Technical Paper Series

PROVIDE PROJECT

The Provincial Decision-making Enabling Project

Technical Paper 2003: 1

Creating a 1995 IES Database in STATA

Elsenburg
September 2003

PROVIDE PROJECT

The Provincial Decision-making Enabling Project

Overview

The Provincial Decision-Making Enabling (PROVIDE) Project aims to facilitate policy design by supplying policymakers with provincial and national level quantitative policy information. The project entails the development of a series of databases (in the format of Social Accounting Matrices) for use in Computable General Equilibrium models.

The National and Provincial Departments of Agriculture are the stakeholders and funders of the PROVIDE Project. The research team is located at Elsenburg in the Western Cape.

PROVIDE Research Team

Project Leader:	Cecilia Punt
Senior Researchers:	Kalie Pauw
	Esther Mohube
Junior Researchers:	Benedict Gilimani
	Lillian Rantho
	Rosemary Leaver
Technical Expert:	Scott McDonald
Associate Researchers:	Lindsay Chant
	Christine Valente

PROVIDE Contact Details

Private Bag X1
Elsenburg, 7607
South Africa

ceciliap@elsenburg.com
+27-21-8085191
+27-21-8085210

For the original project proposal and a more detailed description of the project, please visit www.elsenburg.com/provide

Creating a 1995 IES Database in STATA[1]

Abstract

This technical paper serves as an outline of the procedures followed to create household level income and expenditure data files in the statistical software package STATA. The datasets are used for the development of various Social Accounting Matrices for South Africa. A national data set, four regional data sets and nine provincial data sets were created; all based on the Income and Expenditure Survey (IES) of 1995. The STATA programming files (do-files) used are attached as a technical appendix to this paper.

[1] The main author of this paper is Benedict Gilimani, Junior Researcher of the PROVIDE Project.

Table of contents

List of figures

List of tables

1. Introduction

This technical paper serves as an outline of the procedures followed to create household level income and expenditure data files in STATA. A national data set, four regional data sets and nine provincial data sets were created; all based on the 1995 Income and Expenditure Survey (SSA, 1997). STATA's statistical analysis software was selected due to its usefulness in analysing large sets of data such as the 1995 Income and Expenditure Survey (IES 1995) dataset.

This paper shows how do-files were set up and used to create datasets in STATA format. The datasets had to be organised in such a way that data needed for the development of various Social Accounting Matrices (SAM) for South Africa[2] could easily be extracted. Section 2 of this paper provides an overview of the datasets that were developed. Reference will be made to the IES 1995 survey questionnaire on which these datasets are based. Section 3 reviews the procedures followed to create the do-files in STATA. Section 4 gives a brief overview of the way in which incomplete or inconsistent data in the survey were treated. The appendix (section 6) includes all the STATA do-files and a brief discussion of how selected STATA commands work.

2. The 1995 Income and Expenditure Survey questionnaire and data

The IES data were collected in 1995 by Statistics South Africa (formerly the Central Statistical Service). The questionnaire gives particulars of income from different sources, expenditure on different commodities, savings and interhousehold transfers. All the questions were coded and these codes were used to map the information to the income and expenditure accounts for households included in the SAM. All the data pertaining to expenditure on commodities were mapped to 134 expenditure categories based on the Standard Industrial Classification (SIC) (CSS, 1993). The 134 expenditure categories were subsequently mapped to 96 expenditure categories. The matching of expenditures to 134 categories was done as an intermediate step to check that the STATA results were consistent with results that were previously derived in Excel for 134 categories. The first 95 of the final 96 expenditure categories correspond to those used by Statistics South Africa in the compilation of their national Supply and Use Tables (SSA, 2001). A 96th category was created for expenditure on domestic labour. Payments to domestic labour are not included in the SAM as a payment directly from households to factors, but as household consumption of the commodity "domestic services", which is supplied by the activity "domestic services". The domestic

[2] The second phase of the PROVIDE Project involves the development of four regional SAMs, a S.A. SAM as well as an integrated National SAM. This paper forms part of the initial data collection phase.

services activity then pays the relevant factor. This reorganisation of data ensures that all payments to factors are recorded in the same SAM submatrix.

In the IES 1995 questionnaire some of the figures were reported as monthly figures and others as annual figures. The figures were all changed to annual figures because SAMs require annual data. STATA allows up to eight digits in the coding system of variables. All variables used in STATA were assigned a different code and labelled to give the description of each variable. For example, the STATA code for the race variable is *race* and its label description is *Population group* and each population group was given a code and those codes were also labelled, for example code 1 was labelled "African", 2 "Coloured", 3 "Indian" and 4 "White". The settlement variable was also recoded. In the original dataset there were 16 categories for the settlement variable. The settlement variable code in the original data set was a two-digit code that contained fairly detailed information on the area (urban, semi-urban and rural). These categories were reduced to either urban (code 1) or rural (code 2).

Dictionary files are used to read the data into STATA from the original fixed-width text data files version. The IES 1995 dictionary files are listed by province. Each variable in the IES 1995 questionnaire is coded. Variables in the dictionary files are stored in a format relating to reference in the IES 1995 questionnaire, for example b07f001 refers to block 07 field 001 of the questionnaire.

The national data set contains data on all provinces. Different codes were given for all the provinces i.e. 1 for Western Cape, 2 for Eastern Cape, 3 for Northern Cape, 4 for Free State Province, 5 for Kwazulu-Natal, 6 for North West, 7 for Gauteng, 8 for Mpumalanga and 9 for Limpompo. In order to construct regional SAMs four regional datasets were constructed as follows: Western Cape and Northern Cape combined form the West Coast region. Kwazulu-Natal and Eastern Cape form the East Coast region. Free State, North West and Gauteng form the Centre region, and Mpumalanga and Limpompo form the Border region. Data on these regions were stored as separate data files called wcoast.dta, ecoast.dta, centre.dta and border.dta. The main aim of creating these data files is to organise information on household income and expenditure in a way that it can be included in the regional and national SAM's.

All households are allocated to either one of the four main population groups in South Africa, namely Africans, Coloureds, Indians and Whites. No provision was made for "other" race groups. The race variable is coded as follows: 1 for Africans, 2 for Coloureds, 3 for Indians and 4 for Whites.

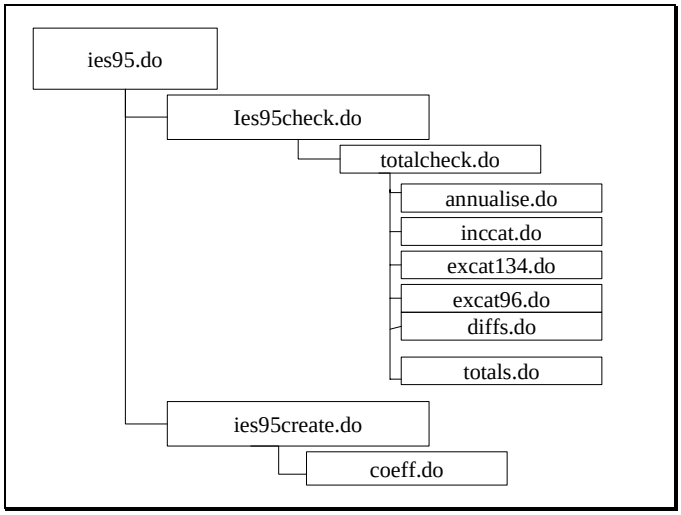
3. Do-files to create Income and Expenditure Survey datasets

A do-file is a file containing STATA commands. A do-file keeps a record of all commands needed to perform a certain task in STATA. The need to repeat all commands manually when an error arises is thus avoided. Do-files are very important, especially because an "undo" command does not form part of the STATA programming language. Various do-files can be combined to form a STATA 'programme' that can be executed at any time. The do-files discussed in this document contain all the commands that were used to create national, regional and provincial data sets.

