Sequence Analysis:
A Non-Parametric Approach to Study Pathways to Adulthood

Rimantas Vosylis

Institute of Psychology at Mykolas Romeris University

What are sequences and where can we find them?

A string of values of a categorical (nominal / ranks) variable

A string of states (state – a certain value of a categorical variable)

It can look like this:

1 1 2 3 4 3 2 1 4 1 1 2 3 4 3 2 1 4

Or this:

A-D-A-D-D-D-D-A-A-A-A-D-D-D

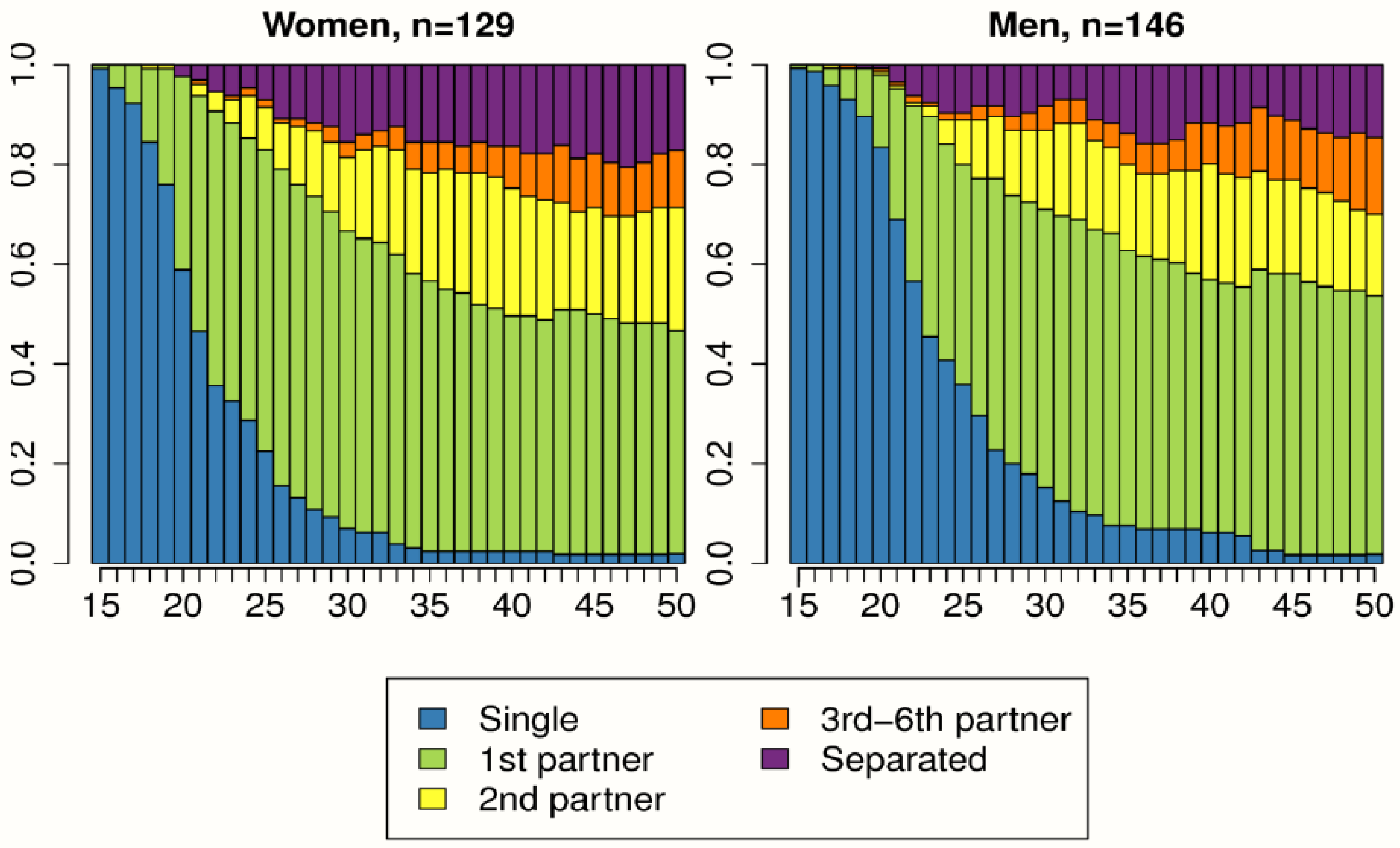
Sequences are ussually obtained from prospective (in psychology) or retrospective (in sociology) longitudinal studies

What can we do with sequence analysis?

- Visualize the (longitudinal categorical) data!
- Investigate the sequences characteristics and link these characteristics to certain variables
- Build a typology of sequences, i.e., uncover certain groups that have similar sequences

Do lots of other cool stuff

Figure 1: State distribution plots of partnership histories for women and men between ages 15–50 in JYLS data



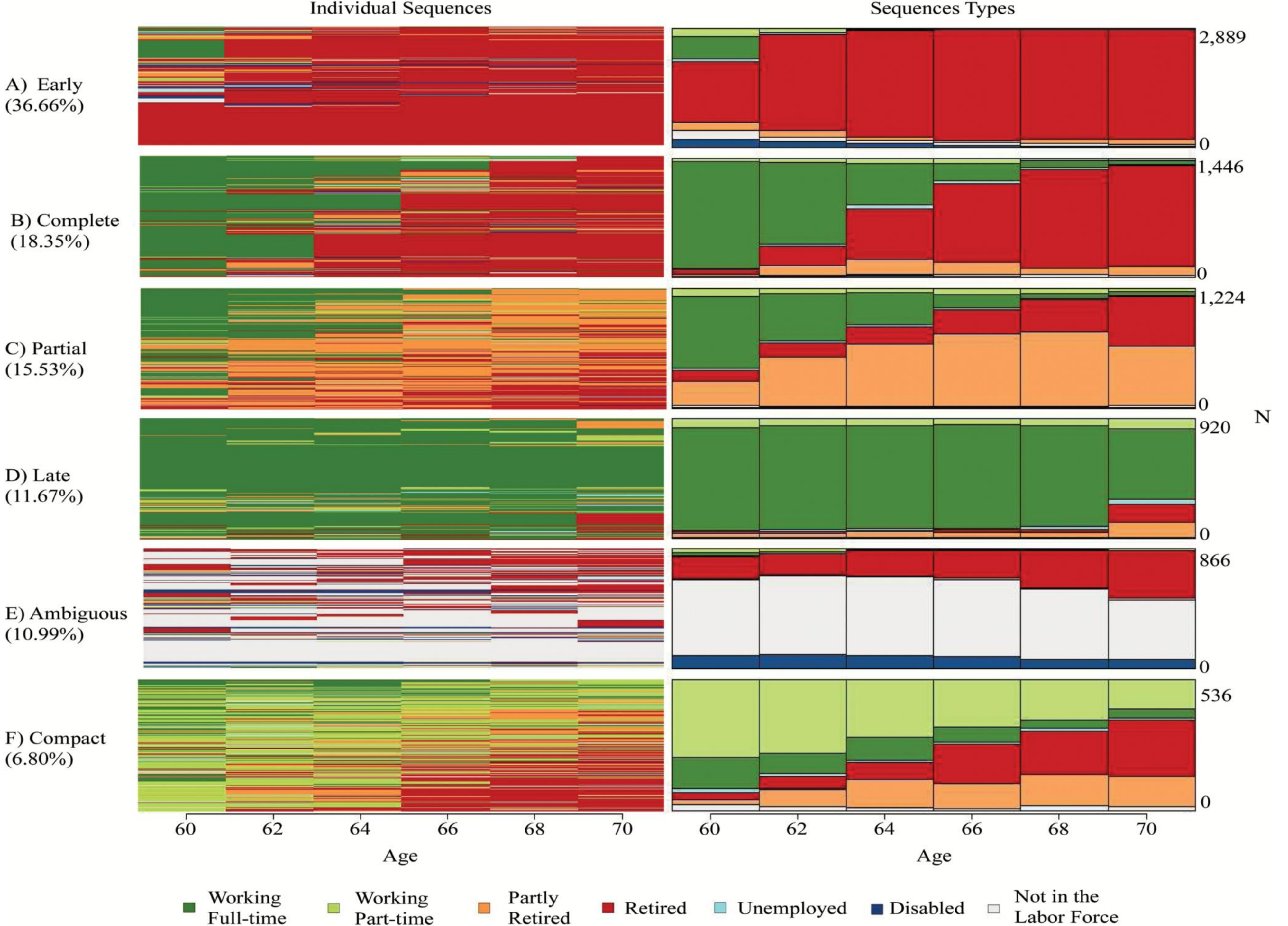
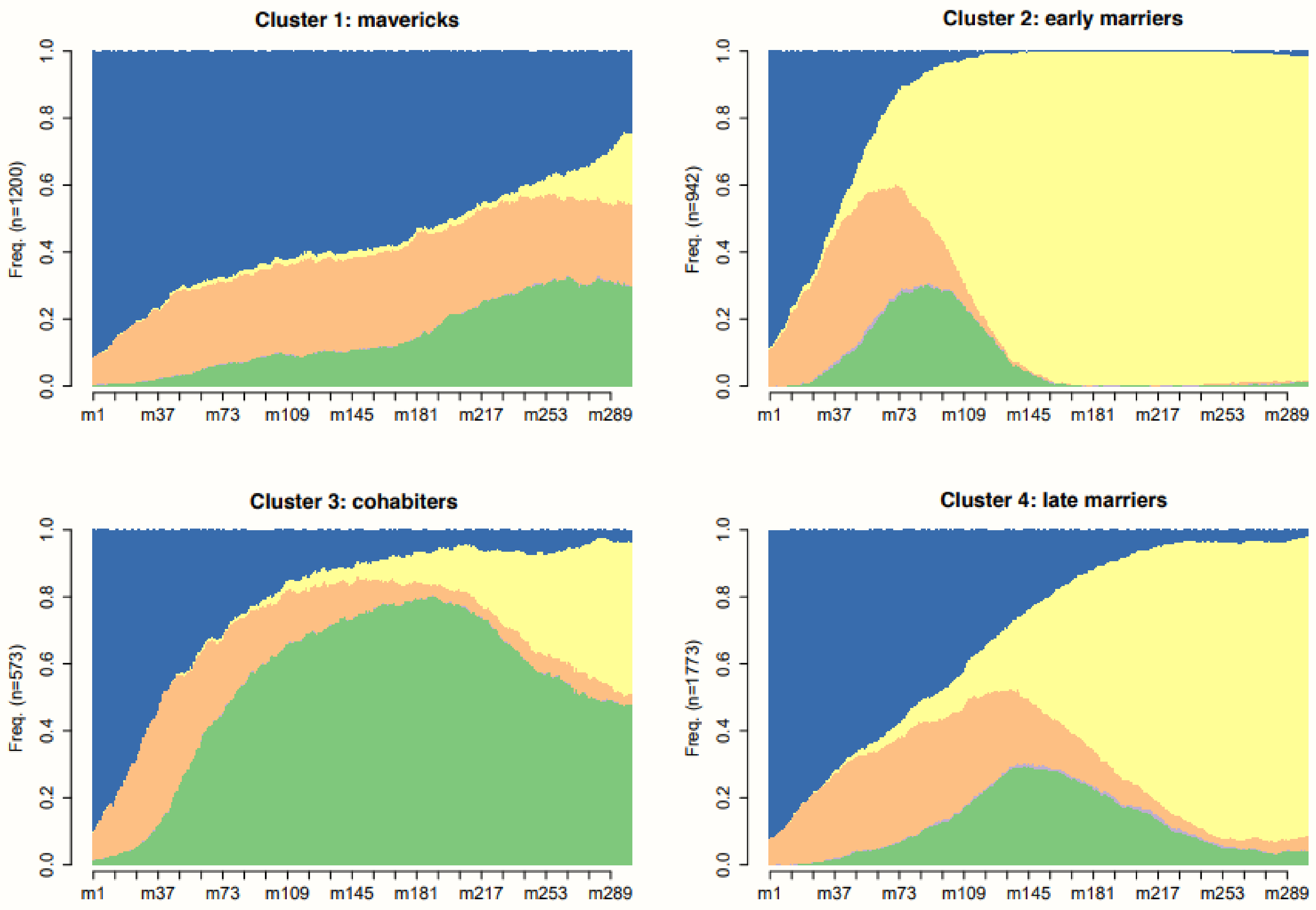
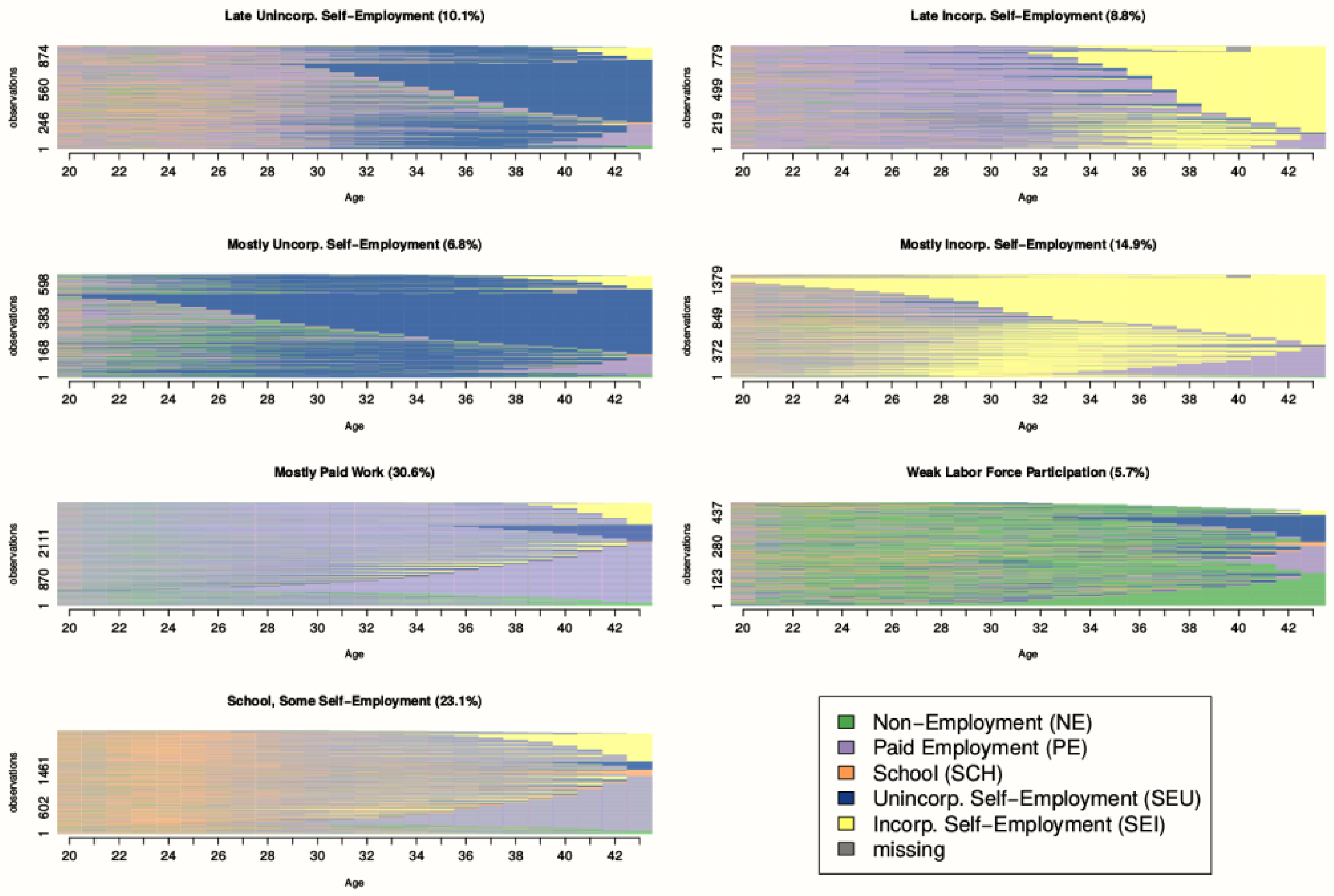

