1

Analysis 3 classes - HK

1.1 Description of data

1.1.1 Data overview

The data were comprised by two parts: the baseline data and the follow-up eGFR data. The baseline data, which were extracted from the Hong Kong Diabetes Registry(HKDR), included the information of clinical assessments and laboratory investigations at enrollment, and the well-defined complication outcomes censored to 31st, January 2009. As here we focused on the ESRD outcome, we only selected those Chinese patients with no history of ESRD which was defined according to the ICD-9 codes and eGFR <15. Therefore, we obtained a cohort consisted of 718 ESRD events and 8810 event-free patients at censoring date. The follow-up data included all the creatinine records from enrollment to 2014, and the eGFR were corresponding calculated using the Chinese-modified MDRD formula.

```
> base_dat <- read.table("../data/ESRD1_Prosp2014_CH-T2D_1218vs8336.csv",header=TRUE,sep=",");
> follow_dat <- read.table("../data/eGFR_19940714_20140630.csv", header=TRUE, sep=",");
> base_dat <- transform(base_dat, doin=cal.yr(date), dob=cal.yr(DOB, "%d/%m/%Y"),
+                        dox=cal.yr(ESRD1_DATE));
> base_dat <- transform(base_dat, dodm = pmin(YEAR_DIA + runif(length(YEAR_DIA)), doin));
> dob_na <- which(is.na(base_dat$dob));
> base_dat$dob[dob_na] <- (floor(base_dat$doin[dob_na]) - base_dat$AGE[dob_na]) + runif(length(dob_na));
> #write.table(base_dat, "trans_base.csv", sep=",", row.names=F, col.names=T);
> base_subdat <- subset(base_dat, select=c("Obs_id", "doin", "dob", "dodm", "dox",
+                                         "SEX", "ESRD1_END"))
> dim(base_subdat);
[1] 9554    7
> names(base_subdat);
[1] "Obs_id"    "doin"      "dob"       "dodm"      "dox"       "SEX"
[7] "ESRD1_END"
> head(base_subdat);
  Obs_id    doin      dob      dodm      dox SEX ESRD1_END
1      1 2002.805 1940.598 2002.493 2014.493  0         0
2      2 1996.757 1939.194 1995.153 2014.493  1         0
3      3 1996.585 1935.172 1983.217 2014.493  1         0
4      4 2001.203 1927.048 1980.747 2004.648  1         0
5      5 1997.381 1924.527 1993.908 2014.493  1         0
6      6 1999.221 1922.345 1991.221 2010.564  0         0
> table(base_subdat$ESRD1_END);
 0     1
8336 1218
```

The outcomes of interest were named as "ESRD1_HIST" (0 represents no ESRD history), "ESRD1_END"(endpoint censored to 2009) and "ESRD1_TIME"(follow-up period).

```
> follow_dat <- transform(follow_dat, dolab=cal.yr(test_date));
> follow_dat <- subset(follow_dat, select=-c(test_date));
> dim(follow_dat);
[1] 391551      4
> head(follow_dat);
  Obs_id F_eGFR0 creatinine   dolab
1      1  80.6474         84 2002.632
2      1  97.9131         71 2002.643
3      1  73.5245         91 2002.663
4      1  91.8750         75 2002.767
5      1  72.5693         92 2002.805
6      1  81.4311         83 2003.914
```

We merged the baseline data and the follow-up eGFR data according to the id of subject:

```
> merged_dat <- merge(base_subdat, follow_dat, by=intersect("Obs_id", "Obs_id"), sort=F);
> dim(merged_dat);
[1] 366122     10
> head(merged_dat);
  Obs_id   doin      dob      dodm      dox SEX ESRD1_END F_eGFR0 creatinine
1      1 2002.805 1940.598 2002.493 2014.493  0         0 102.8468         66
2      1 2002.805 1940.598 2002.493 2014.493  0         0  91.8750         75
3      1 2002.805 1940.598 2002.493 2014.493  0         0  96.8863         71
4      1 2002.805 1940.598 2002.493 2014.493  0         0  80.6474         84
5      1 2002.805 1940.598 2002.493 2014.493  0         0  87.7935         77
6      1 2002.805 1940.598 2002.493 2014.493  0         0 103.2903         66
  dolab
1 2014.359
2 2002.767
3 2005.951
4 2002.632
5 2007.558
6 2012.812
> addmargins(table(table(merged_dat$Obs_id)));
      1      2      3      4      5      6      7      8      9     10     11     12     13     14     15     16
350   65   69   74   94  141  182  179  158  174  160  150  155  172  158  176
  17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
172  222  214  200  220  160  174  196  178  164  177  159  159  164  153  119
  33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48
130  129  132  125  112  105  117  126  123  118  113  99   88  122   76   93
  49   50   51   52   53   54   55   56   57   58   59   60   61   62   63   64
  93   98   73   75   69   62   54   56   68   59   62   55   54   48   43   34
  65   66   67   68   69   70   71   72   73   74   75   76   77   78   79   80
  32   38   36   43   38   36   27   34   26   29   40   30   28   28   30   21
  81   82   83   84   85   86   87   88   89   90   91   92   93   94   95   96
  25   22   22   23   25   29   23   23   25   24   19   16   21   16   17   14
  97   98   99  100  101  102  103  104  105  106  107  108  109  110  111  112
  10   18   22   12   14   10   12   17   11   13   16   13   12   10   8   12
113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128
   9    8   13    8    8   10   11   12   10    6    7    4    5    5    4    5
129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144
   6   10    6    8    2    9    9    1    9    5    6    3    2    2    4    3
145  146  147  148  149  150  151  153  154  155  156  158  159  160  161  162
   1    2    7    4    7    7    4    2    5    4    2    7    4    2    2    1
163  164  165  166  167  169  170  171  172  173  175  176  177  178  179  181
   1    5    2    1    4    2    3    2    1    6    3    4    3    4    4    1
182  183  186  187  191  193  194  195  196  197  198  199  200  204  206  207
   2    2    1    3    2    1    1    3    1    2    1    2    2    1    1    1
```

208	209	210	211	212	213	216	218	221	223	224	225	228	231	232	238
1	1	1	1	1	1	2	1	1	1	1	1	2	1	1	1
240	241	243	249	250	251	252	254	258	267	282	284	288	293	301	308
1	1	1	1	1	2	2	1	1	3	1	1	1	1	1	1
319	353	431	725	Sum											
1	1	1	1	9554											

We only selected those records between baseline and event/censoring dates, that said, those eGFR records before baseline or after event/censoring dates were removed. Moreover, we also calculated the follow-up age, duration of diabetes, and the backward time gap between event/censoring date and measurement date of eGFR, named "F_AGE", "F_DMAGE" and "BW_TIME", respectively.

```
> sub_merged <- subset(merged_dat, (1:nrow(merged_dat)) %in% intersect(which(dolab >= doin), which(dolab <= dodm)))
> sub_merged <- transform(sub_merged, F_AGE=dolab-dob, F_DMAGE=dolab-dodm, BW_TIME=dolab-dox);
> dim(sub_merged);
[1] 282607      13
> head(sub_merged);
  Obs_id   doin   dob   dodm   dox SEX ESRD1_END F_eGFR0 creatinine
1      1 2002.805 1940.598 2002.493 2014.493 0         0 102.8468         66
3      1 2002.805 1940.598 2002.493 2014.493 0         0  96.8863         71
5      1 2002.805 1940.598 2002.493 2014.493 0         0  87.7935         77
6      1 2002.805 1940.598 2002.493 2014.493 0         0 103.2903         66
7      1 2002.805 1940.598 2002.493 2014.493 0         0  81.2613         83
8      1 2002.805 1940.598 2002.493 2014.493 0         0  81.4311         83
  dolab F_AGE F_DMAGE BW_TIME
1 2014.359 73.76044 11.865461 -0.1341547
3 2005.951 65.35250  3.457521 -8.5420945
5 2007.558 66.95962  5.064640 -6.9349760
6 2012.812 72.21355 10.318575 -1.6810404
7 2004.568 63.96988  2.074906 -9.9247091
8 2003.914 63.31554  1.420560 -10.5790554
> str(sub_merged);
'data.frame':      282607 obs. of  13 variables:
 $ Obs_id      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ doin        : num  2003 2003 2003 2003 2003 ...
 $ dob         : num  1941 1941 1941 1941 1941 ...
 $ dodm        : num  2002 2002 2002 2002 2002 ...
 $ dox         : num  2014 2014 2014 2014 2014 ...
 $ SEX         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ESRD1_END   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ F_eGFR0     : num  102.8 96.9 87.8 103.3 81.3 ...
 $ creatinine  : num  66 71 77 66 83 83 92 74 73 69 ...
 $ dolab       : num  2014 2006 2008 2013 2005 ...
 $ F_AGE       : num  73.8 65.4 67 72.2 64 ...
 $ F_DMAGE     : num  11.87 3.46 5.06 10.32 2.07 ...
 $ BW_TIME     : num  -0.134 -8.542 -6.935 -1.681 -9.925 ...
> range(sub_merged$BW_TIME);
[1] -21.03491    0.00000
> #write.table(sub_merged, "effective_subdat.csv", sep="," , row.names=F, col.names=T)
```

1.1.2 Outcomes

We removed those records with missing data. We first plotted the observed creatinine and eGFR values (Figure ??). From the figure, we can see there are some abnormal records, which may be due to measurement or typo errors.

```

> nomiss_dat <- sub_merged[complete.cases(sub_merged), ];
> dim(nomiss_dat);
[1] 279002      13
> with(nomiss_dat, table((F_eGFR0>300) + (F_eGFR0>1000)));
      0      1      2
278701  265   36

> with(nomiss_dat, plot(dolab, F_eGFR0, pch=16, cex=0.3,
+                       xlab="Date of measurement", ylab="eGFR"));

> with(nomiss_dat, plot(dolab, creatinine, pch=16, cex=0.3,
+                       xlab="Date of measurement", ylab="Creatinine"));

```

We removed those records with eGFR ≥ 300 , which were considered to be errors. The updated distributions were shown in Figure 2.1.

```

> sub_nomiss <- subset(nomiss_dat, F_eGFR0<300);
> dim(sub_nomiss);
[1] 278701      13

> with(sub_nomiss[sample(1:nrow(sub_nomiss), 2000),],
+       plot(BW_TIME, F_eGFR0, pch=16, cex=0.3,
+            xlab="Time before ESRD", ylab="eGFR"));
> abline(h=15, col="red");

> with(sub_nomiss[sample(1:nrow(sub_nomiss), 2000),],
+       plot(BW_TIME, creatinine, pch=16, cex=0.3, #ylim=c(0,400),
+            xlab="Time before ESRD", ylab="Creatinine"))

```

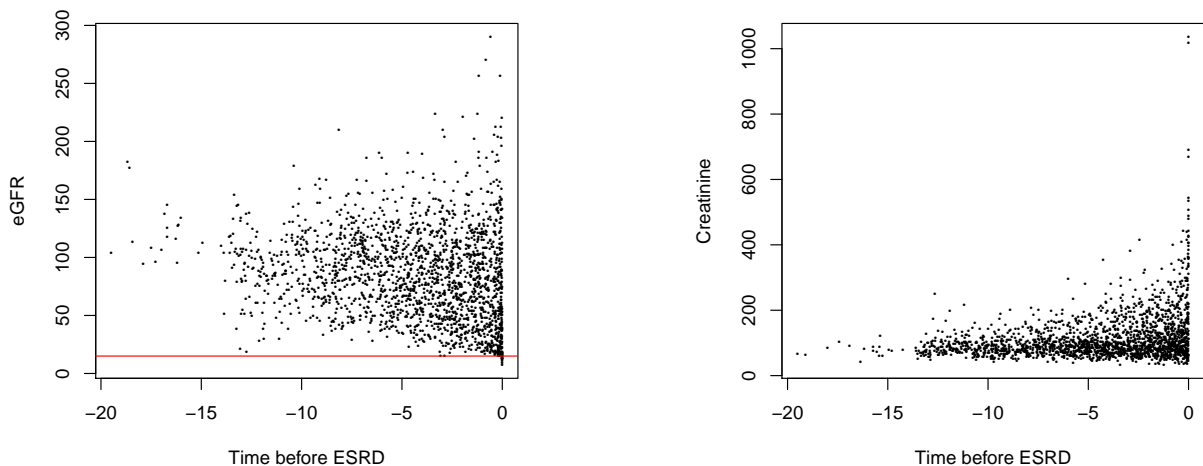


Figure 1.1: *Distribution of eGFR and creatinine with 2000 samples after removing eGFR ≥ 300 . The red line represents eGFR=15.*

As we here only focused on the patients developed ESRD, we extracted those subjects and plotted the distribution of eGFR. We removed those patients with only one measurement, and built the models on event and non-event individually. The summaries of the number of measurement for each subject were shown in Figure 2.2.

```

> num_test <- table(sub_nomiss$Obs_id);
> id_keep <- names(which(num_test>1)); #removed those patients with only one
> sub_nomiss <- subset(sub_nomiss, Obs_id %in% id_keep);
> sub_nomiss <- transform(sub_nomiss, log_eGFR=log(sub_nomiss$F_eGFR));
> event_dat <- subset(sub_nomiss, ESRD1_END==1); #event subjects
> dim(event_dat);

[1] 41412 14

> length(unique(event_dat$Obs_id));

[1] 1167

> (num_event <- table(table(event_dat$Obs_id)));

 2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
51 21 23 18 17 19 23 18 21 21 22 17 21 23 20 18 18 23 14 16
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
23 17 20 18 15 17 17 32 17 13 11 24 18 13 23 14 20 24 15 16
42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61
14 16 10 14 11 11 10 10 8 6 11 10 7 8 7 7 7 7 7 8
62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81
10 11 4 8 10 5 5 8 4 2 4 2 3 3 4 8 2 3 3 1
82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 98 99 100 102 103
3 4 5 5 1 3 3 2 1 4 4 2 3 1 2 1 2 1 1 1
104 105 106 108 109 111 112 113 114 115 117 120 122 125 126 128 129 131 132 133
4 1 2 1 1 2 1 2 1 3 4 1 1 1 1 1 1 1 1 1
136 142 146 147 148 153 173
1 1 1 1 1 1 1

> nevent_dat <- subset(sub_nomiss, ESRD1_END==0); #non-event subjects
> dim(nevent_dat);

[1] 236907 14

> length(unique(nevent_dat$Obs_id));

[1] 7852

> (num_nevent <- table(table(nevent_dat$Obs_id)));

 2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
77 70 81 119 175 214 205 181 213 172 193 183 175 186 210 203 219 193 219 200
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
173 169 164 186 146 162 169 128 141 119 112 126 101 109 95 91 77 94 86 100
42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61
86 90 88 84 79 57 62 60 60 53 54 57 37 34 38 42 40 38 38 33
62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81
42 27 20 22 24 22 19 19 22 22 18 19 13 16 19 13 8 8 19
82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 101 102
10 10 9 6 14 11 10 10 6 7 8 5 12 8 7 7 9 5 2 4
103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 123
5 4 2 3 2 10 8 2 2 6 5 1 1 2 2 2 6 4 1 4
124 126 127 128 129 130 131 132 134 137 139 142 143 144 145 147 148 151 155 156
2 4 2 3 1 1 2 1 3 2 1 3 1 1 1 3 1 2 1 1
157 158 159 161 163 166 167 168 176 183 185 187 190 194 199 203 207 220 221 288
1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1
295 621
1 1

> par(mar=c(5,4,1,2));
> plot(as.numeric(names(num_event)), num_event, type="h", lwd=3, xaxs="i", xlim=c(0,140),
+ xlab="No. of measurements", ylab="No. of subjects", main="ESRD subjects", yaxt="n");
> axis(side=2);

> par(mar=c(5,4,1,2));
> plot(as.numeric(names(num_nevent)), num_nevent, type="h", lwd=3, xaxs="i", xlim=c(0,140),
+ xlab="No. of measurements", ylab="No. of subjects", main="non-ESRD subjects", yaxt="n");
> axis(side=2);

```

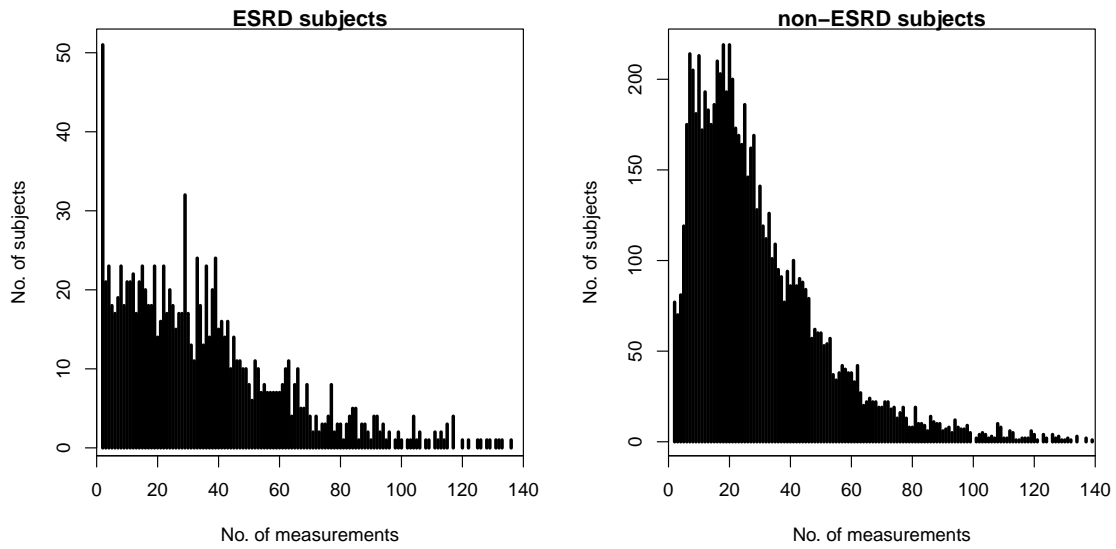


Figure 1.2: Number of eGFR records for event and non-event subjects.

```
> dest_eGFR <- density(event_dat$F_eGFR0);
> plot(dest_eGFR, xlim=c(0,150), xlab="eGFR", lwd=3, yaxs="i",
+      ylab="Density", main="ESRD subjects", bty="n");
> abline(v=quantile(event_dat$F_eGFR0, probs=c(50, 75, 90)/100),
+      col="red");

> dest_logeGFR <- density(log(event_dat$F_eGFR0));
> plot(dest_logeGFR, lwd=3, xlab="Ln(eGFR)",
+      ylab="Density", main="ESRD subjects");
```

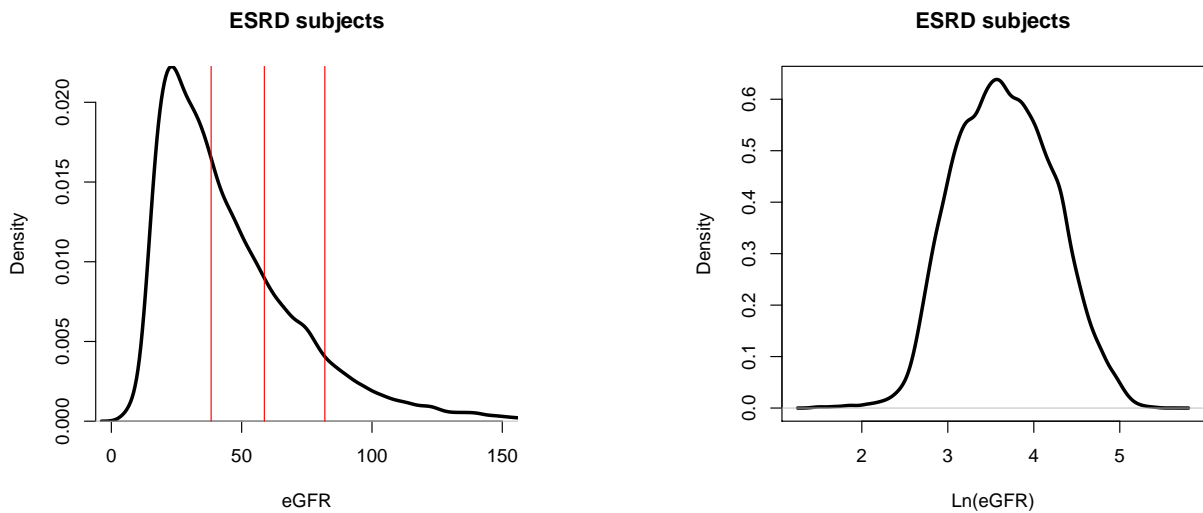


Figure 1.3: Distribution of eGFR and $\log(eGFR)$ for ESRD subjects. Red lines represents the 50, 75 and 90 percent of subjects, respectively

As shown in Figure 2.3, the original distribution of eGFR is skewed, whilst it's close to be normal after log-transformed.