The "Master do-file" is called ies95.do. This do-file contains two "sub-do-files".

Figure 1 shows the structure of the master do-file, its sub-do-files and various other do-files also contained in the programme structure. By executing ies95.do all other do-files containing various commands are called up and executed systematically, thus forming a kind of executable computer programme in STATA code. In order to run the master do-file (ies95.do), all the do-files, dictionary files and fixed-width data files need to be saved in a single folder. STATA can now be loaded and the path (i.e. directory name in which the above-mentioned files are stored) needs to be set to point at the file in which the various files mentioned above are saved, e.g. "cd c:/data/ies95". Now type, "do ies95". The programme will execute and create various datasets that will be stored in the same folder. Each do-file is discussed in more detail in sections 3.1 and 3.2.

Figure 1: Structure of ies95.do



3.1. Ies95check.do

Ies95check.do runs a series of do-files that checks data for consistency. Once the data is read in from the ASCII files a do-file called *totalcheck.do* is called up for each provincial dataset. This do-file checks the various totals and sub-totals as reported in the original IES 1995 dataset for consistency. Any inconsistencies are pointed out (see discussion in section 4). *Annualise.do* converts monthly data to annual data. All figures used hereafter are thus annual. *Excat134.do* generates the 134 expenditure categories based on the SIC codes discussed in section 2. *Exact96.do* generates 96 expenditure categories from the original 134 expenditure categories. *Inccat.do* creates income categories by grouping income from various income sources. These include: inclab (household income from labour), incgos (household income from gross operating surplus), inctrans (inter-household transfers), inccorp (income from corporation), incgov (transfers from central government), incother (other income (usually netted out)) and inctot (reported income total in IES 1995). *Totals.do* matches IES 1995 expenditure category totals with those reported on the summary page of the IES 1995 questionnaire. This only applies to those variables actually used in the analysis. E.g., the monthly summary variables b89f001-b89f024 were not annualised since they are not used for the analysis. And *diffs.do* checks data for consistency and reports any differences in reported and calculated totals. The do-files are included in the appendix.

3.2. Ies95create.do

Ies95create.do creates the regional and the provincial data sets after all checks have been performed and inconsistent observations have been dropped (see section 4). *Ies95create.do* also runs the do-file called *Coeffs.do* in order to derive the proportional spending of households per one Rand spent.

4. Treatment of inconsistent/incomplete data

When *ies95check.do* was run certain reporting errors were found. This was especially the case in the housing expenditure section. Some households reported only figures for total housing expenditure without reporting the expenditures for each sub-category that make up total housing expenditure. There was also a problem with the way in which VAT was reported in the housing section. The transport section also had a few problems that are discussed below. Problematic observations were either dropped or “dealt with” in an appropriate manner. This was necessary because the incomplete information caused a discrepancy between reported and calculated household expenditure totals.

4.1. Cost of housing totals (b07f100 and b07f200)

The two totals (b07f100 and b07f200) in the cost of housing section were problematic for quite a number of observations. It was discovered that some households only reported an amount for total expenditure on housing without providing a breakdown of how this total was arrived at. Total cost of housing is made up of various sub-totals, including rent, bond payments and services such as electricity, sanitation etc. These sub-totals are needed in the SAM as they fall under various separate expenditure categories. The reported total amounts therefore do not contribute sufficient information for use in the SAM. Thus, whenever a household reported total expenditure without providing the proper breakdown, the entire observation was deleted.

Sometimes dropping variables may be problematic as it may affect the randomness of the sample. In order to determine whether this is a real problem in this case, a profile of the dropped variables is given, based on settlement, race and province. As is shown in Tables 1 to 3 there was no real pattern or common characteristics among the dropped variables. It is thus assumed that the randomness of the sample is unaffected by the dropping of observations.

There were only three observations with problems relating to variable b07f100. One rural African household in the Eastern Cape was dropped, while two rural African households in the Limpopo province were dropped. This had virtually no impact on the national dataset. There were however more problems relating to variable b07f200. Table 1 shows the number of households in each category (by race, settlement and province) that have been dropped due to problems with variable b07f200. The percentages in brackets refer to the number of households dropped as a percentage of the total number of households in the sample that fall under that category.

Most provinces had between zero and six problematic observations that had to be dropped. Gauteng had slightly more (ten), all of which were urban households. The most problematic one was KwaZulu-Natal, with 19 observations (or 0.37% of the sample) dropped. These were mainly White urban households.

In total 48 observations were dropped, 19 of which were from KwaZulu-Natal. These 48 households make up 0.16% of the entire IES 1995 sample. Only 10 of these households were from rural areas and 38 of them were from urban areas. Most of the households were African, although in relative terms more White households (mainly urban) were dropped. None of the percentages above seem to indicate that a very large number of households from a specific households group (apart maybe for White urban households in KwaZulu-Natal) were dropped.

Table 1: The number of dropped observations with regard to b07f200 (cost of housing)

	Race	Urban	Rural	Total
Western Cape	African			
	Coloured			
	Indian			
	White			
	Total	*0*	*0*	*0*
Eastern Cape	African	5 (0.35%)		*5 (0.12%)*
	Coloured			
	Indian			
	White			
	Total	*5 (0.20%)*		*5 (0.09%)*
Northern Cape	African			
	Coloured			
	Indian			
	White			
	Total			
Free State	African	4 (0.32%)		*4 (0.17%)*
	Coloured			
	Indian			
	White	1 (0.16%)		*1 (0.15%)*
	Total	*5 (0.24%)*		*5 (0.15%)*
KwaZulu-Natal	African	5 (0.39%)	2 (0.08%)	*7 (0.19%)*
	Coloured			
	Indian			
	White	11 (2.12%)	1 (0.64%)	*12 (1.78%)*
	Total	*16 (0.63%)*	*3 (0.11%)*	*19 (0.37%)*
North-West	African			
	Coloured			
	Indian			
	White	1 (0.33%)		*1 (0.29%)*
	Total	*1 (0.08%)*		*1 (0.04%)*
Gauteng	African	2 (0.12%)		*2 (0.11%)*
	Coloured	1 (0.41%)		*1 (0.38%)*
	Indian			
	White	7 (0.73%)		*7 (0.62%)*
	Total	*10 (0.32%)*		*10 (0.29%)*
Mpumalanga	African		2 (0.15%)	*2 (0.10%)*
	Coloured			
	Indian			
	White			
	Total		*2 (0.14%)*	*2 (0.08%)*
Northern Province	African		5 (0.26%)	*5 (0.20%)*
	Coloured			
	Indian			
	White	1 (0.74%)		*1 (0.52%)*
	Total	*1 (0.15%)*	*5 (0.25%)*	*6 (0.22%)*
Total	African	16 (0.19%)	9 (0.08%)	*25 (0.13%)*
	Coloured	1 (0.03%)		*1 (0.03%)*
	Indian			
	White	21 (0.45%)	1 (0.12%)	*22 (0.40%)*
	Total	*38 (0.22%)*	*10 (0.08%)*	*48 (0.16%)*

4.2. Public and hired transport (b56f100)

A similar problem of inconsistency occurred on the section covering public and hired transport. For some observations the items in this section did not add up to the total (variable b56f100). This was due to the treatment of furniture removal, which was noted in the questionnaire as a monthly figure, but was actually included in the total as an annual figure. This was discovered through summation checks that were imposed on all data. Some of the reported totals therefore did not match the calculated totals when adding up the items in this section. The problem was not too severe, with reported totals for 38 households not matching calculated totals. These households were dropped. Table 2 shows the number and percentages of households that were dropped in each province by race and settlement.