Individual Sequences
Sequences Types
A) Early
(36.66%)
B) Complete
(18.35%)
C) Partial
(15.53%)
D) Late
(11.67%)
E) Ambiguous
(10.99%)
F) Compact
(6.80%)
2,889
0
1,446
0
1,224
0
920
0
866
0
536
0
N
60
62
64
66
68
70
Age
Working Full-time
Working Part-time
Partly Retired
Retired
Unemployed
Disabled
Not in the Labor Force

Figure 10: Distribution of partnership statuses in clusters 1–4

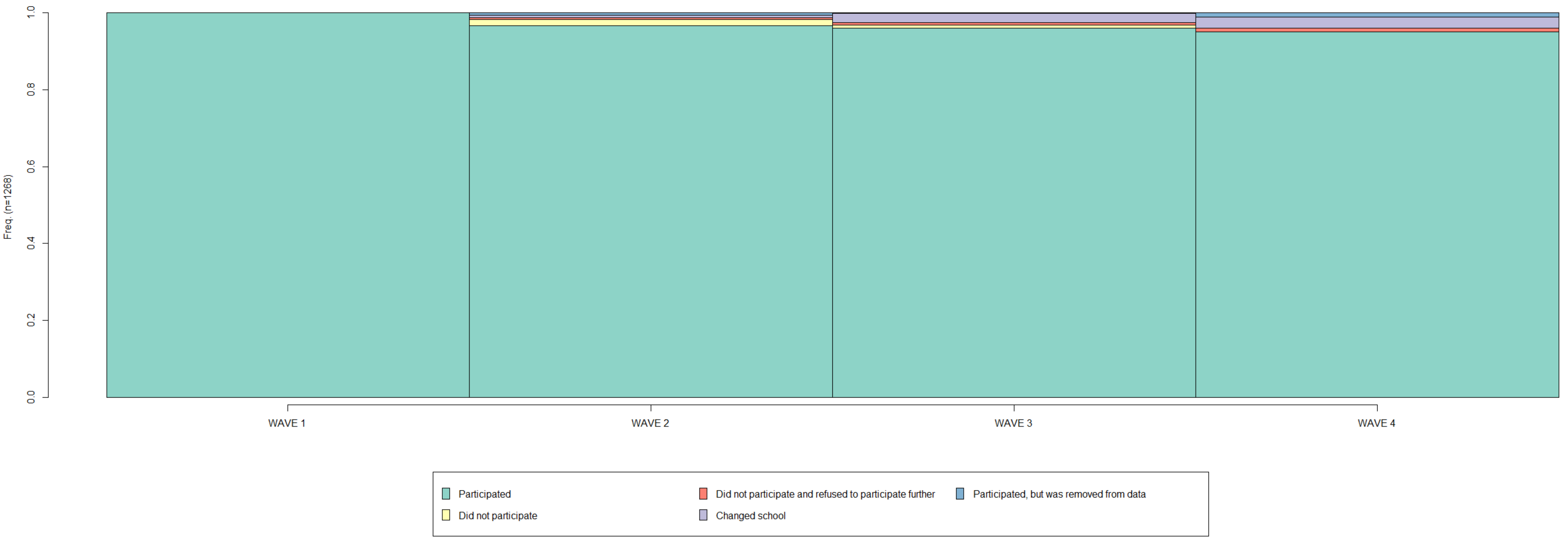


Source: Pairfam waves 1–6, own research.

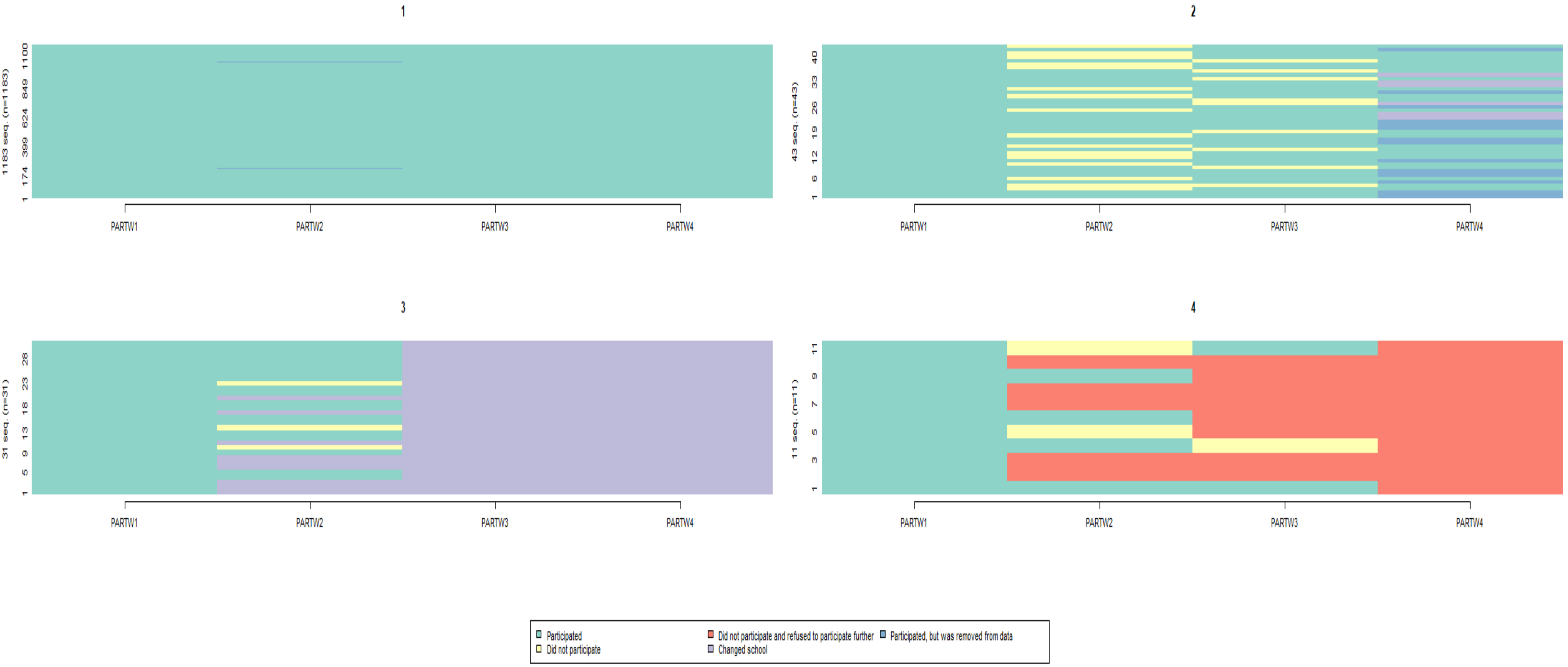
Figure 3: Clusters of Life Cycles Involving Self-Employment (1970 birth cohort)



Notes: Figure shows life employment profiles of all Swedish males born in 1970 who are ever self-employed between 1990 and 2013.



Freq. (n=1268)
0.0
0.2
0.4
0.6
0.8
1.0
WAVE 1
WAVE 2
WAVE 3
WAVE 4
Participated
Did not participate
Did not participate and refused to participate further
Changed school
Participated, but was removed from data


1
1183 seq. (n=1183)
1 174 399 624 849 1100
PARTW1
PARTW2
PARTW3
PARTW4
2
43 seq. (n=43)
1 6 12 19 26 33 40
3
31 seq. (n=31)
1 5 9 13 18 23 28
4
11 seq. (n=11)
1 3 5 7 9 11
Participated
Did not participate
Did not participate and refused to participate further
Changed school
Participated, but was removed from data

Instruments: Life History Calendar (LHC)

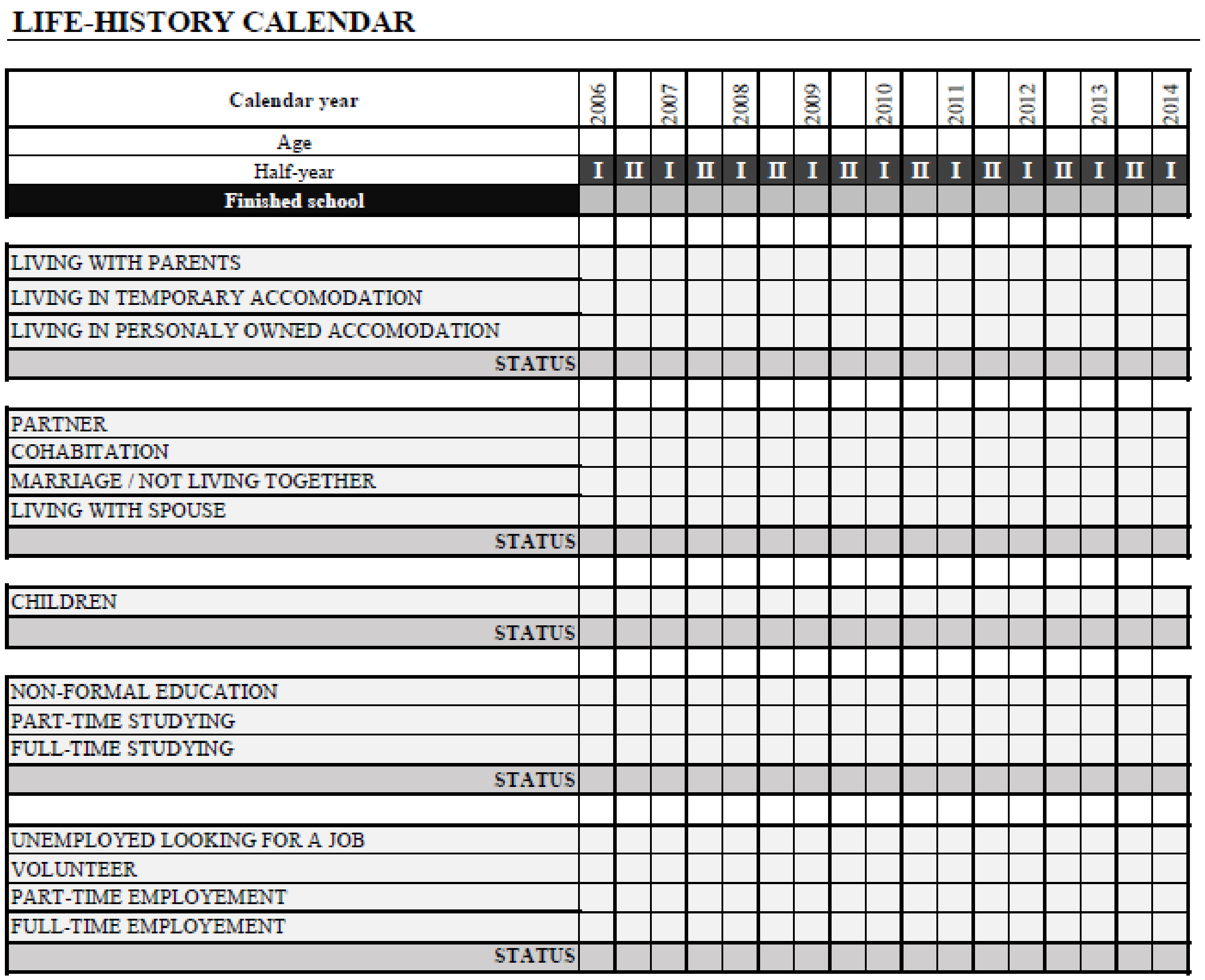
- LHC is an interview-based assessment that was used to collect data on **partner, parenthood, living arrangements, educational and work status histories.**

Interviews took about 15-25 minutes to complete.

