A Practical Introduction to the Family and Working Lives Survey

Götz Rohwer

May 1996

Report prepared for the Department for Education and Employment, London.

Author's address: Götz Rohwer, Max Planck Institute for Human Development and Education. Lentzeallee 94, 14195 Berlin, Germany.
Email: rohwer @ mpib-berlin.mpg.de.

Contents

1 Introduction

The Family and Working Lives (FWL) survey provides researchers with representative information about people living in Britain today. The focus of the survey is not only on current living conditions; in addition, the survey includes a broad variety of retrospective questions about family and working lives of the respondents. In this way, the survey tries to get information about basic event histories beginning with the 16th birthday of the respondents. The fieldwork for this survey was done in 1994 and 1995. The final sample consists of information for 11237 respondents.

This report provides a *practical* introduction to the currently available data. The focus is on preparing the raw data for cross-sectional and longitudinal analyses with the computer program TDA (Transition Data Analysis). This program has been specifically designed for statistical analyses with longitudinal data but can also be used with cross-sectional data.[1] Another advantage of TDA is that data can be stored in a highly compressed data archive that can be used without further need to decompress the data separately. However, using the FWL data is in no way restricted to using this program. The data files that will be discussed in subsequent sections can also be used with many other statistical packages.

The report consists of the following sections.

- Section 2 describes the FWL raw data file that is currently available. It is one big data file (about 68 Mega bytes) and consists of several different record types. It would be difficult, therefore, to use this raw data file directly. Instead, our strategy will be to split this raw data file into several separate files that can be used more easily for different purposes.
- Section 3 deals with cross-section data. It is shown how to create a new data file containing all cross-sectional record types of the FWL raw data file. Then it is explained how to create a TDA data archive, and finally there will be some examples illustrating how to use the archive.
- Section 4 discusses how to create event history data based on the respondent's event matrix. We create a new data file, containing the

[1] See the preliminary report on using TDA, based on a pilot sample for the FWL survey (Rohwer, 1994). An introduction to using TDA for event history analysis can be found in Blossfeld and Rohwer (1995).

information given in the respondent's event matrix and add this data file to our archive. Then we illustrate using the data file with some examples.

- Section 5 discusses the basic information about respondent's partners.
- Section 6 describes how we have created an event history data file based on the information in the partner's event matrix. It is further explained how the new data file is added to our data archive and its usage is illustrated with examples.
- Section 7 explains how we have created an event history data file for the respondent's job histories provided in their so-called *job grid*.
- Section 8 describes an analogously created job history file based on the partner's job grid data. This section includes a description of the final data archive.

2 The Raw Data

The currently available FWL raw data are contained in one big ASCII data file, subsequently called `fwl2.dat`. The size of this data file is 68,406,959 bytes. The file contains 1,551,086 records for 11237 respondents.

Box 2.1 Data for the first person in the FWL raw data file

```
999900010011001  0001001 1PIKE825095012000100100000000100015160930681010680703196920 3  26   09990
999900010011002001                          1 12    026   027026            3 33         2
99990001001100300 2        1   1 1          1 11
99990001001100400         00       1          1   1           6       2 31   1  11 12
99990001001100500     1                                                   031
99990001001100600 16154612161605208211166                    10     0802
99990001001100700                     87     9394
999900010011008008710            93089402
99990001001100900                               101001
99990001001101000        888990                                               881089109001
99990001001101100
99990001001101200
99990001001101300
99990001001101400                         017   017   017
99990001001101500
99990001001101600                          133                       050303
99990001001101700                            444
99990001001101800
99990001001101900
99990001001102000                                                              333
99990001001102100            222221                    208942  41 02192
99990001001102200    1628341162856 1           2  2
99990001001102300                                                                2
99990001001102400
99990001001102500                               10            16333311633391   00011004
99990001001102600000200049  0000   3311      1                   2      2
99990001001102700     111 2    2                           2       2              2
99990001001102800222   1 11              3             1639031163908
99990001001102900          2             1639161163921            65  194        1    1
9999000100110300000        2               2    2
99990001001103100
99990001001103200                            16434711643502
99990001001103300
99990001001103400                                  13164505
99990001001103500
99990001001103600
99990001001103700
99990001001103800
99990001001103900                                                            16450512164523
99990001001104000
99990001001104100164523 12164537
99990001001104200                   2216454811080-         165213
99990001001104300
99990001001104400                              165213    1       04 114216501981 01
99990001001104500                                             10.315
99990001001105200                  45
99990001001105300
99990001001105400
08870001001106384I10                     U10          S10                                 08250PIKE
08870001001106386                                     S07         A07                     08250PIKE
08870001001106387                        W07U10       U10         A07                     08250PIKE
08870001001106388                        W08U09                                           08250PIKE
08870001001106389                        W08U09                                           08250PIKE
08870001001106393                        W08          U06         A06A08                  08250PIKE
08870001001106401  11 0787108705  22       89900000 22 000622311                          08250PIKE
08870001001106402  11 0888098805  22       80900000 22 020015611                          08250PIKE
08870001001106403  11 0889098905  22       80900000 22 010015331                          08250PIKE
08870001001106404  11 0893999998  11       22000000 22 200085111                          08250PIKE
```

Box 2.1 (continued) Data for the first person in the FWL raw data file

```
088700010011064 98     10840687
088700010011064 98     10880789
088700010011064 98     10890793
088700010011064 98     11870788
088700010011070 01    3                  1          1
088700010011070 04  12          2        1          1
088700010011070 98
088700010011070 98
088700010011070 98
088700010011070 98
088700010011071 01
088700010011071 02
088700010011071 03
088700010011071 04
088700010011071 98
088700010011071 98
088700010011071 98
088700010011071 98
9999000100110 74               3                                        2
9999000100110 75      5                                     1234  7      3          456789  1        0
9999000100110 76
9999000100110 77
9999000100110 78
9999000100110 79
9999000100110 80
9999000100110 81
9999000100110 82
9999000100110 83
9999000100110 84          8   2                             1                    1  4
9999000100110 85               3                                                   3
9999000100110 86                                            1
9999000100110 87
9999000100110 88
9999000100110 89
9999000100110 90
9999000100110 91
9999000100110 92
9999000100110 93
9999000100110 94                                             2
9999000100110 95
9999000100110 96                                                                    4
9999000100110 97                             6
9999000100110 98
9999000100110 99
9999000100111 00                                                                 1
9999000100111 01
9999000100111 02
9999000100111 03                                            1          1
9999000100111 04
9999000100111 05
9999000100111 06
9999000100111 07
9999000100111 08
9999000100111 09
9999000100111 10
9999000100111 11
9999000100111 12
9999000100111 13
9999000100111 14
9999000100111 15
9999000100111 16
9999000100111 17
9999000100111 18
9999000100111 19
9999000100111 20
9999000100111 21
9999000100111 22
9999000100111 23
9999000100111 24
9999000100111 25
9999000100111 26
9999000100111 27
9999000100111 28
9999000100111 29
9999000100111 30
```


Box 2.2 Record types for event history data

```
Record Type
--------------------------------------
     63     Respondent's event matrix
     64     Respondent's job grid
     66     Partner's event matrix
     67     Partner's job grid
     70     Respondent's job grid (continued)
     71     Respondent's job grid (continued)
     72     Partner's job grid (continued)
     73     Partner's job grid (continued)
```

The records have variable length; end of record characters conform to DOS conventions, that is, each record is terminated by a carriage return-line feed sequence. Values of variables are stored in a fixed format (positions can be found in the codebook).

Each record is identified by two pieces of information: An 8-digit identifier for respondents (columns 5 – 12), and a 3-digit record type (columns 13 – 15). Box 2.1 shows the raw data for the first respondent (ID = 00010011).

Records types are in the range 1 – 130. There are two groups of records. In the first group there is exactly one record for each respondent. Subsequently we will refer to these records as *cross-sectional records*. The record types in this group are: 1 – 45, 52 – 54, and 74 – 130, altogether 105 records that are available for each respondent.

The second group consists of special records used to store event history information for respondents and their partners. The record types are in the range 63 – 73, see Box 2.2. Subsequently, we will refer to these records as *event history records*. Contrary to cross-sectional records, there may be a variable number of event history records for each respondent (including his or her partner), and some of the record types shown in Box 2.2 may be totally absent. Consequently, the total number of records varies across respondents.

Our strategy to cope with this specific structure will be to split the raw data file into separate data files with a more simple data structure. The first task will be to generate a new data file that contains all cross-sectional record types. The result will be a simple rectangular data file that can be used in a straightforward way with most statistical packages.

3 Cross-Sectional Data

This section describes how we have created a cross-sectional data file from the FWL raw data and how this file can be used with TDA. In particular, we show how to create a TDA data archive and explain its possible use with some examples.

3.1 The Cross-Sectional Data File

Creating a cross-sectional data file from the FWL raw data is simple, one just has to put all cross-sectional records into a separate data file. This new data file will be called `fwl2a.dat`. The basic points about this data file are as follows.

1. There are 105 cross-sectional records for each respondent. Given the number of 11237 respondents, the file contains 1,179,885 records. The records for each respondent are ordered with respect to the record type numbers shown in Box 3.1.
2. Contrary to the raw data file, `fwl2.dat`, records in the cross-sectional data file (`fwl2a.dat`) will have a fixed record length: 99 data columns plus 1 end-of-record character (line-feed).[2] Consequently, the *physical* record length is 100 bytes, and it is easy to view the data for each respondent alternatively as a sequence of 105 records, or as a single record having a physical length of 10500 bytes. The physical size of the new data file is, of course, 117,988,500 bytes.
3. To assess variables in the new data file one can use their definition in the FWL codebook. Each variable is defined, there, by two pieces of information: a record type number and the physical columns where the values of the variable can be found. For instance, the definition of the variable `S1AY` (birth year) is `1 (72-73)` meaning that this variable is in columns 72 – 73 in the records of type 1.

   It is also easy to locate the variables if we view the data for each respondent as one long record. Box 3.1 shows, for each record type number, the corresponding *offset*, so one only has to add this offset to the variable's columns. For instance, the variable `S2B_1` (sex of respondent) is defined as `2(44)`, that is, column 44 in record type 2. Since record type 2 has offset 100, this variable is located at column 144 in the combined records.

[2] This conforms to the UNIX convention, but most programs are able to use a file with this end-of-record convention also in a DOS environment.

Box 3.1 Record types and offsets in `fwl2a.dat`

RTYPE	Offset	RTYPE	Offset	RTYPE	Offset	RTYPE	Offset	RTYPE	Offset
001	0	022	2100	043	4200	089	6300	110	8400
002	100	023	2200	044	4300	090	6400	111	8500
003	200	024	2300	045	4400	091	6500	112	8600
004	300	025	2400	052	4500	092	6600	113	8700
005	400	026	2500	053	4600	093	6700	114	8800
006	500	027	2600	054	4700	094	6800	115	8900
007	600	028	2700	074	4800	095	6900	116	9000
008	700	029	2800	075	4900	096	7000	117	9100
009	800	030	2900	076	5000	097	7100	118	9200
010	900	031	3000	077	5100	098	7200	119	9300
011	1000	032	3100	078	5200	099	7300	120	9400
012	1100	033	3200	079	5300	100	7400	121	9500
013	1200	034	3300	080	5400	101	7500	122	9600
014	1300	035	3400	081	5500	102	7600	123	9700
015	1400	036	3500	082	5600	103	7700	124	9800
016	1500	037	3600	083	5700	104	7800	125	9900
017	1600	038	3700	084	5800	105	7900	126	10000
018	1700	039	3800	085	5900	106	8000	127	10100
019	1800	040	3900	086	6000	107	8100	128	10200
020	1900	041	4000	087	6100	108	8200	129	10300
021	2000	042	4100	088	6200	109	8300	130	10400

Some basic variables contained in the cross-sectional data file, `fwl2a.dat`, are shown in Box 3.2. A complete list of variables is provided by the FWL codebook.

3.2 The TDA Data Archive

We will now create a TDA data archive containing the cross-sectional data file `fwl2a.dat`. (Later on, we will add more data files to that archive.) The procedure is based on the program ZOO, a general purpose archive program that can be used to create compressed archives of arbitrary files.[3] We will call the archive `fwl2.zoo`. It is created in three steps.[4]

[3] ZOO has been written and made available as freeware by Rahul Dhesi. The program is available for many different computer platforms, including DOS and UNIX.