1.2 Analysis

We first fit the model using the "lcm" function, and specified "linear" as the link function. The "lcm" with linear link function is similar with the "hlme" function, but the reason for using "lcm" function is that the confidence interval of predict values can be obtained by using the corresponding prediction function.

```
> (kn <- seq(-12, 0, , 5));
[1] -12 -9 -6 -3 0
> MM <- Ns(event_dat$BW_TIME, knots=kn);
> dim(MM);
[1] 41412 4
> MM <- detrend(MM, event_dat$BW_TIME);
> dim(MM);
[1] 41412 3
> head(MM);
      1      2      3
[1,] 0.04689555 0.38709327 -0.10183513
[2,] -0.02867914 0.38150137 -0.07236472
[3,] -0.06988576 0.30174461 -0.04175845
[4,] -0.05881167 -0.13637325 0.03674734
[5,] -0.07504976 0.02845691 0.01176823
[6,] -0.34770860 -0.36093378 0.33529565
> (colnames(MM) <- paste("x", colnames(MM), sep=""));
[1] "x1" "x2" "x3"
> event_dat <- cbind(event_dat, MM);
> dim(event_dat);
[1] 41412 17

> event_model <- hlme(log_eGFR ~ BW_TIME + x1 + x2 + x3 + F_AGE + SEX + F_DMAGE,
+                        mixture =~ BW_TIME + x1 + x2 + x3,
+                        random =~ BW_TIME,
+                        subject = "Obs_id", ng=3, data=event_dat);
Be patient, hlme is running ...
The program took 5497.56 seconds
> event_model;
Heterogenous linear mixed model
fitted by maximum likelihood method

hlme(fixed = log_eGFR ~ BW_TIME + x1 + x2 + x3 + F_AGE + SEX +
      F_DMAGE, mixture = ~BW_TIME + x1 + x2 + x3, random = ~BW_TIME,
      subject = "Obs_id", ng = 3, data = event_dat)

Statistical Model:
  Dataset: event_dat
  Number of subjects: 1167
  Number of observations: 41412
  Number of latent classes: 3
  Number of parameters: 24

Iteration process:
  Convergence criteria satisfied
  Number of iterations: 16
  Convergence criteria: parameters= 3e-09
                      : likelihood= 1.8e-07
                      : second derivatives= 4.9e-14

Goodness-of-fit statistics:
  maximum log-likelihood: -4507.4
  AIC: 9062.8
  BIC: 9184.29
```

```

> postprob(event_model);
Posterior classification:
  class1 class2 class3
N 303.00 593.00 271.00
% 25.96 50.81 23.22

Posterior classification table:
--> mean of posterior probabilities in each class
  prob1 prob2 prob3
class1 0.8997 0.0996 0.0007
class2 0.0912 0.8463 0.0625
class3 0.0086 0.1246 0.8669

Posterior probabilities above a threshold (%):
  class1 class2 class3
prob>0.7 85.15 76.05 80.44
prob>0.8 78.55 65.60 70.85
prob>0.9 68.32 55.99 60.15
> str(event_model$pprob);
'data.frame':   1167 obs. of  5 variables:
 $ Obs_id: num  16 21 22 36 37 42 44 54 56 71 ...
 $ class : int   2 1 3 1 2 1 1 2 3 2 ...
 $ prob1 : num  5.33e-03 9.96e-01 2.51e-08 9.35e-01 1.71e-03 ...
 $ prob2 : num  0.942052 0.003643 0.000482 0.065047 0.998282 ...
 $ prob3 : num  5.26e-02 7.25e-31 1.00 5.41e-11 3.14e-06 ...
> ng <- event_model$ng;

> #nevent_model <- hlme(log_eGFR ~ BW_TIME + x1 + x2 + x3 + F_AGE + SEX + F_DMAGE,
> #                      random =~ BW_TIME,
> #                      subject = "Obs_id", ng=1, data=nevent_dat);
> #nevent_model;

```

Now we can further investigate the posterior probabilities of subjects in each class (Figure 2.4 and 2.5).

```

> clr = c("black", "red", "blue")
> par(mfrow=c(1,3), mar=c(3,0,1,1), oma=c(0,3,0,0),
+     las=1, bty="n", mgp=c(3,1,0)/1.6);
> ppr <- event_model$pprob;
> num <- table(ppr$class);
> for(i in 1:ng)
+ {
+   hist(ppr[class==i,i+2], breaks=0:50/50,
+       col=clr[i], border=clr[i], ylim=c(0,60),
+       main="", xlab="", yaxt="n", yaxs="i");
+   if(i==1) axis(side=2);
+   text(0.4,30,paste("Class",i," n=",num[i]),
+       font=2,col=clr[i], cex=1);
+ }
> #plot_ppr
> #plot_ppr(event_model,clr);

> ng <- event_model$ng;
> post_pr <- event_model$pprob;
> #par(bty="o");
> pairs(post_pr[, 2+1:ng], pch=16, col=clr[post_pr$class],
+       cex=0.4, gap=0);

```

We built a data set for prediction to plot the estimated trajectories. Here the median age 65, median DM duration 12 and sex as male were used. The function "plot_predictY" incorporated the prediction function "predictY" in LCMM package (Figure 2.6).

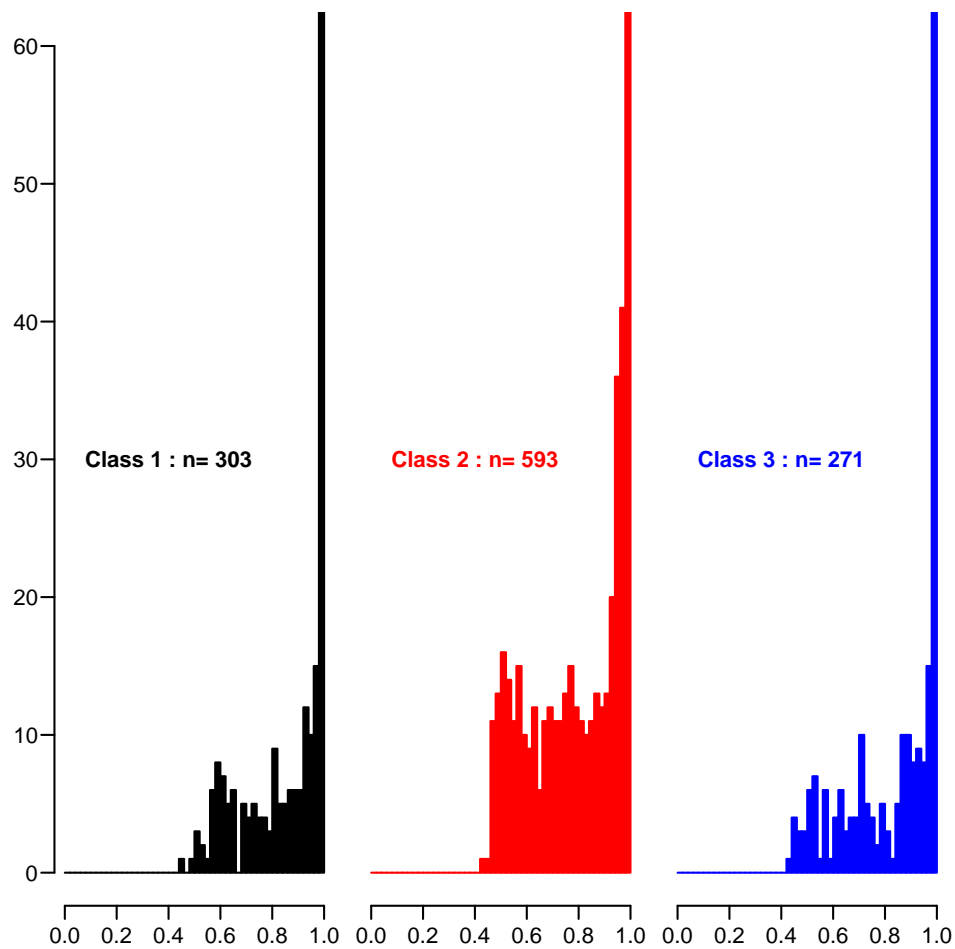


Figure 1.4: *Posterior probabilities of 3 classes for the ESRD model.*

```
> (adjust_var <- event_model$Xnames[-c(1:5)]);
[1] "F_AGE" "SEX" "F_DMAGE"
> length(table(event_dat$BW_TIME))
[1] 6964
> x <- sort(unique(event_dat$BW_TIME));
> length(x);
[1] 6964
> wh <- match(x, event_dat$BW_TIME)[1:69*100];
> length(wh);
[1] 69
> plotdata <- data.frame(1, event_dat$BW_TIME[wh], MM[wh,],
+                        F_AGE = 65 + event_dat$BW_TIME[wh],
+                        SEX = 1,
+                        F_DMAGE = 15 + event_dat$BW_TIME[wh]);
> names(plotdata) <- event_model$Xnames;
> head(plotdata);
```

	intercept	BW_TIME	x1	x2	x3	F_AGE	SEX
1	1	-14.22040	-0.6919963	-0.001011389	-0.621725804	50.77960	1
2	1	-12.62149	-0.6178121	-0.138170718	-0.250384157	52.37851	1
3	1	-11.96715	-0.5874523	-0.194302593	-0.098414192	53.03285	1

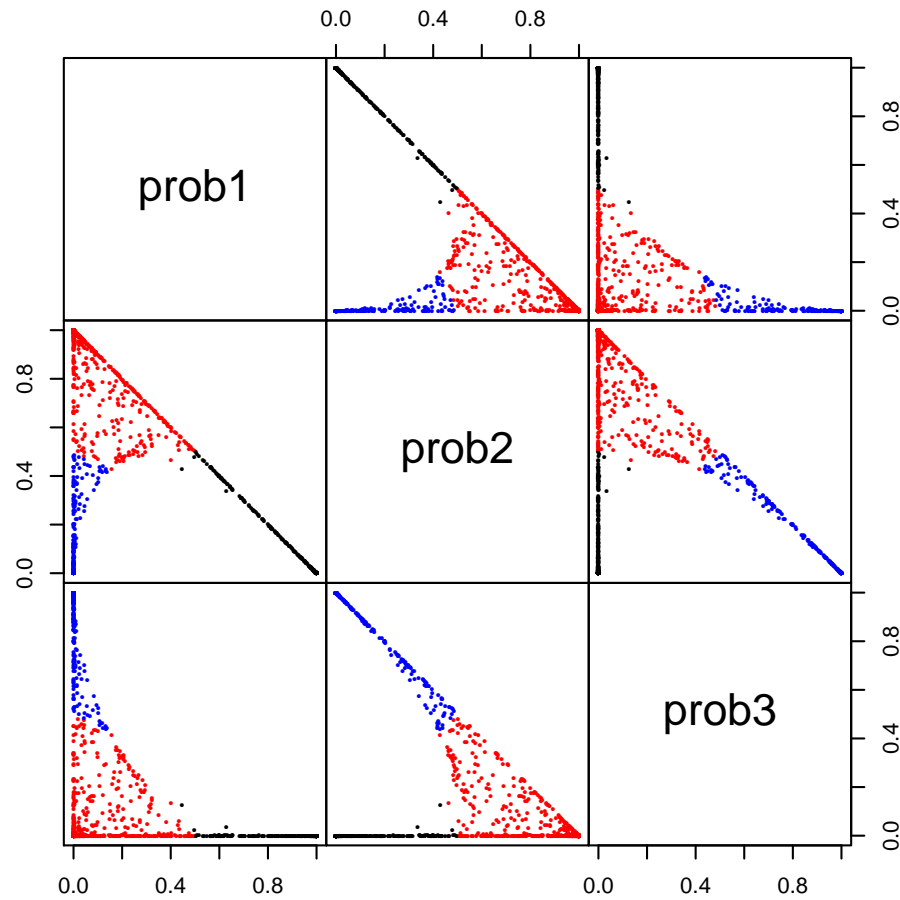


Figure 1.5: Pairwise posterior probabilities from the fitted model using "lcm".

```

4      1 -11.56468 -0.5682702 -0.228569177 -0.005717254 53.43532 1
5      1 -11.20602 -0.5490492 -0.258027568 0.073654734 53.79398 1
6      1 -10.87201 -0.5277820 -0.283754898 0.142452077 54.12799 1
      F_DMAGE
1 0.779603
2 2.378508
3 3.032854
4 3.435318
5 3.793977
6 4.127995

> range(plotdata$BW_TIME);
[1] -14.22039699 -0.08761123

> range(plotdata$F_DMAGE);
[1] 0.779603 14.912389

> pred_event <- exp(predictY(event_model, plotdata, var.time="BW_TIME", draws=TRUE)$pred);
> ylim <- range(pred_event);
> lwd_main <- 4;
> lwd_ci <- 1;
> for (i in 1:ng)
+ {
+   plot(y = pred_event[, i], x = plotdata$BW_TIME, type = "l", col= clrs[i],
```

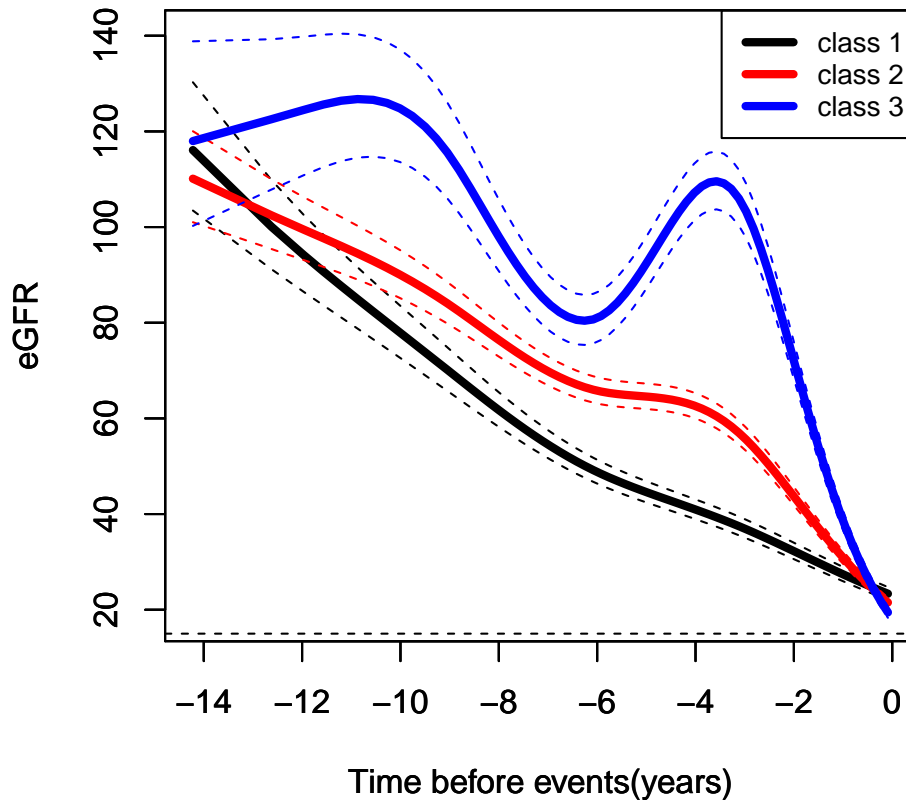


Figure 1.6: Mean trajectories of eGFR for the three latent classes of subjects developing ESRD. The horizontal dashed line represents eGFR is equal to 15

```
+      ylim = ylim, lwd = lwd_main, xlab = "Time before events(years)", ylab = "eGFR");
+      points(y = pred_event[, i + ng], x = plotdata$BW_TIME, type = "l", lty = "dashed", col = clrsl
+      ylim = ylim, lwd = lwd_ci);
+      points(y = pred_event[, i + 2*ng], x = plotdata$BW_TIME, type = "l", lty = "dashed", col= clrsl
+      ylim = ylim, lwd = lwd_ci);
+      par(new = TRUE);
+ }
> abline(h = 15, lty = "dashed");
> legend("topright", legend = paste("class", 1:ng), col=clrs, lty=1, lwd=lwd_main, cex=0.8);
> par(new = FALSE);
```

Chapter 2

Analysis 4 classes - HK

2.1 Description of data

2.1.1 Data overview

The data were comprised by two parts: the baseline data and the follow-up eGFR data. The baseline data, which were extracted from the Hong Kong Diabetes Registry(HKDR), included the information of clinical assessments and laboratory investigations at enrollment, and the well-defined complication outcomes censored to 31st, January 2009. As here we focused on the ESRD outcome, we only selected those Chinese patients with no history of ESRD which was defined according to the ICD-9 codes and eGFR <15. Therefore, we obtained a cohort consisted of 718 ESRD events and 8810 event-free patients at censoring date. The follow-up data included all the creatinine records from enrollment to 2014, and the eGFR were corresponding calculated using the Chinese-modified MDRD formula.

```
> base_dat <- read.table("../data/ESRD1_Prospect2014_CH-T2D_1218vs8336.csv",header=TRUE,sep=",");
> base_dat <- transform(base_dat, doin=cal.yr(date), dob=cal.yr(DOB, "%d/%m/%Y"),
+                        dox=cal.yr(ESRD1_DATE));
> base_dat <- transform(base_dat, dodm = pmin(YEAR_DIA + runif(length(YEAR_DIA)), doin));
> dob_na <- which(is.na(base_dat$dob));
> base_dat$dob[dob_na] <- (floor(base_dat$doin[dob_na]) - base_dat$AGE[dob_na]) + runif(length(dob_na));
> #write.table(base_dat, "trans_base.csv", sep=",", row.names=F, col.names=T);
> base_subdat <- subset(base_dat, select=c("Obs_id", "doin", "dob", "dodm", "dox",
+                                         "SEX", "ESRD1_END"))
> dim(base_subdat);
[1] 9554      7
> names(base_subdat);
[1] "Obs_id"      "doin"        "dob"         "dodm"        "dox"         "SEX"
[7] "ESRD1_END"
> head(base_subdat);
  Obs_id   doin      dob      dodm      dox SEX ESRD1_END
1      1 2002.805 1940.598 2002.493 2014.493  0         0
2      2 1996.757 1939.194 1995.153 2014.493  1         0
3      3 1996.585 1935.172 1983.217 2014.493  1         0
4      4 2001.203 1927.048 1980.747 2004.648  1         0
5      5 1997.381 1924.527 1993.908 2014.493  1         0
6      6 1999.221 1922.345 1991.221 2010.564  0         0
> table(base_subdat$ESRD1_END);
  0      1
8336 1218
```

The outcomes of interest were named as "ESRD1_HIST" (0 represents no ESRD history), "ESRD1_END"(endpoint censored to 2009) and "ESRD1_TIME"(follow-up period).

```
> follow_dat <- read.table("../data/eGFR_19940714_20140630.csv", header=TRUE, sep=",");
> follow_dat <- transform(follow_dat, dolab=cal.yr(test_date));
> follow_dat <- subset(follow_dat, select=-c(test_date));
> dim(follow_dat);
```

```
[1] 391551      4
```

```
> head(follow_dat);
```

	Obs_id	F_eGFR0	creatinine	dolab
1	1	80.6474	84	2002.632
2	1	97.9131	71	2002.643
3	1	73.5245	91	2002.663
4	1	91.8750	75	2002.767
5	1	72.5693	92	2002.805
6	1	81.4311	83	2003.914

We merged the baseline data and the follow-up eGFR data according to the id of subject:

```
> merged_dat <- merge(base_subdat, follow_dat, by=intersect("Obs_id", "Obs_id"), sort=F);
> dim(merged_dat);
```

```
[1] 366122      10
```

```
> head(merged_dat);
```

	Obs_id	doin	dob	dodm	dox	SEX	ESRD1_END	F_eGFR0	creatinine
1	1	2002.805	1940.598	2002.493	2014.493	0	0	102.8468	66
2	1	2002.805	1940.598	2002.493	2014.493	0	0	91.8750	75
3	1	2002.805	1940.598	2002.493	2014.493	0	0	96.8863	71
4	1	2002.805	1940.598	2002.493	2014.493	0	0	80.6474	84
5	1	2002.805	1940.598	2002.493	2014.493	0	0	87.7935	77
6	1	2002.805	1940.598	2002.493	2014.493	0	0	103.2903	66

```
dolab
```

```
1 2014.359
2 2002.767
3 2005.951
4 2002.632
5 2007.558
6 2012.812
```

```
> addmargins(table(table(merged_dat$Obs_id)));
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
350	65	69	74	94	141	182	179	158	174	160	150	155	172	158	176
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
172	222	214	200	220	160	174	196	178	164	177	159	159	164	153	119
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
130	129	132	125	112	105	117	126	123	118	113	99	88	122	76	93
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
93	98	73	75	69	62	54	56	68	59	62	55	54	48	43	34
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
32	38	36	43	38	36	27	34	26	29	40	30	28	28	30	21
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
25	22	22	23	25	29	23	23	25	24	19	16	21	16	17	14
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
10	18	22	12	14	10	12	17	11	13	16	13	12	10	8	12
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
9	8	13	8	8	10	11	12	10	6	7	4	5	5	4	5
129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
6	10	6	8	2	9	9	1	9	5	6	3	2	2	4	3
145	146	147	148	149	150	151	153	154	155	156	158	159	160	161	162
1	2	7	4	7	7	4	2	5	4	2	7	4	2	2	1
163	164	165	166	167	169	170	171	172	173	175	176	177	178	179	181
1	5	2	1	4	2	3	2	1	6	3	4	3	4	4	1
182	183	186	187	191	193	194	195	196	197	198	199	200	204	206	207

2	2	1	3	2	1	1	3	1	2	1	2	2	1	1	1
208	209	210	211	212	213	216	218	221	223	224	225	228	231	232	238
1	1	1	1	1	1	2	1	1	1	1	1	2	1	1	1
240	241	243	249	250	251	252	254	258	267	282	284	288	293	301	308
1	1	1	1	1	2	2	1	1	3	1	1	1	1	1	1
319	353	431	725	Sum											
1	1	1	1	9554											

We only selected those records between baseline and event/censoring dates, that said, those eGFR records before baseline or after event/censoring dates were removed.

Moreover, we also calculated the follow-up age, duration of diabetes, and the backward time gap between event/censoring date and measurement date of eGFR, named

"F_AGE", "F_DMAGE" and "BW_TIME", respectively.