Starting point of the life story in terms of role statuses, was finishing school

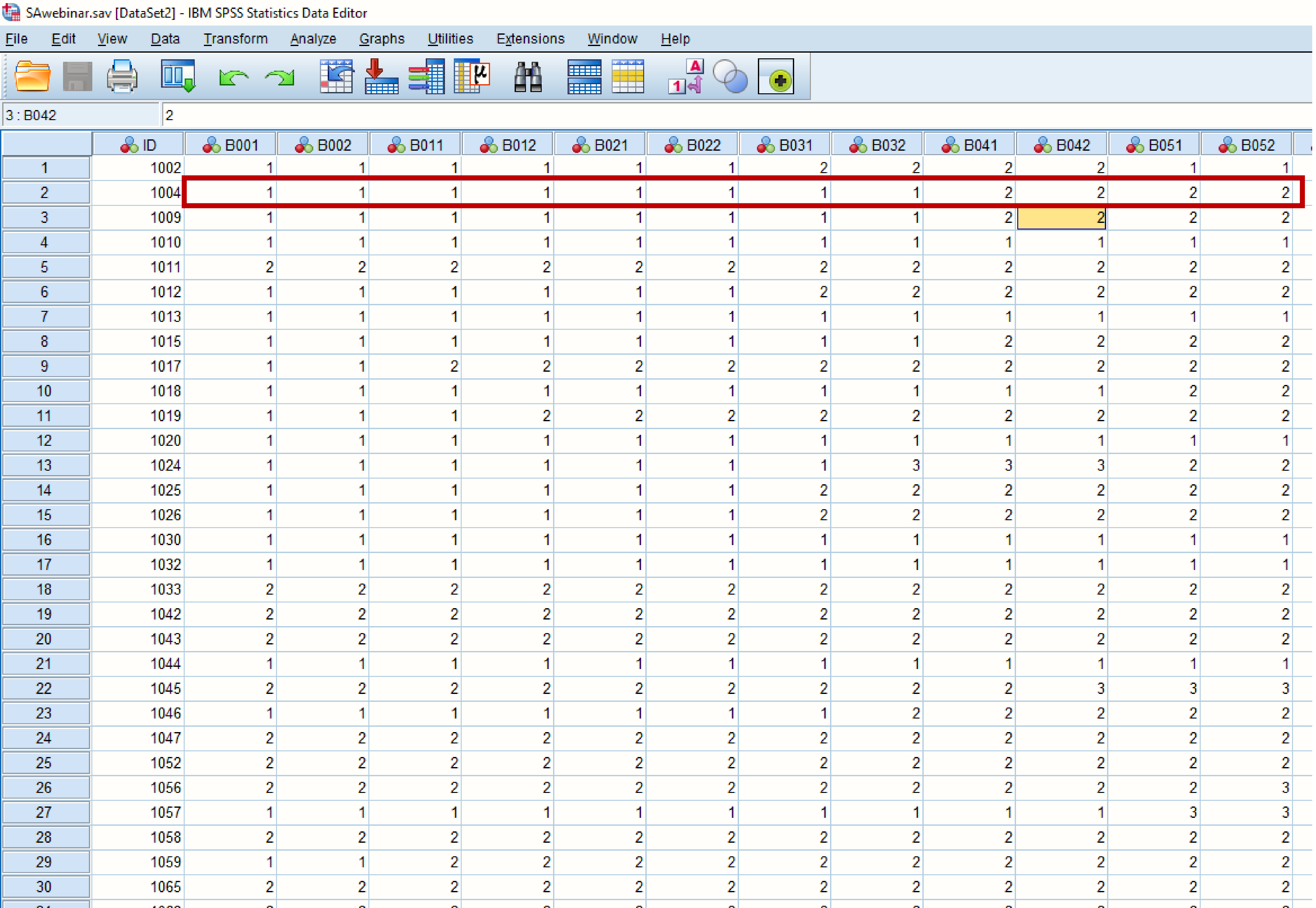
Finishing point was – current moment

Minimal interval - six months



LIFE-HISTORY CALENDAR

Calendar year	2006		2007		2008		2009		2010		2011		2012		2013		2014
Age																	
Half-year	I	II	I	II	I	II	I	II	I	II	I	II	I	II	I	II	I
Finished school																	
LIVING WITH PARENTS																	
LIVING IN TEMPORARY ACCOMODATION																	
LIVING IN PERSONALY OWNED ACCOMODATION																	
STATUS																	
PARTNER																	
COHABITATION																	
MARRIAGE / NOT LIVING TOGETHER																	
LIVING WITH SPOUSE																	
STATUS																	
CHILDREN																	
STATUS																	
NON-FORMAL EDUCATION																	
PART-TIME STUDYING																	
FULL-TIME STUDYING																	
STATUS																	
UNEMPLOYED LOOKING FOR A JOB																	
VOLUNTEER																	
PART-TIME EMPLOYEMENT																	
FULL-TIME EMPLOYEMENT																	
STATUS																	



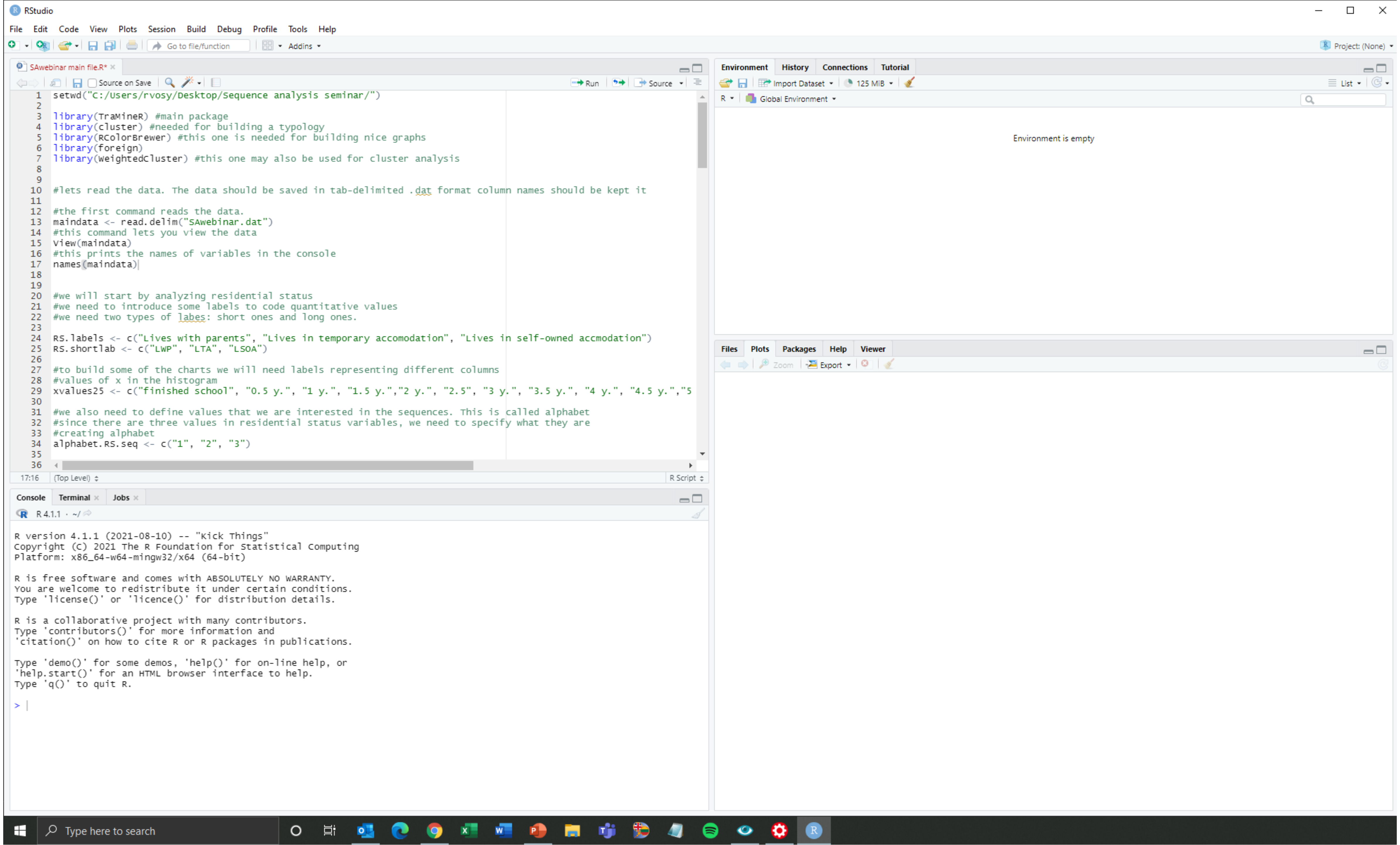
3 : B042 2

	ID	B001	B002	B011	B012	B021	B022	B031	B032	B041	B042	B051	B052
1	1002	1	1	1	1	1	1	2	2	2	2	1	1
2	1004	1	1	1	1	1	1	1	1	2	2	2	2
3	1009	1	1	1	1	1	1	1	1	2	2	2	2
4	1010	1	1	1	1	1	1	1	1	1	1	1	1
5	1011	2	2	2	2	2	2	2	2	2	2	2	2
6	1012	1	1	1	1	1	1	2	2	2	2	2	2
7	1013	1	1	1	1	1	1	1	1	1	1	1	1
8	1015	1	1	1	1	1	1	1	1	2	2	2	2
9	1017	1	1	2	2	2	2	2	2	2	2	2	2
10	1018	1	1	1	1	1	1	1	1	1	1	2	2
11	1019	1	1	1	2	2	2	2	2	2	2	2	2
12	1020	1	1	1	1	1	1	1	1	1	1	1	1
13	1024	1	1	1	1	1	1	1	3	3	3	2	2
14	1025	1	1	1	1	1	1	2	2	2	2	2	2
15	1026	1	1	1	1	1	1	2	2	2	2	2	2
16	1030	1	1	1	1	1	1	1	1	1	1	1	1
17	1032	1	1	1	1	1	1	1	1	1	1	1	1
18	1033	2	2	2	2	2	2	2	2	2	2	2	2
19	1042	2	2	2	2	2	2	2	2	2	2	2	2
20	1043	2	2	2	2	2	2	2	2	2	2	2	2
21	1044	1	1	1	1	1	1	1	1	1	1	1	1
22	1045	2	2	2	2	2	2	2	2	2	3	3	3
23	1046	1	1	1	1	1	1	1	2	2	2	2	2
24	1047	2	2	2	2	2	2	2	2	2	2	2	2
25	1052	2	2	2	2	2	2	2	2	2	2	2	2
26	1056	2	2	2	2	2	2	2	2	2	2	2	3
27	1057	1	1	1	1	1	1	1	1	1	1	3	3
28	1058	2	2	2	2	2	2	2	2	2	2	2	2
29	1059	1	1	2	2	2	2	2	2	2	2	2	2
30	1065	2	2	2	2	2	2	2	2	2	2	2	2

First steps

- Download R software
- Download R Studio software

- Save data from SPSS into *.dat file **with column names included in the dataset**
- Preinstall libraries



```r
setwd("C:/Users/rvosy/Desktop/Sequence analysis seminar/")

library(TraMineR) #main package
library(cluster) #needed for building a typology
library(RColorBrewer) #this one is needed for building nice graphs
library(foreign)
library(WeightedCluster) #this one may also be used for cluster analysis


#lets read the data. The data should be saved in tab-delimited .dat format column names should be kept it

#the first command reads the data.
maindata <- read.delim("SAwebinar.dat")
#this command lets you view the data
View(maindata)
#this prints the names of variables in the console
names(maindata)


#we will start by analyzing residential status
#we need to introduce some labels to code quantitative values
#we need two types of labes: short ones and long ones.

RS.labels <- c("Lives with parents", "Lives in temporary accomodation", "Lives in self-owned accmodation")
RS.shortlab <- c("LWP", "LTA", "LSOA")

#to build some of the charts we will need labels representing different columns
#values of x in the histogram
xvalues25 <- c("finished school", "0.5 y.", "1 y.", "1.5 y.","2 y.", "2.5", "3 y.", "3.5 y.", "4 y.", "4.5 y.","5

#we also need to define values that we are interested in the sequences. This is called alphabet
#since there are three values in residential status variables, we need to specify what they are
#creating alphabet
alphabet.RS.seq <- c("1", "2", "3")