[4] A full description of how to create and use TDA data archives can be found in the TDA User's Manual.

Box 3.2 Basic variables in data file `fwl2a.dat`

```
Name    RTYPE  Columns
--------------------------------------------------
PID      all    5 - 12    Respondent's ID
RTYPE    all   13 - 15    Record type
INTY       1   34 - 35    Interview date: year
INTM       1   36 - 37    Interview date: month
INTD       1   38 - 39    Interview date: day
SEG       44   65 - 66    Socio economic group
POSTC     44   68 - 71    Post code
WARD      44   72 - 76    WARD / Local authority
COUNTY    44   77 - 78    County
IDENT     44        79    Subsample identifier
NHH       44   80 - 82    Number of households at the address

WEIGHT    45   60 - 67    Main sample weights (format F8.3)
MWEIGHT   45   70 - 77    Ethnic minority sample weights (F8.3)
```

Step 1: Compressing the data file. In a first step, the data file must be compressed and put into the archive. This is done with the following ZOO command:[5]

```
zoo ah fwl2.zoo fwl2a.dat
```

This command creates the archive fwl2.zoo (since the archive does not already exist), compresses the file `fwl2a.dat` and adds it to the archive. The result can be viewed by using ZOO's list command:

```
zoo l fwl2.zoo
```

resulting in a table of contents for the archive `fwl2.zoo`. It shows that we reach a compression factor of 94 %, that is, the size of the file `fwl2.dat` is reduced from 117,988,500 to 8,239,552 bytes.

Step 2: Adding a variable description file. ZOO is a general purpose archive program and not specifically concerned with data files consisting of data records and variables. To use the archive as a TDA data archive, one has to add a *variable description file* containing information about all variables that should possibly be retrieved from the archive.

The variable description file for our data archive will be called `fwl2.var`; Box 3.3 shows the first records corresponding to the variables in Box 3.2. For each variable, TDA needs the following information: (1) The

[5] A full description of ZOO and its many different commands is in a manual, `zoo.man`, that is part of every ZOO distribution.

Box 3.3 First records of variable description file `fwl2.var`

```
Name    FN   OFFS  LEN   Label
-----------------------------------------------------------
PID     1    4       8   respondent's identification number
RTYPE   1    12      3   record type (always 1)
INTY    1    33      2   interview date: year
INTM    1    35      2   interview date: month
INTD    1    37      2   interview date: day
SEG     1    4364    2   socio-economic group
POSTC   1    4367    4   post code
WARD    1    4371    5   ward/ local authority
COUNTY  1    4376    2   county
IDENT   1    4378    1   main/boost identifier
    1   main sample without ethnic minorities
    2   boost sample
    3   ethnic minorities from the main sample
NHH     1    4379    3   number of households at the address
SAMPLE  1    4383    1   green sample (X or blank)
WEIGHT  1    4459    8.3 main sample weights
MWEIGHT 1    4469    8.3 ethnic minority sample weights
```

variable's name;[6] we have generally used the same names as in the FWL codebook. (2) The logical file number (FN) of the data file the variable belongs to. This is necessary because a TDA data archive may contain several different data files. The logical file number of our first data file, `fwl2a.dat`, is 1. (3) The offset defining where the values of a variable begin in the records of a data file. Note that offsets begin with zero for the first physical column in a record. The definition of variables for the data file `fwl2a.dat` is based on the view that there is a single logical record for each respondent (consisting of 10500 consecutive bytes). This is taken into account when defining the variable's offsets. For instance, the variable `SEG` is in columns 65 – 66 in record type 44. As shown in Box 3.1, this record type has offset 4300, so the offset of variable `SEG` is 4364. (4) Then comes the number of columns (LEN) occupied by the variable. It suffices to give the number of columns since TDA will automatically recognize whether there are integer or floating point values. As an option (see, for instance, the definition of the weights variables), one can use a FORTRAN-like format statement that will be used then, by TDA, to define the *print format* of the variable. (5) One can add a *variable label*. This is optional and only used, by TDA, for informational purposes in print outs. Any non-blank strings, following the definition of the number

[6] Note that variable names must begin with an upper case letter but are, in general, case sensitive, that is, TDA distinguishes upper and lower case letters.

Box 3.4 Preliminary version of the archive description file `fwl2.des`

```
# archive description file for fwl2.zoo

fwl2.zoo

 1  fwl2a.dat   1   10500  11237  1603
99  fwl2.var    2       0   6512     0
```

of columns (LEN) until the end of the current record, is regarded as a variable label. (6) As a further option one can add *value labels*. This can be any text immediately following the definition of a variable in subsequent records that must begin, however, with at least one blank character (see the example for variable `IDENT` in Box 3.3).

Having created the variable description file, it must be added to the archive. The ZOO command is now

```
zoo ah fwl2.zoo fwl2.var
```

This command compresses the file `fwl2.var` and then adds the compressed file to the archive `fwl2.zoo`. The archive now contains all that is required by TDA to retrieve data: the data file (`fwl2a.dat`) and a description of its variables (`fwl2.var`).[7]

Step 3: Creating an archive description file. The final step is to create an *archive description file* that describes the archive to be used with TDA. This file contains two pieces of information (see the example in Box 3.4. First, the name of a data archive, optionally preceded by a path. And second, a description of each file that is contained in the archive *and* should be recognized by TDA.[8] There is one record for each file, containing: (1) The logical file number. (2) the name of the file.

[7] Note that to create the variable description file `fwl2.var`, we have used the machine-readable version of the FWL codebook. This has been a little bit cumbersome because the variable descriptions in the codebook do not follow a totally fixed format. It was therefore necessary to edit the variable description file by hand and so there may be some errors in the current variable description file. Also we have omitted some variables. It is always possible, however, to modify the variable description file. Simply extract the file from the archive (the command is: `zoo x fwl2.zoo fwl2.var`), then edit the file which is a plain ASCII file, and finally put the file back into the archive (the command is: `zoo ah fwl2.zoo fwl2.var`). Note that the archive should be repacked after replacement of a file; the command is `zoo P fwl2.zoo`. Note also that all variable names in a variable description file must be unique; this can be checked with the command `tda arcvchk=fname` where `fname` is the name of a variable description file.

[8] A TDA data archive may contain additional files that are not recognized by TDA.

(3) The type of the file (1 = data file, 2 = variable description file). (4) The physical length (in bytes) of the file's logical records (or zero if the records have variable length). (5) The number of logical records in the file. (6) The number of variables in the file.[9]

Checking the data archive. Before a data archive is used for the first time (and whenever the archive is updated with additional data files or with a modified variable description file) it should be checked whether it can be correctly used with TDA. The TDA command to perform such a check is

```
tda arcd=fwl2.des arcc
```

TDA is invoked, in this example, with two arguments (commands). The first command is `arcd=fwl2.des` telling TDA that it shall use a file with the name `fwl2.des` as an archive description file. (This `arcd` command must always be given when TDA shall use a data archive.) The second command is `arcc` requesting that the archive defined with the `arcd` command shall be checked. The check consists of the following steps: (1) TDA tries to open the archive description file and tries to interpret all entries in this file. (2) TDA tries to open the ZOO archive defined in the archive description file. (3) TDA tries to read (and internally decompress) all files from the archive that are defined in the archive description file. If any errors occur, or if there are any discrepancies between the file's description in the archive description file and what is found while extracting the files from the archive, an error message will be given in the program's standard output. Otherwise the program ends with the message „no errors found“.

3.3 Examples

Having set up the data archive as described above it can be used to retrieve selected variables and to create selected data matrices for subsequent statistical analyses. In the following we give some examples.

Example 1: Information about variables. A first exercise should be to retrieve information about variables contained in the archive. The TDA command is

```
tda arcd=fwl2.des arcv1 > var.lis
```

[9] This last entry is only for informational purposes and not checked by TDA.

Box 3.5 Command file `fwl1.cf`

```
arcd = fwl2.des;    archive description file

INTY   <1>[2.0] = A:INTY   ; (1) interview date: year
INTM   <1>[2.0] = A:INTM   ; (1) interview date: month

freq = INTY,INTM;   joint frequency distribution
```

Box 3.6 Output of command file `fwl1.cf`

```
TDA. Analysis of Transition Data (6.0). Tue Mar 19 11:12:16 1996
Command(s): cf=fwl1.cf
--------------------------------------------------------------------
Module: tda_arc [data archives]. Current memory: 134604 bytes.

Reading archive description file: fwl2.des
ZOO data archive: fwl2.zoo
Checking definitions of files in archive.

End of module tda_arc (135480).
--------------------------------------------------------------------
Module: tda_gd [data generation]. Current memory: 135480 bytes.

Variable  File       FN  Off  Len  Label
----------------------------------------
INTY      fwl2a.dat   1   33    2  interview date: year
INTM      fwl2a.dat   1   35    2  interview date: month

Using archive data file: fwl2a.dat
Number of records: 11237. Record length: 10500.
Set maximum number of cases to: 11237
Will generate a new data matrix with a maximum of 11237 cases.

Idx Variable T S  PFmt  Definition
----------------------------------
  1 INTY     2 1   2.0  A:INTY
  2 INTM     2 1   2.0  A:INTM

Read records: 11237
Created data for 2 variable(s) and 11237 cases.
Created a new data matrix.
Missing values (blank,star,point,general): none

End of module tda_gd (289659).
```

In this example, TDA is invoked with two commands. The **arcd** command defines the archive to be used (via the name of an archive descrip-

Box 3.6 (continued) Output of command file `fwl1.cf`

```
Module: tda_ds [descriptive statistics]. Current memory: 289659 bytes.

Frequency distribution: freq=INTY,INTM
Max number of categories: 11237

Index INTY INTM  Frequency   Pct   Cumulated    Pct
---------------------------------------------------
    1   94    1       5.00  0.04        5.00   0.04
    2   94    2       4.00  0.04        9.00   0.08
    3   94    6       8.00  0.07       17.00   0.15
    4   94    7    2014.00 17.92     2031.00  18.07
    5   94    8    3206.00 28.53     5237.00  46.60
    6   94    9    1230.00 10.95     6467.00  57.55
    7   94   10     578.00  5.14     7045.00  62.69
    8   94   11    1293.00 11.51     8338.00  74.20
    9   94   12     729.00  6.49     9067.00  80.69
   10   95    1    1036.00  9.22    10103.00  89.91
   11   95    2     292.00  2.60    10395.00  92.51
   12   95    3     268.00  2.38    10663.00  94.89
   13   95    4     327.00  2.91    10990.00  97.80
   14   95    5     241.00  2.14    11231.00  99.95
   15   95    7       1.00  0.01    11232.00  99.96
   16   95    8       1.00  0.01    11233.00  99.96
   17   95   12       4.00  0.04    11237.00 100.00
---------------------------------------------------
Sum                11237.00 100.00

End of module tda_ds (289659).
--------------------------------------------------------------------
End of program. Memory used: 559595 bytes. Tue Mar 19 11:14:41 1996
```

tion file), and the **arcv1** command requests that a list of all variables contained in the archive is to be written into TDA's standard output. Finally, the standard output is redirected into an output file, **var.lis** in this example. This output file can be used, then, as a starting point to create a TDA *command file*.

Example 2: Interview dates. To illustrate how to create frequency distributions of selected variables we use the variables **INTY** and **INTM** (year and month of interview). To create a TDA command file we use the output file **var.lis** of the first example. The command file, called **fwl1.cf**, is shown in Box 3.5. There are four commands. The **arcd** command defines the archive. The following two commands specify the two variables to be used. The final **freq** command requests a joint

Box 3.7 Command file `fwl2.cf`

```
arcd = fwl2.des;        archive description file

PID    <8>[8.0] = A:PID    ; (1) respondent's identification number
INTY   <1>[2.0] = A:INTY   ; (1) interview date: year
INTM   <1>[2.0] = A:INTM   ; (1) interview date: month
INTD   <1>[2.0] = A:INTD   ; (1) interview date: day

pdata = fwl2.d1;    data written to output file fwl2.d1;
dtda  =  tda.d1;    TDA description of fwl2.d1 written to tda.d1;
```

Box 3.8 Command file `fwl2a.cf`

```
dfile = fwl2.d1;          data file
noc = 11237;              (maximum) number of cases

PID  <8>[8.0] = c1;       variables
INTY <1>[2.0] = c2;
INTM <1>[2.0] = c3;
INTD <1>[2.0] = c4;

freq = INTY,INTM;         frequency distribution
```

frequency distribution for the two variables. To execute this command file TDA is invoked as

```
tda cf=fwl1.cf > fwl1.out
```

The `cf` command tells TDA about the name of the command file to be used. The standard output is now redirected into the output file `fwl1.out` shown in Box 3.6 (two parts).