```
> sub_merged <- subset(merged_dat, (1:nrow(merged_dat)) %in% intersect(which(dolab >= doin), which(
> sub_merged <- transform(sub_merged, F_AGE=dolab-dob, F_DMAGE=dolab-dodm, BW_TIME=dolab-dox);
> dim(sub_merged);
[1] 282607      13
> head(sub_merged);
  Obs_id   doin    dob    dodm    dox SEX ESRD1_END F_eGFR0 creatinine
1      1 2002.805 1940.598 2002.493 2014.493  0         0 102.8468         66
3      1 2002.805 1940.598 2002.493 2014.493  0         0  96.8863         71
5      1 2002.805 1940.598 2002.493 2014.493  0         0  87.7935         77
6      1 2002.805 1940.598 2002.493 2014.493  0         0 103.2903         66
7      1 2002.805 1940.598 2002.493 2014.493  0         0  81.2613         83
8      1 2002.805 1940.598 2002.493 2014.493  0         0  81.4311         83
      dolab   F_AGE   F_DMAGE   BW_TIME
1 2014.359 73.76044 11.865461 -0.1341547
3 2005.951 65.35250  3.457521 -8.5420945
5 2007.558 66.95962  5.064640 -6.9349760
6 2012.812 72.21355 10.318575 -1.6810404
7 2004.568 63.96988  2.074906 -9.9247091
8 2003.914 63.31554  1.420560 -10.5790554
> str(sub_merged);
'data.frame':   282607 obs. of  13 variables:
 $ Obs_id      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ doin        : num  2003 2003 2003 2003 2003 ...
 $ dob         : num  1941 1941 1941 1941 1941 ...
 $ dodm        : num  2002 2002 2002 2002 2002 ...
 $ dox         : num  2014 2014 2014 2014 2014 ...
 $ SEX         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ESRD1_END   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ F_eGFR0     : num  102.8 96.9 87.8 103.3 81.3 ...
 $ creatinine  : num  66 71 77 66 83 83 92 74 73 69 ...
 $ dolab       : num  2014 2006 2008 2013 2005 ...
 $ F_AGE       : num  73.8 65.4 67 72.2 64 ...
 $ F_DMAGE     : num  11.87 3.46 5.06 10.32 2.07 ...
 $ BW_TIME     : num  -0.134 -8.542 -6.935 -1.681 -9.925 ...
> range(sub_merged$BW_TIME);
[1] -21.03491  0.00000
> #write.table(sub_merged, "effective_subdat.csv", sep=",", row.names=F, col.names=T)
```

2.1.2 Outcomes

We removed those records with missing data. We first plotted the observed creatinine and eGFR values (Figure ??). From the figure, we can see there are some abnormal records, which may be due to measurement or typo errors.

```
> nomiss_dat <- sub_merged[complete.cases(sub_merged), ];
> dim(nomiss_dat);
[1] 279002      13
> with(nomiss_dat, table((F_eGFR0>300) + (F_eGFR0>1000)));
      0      1      2
278701  265   36

> with(nomiss_dat, plot(dolab, F_eGFR0, pch=16, cex=0.3,
+                       xlab="Date of measurement", ylab="eGFR"));

> with(nomiss_dat, plot(dolab, creatinine, pch=16, cex=0.3,
+                       xlab="Date of measurement", ylab="Creatinine"));
```

We removed those records with eGFR ≥ 300 , which were considered to be errors. The updated distributions were shown in Figure 2.1.

```
> sub_nomiss <- subset(nomiss_dat, F_eGFR0<300);
> dim(sub_nomiss);
[1] 278701      13

> with(sub_nomiss[sample(1:nrow(sub_nomiss), 2000),],
+       plot(BW_TIME, F_eGFR0, pch=16, cex=0.3,
+           xlab="Time before ESRD", ylab="eGFR"));
> abline(h=15, col="red");

> with(sub_nomiss[sample(1:nrow(sub_nomiss), 2000),],
+       plot(BW_TIME, creatinine, pch=16, cex=0.3, #ylim=c(0,400),
+           xlab="Time before ESRD", ylab="Creatinine"))
```

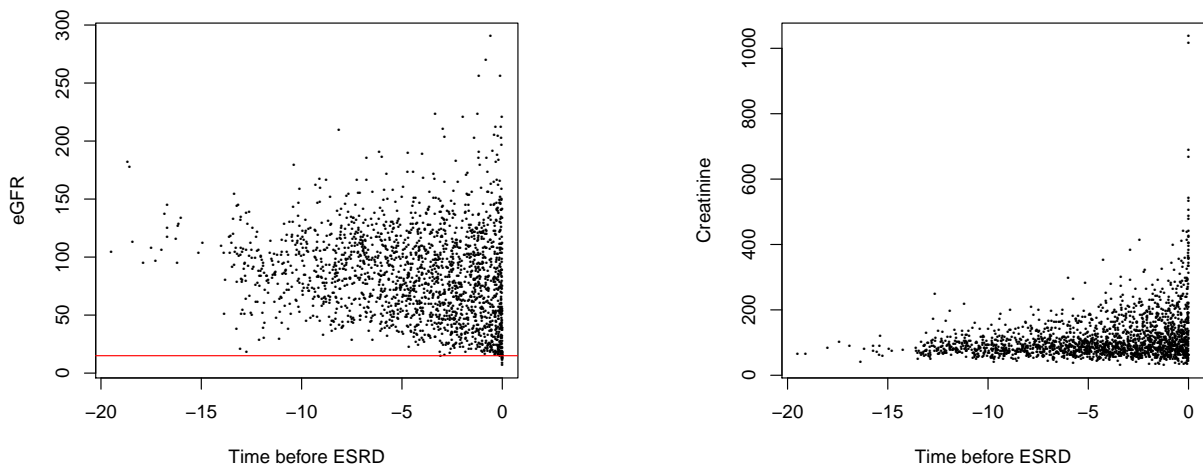


Figure 2.1: *Distribution of eGFR and creatinine with 2000 samples after removing eGFR ≥ 300 . The red line represents eGFR=15.*

As we here only focused on the patients developed ESRD, we extracted those subjects and plotted the distribution of eGFR. We then removed those patients with only one measurement, and summarized the number of measurement for each subject (Figure 2.2).

```

> event_dat <- subset(sub_nomiss, ESRD1_END==1);
> dim(event_dat);

[1] 41444    13

> length(unique(event_dat$Obs_id));

[1] 1199

> num_test <- table(event_dat$Obs_id);
> id_keep <- names(which(num_test>1));           #removed those patients with only one
> event_dat <- subset(event_dat, Obs_id %in% id_keep);
> (num_stat <- table(table(event_dat$Obs_id)));

  2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
51  21  23  18  17  19  23  18  21  21  22  17  21  23  20  18  18  23  14  16
22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
23  17  20  18  15  17  17  32  17  13  11  24  18  13  23  14  20  24  15  16
42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61
14  16  10  14  11  11  10  10  8   6  11  10  7   8   7   7   7   7   7   8
62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81
10  11  4   8  10   5   5   8   4   2   4   2   3   3   4   8   2   3   3   1
82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  98  99 100 102 103
 3   4   5   5   1   3   3   2   1   4   4   2   3   1   2   1   2   1   1   1
104 105 106 108 109 111 112 113 114 115 117 120 122 125 126 128 129 131 132 133
 4   1   2   1   1   2   1   2   1   3   4   1   1   1   1   1   1   1   1   1
136 142 146 147 148 153 173
 1   1   1   1   1   1   1   1

> dim(event_dat);

[1] 41412    13

> length(unique(event_dat$Obs_id));

[1] 1167

> event_dat <- transform(event_dat, log_eGFR=log(event_dat$F_eGFR0));

> par(mar=c(5,4,1,2));
> plot(as.numeric(names(num_stat)),num_stat,type="h",lwd=3,xaxs="i",xlim=c(0,140),
+       xlab="No. of measurements",ylab="No. of subjects",yaxt="n");
> axis(side=2);

> dest_eGFR <- density(event_dat$F_eGFR0);
> plot(dest_eGFR, xlim=c(0,150), xlab="eGFR", lwd=3, yaxs="i",
+       ylab="Density", main="ESRD subjects", bty="n");
> abline(v=quantile(event_dat$F_eGFR0, probs=c(50, 75, 90)/100),
+        col="red");

> dest_logeGFR <- density(log(event_dat$F_eGFR0));
> plot(dest_logeGFR, lwd=3, xlab="Ln(eGFR)",
+       ylab="Density", main="ESRD subjects");

```

As shown in Figure 2.3, the original distribution of eGFR is skewed, whilst it's close to be normal after log-transformed.

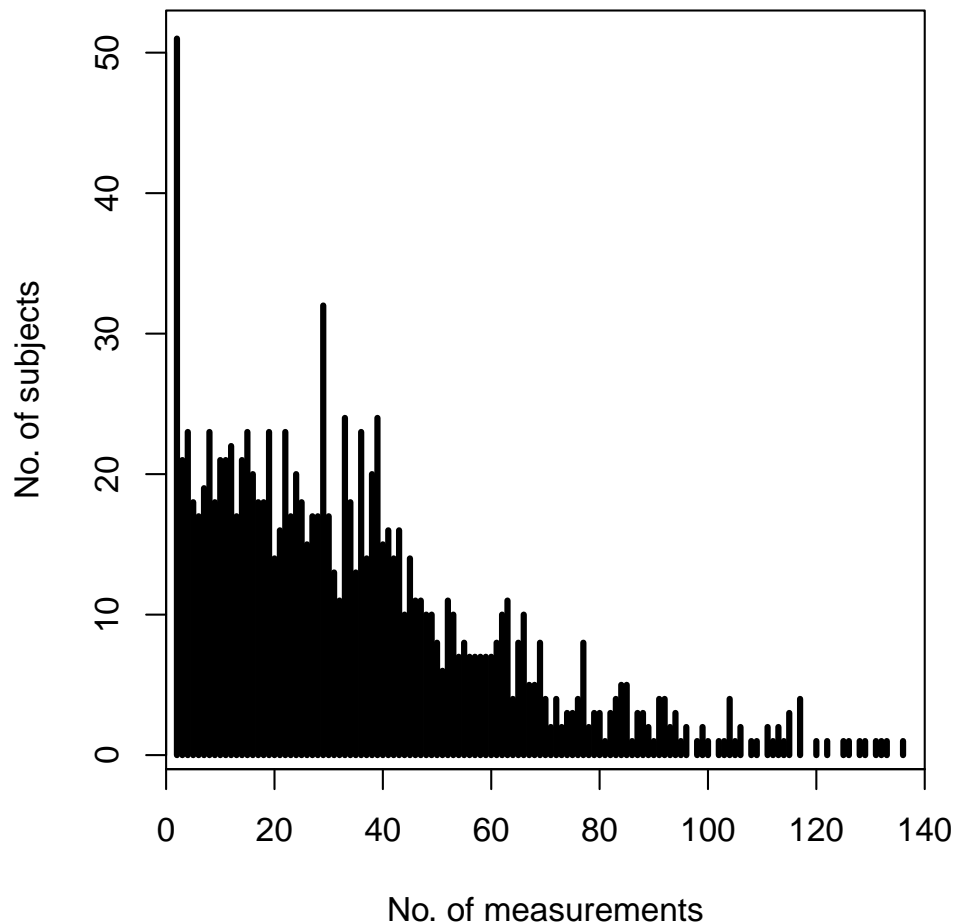


Figure 2.2: *Number of eGFR records for patients.*

2.2 Analysis

We first fit the model using the "lmm" function, and specified "linear" as the link function. The "lmm" with linear link function is similar with the "hlme" function, but the reason for using "lmm" function is that the confidence interval of predict values can be obtained by using the corresponding prediction function.

```
> (kn <- seq(-12, 0, , 5));
[1] -12 -9 -6 -3 0
> MM <- Ns(event_dat$BW_TIME, knots=kn);
> dim(MM);
[1] 41412 4
> MM <- detrend(MM, event_dat$BW_TIME);
> dim(MM);
[1] 41412 3
> head(MM);

      1      2      3
[1,] 0.04689555 0.38709327 -0.10183513
[2,] -0.02867914 0.38150137 -0.07236472
```

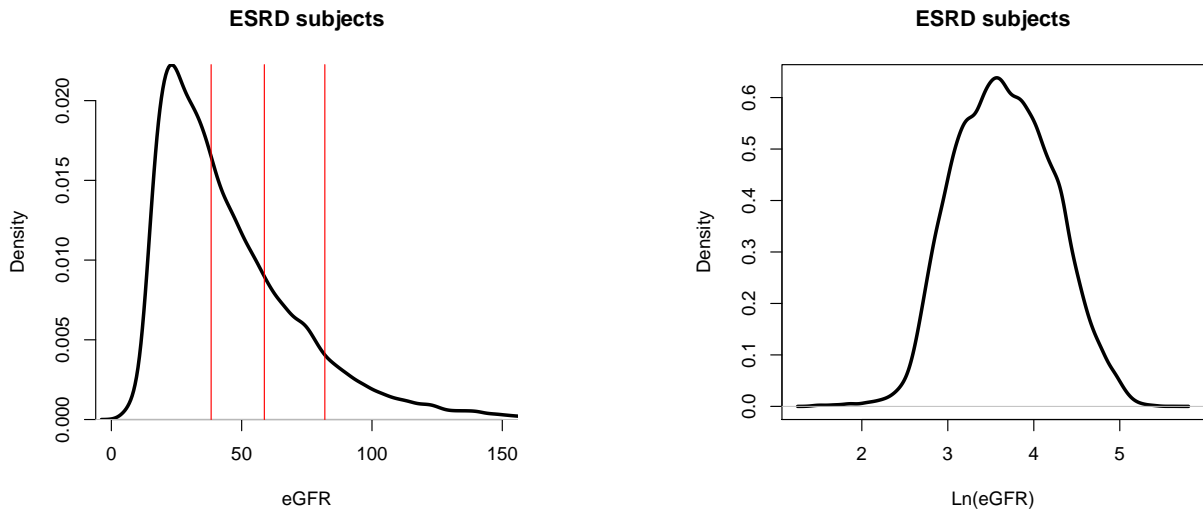


Figure 2.3: Distribution of $eGFR$ and $\log(eGFR)$ for ESRD subjects. Red lines represents the 50, 75 and 90 percent of subjects, respectively

```
[3,] -0.06988576  0.30174461 -0.04175845
[4,] -0.05881167 -0.13637325  0.03674734
[5,] -0.07504976  0.02845691  0.01176823
[6,] -0.34770860 -0.36093378  0.33529565

> (colnames(MM) <- paste("x", colnames(MM), sep=""));
[1] "x1" "x2" "x3"
> event_dat <- cbind(event_dat, MM);
> dim(event_dat);
[1] 41412    17

> event_model <- hlme(log_eGFR ~ BW_TIME + x1 + x2 + x3 + F_AGE + SEX + F_DMAGE,
+                       mixture =~ BW_TIME + x1 + x2 + x3,
+                       random =~ BW_TIME,
+                       subject = "Obs_id", ng=4, data=event_dat);

Be patient, hlme is running ...
The program took 10768.32 seconds

> event_model;

Heterogenous linear mixed model
  fitted by maximum likelihood method

hlme(fixed = log_eGFR ~ BW_TIME + x1 + x2 + x3 + F_AGE + SEX +
      F_DMAGE, mixture = ~BW_TIME + x1 + x2 + x3, random = ~BW_TIME,
      subject = "Obs_id", ng = 4, data = event_dat)

Statistical Model:
  Dataset: event_dat
  Number of subjects: 1167
  Number of observations: 41412
  Number of latent classes: 4
  Number of parameters: 30

Iteration process:
  Convergence criteria satisfied
  Number of iterations: 24
  Convergence criteria: parameters= 9e-08
```

```

: likelihood= 3.5e-06
: second derivatives= 1.4e-12

```

```

Goodness-of-fit statistics:
  maximum log-likelihood: -4265.84
  AIC: 8591.69
  BIC: 8743.55

```

Now we can further investigate the posterior probabilities of subjects in each class (Figure 2.4 and 2.5).