Table 2: The number of dropped observations with regard to b56f100 (public and hired transport)

	Race	Urban	Rural	Total
Western Cape	African Coloured Indian White	1 (0.08%)		1 (0.06%)
	Total	*1 (0.04%)*		*1 (0.03%)*
Eastern Cape	African Coloured Indian White	3 (0.21%)	2 (0.08%)	5 (0.12%)
	Total	*3 (0.12%)*	*2 (0.07%)*	*5 (0.09%)*
Northern Cape	African Coloured Indian White			
	Total	*0*	*0*	*0*
Free State	African Coloured Indian White	2 (0.31%)		2 (0.30%)
	Total	*2 (0.10%)*		*2 (0.06%)*
KwaZulu-Natal	African Coloured Indian White	1 (0.19%)	2 (0.08%)	2 (0.05%) 1 (0.15%)
	Total	*1 (0.04%)*	*2 (0.07%)*	*3 (0.06%)*
North-West	African Coloured Indian White	1 (0.12%)		1 (0.05%)
	Total	*1 (0.08%)*		*1 (0.04%)*
Gauteng	African Coloured Indian White	4 (0.24%) 3 (1.23%) 5 (0.52%)	4 (1.94%) 1 (0.57%)	8 (0.42%) 3 (1.15%) 6 (0.53%)
	Total	*12 (0.39%)*	*5 (1.25%)*	*17 (0.49%)*
Mpumalanga	African		2 (0.15%)	2 (0.10%)

	Race	Urban	Rural	Total
	Coloured			
	Indian			
	White	1 (0.23%)	1 (2.38%)	2 (0.42%)
	Total	*1 (0.09%)*	*3 (0.22%)*	*4 (0.16%)*
Northern Province	African	2 (0.38%)	3 (0.16%)	5 (0.20%)
	Coloured			
	Indian			
	White			
	Total	*2 (0.29%)*	*3 (0.15%)*	*5 (0.19%)*
Total	African	10 (0.12%)	13(0.12%)	23 (0.12%)
	Coloured	4 (0.14%)		4 (0.11%)
	Indian			
	White	9 (0.19%)	2 (0.25%)	11 (0.20%)
	Total	*23 (0.14%)*	*15(0.12%)*	*38 (0.13%)*

4.3. Value added Tax (VAT) in the cost of housing section

Section 1.7 of the questionnaire deals with payments for housing services. Initially the VAT element under the housing section (b07f017 in the questionnaire) was included in the variable hhindtax (household indirect taxes). However, it was subsequently realized that this was an error: all reported expenditures should be inclusive of VAT in the SAM. This component should therefore be added (*pro-rata*) to water (b07f011), electricity (b07f012), gas (b07f014), sanitary services (b07f015) and refuse (b07f016) (it is assumed that assessment rates and taxes are not taxable). For some inexplicable reason some households reported an expense for VAT but no other expenses under section 1.7. These expenses were added to the variable hhindtax. No observations were dropped.

4.4. Summary of the dropped households

In total there were 89 households dropped from the original 29 595 households. Only three households were dropped as a result of reporting errors related to the sub-total on housing (b07f100). The overall total for housing (b07f200) had the most inconsistencies and 48 observations were dropped. 38 households were dropped from the sample due to errors in public and hired transport (b56f100). Table 3 below shows the percentages of households dropped by race and settlement. In total only 0.3% of all observations were dropped.

Table 3: The number of dropped observations from the national data set

Summary – percentage of households dropped by race and settlement			
	Urban	Rural	Total
African	0.35%	0.20%	0.26%
Coloured	0.17%	0.00%	0.13%
Indian	0.00%	0.00%	0.00%
White	0.64%	0.37%	0.60%
Total	*0.38%*	*0.20%*	*0.30%*

5. References

CSS (1993). SIC Standard Industrial Classification of all Economic Activities. Report 09-90-02. Pretoria: Central Statistical Service.

SSA (1997). *Household Income and Expenditure Survey,1995*. Pretoria: Statistics South Africa.

SSA (2001). 1993 Supply and Use matrices for South Africa, Report No. 04-04-01. Pretoria: Statistics South Africa.

6. Technical appendix

This appendix provides the STATA code used to produce the results in the text. Enough comments have been added to the code to make it widely comprehensible and to aid those who wish to translate it into languages other than STATA.

6.1. STATA commands

The following boxes explain the commands that are frequently used in STATA.

Box 1 The #delimit command

The *#delimit command* means that every command line will end with a semi-colon (;).

Box 2 The set mem command

The *set memory command* allocates computer memory to STATA for exclusive use by this programme.

Box 3 The change directory (cd) command

The *change directory (cd) command* indicates the directory in which to open the do-files and data files.

Box 4 The set more off

The *set more off command* is used so that the computer can continuously run without pausing at every frame or window.

Box 5 The infile command

The *infile command* calls up a *dictionary file* (the file that stores data), which in turn reads the data from the specific provincial ASCII data set and stores it as variables in STATA format. Each province is given a code to distinguish it from the rest once the provincial data sets are merged.

Box 6 The gen (generate) command

The *generate command* is used to create a new variable.

Box 7 The replace command

The *replace command* overwrites the file or variable, if it already exists, in other words it changes the value of an existing variable.

Box 8 The drop command

The *drop command* will remove from memory the variables that are listed after the command.

Box 9 The keep command

The *keep command* will retain only the listed variables and drop all the others.

Box 10 The label var command

The *label var command* gives the description of the variable in order to identify it by name.

Box 11 The append command

The *append command* adds all the datasets in order to have a national data set.

Box 12 The sum (summary) command

The *summary command* is used to compute the mean of a variable in STATA.

6.2. Ies95.do

```
#delimit;

* This is the MASTER do-file. Various checks are performed on the data
*     to find any inconsistencies. Some observations are dropped in the
*     process. Once the entire dataset is 'cleaned' datafiles for various
*     provinces and regions, as well as a national datafile are created.

      NOTE: Prior to running this do-file, a command line stating the path
      needs to be entered, e.g. "cd c:\data\ies95\". All other do-files,
      dictionary files and ASCII files should also be stored in this folder
      for the programme to run correctly. All output (log-files and dta-
      files) is also stored in this directory.


set mem 100m;
```

```
set more off;

      do ies95check.do;
      do ies95create.do;
```