Instead of the `freq` command one could use the `freq2` command resulting in another format for the two-dimensional frequency table (including row and column percentages).

Example 3: Creating data files. TDA (normally) always creates an internal data matrix before any statistical calculations are applied to the selected data. This internal data matrix (optionally only a part of it to be selected with the `keep` or `drop` commands) can be written into an output file. The command file `fwl2.cf`, shown in Box 3.7 gives a simple illustration. There are now four variables selected for the data matrix. In addition there are two commands: The `pdata` command requests that the data matrix is written into an output file, `fwl2.d1` in this example; and the `dtda` requests a TDA description of the data file.

Box 3.9 Command file `fwl3.cf`

```
arcd = fwl2.des;        archive description file

IDENT  <1>[2.0] = A:IDENT  ; (1) main/boost identifier
WEIGHT <8>[8.3] = A:WEIGHT ; (1) main sample weights
MWEIGHT<8>[8.3] = A:MWEIGHT; (1) ethnic minority sample weights

# here we create indicator variables for available weights
IW  = gt(WEIGHT,0);
IMW = gt(MWEIGHT,0);

freq = IDENT,IW,IMW;    joint frequency
```

Box 3.10 Output of command file `fwl3.cf`

```
Index IDENT  IW IMW  Frequency   Pct   Cumulated   Pct
------------------------------------------------------
    1     1   1   0    8849.00  78.75    8849.00  78.75
    2     2   0   1    2098.00  18.67   10947.00  97.42
    3     3   1   1     290.00   2.58   11237.00 100.00
------------------------------------------------------
Sum                   11237.00 100.00
```

The resulting data file, `fwl2.d1`, contains the four variables in a free format and can be used as an input data file in subsequent calls of TDA (or with other statistical packages). For illustration, we create a new command file, `fwl2a.cf`, based on `tda.d1` (see Box 3.8). This command file reads the data file `fwl2.d1` and creates again a two-dimensional frequency distribution for the year and month of the interview. Of course, the resulting frequency table will be the same as shown in Box 3.6.

Example 4: Available sample weights. Here is an example that checks the available sample weights. Box 3.9 shows the command file, `fwl3.cf`. It first defines three variables from the archive: `IDENT` (subsample identifier), `WEIGHT` (main sample weights) and `MWEIGHT` (weights for ethnic minorities). Then there is a definition of two indicator variables taking the value 1 if there is a positive weight and taking the value 0 otherwise. Finally we request a joint frequency distribution of `IDENT` and the two indicator variables. The resulting frequency table in Box 3.10 shows the consistency of the available information.

Example 5: Using sample weights. There are different possibilities to select subsamples. Command file `fwl4.cf`, shown in Box 3.11, shows five different possibilities. For each of the selected subsamples the

Box 3.11 Command file `fwl4.cf`

```
section = "data";

arcd = fwl2.des;        archive description file

IDENT  <1>[2.0] = A:IDENT  ; (1) main/boost identifier
WEIGHT <8>[8.3] = A:WEIGHT ; (1) main sample weights
MWEIGHT<8>[8.3] = A:MWEIGHT; (1) ethnic minority sample weights
S2B_1  <1>[2.0] = A:S2B_1  ; (1) person 1: sex

section = "main sample without ethnic minorities";
tsel = IDENT[1];    select main sample
freq = S2B_1;       distribution of SEX

section = "boost sample";
tsel = IDENT[2];    select boost sample (ethnic minorities)
freq = S2B_1;       distribution of SEX

section = "ethnic minorities from main sample";
tsel = IDENT[3];    select ethnic minorities from main sample
freq = S2B_1;       distribution of SEX

section = "main sample";
tsel = IDENT[1,3];  select main sample
freq = S2B_1;       distribution of SEX

section = "ethnic minorities";
tsel = IDENT[2,3];  select ethnic minorities
freq = S2B_1;       distribution of SEX
```

Box 3.12 Output of command file `fwl4.cf`

```
Section 2. [main sample without ethnic minorities]

Executing command: tsel=IDENT[1]
While active, only 8849 cases of the data matrix will be used.

Index S2B_1  Frequency   Pct   Cumulated   Pct
----------------------------------------------
    1     1    3927.00  44.38    3927.00  44.38
    2     2    4922.00  55.62    8849.00 100.00
----------------------------------------------
Sum            8849.00 100.00
```

command file requests an unweighted frequency distribution of the respondent's sex. Part of the output file (only for the main sample without ethnic minorities) is shown in Box 3.12. Now, to calculate a weighted

Box 3.13 Command file `fwl4a.cf`

```
arcd = fwl2.des;        archive description file
isel = IDENT[1];        select main sample without ethnic minorities

IDENT  <1>[2.0] = A:IDENT  ; (1) main/boost identifier
WEIGHT <8>[8.3] = A:WEIGHT ; (1) main sample weights
MWEIGHT<8>[8.3] = A:MWEIGHT; (1) ethnic minority sample weights
S2B_1  <1>[2.0] = A:S2B_1  ; (1) person 1: sex

cwt  = WEIGHT;      define case weights
dstat;              simple descriptive statistics
freq = S2B_1;       distribution of SEX
```

Box 3.14 Output of command file `fwl4a.cf`

```
Executing command: cwt=WEIGHT
Sum of weights: 37144.1150

Executing command: freq=S2B_1
Using case weights defined by: WEIGHT

Index S2B_1  Frequency   Pct   Cumulated   Pct
----------------------------------------------
    1     1   18571.13  50.00   18571.13  50.00
    2     2   18572.99  50.00   37144.11 100.00
----------------------------------------------
Sum           37144.11 100.00
```

frequency distribution, one has to use the `WEIGHT` variable. This is shown in command file **`fwl4a.cf`** shown in Box 3.13. Part of the output of using this command file is shown in Box 3.14. The frequency distribution is now weighted, the sum of weights is 37144.115.[10]

[10] Interpretation is limited by the fact that there is currently no information about how the weights have been created.

4 Respondent's Event Matrix

The FWL raw data file contains two types of event history data: an *event matrix* and a *job grid*, both for respondents and (if present) their partners. This section describes a data file that contains the event matrix information for respondents.

Box 4.1 Variables in data file `fwl2b.dat`

	Name	Columns	
1	REID	1 - 8	Respondent's ID
2	RENE	10 - 12	Number of events
3	REEC	14 - 16	Event (record) count
4	REIDENT	18	Subsample identifier
5	RESEX	20	Sex (1 men, 2 women)
6	REBY	22 - 23	Birth year
7	REBM	25 - 26	Birth month
8	REIY	28 - 29	Interview year
9	REIM	31 - 32	Interview month
10	REY	34 - 35	Date of event: year
11	REM	37 - 38	Date of event: month
12	REER	40	Error code for REM
13	RETYP	42 - 43	Type of event
14	RETYP1	45	Subtype of event
15	RETYP2	47 - 48	First/second event or number of child

4.1 The Event Data File

The idea is to create a new data file where each event from the respondent's event matrix is recorded in a separate record. The data file will be called **`fwl2b.dat`**. As shown in Box 4.1, the data file will contain 15 variables. The type of event is described by three variables. Basic information is in variables `RETYP` and `RETYP1` as defined in Box 4.2. Variable `RETYP2` contains additional information:

1. Information for events of type 4 (training), type 5 (other), and type 6 (benefits) should occur in pairs, one event should refer to the beginning, a second one should refer to the end of a period of the specified state. This is recorded in variable `RETYP2`. A value of 1 means that the event refers to the beginning, a value of 2 means that the event refers to the end of the period.

Box 4.2 Values of `RETYP` and `RETYP1` variables

RETYP	RETYP1		
1	1	Cohabitation	C: cohabitation
	2		M: marriage
	3		D: divorce
	4		W: widowed
	5		S: separated
	6		I: independent/single
2	1	Children	B: birth of a child
	2		S: still born
	3		D: death of child
	4		L: child left home
	5		R: child returned home
	6		A: adopted child
	7		P: step child
3	1	Work	W: any paid work
	2		U: unemployed and looking for work
	3		N: not working
4	1	Education	S: school
	2		C: college
	3		U: university/polytechnic
	4		G: government training scheme
	5		T: other training scheme
	6		O: open university
5	1	Other	V: voluntary work
	2		D: disability
	3		C: caring for adults
	4		H: stayed at home
	5		E: entered country for first time
	6		P: prison
	7		I: institutionalised
	8		N: national service
	9		A: abroad
6	1	Benefits	A: unemployment benefit
	2		B: income support
	3		C: child benefit / one parent benefit
	4		D: family credit
	5		E: housing benefit
	6		F: sickness and disability benefits
	7		G: invalid care allowance
	8		H: retirement pension
	9		J: widows benefit
	0		K: other state benefits

Box 4.3 Records in data file `fwl2b.dat` for first two respondents

```
10011  17    1 1 1 68 10 95  1 84 10 0  1 6  0
10011  17    2 1 1 68 10 95  1 84 10 0  3 3  0
10011  17    3 1 1 68 10 95  1 84 10 0  4 1  1
10011  17    4 1 1 68 10 95  1 86  7 0  4 1  2
10011  17    5 1 1 68 10 95  1 86  7 0  5 9  1
10011  17    6 1 1 68 10 95  1 87  7 0  3 1  0
10011  17    9 1 1 68 10 95  1 87  7 0  5 9  2
10011  17    7 1 1 68 10 95  1 87 10 0  3 3  0
10011  17    8 1 1 68 10 95  1 87 10 0  4 3  1
10011  17   10 1 1 68 10 95  1 88  8 0  3 1  0
10011  17   11 1 1 68 10 95  1 88  9 0  3 3  0
10011  17   12 1 1 68 10 95  1 89  8 0  3 1  0
10011  17   13 1 1 68 10 95  1 89  9 0  3 3  0
10011  17   15 1 1 68 10 95  1 93  6 0  4 3  2
10011  17   16 1 1 68 10 95  1 93  6 0  5 9  1
10011  17   14 1 1 68 10 95  1 93  8 0  3 1  0
10011  17   17 1 1 68 10 95  1 93  8 0  5 9  2

10101  15    1 1 2 72  2 95  1 88  2 0  1 6  0
10101  15    2 1 2 72  2 95  1 88  2 0  3 3  0
10101  15    3 1 2 72  2 95  1 88  2 0  4 1  1
10101  15    4 1 2 72  2 95  1 90  7 0  3 1  0
10101  15    6 1 2 72  2 95  1 90  7 0  4 1  2
10101  15    5 1 2 72  2 95  1 90  9 0  3 3  0
10101  15    7 1 2 72  2 95  1 90  9 0  4 2  1
10101  15    8 1 2 72  2 95  1 91  7 0  3 1  0
10101  15   10 1 2 72  2 95  1 91  7 0  4 2  2
10101  15    9 1 2 72  2 95  1 91 10 0  3 3  0
10101  15   11 1 2 72  2 95  1 91 10 0  4 3  1
10101  15   13 1 2 72  2 95  1 94  7 0  4 3  2
10101  15   14 1 2 72  2 95  1 94  7 0  6 1  1
10101  15   12 1 2 72  2 95  1 94  8 0  3 1  0
10101  15   15 1 2 72  2 95  1 94  8 0  6 1  2
```

One should note, however, that there are a lot of cases without a matching event that closes the period (5166 cases for events of type 4, 3655 cases for events of type 5, and 8853 cases for events of type 6). Of course, some of these cases can be due to right censoring.[11]

2. In case of type 2 events (children), the variable `RETYP2` is used to number the children. Note that this information is taken from the raw data file and may contain missing values (coded as -1). The Instruction Manual for the FWL Pilot Study says the following about this topic: „To be able to identify the children they should be numbered

[11] Unfortunately it is not possible to decide whether the missing of matching events is due to censoring or to data errors.

