

# Follow-up data with the Epi package

Autumn 2007.

Michael Hills    Retired  
Highgate, London

Martyn Plummer    International Agency for Research on Cancer, Lyon  
[plummer@iarc.fr](mailto:plummer@iarc.fr)

Bendix Carstensen    Steno Diabetes Center, Gentofte, Denmark  
& Department of Biostatistics, University of Copenhagen  
[bxc@steno.dk](mailto:bxc@steno.dk)  
[www.pubhealth.ku.dk/~bxc](http://www.pubhealth.ku.dk/~bxc)



# Contents

<b>1</b>	<b>Follow-up data in the Epi package</b>	<b>1</b>
1.1	Timescales . . . . .	1
1.2	Splitting the follow-up time along a timescale . . . . .	5
1.3	Splitting time at a specific date . . . . .	7
1.4	Competing risks — multiple types of events . . . . .	9
1.5	Multiple events of the same type (recurrent events) . . . . .	11



# Chapter 1

## Follow-up data in the Epi package

In the `Epi`-package, follow-up data is represented by adding some extra variables to a dataframe. Such a dataframe is called a `Lexis` object. The tools for handling follow-up data then use the structure of this for special plots, tabulations etc.

Follow-up data basically consists of a time of entry, a time of exit and an indication of the status at exit (normally either “alive” or “dead”). Implicitly is also assumed a status *during* the follow-up (usually “alive”).

These three variables are specific for each *type* of outcome, i.e. cancer, cardiovascular event, death, ...

### 1.1 Timescales

A timescale is a variable that varies deterministically *within* each person during follow-up, *e.g.*:

- Age
- Calendar time
- Time since treatment
- Time since relapse

All timescales advance at the same pace, so the time followed is the same on all timescales. Therefore, it suffices to use only the entry point on each of the time scale, for example:

- Age at entry.
- Date of entry.
- Time since treatment (*at* treatment this is 0).
- Time since relapse.

In the `Epi` package, follow-up in a cohort is represented in a `Lexis` object. A `Lexis` object is a dataframe with a bit of extra structure representing the follow-up. For the `nickel` data we would construct a `Lexis` object by:

```
> data( nickel )
> nicL <- Lexis( entry = list( per=agein+dob,
+                               age=agein,
```

```
+               tfh=agein-age1st ),
+               exit = list( age=ageout ),
+               exit.status = ( icd %in% c(162,163) )*1,
+               data = nickel )
```

The `entry` argument is a *named* list with the entry points on each of the timescales we want to use. It defines the names of the timescales and the entry points. The `exit` argument gives the exit time on *one* of the timescales. This is sufficient, because the follow-up time on all time scale is the same, in this case `ageout - agein`. Now take a look at the result:

```
> str( nickel )
```

```
'data.frame':      679 obs. of  7 variables:
 $ id      : num  3 4 6 8 9 10 15 16 17 18 ...
 $ icd     : num  0 162 163 527 150 163 334 160 420 12 ...
 $ exposure: num  5 5 10 9 0 2 0 0.5 0 0 ...
 $ dob     : num 1889 1886 1881 1886 1880 ...
 $ age1st  : num 17.5 23.2 25.2 24.7 30.0 ...
 $ agein   : num 45.2 48.3 53.0 47.9 54.7 ...
 $ ageout  : num 93.0 63.3 54.2 69.7 76.8 ...
```

```
> str( nicL )
```

```
Classes 'Lexis' and 'data.frame':      679 obs. of  14 variables:
 $ per     : num 1934 1934 1934 1934 1934 ...
 $ age     : num 45.2 48.3 53.0 47.9 54.7 ...
 $ tfh     : num 27.7 25.1 27.7 23.2 24.8 ...
 $ lex.dur : num 47.75 15.00 1.17 21.77 22.10 ...
 $ lex.Cst : num 0 0 0 0 0 0 0 0 0 0 ...
 $ lex.Xst : num 0 1 1 0 0 1 0 0 0 0 ...
 $ lex.id  : int 1 2 3 4 5 6 7 8 9 10 ...
 $ id      : num 3 4 6 8 9 10 15 16 17 18 ...
 $ icd     : num 0 162 163 527 150 163 334 160 420 12 ...
 $ exposure: num 5 5 10 9 0 2 0 0.5 0 0 ...
 $ dob     : num 1889 1886 1881 1886 1880 ...
 $ age1st  : num 17.5 23.2 25.2 24.7 30.0 ...
 $ agein   : num 45.2 48.3 53.0 47.9 54.7 ...
 $ ageout  : num 93.0 63.3 54.2 69.7 76.8 ...
 - attr(*, "time.scales")= chr "per" "age" "tfh"
 - attr(*, "breaks")=List of 3
 ..$ per: NULL
 ..$ age: NULL
 ..$ tfh: NULL
```

```
> head( nicL )
```

