CO

INSTRUCTION MANUAL

Paul Williamson

Working Paper 2007/1
(v. 07.06.25)

Population Microdata Unit,
Department of Geography,
University of Liverpool

June 2007

CONTENTS

Document History

Revision of Williamson P and Huang Z (2002) Pop91 Instruction Manual v2.0, Working Paper 2002/1, Population Microdata Unit, Dept. of Geography, University of Liverpool to take account of code streamlining and generalisation undertaken in 2005 and 2007.

CO: a computer algorithm for reweighting microdada to fit small-area constraints

OVERVIEW

1. Program structure

The structure of CO is shown in Figure 1. The main program, *CO.exe*, requires as inputs (i) a set of survey microdata, suitably formatted for input to CO; (ii) a set of constraints to which these survey data are to be reweighted. The main outputs are a set of survey weights which 'best fit' the supplied external constraints, and a summary file reporting the fit of these weights to the supplied constraints for each estimation area.

The root directory of the programme suite is *CO*, with program code, input data and documentation arranged within sub-folders is outlined below. For test purposes, CO comes pre-supplied with a set of 'dummy' input datasets.

Folder	Content
Documentation, code and inputs	
Code	Synthetic estimation program code (Fortran 95)
Code/Release/Win23	Program code pre-compiled into linkable 'object' files
Data	CO run-time input data, including user-defined run-time parameters
Data/Original Survey	Original survey data and SPSS syntax for converting survey data to format required by CO
Documentation	CO documentation
Output	
Estimates	Constraints and their estimated equivalents for each estimation area
Estimate_Fit	Summary goodness-of-fit statistics for each estimated area
Combinations	The sets of households (combinations) identified as most representative (optimal) for each estimated area
Weights	Area-specific survey weights produced by CO

The source code assumes that all files are located relative to the root folder *C0*. Provided that the file structure and naming convention below this root directory are left unchanged, the program can be located wherever required (e.g. D:\Synthetic Population\CO; C:\CO).

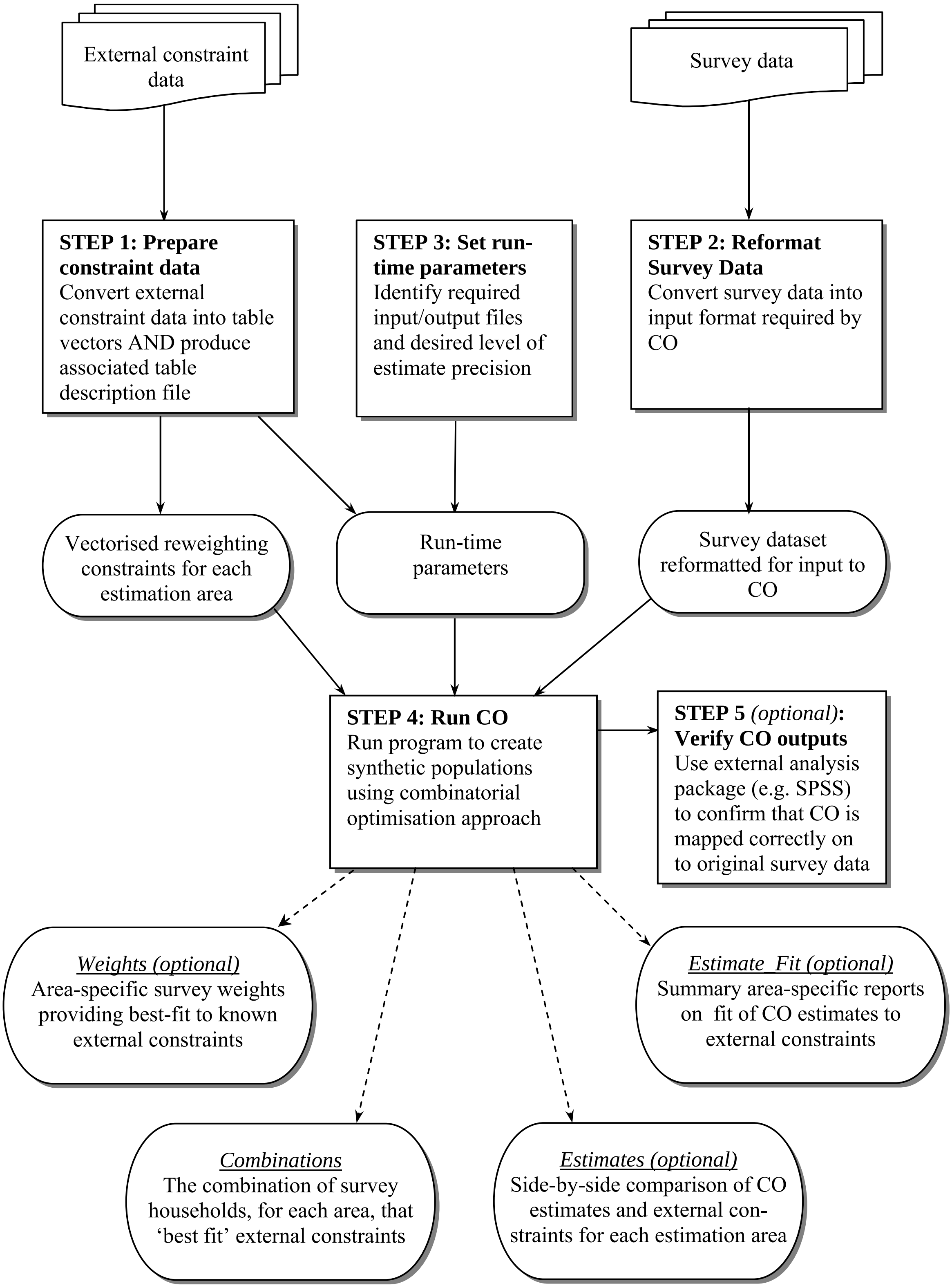

External constraint data
Survey data
STEP 1: Prepare constraint data
Convert external constraint data into table vectors AND produce associated table description file
STEP 3: Set run-time parameters
Identify required input/output files and desired level of estimate precision
STEP 2: Reformat Survey Data
Convert survey data into input format required by CO
Vectorised reweighting constraints for each estimation area
Run-time parameters
Survey dataset reformatted for input to CO
STEP 4: Run CO
Run program to create synthetic populations using combinatorial optimisation approach
STEP 5 (optional): Verify CO outputs
Use external analysis package (e.g. SPSS) to confirm that CO is mapped correctly on to original survey data
Weights (optional)
Area-specific survey weights providing best-fit to known external constraints
Estimate_Fit (optional)
Summary area-specific reports on fit of CO estimates to external constraints
Combinations
The combination of survey households, for each area, that 'best fit' external constraints
Estimates (optional)
Side-by-side comparison of CO estimates and external con-straints for each estimation area

Figure 1 CO program structure

2. Program distribution

The CO programme suite is publicly available via the project website (http://pcwww.liv.ac.uk/~william/microdata). It is supplied with Windows-compatible executable code, the source code for this executable, a set of dummy input data (for test purposes), a set of accompanying documentation and some results analysis/extraction programmes. This manual outlines the steps required to (i) prepare data for input to CO; (ii) use CO for survey reweighting; (iii) independently verify CO results.

Users are welcome to copy and adapt the source code as they require, provided that appropriate acknowledgement of the original source is made. Throughout, standard Fortran 95 code is used, with the exception of the use of calls to a Salford Fortran95 extension, MKDIR@, used to create output folders, using a nested county and district hierarchy where appropriate. If a Salford compiler is unavailable, and program recompilation is required, the lines calling the MKDIR@ subroutine in each program can be commented out, and the subfolders created 'manually' via Windows Explorer.

3. Getting started

To run CO, two sets of data are required: (i) a set of known area distributions (constraints); (ii) a set of survey microdata. Dummy versions of these two datasets are supplied, and are referred to throughout this manual for illustrative purposes. CO also requires a number of run-time parameters, supplied via a range of control files. The supplied control files, left unaltered, will prompt CO to produce estimates using the supplied dummy data.