```

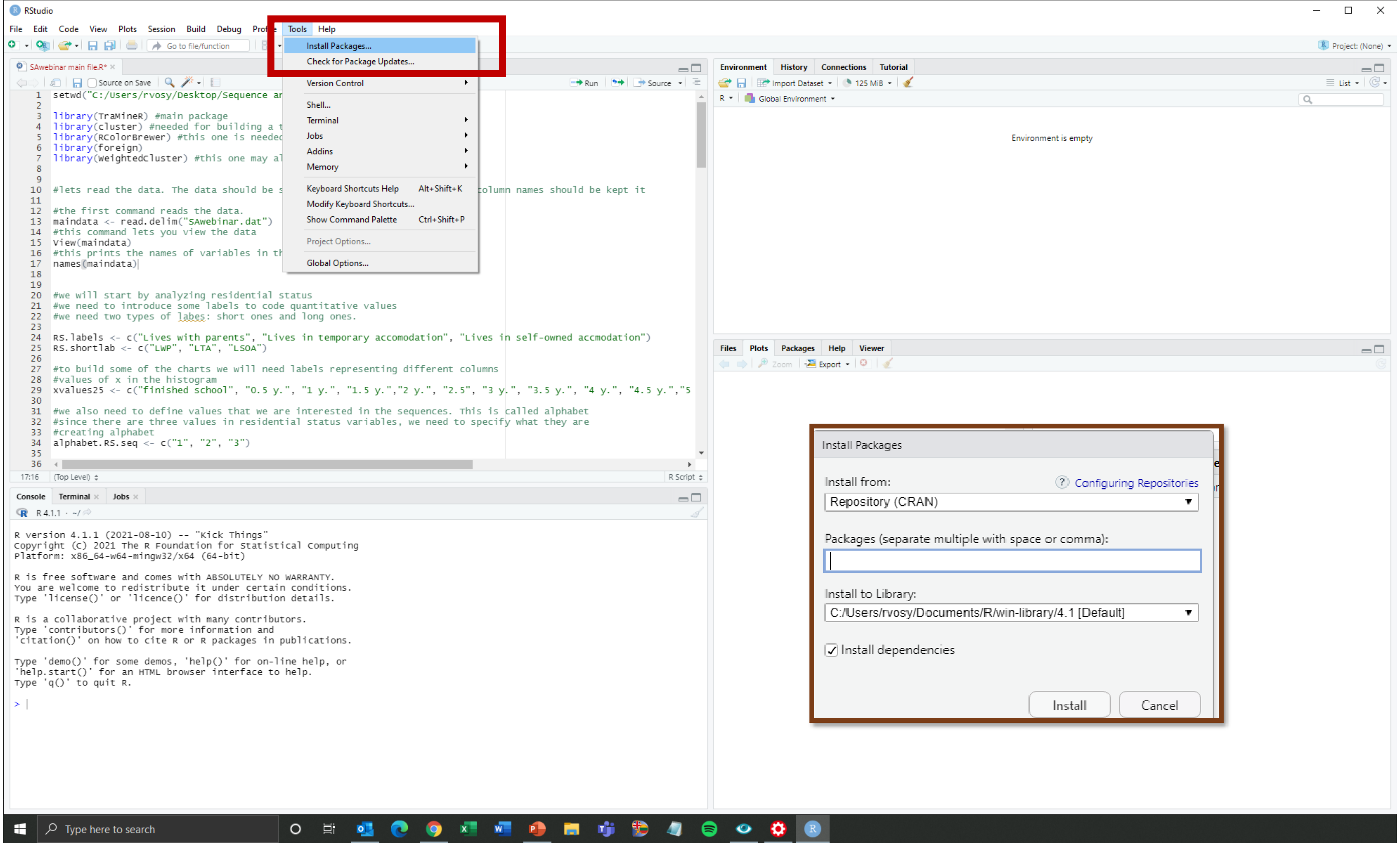
```
R version 4.1.1 (2021-08-10) -- "Kick Things"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```



```
setwd("C:/Users/rvosy/Desktop/Sequence ar

library(TraMineR) #main package
library(cluster) #needed for building a t
library(RColorBrewer) #this one is needec
library(foreign)
library(WeightedCluster) #this one may al


#lets read the data. The data should be s                                column names should be kept it

#the first command reads the data.
maindata <- read.delim("SAwebinar.dat")
#this command lets you view the data
View(maindata)
#this prints the names of variables in th
names(maindata)


#we will start by analyzing residential status
#we need to introduce some labels to code quantitative values
#we need two types of labes: short ones and long ones.

RS.labels <- c("Lives with parents", "Lives in temporary accomodation", "Lives in self-owned accmodation")
RS.shortlab <- c("LWP", "LTA", "LSOA")

#to build some of the charts we will need labels representing different columns
#values of x in the histogram
xvalues25 <- c("finished school", "0.5 y.", "1 y.", "1.5 y.","2 y.", "2.5", "3 y.", "3.5 y.", "4 y.", "4.5 y.","5

#we also need to define values that we are interested in the sequences. This is called alphabet
#since there are three values in residential status variables, we need to specify what they are
#creating alphabet
alphabet.RS.seq <- c("1", "2", "3")
```

```
R version 4.1.1 (2021-08-10) -- "Kick Things"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

Install Packages

Install from: Repository (CRAN) — Configuring Repositories

Packages (separate multiple with space or comma):

Install to Library: C:/Users/rvosy/Documents/R/win-library/4.1 [Default]

✓ Install dependencies

Install | Cancel

Required R libraries

```
setwd("C:/Users/xxxx/Desktop
/Sequence analysis seminar/")

library(TraMineR)
library(cluster)
library(RColorBrewer)
library(foreign)
library(WeightedCluster)
```

TraMineR is the main sequence analysis package, however, few additional ones are needed for cluster analysis and for making “cool” looking graphs

Cluster and WeightedCluster are two cluster analysis packages

RColorBrewer helps coloring the charts to make them more readable

Reading the data

```
maindata <-
read.delim("SAwebinar.dat")
View(maindata)
names(maindata)
```

The first command reads the data and creates a data frame object in R

The second command lets you view the data

The last command prints the names of variables in the console

Alphabet, long labels, short labels

```
alphabet.RS.seq <- c("1", "2", "3")

RS.labels <- c("Lives with parents", "Lives in temporary accomodation", "Lives in self-owned accomodation")

RS.shortlab <- c("LWP", "LTA", "LSOA")
```

***note that the you can name the object in any way you like, but try using the names that you will later remember

In order to build a sequence object (a data frame that contains sequences, which we will later analyze), we first need to create three additional objects that will contain some important information for the sequence object

Alphabet. Specifying the alphabet means that you state the values that will appear in the sequences. If you create a sequence object without specifying the alphabet option, all possible states are supposed to be present in the data set and the alphabet is set by listing the distinct states encountered. However, in some cases, we may have to consider states that are not present in the data set used to create the sequence object. In the example dataset sequences have three distinct values: 1, 2, 3. The first command states that these values will appear in the sequences.

Short labels. These ussually are the short names for the possible values that may appear in the sequences. These areused in some particular charts or dataframes, which will be created later.

Long labels. Just another type of labels that may be used in some particular charts.

One more object...

```
xvalues25 <- c("finished school", "0.5 y.", "1 y.", "1.5 y.","2 y.", "2.5", "3 y.", "3.5 y.", "4 y.", "4.5 y.","5 y.", "5.5 y.", "6 y.", "6.5 y.")
```

Specifically for this analysis we will create an additional objects, which will contain the names of the columns in several charts that we will create later.

The labels here represent a specific period of life course that were assessed in the study: the moment when participants finished high school, then six months later and so on...

…and now, the most important object! Defining a sequence object

```
RS.seq <- seqdef(maindata,
var= 2:15, label=RS.labels,
states=RS.shortlab,
alphabet=alphabet.RS.seq)
```

```
View(RS.seq)

names(RS.seq)
```

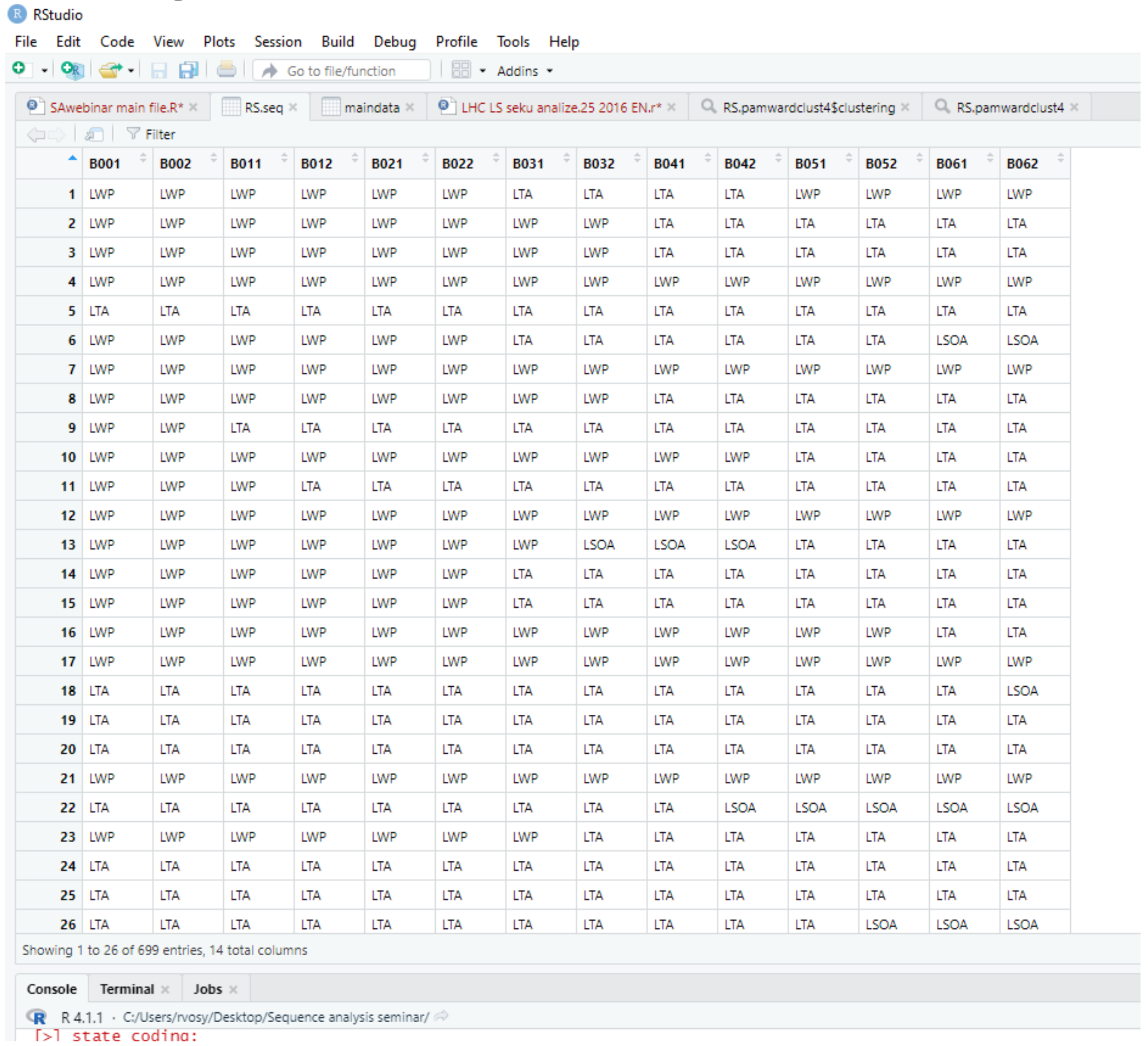
Seqdef is the command that defines and creates the state sequence object

- The first argument specifies the dataset object
- The second argument (var=…) specifies with columns represent a sequence
- The third argument (states) specifies long labels
- The fourth – specifies short labels
- Fifth – the alphabet

Viewing the sequence object

If everything went well so far, then you should see the new window in R, which looks something like this →

Note that the short labels are presented instead of actual variable values



RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
SAwebinar main file.R*
RS.seq
maindata
LHC LS seku analize.25 2016 EN.r*
RS.pamwardclust4$clustering
RS.pamwardclust4
Filter

	B001	B002	B011	B012	B021	B022	B031	B032	B041	B042	B051	B052	B061	B062
1	LWP	LWP	LWP	LWP	LWP	LWP	LTA	LTA	LTA	LTA	LWP	LWP	LWP	LWP
2	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LTA	LTA	LTA	LTA	LTA	LTA
3	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LTA	LTA	LTA	LTA	LTA	LTA
4	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP
5	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA
6	LWP	LWP	LWP	LWP	LWP	LWP	LTA	LTA	LTA	LTA	LTA	LTA	LSOA	LSOA
7	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP
8	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LTA	LTA	LTA	LTA	LTA	LTA
9	LWP	LWP	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA
10	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LTA	LTA	LTA	LTA
11	LWP	LWP	LWP	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA
12	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP
13	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LSOA	LSOA	LSOA	LTA	LTA	LTA	LTA
14	LWP	LWP	LWP	LWP	LWP	LWP	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA
15	LWP	LWP	LWP	LWP	LWP	LWP	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA
16	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LTA	LTA
17	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP
18	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LSOA
19	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA
20	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA
21	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LWP
22	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LSOA	LSOA	LSOA	LSOA	LSOA
23	LWP	LWP	LWP	LWP	LWP	LWP	LWP	LTA	LTA	LTA	LTA	LTA	LTA	LTA
24	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA
25	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA
26	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LTA	LSOA	LSOA	LSOA

Showing 1 to 26 of 699 entries, 14 total columns

```
R 4.1.1 · C:/Users/rvosy/Desktop/Sequence analysis seminar/
[>] state coding:
```

Lets attribute some colors to different states

```
attr(RS.seq, "cpal") <-
c("#7b3d17", "#565a3c",
#355d7e")
```

I picked up three different color codes from the https://colorbrewer2.org.

If you would like to use some different colors, please visit the page and choose the colors you like

Now we are ready to build some cool looking graphs

State distribution plot

```
seqdplot(RS.seq, withlegend = T,
border = T, xtlab=xvalues25)
```

The seqdplot() function plots a graphic showing the state distribution at each time point (the columns of the sequence object).