Box 4.4 Information about `REER` in data file `fwl2b.dat`

```
REER   Events   Description
-------------------------------------------------------------
  0    154353   month given in range 1 - 12
  1      7873   event month changed from  0 to  7  (don't know)
  2       410   event month changed from 13 to 12  (winter)
  3       517   event month changed from 14 to  3  (spring)
  4       734   event month changed from 15 to  7  (summer)
  5       408   event month changed from 16 to 10  (autumn)
```

in the box that is attached to the larger box, as in when the first child is born it should be B1, second child B2 and so on, therefore when child two leaves home it is L2 and when they return again it is R2, we can then calculate their age and so on.“

The data file `fwl2b.dat` contains 164,295 records for 11117 respondents (120 respondents do not have any valid entries and are excluded). The size of the data file is 8,050,455 bytes. For each respondent there is a variable number of records depending on the number of events. The records are ordered in ascending order with respect to the variables `REY` (year of event), `REM` (month of event), and `RETYP` (type of event). The range of variable `RENE` (the number of events per respondent) is 1 – 100.

Box 4.3 shows the records for the first two respondents. The first respondent (`REID` = 10011) has 17 records (events). The first event has type (1,6), that is, independent/single, recorded for date 10/84 (16th birthday). At the same date, we have the events (3,3), that is, not working, and (4,1), that is, visiting school. The last event should occur twice and, in fact, it does. The same event (4,1) occurs at 7/86 with `RETYP2` = 2, meaning that the period of visiting school ends in July, 1986. At the same time we find the event (5,9) (going abroad); the period of being abroad is until 7/86 when the same event occurs with `RETYP2` = 2. At the same time we find the event (3,1), that is, beginning paid work. And so on. The final events are recorded for 8/93. There are, in fact, two events: coming back from abroad (5,9) and beginning paid work (3,1).

Note that we sometimes do not know the exact month of an event. In these cases the raw data file contains codes for approximate dates indicating, for instance, that an event happened in spring or in winter. For the data file `fwl2b.dat` we have changed this and substituted „exact“ dates. However, an additional variable, `REER`, is used to record these changes as shown in Box 4.4.

Box 4.5 Additional variables for data file `fwl2b.dat`

```
Name     FN  Offset  Width  Label
-----------------------------------------
REID      2      0     8    Respondent's ID
RENE      2      9     3    Number of events
REEC      2     13     3    Event (record) count
REIDENT   2     17     1    Subsample identifier
RESEX     2     19     1    Sex (1 men, 2 women)
REBY      2     21     2    Birth year
REBM      2     24     2    Birth month
REIY      2     27     2    Interview year
REIM      2     30     2    Interview month
REY       2     33     2    Date of event: year
REM       2     36     2    Date of event: month
REER      2     39     1    Error code for REM
RETYP     2     41     2    Type of event
RETYP1    2     44     1    Subtype of event
RETYP2    2     46     2    First/second event or number of child
```

Box 4.6 New version of the archive description file (`fwl2.des`)

```
# archive description file for fwl2.zoo

fwl2.zoo

 1  fwl2a.dat   1   10500  11237  1603
 2  fwl2b.dat   1      49 164295    15
99  fwl2.var    2       0   6526     0
```

4.2 Updating the Archive

In order to put the new data file, `fwl2b.dat`, into the data archive, `fwl2.zoo`, the command is simply

```
zoo ah fwl2.zoo fwl2b.dat
```

This command puts the compressed file `fwl2b.dat` into the already existing archive `fwl2.zoo`. In addition, one has to update the variable description file. This is simply done by appending the list of new variables (see Box 4.5) to the end of the already existing variable description file `fwl2.var`. Note that we have used the logical file number 2 for the new data file.

Box 4.7 Command file `fwl7.cf`

```
arcd = fwl2.des;

RESEX  <1>[2.0] = A:RESEX  ; (2) Sex (1 men, 2 women)
REBY   <1>[2.0] = A:REBY   ; (2) Birth year
REBM   <1>[2.0] = A:REBM   ; (2) Birth month
REIY   <1>[2.0] = A:REIY   ; (2) Interview year
REIM   <1>[2.0] = A:REIM   ; (2) Interview month
REY    <1>[2.0] = A:REY    ; (2) Date of event: year
REM    <1>[2.0] = A:REM    ; (2) Date of event: month
REER   <1>[2.0] = A:REER   ; (2) Error code for REM
RETYP  <1>[2.0] = A:RETYP  ; (2) Type of event
RETYP1 <1>[2.0] = A:RETYP1 ; (2) Subtype of event
RETYP2 <1>[2.0] = A:RETYP2 ; (2) First/second event or # of child

dstat;
freq1 = REBY,REBM,RETYP2,REER;
```

Box 4.8 Information about variables in `fwl2b.dat`

```
Variable    Minimum    Maximum       Mean   Std.Dev.
----------------------------------------------------
RESEX        1.0000     2.0000     1.5798     0.4936
REBY        -1.0000    78.0000    50.6314    14.2857
REBM        -1.0000    12.0000     6.4727     3.4492
REIY        94.0000    95.0000    94.1776     0.3821
REIM         1.0000    12.0000     7.5705     2.9328
REY         38.0000    95.0000    77.7186    13.1526
REM          1.0000    12.0000     6.6340     3.1053
REER         0.0000     5.0000     0.0926     0.4596
RETYP        1.0000     6.0000     3.3525     1.5950
RETYP1       1.0000     9.0000     2.7589     2.0083
RETYP2      -1.0000    13.0000     0.9053     0.9995
```

In addition, we have to update the archive description file, `fwl2.des`, by adding one line describing the new data file. The updated description file is shown in Box 4.6.

Finally, the new archive can be checked with the TDA command

```
tda arcd=fwl2.des arcc
```

Example: Descriptive statistics. To illustrate using the new data file, we begin with a small example to get basic descriptive information about the new variables. The command is `fwl7.cf` shown in Box 4.7. The `dstat` command provides this information, the `freq1` command additionally provides frequency distributions of the variables given on

Box 4.9 Illustration of event history data

ID	SN	ORG	DES	TS	TF	TS1	TF1
1	1	1	2	0	20	0	20
1	2	2	3	20	50	0	30
1	3	3	3	50	75	0	25
2	1	2	1	0	30	0	30
2	2	1	1	30	70	0	40

the right-hand side. Box 4.8 shows the result of the `dstat` command. As already noted, there are some missing values (-1) for the respondent's birth date and for the `RETYP2` variable.

4.3 Event History Data

In an event history data file there is (normally) one record for each episode. The information required depends on whether there is single or multi-episode data.

In a set of *single episode data*, each episode is sufficiently described by: an *origin state* (`ORG`), that is, the state entered when the episode begins, and a *destination state* (`DES`), that is, the state entered when the current episode ends; and a *starting time* (`TS`) and *ending time* (`TF`) to record when the episode begins and ends. If an episode is right censored, then the destination state equals the origin state, and the ending time is the time when the episode becomes right censored.[12]

In a set of *multi-episode data* each individual can contribute a sequence of episodes, so we need two more pieces of information: an identification number (`ID`) showing to which individual the episode belongs, and a *spell number* (`SN`) to count the episodes and provide a temporal ordering.

Box 4.9 illustrates multi-episode data. There are two individuals, the first with three, the second with two episodes. The process time axis begins at time zero, and in both cases it is assumed that the last episode is right censored. Note that one can regard the same data as a set of single episode data by just ignoring the `ID` and `SN` information.

How to define the time axis depends on the model to analyse the data. In TDA it is required that the starting time of each episode is not less

12 There is no specific convention to code left censored data.

Box 4.10 Command file `fwl8.cf` to create episode data

```
arcd = fwl2.des;        archive description file
dblock(110) = REID;     define block mode

# select records with valid birth year,
# and with a marriage event or the first event

isel = gt(REBY,0) . REBM[1,,12] .
          ((RETYP[1] . RETYP1[2])  + REEC[1]);

REID   <8>[8.0] = A:REID   ; (2) Respondent's ID
REEC   <2>[3.0] = A:REEC   ; (2) Event (record) count
REBY   <1>[2.0] = A:REBY   ; (2) Birth year
REBM   <1>[2.0] = A:REBM   ; (2) Birth month
REIY   <1>[2.0] = A:REIY   ; (2) Interview year
REIM   <1>[2.0] = A:REIM   ; (2) Interview month
REY    <1>[2.0] = A:REY    ; (2) Date of event: year
REM    <1>[2.0] = A:REM    ; (2) Date of event: month
RETYP  <1>[2.0] = A:RETYP  ; (2) Type of event
RETYP1 <1>[2.0] = A:RETYP1 ; (2) Subtype of event

MARR [1.0] = RETYP[1] . RETYP1[2];    dummy for marriage events

MARRN = grec(MARR);
MARRCNT = if MARR then MARRN else 0;  count marriages

# select all cases with first marriage or right censored
SEL = eq(MARRCNT,1) + eq(bnrec,1);
vsel = SEL;

BDate [4.0] = REBY * 12 + REBM;  birth date in century months
IDate [4.0] = REIY * 12 + REIM;  interview date in century months
XDate [4.0] = REY * 12 + REM;    date for marriage or other event

# now we have the date for first marriage, or the interview date if
# the spell is right censored.
EDate [4.0] = if MARR then XDate else IDate;

# duration in month from 16th birth day to marriage or interview
DUR [4.0] = EDate - BDate - 16 * 12;

keep  = REID,MARR,DUR;
pdata = t8.dat;   write data to output file t8.dat
dtda  = t8.tda;   TDA description of output file
```

than zero, and that each episode has a positive duration.[13] There is,

13 Note that duration is always calculated as the difference between starting and ending time.

Box 4.11 Command file fwl8a.cf for product-limit estimation

```
dfile = t8.dat;          data file
noc = 11060;             max number of records
isel = gt(c3,0);         input select

ID <5> [8.0] = c1;       definition of variables
DES    [2.0] = c2;
TF     [4.0] = c3;

org = 0;                 definition of episode data
des = DES;
ts  = 0;
tf  = TF;

ple = ple.8a;            request product-limit estimation
```

however, an important additional assumption when estimating models for event history data: it is assumed, then, that each episode is defined on a process time axis that begins with time zero. Or put otherwise: if the starting time of an episode is greater than zero, it is assumed that the episode is left truncated up to the given starting time.

Example 1: Duration until first marriage. To illustrate how to create event history data based on the data file `fwl2b.dat`, we take as an example the duration until the first marriage. We proceed in two steps. In a first step, we create an data file containing for each individual a single episode. In a second step we use this data file to estimate a survivor function for duration until first marriage.

Creating the episode data file is done with command file `fwl8.cf` shown in Box 4.10.

The output data file is `t8.dat`. It contains one episode for each individual, described by three variables: `REID` is the identification number for the individuals; `MARR` is an indicator variable with value 1 if there was a first marriage until the interview data, and value 0 otherwise; and `DUR` is the duration in months, measured from the individual's 16th birthday until first marriage or, if there was no marriage, until the interview date.

The data file contains 11094 records. 23 individuals have been excluded because there is no valid birth data.

Having create the episode data file, we use command file `fwl8a.cf` (Box 4.11) for product-limit estimation of a survivor function. The data file contains 77 episodes with a negative or zero duration that must be excluded. The table containing the estimates for the survivor function is written into a new output file, `ple.8a`. Since this table is fairly long, we do not document it here. Instead we present a plot of the survivor function, created with command file `fwl8b.cf` shown in Box 4.12.

Box 4.12 Command file fwl8b.cf to plot the survivor function

```
dfile = ple.8a;  data file with PL estimates

T  = c3;                 time axis
S  = c7;                 survivor function
T1 = 16 + T / 12;        age in years

postscript = fwl8b.ps;   PostScript output file
pxlen = 100;             size of plot
pylen = 55;
pxa(5,5) = 15,45;        logical X axis
pya(0.5,5) = 0,1;        logical Y axis
pyfmt = 3.1;
plot = T1,S;             plot T1 vs. S
```

Example 2: Adding variables. We now want to add information about respondent's sex to the data file `t8.dat` created in example 1. This information is in the cross-sectional data file `fwl2a.dat`, so we have to merge data. How this can be accomplished with TDA is shown

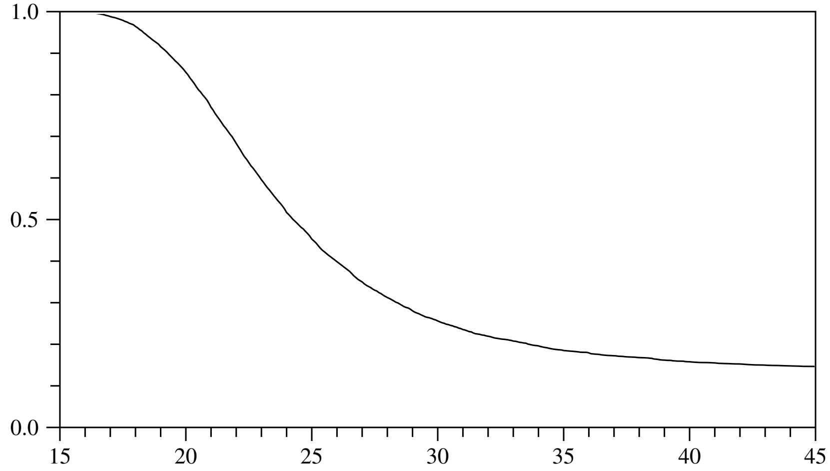

Figure 4.1 Survivor function for duration until first marriage, duration measured in months from 16th birthday. X axis: age (in years).