Users are advised to familiarise themselves with CO by first running CO using these dummy datasets. To do this, unzip the CO program suite, navigate to the root folder CO using *Windows Explorer* or similar, then double-click on the file *CO.exe*. Provided that the folder structure outlined in section 1 above has been maintained, and a copy of the Salford Fortran 95 run-time library (*salflibc.dll*) is located in the same folder as *CO.exe*, the program will execute. CO will then produce a set of outputs which the user may explore following the information given in section of the manual entitled 'STEP 4: Running CO and understanding the ouputs'.

The remainder of this manual describes how users can customise the programme suite to their own needs, substituting the supplied dummy data with their own constraint and survey data (STEPS 1 and 2), and amending the control files accordingly (STEP 3). Of course, all software has limitations. The current operational limits of CO (maximum number of constraints; maximum size of survey microdata file etc.) are summarised in Table 1.

Table 1 CO operational limitations

Input / Outputs	Maximum no.	Comment/Manual reference
Estimation areas [a]	1,000	No limit if Weights reporting option is turned off
Households per estimation area	100,000	STEP 3, sub-section 1
Constraint tables	20	STEP 1, sub-section 1
Constraints per table	120	STEP 1, sub-section 2
Survey households	500,000	STEP 2, sub-section 4
Survey persons	2,000,000	STEP 2, sub-section 3
Replications	100	STEP 3, sub-section 3.1

STEP 1: Prepare constraint data

1 Background

The purpose of the CO algorithm is to reweight a set of survey microdata such that, when aggregated, the weights add up to match a set of known external constraints for a specified geographical area. These constraints might comprise a simple series of unrelated counts, such as those illustrated in Table 2(a). However, more commonly the reweighting constraints are likely to be drawn from a set of one or more tabulations, whether uni- or multi-variate (Table 2b,c).

Table 2 Example constraints

(a) Set of stand-alone counts

Constraint	Count
Residents	209
Households	71
Unemployed	32

(b) Set of univariate tables

Age	Count
0-15	60
16-64	97
65+	52

Sex	Count
Male	72
Female	137

Cars	Count
0	15
1	21
2	24
3+	11

H/hold size	Count
1 persons	15
2 persons	10
3 persons	22
4+ persons	24

(c) Set of multivariate tables

	Sex	
Age	Male	Female
0-15	11	49
16-64	40	57
65+	21	31

	H/hold size (Persons)			
Cars	1	2	3	4+
0	*5*	*2*	*3*	*5*
1	*3*	*6*	*8*	*4*
2	*7*	*1*	*8*	*8*
3+	*0*	*1*	*3*	*7*

For the purposes of CO, any set of mutually exclusive constraints drawn from the same tier of a population hierarchy (i.e. persons OR households) comprises a 'table'. On this basis each of the contraints in 2(a) is a 'table' ('unemployed' and 'resident' are not mutually exclusive categories; 'households' is a household-level count); as is each of the univariate distributions in 2(a) and each of the bivariate distributions in 2(c). In particular note that, if required, table marginals (table row/column totals) should be supplied as univariate constraint 'tables' separate from the multivariate table of interior table counts to which they relate (as illustrated by Table 2(b) and 2(c) above).

To run CO, a file containing constraint data in a suitable format is required. A set of dummy constraint data is provided with CO for test purposes (section 2), the format of which is explained in section 3. Also required is a file providing CO with a description of each constraint table (section 4).

2. CO dummy constraint data

The dummy constraint data supplied with CO is located in folder *CO/Data*, in the file *CO_Dummy_estimation_contraints.txt*. This file contains the constraints described in Table 3 for each of 10 areas. (For full specification of variable categories, see Table 2c above.)

Table 3 Dummy constraint data

Order processed	Table Name	Tabulation	Constraints per table
1	P1	Age (3 categories) x Sex (2 categories)	6
2	H1	Cars in household (0-3) x Household Size (1-4 persons)	16
Total constrained cells			22

3. Constraint data input format

As indicated by the file *CO_Dummy_estimation_contraints.txt* the required constraint data input format is as follows:

AreaCode unique area identier (name or alphanumeric code) [max. 20 characters; no spaces]

Households Number of occupied households in area (control target) – required because disclosure control may lead to differences in household totals across published tables. (If only person-level constraints are being used, replace with the target number of persons) [Max. 100,000]

Constraint 1
Constraint 2
…
Constraint n
} List of area constraints, with any person-level contraint tables (e.g. age x sex) PRECEDING any household-level constraint tables (e.g. cars in household x persons in household) [Max. 120 counts per constraint table]

It is recommended, but not essential, that the constraints from each constraint table are listed in order from top-left to bottom-right.

In the following example, the constraints in Table 2(c) have been appropriately reformatted:

```
Area_1    71     11  49  40  57  21  31     5  2  3  5  3  6  8  4  7  1  8  8  0  1  3  7
AreaCode  House-    Person-level constraints          Household-level constraints
          holds
```

Example 1 Estimation constraints formatted for input to CO

Although the data items above are space-separated, comma- and tab-separated formats are also permissible.

Nor do the constraints all have to be placed on a single row (although this aids data inspection and error checking). Hence both Examples 2 and 3 illustrate permissible formats. However, note that each new area MUST start on a new row.

```
Area_1   71   11   49   40   57   21   31   5   2   3   5   3   6   8   4   7   1   8   8   0   1   3   7
Area_2   39    6    3   11    8    3    5   3   2   4   2   3   0   1   3   4   1   2   4   1   2   4   3
```

Example 2 Estimation constraints formatted for input to CO (one row per area)

```
Area_1   71   11   49   40   57   21   31   5
  2   3   5   3   6   8   4   7   1   8
  8   0   1   3   7
Area_2   39    6    3   11    8    3    5   3
  2   4   2   3   0   1   3   4   1   2
  4   1   2   4   3
```

Example 3 Estimation constraints formatted for input to CO (multiple rows per area)

Adding further sets of constraints from other tables is simply a matter of extension. In Example 4, in the additional person-level constraints presented in Table 4 have been added (remembering to list ALL person-level constraints before any household-level constraints).

Table 4 Additional person-level constraints

Marital	Health status	
status	Ill	Well
Single	2	8
Married	5	10
Widowed	6	4
Divorced	4	1

```
Area_1   40   12   12   10   16   13   16   2   8   5   10   6   4   4   1   8   2   0   0   6   6   3   0   4
4   2   0   2   0   3   0
```

Example 4 Formatted estimation constraints drawn from three separate constraint tables

The table description file supplied with CO, *CO_Dummy_constraining_tables_ info.txt*, is located in the folder *CO/Data*. User-supplied estimation constraints should be saved as a plain-text file in the folder *CO/Data*, using a filename of up to 50 alpha-numeric characters. There is no operational limit, other than processing time, on the number of input estimation areas this file can contain.

4. Creating a table information file

In addition to creating a formatted set of constraint data it is necessary to create a file that provides CO with brief information about the tables to be used as estimation constraints. The format of this second file is as follows:

Line 1: Number of user-supplied constraint tables

Subsequent lines, using one line (row) per constraint table:
Column 1 – Table Name (max. 4 characters; no spaces)
Column 2 – No. of constraining counts in table
Column 3 – Chi-square critical value for table (p=0.05). (This value can be calculated in Excel using the formula **=chiinv(0.05,*X*)**, where *X* = no. of cells in table vector (see Voas and Williamson (2001) for justification)
Column 4 – Table type (1=person-level; 2=household-level)
Column 5 – Table switch (1=on; 0=off) (Switch)

The list of constraints in Example 4 draws upon counts from three separate constraint tables. Example 5 illustrates the table description that would required by CO to accompany the formatted constraint counts. Note that: (i) the constraint table information **MUST** be listed in **precisely** the same order as the tables are listed in the formatted constraint data; (ii) the first line of the table information file indicates the number of constraint table descriptions to follow.

```
3
P1    6  12.59 1 1
P2    8    7.34 1 1
H1   16  26.30 2 1
```

Example 5 Table information file for constraints listed in Example 4

The table information file supplied with CO, *CO_Dummy_constraining_tables_ info.txt*, is located in the folder *CO/Data*. Any user-supplied table information file should be saved in the same folder, in plain-text format, using a filename of up to 50 alpha-numeric characters.