```

> clr<- c("black","red","blue", "yellow");
> postprob(event_model);
Posterior classification:
  class1 class2 class3 class4
N 273.00 304.00 436.00 154.0
% 23.39 26.05 37.36 13.2

Posterior classification table:
--> mean of posterior probabilities in each class
      prob1 prob2 prob3 prob4
class1 0.8572 0.1229 0.0195 0.0004
class2 0.0972 0.7891 0.1110 0.0027
class3 0.0420 0.1330 0.7514 0.0736
class4 0.0044 0.0194 0.1151 0.8612

Posterior probabilities above a threshold (%):
      class1 class2 class3 class4
prob>0.7  77.29  65.79  58.03  79.22
prob>0.8  67.77  52.30  45.18  71.43
prob>0.9  59.71  40.46  35.55  59.74
> str(event_model$pprob);
'data.frame':
  1167 obs. of  6 variables:
 $ Obs_id: num  16 21 22 36 37 42 44 54 56 71 ...
 $ class : int   3 1 4 2 2 1 1 3 3 3 ...
 $ prob1 : num  3.12e-03 7.50e-01 4.97e-09 4.06e-01 3.54e-05 ...
 $ prob2 : num  2.39e-01 2.50e-01 1.81e-06 5.94e-01 9.15e-01 ...
 $ prob3 : num  7.55e-01 8.15e-10 4.89e-03 2.42e-04 8.47e-02 ...
 $ prob4 : num  2.58e-03 5.46e-45 9.95e-01 1.04e-15 1.78e-11 ...
> plot_ppr
function(fit_model,clr)
{
  par(mfrow=c(1,3), mar=c(3,0,1,1), oma=c(0,3,0,0),
      las=1, bty="n", mgp=c(3,1,0)/1.6);
  ng <- fit_model$ng;
  ppr <- fit_model$pprob;
  num <- table(ppr$class);
  for(i in 1:ng) {
    hist(ppr[ppr$class==i,i+2],breaks=0:50/50,
         col=clr[i], border=clr[i], ylim=c(0,60),
         main="", xlab="", yaxt="n", yaxs="i");
    if(i==1) axis(side=2);
    text(0.4,30,paste("Class",i,": n=",num[i]),
         font=2,col=clr[i], cex=1);
  }
}
> plot_ppr(event_model,clr);

> ng <- event_model$ng;
> post_pr <- event_model$pprob;
> #par(bty="o");
> pairs(post_pr[, 2+1:ng], pch=16, col=clr[post_pr$class],
+       cex=0.4, gap=0);

```

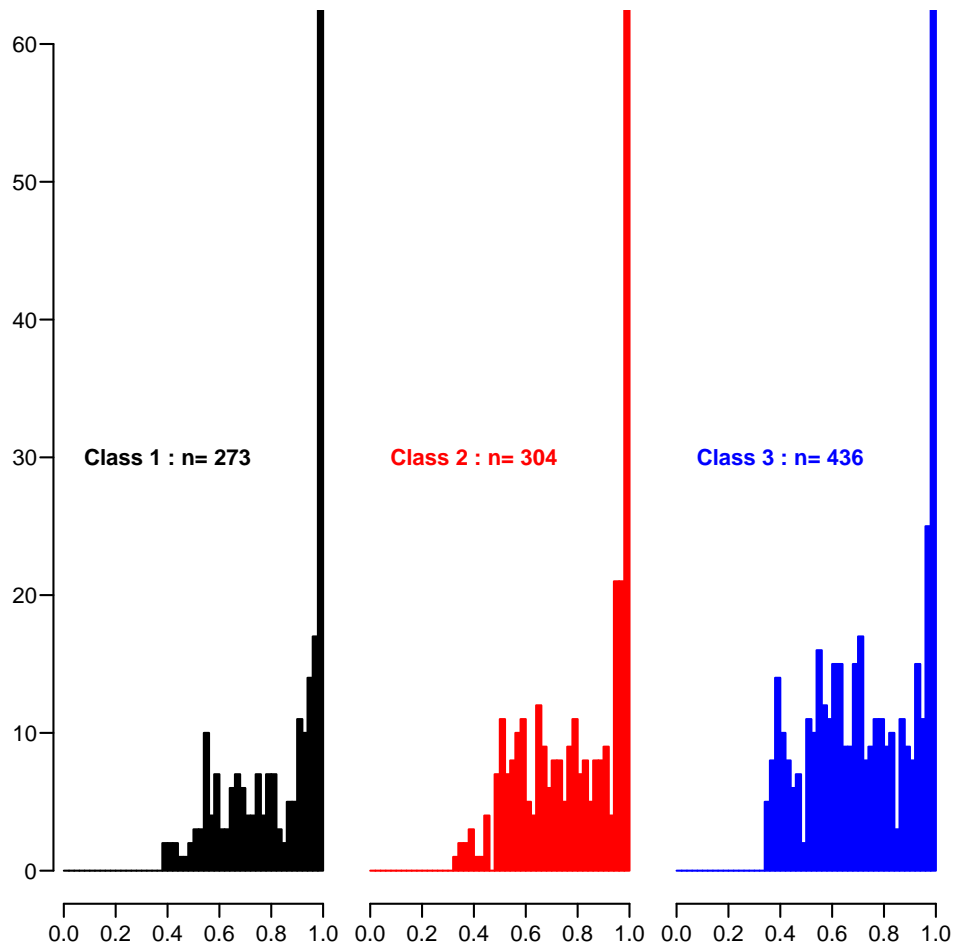


Figure 2.4: Posterior probabilities of 3 classes for the ESRD model.

We built a data set for prediction to plot the estimated trajectories. Here the median age 65, median DM duration 12 and sex as male were used. The function "plot_predictY" incorporated the prediction function "predictY" in LCMM package (Figure 2.6).

```
> (adjust_var <- event_model$Xnames[-c(1:5)]);
[1] "F_AGE" "SEX" "F_DMAGE"
> length(table(event_dat$BW_TIME))
[1] 6964
> x <- sort(unique(event_dat$BW_TIME));
> length(x);
[1] 6964
> wh <- match(x, event_dat$BW_TIME)[1:69*100];
> length(wh);
[1] 69
> plotdata <- data.frame(1, event_dat$BW_TIME[wh], MM[wh,],
+                        F_AGE = 65 + event_dat$BW_TIME[wh],
+                        SEX = 1,
+                        F_DMAGE = 15 + event_dat$BW_TIME[wh]);
> names(plotdata) <- event_model$Xnames;
> head(plotdata);
```

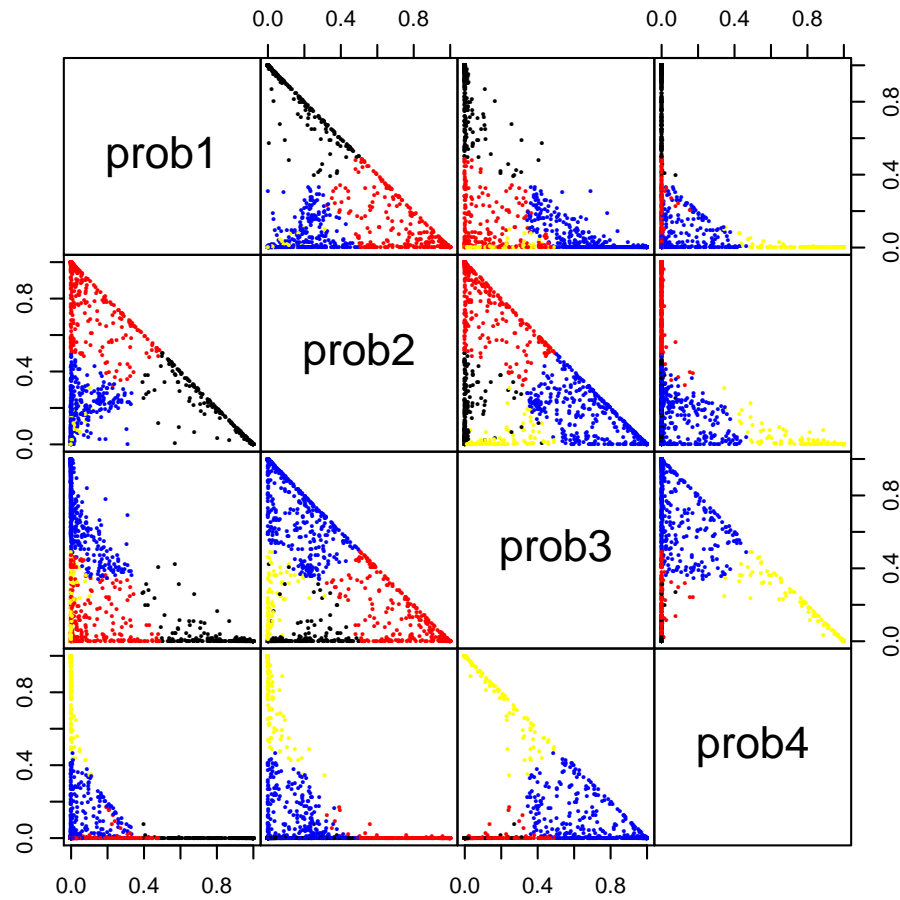


Figure 2.5: Pairwise posterior probabilities from the fitted model using "lcm".

```

intercept  BW_TIME      x1      x2      x3  F AGE SEX
1          1 -14.22040 -0.6919963 -0.001011389 -0.621725804 50.77960 1
2          1 -12.62149 -0.6178121 -0.138170718 -0.250384157 52.37851 1
3          1 -11.96715 -0.5874523 -0.194302593 -0.098414192 53.03285 1
4          1 -11.56468 -0.5682702 -0.228569177 -0.005717254 53.43532 1
5          1 -11.20602 -0.5490492 -0.258027568 0.073654734 53.79398 1
6          1 -10.87201 -0.5277820 -0.283754898 0.142452077 54.12799 1

F_DIMAGE
1 0.779603
2 2.378508
3 3.032854
4 3.435318
5 3.793977
6 4.127995

> range(plotdata$BW_TIME);
[1] -14.22039699 -0.08761123

> pred_event <- exp(predictY(event_model, plotdata, var.time="BW_TIME", draws=TRUE)$pred);
> ylim <- range(pred_event);
> lwd_main <- 4;
> lwd_ci <- 1;
> for (i in 1:ng)
+ {

```

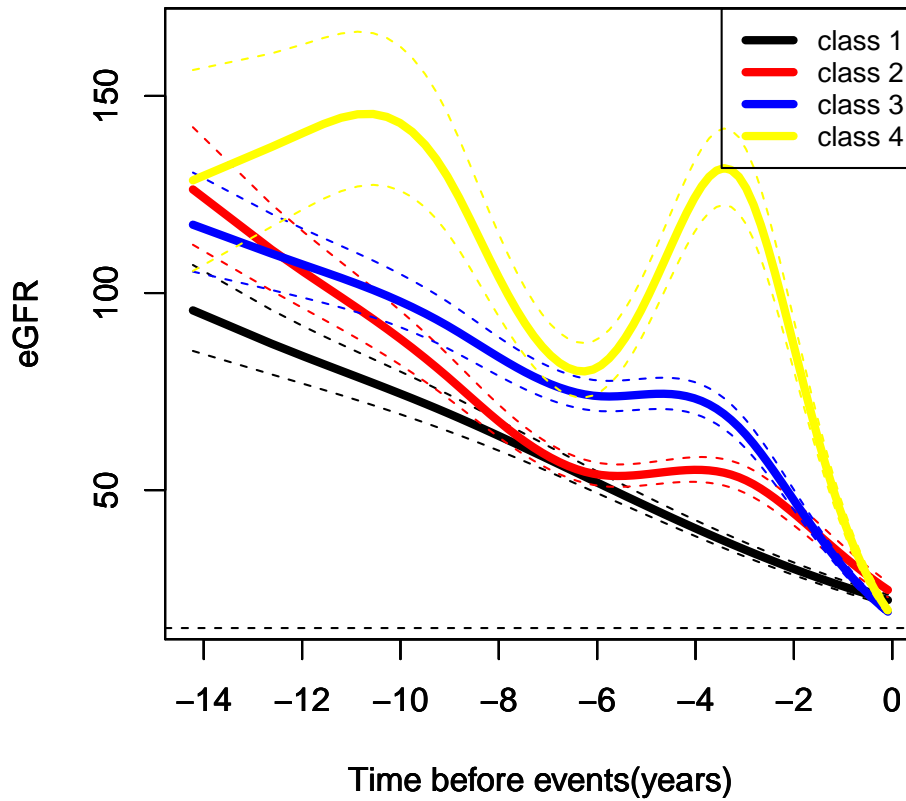


Figure 2.6: Mean trajectories of eGFR for the three latent classes of subjects developing ESRD. The horizontal dashed line represents eGFR is equal to 15

```
+ plot(y = pred_event[, i], x = plotdata$BW_TIME, type = "l", col= clrs[i],
+      ylim = ylim, lwd = lwd_main, xlab = "Time before events(years)", ylab = "eGFR");
+ points(y = pred_event[, i + ng], x = plotdata$BW_TIME, type = "l", lty = "dashed", col = clrs[i],
+        ylim = ylim, lwd = lwd_ci);
+ points(y = pred_event[, i + 2*ng], x = plotdata$BW_TIME, type = "l", lty = "dashed", col= clrs[i],
+        ylim = ylim, lwd = lwd_ci);
+ par(new = TRUE);
+ }
> abline(h = 15, lty = "dashed");
> legend("topright", legend = paste("class", 1:ng), col=clrs, lty=1, lwd=lwd_main, cex=0.8);
> par(new = FALSE);
```

Chapter 3

Reading data - DK

3.1 Reading the SDC clinical data

The data have been extracted in SAS-Xport format and in order to preserve the long variable names we have also exported a data-set of the original variable names:

```
1                                "Program: xn.sas"                                13:19 Friday, February 6, 2015
```

```
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          2.49 seconds  
      cpu time           0.43 seconds
```

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

```
-----  
C:\Bendix\Steno\HongKong\data\xn.sas  
-----
```

```
NOTE: Libref HER was successfully assigned as follows:
```

```
      Engine:           V9  
      Physical Name: C:\Bendix\Steno\HongKong\data
```

```
NOTE: Libname DATA refers to the same physical library as HER.
```

```
NOTE: Libref DATA was successfully assigned as follows:
```

```
      Engine:           V9  
      Physical Name: C:\Bendix\Steno\HongKong\data
```

```
NOTE: AUTOEXEC processing completed.
```

```
1          * First export the SAS-dataset ;  
2  
3          option validvarname = v6 ;  
4          libname rd xport './EPJnef.xpt' ;  
NOTE: Libref RD was successfully assigned as follows:  
      Engine:           XPORT  
      Physical Name: C:\Bendix\Steno\HongKong\data\EPJnef.xpt  
5          proc copy in = her out = rd ;  
6          select epjnef ;  
7          run ;
```

```
NOTE: Copying HER.EPJNEF to RD.EPJNEF (memtype=DATA).
```

```
NOTE: The variable name DEBUT_DIABETES has been truncated to DEBUT_DI.
```

```
NOTE: The variable DEBUT_DI now has a label set to DEBUT_DIABETES.
```

```
NOTE: The variable name D_STENOSTART has been truncated to D_STENOS.
```

```
NOTE: The variable D_STENOS now has a label set to D_STENOSTART.
```

```
NOTE: The variable name D_STENOSLUT has been truncated to D_STENOS.
```

```
NOTE: The variable D_STENOS now has a label set to D_STENOSLUT.
```

```
NOTE: Variable D_STENOS already exists on file RD.EPJNEF, using D_STENO2 instead.
```

```
NOTE: The variable name DATO_LABKALIVSSTIL has been truncated to DATO_LAB.
```

```
NOTE: The variable name ALAT_ENHED has been truncated to ALAT_ENH.
```

```
NOTE: The variable name ABDOMINALOMFANG has been truncated to ABDOMINA.
```

```
NOTE: The variable name ABDOMINALOMFANG_ENHED has been truncated to ABDOMINA.
```

```
NOTE: Variable ABDOMINA already exists on file RD.EPJNEF, using ABDOMIN2 instead.
```

```
NOTE: The variable name ASAT_ENHED has been truncated to ASAT_ENH.
```

```
NOTE: The variable name BAS_ENHED has been truncated to BAS_ENHE.
```

```
NOTE: The variable name CPEPTID_ENHED has been truncated to CPEPTID_.
```

```

NOTE: The variable name DUNA_ENHED has been truncated to DUNA_ENH.
NOTE: The variable name D_VITAMIN has been truncated to D_VITAMI.
NOTE: The variable name D_VITAMIN_ENHED has been truncated to D_VITAMI.
NOTE: Variable D_VITAMI already exists on file RD.EPJNEF, using D_VITAM2 instead.
NOTE: The variable name DIASTOLISKEPJ has been truncated to DIASTOLI.
NOTE: The variable name DIASTOLISKEPJ_ENHED has been truncated to DIASTOLI.
NOTE: Variable DIASTOLI already exists on file RD.EPJNEF, using DIASTOL2 instead.
NOTE: The variable name DIASTOLISKHJEMME has been truncated to DIASTOLI.
NOTE: Variable DIASTOLI already exists on file RD.EPJNEF, using DIASTOL3 instead.
NOTE: The variable name DIASTOLISKHJEMME_ENHED has been truncated to DIASTOLI.
NOTE: Variable DIASTOLI already exists on file RD.EPJNEF, using DIASTOL4 instead.
NOTE: The variable name DIURESE_ENHED has been truncated to DIURESE_.
NOTE: The variable name DUALB_ENHED has been truncated to DUALB_EN.
NOTE: The variable name FERATIO_ENHED has been truncated to FERATIO_.
NOTE: The variable name GAD_ENHED has been truncated to GAD_ENHE.
NOTE: The variable name GFR_ENHED has been truncated to GFR_ENHE.
NOTE: The variable name GLUC_ENHED has been truncated to GLUC_ENH.
NOTE: The variable name HDL_ENHED has been truncated to HDL_ENHE.
NOTE: The variable name HVILEPULS has been truncated to HVILEPUL.
NOTE: The variable name HVILEPULS_ENHED has been truncated to HVILEPUL.
NOTE: Variable HVILEPUL already exists on file RD.EPJNEF, using HVILEPU2 instead.
NOTE: The variable name KALIUM_ENHED has been truncated to KALIUM_E.
NOTE: The variable name LDL_ENHED has been truncated to LDL_ENHE.
NOTE: The variable name NATRIUM_ENHED has been truncated to NATRIUM_.
NOTE: The variable name PCREATININ has been truncated to PCREATIN.
NOTE: The variable name PCREATININ_ENHED has been truncated to PCREATIN.
NOTE: Variable PCREATIN already exists on file RD.EPJNEF, using PCREATI2 instead.
NOTE: The variable name SYSTOLISKEPJ has been truncated to SYSTOLIS.
NOTE: The variable name SYSTOLISKEPJ_ENHED has been truncated to SYSTOLIS.
NOTE: Variable SYSTOLIS already exists on file RD.EPJNEF, using SYSTOLI2 instead.
NOTE: The variable name SYSTOLISKHJEMME has been truncated to SYSTOLIS.
NOTE: Variable SYSTOLIS already exists on file RD.EPJNEF, using SYSTOLI3 instead.
NOTE: The variable name SYSTOLISKHJEMME_ENHED has been truncated to SYSTOLIS.
NOTE: Variable SYSTOLIS already exists on file RD.EPJNEF, using SYSTOLI4 instead.
NOTE: The variable name TCHOL_ENHED has been truncated to TCHOL_EN.
NOTE: The variable name TSH_ENHED has been truncated to TSH_ENHE.
NOTE: The variable name TRANS_ENHED has been truncated to TRANS_EN.
NOTE: The variable name TRIGLYCERID has been truncated to TRIGLYCE.
NOTE: The variable name TRIGLYCERID_ENHED has been truncated to TRIGLYCE.
NOTE: Variable TRIGLYCE already exists on file RD.EPJNEF, using TRIGLYC2 instead.
NOTE: The variable name UALBCREA_ENHED has been truncated to UALBCREA.
NOTE: Variable UALBCREA already exists on file RD.EPJNEF, using UALBCRE2 instead.
NOTE: The variable name VLDL_ENHED has been truncated to VLDL_ENH.
NOTE: The variable name B12_ENHED has been truncated to B12_ENHE.
NOTE: The variable name BLODGLUKOSE has been truncated to BLODGLUK.
NOTE: The variable name BLODGLUKOSE_ENHED has been truncated to BLODGLUK.
NOTE: Variable BLODGLUK already exists on file RD.EPJNEF, using BLODGLU2 instead.
NOTE: The variable name BMI_ENHED has been truncated to BMI_ENHE.
NOTE: The variable name CIVILSTANDSKODE has been truncated to CIVILSTA.
NOTE: The variable name DIASTOLISK_ARM has been truncated to DIASTOLI.
NOTE: Variable DIASTOLI already exists on file RD.EPJNEF, using DIASTOL5 instead.
NOTE: The variable name DIASTOLISK_ARM_ENHED has been truncated to DIASTOLI.
NOTE: Variable DIASTOLI already exists on file RD.EPJNEF, using DIASTOL6 instead.
NOTE: The variable name EGFR_ENHED has been truncated to EGFR_ENH.
NOTE: The variable name HAEMOGLOBIN has been truncated to HAEMOGLO.
NOTE: The variable name HAEMOGLOBIN_ENHED has been truncated to HAEMOGLO.
NOTE: Variable HAEMOGLO already exists on file RD.EPJNEF, using HAEMOGL2 instead.
NOTE: The variable name HBA1C_ENHED has been truncated to HBA1C_EN.
NOTE: The variable name HEIGHT_ENHED has been truncated to HEIGHT_E.
NOTE: The variable name MIDDELBLDGLUKOSEEPJ has been truncated to MIDDELBL.
NOTE: The variable name MIDDELBLDGLUKOSEEPJ_ENHED has been truncated to MIDDELBL.
NOTE: Variable MIDDELBL already exists on file RD.EPJNEF, using MIDDELBL2 instead.
NOTE: The variable name SYGDOM_DIABETES has been truncated to SYGDOM_D.
NOTE: The variable SYGDOM_D now has a label set to SYGDOM_DIABETES.
NOTE: The variable name SYSTOLISK_ARM has been truncated to SYSTOLIS.
NOTE: Variable SYSTOLIS already exists on file RD.EPJNEF, using SYSTOLI5 instead.
NOTE: The variable name SYSTOLISK_ARM_ENHED has been truncated to SYSTOLIS.
NOTE: Variable SYSTOLIS already exists on file RD.EPJNEF, using SYSTOLI6 instead.
NOTE: The variable name WEIGHT_ENHED has been truncated to WEIGHT_E.
NOTE: There were 517919 observations read from the data set HER.EPJNEF.
NOTE: The data set RD.EPJNEF has 517919 observations and 96 variables.
WARNING: Labels exceeding length 40 are not supported by engine XPORT and are being truncated.
NOTE: PROCEDURE COPY used (Total process time):
      real time          1:02.77
      cpu time           5.66 seconds

```