	per	age	tfh	lex.dur	lex.Cst	lex.Xst	lex.id	id	icd	exposure
1	1934.246	45.2273	27.7465	47.7535	0	0	1	3	0	5
2	1934.246	48.2684	25.0820	15.0028	0	1	2	4	162	5
3	1934.246	52.9917	27.7465	1.1727	0	1	3	6	163	10
4	1934.246	47.9067	23.1861	21.7727	0	0	4	8	527	9
5	1934.246	54.7465	24.7890	22.0977	0	0	5	9	150	0
6	1934.246	44.3314	23.0437	18.2099	0	1	6	10	163	2
	dob	age1st	agein	ageout						
1	1889.019	17.4808	45.2273	92.9808						
2	1885.978	23.1864	48.2684	63.2712						
3	1881.255	25.2452	52.9917	54.1644						
4	1886.340	24.7206	47.9067	69.6794						
5	1879.500	29.9575	54.7465	76.8442						
6	1889.915	21.2877	44.3314	62.5413						

The `Lexis` object `nicL` has a variable for each timescale which is the entry point on this timescale. The follow-up time is in the variable `lex.dur` (**d**uration).

We defined the exit status to be death from lung cancer (ICD7 162,163), i.e. this variable is 1 if follow-up ended with a death from this cause. If follow-up ended alive or by death from another cause, the exit status is coded 0, i.e. aa a censoring.

Note that the exit status is in the variable `lex.Xst` (**eXit status**). The variable `lex.Csat` is the state where the follow-up takes place (**C**urrent **s**tatus), in this case 0 (alive).

It is possible to get a visualization of the follow-up along the timescales chosen by using the `plot` method for `Lexis` objects. `nicL` is an object of *class* `Lexis`, so using the function `plot()` on it means that R will look for the function `plot.Lexis` and use this function.

```
> plot( nicL )
```

The function allows a lot of control over the output, and a `points.Lexis` function allows plotting of the endpoints of follow-up.

```
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> plot( nicL, 1:2, lwd=1, col=c("blue","red")[(nicL$exp>0)+1],
+       grid=TRUE, lty.grid=1, col.grid=gray(0.7),
+       xlim=1900+c(0,90), xaxs="i",
```