6.3. Ies95check.do

```
#delimit;

*Checks for inconsistencies and errors in observations. Each province's
*     ASCII file is read in using infile command, a province variable is
*     created and the settle variable is modified to only have two sub-
*     classes, namely urban and rural. Provincial datasets are stored
*     as .dta files, with filename corresponding to the PROVIDE Project
*     naming conventions.
;

*Reading data using infile command. Note that .dct is a Stata dictionary
file
*     that reads the ASCII-format datafiles and stores this data as Stata
      variables.;

      infile using dict_ec.dct;
      gen province = 2;
      replace settle = 1 if settle < 30;
      replace settle = 2 if settle > 30;
      save ecape.dta, replace;
      clear;

      infile using dict_fs.dct;
      gen province = 4;
      replace settle = 1 if settle < 30;
      replace settle = 2 if settle > 30;
      save fstate.dta, replace;
      clear;

      infile using dict_gt.dct;
      gen province = 7;
      replace settle = 1 if settle < 30;
      replace settle = 2 if settle > 30;
      save gteng.dta, replace;
      clear;

      infile using dict_kz.dct;
      gen province = 5;
      replace settle = 1 if settle < 30;
      replace settle = 2 if settle > 30;
      save kzn.dta, replace;
      clear;

      infile using dict_mp.dct;
      gen province = 8;
      replace settle = 1 if settle < 30;
      replace settle = 2 if settle > 30;
      save mpum.dta, replace;
      clear;

      infile using dict_nc.dct;
      gen province = 3;
      replace settle = 1 if settle < 30;
      replace settle = 2 if settle > 30;
      save ncape.dta, replace;
      clear;
```

```
      infile using dict_np.dct;
      gen province = 9;
      replace settle = 1 if settle < 30;
      replace settle = 2 if settle > 30;
      save lpopo.dta, replace;
      clear;

      infile using dict_nw.dct;
      gen province = 6;
      replace settle = 1 if settle < 30;
      replace settle = 2 if settle > 30;
      save nwest.dta, replace;
      clear;

      infile using dict_wc.dct;
      gen province = 1;
      replace settle = 1 if settle < 30;
      replace settle = 2 if settle > 30;
      save wcape.dta, replace;
      clear;

      *A log-file ies95check.log is created that contains summary data
      after checks have been performed and observations containing errors
      have been dropped.*
*     Do-file totalcheck.do is performed on each provincial dataset stored
      before.*
;

log using ies95check.log, replace;

      use ecape.dta, clear;
      quietly log off;
      do totalcheck.do;
      save ecape.dta, replace;

      use kzn.dta, clear;
      quietly log off;
      do totalcheck.do;
      save kzn.dta, replace;

      use wcape.dta, clear;
      quietly log off;
      do totalcheck.do;
      save wcape.dta, replace;

      use ncape.dta, clear;
      quietly log off;
      do totalcheck.do;
      save ncape.dta, replace;

      use nwest.dta, clear;
      quietly log off;
      do totalcheck.do;
      save nwest.dta, replace;

      use fstate.dta, clear;
      quietly log off;
      do totalcheck.do;
      save fstate.dta, replace;

      use gteng.dta, clear;
      quietly log off;
      do totalcheck.do;
      save gteng.dta, replace;
```

```
use mpum.dta, clear;
quietly log off;
do totalcheck.do;
save mpum.dta, replace;

use lpopo.dta, clear;
quietly log off;
do totalcheck.do;
save lpopo.dta, replace;

log close;
```

6.4. Ies95create.do

```
#delimit;

      After checks have been completed, final versions of datasets are
      stored. All redundant variables are dropped, while coeffs.do creates
      coefficients of income and expenditure. These coefficients are stored
      as variables starting with c* and are used when comparing income or
      expenditure patterns between household groups
                              *
;

      use ecape.dta, clear;
      rename b89f065 extot;
      drop inctot;
      rename b89f068 inctot;
      label var extot "Total reported expenditure";
      drop b* c* excat* hhdomser extot134 extot96 t* d* tot* vat*;
      do coeffs.do;
      save ecape.dta, replace;

      use fstate.dta, clear;
      rename b89f065 extot;
      drop inctot;
      rename b89f068 inctot;
      label var extot "Total reported expenditure";
      drop b* c* excat* hhdomser extot134 extot96 t* d* tot* vat*;
      do coeffs.do;
      save fstate.dta, replace;

      use gteng.dta, clear;
      rename b89f065 extot;
      drop inctot;
      rename b89f068 inctot;
      label var extot "Total reported expenditure";
      drop b* c* excat* hhdomser extot134 extot96 t* d* tot* vat*;
      do coeffs.do;
      save gteng.dta, replace;

      use kzn.dta, clear;
      rename b89f065 extot;
      drop inctot;
      rename b89f068 inctot;
      label var extot "Total reported expenditure";
      drop b* c* excat* hhdomser extot134 extot96 t* d* tot* vat*;
      do coeffs.do;
      save kzn.dta, replace;

      use mpum.dta, clear;
      rename b89f065 extot;
      drop inctot;
      rename b89f068 inctot;
```

```
      label var extot "Total reported expenditure";
      drop b* c* excat* hhdomser extot134 extot96 t* d* tot* vat*;
      do coeffs.do;
      save mpum.dta, replace;

      use ncape.dta, clear;
      rename b89f065 extot;
      drop inctot;
      rename b89f068 inctot;
      label var extot "Total reported expenditure";
      drop b* c* excat* hhdomser extot134 extot96 t* d* tot* vat*;
      do coeffs.do;
      save ncape.dta, replace;

      use lpopo.dta, clear;
      rename b89f065 extot;
      drop inctot;
      rename b89f068 inctot;
      label var extot "Total reported expenditure";
      drop b* c* excat* hhdomser extot134 extot96 t* d* tot* vat*;
      do coeffs.do;
      save lpopo.dta, replace;

      use nwest.dta, clear;
      rename b89f065 extot;
      drop inctot;
      rename b89f068 inctot;
      label var extot "Total reported expenditure";
      drop b* c* excat* hhdomser extot134 extot96 t* d* tot* vat*;
      do coeffs.do;
      save nwest.dta, replace;

      use wcape.dta, clear;
      rename b89f065 extot;
      drop inctot;
      rename b89f068 inctot;
      label var extot "Total reported expenditure";
      drop b* c* excat* hhdomser extot134 extot96 t* d* tot* vat*;
      do coeffs.do;
      save wcape.dta, replace;


*Creating a national dataset*
;
      append using ecape.dta;
      append using fstate.dta;
      append using gteng.dta;
      append using kzn.dta;
      append using mpum.dta;
      append using ncape.dta;
      append using lpopo.dta;
      append using nwest.dta;
      save ies95.dta, replace;

      label var province "Province";
      label define provlab    2 "Eastern Cape"
                              4 "Freestate"
                              7 "Gauteng"
                              5 "Kwazulu-Natal"
                              8 "Mpumalanga"
                              3 "Northern Cape"
                              9 "Limpopo"
                              6 "North-West"
                              1 "Western Cape";

      label values province provlab;
```

```
      label var race "Population group";
      label define racelab 1 "African" 2 "Coloured" 3 "Indian" 4 "White";
      label values race racelab;

      label var settle "Settlement";
      label define setlab 1 "Urban" 2 "Rural";
      label values settle setlab;


      gen region = .;
      replace region = 1 if province == 1 | province == 3;
      replace region = 2 if province == 2 | province == 5;
      replace region = 3 if province == 8 | province == 9;
      replace region = 4 if province == 4 | province == 6 | province == 7;

      label define reglab 1 "Wcoast" 2 "Ecoast" 3 "Border" 4 "Centre";
      label values region reglab;
      label var region "Regions";

      save ies95.dta, replace;

* Creating regional datasets*
;

      use ies95.dta,clear;
      keep if region == 1;
      save wcoast.dta, replace;

      use ies95.dta, clear;
      keep if region == 2;
      save ecoast.dta, replace;

      use ies95.dta, clear;
      keep if region == 3;
      save border.dta, replace;

      use ies95.dta, clear;
      keep if region == 4;
      save centre.dta, replace;

      Recreating provincial datasets - these now include new labels created
      above*
;

      use ies95.dta, clear;
      keep if province == 1;
      save wcape.dta, replace;

      use ies95.dta, clear;
      keep if province == 2;
      save ecape.dta, replace;

      use ies95.dta, clear;
      keep if province == 3;
      save ncape.dta, replace;

      use ies95.dta, clear;
      keep if province == 4;
      save fstate.dta, replace;

      use ies95.dta, clear;
      keep if province == 5;
      save kzn.dta, replace;

      use ies95.dta, clear;
```

```
      keep if province == 6;
      save nwest.dta, replace;

      use ies95.dta, clear;
      keep if province == 7;
      save gteng.dta, replace;

      use ies95.dta, clear;
      keep if province == 8;
      save mpum.dta, replace;

      use ies95.dta, clear;
      keep if province == 9;
      save lpopo.dta, replace;
      clear;
```