STEP 2: Reformat Survey Data

1. Overview

In order for CO to run successfully, the user has to map each person and household in their survey microdata onto the relevant person- and household-level constraints in their constraint data. As currently configured, CO expects two input files. The first (section 3) maps survey individuals onto person-level constraints. The second (section 4) map survey households onto household-level constraints AND identifies which survey individuals fall in which survey households. Section 5 outlines one method for automating the conversion of raw survey data into the required CO-formatted inputs files. But as a preliminary to all of the above, it is first necessary to uniquely identify each constraint table cell (section 2).

2. Numbering constraint table cells

In order to uniquely identify each constraint table cell it is recommended that the cells within each user-supplied constraint table are numbered in order from top-left to bottom-right, as in the example below. The combination of table name and cell number then uniquely identifies each constraint table cell.

Table P1		
	Sex	
Age	Male	Female
0-15	1	2
16-64	3	4
65+	5	6

Table H1				
	H/hold size (Persons)			
Cars	1	2	3	4+
0	*1*	*2*	*3*	*4*
1	*5*	*6*	*7*	*8*
2	*9*	*10*	*11*	*12*
3+	*13*	*14*	*15*	*16*

Example 6 Numbering constraint table cells

3. Mapping onto person-level constraints

3.1 The underlying principle

Table 5 provides an illustrative extract from *CO_Dummy_survey.txt*, the dummy survey file supplied with CO, which may be found in the folder *CO/Data/Original Survey*.

Table 5 Extract of supplied dummy survey microdata

Person	Household	Household member	Age	Sex [1=male; 2=female]	Cars in household	Persons in household
1	1	1	60	1	0	2
2	1	2	7	2	0	2
3	2	1	54	1	2	3
4	2	2	60	2	2	3
5	2	3	34	1	2	3

Using the table constraint numbering set out in Example 6, the individuals in Table 5 can be 'mapped' on to specific cells in the person-level table P1 as shown in Table 6.

Table 6 Mapping of survey individuals onto constraint table cells

Person	Household	Household member	Cell in Table P1
1	1	1	3
2	1	2	2
3	2	1	3
4	2	2	4
5	2	3	3

3.2 Required input format

A file containing a mapping of each survey individual onto each person-level constraint table is an essential CO input. The required format of the CO input file recording these mappings is as follows:

Row 1: Mapping of first person listed in survey microdata onto each person-level constraint table forming part of the supplied estimation constraints. NOTE: Person-level table mappings MUST be listed in the precise order used in the input tabular constraint data (c.f. STEP 1, section 3).

Row 2: person-level mapping(s) for second person listed in survey microdata

Row X: person-level mapping(s) for X^{th} person listed in survey microdata

If table P1 were the only person-level constraint file, then the person-level survey mapping file would comprise only the contents of column 4 from Table 6.

Given the CO definition of a constraint 'table' (STEP 1, section 1), note that each person in a survey can fall into one, and only one, cell within any given person-level constraint table. If an individual does not fall in a particular table (e.g. because they are a 'visitor', whereas the table covers 'residents' only), this should be indicated by recording a mapping of '**0**' .

The file *CO_Dummy_constraint_mapped_survey_Persons.txt*, supplied with CO in folder *CO/Data*, contains the person-level mapping of *CO_Dummy_Survey.txt*. Section 5 describes how the creation of this mapping file was automated using SPSS. Any user-supplied person-level mapping file should be saved as plain-text, with all mappings space-, comma- or tab-separated, and placed in the folder *CO/Data* using a filename up to 50 characters long,

4. Mapping onto household-level constraints

4.1 The underlying principle

Table 7 shows how the housholds in the survey extract presented above (Table 5) may be 'mapped' on to specific cells in the household-level table, H1, using the cell numbering in Example 6.

Table 7 Mapping of survey households onto constraint table cells

Household	Cell in Table H1	First Person in h/hold	Last Person in h/hold
1	2	1	2
2	11	3	5

4.2 Required input format

A file containing a mapping of each survey household onto each hosuehold-level constraint table is an essential CO input. The required format of the CO input file recording these mappings is as follows:

Row 1: (i) unique household ID; (ii) household-level mapping(s) for first household listed in survey microdata; (iii) unique person ID of first person listed in household; (iv) unique person ID of last person listed in household. (i.e. the contents of columns 1 to 4 in Table 7 above.)

If more than one household-level constraint table exists, ensure that the mappings for these other household-level tables are listed consecutively on the same row, in precisely the same order as that used in when inputting the tabular constraint data (c.f. STEP 1, section 3).

Row 2: household-level mapping(s) for second household listed in survey microdata

Row X: household-level mapping(s) for X^{th} household listed in survey microdata

Given the CO definition of a constraint 'table', note that each household in a survey can fall into one, and only one, cell within any given household-level constraint table. If a household does not fall in a particular table (e.g. because it is a 'visitor' household, whereas the table covers 'resident' households only), this should be indicated by recording a mapping of '**0**' .

The file *CO_Dummy_constraint_mapped_survey_Households.txt*, supplied with CO in folder *CO/Data*, contains the household-level mapping of *CO_Dummy_Survey.txt*. Section 5 describes how the creation of this mapping file was automated using SPSS. Any user-supplied household-level mapping file should be saved as plain-text, with all mappings space-, comma- or tab-separated, and placed in the folder *CO/Data* using a filename up to 50 characters long,

5. Using SPSS to create a CO-formatted version of original survey microdata

The precise method by which raw survey data is converted into the required CO-format, with each person and household is mapped onto a specific cell in each constraint table, is left to the user. However, the SPSS syntax code used to convert the file *CO_Dummy_survey.txt* into the CO-formatted input files

CO_Dummy_constraint_mapped_survey_Persons.txt
and *CO_Dummy_constraint_mapped_survey_Hholds.txt*

is supplied as part of the software suite (*Format survey data for CO.sps*, located in folder *CO/Data/Original Survey*), and is listed in Box 1, for adaption by the user as required.