Box 4.13 Example command file `fwl9.cf` to merge data

```
section;  first section to get event data

      this is identical to command file fwl8.cf except
      for the commands to write the output data file.

section; second section to add variables

# first we define the additional variables

PID    <8>[8.0] = A:PID    ; (1) respondent's identification number
S2B_1  <1>[2.0] = A:S2B_1  ; (1) person 1: sex

match = PID,REID;  this command shows how to merge the new data
                ;  to the already existing data matrix

section;  third section to write selected variables into output file

keep  = REID,MARR,DUR,S2B_1;
pdata = t9.dat;   write data to output file t8.dat
dtda  = t9.tda;   TDA description of output file
```

in the example command file `fwl9.cf` (Box 4.13). Since merging data has to be done in separate sections of a command file, we now have three sections. The first section is identical to command file `fwl8.cf` used in example 1, except for the command to write the output file. At the end of this section we have a data matrix containing all variables defined in that section.

Then comes a second section where we define additional variables. Since these additional variables have to be retrieved from another data file (`fwl2a.dat`) one has to specify how to merge these new variables to the already existing data matrix. This is done with the command

```
match = PID,REID;
```

meaning that records should correspond where `PID` (one of the new variables) is equal to `REID` (one of the already existing variables).

Finally, there is a third section to write the output data file, `t9.dat`, now containing `S2B_1` (respondent's sex) as an additional variable.

5 Information about Partners

A specific feature of the FWL survey is that it contains information about the respondent's partners. There are four pieces of information.

1. Basic information to record whether the respondent currently has a partner is in the cross-sectional data file `fwl2a.dat`. Most important are the variables for the partner's date of birth and his/her relationship to the respondent.
2. Information about the current situation of partners is in module Q of the questionnaire. The corresponding variables are in the main cross-sectional data file (`fwl2a.dat`).
3. In addition there is an event matrix for the partners. It is similar to the respondent's event matrix but the amount of information is somewhat reduced (concerning only the categories: *work*, *training and education*, *other*, and *state benefits*). This will be further discussed in Section 6.
4. Finally, there is also a job grid for the partners. This will be discussed in Section 8.

Basic Information

Basic information about whether a respondent currently has a partner is in variables `S1BY`, `S1BM`, and `S1BD` (birth date of partner: year, month, and day) and in variable `S5_2` (relationship of partner to respondent). To check this information we use command file `fwl10.cf` shown in Box 5.1.

We define two indicator variables: `I1` has value 1 if there is a positive year in `S1BY`, and `I2` has value 1 if variable `S5_2` has value 1, that is, if the second person in the respondent's household (that should be the partner, if any) is the respondent's husband, wife, or partner. The result of cross-tabulating these two indicator variables (requested with the `freq` command) is shown in Box 5.2. There are, obviously, some inconsistencies. 79 respondents have a partner (`I2 = 1`) but there is no information about the partner's year of birth. On the other hand, 109 respondents have identified a partner by giving his/her year of birth, but – quite possible – these partners are not part of the respondent's current household.

Box 5.1 Command file `fwl10.cf`

```
arcd = fwl2.des;

PID    <8>[8.0] = A:PID    ; (1) respondent's identification number
S1BY   <2>[4.0] = A:S1BY   ; (1) partner's date of birth: year
S5_2   <1>[2.0] = A:S5_2   ; (1) person 2: relationship to respondent

I1 = gt(S1BY,0);     indicator for partner's birth year
I2 = S5_2[1];        indicator for second person in household
                     # relationship to resp: husband, wife, or partner

freq2 = I1,I2;       cross tabulation
```

Box 5.2 Cross-tabulation of I1 and I2 defined in `fwl10.cf`

```
  Index  I1  I2  Frequency   Pct   Cumulated   Pct
  ------------------------------------------------
      1   0   0    4462.00  39.71    4462.00  39.71
      2   0   1      79.00   0.70    4541.00  40.41
      3   1   0     109.00   0.97    4650.00  41.38
      4   1   1    6587.00  58.62   11237.00 100.00
  ------------------------------------------------
  Sum             11237.00 100.00
```


6 Partner's Event Matrix

This section describes the event matrix data for partners. In basically the same way as done in Section 4, we create a new data file, **`fwl2c.dat`**, containing the event matrix information for partners. Then this data file will be added to our archive, **`fwl2.zoo`**, and we illustrate how to use the data file with some examples.

Box 6.1 Variables in data file `fwl2c.dat`

```
     Name     Columns
 ----------------------------------------
  1  PEID       1 -  8    Respondent's ID
  2  PENE      10 - 12    Number of events
  3  PEEC      14 - 16    Event (record) count
  4  PEIDENT        18    Subsample identifier
  5  PESEX     20 - 21    Partner's sex (1 men, 2 women)
  6  PEBY      23 - 26    Partner's birth year
  7  PEBM      28 - 29    Partner's birth month
  8  PEIY      31 - 32    Interview year
  9  PEIM      34 - 35    Interview month
 10  PEY       37 - 38    Date of event: year
 11  PEM       40 - 41    Date of event: month
 12  PEER           43    Error code for PEM
 13  PETYP     45 - 46    Type of event
 14  PETYP1         48    Subtype of event
 15  PETYP2    50 - 51    First/second event
```

Information about partner's event matrix is contained in the raw data file in records of type 66. The coding is identical to the event matrix for respondents, but there is less information. There is no information about *cohabitation* events and *children*. The recorded events, and there numerical codes, are shown in Box 6.2.

The variables contained in **`fwl2c.dat`** are shown in Box 6.1. Comparing this list of variables to the variables in respondent's event matrix data file (see Box 6.1), one should note:

1. **`PEID`** is the respondent's identification number; there is no separate identification number for partners. (This is, of course, the correct variable to merge data from separate files.)
2. **`PESEX`** is now the partner's sex and is identical to **`S2B_2`** in the cross-sectional data file. The variable now occupies 2 columns because there are some missing values coded as -1.

Box 6.2 Values of RETYP and RETYP1 variables in `fwl2c.dat`

```
RETYP  RETYP1
--------------------------------------------
   3      1     Work         W: any paid work
          2                  U: unemployed and looking for work
          3                  N: not working

   4      1     Education    S: school
          2                  C: college
          3                  U: university/polytechnic
          4                  G: government training scheme
          5                  T: other training scheme
          6                  O: open university

   5      1     Other        V: voluntary work
          2                  D: disability
          3                  C: caring for adults
          4                  H: stayed at home
          5                  E: entered country for first time
          6                  P: prison
          7                  I: institutionalised
          8                  N: national service
          9                  A: abroad

   6      1     Benefits     A: unemployment benefit
          2                  B: income support
          3                  C: child benefit / one parent benefit
          4                  D: family credit
          5                  E: housing benefit
          6                  F: sickness and disability benefits
          7                  G: invalid care allowance
          8                  H: retirement pension
          9                  J: widows benefit
          0                  K: other state benefits
```

3. `PEBY` and `PEBM` are the partner's birth year and birth month, respectively, and are identical to the variables `S1BY` and `S1BM` in the cross-sectional data file. Note that the birth year in `PEBY` is given as 4 digits. (While the questionnaire selects respondents with age, at interview time, from 16 to 69, there may be quite different birth years for partners.)
4. All other variables have identical meaning as for the respondent's event matrix file.

The partner's event matrix file, `fwl2c.dat`, contains the information taken from all records of type 066 in the raw data file. The file contains

Box 6.3 Additional variables for data file `fwl2c.dat`

```
Name    FN  Offset  Width  Label
-------------------------------------------
PEID     3      0     8    Respondent's ID
PENE     3      9     3    Number of events
PEEC     3     13     3    Event (record) count
PEIDENT  3     17     1    Subsample identifier
PESEX    3     19     2    Partner's sex (1 men, 2 women)
PEBY     3     22     4    Partner's birth year
PEBM     3     27     2    Partner's birth month
PEIY     3     30     2    Interview year
PEIM     3     33     2    Interview month
PEY      3     36     2    Date of event: year
PEM      3     39     2    Date of event: month
PEER     3     42     1    Error code for PEM
PETYP    3     44     2    Type of event
PETYP1   3     47     1    Subtype of event
PETYP2   3     49     2    First/second event
```

Box 6.4 New version of the archive description file (`fwl2des`)

```
# archive description file for fwl2.zoo

fwl2.zoo

 1  fwl2a.dat   1   10500  11237  1603
 2  fwl2b.dat   1      49 164295    15
 3  fwl2c.dat   1      52  21599    15
99  fwl2.var    2       0   6543     0
```

21599 records (events) for 4233 persons. So the number of partners with at least one event is 4233. Comparing this to the number of partners found in Section 5, we find that we do not have event matrix entries for about 36 % of the partners.[14]

Updating the Archive

In order to put the new data file, `fwl2c.dat`, into the data archive, `fwl2.zoo`, the command is simply

```
zoo ah fwl2.zoo fwl2c.dat
```

This command puts the compressed file `fwl2c.dat` into the already existing archive `fwl2.zoo`. In addition, one has to update the variable

[14] As shown in Box 5.2, there are at least 6587 respondents with a partner.

Box 6.5 Command file `fwl12.cf`

```
arcd = fwl2.des;

PENE   <2>[3.0] = A:PENE   ; (3) Number of events
PEIDENT<1>[2.0] = A:PEIDENT; (3) Subsample identifier
PESEX  <1>[2.0] = A:PESEX  ; (3) Partner's sex (1 men, 2 women)
PEBY   <2>[4.0] = A:PEBY   ; (3) Partner's birth year
PEBM   <1>[2.0] = A:PEBM   ; (3) Partner's birth month
PEIY   <1>[2.0] = A:PEIY   ; (3) Interview year
PEIM   <1>[2.0] = A:PEIM   ; (3) Interview month
PEY    <1>[2.0] = A:PEY    ; (3) Date of event: year
PEM    <1>[2.0] = A:PEM    ; (3) Date of event: month
PEER   <1>[2.0] = A:PEER   ; (3) Error code for PEM
PETYP  <1>[2.0] = A:PETYP  ; (3) Type of event
PETYP1 <1>[2.0] = A:PETYP1 ; (3) Subtype of event
PETYP2 <1>[2.0] = A:PETYP2 ; (3) First/second event

dstat;
freq1 = PESEX,PEBY,PEBM,PETYP2,PEER;
```

description file. This is simply done by appending the list of new variables (see Box 6.3) to the end of the already existing variable description file `fwl2.var`. Note that we have used the logical file number 3 for the new data file.

In addition, we have to update the archive description file, `fwl2des`, by adding one line describing the new data file. The updated description file is shown in Box 6.4.

Finally, the new archive can be checked with the TDA command

```
tda arcd=fwl2.des arcc
```

Example 1: Descriptive statistics. To illustrate using the new data file, we begin with a small example to get basic descriptive information about the new variables. The command is `fwl12.cf` shown in Box 6.5. The `dstat` command provides this information, the `freq1` command additionally provides frequency distributions of the variables given on the right-hand side. Box 6.6 shows the result of the `dstat` command. As already noted, there are some missing values. 17 partners do not have a valid sex information, for 181 partners the birth year is missing (range is 1910 – 1979), and for 52 partners the birth month is missing.