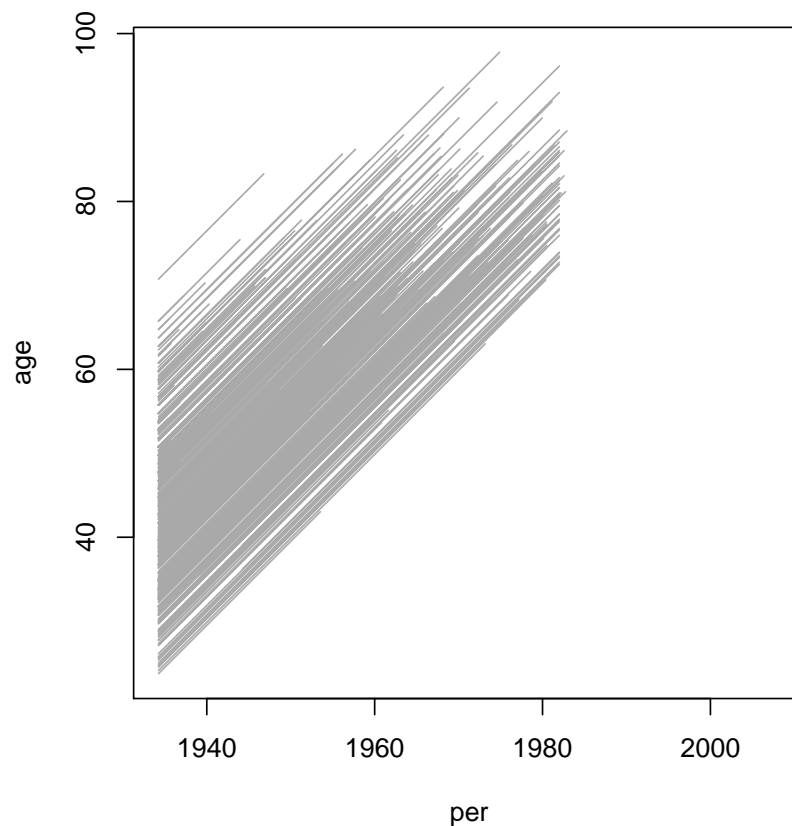


Figure 1.1: *Lexis diagram of the nickel dataset.*

```

+      ylim= 10+c(0,90), yaxs="i", las=1 )
> points( nicL, 1:2, pch=c(NA,3)[nicL$lex.Xst+1],
+        col="lightgray", lwd=3, cex=1.5 )
> points( nicL, 1:2, pch=c(NA,3)[nicL$lex.Xst+1],
+        col=c("blue","red")[(nicL$exp>0)+1], lwd=1, cex=1.5 )

```

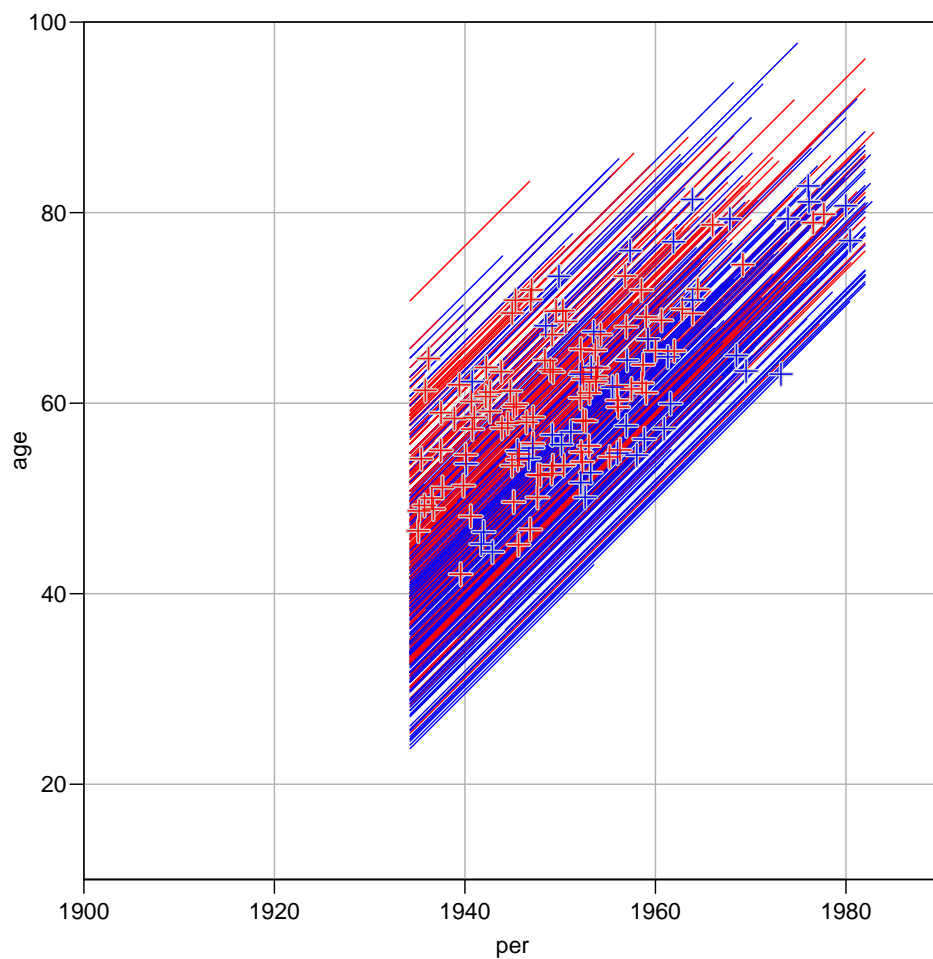


Figure 1.2: *Lexis diagram of the **nickel** dataset, with bells and whistles. The red lines are for persons with  $exposure > 0$ , so it is pretty evident that the oldest ones are the exposed part of the cohort.*