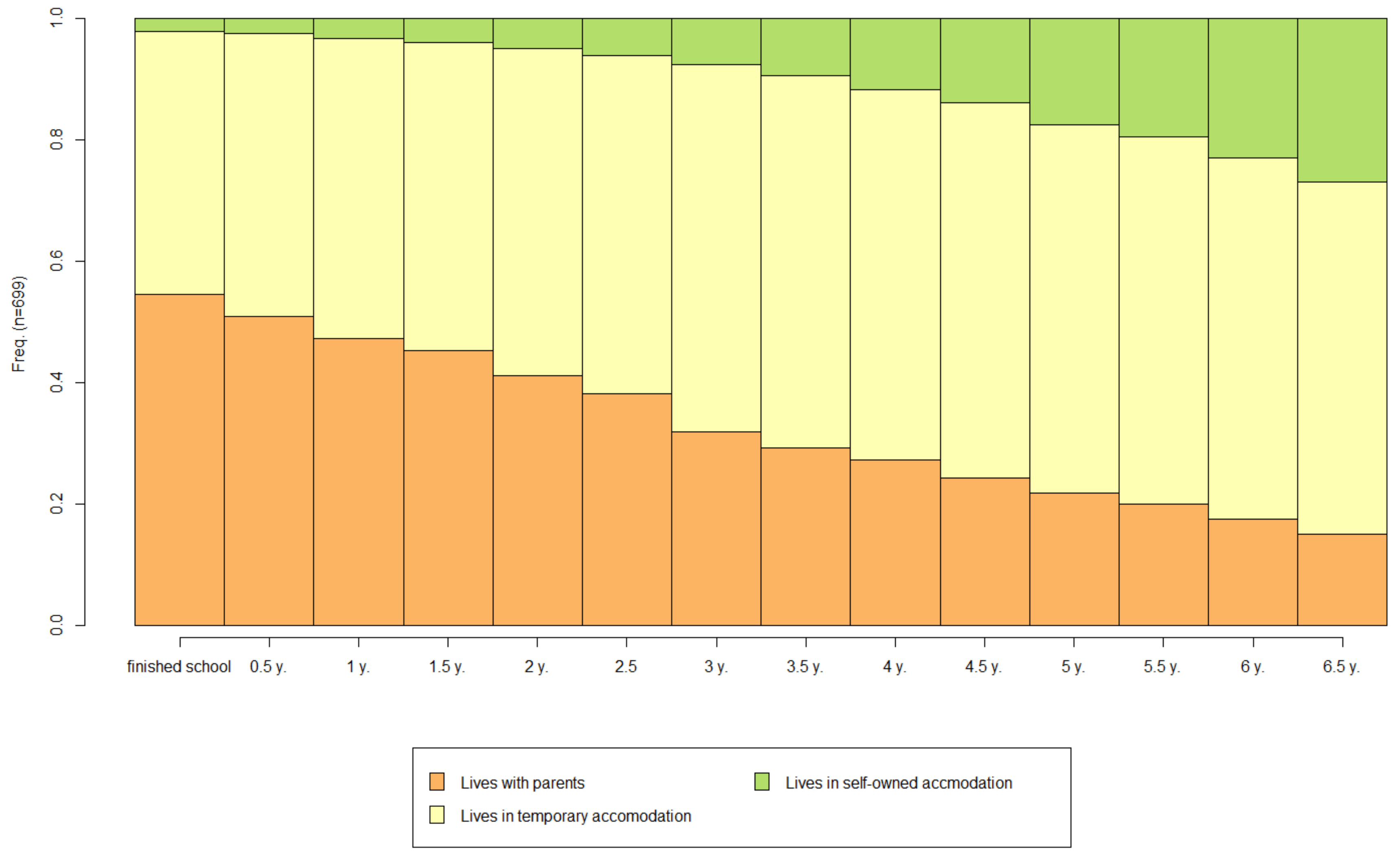
The first arguments specifies the sequence object

With the second one you can specify if you want a legend or not

The third object specifies if you want column borders to be visible or not

The fourth argument specifies the column labels (X-axis values)

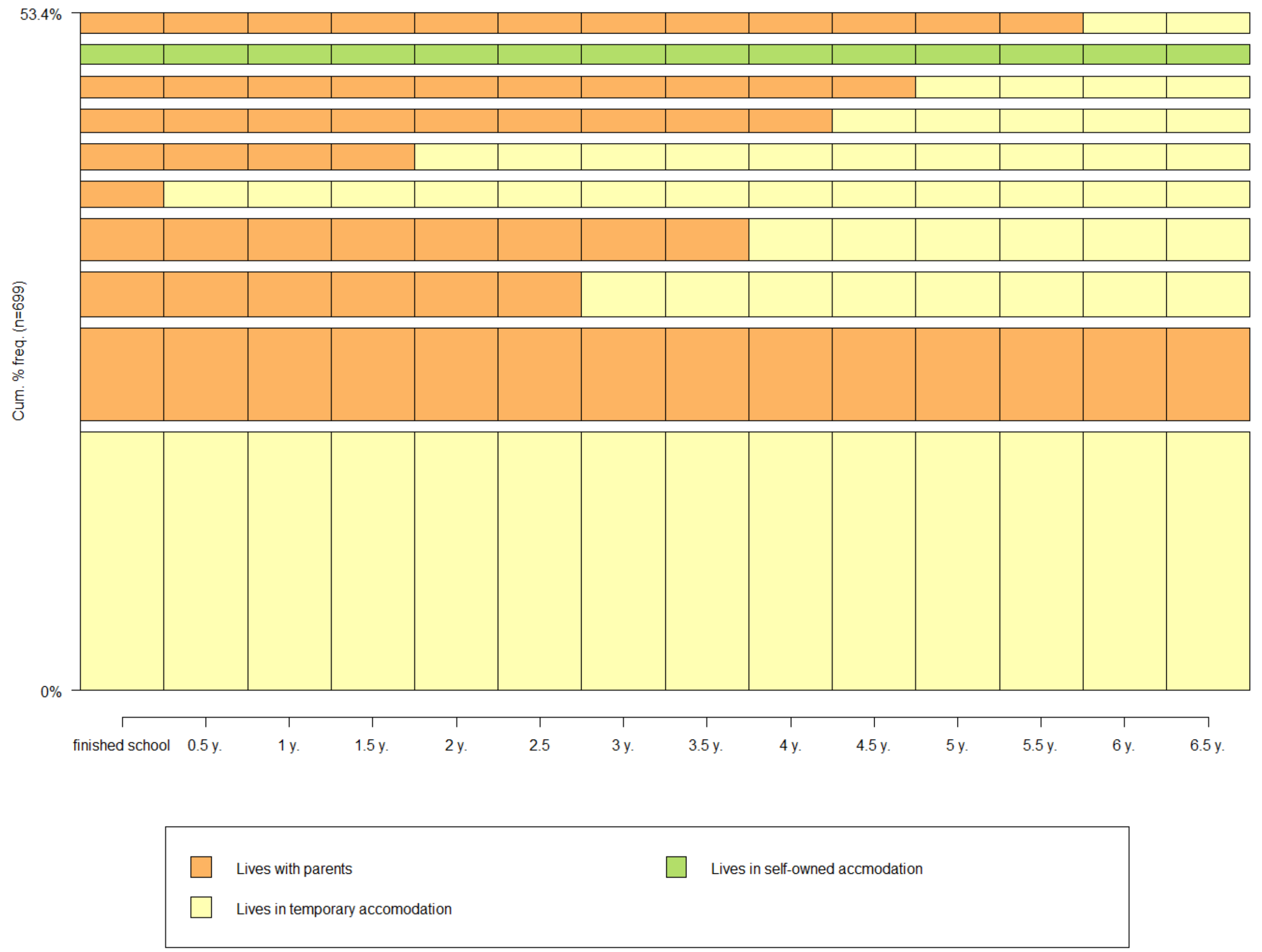
State distribution plot



This is what you should see if everything went well in the previous steps

Plot 10 most frequent sequences

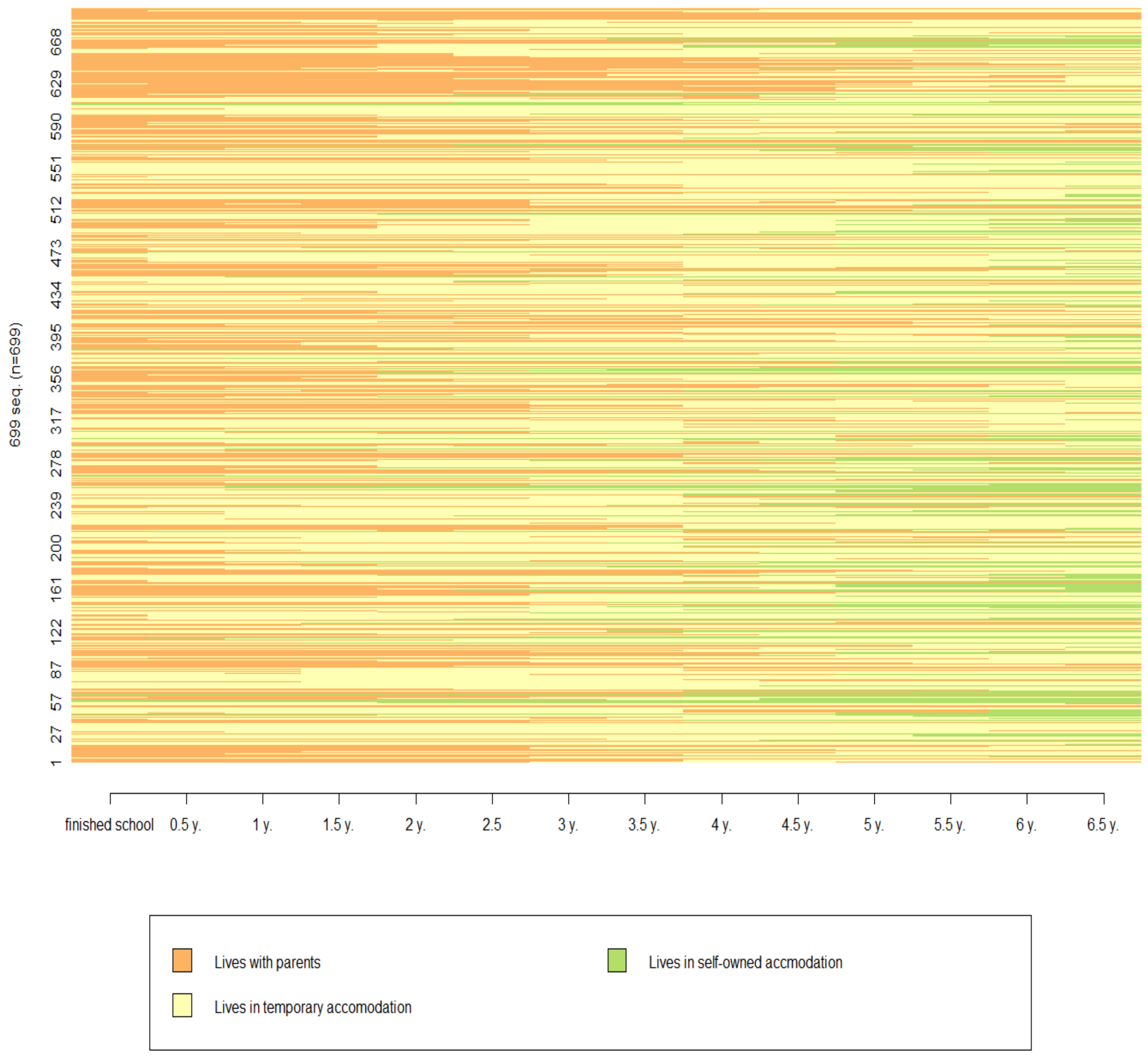
```
seqfplot(RS.seq,
withlegend = T,
xtlab=xvalues25)
```



Plot ALL sequences

seqIplot(RS.seq, withlegend = T, xtlab=xvalues25, border = NA)

**note that I ask to remove borders

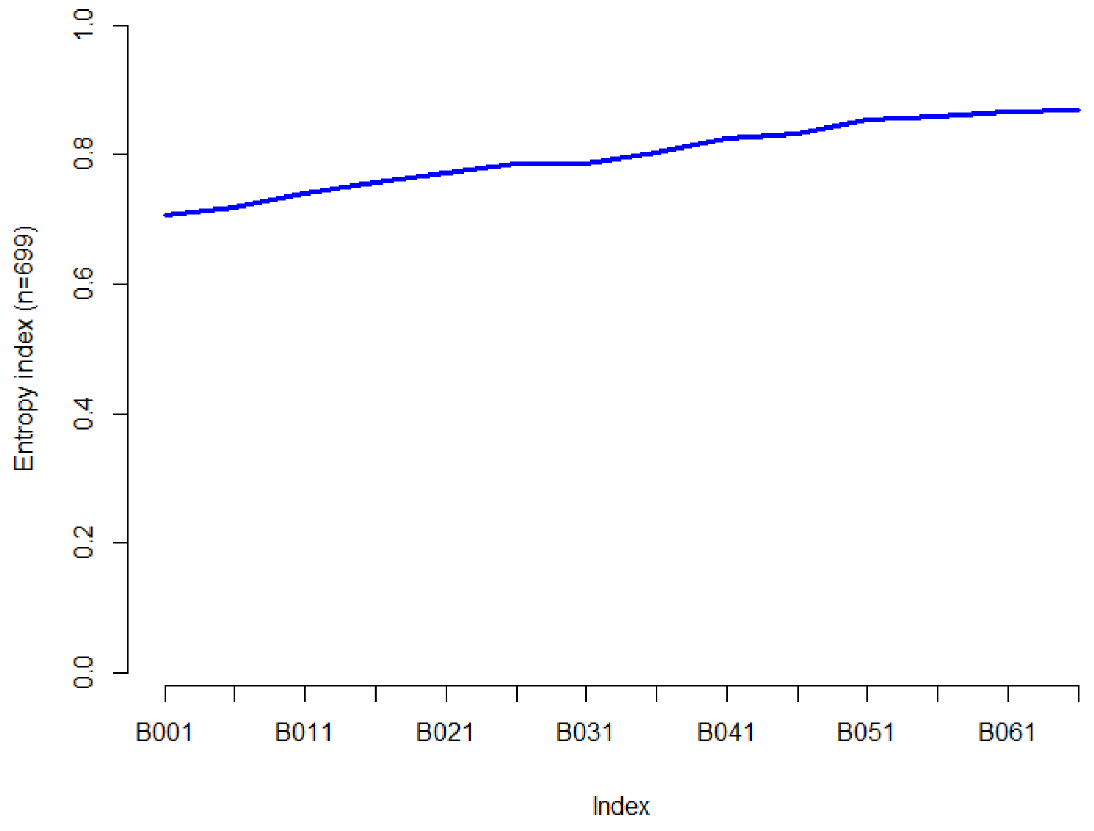


Within column entropy

seqHtplot(RS.seq)

**entropy close to 1 indicates that all states are equally distributed at some specific time (column)

**entropy close to 0 indicates that at some particular time (column), only one state is present



Characteristics of individual sequences

seqtransn(RS.seq)

- This command prints out how many times individual has moved from one state to another

seqient(RS.seq)

- This command displays within row entropy (interpretation is the same as for columns)

seqST(RS.seq)

- This command displays turbulence characteristic for each sequence. Turbulence is a bit hard to define, but it is good to think of it as a level of chaos in the sequence.