```

8
9      * Names are long in the original dataset and so funny in the Xport
10     dataset, so we retrieve them for inclusion in the R-dataset ;
11
12     proc contents data = her.epjnef out = vnam noprint ; run ;

```

```

NOTE: The data set WORK.VNAM has 96 observations and 40 variables.
NOTE: PROCEDURE CONTENTS used (Total process time):
      real time          0.12 seconds
      cpu time           0.03 seconds

```

```
13      proc sort  data = vnam ( keep = varnum name ) ; by varnum ; run ;
```

NOTE: There were 96 observations read from the data set WORK.VNAM.

NOTE: The data set WORK.VNAM has 96 observations and 2 variables.

NOTE: PROCEDURE SORT used (Total process time):

```
real time      0.01 seconds
cpu time       0.00 seconds
```

```
14      proc print data = vnam ; run ;
```

NOTE: There were 96 observations read from the data set WORK.VNAM.

NOTE: The PROCEDURE PRINT printed page 1.

NOTE: PROCEDURE PRINT used (Total process time):

```
real time      0.00 seconds
cpu time       0.00 seconds
```

```
15
```

```
16      libname rn xport './EPJnam.xpt' ;
```

NOTE: Libref RN was successfully assigned as follows:

```
Engine:        XPORT
```

```
Physical Name: C:\Bendix\Steno\HongKong\data\EPJnam.xpt
```

```
17      proc copy  in = work  out = rn ;
```

```
18      select vnam ;
```

```
19      run ;
```

NOTE: Copying WORK.VNAM to RN.VNAM (memtype=DATA).

WARNING: Engine XPORT does not support SORTEDBY operations. SORTEDBY information cannot be copied.

NOTE: There were 96 observations read from the data set WORK.VNAM.

NOTE: The data set RN.VNAM has 96 observations and 2 variables.

NOTE: PROCEDURE COPY used (Total process time):

```
real time      0.16 seconds
cpu time       0.01 seconds
```

```
20
```

NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414

NOTE: The SAS System used:

```
real time      1:05.81
cpu time       6.19 seconds
```

The SAS System

13:19 Friday, February 6, 2015 1

Obs	NAME	VARNUM
1	NEWID	1
2	SEX	2
3	DOB	3
4	DEBUT_DIABETES	4
5	D_DTH	5
6	D_STENOSTART	6
7	D_STENOSLUT	7
8	DATO_LABKALIVSSTIL	8
9	ALAT	9
10	ALAT_ENHED	10
11	ABDOMINALOMFANG	11
12	ABDOMINALOMFANG_ENHED	12
13	ALKOHOL	13
14	ASAT	14
15	ASAT_ENHED	15
16	BAS	16
17	BAS_ENHED	17
18	CPEPTID	18
19	CPEPTID_ENHED	19
20	DUNA	20
21	DUNA_ENHED	21
22	D_VITAMIN	22
23	D_VITAMIN_ENHED	23
24	DIASTOLISKEPJ	24
25	DIASTOLISKEPJ_ENHED	25
26	DIASTOLISKHJEMME	26
27	DIASTOLISKHJEMME_ENHED	27
28	DIURESE	28
29	DIURESE_ENHED	29
30	DUALB	30
31	DUALB_ENHED	31
32	FE	32
33	FE_ENHED	33
34	FERATIO	34
35	FERATIO_ENHED	35
36	GAD	36
37	GAD_ENHED	37
38	GFR	38
39	GFR_ENHED	39
40	GLUC	40
41	GLUC_ENHED	41
42	HDL	42

43	HDL_ENHED	43
44	HVILEPULS	44
45	HVILEPULS_ENHED	45
46	KALIUM	46
47	KALIUM_ENHED	47
48	LDL	48
49	LDL_ENHED	49
50	NATRIUM	50
51	NATRIUM_ENHED	51
52	PCREATININ	52
53	PCREATININ_ENHED	53
54	SYSTOLISKEPJ	54
55	SYSTOLISKEPJ_ENHED	55
56	SYSTOLISKHJEMME	56
57	SYSTOLISKHJEMME_ENHED	57
58	TCHOL	58
59	TCHOL_ENHED	59
60	TSH	60
61	TSH_ENHED	61
62	TRANS	62
63	TRANS_ENHED	63
64	TRIGLYCERID	64
65	TRIGLYCERID_ENHED	65
66	UALBCREA	66
67	UALBCREA_ENHED	67
68	VLDL	68
69	VLDL_ENHED	69
70	_MERGE	70
71	B12	71
72	B12_ENHED	72
73	BLODGLUKOSE	73
74	BLODGLUKOSE_ENHED	74
75	BMI	75
76	BMI_ENHED	76
77	CIVILSTANDSKODE	77
78	DIASTOLISK_ARM	78
79	DIASTOLISK_ARM_ENHED	79
80	EGFR	80
81	EGFR_ENHED	81
82	HAEMOGLOBIN	82
83	HAEMOGLOBIN_ENHED	83
84	HBA1C	84
85	HBA1C_ENHED	85
86	HEIGHT	86
87	HEIGHT_ENHED	87
88	MIDDELBLODGLUKOSEEPJ	88
89	MIDDELBLODGLUKOSEEPJ_ENHED	89
90	MOTION	90
91	RYGNING	91
92	SYGDOM_DIABETES	92
93	SYSTOLISK_ARM	93
94	SYSTOLISK_ARM_ENHED	94
95	WEIGHT	95
96	WEIGHT_ENHED	96

These are now read by R and used to devise a groomed dataset from the clinic:

```
> library( Epi )
> library( foreign )
> clear()
> system.time( sdc <- read.xport( "../data/EPJnef.xpt" ) )
   user system elapsed 
2.949   0.258   7.635

> dim( sdc )
[1] 517919    96

> names( sdc )
[1] "NEWID"      "SEX"        "DOB"        "DEBUT_DI"   "D_DTH"      "D_STENOS"   "D_STENO2"
[8] "DATO_LAB"   "ALAT"       "ALAT_ENH"   "ABDOMINA"   "ABDOMIN2"   "ALKOHOL"    "ASAT"
[15] "ASAT_ENH"   "BAS"        "BAS_ENHE"   "CPEPTID"    "CPEPTID_"   "DUNA"       "DUNA_ENH"
[22] "D_VITAMI"   "D_VITAM2"   "DIASTOLI"   "DIASTOL2"   "DIASTOL3"   "DIASTOL4"   "DIURESE"
[29] "DIURESE_"   "DUALB"      "DUALB_EN"   "FE"         "FE_ENHED"   "FERATIO"    "FERATIO_"
[36] "GAD"        "GAD_ENHE"   "GFR"        "GFR_ENHE"   "GLUC"       "GLUC_ENH"   "HDL"
[43] "HDL_ENHE"   "HVILEPUL"   "HVILEPU2"   "KALIUM"     "KALIUM_E"   "LDL"        "LDL_ENHE"
[50] "NATRIUM"    "NATRIUM_"   "PCREATIN"   "PCREATI2"   "SYSTOLIS"   "SYSTOLI2"   "SYSTOLI3"
[57] "SYSTOLI4"   "TCHOL"      "TCHOL_EN"   "TSH"        "TSH_ENHE"   "TRANS"      "TRANS_EN"
[64] "TRIGLYCE"   "TRIGLYC2"   "UALBCREA"   "UALBCRE2"   "VLDL"       "VLDL_ENH"   "X_MERGE"
[71] "B12"        "B12_ENHE"   "BLODGLUK"   "BLODGLU2"   "BMI"        "BMI_ENHE"   "CIVILSTA"
```

```
[78] "DIASTOL5" "DIASTOL6" "EGFR" "EGFR_ENH" "HAEMOGLO" "HAEMOGL2" "HBA1C"
[85] "HBA1C_EN" "HEIGHT" "HEIGHT_E" "MIDDELBL" "MIDDELB2" "MOTION" "RYGNING"
[92] "SYGDOM_D" "SYSTOLI5" "SYSTOLI6" "WEIGHT" "WEIGHT_E"
```

```
> nam <- read.xport( "./data/EPJnam.xpt" )
> head( nam )
```

	NAME	VARNUM
1	NEWID	1
2	SEX	2
3	DOB	3
4	DEBUT_DIABETES	4
5	D_DTH	5
6	D_STENOSTART	6

```
> nnam <- gsub( "_", ".", tolower(nam$NAME) )
> cbind( names(sdc), nnam )
```

		nnam
[1,]	"NEWID"	"newid"
[2,]	"SEX"	"sex"
[3,]	"DOB"	"dob"
[4,]	"DEBUT_DI"	"debut.diabetes"
[5,]	"D_DTH"	"d.dth"
[6,]	"D_STENOS"	"d.stenostart"
[7,]	"D_STENO2"	"d.stenoslut"
[8,]	"DATO_LAB"	"dato.labkalivsstil"
[9,]	"ALAT"	"alat"
[10,]	"ALAT_ENH"	"alat.enhed"
[11,]	"ABDOMINA"	"abdominalomfang"
[12,]	"ABDOMIN2"	"abdominalomfang.enhed"
[13,]	"ALKOHOL"	"alkohol"
[14,]	"ASAT"	"asat"
[15,]	"ASAT_ENH"	"asat.enhed"
[16,]	"BAS"	"bas"
[17,]	"BAS_ENHE"	"bas.enhed"
[18,]	"CPEPTID"	"cpeptid"
[19,]	"CPEPTID_"	"cpeptid.enhed"
[20,]	"DUNA"	"duna"
[21,]	"DUNA_ENH"	"duna.enhed"
[22,]	"D_VITAMI"	"d.vitamin"
[23,]	"D_VITAM2"	"d.vitamin.enhed"
[24,]	"DIASTOLI"	"diastoliskepj"
[25,]	"DIASTOL2"	"diastoliskepj.enhed"
[26,]	"DIASTOL3"	"diastoliskhjemme"
[27,]	"DIASTOL4"	"diastoliskhjemme.enhed"
[28,]	"DIURESE"	"diurese"
[29,]	"DIURESE_"	"diurese.enhed"
[30,]	"DUALB"	"dualb"
[31,]	"DUALB_EN"	"dualb.enhed"
[32,]	"FE"	"fe"
[33,]	"FE_ENHED"	"fe.enhed"
[34,]	"FERATIO"	"feratio"
[35,]	"FERATIO_"	"feratio.enhed"
[36,]	"GAD"	"gad"
[37,]	"GAD_ENHE"	"gad.enhed"
[38,]	"GFR"	"gfr"
[39,]	"GFR_ENHE"	"gfr.enhed"
[40,]	"GLUC"	"gluc"
[41,]	"GLUC_ENH"	"gluc.enhed"
[42,]	"HDL"	"hdl"
[43,]	"HDL_ENHE"	"hdl.enhed"
[44,]	"HVILEPUL"	"hvilepuls"
[45,]	"HVILEPU2"	"hvilepuls.enhed"
[46,]	"KALIUM"	"kalium"
[47,]	"KALIUM_E"	"kalium.enhed"
[48,]	"LDL"	"ldl"
[49,]	"LDL_ENHE"	"ldl.enhed"

```

[50,] "NATRIUM" "natrium"
[51,] "NATRIUM_" "natrium.enhed"
[52,] "PCREATIN" "pcreatinin"
[53,] "PCREATI2" "pcreatinin.enhed"
[54,] "SYSTOLIS" "systoliskepj"
[55,] "SYSTOLI2" "systoliskepj.enhed"
[56,] "SYSTOLI3" "systoliskhjemme"
[57,] "SYSTOLI4" "systoliskhjemme.enhed"
[58,] "TCHOL" "tchol"
[59,] "TCHOL_EN" "tchol.enhed"
[60,] "TSH" "tsh"
[61,] "TSH_ENHE" "tsh.enhed"
[62,] "TRANS" "trans"
[63,] "TRANS_EN" "trans.enhed"
[64,] "TRIGLYCE" "triglycerid"
[65,] "TRIGLYC2" "triglycerid.enhed"
[66,] "UALBCREA" "ualbcrea"
[67,] "UALBCRE2" "ualbcrea.enhed"
[68,] "VLDL" "vldl"
[69,] "VLDL_ENH" "vldl.enhed"
[70,] "X_MERGE" ".merge"
[71,] "B12" "b12"
[72,] "B12_ENHE" "b12.enhed"
[73,] "BLODGLUK" "blodglukose"
[74,] "BLODGLU2" "blodglukose.enhed"
[75,] "BMI" "bmi"
[76,] "BMI_ENHE" "bmi.enhed"
[77,] "CIVILSTA" "civilstandskode"
[78,] "DIASTOL5" "diastolisk.arm"
[79,] "DIASTOL6" "diastolisk.arm.enhed"
[80,] "EGFR" "egfr"
[81,] "EGFR_ENH" "egfr.enhed"
[82,] "HAEMOGLO" "haemoglobin"
[83,] "HAEMOGL2" "haemoglobin.enhed"
[84,] "HBA1C" "hba1c"
[85,] "HBA1C_EN" "hba1c.enhed"
[86,] "HEIGHT" "height"
[87,] "HEIGHT_E" "height.enhed"
[88,] "MIDDELBL" "middelblodglukoseepj"
[89,] "MIDDELB2" "middelblodglukoseepj.enhed"
[90,] "MOTION" "motion"
[91,] "RYGNING" "rygning"
[92,] "SYGDOM_D" "sygdom.diabetes"
[93,] "SYSTOLI5" "systolisk.arm"
[94,] "SYSTOLI6" "systolisk.arm.enhed"
[95,] "WEIGHT" "weight"
[96,] "WEIGHT_E" "weight.enhed"

> names(sdc)<-nnam
> sdc[1:10,1:8]
  newid sex   dob      debut.diabetes d.dth d.stenostart d.stenoslut
1     1 Male -2348          1993 19935          15593          15790
2     1 Male -2348          1993 19935          15593          15790
3     1 Male -2348          1993 19935          15593          15790
4     1 Male -2348          1993 19935          15593          15790
5     1 Male -2348          1993 19935          15593          15790
6     1 Male -2348          1993 19935          15593          15790
7     1 Male -2348          1993 19935          15593          15790
8     2      1336              NA              NA              NA
9     3 Male -14807          1976 17834          12329          17834
10    3 Male -14807          1976 17834          12329          17834
  dato.labkalivsstil
1          15593
2          15628
3          15644
4          15664
5          15713

```

```

6          15735
7          15782
8           NA
9          13883
10         13884

```

Sex is coded a little funny, but it appears that no person is coded as both male and female:

```

> tt <- with( sdc, table( newid, sex ) )
> head(tt)

      sex
newid  Female Male
  1 16         0    7
  2 13         0    0
  3 11         0   46
  4 12         0   24
  5 20         0   63
  6  0         0    2

> wh <- apply( tt[,-1], 1, min )
> summary( wh )

      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
       0       0       0       0       0       0

```

Thus we can determine sex for all those values of `newid` for which a male or female code exists, and subsequently merge these into the `sdc` dataframe and only maintain those with a valid value of sex (using `all=TRUE`).

```

> kn <- as.integer( sdc$sex )
> table( kn, sdc$sex )

kn      Female    Male
 1 78325         0      0
 2  0 199138         0
 3  0         0 240456

> kn <- ave( kn, sdc$newid, FUN=max )
> table( kn, sdc$sex )

kn      Female    Male
 1  2603         0      0
 2 32747 199138         0
 3 42975         0 240456

> sdc$sex <- factor( kn, levels=3:2, labels=c("M","F") )
> table( kn, sdc$sex, exclude=NULL )

kn      M      F  <NA>
 1      0      0  2603
 2      0 231885      0
 3 283431      0      0
<NA>      0      0      0

> sdc <- subset( sdc, !is.na(sex) )
> table( sdc$sex, useNA="ifany" )

      M      F
283431 231885

```

Thus, the dataframe `sdc` is now cut down to persons with known sex.

3.1.1 Measurement unit variables

A large fraction of the variables are just uniform repeats of the units used for the actually measured variables, so we just make a table of the actually occurring values for measurement units for each variable and paste them in order to check if different units have been used in the same variable. Finally we remove the corresponding variables from the data frame:

```
> en <- grep( ".enhed", names(sdc) )
> what <- function(x) paste( names(table(x)), collapse=" " )
> units <- sapply( sdc[,en], what )
> names( units ) <- gsub( ".enhed", "", names(units) )
> cbind( units )
```

	units
alat	" U/l"
abdominalomfang	" cm"
asat	" U/l"
bas	" U/l"
cpeptid	" pmol/l"
duna	" mmol/d"
d.vitamin	" nmol/l"
diastoliskepj	" mm Hg"
diastoliskhjemme	" mm Hg"
diurese	" ml"
dualb	" mg/d"
fe	" \xb5mol/l"
feratio	" %"
gad	" kIU/l"
gfr	" ml/min"
gluc	" mmol/l"
hdl	" mmol/l"
hvillepuls	" slag/min slag/min."
kalium	" mmol/l"
ldl	" mmol/l"
natrium	" mmol/l"
pcreatinin	" \xb5mol/l"
systoliskepj	" mm Hg"
systoliskhjemme	" mm Hg"
tchol	" mmol/l"
tsh	" \xd7 10⁻³ mIU/l mIU/l"
trans	" \xb5mol/l"
triglycerid	" mmol/l"
ualbcrea	" mg/g"
vldl	" mmol/l"
b12	" pmol/l"
blodglukose	" mmol/l"
bmi	" Kg/m ² kg/m²"
diastolisk.arm	" mmHg mm Hg"
egfr	" ml/min"
haemoglobin	" mmol/l"
hba1c	" mmol/mol"
height	" m"
middelblodglukoseepj	" mmol/l"
systolisk.arm	" mmHg mm Hg"
weight	" kg Kg"

```
> sdc <- sdc[,-en]
```

Note that the reason that the database is constructed this way is that it enables a smooth change of measurement units over time; but as seen from the above table of units, there are no such changes in this dataset.

We then groom and rename the date variables; for some odd reason the date of diagnosis has ended up as a factor - presumably because it only is recorded by year:

```
> str( sdc )
'data.frame':      515316 obs. of  55 variables:
 $ newid      : num  1 1 1 1 1 1 1 3 3 3 ...
 $ sex        : Factor w/ 2 levels "M","F": 1 1 1 1 1 1 1 1 1 1 ...
 $ dob        : num  -2348 -2348 -2348 -2348 -2348 ...
 $ debut.diabetes : Factor w/ 83 levels "", "1933",...: 62 62 62 62 62 62 62 45 4
 $ d.dth      : num  19935 19935 19935 19935 19935 ...
 $ d.stenostart : num  15593 15593 15593 15593 15593 ...
 $ d.stenoslut  : num  15790 15790 15790 15790 15790 ...
 $ dato.labkalivsstil : num  15593 15628 15644 15664 15713 ...
 $ alat       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ abdominalomfang : num  NA NA NA NA NA NA NA NA NA NA ...
 $ alkohol    : Factor w/ 11 levels "", "0 Genstande/dag",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ asat       : num  NA 28 NA 24 NA 27 NA NA NA NA ...
 $ bas        : num  NA 177 NA 159 NA 152 NA NA NA NA ...
 $ cpeptid    : num  NA 787 NA NA NA NA NA NA NA NA ...
 $ duna       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ d.vitamin  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ diastoliskepj : num  NA 93 95 NA 85 92 NA NA NA NA ...
 $ diastoliskhjemme : num  NA NA NA NA NA NA NA NA NA NA ...
 $ diurese    : num  NA NA NA NA NA NA NA NA NA NA ...
 $ dualb      : num  NA NA NA NA NA NA NA NA NA NA ...
 $ fe         : num  NA 18 NA NA NA NA NA NA NA NA ...
 $ feratio    : num  NA 22 NA NA NA NA NA NA NA NA ...
 $ gad        : num  NA NA NA NA NA NA NA NA NA NA ...
 $ gfr        : num  NA NA NA NA NA NA NA NA NA NA ...
 $ gluc       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ hdl        : num  NA 1.3 NA 1.25 NA ...
 $ hvilepuls  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ kalium     : num  NA 3.8 NA NA NA ...
 $ ldl        : num  NA 3.2 NA 2.6 NA ...
 $ natrium    : num  NA 136 NA NA NA 140 NA NA NA NA ...
 $ pcreatinin : num  NA 82 NA NA NA 87 NA NA NA NA ...
 $ systoliskepj : num  NA 143 150 NA 140 144 NA NA NA NA ...
 $ systoliskhjemme : num  NA NA NA NA NA NA NA NA NA NA ...
 $ tchol      : num  NA 5.2 NA 4.2 NA ...
 $ tsh        : num  NA 2 NA NA NA NA NA NA NA NA ...
 $ trans      : num  NA 41 NA NA NA NA NA NA NA NA ...
 $ triglycerid : num  NA 1.45 NA 0.85 NA ...
 $ ualbrecrea : num  4 NA NA NA NA 4 NA NA NA NA ...
 $ vldl       : num  NA 0.7 NA 0.4 NA ...
 $ .merge     : num  1 1 1 1 1 1 1 1 1 1 ...
 $ b12        : num  NA NA NA NA NA NA NA NA NA NA ...
 $ blodglukose : num  NA NA NA NA NA NA NA NA NA NA ...
 $ bmi        : num  30 NA NA NA NA NA NA NA NA ...
 $ civilstandskode : Factor w/ 10 levels "", "D\xfb8d", "Enke/Enkemand",...: 2 2 2 2 2 2 2 2 2 ...
 $ diastolisk.arm : num  NA NA NA NA NA NA NA NA NA NA ...
 $ egfr       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ haemoglobin : num  NA 10.2 NA NA NA ...
 $ hba1c      : num  63 57 NA 53 51 48 48 79 83 78 ...
 $ height     : num  1.79 NA NA NA NA ...
 $ middelblodglukoseepj : num  NA NA NA NA NA NA NA NA ...
 $ motion     : Factor w/ 10 levels "", "30x7MI", "<30x7MI",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ rygning    : Factor w/ 16 levels "", ">20 cigaretter/dag",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ sygdom.diabetes : Factor w/ 39 levels "", "DE10", "DE100",...: 14 14 14 14 14 14 14 14 14 14 ...
 $ systolisk.arm : num  NA NA NA NA NA NA NA NA NA NA ...
 $ weight     : num  97.3 99.7 96.5 94.3 NA ...

> sdc[1:10,1:9]
      newid sex  dob      debut.diabetes d.dth d.stenostart d.stenoslut
1         1  M -2348          1993 19935          15593          15790
2         1  M -2348          1993 19935          15593          15790
3         1  M -2348          1993 19935          15593          15790
4         1  M -2348          1993 19935          15593          15790
5         1  M -2348          1993 19935          15593          15790
6         1  M -2348          1993 19935          15593          15790
```

```

7      1      M  -2348          1993 19935          15593          15790
9      3      M -14807          1976 17834          12329          17834
10     3      M -14807          1976 17834          12329          17834
11     3      M -14807          1976 17834          12329          17834
      dato.labkalivsstil alat
1              15593      NA
2              15628      NA
3              15644      NA
4              15664      NA
5              15713      NA
6              15735      NA
7              15782      NA
9              13883      NA
10             13884      NA
11             14005      NA

> dvar <- c(3:8)
> cbind( names(sdc)[dvar], nnam <- c("dob","dodm","dodd","doin","dox","dolab") )

      [,1]      [,2]
[1,] "dob"      "dob"
[2,] "debut.diabetes" "dodm"
[3,] "d.dth"     "dodd"
[4,] "d.stenostart" "doin"
[5,] "d.stenoslut" "dox"
[6,] "dato.labkalivsstil" "dolab"

>      names(sdc)[dvar]<-nnam
> sdc[,dvar[-2]] <- sdc[,dvar[-2]]/365.25+1960
> sdc[,dvar[ 2]] <- as.numeric( as.character(sdc[,dvar[ 2]]) )
> sdc[1:10,1:12]

      newid sex      dob dodm      dodd      doin      dox      dolab alat abdominalomfang
1         1  M 1953.572 1993 2014.579 2002.691 2003.231 2002.691      NA              NA
2         1  M 1953.572 1993 2014.579 2002.691 2003.231 2002.787      NA              NA
3         1  M 1953.572 1993 2014.579 2002.691 2003.231 2002.831      NA              NA
4         1  M 1953.572 1993 2014.579 2002.691 2003.231 2002.886      NA              NA
5         1  M 1953.572 1993 2014.579 2002.691 2003.231 2003.020      NA              NA
6         1  M 1953.572 1993 2014.579 2002.691 2003.231 2003.080      NA              NA
7         1  M 1953.572 1993 2014.579 2002.691 2003.231 2003.209      NA              NA
9         3  M 1919.461 1976 2008.827 1993.755 2008.827 1998.010      NA              NA
10        3  M 1919.461 1976 2008.827 1993.755 2008.827 1998.012      NA              NA
11        3  M 1919.461 1976 2008.827 1993.755 2008.827 1998.344      NA              NA
      alkohol asat
1              NA
2              28
3              NA
4              24
5              NA
6              27
7              NA
9              NA
10             NA
11             NA

> names( sdc )

      [1] "newid"      "sex"      "dob"
      [4] "dodm"      "dodd"      "doin"
      [7] "dox"      "dolab"      "alat"
     [10] "abdominalomfang" "alkohol"      "asat"
     [13] "bas"      "cpeptid"      "duna"
     [16] "d.vitamin" "diastoliskepj" "diastoliskhjemme"
     [19] "diurese"      "dualb"      "fe"
     [22] "feratio"      "gad"      "gfr"
     [25] "gluc"      "hdl"      "hvilepuls"
     [28] "kalium"      "ldl"      "natrium"
     [31] "pcreatinin" "systoliskepj" "systoliskhjemme"
     [34] "tchol"      "tsh"      "trans"

```

```

[37] "triglycerid"      "ualbcrea"      "vldl"
[40] ".merge"          "b12"           "blodglukose"
[43] "bmi"             "civilstandskode" "diastolisk.arm"
[46] "egfr"            "haemoglobin"    "hba1c"
[49] "height"          "middelblodglukoseepj" "motion"
[52] "rygning"         "sygdom.diabetes" "systolisk.arm"
[55] "weight"

```

```
> addmargins( table(table(sdc$newid)) )
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
357	390	356	290	263	305	336	366	378	373	349	339	314	308	305	
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
292	261	232	257	261	261	254	224	203	226	207	178	197	165	171	
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	
179	162	158	144	136	126	136	128	122	122	125	121	106	106	117	
46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	
121	115	94	119	90	106	97	96	76	106	91	97	96	82	85	
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	
89	102	87	100	99	91	71	81	83	84	79	85	87	68	79	
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	
78	74	67	86	67	75	61	58	63	65	49	52	58	44	42	
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	
37	39	32	30	27	31	30	23	32	18	20	24	28	22	13	
106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	
20	19	16	10	10	9	18	12	6	7	13	7	7	15	7	
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	
3	10	5	9	7	4	8	5	8	8	8	2	11	1	2	
136	137	138	139	140	141	142	143	144	145	146	147	148	149	151	
5	2	4	5	5	2	2	2	4	1	1	5	3	3	2	
152	153	155	158	159	160	161	162	163	164	165	166	171	172	173	
2	2	1	3	1	1	1	1	1	1	1	1	3	1	2	
174	175	181	185	187	190	200	202	204	208	211	215	Sum			
2	2	1	1	1	1	1	1	1	1	2	1	14857			

3.1.2 Renal endpoints

Furthermore we want the dates of CKD and ESRD as well as indicators for any of these two endpoints, and additionally also dates and indicators for micro- and macro-albuminuria.

To this end the following outcome variables are of interest; albumin, creatinine and GFR/eGFR.

OBS! OBS! Skal creatinin (pcreatinin) bruges i definitionen af ESRD?

We will be using both `egfr` and `gfr`, as well as `ualbcrea` and `dualb` in the definitions of the renal endpoints:

```

> with( sdc, table(eGFR=!is.na(egfr),GFR=!is.na(gfr)) )
      GFR
eGFR    FALSE    TRUE
FALSE 480716    8458
TRUE   26142      0

> with( sdc, table(ucr=!is.na(ualbcrea),dualb=!is.na(dualb)) )
      dualb
ucr    FALSE    TRUE
FALSE 364541    51935
TRUE   98277     563

```

We see that for some visits, both `ualbcrea` and `dualb` are known, but the GFR-variables are mutually exclusive.

Currently we have not linked data with that from the national patient register, so the definition of ESRD is entirely based on GFR being defined as (e)GFR < 15 but for

completeness we define the kidney state by cutpoints usually employed where $GFR < 60$ is defined as CKD (Chronic Kidney Disease) a.k.a. DKD (Diabetic Kidney Disease) and $GFR < 15$ as ESRD. We see that there is a certain small misclassification in using concurrent measurements of `dualb` and `ualbcrea`:

```
> with( sdc, table( max = cut( pmax(dualb,ualbcrea,na.rm=TRUE),
+                               breaks=c(0,30,300,Inf),
+                               right=FALSE ),
+                               min = cut( pmin(dualb,ualbcrea,na.rm=TRUE),
+                                           breaks=c(0,30,300,Inf),
+                                           right=FALSE )) )
```

	min			
max	[0,30)	[30,300)	[300,Inf)	
[0,30)	96686	0	0	
[30,300)	45	38625	0	
[300,Inf)	1	32	15386	

With this in mind we can define the desired variables from `gfr` and `egfr` and the albumin variabes `dualb` and `ualbcrea`:

```
> sdc <- transform( sdc, GFR = pmin( egfr, gfr, na.rm=TRUE ),
+                     ren.st = cut( pmin( egfr, gfr, na.rm=TRUE ),
+                                   breaks=c(0,15,30,45,60,90,Inf),
+                                   include.lowest=TRUE ),
+                     alb.st = cut( pmax(dualb,ualbcrea,na.rm=TRUE),
+                                   breaks=c(0,30,300,Inf),
+                                   right=FALSE ) )
> non.miss <- function(x) sum(x[-length(x)])
> with( sdc, addmargins( table( ren.st, alb.st, useNA="ifany" ),
+                               FUN=list(list(sum,non.miss),list(sum,non.miss)),
+                               quiet=TRUE ) ) [c(1:6,9,7,8),c(1:3,6,4,5)]
```

	alb.st					
ren.st	[0,30)	[30,300)	[300,Inf)	non.miss	<NA>	sum
[0,15]	15	34	89	138	151	289
(15,30]	231	385	427	1043	875	1918
(30,45]	630	728	644	2002	1736	3738
(45,60]	1097	936	558	2591	2118	4709
(60,90]	7835	2634	1174	11643	9606	21249
(90,Inf]	142	298	237	677	2020	2697
non.miss	9950	5015	3129	18094	16506	34600
<NA>	86736	33655	12290	132681	348035	480716
sum	96686	38670	15419	150775	364541	515316

Thus, based on the GFR-measurements from SDC we see that we have a mere 289 values in the ESRD state. However, the table is a table of visits (lab-dates) so there might be fewer *persons* with ESRD.

We now define CKD and ESRD and the dates of these, by taking the earliest date where the criterion is met:

```
> minna <- function(x) ifelse((mx<-min(x,na.rm=TRUE))<Inf,mx,NA)
> sdc$doESRD <- sdc$dolab
> sdc$doESRD[as.integer(sdc$ren.st)>1|is.na(sdc$ren.st)] <- NA
> sdc$doESRD <- with( sdc, ave( doESRD, newid, FUN=minna ) )
> sdc$ESRD <- !is.na( sdc$doESRD )
> sdc$doCKD <- sdc$dolab
> sdc$doCKD[as.integer(sdc$ren.st)>4|is.na(sdc$ren.st)] <- NA
> sdc$doCKD <- with( sdc, ave( doCKD, newid, FUN=minna ) )
> sdc$CKD <- !is.na( sdc$doCKD )
```

3.2 Saving data

As always there are a few fishy datapoints:

```
> subset( sdc, dolab<1990 )
      newid sex dob dodm dodd doin dox      dolab alat abdominalomfang      alkohol
463614  4516  M  NA   NA   NA   NA   NA 1913.218   NA                      NA <14 Genstande/uge
      asat bas cpeptid duna d.vitamin diastoliskepjd diastoliskhjemme diurese dualb fe
463614   NA  NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
      feratio gad gfr gluc hdl hvilepuls kalium ldl natrium pcreatinin systoliskepjd
463614   NA  NA  NA   NA  NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
      systoliskhjemme tchol tsh trans triglycerid ualbcrea vldl .merge b12 blodglukose
463614   NA   NA  NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
      bmi civilstandskode diastolisk.arm egfr haemoglobin hba1c height
463614   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
      middelblodglukoseepjd motion rygning sygdom.diabetes systolisk.arm weight
463614   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
      GFR ren.st alb.st doESRD ESRD doCKD CKD
463614   NA  <NA>  <NA>   NA FALSE   NA FALSE
```

```
> sdc <- subset( sdc, dolab>1990 )
```

Finally we can save the dataset and the table of measurement units for the variables:

```
> addmargins( table(table(sdc$newid)) )
      1      2      3      4      5      6      7      8      9     10     11     12     13     14     15
357   390   356   290   263   305   336   366   378   373   349   339   314   308   305
16     17     18     19     20     21     22     23     24     25     26     27     28     29     30
292   261   232   257   261   261   254   224   204   225   207   178   197   165   171
31     32     33     34     35     36     37     38     39     40     41     42     43     44     45
179   162   158   144   136   126   136   128   122   122   125   121   106   106   117
46     47     48     49     50     51     52     53     54     55     56     57     58     59     60
121   115   94    119   90    106   97    96    76    106   91    97    96    82    85
61     62     63     64     65     66     67     68     69     70     71     72     73     74     75
89    102     87    100   99    91    71    81    83    84    79    85    87    68    79
76     77     78     79     80     81     82     83     84     85     86     87     88     89     90
78     74     67     86     67     75     61     58     63     65     49     52     58     44     42
91     92     93     94     95     96     97     98     99    100    101    102    103    104    105
37     39     32     30     27     31     30     23     32     18     20     24     28     22     13
106   107   108   109   110   111   112   113   114   115   116   117   118   119   120
20     19     16     10     10     9     18     12     6     7     13     7     7     15     7
121   122   123   124   125   126   127   128   129   130   131   132   133   134   135
3      10     5     9     7     4     8     5     8     8     8     2     11     1     2
136   137   138   139   140   141   142   143   144   145   146   147   148   149   151
5      2     4     5     5     2     2     2     4     1     1     5     3     3     2
152   153   155   158   159   160   161   162   163   164   165   166   171   172   173
2      2     1     3     1     1     1     1     1     1     1     3     1     1     2
174   175   181   185   187   190   200   202   204   208   211   215   Sum
2      2     1     1     1     1     1     1     1     1     2     1 14857
```

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

Chapter 4

Descriptives - DK

4.1 Date variables

First we provide an overview of the date variables paired, so that we can see to what extent they are in the wrong order. We only plot for 5000 records instead of all 500,000, in order to keep the size of the graph manageable:

```
> load( file="./data/sdc.Rda" )
> ( dn <- grep("do",names(sdc)) )
[1] 3 4 5 6 7 8 10 53 59 61
> names(sdc)[dn]
[1] "dob"      "dodm"      "dodd"      "doin"
[5] "dox"      "dolab"     "abdominalomfang" "sygdom.diabetes"
[9] "doESRD"   "doCKD"
> par( bty="o" )
> pairs( sdc[sample(1:nrow(sdc),5000),dn[c(1,2,4,6,10,9,5,3)]], gap=0, pch=16, cex=0.2,
+       panel=function(x,y,...) {points(x,y,...);abline(0,1,col="red")})
```

4.2 Data overview

First we just show the first few records of the data frame:

```
> head( sdc )
```

	newid	sex	dob	dodm	dodd	doin	dox	dolab	alat	abdominalomfang			
1	1	M	1953.572	1993	2014.579	2002.691	2003.231	2002.691	NA	NA			
2	1	M	1953.572	1993	2014.579	2002.691	2003.231	2002.787	NA	NA			
3	1	M	1953.572	1993	2014.579	2002.691	2003.231	2002.831	NA	NA			
4	1	M	1953.572	1993	2014.579	2002.691	2003.231	2002.886	NA	NA			
5	1	M	1953.572	1993	2014.579	2002.691	2003.231	2003.020	NA	NA			
6	1	M	1953.572	1993	2014.579	2002.691	2003.231	2003.080	NA	NA			
			alkohol	asat	bas	cpeptid	duna	d.vitamin	diastoliskep	j diastoliskhemme	diurese	dualb	fe
1			NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
2			28	177	787	NA	NA	93	NA	NA	NA	18	
3			NA	NA	NA	NA	NA	95	NA	NA	NA	NA	
4			24	159	NA	NA	NA	NA	NA	NA	NA	NA	
5			NA	NA	NA	NA	NA	85	NA	NA	NA	NA	
6			27	152	NA	NA	NA	92	NA	NA	NA	NA	
			feratio	gad	gfr	gluc	hdl	hvilepuls	kalium	ldl	natrium	pcreatinin	systoliskep
1			NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
2			22	NA	NA	NA	1.30	NA	3.8	3.2	136	82	143
3			NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	150
4			NA	NA	NA	NA	1.25	NA	NA	2.6	NA	NA	NA