## 1.2 Splitting the follow-up time along a timescale

The follow-up time in a cohort can be subdivided by for example current age. This is achieved by the `splitLexis` (note it is *not* called `split.Lexis`). This requires that the timescale and the breakpoints on this timescale are supplied. Try:

```
> nicS1 <- splitLexis( nicL, "age", breaks=seq(0,100,10) )
> str( nicL )
```

Classes 'Lexis' and 'data.frame': 679 obs. of 14 variables:

```
$ per      : num 1934 1934 1934 1934 1934 ...
$ age      : num 45.2 48.3 53.0 47.9 54.7 ...
$ tfh      : num 27.7 25.1 27.7 23.2 24.8 ...
$ lex.dur   : num 47.75 15.00 1.17 21.77 22.10 ...
$ lex.Cst   : num 0 0 0 0 0 0 0 0 0 0 ...
$ lex.Xst   : num 0 1 1 0 0 1 0 0 0 0 ...
$ lex.id    : int 1 2 3 4 5 6 7 8 9 10 ...
$ id        : num 3 4 6 8 9 10 15 16 17 18 ...
$ icd       : num 0 162 163 527 150 163 334 160 420 12 ...
$ exposure : num 5 5 10 9 0 2 0 0.5 0 0 ...
$ dob       : num 1889 1886 1881 1886 1880 ...
$ age1st    : num 17.5 23.2 25.2 24.7 30.0 ...
$ agein     : num 45.2 48.3 53.0 47.9 54.7 ...
$ ageout    : num 93.0 63.3 54.2 69.7 76.8 ...
- attr(*, "time.scales")= chr "per" "age" "tfh"
- attr(*, "breaks")=List of 3
..$ per: NULL
..$ age: NULL
..$ tfh: NULL
```

```
> str( nicS1 )
```

Classes 'Lexis' and 'data.frame': 2210 obs. of 14 variables:

```
$ lex.id    : int 1 1 1 1 1 1 2 2 2 3 ...
$ per       : num 1934 1939 1949 1959 1969 ...
$ age       : num 45.2 50.0 60.0 70.0 80.0 ...
$ tfh       : num 27.7 32.5 42.5 52.5 62.5 ...
$ lex.dur    : num 4.77 10.00 10.00 10.00 10.00 ...
$ lex.Cst    : num 0 0 0 0 0 0 0 0 0 0 ...
$ lex.Xst    : num 0 0 0 0 0 0 0 0 1 1 ...
$ id         : num 3 3 3 3 3 3 4 4 4 6 ...
$ icd        : num 0 0 0 0 0 0 162 162 162 163 ...
$ exposure   : num 5 5 5 5 5 5 5 5 5 10 ...
$ dob        : num 1889 1889 1889 1889 1889 ...
$ age1st     : num 17.5 17.5 17.5 17.5 17.5 ...
$ agein      : num 45.2 45.2 45.2 45.2 45.2 ...
$ ageout     : num 93 93 93 93 93 ...
- attr(*, "breaks")=List of 3
..$ per: NULL
..$ age: num 0 10 20 30 40 50 60 70 80 90 ...
..$ tfh: NULL
- attr(*, "time.scales")= chr "per" "age" "tfh"
```

```
> round( subset( nicS1, id %in% 8:10 ), 2 )
```