Building a typology of sequence

Preliminary steps are:

1. Calculate transitions rates
2. Create transition costs

Main steps are:

1. Create a distance matrix using optimal matching (OM) or other method
2. Cluster analyze the sequences using Ward’s hierarchical clustering
3. Cluster analyze the sequence again using the PAM method
4. Inspect quality and characteristics of different cluster solutions and make a decision
5. Describe the clusters

Sequence (dis)similarity metrics

1. Number of matching positions
   - How many positions in the two sequences have the same state (value)?
2. Longest common prefix
   - How long is a common prefix?
3. Longest common subsequence
   - How long is a common subsequence?
4. Optimal Matching (OM)
   - How many in/del (insert/delete) and/or subs (substitute) operations we need to do to transform one sequence to another?
5. Variants of OM

Examples of sequence (dis)similarity metrics

Method	Example	Sequence similarity
Number of matching positions	**A**-A-**B-B-C**-B-B-**C**-A-**A** **A**-B-**B-B-C**-C-C-**C**-C-**A**	6 matching positions, so 10-6=4 Distance is 4
Longest common prefix	**A**-A-B-B-C-B-B-C-A-A **A**-B-B-B-C-C-C-C-C-A	1 state is the same in the prefix, so distance is 10-1=9
Longest common subsequence	A-A-**B-B-C**-B-B-C-A-A A-B-**B-B-C**-C-C-C-C-A	Distance = 7
Classical Optimal Matching (OM)	A-A-B-B-C-B-B-C-A-A A-B-B-B-C-C-C-C-C-A A-B-A-B-A-B-A B-A-B-A-B-A-B	Based on my calculations ~ 8, assuming in/del and substitute costs are set to 2

Specifics of OPTIMAL MATCHING

To use OPTIMAL MATCHING, we need to define the “costs” (points given) of each operation of in/del or substitute.

Traditionally in/del costs are set to 2 (1 for insert and 1 for delete)

Substitute costs, however, are set by:

- Using the value of 2 (because it is equal to in/del costs)
- Using the theoretical criteria
- Using the information about transition rates

NP-NP-NP-C-C

NP-NP-NP-NP-C

NP-NP-NP-NP-M

We will focus on the third options as this is the one used most often

Calculating transitions rates

First, we need to calculate transitions rates and create an object that stores these transition rates

```
RS.trate <- seqtrate(RS.seq)
Output:
           [-> LWP]    [-> LTA]    [-> LSOA]
[LWP ->]  0.877310389 0.103569152 0.01912046
[LTA ->]  0.021297574 0.955630053 0.02307237
[LSOA ->] 0.001138952 0.003416856 0.99544419
```

This command would create a new object, which we use later in OM command

Transitions rates are important for measuring distances between the sequences

Calculating transitions costs

Then, we need to calculate transitions costs and create an object that stores these transition costs

```
RS.seq.scost <- seqsubm(RS.seq, method = "TRATE")
View(RS.seq.scost)
```

```
         LWP->     LTA->      LSOA->
LWP->  0.000000 1.875133 1.979741
LTA->  1.875133 0.000000 1.973511
LSOA-> 1.979741 1.973511 0.000000
```

and now… we create OM distance matrix

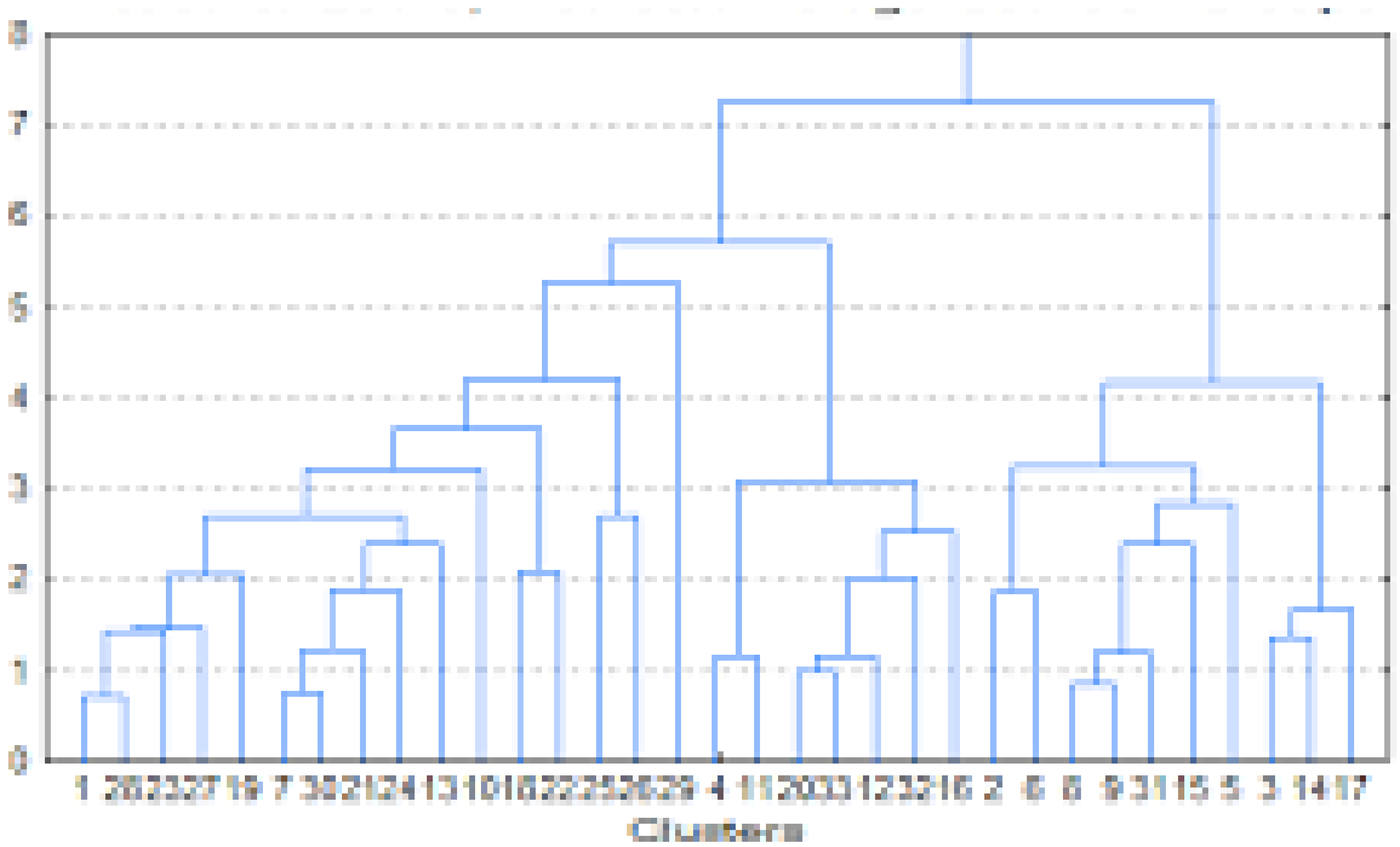
```
RS.sec.full.distOM <- seqdist(RS.seq, method="OM", norm=TRUE,
indel=1, sm=RS.seq.scost, full.matrix=TRUE)
View(RS.sec.full.distOM)
```

This will create a huge matrix (rows = columns = N of sequence) and store it as an object called RS.sec.full.distOM

Running hierarchical cluster analysis with Ward’s algorithm

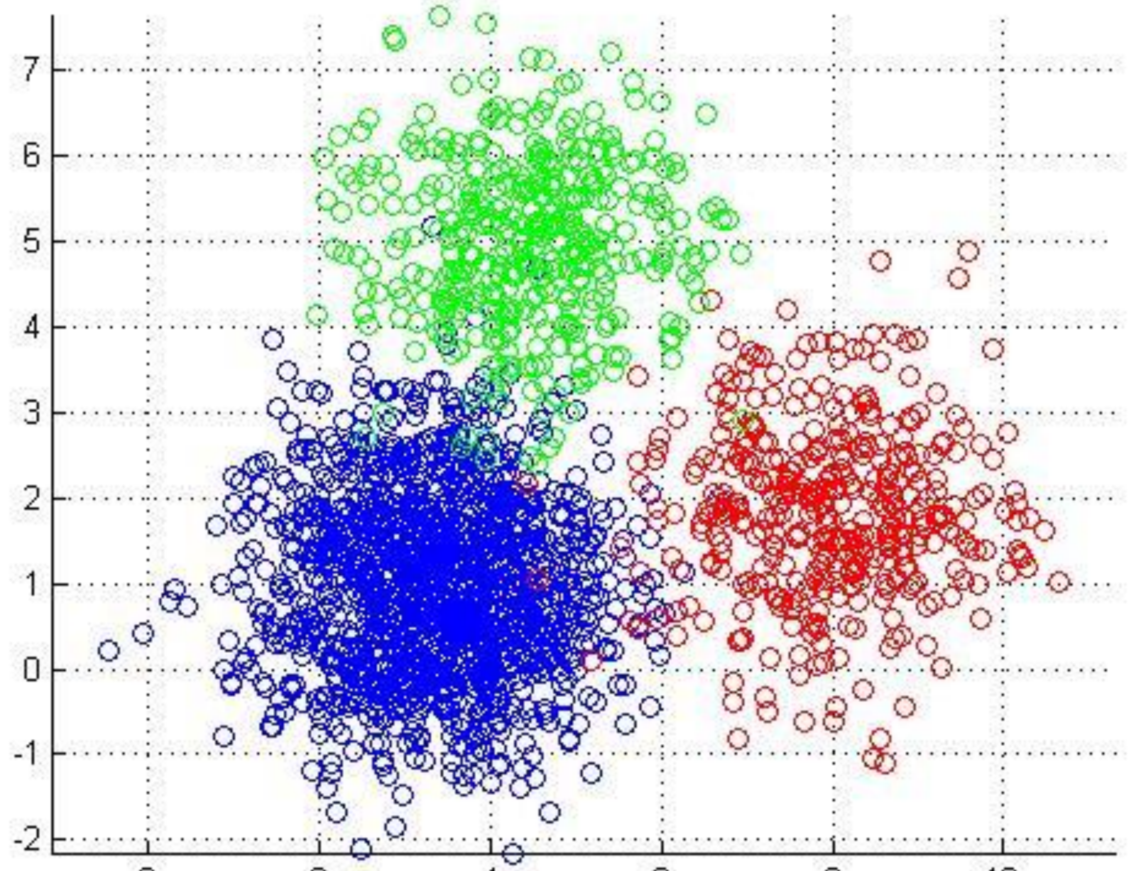
```
RS.wardCluster <- hclust(as.dist(RS.sec.full.distOM), method = "ward")
RS.wardTree <- as.seqtree(RS.wardCluster, seqdata=RS.seq,
diss=RS.sec.full.distOM, ncluster=8)

RS.wC.clust8 <- cutree(RS.wardCluster, k = 8)
RS.wC.clust7 <- cutree(RS.wardCluster, k = 7)
RS.wC.clust6 <- cutree(RS.wardCluster, k = 6)
RS.wC.clust5 <- cutree(RS.wardCluster, k = 5)
RS.wC.clust4 <- cutree(RS.wardCluster, k = 4)
RS.wC.clust3 <- cutree(RS.wardCluster, k = 3)
RS.wC.clust2 <- cutree(RS.wardCluster, k = 2)
```



8
7
6
5
4
3
2
1
0
1 28 23 27 19 7 30 21 24 13 10 18 22 25 26 29 4 11 20 33 12 32 16 2 6 8 9 31 15 5 3 14 17
Clusters