Box 6.6 Information about variables in `fwl2c.dat`

```
Variable    Minimum    Maximum       Mean   Std.Dev.
----------------------------------------------------
PENE         1.0000    46.0000     8.8616     6.2334
PEIDENT      1.0000     3.0000     1.1923     0.4419
PESEX       -1.0000     2.0000     1.5845     0.4976
PEBY        -1.0000  1979.0000  1931.4006   178.1536
PEBM        -1.0000    12.0000     6.2761     3.4719
PEIY        94.0000    95.0000    94.1444     0.3515
PEIM         1.0000    12.0000     7.7201     2.8168
PEY         40.0000    95.0000    81.3636    11.3673
PEM          1.0000    12.0000     6.5819     3.2281
PEER         0.0000     5.0000     0.0801     0.4432
PETYP        3.0000     6.0000     4.0851     1.2372
PETYP1       1.0000     9.0000     2.7258     1.9240
PETYP2       0.0000     2.0000     0.6448     0.7423
```

7 Respondent's Job Grid

The respondent's *job grid* provides detailed information about the episodes identified by the *work* category in the event matrix. The information is contained in records of type 64, 70, and 71 of the raw data file. This section describes how we have constructed an event history data file containing the job grid information.

7.1 The Job History Data File

The event history data file constructed from the respondent's job grid data will be called `fwl2d.dat`. It contains two types of variables: basic variables and additional variables. The basic variables, shown in Box

Box 7.1 Basic variables in the respondent's job grid data file `fwl2d.dat`

```
     Name    Columns  Description
   -----------------------------------------------------
    1 RJID     1 -  8  Respondent's ID
    2 RJNREC  10 - 11  Number of records
    3 RJRECN  13 - 14  Record number (1,...,RJNREC)
    4 RJNEP   16 - 17  Number of episodes
    5 RJEPN   19 - 20  Episode number (1,...,RJNEP)
    6 RJORG        22  Origin state
                        1 unemployed
                        2 not working
                        3 working but no job grid information
                        4 working, one employer
                        5 working, self employed
                        6 working with various employers
    7 RJDES   24 - 25  Destination state
    8 RJTSY   27 - 28  Starting time of episode: year
    9 RJTSM   30 - 31  Starting time of episode: month
   10 RJTSE        33  Starting time of episode: error flag
   11 RJTFY   35 - 36  Ending time of episode: year
   12 RJTFM   38 - 39  Ending time of episode: month
   13 RJTFE        41  Ending time of episode: error flag
   14 RJNEP1  43 - 44  Number of subepisodes
   15 RJEPN1  46 - 47  Subepisode number (1,...,RJNSEP)
   16 RJTSY1  49 - 50  Starting time of subepisode: year
   17 RJTSM1  52 - 53  Starting time of subepisode: month
   18 RJTSE1       55  Starting time of subepisode: error flag
   19 RJTFY1  57 - 58  Ending time of subepisode: year
   20 RJTFM1  60 - 61  Ending time of subepisode: month
   21 RJTFE1       63  Ending time of subepisode: error flag
```


Box 7.2 Some records in `fwl2d.dat` with basic variables (Box 7.1)

```
10011  8  1  8  1 2  4 84 10 0 87  6 0  1  1 84 10 0 87  6 0
10011  8  2  8  2 4  2 87  7 0 87 10 0  1  1 87  7 0 87 10 0
10011  8  3  8  3 2  4 87 11 0 88  7 0  1  1 87 11 0 88  7 0
10011  8  4  8  4 4  2 88  8 0 88  9 0  1  1 88  8 0 88  9 0
10011  8  5  8  5 2  4 88 10 0 89  7 0  1  1 88 10 0 89  7 0
10011  8  6  8  6 4  2 89  8 0 89  9 0  1  1 89  8 0 89  9 0
10011  8  7  8  7 2  4 89 10 0 93  7 0  1  1 89 10 0 93  7 0
10011  8  8  8  8 4  4 93  8 0 95  1 0  1  1 93  8 0 95  1 0
------------------------------------------------------------
10101  6  1  6  1 2  4 88  2 0 90  6 0  1  1 88  2 0 90  6 0
10101  6  2  6  2 4  2 90  7 0 90  9 0  1  1 90  7 0 90  9 0
10101  6  3  6  3 2  4 90 10 0 91  6 0  1  1 90 10 0 91  6 0
10101  6  4  6  4 4  2 91  7 0 91 10 0  1  1 91  7 0 91 10 0
10101  6  5  6  5 2  4 91 11 0 94  7 0  1  1 91 11 0 94  7 0
10101  6  6  6  6 4  4 94  8 0 95  1 0  1  1 94  8 0 95  1 0
------------------------------------------------------------
40211  7  1  5  1 2  4 77  9 0 77 12 0  1  1 77  9 0 77 12 0
40211  7  2  5  2 4  4 78  1 0 82  1 0  3  1 78  1 0 80  1 0
40211  7  3  5  2 4  4 78  1 0 82  1 0  3  2 80  1 0 81 10 0
40211  7  4  5  2 4  4 78  1 0 82  1 0  3  3 81 10 0 82  1 0
40211  7  5  5  3 4  4 82  1 0 84  5 0  1  1 82  1 0 84  5 0
40211  7  6  5  4 4  2 84  5 0 87  8 0  1  1 84  5 0 87  8 0
40211  7  7  5  5 2  2 87  9 0 95  1 0  1  1 87  9 0 95  1 0
------------------------------------------------------------
50591 11  1  7  1 4  4 63  7 0 63  9 0  1  1 63  7 0 63  9 0
50591 11  2  7  2 4  4 63  9 0 65 11 0  1  1 63  9 0 65 11 0
50591 11  3  7  3 4  4 65 11 0 66  2 0  1  1 65 11 0 66  2 0
50591 11  4  7  4 4  1 66  2 0 66  7 0  1  1 66  2 0 66  7 0
50591 11  5  7  5 1  4 66  8 0 66  9 0  1  1 66  8 0 66  9 0
50591 11  6  7  6 4  2 66 10 0 92 10 0  5  1 66 10 0 69 12 0
50591 11  7  7  6 4  2 66 10 0 92 10 0  5  2 69 12 0 80  7 1
50591 11  8  7  6 4  2 66 10 0 92 10 0  5  3 80  7 1 80  7 1
50591 11  9  7  6 4  2 66 10 0 92 10 0  5  4 80  7 1 85  7 1
50591 11 10  7  6 4  2 66 10 0 92 10 0  5  5 85  7 1 92 10 0
50591 11 11  7  7 2  2 92 11 0 94  9 0  1  1 92 11 0 94  9 0
```

7.1, represent the structure of the data file and will be discussed first. To support this discussion, Box 7.2 shows values of the basic variables for some (selected) respondents.

1. The data file is organized according to the respondent's ID, given in variable `RJID`. For each respondent (with at least one valid job grid entry) there is a variable number of records. The number of records is given by variable `RJNREC`; variable `RJRECN` provides a sequence number for the records.

2. Variable `RJNEP` provides the number of episodes, and `RJEPN` gives a sequence number for the episodes. Note that the number of episodes can be less than the number of records (`RJNREC`) because an episode may consist of subepisodes.
3. Episodes have one of the origin states described in Box 7.1 and provided by variable `RJORG`. All episodes correspond to an entry in the respondent's event matrix. Episodes with origin state 1 (unemployed), 2 (not working), and 3 (working, but no job grid entry) are taken directly from the event matrix. Episodes with origin state 4, 5, or 6, should also correspond to event matrix entries but are differentiated by the job grid information.
4. Variable `RJDES` provides the destination state of an episode and is calculated as follows. For the last episode, the destination state equals the origin state since the episode is right censored. For all other episodes the destination state is `ORGN` or `ORGN + 10` where `ORGN` is the origin state of the following episode. It is `ORGN` if the starting time of the following episode is less than or equal to the ending time of the current episode plus one month; otherwise, to flag a transition into an unknown destination state, it is `ORGN + 10`.
5. Variables `RJTSY` and `RJTSM` provide the year and month for the starting time of the episode, and `RJTFY` and `RJTFM` provide the ending time. `RJTSE` and `RJTFE` provide error flags for the dates and will be explained below.
6. As noted, episodes may consist of subepisodes; see for instance the respondent with ID 40211. In this example, the second episode consists of three subepisodes identified by variables `RJNEP1` and `RJEPN1`. Subepisodes are changes in an employment relation during a main episode. Of course, they can only occur if the origin state of the main episode is 4, 5, or 6.

Starting and ending times. As to be expected, starting times for the episodes and subepisodes are not always given to the correct month and not always consistent. Here are the steps we have taken to correct for these problems. First for the main episodes.

1. If the year when the episode begins or ends is missing the episode is not written into the output file. This has happened in 152 cases.
2. Also if the year when the episode begins is greater than the interview year, the episode is not written into the output file. This has happened in 6 cases.

3. If an ending date has value 99, meaning that the episode is not finished up to the interview date, the episode is assumed to be right censored at the interview date.
4. If the month for the starting or ending date of the episode is not exactly given, it is substituted by values according to Box 4.4 and this is recorded in the error variables `RJTSE`, `RJTFE`, `RJTSE1`, and `RJTFE1`.
5. If the ending date of the episode is greater than the interview date it is substituted by the interview date. The error flags get then the value 6. This has happened in 75 cases.
6. If the ending time is less than the starting time of an episode, both dates have been exchanged and the error flags, for both the starting and ending times, get the value 7. This happened in 13 cases.

Additional covariates. In addition to the basic variables shown in Box 7.1, data file `fwl2d.dat` contains additional covariates, most of them taken from the respondent's job grid. The following list, continuing Box 7.1, describes these covariates (with reference to the job grid description in the FWL code book).

```
    Name      Columns  Description
------------------------------------------------------------------
22  RJNJOBS   65 - 66  Number of jobs (E3 or V3)
                       If RJORG = 4 then E3
                       If RJORG = 6 then V3

23  RJSJOBS   68 - 69  Kind of jobs (E4 or V4)
                       If RJORG = 4 and RJNJOBS = 2 then E4
                       If RJORG = 6 then V4

24  RJRFIN1   71 - 72  First reason for end of job (E7A)
                       If RJORG = 4 or 5 then E7A

25  RJRFIN2   74 - 75  Second reason for end of job (E7B)
                       If RJORG = 4 or 5 then E7B

26  RJTUMEM   77 - 78  Trade union member (E8)
                       If RJORG = 4 or 5 then E8

27  RJPENS    80 - 81  Member of pension scheme (E9 or V11)
                       If RJORG = 4 or 5 then E9
                       If RJORG = 6 then V11
```

```
    Name      Columns  Description
---------------------------------------------------------------------
28  RJTRAIN   83 -  84  Training (E10)
                        1 = on the job training
                        2 = off the job training
                        3 = both, on and off the job training
                        4 = DK
                        5 = NA

29  RJTRAIND  86 -  87  Duration of training
                        1 = any over a month
                        2 = any less than a month
                        3 = DK
                        4 = NA

30  RJSOC     89 -  91  Type of occupation (E14 SOC)
                        If RJORG = 4 or 5 then E14 SOC

31  RJSEG     93 -  94  Type of occupation (E14 SEG)
                        If RJORG = 4 or 5 then E14 SEG

32  RJSOCV1   96 -  97  Type of job if RJORG = 6 (V5-1)

33  RJSOCV2   99 - 100  Type of job if RJORG = 6 (V5-2)

34  RJSOCV3  102 - 103  Type of job if RJORG = 6 (V5-3)

35  RJSOCV4  105 - 106  Type of job if RJORG = 6 (V5-4)

36  RJWTIME  108 - 109  Working time (E16 or V6)
                        If RJORG = 4 or 5 then E16
                        If RJORG = 6 then V6
                        1 = full time
                        2 = part time
                        3 = DK/NA

37  RJEMPL   111 - 112  Type of employment (E17 or V8)
                        If RJORG = 4 or 5 then E17
                        1 = self employed
                        2 = not self employed
                        If RJORG = 6 then V8
                        1 = self employed
                        2 = employed
                        3 = government scheme (trainee)
                        4 = self employed / employed
                        5 = employed / government scheme
                        6 = self employed / government scheme
                        7 = self employed / employed / gov. scheme
```



```
    Name      Columns  Description
---------------------------------------------------------------------
38  RJEMPLT  114 - 115  Duration of contracts (E18 or V9)
                        If RJORG = 4 or 5 then E18
                        1 = permanent
                        2 = temporary
                        3 = seasonal
                        4 = DK
                        5 = NA
                        If RJORG = 6 then V9
                        1 = permanent
                        2 = temporary
                        3 = seasonal
                        4 = permanent / temporary
                        5 = temporary / seasonal
                        6 = permanent / seasonal
                        7 = permanent / temporary / seasonal

39  RJWORKT1 117 - 118  Type 1 of working (E20 or V10)
                        If RJORG = 4 or 5 then E20
                        If RJORG = 6 then V10
                        1 = working on employer's premises

40  RJWORKT2 120 - 121  Type 2 of working (E20 or V10)
                        If RJORG = 4 or 5 then E20
                        If RJORG = 6 then V10
                        1 = working home

41  RJWORKT3 123 - 124  Type 3 of working (E20 or V10)
                        If RJORG = 4 or 5 then E20
                        If RJORG = 6 then V10
                        1 = working on own business premises

42  RJWORKT4 126 - 127  Type 4 of working (V10)
                        If RJORG = 6 then V10
                        1 = other premises

43  RJEK0    129 - 130  Description of employment period (V7-0)
                        If RJORG = 6 then V7 (0)
                        1 = period included unemployment

44  RJEK1    132 - 133  Description of employment period (V7-1)
                        If RJORG = 6 then V7 (1)
                        2 = period included studying

45  RJEK2    135 - 136  Description of employment period (V7-2)
                        If RJORG = 6 then V7 (2)
                        3 = period included being on government program
46  RJEK3    138 - 139  Description of employment period (V7-3)
                        If RJORG = 6 then V7 (3)
                        4 = period included not working
```

```
    Name      Columns  Description
-----------------------------------------------------------------
47  RJEK4     141 - 142  Description of employment period (V7-4)
                         If RJORG = 6 then V7 (4)
                         5 = period included traveling

48  RJEK5     144 - 145  Description of employment period (V7-5)
                         If RJORG = 6 then V7 (5)
                         6 = work was continuous

49  RJFSIZE  147 - 150  Firm size (E21)
                         If RJORG = 4 or 5 then E21

50  RJSTAFF  152 - 155  Size of staff (E15)
                         If RJORG = 4 or 5 then E15

51  RJSIC     157 - 160  Employer's SIC code (E22)
                         If RJORG = 4 or 5 then E22

52  RJLOC     162 - 163  Employer's location (E23)
                         If RJORG = 4 or 5 then E23
                         1 = UK
                         2 = abroad
                         3 = both

53  RJSEX     165 - 166  Respondent's sex (1 men, 2 women)
54  RJBY      168 - 169  Respondent's birth year
55  RJBM      171 - 172  Respondent's birth month
56  RJIY      174 - 175  Date of interview: year
57  RJIM      177 - 178  Date of interview: month
```