	lex.id	per	age	tfh	lex.dur	lex.Cst	lex.Xst	id	icd	exposure	dob
11	4	1934.25	47.91	23.19	2.09	0	0	8	527	9	1886.34
12	4	1936.34	50.00	25.28	10.00	0	0	8	527	9	1886.34
13	4	1946.34	60.00	35.28	9.68	0	0	8	527	9	1886.34
14	5	1934.25	54.75	24.79	5.25	0	0	9	150	0	1879.50

```

15      5 1939.50 60.00 30.04   10.00      0      0 9 150      0 1879.50
16      5 1949.50 70.00 40.04    6.84      0      0 9 150      0 1879.50
17      6 1934.25 44.33 23.04    5.67      0      0 10 163     2 1889.91
18      6 1939.91 50.00 28.71   10.00      0      0 10 163     2 1889.91
19      6 1949.91 60.00 38.71    2.54      0      1 10 163     2 1889.91
      age1st agein ageout
11  24.72 47.91  69.68
12  24.72 47.91  69.68
13  24.72 47.91  69.68
14  29.96 54.75  76.84
15  29.96 54.75  76.84
16  29.96 54.75  76.84
17  21.29 44.33  62.54
18  21.29 44.33  62.54
19  21.29 44.33  62.54

```

The resulting object is again a Lexis object, and so follow-up may be split further along another timescale. Try this and list the result for individuals 4 and 6:

```

> nicS2 <- splitLexis( nicS1, "tfh", breaks=c(0,1,5,10,20,30,100) )
> round( subset( nicS2, id %in% 8:10 ), 2 )

```

```

      lex.id      per      age      tfh lex.dur lex.Cst lex.Xst id icd exposure      dob
13      4 1934.25 47.91 23.19    2.09      0      0 8 527      9 1886.34
14      4 1936.34 50.00 25.28    4.72      0      0 8 527      9 1886.34
15      4 1941.06 54.72 30.00    5.28      0      0 8 527      9 1886.34
16      4 1946.34 60.00 35.28    9.68      0      0 8 527      9 1886.34
17      5 1934.25 54.75 24.79    5.21      0      0 9 150      0 1879.50
18      5 1939.46 59.96 30.00    0.04      0      0 9 150      0 1879.50
19      5 1939.50 60.00 30.04   10.00      0      0 9 150      0 1879.50
20      5 1949.50 70.00 40.04    6.84      0      0 9 150      0 1879.50
21      6 1934.25 44.33 23.04    5.67      0      0 10 163     2 1889.91
22      6 1939.91 50.00 28.71    1.29      0      0 10 163     2 1889.91
23      6 1941.20 51.29 30.00    8.71      0      0 10 163     2 1889.91
24      6 1949.91 60.00 38.71    2.54      0      1 10 163     2 1889.91
      age1st agein ageout
13  24.72 47.91  69.68
14  24.72 47.91  69.68
15  24.72 47.91  69.68
16  24.72 47.91  69.68
17  29.96 54.75  76.84
18  29.96 54.75  76.84
19  29.96 54.75  76.84
20  29.96 54.75  76.84
21  21.29 44.33  62.54
22  21.29 44.33  62.54
23  21.29 44.33  62.54
24  21.29 44.33  62.54

```

If we want to model the effect of these timescales we will for each interval use either the value of the left endpoint in each interval or the middle. There is a function `timeBand` which returns these. Try:

```

> timeBand( nicS2, "age", "middle" )[1:20]

[1] 45 45 55 65 75 85 95 45 55 55 65 55 45 55 55 65 55 55 65 75

> # For nice printing and column labelling use the data.frame() function:
> data.frame( nicS2[,c("id","lex.id","per","age","tfh","lex.dur")],
+             mid.age=timeBand( nicS2, "age", "middle" ),
+             mid.tfh=timeBand( nicS2, "tfh", "middle" ) )[1:20,]

```



```

4100 1886.340 24.7206 47.9067 69.6794 30.27616
5    1879.500 29.9575 54.7465 76.8442      Inf
6    1889.915 21.2877 44.3314 62.5413 46.28770
680  1889.915 21.2877 44.3314 62.5413 46.28770

```

(The `cens=` argument is explained below). Note that individual 6 has had his follow-up split at age 25 where 50 exposure-years were attained. This could also have been achieved in the split dataset, try:

```
> subset( nicS2, id %in% 8:10 )
```