Box 1 SPSS code for converting survey data into CO-formatted input files

NOTE: Syntax below will work ONLY if each comment line (starting with an asterisk) is followed by a blank line

```
*(1) Read in raw survey data and save as SPSS datafile

GET DATA  /TYPE = TXT
 /FILE = 'CO_Dummy_survey.txt'
 /DELCASE = LINE
 /DELIMITERS = " "
 /ARRANGEMENT = DELIMITED
 /FIRSTCASE = 1
 /IMPORTCASE = ALL
 /VARIABLES =
 Record F2.0
 HH_ID F2.0
 Person F1.0
 Age F2.0
 Sex F1.0
 Cars F1.0
 HH_Size F1.0
 .
CACHE.
EXECUTE.
DATASET NAME DataSet1 WINDOW=FRONT.

Variable label Record 'Unique record [person] identifier'.
Variable label HH_ID 'Unique household identifier'.
Variable label Person 'Person no. within household'.
Variable label Age 'Age (years)'.
Variable label Sex 'Sex'.
Variable label Cars 'Cars in household'.
Variable label HH_Size 'Persons in household'.

Value labels Sex 1 'Male' 2 'Female'.

SAVE OUTFILE='CO_Dummy_survey.sav'
 /COMPRESSED.


*(2) Recode survey variables to match categories/classes used in constraint tables

*(a) Age

RECODE  Age  (0 thru 15=1)  (16 thru 64=2)  (65 thru Highest=3)  INTO  Age_gp .
VARIABLE LABELS Age_gp 'Grouped Age'.
VALUE LABELS Age_gp 1 'Child' 2 'Adult' 3 'Pensioner'.
EXECUTE .

*(b) Sex [already in required categories]

*(c) Cars [already in required categories]

*(d) HH_Size [already in required categories]
```

```
*(3) For each constraint table in turn, identify the cell within which person/household falls

*(a) Constraint Table P1: Age (child/adult/pensioner) x Sex (male/female)
* [Value initially set to 0; persons not falling in table will retain this coding of 0]

COMPUTE Table_P1=0.
IF ((Age_gp=1) and (Sex=1)) Table_P1 = 1 .
IF ((Age_gp=1) and (Sex=2)) Table_P1 = 2 .
IF ((Age_gp=2) and (Sex=1)) Table_P1 = 3 .
IF ((Age_gp=2) and (Sex=2)) Table_P1 = 4 .
IF ((Age_gp=3) and (Sex=1)) Table_P1 = 5 .
IF ((Age_gp=3) and (Sex=2)) Table_P1 = 6 .

VARIABLE LABELS Table_P1 'Constraint Table P1' .
FORMAT Table_P1 (f2.0).
EXECUTE .

*(b) Constraint Table H1: HH_Size (1-4 persons) x Cars in household (0-3)
* [Value initially set to 0; households not falling in table will retain this coding of 0]

COMPUTE Table_H1=0.
IF ((HH_Size=1) and (Cars=0)) Table_H1 = 1 .
IF ((HH_Size=2) and (Cars=0)) Table_H1 = 2 .
IF ((HH_Size=3) and (Cars=0)) Table_H1 = 3 .
IF ((HH_Size=4) and (Cars=0)) Table_H1 = 4 .
IF ((HH_Size=1) and (Cars=1)) Table_H1 = 5 .
IF ((HH_Size=2) and (Cars=1)) Table_H1 = 6 .
IF ((HH_Size=3) and (Cars=1)) Table_H1 = 7 .
IF ((HH_Size=4) and (Cars=1)) Table_H1 = 8 .
IF ((HH_Size=1) and (Cars=2)) Table_H1 = 9 .
IF ((HH_Size=2) and (Cars=2)) Table_H1 = 10 .
IF ((HH_Size=3) and (Cars=2)) Table_H1 = 11 .
IF ((HH_Size=4) and (Cars=2)) Table_H1 = 12 .
IF ((HH_Size=1) and (Cars=3)) Table_H1 = 13 .
IF ((HH_Size=2) and (Cars=3)) Table_H1 = 14 .
IF ((HH_Size=3) and (Cars=3)) Table_H1 = 15 .
IF ((HH_Size=4) and (Cars=3)) Table_H1 = 16 .

VARIABLE LABELS Table_H1 'Constraint Table H1' .
FORMAT Table_H1 (f2.0).
EXECUTE .


*(4) Identify the record number of the first and last person listed in each household
*** FILE MUST BE IN ASCENDING ORDER OF (i) HH_ID and (ii) PERSONS within household BEFORE doing this***

*(a) Sort survey by (i) HH_ID; (ii) persons within household to ensure file in required order

SORT CASES BY HH_ID (A) Person (A) .

*(b) Find first/last person listed in each household

AGGREGATE
  /OUTFILE=*
  MODE=ADDVARIABLES
  /BREAK=HH_ID
  /First_Person = FIRST(Record) /Last_Person = LAST(Record).
```

```
*(5) Export selected data from SPSS into CO-formatted inputs files
* (Two output files: one for person-level tables; one for household-level tables;
* results placed in folder CO_Dummy/Data)

*** FILE MUST BE IN ASCENDING ORDER OF (i) HH_ID and (ii) PERSONS within household BEFORE EXPORT ***
*** (done in (4) above); otherwise later matching of survey with CO weights file may fail          ***

*(a) Person-level tables first...
*[IMPORTANT: ensure that table mappings are written out in the same order used in CO-formatted table constraint
input file]


*(a) Person-level tables first...
*[IMPORTANT: ensure that table mappings are written out in the same order used in CO-formatted table constraint
input file]

WRITE OUTFILE='../CO_Dummy_constraint_mapped_survey_Persons.txt'
  TABLE
  /Table_P1 (F4.0) .
EXECUTE.

*(b) household-level tables

*(i) Delete excess cases, retaining only the first record (person) from each household
* [i.e. retaining only records where 'record' (unique person id) = 'first_person' (first person in household)]

USE ALL.
FILTER OFF.
USE ALL.
SELECT IF(Record=First_Person).
EXECUTE .

*(ii) write-out household-level mappings
*[IMPORTANT: ensure that table mappings are written out in the same order used in CO-formatted table constraint
* input file, preceded by survey household ID and followed by First_Person, then Last_Person]

WRITE OUTFILE='../CO_Dummy_constraint_mapped_survey_Hholds.txt'
  TABLE
  /HH_ID (F8.0) Table_H1 (F4.0) First_Person (F8.0) Last_Person (F8.0) .
EXECUTE.
```

STEP 3: Set run-time parameters

Three input files contain information used by CO to control program operation. The first (section 1) lists the set of input areas for which CO is to produce area-specific survey weights. The second (section 2), allows the user to specify the input files CO should use. A final file (section 3) contains a range of general and more advanced options to allow the user to (i) specify the outputs required; (ii) select the measure-of-fit CO should try to minimise whilst attempting to meet user-supplied estimation constraints; (iii) 'tweak' algorithm performance.

1. CO_Dummy_area_list.txt

This file is used to control which areas CO produces estimates for. To produce estimates for each input area listed in *CO_Dummy_contraints.txt* (or equivalent), place a list of each of the input area names (case-sensitive), one row per name, in a separate file (e.g. *CO_Dummy_area_list.txt*), taking care to retain the input area order. To produce estimates for a sub-set of the input areas supplied via *CO_Dummy_estimation_constraints.txt* (e.g. to run CO for only one or two areas for test purposes), simply truncate the list of estimation areas in this output control file appropriately, as illustrated in Table 8.

Table 8 Alternative area lists and their impact on CO outputs

Input List	*Control List*	*Control List*	*Control List*	*Control List*
Area_1	Area_1	area_1	Area_1	Area_2
Area_2	Area_2	area_2	Area_2	Area_3
Area_3	Area_3	area_3		
Area_4	Area_4	area_4		
Area_5	Area_5	area_5		
Area_6	Area_6	area_6		
Area_7	Area_7	area_7		
Area_8	Area_8	area_8		
Area_9	Area_9	area_9		
Area_10	Area_10	area_10		
Estimates produced for input areas…	ALL	NONE (CO is case-senstive)	Area_1 and Area_2	NONE (truncate list only)

If the full Weights reporting option is turn on (section 3.2), CO is able to process a maximum of 1000 estimation areas per run. If this option is turned off, there is no operational limit, other than processing time, on the number of input estimation areas that CO can process in a given run.

2. CO_filelist.txt

To provide maximum flexibility, the choice of filenames for the input files created as part of Steps 1 and 2 has been left to the user. In addition to allowing the user to choose filenames that make sense in a particular user context, this flexibility also means that the user can maintain multiple input files for use as and when required (e.g. *English_contstraining_tables.txt*; *Welsh_constraining_tables.txt*;…).

Prior to running CO, therefore, it is necessary to update the contents of *CO_filelist.txt* (located in the folder *CO/Data*) to point CO towards the required set of run-time input/output files. All filenames must be given *relative* to the folder in which the executable code *CO.exe* is located, and must also be given inside quote marks. Each relative filename may comprise up to 50 alpha-numeric characters. Note that the name of *CO_filelist.txt* MUST remain unchanged.