6.5. Totalcheck.do

```
#delimit;

      Checks each province for errors and drops observations that cause
      disparities between calculated totals and reported totals.
                                        *
;

do annualise.do;
annualises all monthly expenditures
           *
;
do inccat.do;
creates income categories based in income sources
                 *
;

*VAT on housing services (see section 1.7 of the questionnaire) is added
pro-rata to the various sub-components that make up total housing services.
This is necessary because reported expenditures in the SAM must be
inclusive of VAT        *
;

      gen vatorig = b07f011 + b07f012 + b07f013 + b07f014 + b07f015 +
      b07f016 ;
      replace b07f011 = b07f011 + (b07f011/vatorig*b07f017) if vatorig > 0
;
      replace b07f012 = b07f012 + (b07f012/vatorig*b07f017) if vatorig > 0
;
      replace b07f013 = b07f013 + (b07f013/vatorig*b07f017) if vatorig > 0
;
      replace b07f014 = b07f014 + (b07f014/vatorig*b07f017) if vatorig > 0
;
      replace b07f015 = b07f015 + (b07f015/vatorig*b07f017) if vatorig > 0
;
      replace b07f016 = b07f016 + (b07f016/vatorig*b07f017) if vatorig > 0
;


*The commands below check that the process of adding VAT was done
correctly, allowing for a 0.05c rounding error. Some households reported
VAT expenditures, but no other expenditures. Although this seems like a
reporting error, these VAT expenditures were simply added to a variable
called hhindtax (see excat134.do).  *
;

      gen vatnew = b07f011 + b07f012 + b07f013 + b07f014 + b07f015 +
      b07f016  - b07f017;
```

```
      gen vatdiff = vatorig-vatnew;
      replace vatdiff = 0 if vatdiff < 0.05 & vatdiff > -0.05;

do excat134.do;
do excat96.do;
above do-files create expenditure categories, first based on 134 SIC
categories, and thereafter on 96 expenditure categories as used by Stats SA
in their national SAM   *
;

do diffs.do;
checks reported totals and compares with calculated totals
           *
;
do totals.do;
checks that reported totals add up to reported total expenditure (b89f065)
                 *
;

*Dropping problem variables: A variable called _problem is created which
shows which observations have errors. If the original IES 1995 dataset is
required it is possible to block off the drop command below by putting a
star (*) in front of the command. Alternatively, this command can be kept
in which case all problem variables (types 1 to 3 - see below) are dropped.
Problem variables of type 1 to 3 are households that typically reported
totals, but provided no breakdowns of those totals are arrived at.
Consequently these observations could not be used. A detailed description
of the dropped observations is provided in the related Technical Paper.
                                                        *
*     Problem variables of type 4 are defined as those households that
                 *
*     reported VAT but no other household services expenditures. These
      observations are not dropped, since the VAT is added to hhindtax as
      mentioned before.       *
;

gen _problem = .;
replace _problem = 1 if d07f100 ~= 0 & t07f100 == 0;
replace _problem = 2 if d07f200 ~= 0 & t07f200 == 0;
replace _problem = 3 if d56f100 ~= 0;
replace _problem = 4 if vatorig == 0 & vatdiff ~= 0;
drop if _problem == 1 | _problem == 2 | _problem == 3;

quietly log on;

list vat* b07f010 *07f100 if d07f100~=0;

sum  d07f100 d07f200 d10f100
     d15f100 d17f100 d18f100 d19f100 d20f100 d21f100 d22f100 d23f100
     d24f100 d25f100 d26f100 d27f100 d28f100 d29f100 d30f100 d31f100
     d32f100 d33f100 d34f100 d35f100 d36f100 d37f100 d38f100 d39f100
     d40f100 d41f100 d42f100 d43f100 d44f100 d45f100 d46f100 d47f100
     d48f100 d49f100 d50f100 d51f100 d52f100 d53f100 d54f100 d55f100
     d56f100 d57f100 d58f100 d59f100 d60f100 d61f100 d62f100 d63f100
     d64f100 d65f100 d66f100 d67f100 d68f100 d69f100 d70f100  ;

*Table below: totdiff should equal zero. Created in totals.do and compares
category totals with grand total b89f065;
sum totdiff;

Table below: extot134 should equal extot96 and b89f065 (grand total);
sum extot* b89f065;
Table below: inctot (calculated total income) should equal b89f068
(reported total income);
sum inctot b89f068;
```