	lex.id	per	age	tfh	lex.dur	lex.Cst	lex.Xst	id	icd	exposure
13	4	1934.246	47.9067	23.1861	2.0933	0	0	8	527	9
14	4	1936.340	50.0000	25.2794	4.7206	0	0	8	527	9
15	4	1941.060	54.7206	30.0000	5.2794	0	0	8	527	9
16	4	1946.340	60.0000	35.2794	9.6794	0	0	8	527	9
17	5	1934.246	54.7465	24.7890	5.2110	0	0	9	150	0
18	5	1939.457	59.9575	30.0000	0.0425	0	0	9	150	0
19	5	1939.500	60.0000	30.0425	10.0000	0	0	9	150	0
20	5	1949.500	70.0000	40.0425	6.8442	0	0	9	150	0
21	6	1934.246	44.3314	23.0437	5.6686	0	0	10	163	2
22	6	1939.915	50.0000	28.7123	1.2877	0	0	10	163	2
23	6	1941.203	51.2877	30.0000	8.7123	0	0	10	163	2
24	6	1949.915	60.0000	38.7123	2.5413	0	1	10	163	2
dob age1st agein ageout										
13	1886.340	24.7206	47.9067	69.6794						
14	1886.340	24.7206	47.9067	69.6794						
15	1886.340	24.7206	47.9067	69.6794						
16	1886.340	24.7206	47.9067	69.6794						
17	1879.500	29.9575	54.7465	76.8442						
18	1879.500	29.9575	54.7465	76.8442						
19	1879.500	29.9575	54.7465	76.8442						
20	1879.500	29.9575	54.7465	76.8442						
21	1889.915	21.2877	44.3314	62.5413						
22	1889.915	21.2877	44.3314	62.5413						
23	1889.915	21.2877	44.3314	62.5413						
24	1889.915	21.2877	44.3314	62.5413						

```

> nicS2$agehi <- nicS2$age1st + 50 / nicS2$exposure
> nicS2C <- cutLexis( data=nicS2, cut=nicS2$agehi, timescale="age", new.state=2, cens=0 )
> subset( nicS2C, id %in% 8:10 )

```

	lex.id	per	age	tfh	lex.dur	lex.Cst	lex.Xst	id	icd	exposure
3142	4	1934.246	47.9067	23.1861	2.0933	2	2	8	527	9
3143	4	1936.340	50.0000	25.2794	4.7206	2	2	8	527	9
3144	4	1941.060	54.7206	30.0000	5.2794	2	2	8	527	9
3145	4	1946.340	60.0000	35.2794	9.6794	2	2	8	527	9
17	5	1934.246	54.7465	24.7890	5.2110	0	0	9	150	0
18	5	1939.457	59.9575	30.0000	0.0425	0	0	9	150	0
19	5	1939.500	60.0000	30.0425	10.0000	0	0	9	150	0
20	5	1949.500	70.0000	40.0425	6.8442	0	0	9	150	0
21	6	1934.246	44.3314	23.0437	1.9563	0	2	10	163	2
3150	6	1936.203	46.2877	25.0000	3.7123	2	2	10	163	2
3151	6	1939.915	50.0000	28.7123	1.2877	2	2	10	163	2
3152	6	1941.203	51.2877	30.0000	8.7123	2	2	10	163	2
3153	6	1949.915	60.0000	38.7123	2.5413	2	1	10	163	2
dob age1st agein ageout agehi										
3142	1886.340	24.7206	47.9067	69.6794	30.27616					
3143	1886.340	24.7206	47.9067	69.6794	30.27616					
3144	1886.340	24.7206	47.9067	69.6794	30.27616					

```

3145 1886.340 24.7206 47.9067 69.6794 30.27616
17   1879.500 29.9575 54.7465 76.8442      Inf
18   1879.500 29.9575 54.7465 76.8442      Inf
19   1879.500 29.9575 54.7465 76.8442      Inf
20   1879.500 29.9575 54.7465 76.8442      Inf
21   1889.915 21.2877 44.3314 62.5413 46.28770
3150 1889.915 21.2877 44.3314 62.5413 46.28770
3151 1889.915 21.2877 44.3314 62.5413 46.28770
3152 1889.915 21.2877 44.3314 62.5413 46.28770
3153 1889.915 21.2877 44.3314 62.5413 46.28770

```

Note that follow-up subsequent to the event is classified as being in state 2, but that the final transition to state 1 (death from lung cancer) is preserved. This is the point of the `cens=` argument. It names the states (in this case 0, “Alive”) that will be over-ridden by `new.state` (in this case 2, “High exposure”). Clearly, state 1 (“Dead”) should not be updated even if it is after the time where the persons moves to state 2.