The supplied version of *CO_filelist.txt* (see Box 2) allows CO to run using the supplied Dummy data.

Box 2 *CO-filelist.txt*

```
'Data/CO_Dummy_constraint_mapped_survey_Hholds.txt'
'Data/CO_Dummy_constraint_mapped_survey_Persons.txt'
'Data/CO_Dummy_estimation_constraints.txt'
'Data/CO_Dummy_area_list.txt'
'Data/CO_Dummy_constraining_tables_info.txt'
'Data/CO_Dummy_control_parameters.txt'
'Data/CO_random_seednumbers.txt'
```

A brief outline of the purpose of each file/foldername listed in *CO_filelist.txt* is given below, along with pointers to more the detailed explanations provided elsewhere in this manual.

Line 1: *Constraint-mapped household-level microdata*
[Default: 'CO_Dummy_constraint_mapped_Hholds.txt']
Mapping of original survey data onto household-level constraint table(s) (STEP 2, sections 4&5).

Line 2: *Constraint-mapped person-level microdata*
[Default: 'CO_Dummy_constraint_mapped_survey_Persons.txt']
Mapping of original survey data onto person-level constraint table(s) (STEP 2, sections 3&5).

Line 3: *Estimation constraints*
[Default: 'CO_Dummy_estimation_constraints.txt']
Estimation contraints for one or more estimation areas (see STEP1, section 3).

Line 4: *List of estimation areas*
[Default: 'CO_Dummy_area_list.txt']
List of areas for which CO estimates are required. (STEP 3, section 1)

Line 5: *Constraint table information*
[Default: 'CO_Dummy_constraining_tables_info.txt']
Information on constraint tables, including the number, size and type (person/household) of these tables, their associated chi-square critical value, and a flag turning the table 'on' or 'off'. (STEP 1, section 4).

Line 6: *CO run-time parameters*
[Default: 'CO_Dummy_control_parameters.txt']
User-specified CO run-time control parameters, including output pathname, output options, measure of fit to minimised and other, more advanced, controls (see STEP 3, section 3).

Line 7: *Random numbers*
[Default: 'CO_random_seednumbers.txt']
A list of 100 random 'seeds' for the random number generation process used by CO (one seed used per 'replication'). If required, can be substituted with an alternative set of 100 random seed numbers (each an integer value), in order to test CO outputs for sensitivity to initially selected household combination (weights).

3. CO_Dummy_control_parameters.txt

Before running CO for a set of user-supplied constraint and survey data, it is imperative that the control paramters contained in this file are amended accordingly. Recommended values/settings for each parameter are outlined below.

3.1 Basic operational controls

Line 1: *Measure of fit to minimise (Measure)*
[Default: RSSZm]
When attempting to fit estimation area constraints, CO will search for the set of survey weights which minimises either TAE or RSSZm. Selecting RSSZm causes CO to focus on minimising proporitional differences between constraints and weighted estimates; selecting TAE causes CO to focus on minimising absolute differences (For full explanation of TAE and RSSZm, see STEP 4, section 3).

Line 2: *Number of replications per area (Reps)*
[Default: 1]
Simulated annealing, which lies at the heart of the CO algorithm, involves the probabilistic acceptance/rejection of a proposed change in household weight (household combination). Therefore runs (replications) of CO starting with differing random seed numbers will produce (slightly) differing results. For most purposes users should ignore the minor stochastic variation between runs, accepting the output from a single run of CO (*Reps*=1) as a 'best' estimate. However, if required, CO is able to replicate the estimation process up to 100 times per estimation area. The outputs from each replication will be saved separately.

3.2 User-controlled outputs

Line 3: *Run name*

[Default: Dummy]

The outputs from a given run of CO (described in STEP 4) are stored in the user-specified subfolder <*Run name*>.

[Maximum length: 40 alpha-numberic characters; place name in quotes if it includes a space]

Line 4: *Weights reporting flag* (*Weights_on*)

[Default: W]

Set flag to **W** to request a full set of household weights for each estimation area (STEP 4, section 5). Set flag to **X** to turn off.

Line 5: *Estimate fit reporting flag (estimate_Fit_on)*

[Default: F]

Set flag to **F** to produce a report summarising the fit, for each area, of the resulting weighted estimates to the user-supplied area constraints. (STEP 4, section 3). Set flag to **X** to turn off.

Line 6: *Estimates reporting flag (Estimates_on)*

[Default: E]

Set flag to **E** to report the full set of constraints and their estimates for each estimation area (STEP 4, section 4). Set flag to **X** to turn-off.

NOTE: The remaing CO output of household 'Combinations' (STEP 4, section 2) is produced automatically and cannot, at present, be switched off.

3.3 Simulated annealing parameters

A brief explanation of the function of these parameters is given below. For a full explanation of simulated annealing and its application in CO, see Williamson *et al*. (1998).

Line 7: *temp0*

[Default: 22]

Initial 'temperature'. (The 'temperature' influences the likelihood of adverse change in weights being accepted, helping to aovid local sub-optimal solutions.) As a rule-of-thumb, set *temp0* to equal the total number of constraint cells contained in the user-supplied constraint tables. (The supplied dummy data contains 22 constraint cells, as highlighted in Table 3.)

Line 8: *limit*

[Default: 220]

Maximum number of changes in household weights (changes in household combination) allowed before temperature decreases. As a rule-of-thumb, setting *limit* to 10 x *temp0* (initial temperature) should ensure good algorithm performance.

Line 9: *decr*

[Default: 0.95]

Rate of temperature reduction. The 'temperature' is reduced, after a series of ‘successful’ household replacements, through multiplication by *decr*. The no. of successful replacements required is determined by *step_size* (sub-section 3.4). The default value of 0.95 is strongly recommended.

3.4 Advanced control options

If run-times are acceptable, users are advised to leave the default values of the advanced control options unchanged.

Rationale behind advanced control options

CO is in iterative algorithm, which tries to ‘optimise’ its estimates by evaluating as many combinations of household weights as possible. Using the advanced control options the user is able to specify exit conditions for the algorithm, relating to both target levels of goodness-of-fit for the resulting weighted estimates, and to the maximum number of evaluations (and hence CPU time) to be used per estimation area. This allows a trade-off to be made between run-time and estimate fit, and allows extra CPU time to be dedicated to the hardest to fit areas. Note that, in the default configuration, CO will expend equal effort fitting each estimation area, making for ‘slow but sure’ progress.

Iteration thresholds

Line 10: *step_size*

[Default: 200000]

Number household combinations (sets of household weights) evaluated between assessments of whether or not user-specified goodness-of-fit thresholds (*AcRSSZ* and *AcOTAE*) have been met. Once the thresholds are met, CO will move on to next estimation area. In setting *step_size* it should be borne in mind that the computational overhead associated with assessing goodness-of-fit means that frequent assessments (small *step_size*) may increase rather than decrease overall run-time.

Line 11: *EvalsThreshold1*

[Default: 200000; should be a multiple of *step_size*]

Defines the minimum number of household combinations (sets of household weights) that CO must evaluate before termination of estimation for current estimation area is first considered. It is strongly recommended that this value is set to *step_size* (line 16).

Line 12: *EvalsThreshold2*

[Default: 200000; MUST be >= *EvalsThreshold1* and <*EvalsThreshold3*; should be a multiple of *step_size*]

Currently has impact on program operation, but value must be set within constraints outlined above to avoid program malfunction.

Line 13: *EvalsThreshold3*

[Default: 4000000; MUST be <= *EvalsThreshold4*; should be a multiple of *step_size*]

If the number of household combinations (sets of household weights) evaluated exceeds *EvalsThreshold3*, the current estimation area is deemed to be ‘hard to fit’. For subsequent evaluations, sampling will be restricted to households already selected represent the area (on the grounds that hard-to-fit areas normally comprise 'rare' households [e.g. student households] which, by this stage, will be present in significant numbers in the already selected household combination. If this 'within combination' sampling phase is not wanted, set *EvalsThreshold3*=*EvalsThreshold4*.