6.6. Annualise.do

```
#delimit;

Changes all monthly variables to annual variables. This only applies to
those variables actually used in the analysis. E.g, the monthly summary
variables b89f001 - b89f024 were not annualised since they are not used for
the analysis;

replace     b07f001     =     b07f001     *     12    ;
replace     b07f002     =     b07f002     *     12    ;
replace     b07f003     =     b07f003     *     12    ;
replace     b07f004     =     b07f004     *     12    ;
replace     b07f005     =     b07f005     *     12    ;
replace     b07f006     =     b07f006     *     12    ;
replace     b07f007     =     b07f007     *     12    ;
replace     b07f008     =     b07f008     *     12    ;
replace     b07f009     =     b07f009     *     12    ;
replace     b07f010     =     b07f010     *     12    ;
replace     b07f011     =     b07f011     *     12    ;
replace     b07f012     =     b07f012     *     12    ;
replace     b07f013     =     b07f013     *     12    ;
replace     b07f014     =     b07f014     *     12    ;
replace     b07f015     =     b07f015     *     12    ;
replace     b07f016     =     b07f016     *     12    ;
replace     b07f017     =     b07f017     *     12    ;
replace     b07f100     =     b07f100     *     12    ;

replace     b10f100     =     b10f100     *     12    ;

replace     b15f001     =     b15f001     *     12    ;
replace     b15f002     =     b15f002     *     12    ;
replace     b15f003     =     b15f003     *     12    ;
replace     b15f004     =     b15f004     *     12    ;
replace     b15f005     =     b15f005     *     12    ;
replace     b15f006     =     b15f006     *     12    ;
replace     b15f007     =     b15f007     *     12    ;
replace     b15f008     =     b15f008     *     12    ;
replace     b15f009     =     b15f009     *     12    ;
replace     b15f010     =     b15f010     *     12    ;
replace     b15f011     =     b15f011     *     12    ;
replace     b15f012     =     b15f012     *     12    ;
replace     b15f013     =     b15f013     *     12    ;
replace     b15f014     =     b15f014     *     12    ;
replace     b15f015     =     b15f015     *     12    ;
replace     b15f100     =     b15f100     *     12    ;
replace     b16f001     =     b16f001     *     12    ;
replace     b16f002     =     b16f002     *     12    ;
replace     b16f003     =     b16f003     *     12    ;
replace     b16f004     =     b16f004     *     12    ;
replace     b16f005     =     b16f005     *     12    ;
replace     b17f006     =     b17f006     *     12    ;
replace     b17f007     =     b17f007     *     12    ;
replace     b17f008     =     b17f008     *     12    ;
replace     b17f009     =     b17f009     *     12    ;
replace     b17f010     =     b17f010     *     12    ;
replace     b17f011     =     b17f011     *     12    ;
replace     b17f012     =     b17f012     *     12    ;
replace     b17f013     =     b17f013     *     12    ;
replace     b17f014     =     b17f014     *     12    ;
replace     b17f015     =     b17f015     *     12    ;
replace     b17f016     =     b17f016     *     12    ;
replace     b17f017     =     b17f017     *     12    ;
```

```
replace     b17f100     =     b17f100     *     12    ;
replace     b18f001     =     b18f001     *     12    ;
replace     b18f002     =     b18f002     *     12    ;
replace     b18f003     =     b18f003     *     12    ;
replace     b18f004     =     b18f004     *     12    ;
replace     b18f005     =     b18f005     *     12    ;
replace     b18f100     =     b18f100     *     12    ;
replace     b19f001     =     b19f001     *     12    ;
replace     b19f002     =     b19f002     *     12    ;
replace     b19f003     =     b19f003     *     12    ;
replace     b19f004     =     b19f004     *     12    ;
replace     b19f005     =     b19f005     *     12    ;
replace     b19f100     =     b19f100     *     12    ;
replace     b20f001     =     b20f001     *     12    ;
replace     b20f002     =     b20f002     *     12    ;
replace     b20f003     =     b20f003     *     12    ;
replace     b20f004     =     b20f004     *     12    ;
replace     b20f005     =     b20f005     *     12    ;
replace     b20f006     =     b20f006     *     12    ;
replace     b20f007     =     b20f007     *     12    ;
replace     b20f008     =     b20f008     *     12    ;
replace     b20f009     =     b20f009     *     12    ;
replace     b20f010     =     b20f010     *     12    ;
replace     b20f011     =     b20f011     *     12    ;
replace     b20f100     =     b20f100     *     12    ;
replace     b21f001     =     b21f001     *     12    ;
replace     b21f002     =     b21f002     *     12    ;
replace     b21f003     =     b21f003     *     12    ;
replace     b21f004     =     b21f004     *     12    ;
replace     b21f005     =     b21f005     *     12    ;
replace     b21f006     =     b21f006     *     12    ;
replace     b21f007     =     b21f007     *     12    ;
replace     b21f008     =     b21f008     *     12    ;
replace     b21f009     =     b21f009     *     12    ;
replace     b21f010     =     b21f010     *     12    ;
replace     b21f011     =     b21f011     *     12    ;
replace     b21f012     =     b21f012     *     12    ;
replace     b21f013     =     b21f013     *     12    ;
replace     b21f014     =     b21f014     *     12    ;
replace     b21f015     =     b21f015     *     12    ;
replace     b21f016     =     b21f016     *     12    ;
replace     b21f017     =     b21f017     *     12    ;
replace     b21f018     =     b21f018     *     12    ;
replace     b21f019     =     b21f019     *     12    ;
replace     b21f020     =     b21f020     *     12    ;
replace     b21f021     =     b21f021     *     12    ;
replace     b21f100     =     b21f100     *     12    ;
replace     b22f001     =     b22f001     *     12    ;
replace     b22f002     =     b22f002     *     12    ;
replace     b22f003     =     b22f003     *     12    ;
replace     b22f004     =     b22f004     *     12    ;
replace     b22f005     =     b22f005     *     12    ;
replace     b22f006     =     b22f006     *     12    ;
replace     b22f007     =     b22f007     *     12    ;
replace     b22f008     =     b22f008     *     12    ;
replace     b22f009     =     b22f009     *     12    ;
replace     b22f010     =     b22f010     *     12    ;
replace     b22f011     =     b22f011     *     12    ;
replace     b22f012     =     b22f012     *     12    ;
replace     b22f013     =     b22f013     *     12    ;
replace     b22f100     =     b22f100     *     12    ;
replace     b23f001     =     b23f001     *     12    ;
replace     b23f002     =     b23f002     *     12    ;
replace     b23f003     =     b23f003     *     12    ;
replace     b23f004     =     b23f004     *     12    ;
replace     b23f100     =     b23f100     *     12    ;
```

```
replace     b24f001     =     b24f001     *     12    ;
replace     b24f002     =     b24f002     *     12    ;
replace     b24f003     =     b24f003     *     12    ;
replace     b24f004     =     b24f004     *     12    ;
replace     b24f005     =     b24f005     *     12    ;
replace     b24f006     =     b24f006     *     12    ;
replace     b24f100     =     b24f100     *     12    ;
replace     b25f001     =     b25f001     *     12    ;
replace     b25f002     =     b25f002     *     12    ;
replace     b25f003     =     b25f003     *     12    ;
replace     b25f004     =     b25f004     *     12    ;
replace     b25f005     =     b25f005     *     12    ;
replace     b25f100     =     b25f100     *     12    ;
replace     b26f001     =     b26f001     *     12    ;
replace     b26f002     =     b26f002     *     12    ;
replace     b26f003     =     b26f003     *     12    ;
replace     b26f004     =     b26f004     *     12    ;
replace     b26f005     =     b26f005     *     12    ;
replace     b26f006     =     b26f006     *     12    ;
replace     b26f007     =     b26f007     *     12    ;
replace     b26f008     =     b26f008     *     12    ;
replace     b26f009     =     b26f009     *     12    ;
replace     b26f010     =     b26f010     *     12    ;
replace     b26f011     =     b26f011     *     12    ;
replace     b26f100     =     b26f100     *     12    ;
replace     b27f001     =     b27f001     *     12    ;
replace     b27f002     =     b27f002     *     12    ;
replace     b27f003     =     b27f003     *     12    ;
replace     b27f100     =     b27f100     *     12    ;
replace     b27f200     =     b27f200     *     12    ;
replace     b28f001     =     b28f001     *     12    ;
replace     b28f002     =     b28f002     *     12    ;
replace     b28f003     =     b28f003     *     12    ;
replace     b28f004     =     b28f004     *     12    ;
replace     b28f100     =     b28f100     *     12    ;
replace     b29f001     =     b29f001     *     12    ;
replace     b29f002     =     b29f002     *     12    ;
replace     b29f003     =     b29f003     *     12    ;
replace     b29f004     =     b29f004     *     12    ;
replace     b29f100     =     b29f100     *     12    ;
replace     b30f001     =     b30f001     *     12    ;
replace     b30f002     =     b30f002     *     12    ;
replace     b30f003     =     b30f003     *     12    ;
replace     b30f004     =     b30f004     *     12    ;
replace     b30f005     =     b30f005     *     12    ;
replace     b30f006     =     b30f006     *     12    ;
replace     b30f007     =     b30f007     *     12    ;
replace     b30f100     =     b30f100     *     12    ;
replace     b31f001     =     b31f001     *     12    ;
replace     b31f002     =     b31f002     *     12    ;
replace     b31f003     =     b31f003     *     12    ;
replace     b31f004     =     b31f004     *     12    ;
replace     b31f005     =     b31f005     *     12    ;
replace     b31f006     =     b31f006     *     12    ;
replace     b31f007     =     b31f007     *     12    ;
replace     b31f100     =     b31f100     *     12    ;
replace     b32f001     =     b32f001     *     12    ;
replace     b32f002     =     b32f002     *     12    ;
replace     b32f003     =     b32f003     *     12    ;
replace     b32f004     =     b32f004     *     12    ;
replace     b32f100     =     b32f100     *     12    ;
replace     b33f001     =     b33f001     *     12    ;
replace     b33f002     =     b33f002     *     12    ;
replace     b33f003     =     b33f003     *     12    ;
replace     b33f004     =     b33f004     *     12    ;
replace     b33f005     =     b33f005     *     12    ;
```

```
replace     b33f006     =     b33f006     *     12    ;
replace     b33f007     =     b33f007     *     12    ;
replace     b33f008     =     b33f008     *     12    ;
replace     b33f009     =     b33f009     *     12    ;
replace     b33f010     =     b33f010     *     12    ;
replace     b33f011     =     b33f011     *     12    ;
replace     b33f012     =     b33f012     *     12    ;
replace     b33f013     =     b33f013     *     12    ;
replace     b33f014     =     b33f014     *     12    ;
replace     b33f015     =     b33f015     *     12    ;
replace     b33f016     =     b33f016     *     12    ;
replace     b33f017     =     b33f017     *     12    ;
replace     b33f018     =     b33f018     *     12    ;
replace     b33f100     =     b33f100     *     12    ;
replace     b34f001     =     b34f001     *     12    ;
replace     b34f002     =     b34f002     *     12    ;
replace     b34f003     =     b34f003     *     12    ;
replace     b34f004     =     b34f004     *     12    ;
replace     b34f005     =     b34f005     *     12    ;
replace     b34f006     =     b34f006     *     12    ;
replace     b34f007     =     b34f007     *     12    ;
replace     b34f008     =     b34f008     *     12    ;
replace     b34f009     =     b34f009     *     12    ;
replace     b34f010     =     b34f010     *     12    ;
replace     b34f100     =     b34f100     *     12    ;
replace     b35f001     =     b35f001     *     12    ;
replace     b35f002     =     b35f002     *     12    ;
replace     b35f003     =     b35f003     *     12    ;
replace     b35f004     =     b35f004     *     12    ;
replace     b35f005     =     b35f005     *     12    ;
replace     b35f006     =     b35f006     *     12    ;
replace     b35f100     =     b35f100     *     12    ;

replace     b56f001     =     b56f001     *     12    ;
replace     b56f002     =     b56f002     *     12    ;
replace     b56f003     =     b56f003     *     12    ;
replace     b56f004     =     b56f004     *     12    ;
replace     b56f005     =     b56f005     *     12    ;
replace     b56f007     =     b56f007     *     12    ;

*NOTE: b56f006 is already an annual figure

;
replace b56f100 = b89f046;

*NOTE: b56f100 not equal to b89f046: this has to do with the fact that
b56f006 is not annualised - Assumption: furniture removal happens just once
a year.Also note that b89f046 is already an annual figure.

;

replace     b59f001     =     b59f001     *     12    ;
replace     b59f002     =     b59f002     *     12    ;
replace     b59f003     =     b59f003     *     12    ;
replace     b59f004     =     b59f004     *     12    ;
replace     b59f005     =     b59f005     *     12    ;
replace     b59f006     =     b59f006     *     12    ;
replace     b59f007     =     b59f007     *     12    ;
replace     b59f008     =     b59f008     *     12    ;
replace     b59f100     =     b59f100     *     12    ;
```