K-means / PAM example



Running PAM (portioning around medoids) cluster analysis

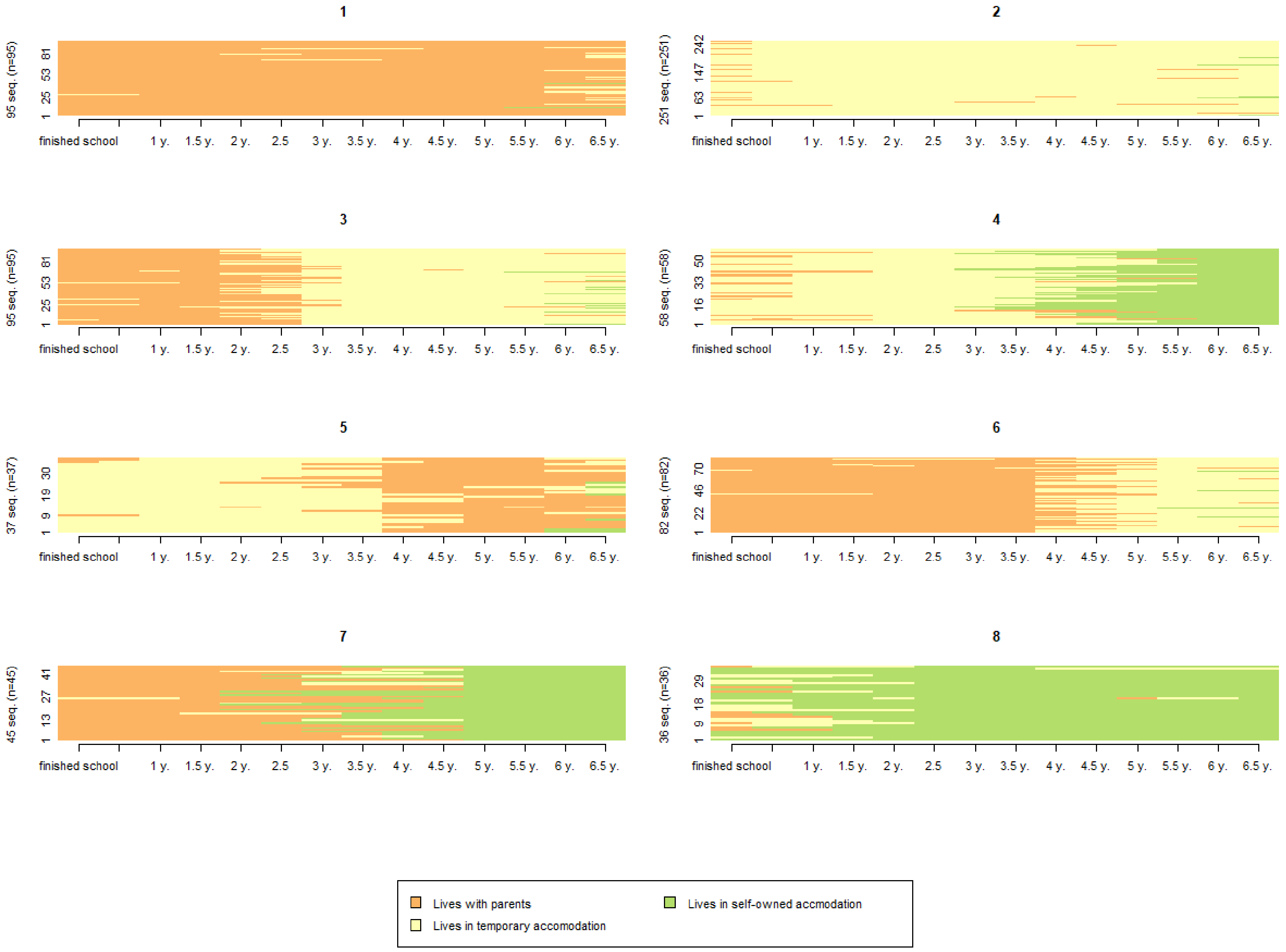
```
RS.pamwardclust2 <- wcKMedoids(RS.sec.full.distOM, k = 2, initialclust = RS.wardCluster)
RS.pamwardclust3 <- wcKMedoids(RS.sec.full.distOM, k = 3, initialclust = RS.wardCluster)
RS.pamwardclust4 <- wcKMedoids(RS.sec.full.distOM, k = 4, initialclust = RS.wardCluster)
RS.pamwardclust5 <- wcKMedoids(RS.sec.full.distOM, k = 5, initialclust = RS.wardCluster)
RS.pamwardclust6 <- wcKMedoids(RS.sec.full.distOM, k = 6, initialclust = RS.wardCluster)
RS.pamwardclust7 <- wcKMedoids(RS.sec.full.distOM, k = 7, initialclust = RS.wardCluster)
RS.pamwardclust8 <- wcKMedoids(RS.sec.full.distOM, k = 8, initialclust = RS.wardCluster)
```

Checking the quality of cluster solutions

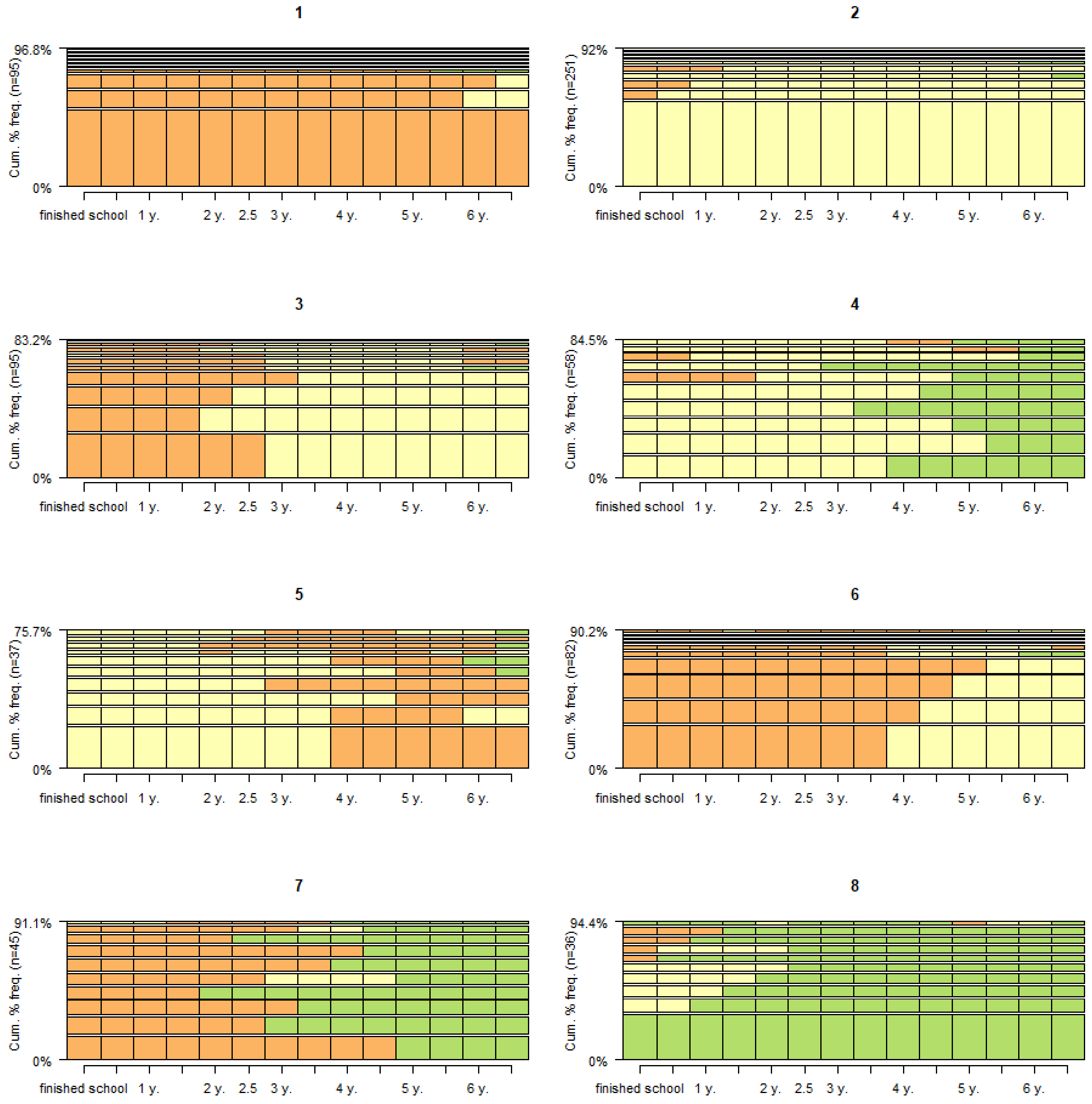
```
RS.pamwardclust2$stats
RS.pamwardclust3$stats
RS.pamwardclust4$stats
RS.pamwardclust5$stats
RS.pamwardclust6$stats
RS.pamwardclust7$stats
RS.pamwardclust8$stats
```

```
> RS.pamwardclust2$stats
      PBC          HG        HGSD         ASW        ASWw          CH          R2        CHsq        R2sq
0.5963015   0.6996396   0.6971913   0.4990412   0.5004807 373.8078249   0.3490896 703.0744734   0.5021693
       HC
0.1437818
```

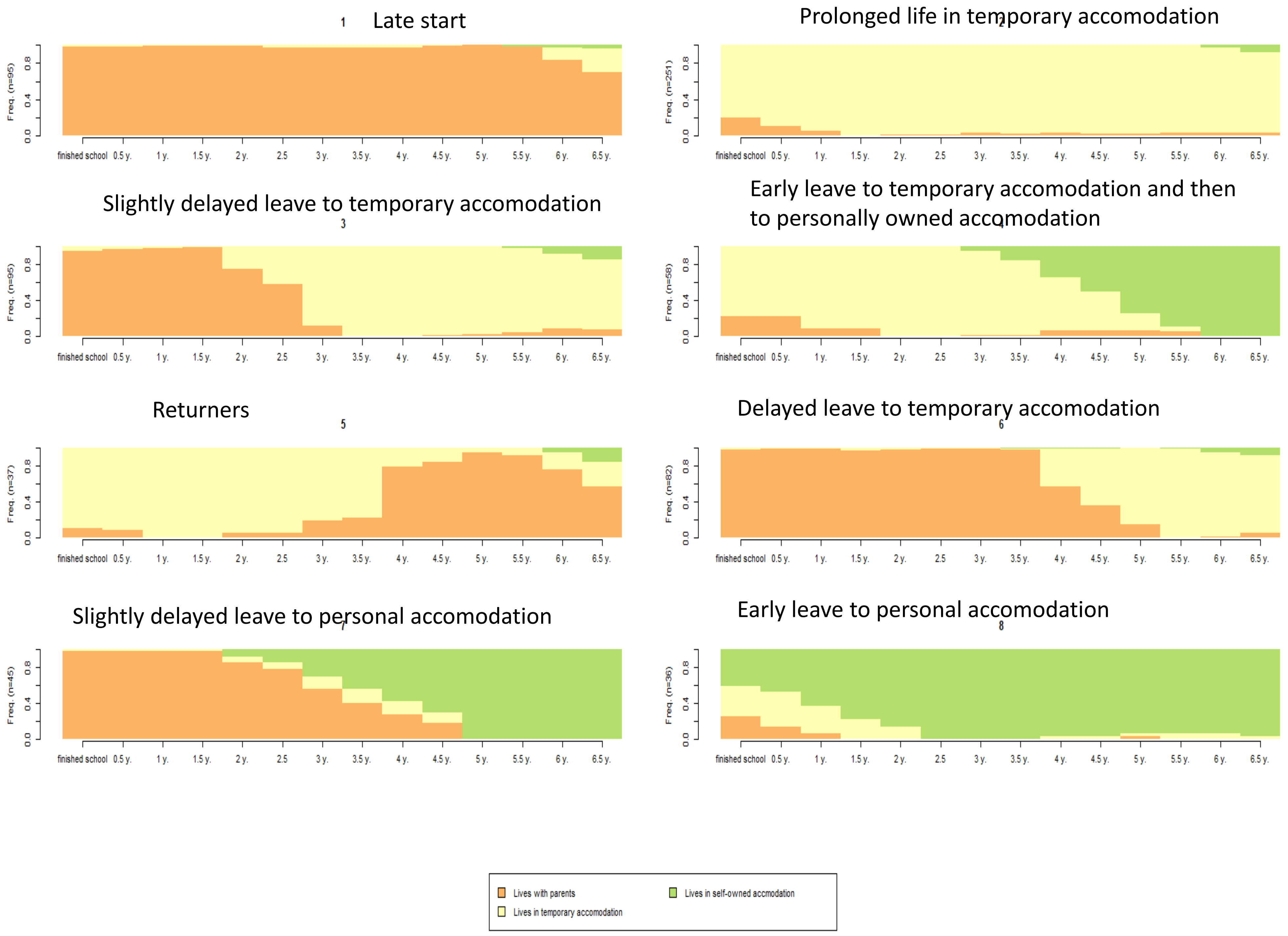
seqIplot(RS.seq, group=maindata $RSPATHS, xtlab=xvalues25, border = NA)



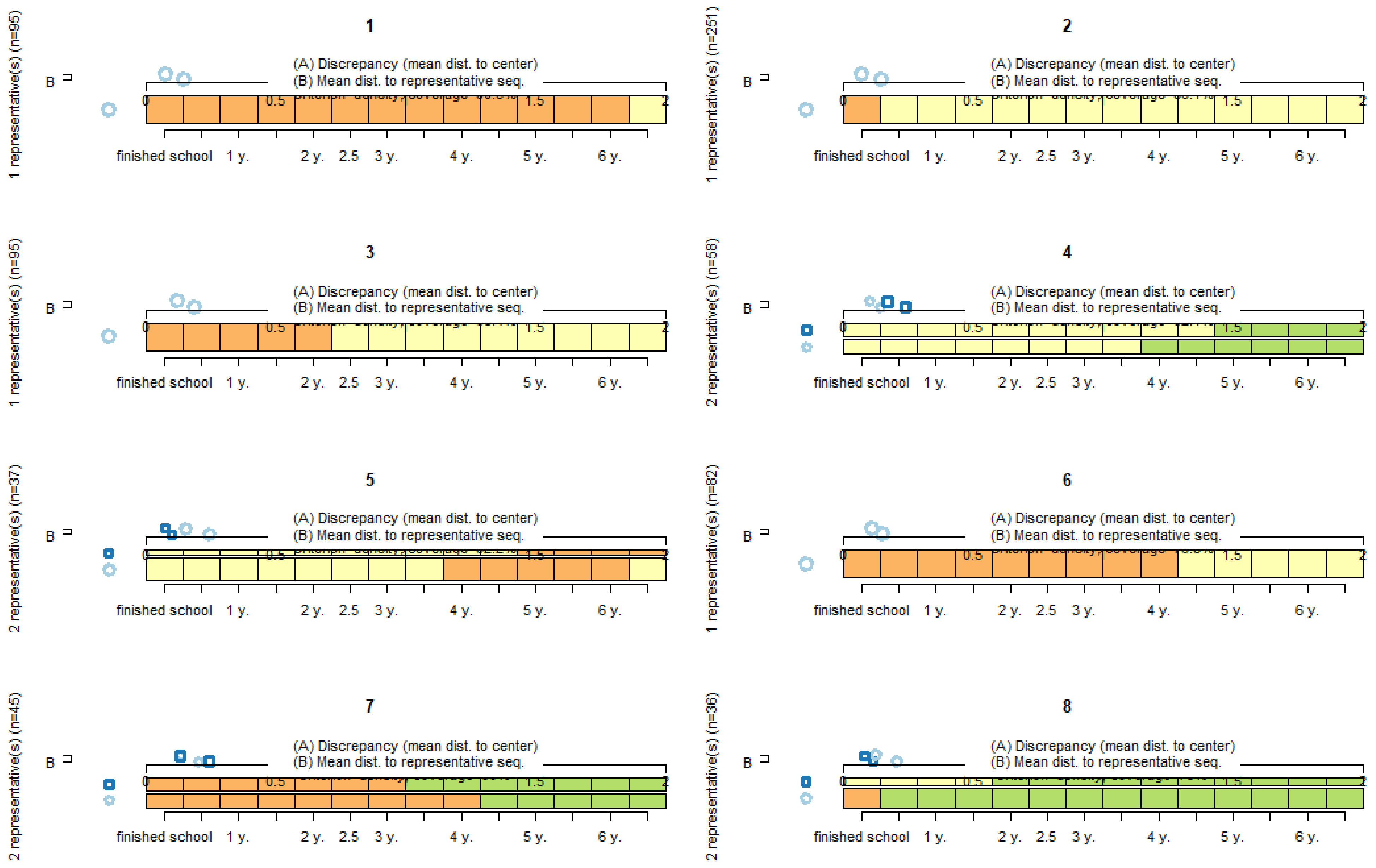
```
seqfplot(RS.seq,
group=maindata$RSPATHS,
xtlab=xvalues25, border = NA)
```



```
seqdplot(RS.seq,
group=maindata
$RSPATHS,
xtlab=xvalues25,
border = NA)
```



```
seqrplot(RS.se
q, criterion =
"density",
withlegend = F,
group=mainda
ta$RSPATHS,
dist.matrix =
RS.sec.full.dist
OM, tsim =
0.10, trep =
0.5,
xtlab=xvalues2
5, cex.plot =
0.1)
```