Line 14: *EvalsThreshold4*

[Default: 5000000; MUST be a multiple of *step_size* (line 16)]

The maximum number of household combinations (sets of household weights) to be considered per estimation area. The greater the size of *EvalsThreshold4*, the longer the potential run-time per estimation area. For quick test runs of CO, to check all inputs supplied in correct format, it can be helpful to set *EvalsThreshold4* to some nominal value (e.g. *step_size*).

Target levels of estimate fit

Line 15: *AcRSSZ*

[Default: 0.0 (floating point number)]

Minimum target value of RSSZ^2 for CO to reach before the estimate for a given estimation area is deemed acceptable. If *AcRSSZ* is set too low compute times may be extended unnecessarily, as the estimation algorithm will only move on to the next estimation area once either the required level of fit is met, or the maximum allowed number of iterations for the current estimation area has been exceeded. A rough rule-of-thumb is to set *AcRSSZ* = .01 x number of constraint cells. However, all things being equal, the greater the number of households per estimation area, the higher the value of *AcRSSZ* should be set. Experimentation using a few pilot estimation areas can help to yield a better problem-specific value. (RSSZ^2 is explained in STEP 4, section 3)

Line 16: *AcOTAE*

[Default: 0 (integer)]

Minimum target value of OTAE for CO to reach before the estimate for a given estimation area is deemed acceptable. If *AcOTAE* is set too low compute times may be extended unnecessarily, as the estimation algorithm will only move on to the next estimation area once either the required level of fit is met, or the maximum allowed number of iterations for the current estimation area has been exceeded. A rough rule-of-thumb is to set the value of *AcOTAE* to 0.5 x average households per estimation area. However, the greater the number of constraint cells, and the greater the number of households per estimation area, the higher the value of AcOTAE should be set. Experimentation with a few pilot estimation areas can help to yield a better problem-specific value (OTAE is explained in STEP 4, section 3).

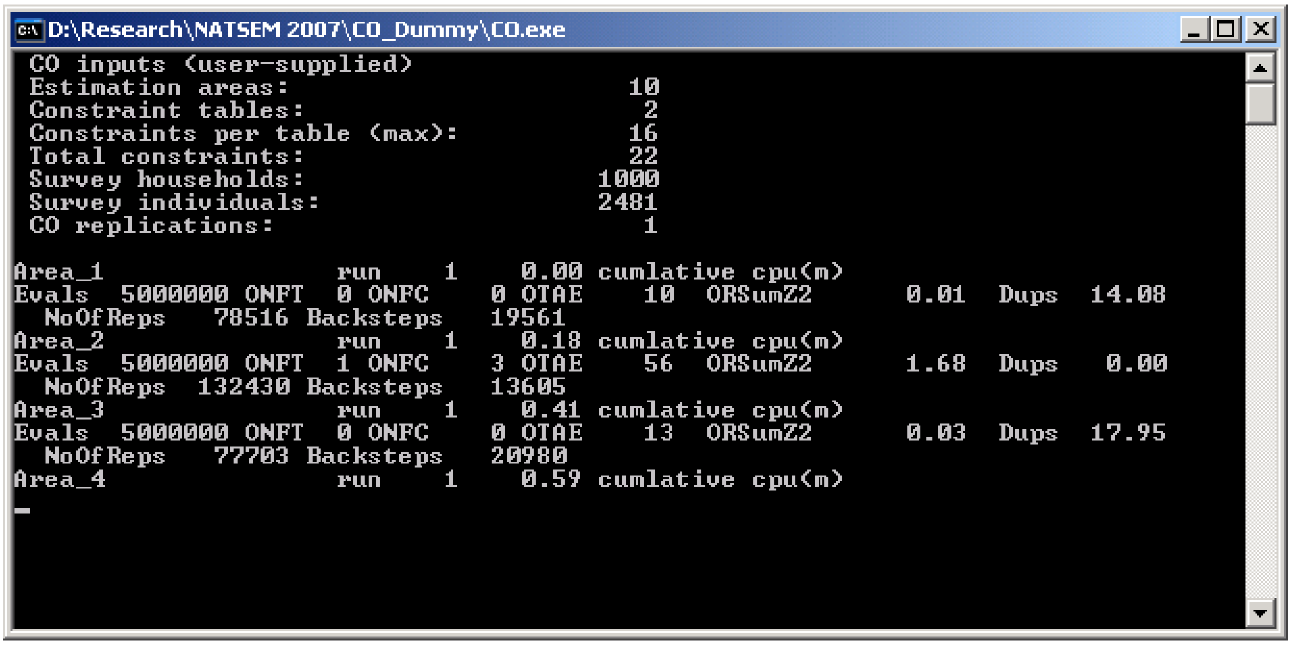
STEP 4: Running CO and understanding the outputs

Having prepared the input files and user-controlled run-time parameters as outlined in STEPS 1-3, all that is required to run CO is to (i) locate the file *CO.exe* via Windows Explorer (located in the root folder *CO*; (ii) double-click on the file icon. A run-time screen will then appear, reporting on the progress of CO (section 1). Once execution has successfully completed, the resulting CO outputs will be stored in four folders: *CO/Combinations* (Section 2); *CO/Estimate_Fit* (section 3); *CO/Estimates* (section 4); *CO/Weights* (section 5). To cancel execution at any time, press the key combination *Ctrl* and *C* .

If CO fails to execute, the resulting error message should give a good clue as to the cause. Most commonly, 'file not found' indicates that one or more of the files listed in *CO_filelist.txt* does not exist, possibly reflecting a mismatch caused by a simple typographical error when entering file names. The second most common problem relates to input files that are either incorrectly formatted or that hold mutually inconsistent information.

1. Monitoring CO progress

Once CO has been started, a DOS window appears, in which run-time progress is reported (Figure 2).



```
 CO inputs (user-supplied)
 Estimation areas:                       10
 Constraint tables:                       2
 Constraints per table (max):            16
 Total constraints:                      22
 Survey households:                    1000
 Survey individuals:                   2481
 CO replications:                         1

Area_1               run    1    0.00 cumlative cpu(m)
Evals  5000000 ONFT  0 ONFC    0 OTAE    10  ORSumZ2      0.01  Dups  14.08
  NoOfReps   78516 Backsteps   19561
Area_2               run    1    0.18 cumlative cpu(m)
Evals  5000000 ONFT  1 ONFC    3 OTAE    56  ORSumZ2      1.68  Dups   0.00
  NoOfReps  132430 Backsteps   13605
Area_3               run    1    0.41 cumlative cpu(m)
Evals  5000000 ONFT  0 ONFC    0 OTAE    13  ORSumZ2      0.03  Dups  17.95
  NoOfReps   77703 Backsteps   20980
Area_4               run    1    0.59 cumlative cpu(m)
```

Figure 2 DOS Window reporting run-time progress

The 2nd-7th lines of the progress report describe the input data as processed by CO:

Estimation areas	Number of areas for which user has supplied constraint data
Constraint tables	Number of user-supplied constraint 'tables' for each estimation area
Constraints per table (max.)	Maximum number of (interior) cells in any constraint table
Total constraints	Total (interior) cells across all user-supplied constraint tables

Survey households	Number of households in user-supplied survey data
Survey individuals	Number of persons in user-supplied survey data
CO replications	Number of stochastic CO estimates to be produced for each estimation area

Subsequent lines report progress on estimating the populations for the user-requested estimation areas, with two lines of output per estimation area. The first identifies the area being estimated, the 'replication' number, and the elapsed processing time (in minutes). The second reports a range of post-estimation statistics:

Evals	No. of household weighting combinations evaluated
ONFT	Overall Non-Fitting Tables
ONFC	Overall Non-Fitting Cells
OTAE	Overall Total Absolute Error
ORSumZ2	Overall Relative Sum of Squared Z-scores
Dups	% of households that appear more than once in selected household combination
NoOfReps	No. of household weight changes allowed that led to improved fit
Backsteps	No. of household weight changes allowed that led to decreased fit

CO may be requested to save all of the above post-estimation statistics, with the exception of *Backsteps*, in a run-specific output file. Full details, including an expanded explanation of the above post-estimation measures, is provided in section 3.