6.7. Excat134

```
#delimit;

*134 expenditure categories are based on the SIC codes for commodities. A
similar procedure was used for the rest of country. Split was done for the
variables, which fall into more than one category. For example, the
category for “Hunting, forestry and fishing”, includes eggs (b20f001) but
eggs are also included in the category of “Other food products”. Therefore,
split in this case needs to be done to distribute the eggs equally since
they appear more than once. The same explanation holds for all other splits
that may be found in this paper.
;

*Category 1 Agriculture, hunting, forestry and fishing*
;
        gen excat001        =   (0.5*b20f011)  +
                                b21f001        +
                                b21f002        +
                                b21f003        +
                                b21f004        +
                                b21f005        +
                                b21f006        +
                                b21f007        +
                                b21f008        +
                                b21f009        +
                                b21f010        +
                                b21f011        +
                                b21f012        +
                                b21f013        +
                                b22f001        +
                                b22f002        +
                                b22f003        +
                                b22f004        +
                                b22f005        +
                                b22f006        +
                                b22f007        +
                                b22f011        +
                                b36f001        +
                                b36f010        +
                                b36f011        +
                                (0.5*b64f002)  +
                                b64f004        ;

*Category 2 Mining of coal and lignite*
;
        gen excat002        =   b36f006        ;

*Category 3 Gold and uranium ore*
;
        gen excat003        =       0          ;

*Category 4 Other mining and quarrying*
;
        gen excat004        =       0          ;

*Category 5 Mining of diamonds*
;
        gen excat005        =       0          ;

*Category 6 Service activities incidental to mining of minerals*
;
        gen excat006        =       0          ;
```

```
*Category 7 Production, processing and preserving of meat and meat
products*
;
        gen excat007        =   b16f001        +
                                b16f002        +
                                b16f003        +
                                b16f004        +
                                b16f005        +
                                b17f006        +
                                b17f007        +
                                b17f008        +
                                b17f009        +
                                b17f010        +
                                b17f011        +
                                b17f012        +
                                b17f013        +
                                b17f014        +
                                b17f015        +
                                b17f017        +
                                b19f005        ;


*Category 8 Processing and preserving of fish and fish products*
;
        gen excat008        =   b18f001        +
                                b18f002        +
                                b18f003        +
                                b18f004        +
                                b18f005        ;


*Category 9 Processing and preserving of fruit and vegetables*
;
        gen excat009        =   b21f014        +
                                b21f015        +
                                b21f016        +
                                b21f017        +
                                b21f018        +
                                b21f019        +
                                b21f021        +
                                b22f008        +
                                b22f009        +
                                b22f010        +
                                b22f013        +
                                b24f002        +
                                b24f003        ;



*Category 10 Vegetable and animal oils and fats*
;
        gen excat010        =   b19f002        +
                                b19f003        ;


*Category 11 Dairy products*
;
        gen excat011        =   b19f001        +
                                b20f001        +
                                b20f002        +
                                b20f003        +
                                b20f004        +
                                b20f005        +
                                b20f006        +
                                b20f007        +
                                b20f008        +
```

```
                                b20f010         ;


*Category 12 Grain mill products*
;
        gen excat012        =   b15f001         +
                                b15f002         +
                                b15f003         +
                                b15f004         +
                                b15f005         +
                                b15f006         +
                                b15f007         +
                                b15f008         +
                                b15f015         +
                                b35f005         ;

*Category 13 Starches and starch products*
;
        gen excat013        =   b23f003         ;


*Category 14 Animal feeds*
;
        gen excat014        =   b64f003         ;

*Category 15 Bakery products*
;
        gen excat015        =   b15f010         +
                                b15f011         +
                                b15f012         +
                                b15f013         ;



*Category 16 Sugar, including golden syrup and castor sugar*
;
        gen excat016        =   b23f001         +
                                b23f002         +
                                b23f004         +
                                b24f001         ;



*Category 17 Cocoa, chocolate and sugar confectionery*
;
        gen excat017        =   b24f004         +
                                b24f005         +
                                b24f006         +
                                b25f005         ;


*Category 18 Macaroni, noodles, couscous and similar farinaceous products*
;
        gen excat018        =   b15f009         ;


*Category 19 Other food products n.e.c.*
;
        gen excat019        =   b15f014         +
                                b17f016         +
                                b19f004         +
                                b20f009         +
```

```
                                (0.5*b20f011)  +
                                b21f020        +
                                b22f012        +
                                b25f001        +
                                b25f002        +
                                b25f003        +
                                b25f004        +
                                b26f001        +
                                b26f002        +
                                b26f003        +
                                b26f005        +
                                b26f006        +
                                b26f007        +
                                b26f008        +
                                b26f009        +
                                b26f010        +
                                b26f011        ;


*Category 20 Distilling, rectifying and blending of spirits, wine*
;
        gen excat020        =   b31f001        +
                                b31f002        +
                                b31f005        +
                                b31f006        +
                                b31f007        ;


*Category 21 Beer and other malt liquors and malt*
;
        gen excat021        =   b31f003        +
                                b31f004        ;


*Category 22 Soft drinks, production of mineral waters*
;
        gen excat022        =   b29f001        +
                                b29f002        +
                                b29f003        +
                                b29f004        ;


*Category 23 Tobacco products*
;
        gen excat023        =   b32f001        +
                                b32f002        +
                                b32f003        ;


*Category 24 Preparation and spinning of textile fibres, weaving of
textiles*
;
        gen excat024        =   b46f001        +
                                b46f002        +
                                b46f003        ;

*Category 25 Finishing of textiles*
;
        gen excat025        =       0          ;


*Category 26 Made-up textile articles, except apparel*
;
        gen excat026        =   b41f001        +
                                b48f001        +
```

```
                                b48f002        +
                                b48f003        +
                                b48f004        +
                                b48f005        +
                                b48f006        +
                                b48f007        +
                                b63f008        ;


*Category 27 Carpets and rugs*
;
        gen excat027        =   b47f008        +
                                b47f009        ;


*Category 28 Cordage, rope, twine and netting*
;
        gen excat028        =        0         ;


*Category 29 Other textiles n.e.c.*
;
        gen excat029        =   b46f004        ;


*Category 30 Knitted and crocheted fabrics and articles*
;
        gen excat030        =   b37f005        +
                                b37f008        +
                                b38f005        +
                                b38f008        +