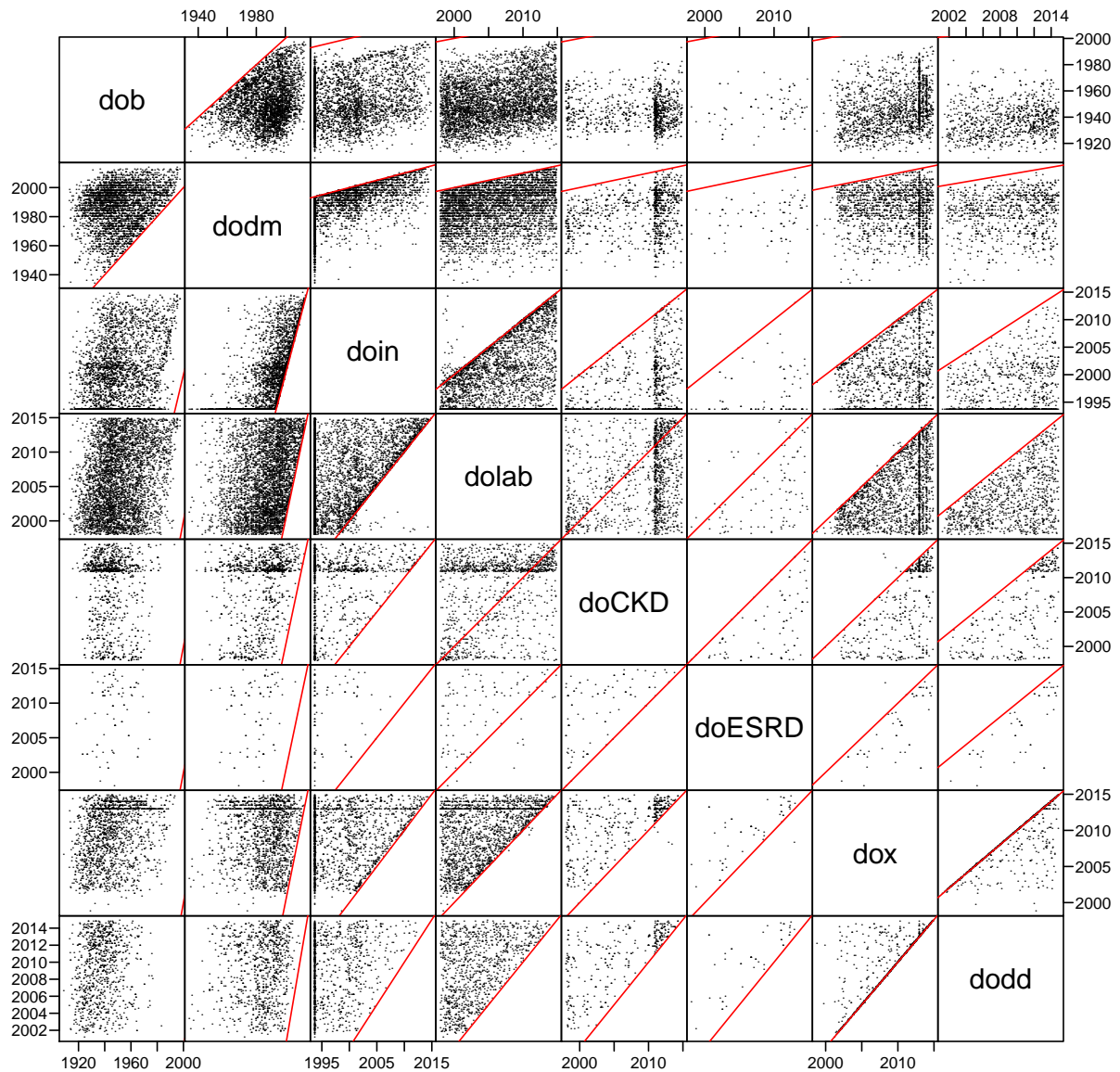


Figure 4.1: Date variables in the SDC clinical dataset. The red lines are the identity lines, meaning that all points should be on the same side of the lines since the date variables are listed in ascending order.

```

5      NA NA NA NA NA      NA      NA NA      NA      NA      140
6      NA NA NA NA 1.21     NA      3.9 NA      140      87      144
systolikhjemme tchol tsh trans triglycerid ualbcrea vldl .merge b12 blodglukose bmi
1      NA      NA NA      NA      NA      NA      4      NA      1 NA      NA      30
2      NA      5.2 2      41      1.45      NA      0.7      1 NA      NA      NA
3      NA      NA NA      NA      NA      NA      NA      NA      1 NA      NA      NA
4      NA      4.2 NA      NA      0.85      NA      NA      0.4      1 NA      NA      NA
5      NA      NA NA      NA      NA      NA      NA      NA      1 NA      NA      NA
6      NA      3.7 NA      NA      1.06      4      0.5      1 NA      NA      NA      NA
civilstandskode diastolisk.arm egfr haemoglobin hba1c height middelblodglukoseepj
1      D\xf8d      NA NA      NA      63      1.79      NA
2      D\xf8d      NA NA      10.2 57      NA      NA
3      D\xf8d      NA NA      NA      NA      NA      NA

```

4	D\xf8d	NA	NA	NA	53	NA		NA
5	D\xf8d	NA	NA	NA	51	NA		NA
6	D\xf8d	NA	NA	NA	48	NA		NA
	motion rygning sygdom.diabetes	systolisk.arm	weight	GFR	ren.st	alb.st	doESRD	ESRD
1	DE11	NA	97.3	NA	<NA>	[0,30)	NA	FALSE
2	DE11	NA	99.7	NA	<NA>	<NA>	NA	FALSE
3	DE11	NA	96.5	NA	<NA>	<NA>	NA	FALSE
4	DE11	NA	94.3	NA	<NA>	<NA>	NA	FALSE
5	DE11	NA	NA	NA	<NA>	<NA>	NA	FALSE
6	DE11	NA	96.6	NA	<NA>	[0,30)	NA	FALSE
	doCKD	CKD						
1	NA	FALSE						
2	NA	FALSE						
3	NA	FALSE						
4	NA	FALSE						
5	NA	FALSE						
6	NA	FALSE						

We then provide an overview over how many visits (well,dates) there are per person:

```
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6, bty="n" )
> tt <- with( sdc, table(table(newid)) )
> plot( as.numeric(names(tt)), tt, type="h", lwd=3, xaxs="i", xlim=c(0,150.5),
+       xlab="No. of dates", ylab="No. of persons", yaxt="n" )
> abline( v=5.5, col="red" )
> axis( side=2 )
```

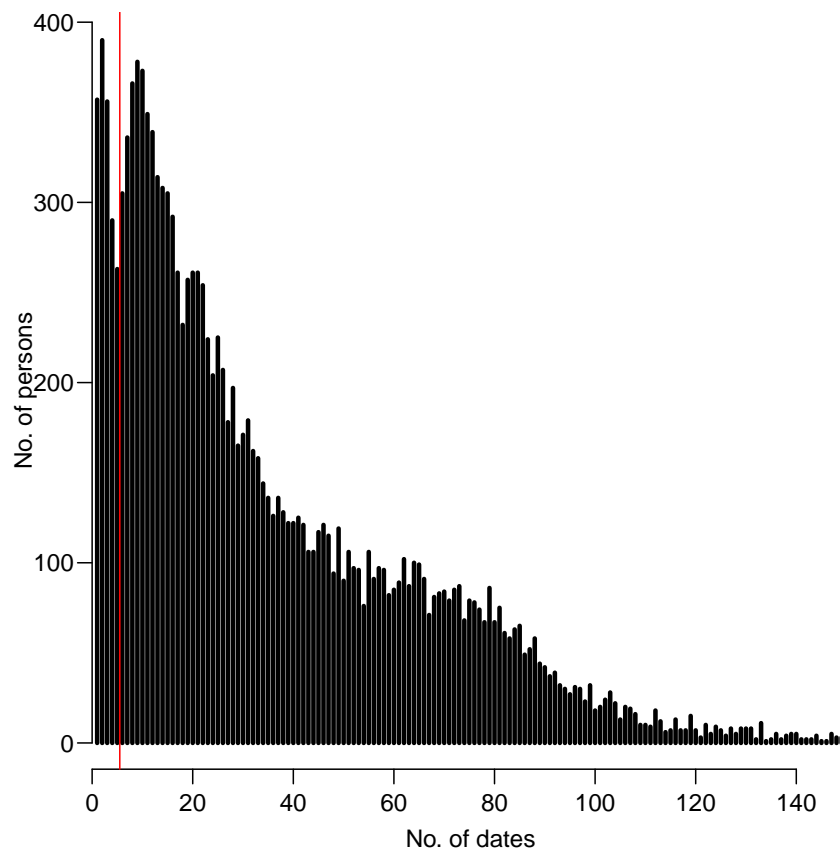


Figure 4.2: Number of recorded dates for patients in the extract — that is patients distributed according to number of records in the dataset. The vertical line is between 5 and 6 visits.

It might however be more illuminating to see how many visits with recordings of central clinical variables that are available:

```
> data.frame( 1:ncol(sdc), names(sdc) )
```

	X1.ncol.sdc.	names.sdc.
1	1	newid
2	2	sex
3	3	dob
4	4	dodm
5	5	dodd
6	6	doin
7	7	dox
8	8	dolab
9	9	alat
10	10	abdominalomfang
11	11	alkohol
12	12	asat
13	13	bas
14	14	cpeptid
15	15	duna
16	16	d.vitamin
17	17	diastoliskepj
18	18	diastoliskhemme
19	19	diurese
20	20	dualb
21	21	fe
22	22	feratio
23	23	gad
24	24	gfr
25	25	gluc
26	26	hdl
27	27	hvilepuls
28	28	kalium
29	29	ldl
30	30	natrium
31	31	pcreatinin
32	32	systoliskepj
33	33	systoliskhemme
34	34	tchol
35	35	tsh
36	36	trans
37	37	triglycerid
38	38	ualbcrea
39	39	vldl
40	40	.merge
41	41	b12
42	42	blodglukose
43	43	bmi
44	44	civilstandskode
45	45	diastolisk.arm
46	46	egfr
47	47	haemoglobin
48	48	hba1c
49	49	height
50	50	middelblodglukoseepj
51	51	motion
52	52	rygning
53	53	sygdom.diabetes
54	54	systolisk.arm
55	55	weight
56	56	GFR
57	57	ren.st
58	58	alb.st
59	59	doESRD
60	60	ESRD
61	61	doCKD
62	62	CKD

```
> wh <- c(
+ "abdominalomfang",
```

```

+ "bmi",
+ "blodglukose",
+ "middelblodglukoseepj",
+ "gluc",
+ "hba1c",
+
+ "diastoliskepj",
+ "diastoliskhjemme",
+ "diastolisk.arm",
+ "systoliskepj",
+ "systoliskhjemme",
+ "systolisk.arm",
+
+ "tchol",
+ "hdl",
+ "ldl",
+ "vldl",
+ "triglycerid",
+ "cpeptid",
+
+ "gfr",
+ "egfr",
+ "dualb",
+ "ualbcrea")
> nval <- NArray( list( var=wh, c("no. pers","median no. values") ) )

> n.vis <-
+ function(vn)
+ {
+ df <- sdc[,c("newid",vn)]
+ df <- df[complete.cases(df),]
+ tt <- table(table(df$newid))
+ plot( as.numeric(names(tt)), tt, type="h", lwd=3, lend=1,
+       xaxs="i", xaxt="n", xlim=c(0,35),
+       yaxs="i", yaxt="n", ylim=c(0,2900) )
+ text( 10, 2000, vn, cex=1.0, font=2, adj=c(0,0) )
+ # Finally return number of persons with at least one valid value of
+ # the variable:
+ c( sum(tt), median( rep(as.numeric(names(tt)),tt) ) )
+ }
> par( mfrow=c(4,6), oma=c(4,4,1,1), mar=c(0,0,0,0), mgp=c(3,1,0)/1.6, bty="n" )
> for( i in 1:length(wh) )
+ {
+   nval[i,] <- n.vis( wh[i] )
+   if( i %in% (0:3*6+1) ) axis(side=2)
+   if( i > 16 ) axis(side=1)
+ }
> mtext( "No. of persons", side=2, line=2, outer=TRUE, las=0 )
> mtext( "No. of dates" , side=1, line=2, outer=TRUE )
> round( nval, 1 )

var                no. pers median no. values
abdominalomfang      714          1
bmi                 12993         1
blodglukose          591          1
middelblodglukoseepj  879          1
gluc                10943         12
hba1c               14810         15
diastoliskepj        4382          6
diastoliskhjemme     1987          1
diastolisk.arm       2749          2
systoliskepj         4382          6
systoliskhjemme      1987          1
systolisk.arm        2188          2
tchol               14618          4
hdl                 14598          4

```

ldl	10976	3
vldl	13725	4
triglycerid	14586	4
cpeptid	11584	1
gfr	1406	4
egfr	8015	2
dualb	6274	6
ualbcrea	12744	5

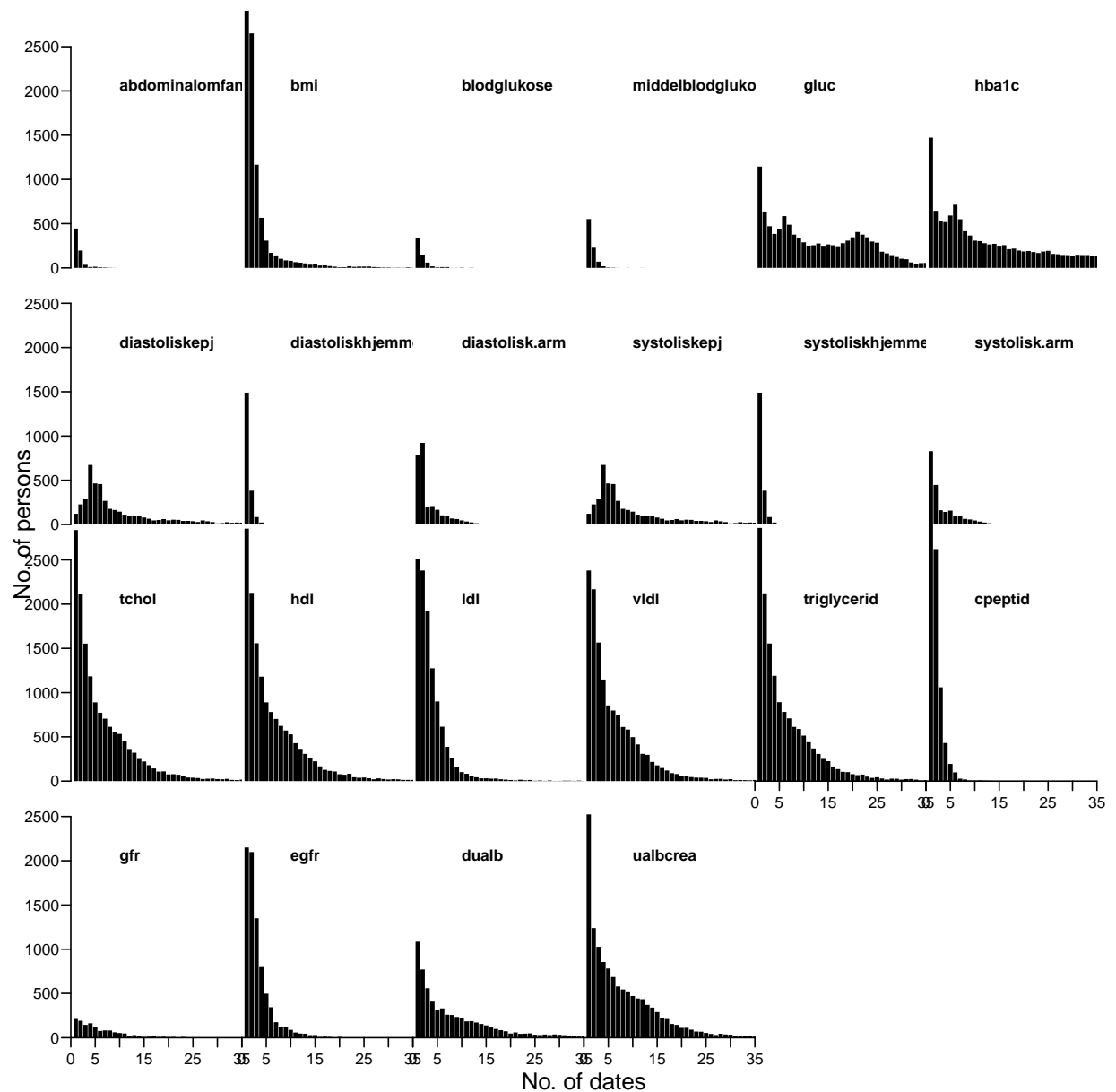


Figure 4.3: Number of recorded dates with valid values of specific measurements for patients in the extract. The total area (the sum of the bar heights) is the number of persons with at least one valid measurement of the variable.

Chapter 5

Analysis - DK

```
> library( Epi )
> load( file="./data/sdc.Rda" )
> sdc[1:10,c(1:8,56,57,59,60)]
```

	newid	sex	dob	dodm	dodd	doin	dox	dolab	GFR	ren.st	doESRD	ESRD
1	1	M	1953.572	1993	2014.579	2002.691	2003.231	2002.691	NA	<NA>	NA	FALSE
2	1	M	1953.572	1993	2014.579	2002.691	2003.231	2002.787	NA	<NA>	NA	FALSE
3	1	M	1953.572	1993	2014.579	2002.691	2003.231	2002.831	NA	<NA>	NA	FALSE
4	1	M	1953.572	1993	2014.579	2002.691	2003.231	2002.886	NA	<NA>	NA	FALSE
5	1	M	1953.572	1993	2014.579	2002.691	2003.231	2003.020	NA	<NA>	NA	FALSE
6	1	M	1953.572	1993	2014.579	2002.691	2003.231	2003.080	NA	<NA>	NA	FALSE
7	1	M	1953.572	1993	2014.579	2002.691	2003.231	2003.209	NA	<NA>	NA	FALSE
9	3	M	1919.461	1976	2008.827	1993.755	2008.827	1998.010	NA	<NA>	NA	FALSE
10	3	M	1919.461	1976	2008.827	1993.755	2008.827	1998.012	NA	<NA>	NA	FALSE
11	3	M	1919.461	1976	2008.827	1993.755	2008.827	1998.344	NA	<NA>	NA	FALSE

```
> sdc[sdc$newid==8,c(1:8,56,57,59,60)]
```

	newid	sex	dob	dodm	dodd	doin	dox	dolab	GFR	ren.st	doESRD	ESRD
189	8	F	1936.701	1985	2002.226	1993.755	2002.226	1998.015	NA	<NA>	1998.779	TRUE
190	8	F	1936.701	1985	2002.226	1993.755	2002.226	1998.070	NA	<NA>	1998.779	TRUE
191	8	F	1936.701	1985	2002.226	1993.755	2002.226	1998.223	NA	<NA>	1998.779	TRUE
192	8	F	1936.701	1985	2002.226	1993.755	2002.226	1998.245	NA	<NA>	1998.779	TRUE
193	8	F	1936.701	1985	2002.226	1993.755	2002.226	1998.439	NA	<NA>	1998.779	TRUE
194	8	F	1936.701	1985	2002.226	1993.755	2002.226	1998.445	NA	<NA>	1998.779	TRUE
195	8	F	1936.701	1985	2002.226	1993.755	2002.226	1998.760	NA	<NA>	1998.779	TRUE
196	8	F	1936.701	1985	2002.226	1993.755	2002.226	1998.779	15	[0,15]	1998.779	TRUE
197	8	F	1936.701	1985	2002.226	1993.755	2002.226	1999.190	NA	<NA>	1998.779	TRUE
198	8	F	1936.701	1985	2002.226	1993.755	2002.226	1999.603	NA	<NA>	1998.779	TRUE
199	8	F	1936.701	1985	2002.226	1993.755	2002.226	1999.814	NA	<NA>	1998.779	TRUE
200	8	F	1936.701	1985	2002.226	1993.755	2002.226	2000.129	NA	<NA>	1998.779	TRUE
201	8	F	1936.701	1985	2002.226	1993.755	2002.226	2000.140	NA	<NA>	1998.779	TRUE
202	8	F	1936.701	1985	2002.226	1993.755	2002.226	2000.397	NA	<NA>	1998.779	TRUE
203	8	F	1936.701	1985	2002.226	1993.755	2002.226	2000.411	NA	<NA>	1998.779	TRUE
204	8	F	1936.701	1985	2002.226	1993.755	2002.226	2000.794	NA	<NA>	1998.779	TRUE
205	8	F	1936.701	1985	2002.226	1993.755	2002.226	2000.890	NA	<NA>	1998.779	TRUE
206	8	F	1936.701	1985	2002.226	1993.755	2002.226	2001.109	NA	<NA>	1998.779	TRUE
207	8	F	1936.701	1985	2002.226	1993.755	2002.226	2001.314	NA	<NA>	1998.779	TRUE
208	8	F	1936.701	1985	2002.226	1993.755	2002.226	2001.637	NA	<NA>	1998.779	TRUE
209	8	F	1936.701	1985	2002.226	1993.755	2002.226	2001.886	NA	<NA>	1998.779	TRUE

5.1 Outcome data

We need to fish out all records with GFR measurements, and subsequently persons with at least two measurements

```

> sdcR <- subset( sdc, !is.na(GFR) & dolab <= doESRD )
> dim( sdcR )
[1] 1244 62

> addmargins( with( sdcR, table(table(newid)) ) )

  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 24 25 26
30 27 14 10 18 10 6  8  9  5  8  6  5  1  3  1  3  1  2  1  1  1  1  1  3

> # Fishing out persons with at least 2 measurements
> tt <- table(sdcR$newid)
> over1 <- names( tt[tt>1] )
> sdcR <- subset( sdcR, newid %in% over1 )
> addmargins( with( sdcR, table(table(newid)) ) )

  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 24 25 26 27
27 14 10 18 10 6  8  9  5  8  6  5  1  3  1  3  1  2  1  1  1  1  1  3  1

```

Since we are going to analyse GFR as a function of time before ESRD, we will need the time to ESRD, `ttESD` as a separate variable:

```

> sdcR <- transform( sdcR, ttESRD = dolab-doESRD )
> hist( sdcR$ttESRD, col="black", breaks=seq(-20,0,0.5) )

```

5.2 Trajectory analyses with latent classes

The following illustrates the use of the `lcmm` package to fit random effects spline models to the trajectories of those that end with ESRD. Thus we are conditioning on the end state renal disease outcome (ESRD), and model how the trajectories of GFR is in these individuals. The purpose of this is to try to identify different *shapes* of GFR-decline up to ESRD.