2. COMBINATIONS

The combinations of survey households selected as best representing the population of each estimation area are saved in the in sub-folder *<RunName>* of *CO/Combinations*, in files named using the format *Comb_<AreaName>_v<replication number>.txt*. *RunName* is taken from *CO_Dummy_control_parameters.txt* and the *AreaName* from the user-supplied list of input areas (e.g. *CO_Dummy_area_list.txt*). The replication number is added by CO automatically.

The format of each results file is as follows:

Line 1: No. of households in area
Line 2: blank
Line 3 onwards: list of survey households (household IDs) chosen as 'best representing' the estimation area.

This default output format is adopted as a space-saving measure (only survey households with non-zero 'weights' are listed). A full set of survey weights may be requested via *CO_Dummy_control_parameters.txt* (STEP 3, section 3). The output from CO produced using the supplied dummy data is shown in Example 7 for one estimation area.

```
  71

 377    21   979   246   664   485   829   392    61   614
 809   782   828    14    48     2   392   225   195   901
 602   492   851   826   516   144   137    48    27    37
 154   631   393   570   891   829    97   770   727   302
 706   731   334   985   703   653    51   995   452    91
 993   977   494   707   390    10   749   643   442   377
 417    27   431   412   818   305   249    50   535   440
  47
```

Example 7 *Comb_Area_1_v1.txt*

3. ESTIMATE_FIT

Upon completion CO records summary measures of fit at cellular, tabular and overall levels for each estimation area in the sub-folder *<Run Name>* of *CO/Estimate_Fit* in a file named *<RunName>_Garea.txt*. *RunName* is taken from *CO_Dummy_control_parameters.txt.*

The format of the file is one row of information per replication, per estimation area. In most cases users will require only one set of estimates (one replication), so there will only be one row per estimation area (Example 8). Where multiple replications are produced, the results for each replication are listed sequentially within estimation areas (Example 9).

								OR							No
							OTAE	Sum	TAE	TAE	RSS	RSS		Dups	Of
Area	time	evals	noofrep	NFT	NFC	OTAE	/HH	Z2	_1	_2	Z_1	Z_2	temp	_%	H
Area_1	10.8	5000000	78516	0	0	10	0.14	0.01	6	4	0.0	0.0	0.0024	14.1	71
Area_2	13.7	5000000	132430	1	3	56	1.44	1.68	16	40	0.0	1.7	0.0001	0.0	39
Area_3	10.8	5000000	77703	0	0	13	0.17	0.03	7	6	0.0	0.0	0.0020	17.9	78
Area_4	11.0	5000000	79866	0	0	23	0.35	0.05	13	10	0.0	0.1	0.0004	31.8	66
Area_5	12.3	5000000	93944	0	0	39	0.48	0.15	21	18	0.0	0.2	0.0001	53.1	81
Area_6	11.3	5000000	84678	0	0	24	0.29	0.05	14	10	0.0	0.0	0.0009	28.0	82
Area_7	12.2	5000000	104467	1	6	68	1.45	2.06	24	44	0.4	1.7	0.0001	14.9	47
Area_8	12.1	5000000	97924	0	0	35	0.51	0.14	19	16	0.0	0.1	0.0001	48.5	68
Area_9	13.1	5000000	120747	1	3	51	1.24	1.24	17	34	0.0	1.2	0.0001	17.1	41
Area_10	11.3	5000000	84008	0	0	45	0.55	0.19	25	20	0.0	0.2	0.0005	34.1	82

Example 8 Contents of *CO/Estimate_Fit/Dummy/Dummy_Garea.txt (1 replication)*

								OR							No
							OTAE	Sum	TAE	TAE	RSS	RSS		Dups	Of
Area	time	evals	noofrep	NFT	NFC	OTAE	/HH	Z2	_1	_2	Z_1	Z_2	temp	_%	H
Area_1	11.4	5000000	78516	0	0	10	0.14	0.01	6	4	0.0	0.0	0.0024	14.1	71
Area_1	11.1	5000000	75409	0	0	10	0.14	0.01	6	4	0.0	0.0	0.0035	14.1	71
Area_2	14.1	5000000	132430	1	3	56	1.44	1.68	16	40	0.0	1.7	0.0001	0.0	39
Area_2	14.0	5000000	128261	1	3	56	1.44	1.68	16	40	0.0	1.7	0.0001	5.1	39

Example 9 Extract from *CO/Estimate_Fit/Dummy/Dummy_Garea.txt* (TWO replications)

The information supplied on estimate fit is explained briefly below. For full details on the measures used, see Voas and Williamson (2001). As a rule-of-thumb, synthetic microdata for estimation areas with an OTAE/HH > 1.0, should be treated with some caution.

Area — Name of estimation area

Time — CPU seconds taken to produce estimate for areea

Evals — Number of potential household replacements (weight changes) evaluated

NoOfRep — Number of actual household replacements (weight changes) made

NFT — Number of 'non-fitting' tables
[In a 'non-fitting' table the sum of Z-scores^2 across all cells in the table exceeds the table-specific chi-square critical value]
(RSSZ_*x* > 1)

NFC — Number of 'non-fitting' cells
[A 'non-fitting' table cell has an associated Z-score >1.96 or < –1.96]

$Z = [(Oi/ \Sigma E_i) - e_i] / [e_i(1-e_i) / \Sigma E_i]^{0.5}$

where E_i = expected (constraint) cell value; O_i = observed (estimated) cell value; n = no. of cells in constraint table x; Σ = sum over all i, from 1 to n; $o_i = O_i / \Sigma O_i$; $e_i = E_i / \Sigma E_i$

OTAE — Overall Total Absolute Error
[Sum of TAE_*x* (see below) across all constraint tables]

OTAE/HH — Overall Total Absolute Error per household
[The Overall Total Absolute Error per household in the estimation area] (=OTAE/NoOfH)

ORSSZ — Overall Relative Sum of Squared Z-scores
[Sum of RSSZ_*x* (see below) across all constraint tables]

TAE_*x* — Total Absolute Error associated with constraint table *x*
[Tables numbered in order listed in constraints file (STEP 1)]

$TAE\_x = \Sigma \mid (O_i - E_i) \mid$

where E_i = expected (constraint) cell value; O_i = observed (estimated) cell value

RSSZ_*x* — Relative Sum of Square Z-scores associated with constraint table *x*
[Tables numbered in order listed in constraints file (STEP 1)]

$$RSSZ\_x = \left(\Sigma \{ [(Oi/ \Sigma E_i) - e_i] / [e_i(1-e_i) / \Sigma E_i]^{0.5} \}^2 \right) / c$$

where E_i = expected (constraint) cell value; O_i = observed (estimated) cell value; n = no. of cells in constraint table x; Σ = sum over all i, from 1 to n; $o_i = O_i/ \Sigma O_i$; $e_i = E_i/ \Sigma E_i$; c = chi-square critical-value for a table with p=0.05 and $d.f. = n$

Temp Final value of ‘temperature’; a simulated annealing control parameter (see Williamson *et al.*, 1998 for details)

Dups_% - % of households present in final weighted combination more than once

NoOfH - Number of households in estimation area

4. ESTIMATES

For users interested in examining estimate fit to specific constraint table cells, CO may also be requested to create a set of files that report each area constraint side-by-side with the CO estimate for that constraint. These files are stored in the sub-folder *CO/Estimates/<RunName>*, in files named using the format *<AreaName>_est_v<replication number>.txt* . *RunName* is taken from *CO_Dummy_control_parameters.txt* and the *AreaName* from the user-supplied list of input areas (e.g. *CO_Dummy_area_list.txt*). The replication number is added by CO automatically.