Basic information about fwl2d.dat. The respondent's job history data file, `fwl2d.dat`, contains 71780 records for 10819 persons. 418 respondents (the difference to the number of respondents in the raw data file) do not have job grid entries,[15] or their job grid entries have serious errors.[16] As shown in Box 7.2, the file is ordered, first with respect to respondent's identification numbers, second with respect to the starting times of the main episodes. If a main episode splits into subepisodes, the information for the main episodes is repeated for each subepisode. The maximum number of records for each respondent is 43; the maximum number of main episodes is 28.

[15] The first two respondents without job grid entries have IDs 41281 and 50391, respectively.

[16] Note, however, that some of our checks may be false as a consequence of errors in the interview date.

Box 7.3 New version of the archive description file (`fwl2.des`)

```
# archive description file for fwl2.zoo

fwl2.zoo

 1  fwl2a.dat   1   10500  11237  1603
 2  fwl2b.dat   1      49 164295    15
 3  fwl2c.dat   1      52  21599    15
 4  fwl2d.dat   1     179  71780    57
99  fwl2.var    2       0   6678     0
```

As explained above, each record of the data file contains 57 variables. The record length is 178 characters plus one end-of-record character (line feed). The total size of the file is 12,848,620 bytes.

Updating the archive. Updating our data archive with the new data file is done in the usual way, already explained in previous sections. First, to put the data file into the archive, the command is simply

```
zoo ah fwl2.zoo fwl2d.dat
```

This command puts the compressed file `fwl2d.dat` into the already existing archive `fwl2.zoo`. In addition, one has to update the variable description file. This is simply done by appending the list of new variables to the end of the already existing variable description file `fwl2.var`. Note that we have used the logical file number 4 for the new data file.

In addition, we have to update the archive description file, `fwl2.des`, by adding one line describing the new data file. The updated description file is shown in Box 7.3.

Finally, the new archive can be checked with the TDA command

```
tda arcd=fwl2.des arcc
```

7.2 Description of Episodes

We now describe the main episodes contained in data file `fwl2d.dat`. In order to get a description we use command file `fwl15.cf` shown in Box 7.4. Here is an explanation of the commands.

1. The `arcd` command is used to open the connection to the data archive `fwl2.zoo`. Note that one has not directly to refer to the archive, but to the archive description file.

Box 7.4 Command file `fwl15.cf`

```
arcd = fwl2.des;    archive description file

dblock(50) = RJID;  define block mode
isel = RJEPN1[1];   select only records where RJEPN1 = 1

RJID   <8>[8.0] = A:RJID   ; (4) Respondent's ID
RJORG  <1>[1.0] = A:RJORG  ; (4) Origin state
RJDES  <1>[2.0] = A:RJDES  ; (4) Destination state
RJTSY  <1>[2.0] = A:RJTSY  ; (4) Starting time of episode: year
RJTSM  <1>[2.0] = A:RJTSM  ; (4) Starting time of episode: month
RJTFY  <1>[2.0] = A:RJTFY  ; (4) Ending time of episode: year
RJTFM  <1>[2.0] = A:RJTFM  ; (4) Ending time of episode: month
RJEPN1 <1>[2.0] = A:RJEPN1 ; (4) Subepisode number (1,...,RJNSEP)

TS = 12 * RJTSY + RJTSM;     starting time in century months
TF = 12 * RJTFY + RJTFM;     ending time in century months
DUR = TF + 1 - TS;           duration

# define single episode data
org = RJORG;
des = RJDES;
ts  = 0;
tf  = DUR;
```

Box 7.5-1 Episodes in `fwl2d.dat` with origin state 1.

SN	Org	Des	Episodes	Weighted	Mean Duration	TS Min	TF Max
1	1	1	1059	1059.00	41.86	0.00	446.00
1	1	2	331	331.00	19.70	0.00	156.00
1	1	3	123	123.00	14.85	0.00	169.00
1	1	4	2608	2608.00	12.62	0.00	308.00
1	1	5	175	175.00	13.81	0.00	98.00
1	1	6	144	144.00	14.11	0.00	205.00
1	1	11	60	60.00	11.52	0.00	62.00
1	1	12	18	18.00	5.94	0.00	17.00
1	1	13	3	3.00	6.00	0.00	11.00
1	1	14	21	21.00	9.33	0.00	55.00
1	1	15	2	2.00	4.50	0.00	6.00

2. The `dblock` command is used then to define block mode, that is, the program reads consecutive records, identified by variable `RJID`, as contiguous blocks of records.
3. The `isel` command provides an input select. While reading the data file, only those records are selected where the expression on the right-hand side is true, that is, takes a value not equal to zero. We select all records where variable `RJEPN1` has value 1, that is, if a main episode consists of more than one subepisode, we only select the first subepisode. The result should be that we get exactly one record for each main episode.

Box 7.5-2 Episodes in `fwl2d.dat` with origin state 2.

SN	Org	Des	Episodes	Weighted	Mean Duration	TS Min	TF Max
1	2	1	508	508.00	34.09	0.00	376.00
1	2	2	4129	4129.00	101.49	0.00	654.00
1	2	3	551	551.00	37.34	0.00	340.00
1	2	4	7071	7071.00	45.27	0.00	534.00
1	2	5	354	354.00	40.48	0.00	307.00
1	2	6	439	439.00	40.67	0.00	341.00
1	2	11	26	26.00	28.96	0.00	119.00
1	2	12	328	328.00	19.77	0.00	200.00
1	2	13	16	16.00	34.50	0.00	121.00
1	2	14	210	210.00	25.06	0.00	350.00
1	2	15	15	15.00	27.07	0.00	116.00
1	2	16	9	9.00	77.22	0.00	164.00

Box 7.5-3 Episodes in `fwl2d.dat` with origin state 3.

SN	Org	Des	Episodes	Weighted	Mean Duration	TS Min	TF Max
1	3	1	155	155.00	27.83	0.00	455.00
1	3	2	636	636.00	37.55	0.00	503.00
1	3	3	201	201.00	57.86	0.00	416.00
1	3	4	1501	1501.00	18.00	0.00	391.00
1	3	5	95	95.00	25.65	0.00	298.00
1	3	6	70	70.00	13.24	0.00	216.00
1	3	11	20	20.00	113.00	0.00	347.00
1	3	12	67	67.00	108.22	0.00	517.00
1	3	13	11	11.00	35.27	0.00	233.00
1	3	14	84	84.00	39.80	0.00	412.00
1	3	15	11	11.00	44.00	0.00	160.00
1	3	16	3	3.00	31.00	0.00	42.00

4. Then we define the necessary variables to be retrieved from the archive, that is, from data file `fwl2d.dat`. Note that a list of all variables can be requested with the command

   ```
   tda arcd=fwl2.des arcv1=fwl2d.dat
   ```

   In fact, we have used this command to get a first version of the command file.
5. The next step is to get starting and ending times of the episodes on an appropriate time axis. We choose *century months*, that is, dates are calculated in months beginning with the first month in this century.

Box 7.5-4 Episodes in `fwl2d.dat` with origin state 4.

SN	Org	Des	Episodes	Weighted	Mean Duration	TS Min	TF Max
1	4	1	2727	2727.00	55.51	0.00	520.00
1	4	2	7696	7696.00	65.04	0.00	613.00
1	4	3	1144	1144.00	50.20	0.00	519.00
1	4	4	19201	19201.00	52.02	0.00	582.00
1	4	5	685	685.00	67.83	0.00	426.00
1	4	6	450	450.00	35.24	0.00	266.00
1	4	11	187	187.00	64.14	0.00	373.00
1	4	12	988	988.00	120.62	0.00	622.00
1	4	13	193	193.00	101.14	0.00	551.00
1	4	14	1413	1413.00	87.75	0.00	604.00
1	4	15	106	106.00	86.74	0.00	474.00
1	4	16	59	59.00	59.36	0.00	200.00

Box 7.5-5 Episodes in `fwl2d.dat` with origin state 5.

SN	Org	Des	Episodes	Weighted	Mean Duration	TS Min	TF Max
1	5	1	145	145.00	46.31	0.00	337.00
1	5	2	288	288.00	91.64	0.00	575.00
1	5	3	58	58.00	67.93	0.00	541.00
1	5	4	332	332.00	47.65	0.00	429.00
1	5	5	707	707.00	99.28	0.00	650.00
1	5	6	22	22.00	76.09	0.00	320.00
1	5	12	9	9.00	110.89	0.00	362.00
1	5	13	1	1.00	104.00	0.00	104.00
1	5	14	70	70.00	92.56	0.00	428.00
1	5	15	20	20.00	112.10	0.00	325.00
1	5	16	6	6.00	101.33	0.00	295.00

Since years are given as 2 digits, this can easily be done by multiplying years with 12 and adding the month.

6. We take our data as a set of single episodes. Therefore each episode should begin at time zero and its ending time should be equal to its duration. The duration is calculated in variable **DUR**. Note that we have added one month because the ending time, as given in the raw data, is (probably) the last month of occupying the origin state.
7. Finally, we define single episode data for TDA. This is done with the commands `org` for the origin state, `des` for the destination state, `ts` for the starting time, and `tf` for the ending time.

Having prepared the command file, it can be executed with **TDA**. The program is invoked as

```
tda cf=fwl15.cf > fwl15.out
```


Box 7.5-6 Episodes in `fwl2d.dat` with origin state 6.

SN	Org	Des	Episodes	Weighted	Mean Duration	TS Min	TF Max
1	6	1	215	215.00	73.43	0.00	445.00
1	6	2	490	490.00	68.02	0.00	552.00
1	6	3	68	68.00	79.96	0.00	425.00
1	6	4	510	510.00	46.60	0.00	319.00
1	6	5	33	33.00	82.64	0.00	289.00
1	6	6	142	142.00	79.63	0.00	559.00
1	6	11	3	3.00	90.33	0.00	189.00
1	6	12	9	9.00	130.33	0.00	469.00
1	6	13	4	4.00	81.00	0.00	163.00
1	6	14	65	65.00	56.55	0.00	309.00
1	6	15	4	4.00	95.75	0.00	187.00
1	6	16	6	6.00	76.50	0.00	256.00

TDA's standard output is redirected into the output file `fwl15.out` (an arbitrary name). This output file will contain information about the results of executing the commands in the command file. In particular, it will contain a table with basic descriptive information about the episode data. This table is shown in Boxes 7-1,...,7-6.