So we first subset the data to those persons who actually get ESRD. Since `lcmm` does not accept the usual model formulae we must explicitly construct the columns of the spline basis (note that the `Ns` is a wrapper from the `Epi` package to simplify definition of natural splines). Also note that `detrend` is a function from `Epi` that makes a projection of the columns of the spline basis on the orthogonal complement to the constant plus the time variable. The resulting columns are thus the non-linear effects of the time variable, in the case `ttESRD`:

```

> library( lcmm )
> library( splines )
> esrd <- subset( sdcR, ESRD )
> esrd$age <- esrd$dolab - esrd$dob
> with( esrd, round( quantile( ttESRD, 0:10/10 ), 1 ) )

  0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
-16.1 -11.8 -9.7 -7.4 -5.5 -3.8 -2.6 -1.7 -1.0  0.0  0.0

> ( kn <- seq(-15,0,,5) )
[1] -15.00 -11.25 -7.50 -3.75  0.00

> MM <- Ns( esrd$ttESRD, knots=kn )
> MM <- detrend( MM, esrd$ttESRD )
> ( colnames(MM) <- paste("x",colnames(MM),sep="" ) )
[1] "x1" "x2" "x3"

> esrd <- cbind( esrd, MM )
> head( MM )

```

	x1	x2	x3
[1,]	-0.268399677	-0.1677629	0.1662526
[2,]	-0.129230404	-0.1982983	0.1940913
[3,]	-0.030780993	-0.1996719	0.1684513
[4,]	-0.030780993	-0.1996719	0.1684513
[5,]	-0.030780993	-0.1996719	0.1684513
[6,]	0.002985959	-0.1977835	0.1554466

We have now set up data to fit the model; the columns `x1`, `x2`, `x3` and `x4` represent the non-linear effects of time before ESRD. This means that that coefficient to `ttESRD` represents the *average* time trend in eGFR over time. Thus it is possible to compare the size of this with the sd of the random effects (that is the between person variation in slopes). The argument `nwg=TRUE` scales the random-effect covariance between classes:

```
> system.time(
+ fitspl <- hlme( GFR ~ x1 + x2 + x3 + ttESRD + age + sex,
+               mixture = ~ x1 + x2 + x3 + ttESRD,
+               random = ~ ttESRD,
+               subject = 'newid', ng = 3, nwg=TRUE, data = esrd ) )
Be patient, hlme is running ...
The program took 52.08 seconds
  user system elapsed
 52.090   0.001  52.084

> fitspl

Heterogenous linear mixed model
  fitted by maximum likelihood method

hlme(fixed = GFR ~ x1 + x2 + x3 + ttESRD + age + sex, mixture = ~x1 +
      x2 + x3 + ttESRD, random = ~ttESRD, subject = "newid", ng = 3,
      nwg = TRUE, data = esrd)

Statistical Model:
  Dataset: esrd
  Number of subjects: 148
  Number of observations: 1214
  Number of latent classes: 3
  Number of parameters: 25

Iteration process:
  Convergence criteria satisfied
  Number of iterations: 24
  Convergence criteria: parameters= 5.1e-08
                      : likelihood= 8.8e-11
                      : second derivatives= 1.7e-15

Goodness-of-fit statistics:
  maximum log-likelihood: -4238.2
  AIC: 8526.41
  BIC: 8601.34

> summary( fitspl )

Heterogenous linear mixed model
  fitted by maximum likelihood method

hlme(fixed = GFR ~ x1 + x2 + x3 + ttESRD + age + sex, mixture = ~x1 +
      x2 + x3 + ttESRD, random = ~ttESRD, subject = "newid", ng = 3,
      nwg = TRUE, data = esrd)

Statistical Model:
  Dataset: esrd
  Number of subjects: 148
  Number of observations: 1214
  Number of latent classes: 3
```

```

Number of parameters: 25

Iteration process:
  Convergence criteria satisfied
  Number of iterations: 24
  Convergence criteria: parameters= 5.1e-08
                      : likelihood= 8.8e-11
                      : second derivatives= 1.7e-15

Goodness-of-fit statistics:
  maximum log-likelihood: -4238.2
  AIC: 8526.41
  BIC: 8601.34

Maximum Likelihood Estimates:

Fixed effects in the class-membership model:
(the class of reference is the last class)

      coef      Se    Wald p-value
intercept class1  1.48259 0.27612  5.369 0.00000
intercept class2 -0.30545 0.47927 -0.637 0.52392

Fixed effects in the longitudinal model:

      coef      Se    Wald p-value
intercept class1 14.47848 1.99227  7.267 0.00000
intercept class2 15.54789 1.95576  7.950 0.00000
intercept class3 26.53283 2.67008  9.937 0.00000
x1 class1        -4.02721 1.81554 -2.218 0.02654
x1 class2       -39.28702 6.05635 -6.487 0.00000
x1 class3         3.77527 6.05364  0.624 0.53287
x2 class1        -2.05853 1.53305 -1.343 0.17935
x2 class2        16.66596 4.22654  3.943 0.00008
x2 class3        30.43634 6.54339  4.651 0.00000
x3 class1        -7.71538 4.13201 -1.867 0.06187
x3 class2        43.85254 15.06561  2.911 0.00361
x3 class3         3.91147 17.12475  0.228 0.81933
ttESRD class1    -4.61879 0.27289 -16.925 0.00000
ttESRD class2    -6.18769 0.93584 -6.612 0.00000
ttESRD class3   -10.03379 1.74118 -5.763 0.00000
age              -0.01017 0.02996 -0.340 0.73419
sexF             -1.41796 0.74607 -1.901 0.05736

Variance-covariance matrix of the random-effects:
      intercept  ttESRD
intercept 17.84505
ttESRD    10.28227 43.57987

      coef      se
Proportional coefficient class1 0.3106424 0.06732407
Proportional coefficient class2 0.3318239 0.15959066
Residual standard error:      6.2460811 0.14336491

```

Once we fitted the model we can have a look at how the posterior probabilities of being in the assigned classes look:

```

> postprob( fitspl )

Posterior classification:
  class1 class2 class3
N      111  15.00  22.00
%       75  10.14  14.86

```

Posterior classification table:

```
--> mean of posterior probabilities in each class
      prob1 prob2 prob3
class1 0.9329 0.0451 0.0219
class2 0.1260 0.7790 0.0950
class3 0.0317 0.0483 0.9200
```

Posterior probabilities above a threshold (%):

```
      class1 class2 class3
prob>0.7  95.50  60.00  90.91
prob>0.8  88.29  53.33  81.82
prob>0.9  78.38  40.00  81.82
```

```
> names( fitspl )
```

```
 [1] "ns"      "ng"      "idea0"   "idprob0" "idg0"    "idcor0"  "loglik"  "best"
[10] "gconv"   "conv"    "call"    "niter"   "dataset" "N"       "idiag"   "pred"
[19] "predRE"  "Xnames"  "Xnames2" "cholesky" "na.action"
```

```
> str( ppr <- fitspl$pprob )
```

```
'data.frame':      148 obs. of  5 variables:
 $ newid: num  60 222 283 406 411 440 507 530 670 709 ...
 $ class: int   1  1  1  1  2  3  2  1  2  3 ...
 $ prob1: num   1  1  0.918 0.598 0.243 ...
 $ prob2: num  8.42e-14 3.19e-12 3.40e-02 3.63e-01 6.09e-01 ...
 $ prob3: num  6.68e-08 1.45e-04 4.85e-02 3.86e-02 1.48e-01 ...
```

```
> ncl <- table( ppr$class )
```

```
> clr <- c("red","black","blue","limegreen")
```

```
> par( mfrow=c(1,3), mar=c(3,0,1,1), oma=c(0,3,0,0), las=1, bty="n", mgp=c(3,1,0)/1.6 )
```

```
> for( i in 1:3 )
```

```
+ {
+   hist( ppr[ppr$class==i,i+2], breaks=0:50/50,
+         col=clr[i], border=clr[i], ylim=c(0,60), main="",
+         xlab="", yaxt="n", yaxis="i" )
+   if( i==1 ) axis( side=2 )
+   text( 0.1, 50, paste( "Class", i, ": n=", ncl[i] ), font=2,
+         col=clr[i], cex=1.5, adj=0 )
+ }
```

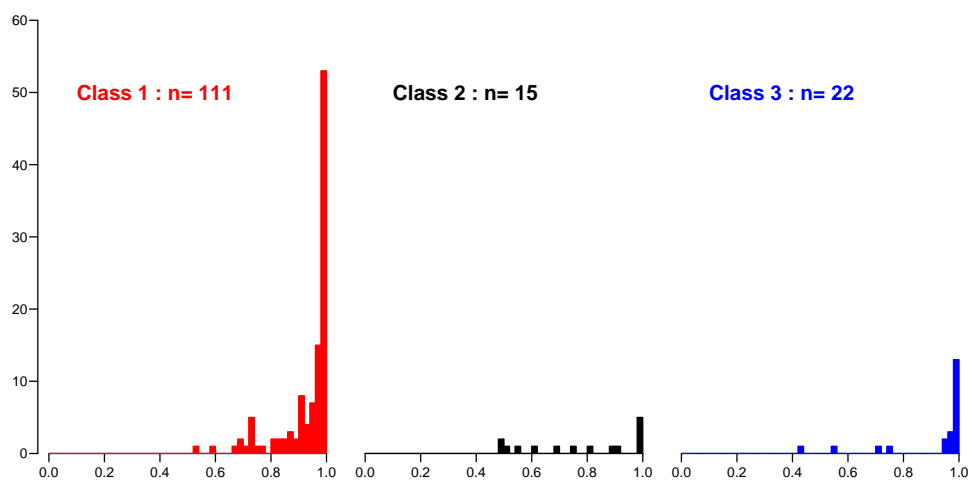


Figure 5.1: Posterior probabilities of class membership for the ESRD cases modelled.

A slightly more informative plot of the posterior probabilities is obtained by looking at the pairwise

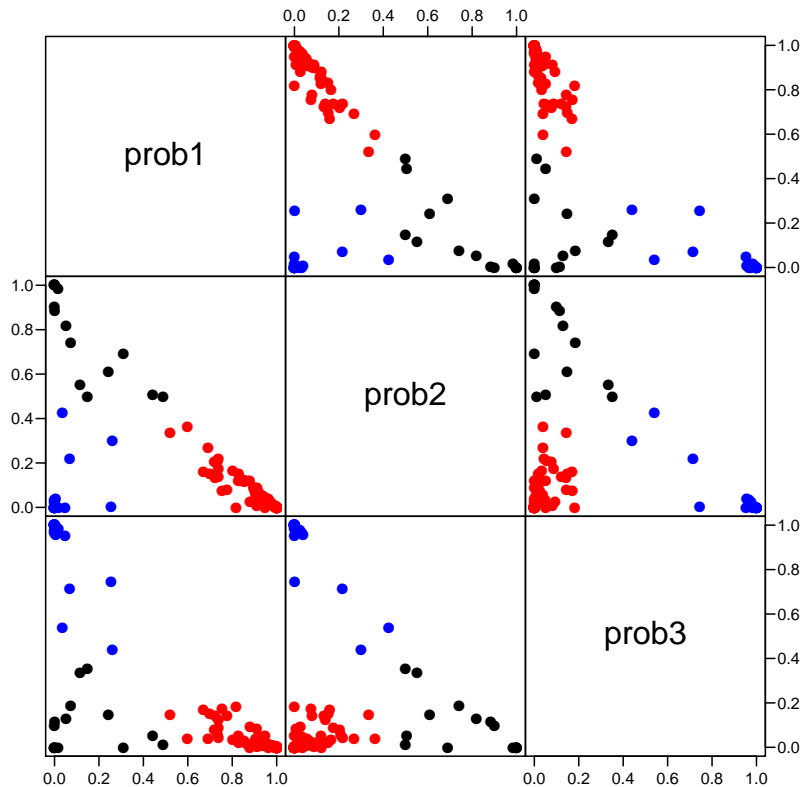


Figure 5.2: *Pairwise posterior probabilities. It is seen that class 1 (red) is not well discriminated from class 2 (black; the largest class)*

```
> par( bty="o")
> pairs( ppr[,2+1:3], pch=16, col=clr[ppr$class], cex=1.5, gap=0 )
```

In order to plot the estimated trajectories we extract a prediction data frame from the analysis data frame. This is necessary because the construction of the de-trended version of the variables depends on data. Incidentally, also the

```
> wh <- match( sort(unique(esrd$ttESRD)), esrd$ttESRD )
> plotdata <- data.frame(1,MM[wh,],esrd$ttESRD[wh],
+                        60+esrd$ttESRD[wh],
+                        sex=factor("M",levels=c("F","M"))) )
> names(plotdata)[-7] <- fitspl$Xnames[-7]
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> outspl <- plot.predict.hlme(fitspl,plotdata,var.time="ttESRD",
+                             legend.loc="topright",col=clr,lwd=4,
+                             ylim=c(0,160), main="")
> str( outspl )
```

```
'data.frame':      865 obs. of  10 variables:
 $ time      : num  -16.1 -15.9 -15.7 -15.6 -15.1 ...
 $ class1    : num   92.7  91.5  90.1  89.5  86.3 ...
 $ class2    : num   110  110  110  110  110 ...
 $ class3    : num   187  185  182  181  175 ...
 $ lower.class1: num   83.1  82.1  80.9  80.5  77.9 ...
 $ lower.class2: num   80.6  81.3  82.1  82.3  83.9 ...
 $ lower.class3: num   131  130  128  127  123 ...
 $ upper.class1: num  102.4 100.9  99.2  98.5  94.8 ...
 $ upper.class2: num   139  139  138  138  137 ...
 $ upper.class3: num   244  240  237  235  227 ...
```

```
> # datasub <- merge(datasub, fitspl$pprob[,1:2], by = "id")
```

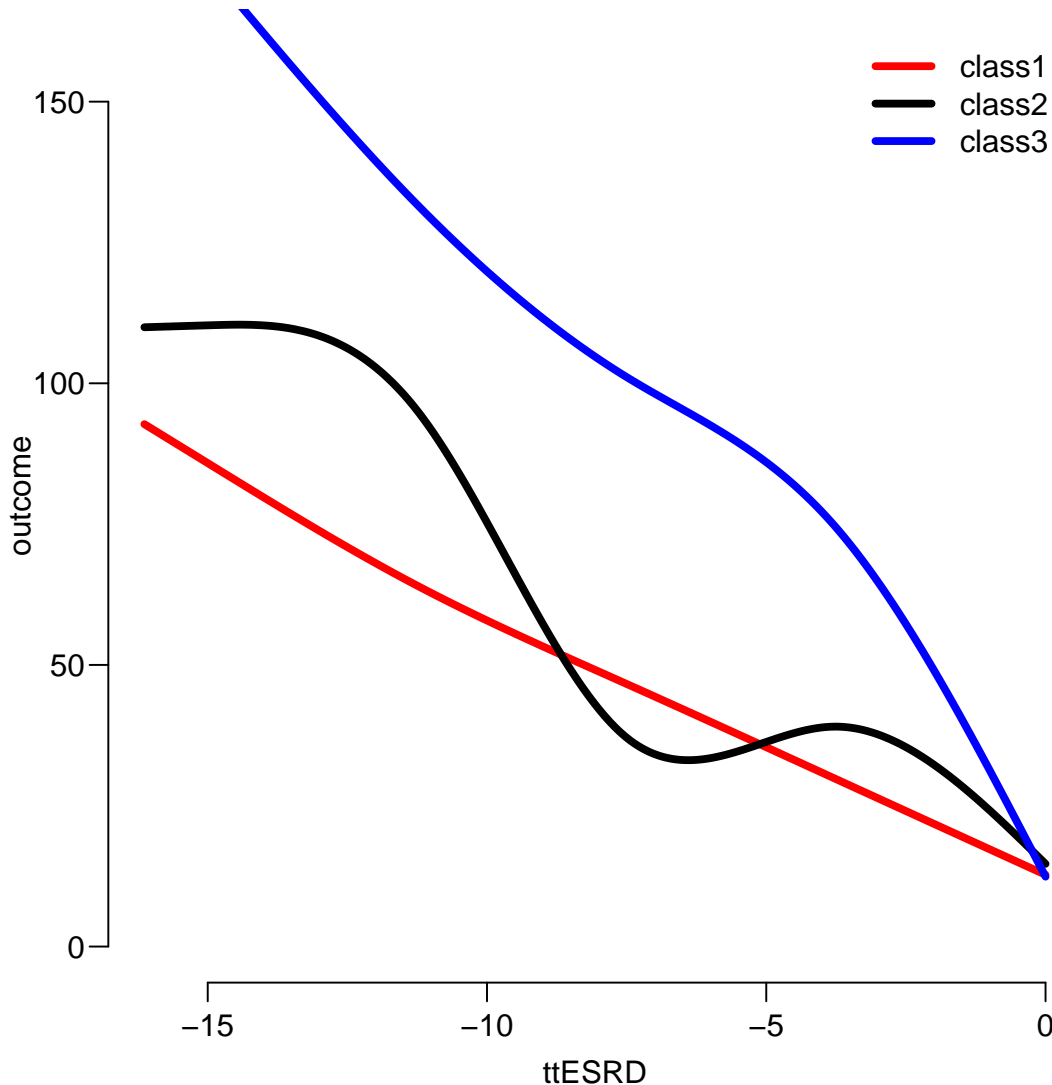


Figure 5.3: Predicted mean trajectories for the three latent classes of persons developing ESRD. Note that the number of persons in the classes as derived are quite unevenly distributed, the classes have 19, 122 and 10 persons in the classes.

5.2.1 2 do next

The latent class trajectory distribution of the persons with event is just a rough guide to the possible patterns; and it would be useful to compare this to the general pattern in GFR among those without event (so far). Hence we will:

- fit separate random effects models to the subgroups identified, allowing us to get estimates of the between-person variation, in order to assess to which extent the variation between persons in different classes is the same.

We shall use the following type of random effects model for GFR-measurement y_{it} :

$$y_{it} = f(t) + \alpha_a + \alpha_s + a_i + b_{it} + e_{it}$$

- extend the models with random slopes and see how these vary between persons.
- plot individual observed trajectories for different classes
- fit a model with linear effect of the time for the patients without ESRD.
- fit a common model for all patients using current age and duration of diabetes as predictors of GFR in addition to sex.
- extend this model with clinical *baseline* variables
- extend this model with *updated* clinical variables