The format of these files is:
File header: Area name, no. of households in area, no. of evaluations, replication number

Then, for each constraint table in turn:
Line 1: table name, no. of constraints in table
Line 2: user-supplied constraints
Line 3: *blank*
Line 4: CO estimates of these constrained values
Line 5: *blank*

```
Area_1       71    5000000        1

 P1        6
  209   11   49   40   57   21   31

  203   11   48   39   55   20   30

 H1        16
   71    5    2    3    5    3    6    8    4    7    1    8    8    0    1    3    7

   71    4    2    3    5    3    6    8    4    6    1    8    9    0    1    3    8
```

Example 10 *Area_1_est_v1.txt*

5. WEIGHTS (optional)

If requested by the user via the CO control parameter file (STEP 3, section 3), CO will report, for each estimation area, the full set of estimation weights, including those households with a weight of 0. For ease of use, the weights for all estimation areas are combined within a single comma-separated file (Example 11). The resulting weights file has a name of the form *<**RunName**>_wgts_v<**replication number**>.csv* (e.g. *Dummy_wgts_v1.csv*), and will be found in the sub-folder *CO/Weights/<**RunName**>*. If more than one replication of the estimation process is requested, separate weights files are produced for each replication (i.e. *Dummy_wgts_v1.csv*; *Dummy_wgts_v2.csv* etc.). *RunName* is taken from *CO_Dummy_control_parameters.txt* and the *AreaName* from the user-supplied list of input areas (e.g. *CO_Dummy_area_list.txt*). The replication number is added by CO automatically.

```
HH_ID, wt_Area_1, wt_Area_2, wt_Area_3,    ...   , wt_Area_9, wt_Area_10
     1,          0,          0,          0,    ...   ,          0,          0
     2,          1,          0,          0,    ...   ,          0,          0
     3,          0,          0,          0,    ...   ,          0,          0
     4,          0,          1,          0,    ...   ,          0,          1
     .
     .
   998,          0,          0,          0,    ...   ,          0,          0
   999,          0,          0,          0,    ...   ,          0,          0
  1000,          0,          0,          0,    ...   ,          0,          0
```

Example 11 Extract from *Dummy_wgts_v1.csv*

STEP 5 (optional): Adding weights to original survey data

For most users the key output from CO will be a set of area-specific household weights (STEP 4, section 5) which, when applied to a survey file, will reweight the survey to fit, as closely as possible, any user-supplied area constraints. When adding these weights to their original survey data the user will need to use their own favoured survey analysis software. By way of example, Box 3 outlines the SPSS syntax code required to (i) import a set of CO-produced weights into SPSS; (ii) add the weights to an existing survey file; (iii) turn 'on' the weights for a given estimation area, for use during subsequent survey analysis; (iv) reproduce the CO estimates (*CO/Dummy/Estimates/Area_1_est_v1.txt*) of the table counts used as estimation constraints (*CO_dummy_estimation_constraints.txt*), providing independent verification of the results reported by CO. (Compare the figures in Examples 12 and 13, produced via SPSS, with those shown in Example 10, produced directly by CO.)

Grouped Age * Sex Crosstabulation

Count

		Sex		Total
		Male	Female	
Grouped Age	Child	**11**	**48**	59
	Adult	**39**	**55**	94
	Pensioner	**20**	**30**	50
Total		70	133	**203**

Example 12 SPSS weighted PERSON-level tabulation for Dummy Area_1

Cars in household * Persons in household Crosstabulation

Count

		Persons in household				Total
		1	2	3	4	
Cars in household	0	**4**	**2**	**3**	**5**	14
	1	**3**	**6**	**8**	**4**	21
	2	**6**	**1**	**8**	**9**	24
	3	**0**	**1**	**3**	**8**	12
Total		13	10	22	26	**71**

Example 13 SPSS weighted HOUSEHOLD-level tabulation for Dummy Area_1

Box 3 SPSS code for adding CO weights to original survey and reproducing CO estimates

NOTE: Syntax below will work ONLY if each comment line (starting with an asterisk) is followed by a blank line

```
*(1) Create SPSS version of CO weights file
*(match file command doesn't work well with temporary files)

*[Note: filenames based on running syntax from folder CO_Dummy/Data/Original Survey]

*(a) Import csv data

GET DATA  /TYPE = TXT
 /FILE = '..\..\Weights\Dummy\Dummy_wgts_v1.csv'
 /DELCASE = LINE
 /DELIMITERS = ","
 /ARRANGEMENT = DELIMITED
 /FIRSTCASE = 2
 /IMPORTCASE = ALL
 /VARIABLES =
 HH_ID F12.0
 wt_Area_1 F10.0
 wt_Area_2 F10.0
 wt_Area_3 F10.0
 wt_Area_4 F10.0
 wt_Area_5 F10.0
 wt_Area_6 F10.0
 wt_Area_7 F10.0
 wt_Area_8 F10.0
 wt_Area_9 F10.0
 wt_Area_10 F10.0
 .
EXECUTE.

DATASET NAME DataSet1.

*(b) Sort cases by HH_ID to allow match with original survey data

SORT CASES BY HH_ID (A)

*(c) Save file

*SAVE OUTFILE='..\..\Weights\Dummy_wgts_v1.sav'
*/COMPRESSED.


*(2) Open original survey data and make active SPSS dataset

GET FILE='CO_Dummy_survey.sav'.
DATASET NAME DataSet2.

*(3) Sort original survey by (i) HH_ID; (ii) persons within household
*      (Otherwise matching with CO weights will fail)

SORT CASES BY HH_ID (A) Person (A) .

*(4) Merge weights with survey data

DATASET ACTIVATE DataSet2.

MATCH FILES /FILE=*
 /TABLE='DataSet1'
 /BY HH_ID.
EXECUTE.

DATASET CLOSE DataSet1.
```

```
*(5) Reproduce constraint tables using CO weights

*(a) Recode variables to match constraint table categories

*(i) Age

RECODE  Age  (0 thru 15=1)  (16 thru 64=2)  (65 thru Highest=3)  INTO  Age_gp .
VARIABLE LABELS Age_gp 'Grouped Age'.
VALUE LABELS Age_gp 1 'Child' 2 'Adult' 3 'Pensioner'.
EXECUTE .

*(ii) Sex [already in required format]
*(iii) Cars [already in required format]
*(iv) HH_Size [already in required format]


*(b) Reweight survey using chosen set of CO area weights

WEIGHT BY wt_Area_1.

*(c) Reproduce constraint tabulations

*(i) Table P1 (Person-level)

CROSSTABS
  /TABLES=Age_gp BY Sex
  /FORMAT= AVALUE TABLES
  /CELLS= COUNT
  /COUNT ROUND CELL .

*(ii) Table H1 (Household-level)
*[FILTER BY FIRST PERSON IN HOUSEHOLD TO PRODUCE HOUSEHOLD COUNTS]

COMPUTE Household_Filter=(Person=1).
VARIABLE LABEL Household_Filter 'Household Filter'.
VALUE LABELS Household_Filter  0 'Not Selected' 1 'Selected'.
FORMAT Household_Filter (f1.0).
FILTER BY Household_Filter.
EXECUTE .

CROSSTABS
  /TABLES=Cars  BY HH_Size
  /FORMAT= AVALUE TABLES
  /CELLS= COUNT
  /COUNT ROUND CELL .

FILTER OFF.
USE ALL.
```

References

Voas D and Williamson P (2001) ‘Evaluating goodness-of-fit measures for synthetic microdata’, *Geographical and Environmental Modelling*, 5, 177-200.

Williamson P, Birkin M and Rees P (1998) ‘The estimation of population microdata using data from small area statistics and samples of anonymised records’, 30, *Environment and Planning A*, 785-816.