Note if the intermediate event is to be used as a time-dependent variable in a Cox-model, then `lex.Cst` should be used as the time-dependent variable, and `lex.Xst==1` as the event.

## 1.4 Competing risks — multiple types of events

If we want to consider death from lung cancer and death from other causes as separate events we can code these as for example 1 and 2.

```

> data( nickel )
> nicL <- Lexis( entry = list( per=agein+dob,
+                             age=agein,
+                             tfh=agein-age1st ),
+               exit = list( age=ageout ),
+               exit.status = ( icd > 0 ) + ( icd %in% c(162,163) ),
+               data = nickel )
> str( nicL )

Classes 'Lexis' and 'data.frame':      679 obs. of  14 variables:
 $ per      : num  1934 1934 1934 1934 1934 ...
 $ age      : num  45.2 48.3 53.0 47.9 54.7 ...
 $ tfh      : num  27.7 25.1 27.7 23.2 24.8 ...
 $ lex.dur  : num  47.75 15.00  1.17 21.77 22.10 ...
 $ lex.Cst  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ lex.Xst  : int  0 2 2 1 1 2 1 1 1 1 ...
 $ lex.id   : int  1 2 3 4 5 6 7 8 9 10 ...
 $ id       : num  3 4 6 8 9 10 15 16 17 18 ...
 $ icd      : num  0 162 163 527 150 163 334 160 420 12 ...
 $ exposure: num  5 5 10 9 0 2 0 0.5 0 0 ...
 $ dob      : num  1889 1886 1881 1886 1880 ...
 $ age1st   : num  17.5 23.2 25.2 24.7 30.0 ...
 $ agein    : num  45.2 48.3 53.0 47.9 54.7 ...
 $ ageout   : num  93.0 63.3 54.2 69.7 76.8 ...
 - attr(*, "time.scales")= chr  "per" "age" "tfh"
 - attr(*, "breaks")=List of 3
 ..$ per: NULL
 ..$ age: NULL
 ..$ tfh: NULL

> head( nicL )

      per      age      tfh lex.dur lex.Cst lex.Xst lex.id id icd exposure
1 1934.246 45.2273 27.7465 47.7535      0      0      1 3  0      5

```

```

2 1934.246 48.2684 25.0820 15.0028      0      2      2 4 162      5
3 1934.246 52.9917 27.7465  1.1727      0      2      3 6 163     10
4 1934.246 47.9067 23.1861 21.7727      0      1      4 8 527      9
5 1934.246 54.7465 24.7890 22.0977      0      1      5 9 150      0
6 1934.246 44.3314 23.0437 18.2099      0      2      6 10 163     2
      dob  age1st  agein  ageout
1 1889.019 17.4808 45.2273 92.9808
2 1885.978 23.1864 48.2684 63.2712
3 1881.255 25.2452 52.9917 54.1644
4 1886.340 24.7206 47.9067 69.6794
5 1879.500 29.9575 54.7465 76.8442
6 1889.915 21.2877 44.3314 62.5413

> subset( nicL, id %in% 8:10 )

      per      age      tfh lex.dur lex.Cst lex.Xst lex.id id icd exposure
4 1934.246 47.9067 23.1861 21.7727      0      1      4 8 527      9
5 1934.246 54.7465 24.7890 22.0977      0      1      5 9 150      0
6 1934.246 44.3314 23.0437 18.2099      0      2      6 10 163     2
      dob  age1st  agein  ageout
4 1886.340 24.7206 47.9067 69.6794
5 1879.500 29.9575 54.7465 76.8442
6 1889.915 21.2877 44.3314 62.5413