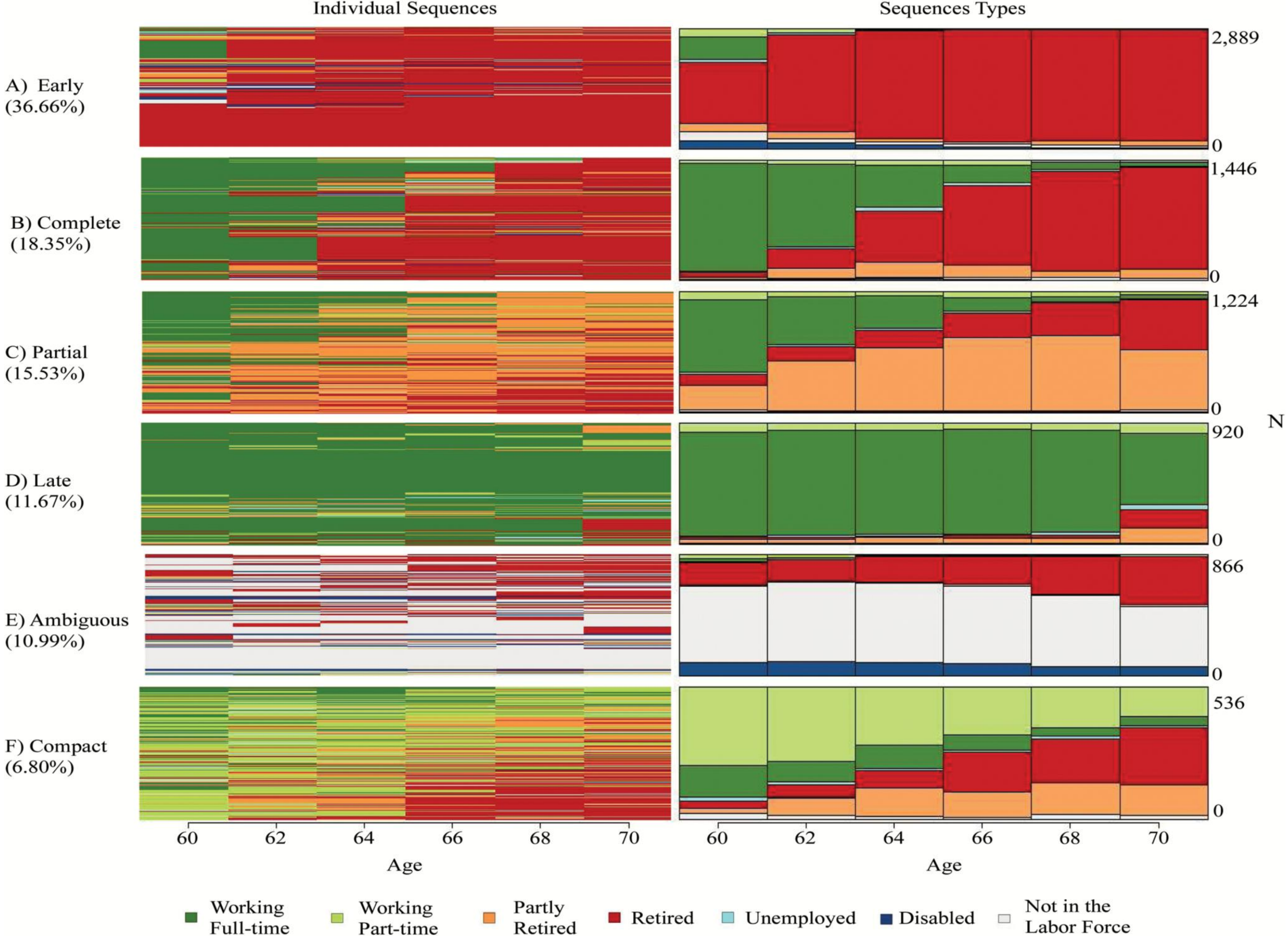
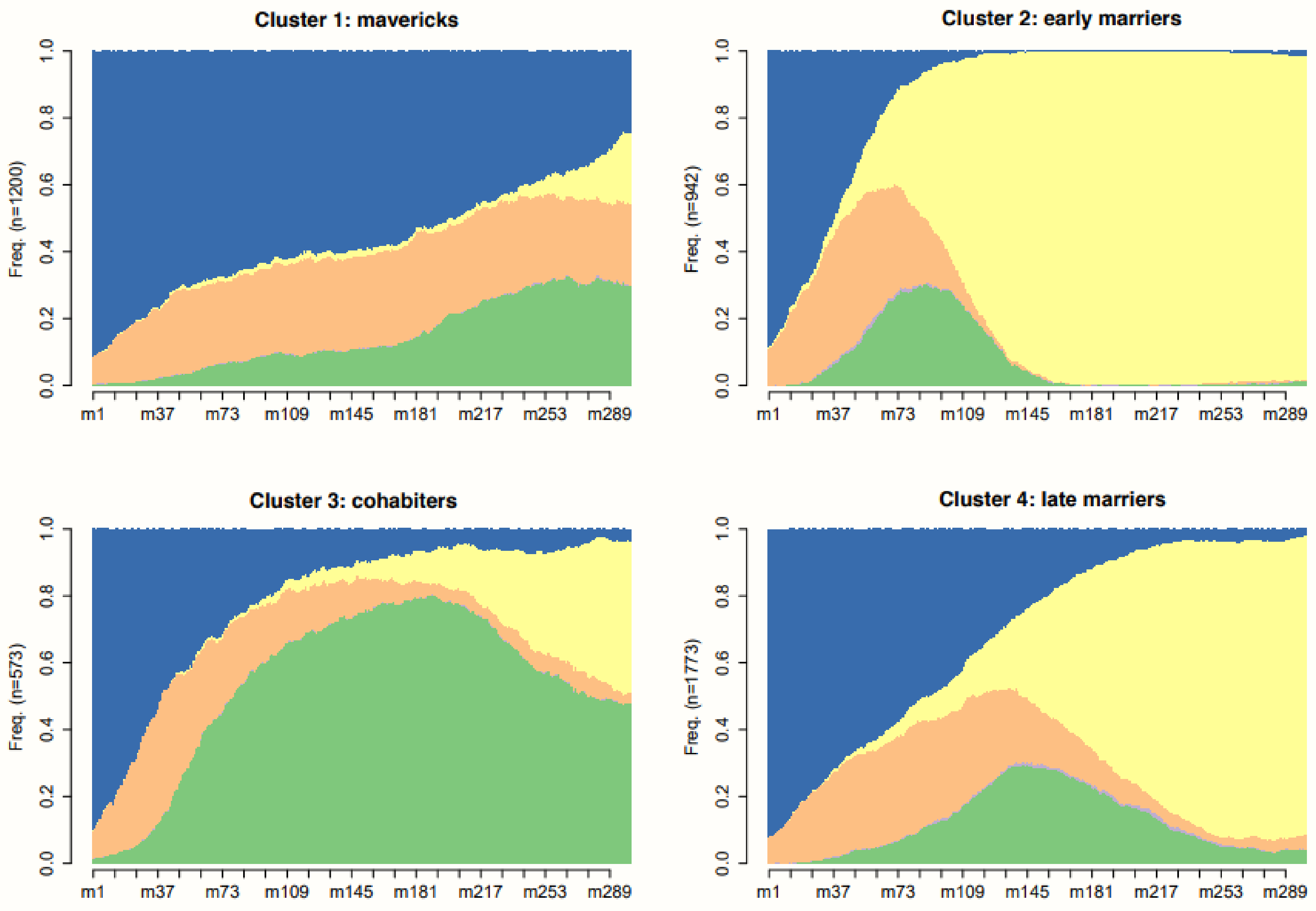
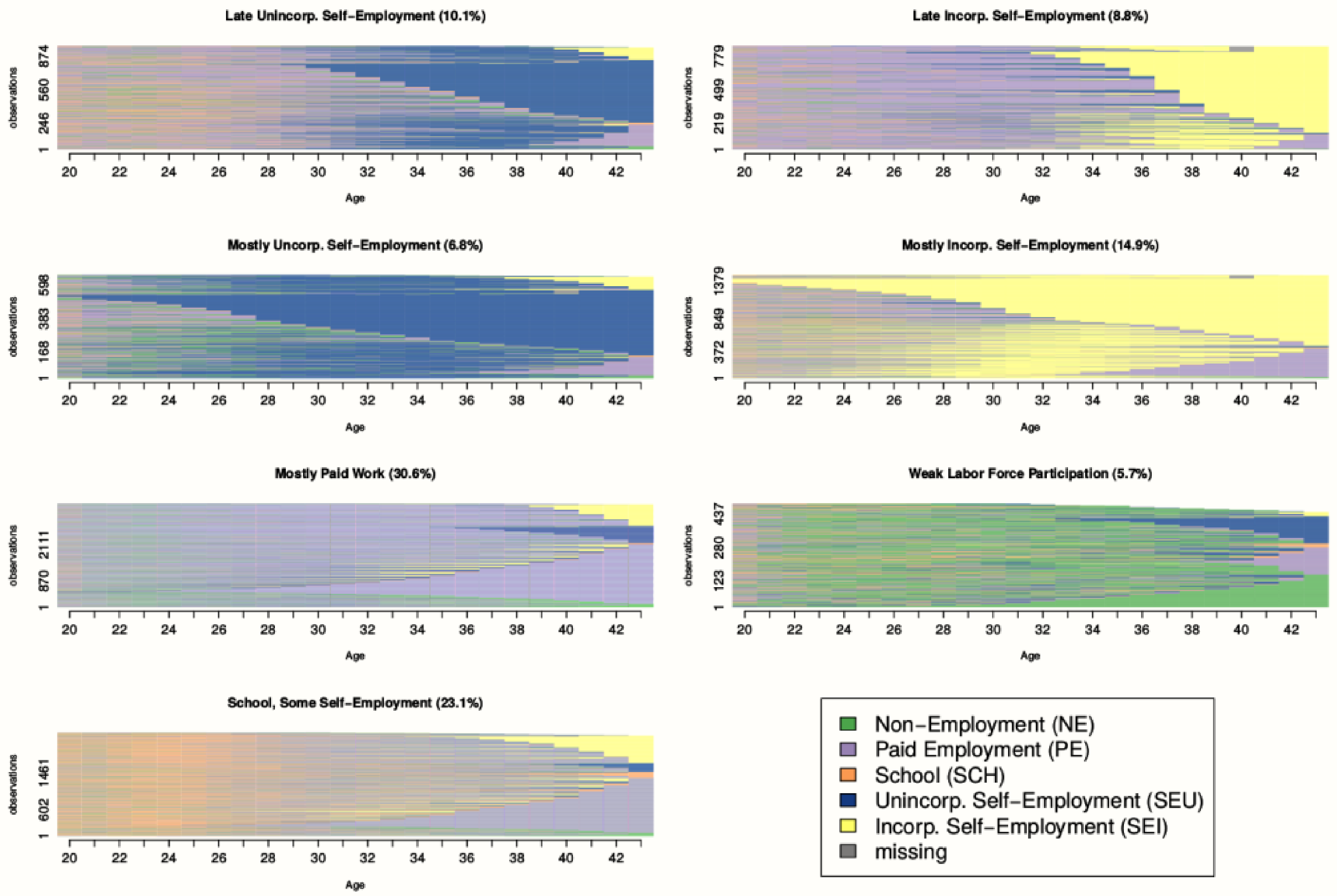

Individual Sequences
Sequences Types
A) Early (36.66%)
B) Complete (18.35%)
C) Partial (15.53%)
D) Late (11.67%)
E) Ambiguous (10.99%)
F) Compact (6.80%)
2,889
1,446
1,224
920
866
536
0
N
60
62
64
66
68
70
Age
Working Full-time
Working Part-time
Partly Retired
Retired
Unemployed
Disabled
Not in the Labor Force

Figure 10: Distribution of partnership statuses in clusters 1–4



Source: Pairfam waves 1–6, own research.

Figure 3: Clusters of Life Cycles Involving Self-Employment (1970 birth cohort)



Notes: Figure shows life employment profiles of all Swedish males born in 1970 who are ever self-employed between 1990 and 2013.

Additional possibilities

If you measure changes in multiple life domains, you can opt for „holistic analysis“:

- Using „multi-channel“ sequences
- Using latent class analysis to distinguish role/state configurations and following up with sequence cluster analysis

If you have large datasets, you can uncover typologies using latent class analysis and then follow up with sequence analysis to describe clusters. Note that, however, sequence analysis can also be used and it does not require large sample or does not make any assumptions about the data distributions

Main references

http://traminer.unige.ch/