Each of these boxes provides a description of episodes beginning in a specific origin state. There are six different origin states, see Box 7.1. Origin state 1 means „being unemployed“. We see that most people being in this state finally get a job (there are 2608 transitions into state 4, that is, working with one employer). The mean duration for episodes ending in this specific destination state is 12.6 months. However, this is not a correct estimate for the mean duration until a transition into employment occurs. A correct estimate would have to take into account the existence of censored episodes and episodes that end into another destination state.

Having successfully read our data file `fwl2d.dat` as a set of episode data, it would be possible to request further statistical procedures to describe and analyse the data. However, this will not be shown here since the focus of the current text is only on preparing the FWL data for subsequent analyses.

8 Partner's Job Grid

In exactly the same way as for respondent's, there is also a job grid for each partner. The information is contained in records of type 67, 72 and 73 of the raw data file. Building on this information, we create a job history data file for partners in basically the same way as done for respondent's in Section 7. The file will be called `fwl2e.dat`.

1. Variables in `fwl2e.dat` are the same as in the respondent's job history file (`fwl2d.dat`) with the following modifications:

   For partners, all variable names have the prefix `PJ` instead of `RJ` for the respondents.

   The variable `PJSEX` is now the partner's sex.

   The variables `PJBY` and `PJBM` now contain the partner's birth year and month, respectively. Note that `PJBY` is given as four digits, instead of two digits for `RJBY`.

   A description of all variables in `fwl2e.dat` can be requested with the command

   ```
   tda arcd=fwl2.des arcv1=fwl2e.dat
   ```

2. The structure of the partner's job history file is exactly the same as of the respondent's job history file, see the selected records in Box 8.1. All basic variables have the same meaning in both data files.
3. In creating `fwl2e.dat`, starting and ending times have been corrected in the same way as described in Section 7 for the respondent's job history file.

Basic information about `fwl2e.dat`. The partner's job history data file, `fwl2e.dat`, contains 16797 records for 4227 persons. Compared to the 4233 partners with at least one entry in the event matrix (see Section 6), we lost 6 persons who do not have job grid entries, or their job grid entries have serious errors. As shown in Box 8.1, the file is ordered in the same way as `fwl2d.dat`, first with respect to respondent's identification numbers, second with respect to the starting times of the main episodes. If a main episode splits into subepisodes, the information for the main episodes is repeated for each subepisode. The maximum number of records for each partner is 30; the maximum number of main episodes is 21.

Box 8.1 Some records in `fwl2e.dat` with basic variables

```
 91461  7  1  7  1 4  4 70  7 1 74  7 1  1  1 70  7 1 74  7 1
 91461  7  2  7  2 4  2 74  7 1 76  7 1  1  1 74  7 1 76  7 1
 91461  7  3  7  3 2  4 76  8 0 77  6 0  1  1 76  8 0 77  6 0
 91461  7  4  7  4 4  2 77  7 1 81 10 0  1  1 77  7 1 81 10 0
 91461  7  5  7  5 2  4 81 11 0 91  6 0  1  1 81 11 0 91  6 0
 91461  7  6  7  6 4  1 91  7 1 94  5 0  1  1 91  7 1 94  5 0
 91461  7  7  7  7 1  1 94  6 0 95  1 0  1  1 94  6 0 95  1 0
-------------------------------------------------------------
160041  2  1  2  1 4  2 83  7 1 87  7 1  1  1 83  7 1 87  7 1
160041  2  2  2  2 2  2 87  8 0 94  8 0  1  1 87  8 0 94  8 0
-------------------------------------------------------------
170041  2  1  1  1 4  4 87 10 0 94 11 0  2  1 87 10 0 -1 -1 0
170041  2  2  1  1 4  4 87 10 0 94 11 0  2  2 -1 -1 0 94 11 0
-------------------------------------------------------------
200151  3  1  3  1 4  2 79  7 1 90  9 0  1  1 79  7 1 90  9 0
200151  3  2  3  2 2  4 90 10 0 90 10 0  1  1 90 10 0 90 10 0
200151  3  3  3  3 4  4 90 11 0 94  8 0  1  1 90 11 0 94  8 0
-------------------------------------------------------------
200461  7  1  5  1 4  3 64  7 1 76  7 1  1  1 64  7 1 76  7 1
200461  7  2  5  2 3  2 76  8 0 76 11 0  1  1 76  8 0 76 11 0
200461  7  3  5  3 2  4 76 12 0 86  6 0  1  1 76 12 0 86  6 0
200461  7  4  5  4 4  2 86  7 1 92  4 0  3  1 86  7 1 87  7 0
200461  7  5  5  4 4  2 86  7 1 92  4 0  3  2 87  7 0 90  1 0
200461  7  6  5  4 4  2 86  7 1 92  4 0  3  3 90  1 0 92  4 0
200461  7  7  5  5 2  2 92  5 0 94  8 0  1  1 92  5 0 94  8 0
```

As explained above, each record of the data file contains 57 variables. The record length is 180 characters plus one end-of-record character (line feed). The total size of the file is 3,040,257 bytes.

8.1 The Final Data Archive

Updating our archive with the new data file is done in the usual way, already explained in previous sections. First, to put the data file into the archive, the command is simply

```
zoo ah fwl2.zoo fwl2e.dat
```

This command puts the compressed file `fwl2e.dat` into the already existing archive `fwl2.zoo`. In addition, one has to update the variable description file. This is simply done by appending the list of new variables to the end of the already existing variable description file `fwl2.var`. Note that we have used the logical file number 5 for the new data file.

Box 7.3 New version of the archive description file (`fwl2.des`)

```
# archive description file for fwl2.zoo

fwl2.zoo

 1  fwl2a.dat   1    10500  11237  1603
 2  fwl2b.dat   1       49 164295    15
 3  fwl2c.dat   1       52  21599    15
 4  fwl2d.dat   1      179  71780    57
 5  fwl2e.dat   1      181  16797    57
99  fwl2.var    2        0   6813     0
```

In addition, we have to update the archive description file, **`fwl2.des`**, by adding one line describing the new data file. The updated description file is shown in Box 8.2.

Finally, the new archive can be checked with the TDA command

```
tda arcd=fwl2.des arcc
```

Our data archive for the FWL data is now ready. It contains all information from the raw data file split into five separate file in order to be more easily accessible. The size of the archive is about 11.5 mega bytes, the size of the data files is about 143 mega bytes.

8.2 Description of Episodes

As a final step, we check whether we are able to read the partner's job history data correctly with TDA. Our command will be **`fwl20.cf`**. It is basically identical to **`fwl15.cf`**, used in Section 7.2, only the variable names have been changed to refer to the partner's job history file (that is, the prefixes have been changed from `RJ` to `PJ`).

Part of TDA's output is shown in Boxes 8-1,...,8-6. Each table describes a set of episodes with a specific origin state.

Box 8.5-1 Episodes in `fwl2e.dat` with origin state 1.

SN	Org	Des	Episodes	Weighted	Mean Duration	TS Min	TF Max
1	1	1	251	251.00	43.52	0.00	425.00
1	1	2	63	63.00	24.10	0.00	191.00
1	1	3	22	22.00	25.73	0.00	128.00
1	1	4	447	447.00	13.21	0.00	139.00
1	1	5	28	28.00	13.39	0.00	43.00
1	1	6	28	28.00	8.68	0.00	94.00
1	1	11	8	8.00	5.75	0.00	15.00
1	1	12	3	3.00	6.67	0.00	11.00
1	1	14	3	3.00	24.67	0.00	41.00
1	1	15	1	1.00	4.00	0.00	4.00

Box 8.5-2 Episodes in `fwl2e.dat` with origin state 2.

SN	Org	Des	Episodes	Weighted	Mean Duration	TS Min	TF Max
1	2	1	73	73.00	31.84	0.00	317.00
1	2	2	1670	1670.00	102.23	0.00	556.00
1	2	3	95	95.00	43.57	0.00	428.00
1	2	4	1586	1586.00	53.04	0.00	375.00
1	2	5	111	111.00	43.05	0.00	312.00
1	2	6	87	87.00	44.71	0.00	211.00
1	2	11	7	7.00	18.00	0.00	71.00
1	2	12	104	104.00	24.42	0.00	256.00
1	2	13	5	5.00	24.40	0.00	50.00
1	2	14	22	22.00	23.50	0.00	84.00
1	2	15	3	3.00	30.00	0.00	61.00
1	2	16	1	1.00	16.00	0.00	16.00

Box 8.5-3 Episodes in `fwl2e.dat` with origin state 3.

SN	Org	Des	Episodes	Weighted	Mean Duration	TS Min	TF Max
1	3	1	27	27.00	41.96	0.00	303.00
1	3	2	98	98.00	34.70	0.00	303.00
1	3	3	109	109.00	96.21	0.00	461.00
1	3	4	301	301.00	15.00	0.00	199.00
1	3	5	28	28.00	22.82	0.00	230.00
1	3	6	10	10.00	21.20	0.00	71.00
1	3	11	4	4.00	38.75	0.00	123.00
1	3	12	14	14.00	31.21	0.00	242.00
1	3	13	8	8.00	15.50	0.00	37.00
1	3	14	7	7.00	14.57	0.00	57.00
1	3	15	1	1.00	165.00	0.00	165.00

Box 8.5-4 Episodes in `fwl2e.dat` with origin state 4.

SN	Org	Des	Episodes	Weighted	Mean Duration	TS Min	TF Max
1	4	1	515	515.00	76.48	0.00	474.00
1	4	2	2014	2014.00	94.89	0.00	612.00
1	4	3	262	262.00	82.97	0.00	482.00
1	4	4	4448	4448.00	76.81	0.00	592.00
1	4	5	217	217.00	74.39	0.00	459.00
1	4	6	49	49.00	52.39	0.00	250.00
1	4	11	31	31.00	102.61	0.00	511.00
1	4	12	244	244.00	139.89	0.00	499.00
1	4	13	35	35.00	97.63	0.00	364.00
1	4	14	256	256.00	112.56	0.00	498.00
1	4	15	26	26.00	108.96	0.00	385.00
1	4	16	10	10.00	83.10	0.00	247.00

Box 8.5-5 Episodes in `fwl2e.dat` with origin state 5.

SN	Org	Des	Episodes	Weighted	Mean Duration	TS Min	TF Max
1	5	1	37	37.00	66.59	0.00	430.00
1	5	2	112	112.00	118.18	0.00	451.00
1	5	3	20	20.00	61.45	0.00	332.00
1	5	4	85	85.00	53.84	0.00	223.00
1	5	5	311	311.00	124.89	0.00	529.00
1	5	6	3	3.00	46.33	0.00	90.00
1	5	11	1	1.00	23.00	0.00	23.00
1	5	12	4	4.00	97.50	0.00	155.00
1	5	13	1	1.00	114.00	0.00	114.00
1	5	14	13	13.00	132.08	0.00	298.00
1	5	15	13	13.00	175.08	0.00	354.00

Box 8.5-6 Episodes in `fwl2e.dat` with origin state 6.

SN	Org	Des	Episodes	Weighted	Mean Duration	TS Min	TF Max
1	6	1	33	33.00	82.91	0.00	397.00
1	6	2	84	84.00	98.45	0.00	568.00
1	6	3	17	17.00	62.88	0.00	217.00
1	6	4	72	72.00	71.21	0.00	314.00
1	6	5	8	8.00	111.38	0.00	386.00
1	6	6	58	58.00	105.64	0.00	408.00
1	6	11	1	1.00	91.00	0.00	91.00
1	6	12	2	2.00	101.00	0.00	168.00
1	6	13	2	2.00	104.50	0.00	193.00
1	6	14	5	5.00	65.80	0.00	131.00
1	6	16	4	4.00	8.00	0.00	8.00

References

Blossfeld, H.-P., Rohwer, G. (1995). Techniques of Event History Modeling. New Approaches to Causal Analysis. Mahwah, NJ: Lawrence Erlbaum 1995.

Rohwer, G. (1994). Using TDA with the Family and Working Lives Survey. Bremen: University of Bremen (mimeo).