```

If we want to label the states, we can enter the names of these in the **states** parameter, try for example:

```

> nicL <- Lexis( entry = list( per=agein+dob,
+                               age=agein,
+                               tfh=agein-age1st ),
+               exit = list( age=ageout ),
+               exit.status = ( icd > 0 ) + ( icd %in% c(162,163) ),
+               data = nickel,
+               states = c("Alive", "D.oth", "D.lung") )
> str( nicL )

Classes 'Lexis' and 'data.frame':      679 obs. of  14 variables:
 $ per      : num  1934 1934 1934 1934 1934 ...
 $ age      : num  45.2 48.3 53.0 47.9 54.7 ...
 $ tfh      : num  27.7 25.1 27.7 23.2 24.8 ...
 $ lex.dur  : num  47.75 15.00  1.17 21.77 22.10 ...
 $ lex.Cst  : Factor w/ 3 levels "Alive","D.oth",...: 1 1 1 1 1 1 1 1 1 ...
 $ lex.Xst  : Factor w/ 3 levels "Alive","D.oth",...: 1 2 2 3 3 2 3 3 3 ...
 $ lex.id   : int   1 2 3 4 5 6 7 8 9 10 ...
 $ id       : num   3 4 6 8 9 10 15 16 17 18 ...
 $ icd      : num   0 162 163 527 150 163 334 160 420 12 ...
 $ exposure: num   5 5 10 9 0 2 0 0.5 0 0 ...
 $ dob      : num  1889 1886 1881 1886 1880 ...
 $ age1st   : num  17.5 23.2 25.2 24.7 30.0 ...
 $ agein    : num  45.2 48.3 53.0 47.9 54.7 ...
 $ ageout   : num  93.0 63.3 54.2 69.7 76.8 ...
 - attr(*, "time.scales")= chr  "per" "age" "tfh"
 - attr(*, "breaks")=List of 3
 ..$ per: NULL
 ..$ age: NULL
 ..$ tfh: NULL

```

You can get an overview of the number of records by state and transitions between states as well as the person-years in each state by using `tab.Lexis()`, try:

```
> tab.Lexis( nicL )

States:
  #records:
    To
From   Alive D.oth D.lung Sum #events: #risk time:      Rate      (95%
  Alive    47   137   495 679      632    15348.06 0.04117785 0.03808940

States:
  #records:
    To
From      c.int.)
  Alive 0.04451673
```

When we cut at a date as in this case, the date where cumulative exposure exceeds 50 exposure-years, we get the follow-up *after* the date classified as being in the new state if it was in a state we

```
> nicL$agehi <- nicL$age1st + 50 / nicL$exposure
> nicC <- cutLexis( data=nicL, cut=nicL$agehi, "age", new.state="HiExp", cens="Alive" )
> subset( nicC, id %in% 8:10 )
```

	per	age	tfh	lex.dur	lex.Cst	lex.Xst	lex.id	id	icd	exposure
4100	1934.246	47.9067	23.1861	21.7727	HiExp	D.lung	4	8	527	9
5	1934.246	54.7465	24.7890	22.0977	Alive	D.lung	5	9	150	0
6	1934.246	44.3314	23.0437	1.9563	Alive	HiExp	6	10	163	2
680	1936.203	46.2877	25.0000	16.2536	HiExp	D.oth	6	10	163	2
	dob	age1st	agein	ageout	agehi					
4100	1886.340	24.7206	47.9067	69.6794	30.27616					
5	1879.500	29.9575	54.7465	76.8442	Inf					
6	1889.915	21.2877	44.3314	62.5413	46.28770					
680	1889.915	21.2877	44.3314	62.5413	46.28770					

```
> tab.Lexis( nicC )

States:
  #records:
    To
From   Alive D.oth D.lung HiExp Sum #events: #risk time:      Rate
  Alive    39    65    279    83 466      427    10772.533 0.03963785
  HiExp     0    72    216     8 296      288     4575.524 0.06294361
  Sum       39   137    495    91 762      715    15348.057 0.04658570

States:
  #records:
    To
From      (95%      c.int.)
  Alive 0.03605095 0.04358162
  HiExp 0.05607809 0.07064967
  Sum   0.04329312 0.05012869
```

Note that the persons-years is the same, but that the number of events has changed. This is because events are now defined as any transition from alive, including the transitions to HiExp.

## 1.5 Multiple events of the same type (recurrent events)

Sometimes more events of the same type are recorded for each person and one would then like to count these and put follow-up time in states accordingly. Essentially, each set of cutpoints

represents progressions from one state to the next. Therefore the states should be numbered, and the numbering of states subsequently occupied be increased accordingly.

This is a behaviour different from the one outlined above, and it is achieved by the argument `count=TRUE` to `cutLexis`. When `count` is set to `TRUE`, the value of the arguments `new.state` and `cens` are ignored.