                                b39f005        +
                                b39f007        +
                                b40f005        +
                                b40f007        ;


*Category 31 Wearing apparel, except fur apparel*
;
        gen excat031        =   b37f001        +
                                b37f002        +
                                b37f003        +
                                b37f004        +
                                b37f006        +
                                b37f007        +
                                b37f010        +
                                b38f001        +
                                b38f002        +
                                b38f003        +
                                b38f004        +
                                b38f006        +
                                b38f007        +
                                b38f009        +
                                b38f010        +
                                b39f001        +
                                b39f002        +
                                b39f003        +
                                b39f004        +
                                b39f006        +
                                b39f008        +
                                b39f010        +
                                b40f001        +
                                b40f002        +
                                b40f003        +
```

```
                                b40f004        +
                                b40f006        +
                                b40f008        +
                                b40f009        +
                                b40f010        +
                                b41f002        +
                                b41f003        ;


*Category 32 Dressing and dyeing of fur, articles of fur*
;
        gen excat032        =        0         ;


*Category 33 Tanning and dressing of leather*
;
        gen excat033        =        0         ;


*Category 34 Luggage, handbags and the like, saddlery and harness*
;
        gen excat034        =  b66f002         ;


*Category 35 Footwear*
;
        gen excat035        =   b42f001        +
                                b42f002        +
                                b42f003        +
                                b42f004        +
                                b42f005        +
                                b43f001        +
                                b43f002        +
                                b43f003        +
                                b43f004        +
                                b43f005        +
                                b44f001        +
                                b44f002        +
                                b44f003        +
                                b44f004        +
                                b44f005        +
                                b45f001        +
                                b45f002        +
                                b45f003        +
                                b45f004        +
                                b45f005        +
                                b42f200        ;


*Category 36 Sawmilling and planing of wood, products of wood, cork, straw
and plaiting materials*
;
        gen excat036        =   b36f002        ;


*Category 37 Pulp, paper and paperboard*
;
        gen excat037        =        0         ;
```

```
*Category 38 Corrugated paper and paperboard, containers of paper and
paperboard*
;
        gen excat038        =        0         ;


*Category 39 Other articles of paper and paperboard*
;
        gen excat039        =   b60f010        +
                                b61f010        +
                                b62f005        +
                                b33f015        +
                                b33f016        +
                                b33f017        +
                                b34f008        +
                                b34f009        ;


*Category 40 Publishing (excluding recorded media) and printing*
;
        gen excat040        =   b60f009        +
                                b61f009        +
                                b62f001        +
                                b62f002        +
                                b62f003        ;


*Category 41 Publishing and reproduction of recorded media*
;
        gen excat041        =   b63f004        ;


*Category 42 Coke oven products, processing of nuclear fuel*
;
        gen excat042        =        0         ;


*Category 43 Petroleum refineries/synthesisers*
;
        gen excat043        =   b36f003        +
                                b36f007        +
                                b36f012        +
                                b55f002        +
                                b55f006        ;

*Category 44 Basic chemicals, except fertilizers and nitrogen compounds*
;
        gen excat044        =        0         ;


*Category 45 Fertilizers and nitrogen compounds*
;
        gen excat045        =   (0.25*b64f002) ;


*Category 46 Plastics in primary forms and synthetic rubber*
;
        gen excat046        =        0         ;


*Category 47 Pesticides and other agrochemical products*
;
        gen excat047        =   (0.25*b64f002) ;
```

```
*Category 48 Paints, varnishes and similar coatings, printing ink and
mastics*
;
        gen excat048        =        0         ;



*Category 49 Pharmaceuticals, medicinal chemicals and botanical products*
;
        gen excat049        =   b51f004        +
                                (0.5*b51f005)  +
                                b51f007        +
                                b52f006        +
                                b52f007        ;

*Category 50 Soap, detergents, cleaning-, polishing-, perfume- and toilet
preparations*
;
        gen excat050        =   b33f005        +
                                b33f006        +
                                b33f007        +
                                b33f008        +
                                b33f009        +
                                b33f010        +
                                b33f011        +
                                b33f012        +
                                b33f013        +
                                b33f014        +
                                b34f001        +
                                b34f002        +
                                b34f003        +
                                b34f004        ;

*Category 51 Other chemical products n.e.c.*
;
        gen excat051        =   b26f004        +
                                (1/3*b64f001)  ;

*Category 52 Man-made fibres*
;
        gen excat052         =        0        ;


*Category 53 Rubber tyres and tubes, retreading, rebuilding of rubber
tyres*
;
        gen excat053        =   b55f007        +
                                b55f008        ;


*Category 54 Other rubber products*
;
        gen excat054        =        0         ;

*Category 55 Plastic products*
;
        gen excat055        =   b34f007        +
                                (1/3*b50f001)  ;

*Category 56 Glass and glass products*
;
        gen excat056        =   (1/3*b50f001)  ;

*Category 57 Non-structural non-refractory ceramicware*
;
        gen excat057        =        0         ;
```

```
*Category 58 Refractory and structural non-refractory clay and ceramic
products*
;
        gen excat058        =        0         ;

*Category 59 Cement, lime and plaster*
;
        gen excat059        =        0         ;

*Category 60 Articles of concrete, cement, plaster, stone and other non-
metallic mineral products n.e.c.*
;
        gen excat060        =   b07f022        +
                                b70f004        ;

*Category 61 Basic iron and steel, casting of iron and steel*
;
        gen excat061        =        0         ;

*Category 62 Basic precious and non-ferrous metals, casting of non-ferrous
metals*
;
        gen excat062        =        0         ;

*Category 63 Structural metal products, tanks, reservoirs and steam
generators*
;
        gen excat063        =        0         ;

*Category 64 Forging, pressing, stamping and roll-forming of metal, powder
metallurgy*
;
        gen excat064        =        0         ;

*Category 65 Treatment and coating of metals, general mechanical
engineering*
;
        gen excat065        =        0         ;

*Category 66 Cutlery, hand tools and general hardware*
;
        gen excat066        =   b50f004        +
                                (1/3*b50f001)  ;


*Category 67 Other fabricated metal products n.e.c.*
;
        gen excat067        =   b34f010        +
                                b50f003        +
                                b50f005        ;

*Category 68 Engines and turbines, except aircraft, vehicle and motor cycle
engines*
;
        gen excat068        =        0         ;


*Category 69 Pumps, compressors, taps and valves*
;
        gen excat069        =   (1/3*b64f001)  ;

*Category 70 Bearings, gears, gearing and driving elements*
;
        gen excat070        =        0         ;
```

```
*Category 71 Ovens, furnaces and furnace burners*
;
        gen excat071        =        0         ;

*Category 72 Lifting and handling equipment*
;
        gen excat072        =        0         ;

*Category 73 Other general purpose machinery*
;
        gen excat073        =        0         ;

*Category 74 Agricultural and forestry machinery*
;
        gen excat074        =   b49f012        +
                                b49f017        ;


*Category 75 Machine-tools*
;
        gen excat075        =   b63f009        ;

*Category 76 Machinery for metallurgy*
;
        gen excat076        =        0         ;

*Category 77 Machinery for mining, quarrying and construction*
;
        gen excat077        =        0         ;

*Category 78 Machinery for food, beverage and tobacco processing*
;
        gen excat078        =        0         ;

*Category 79 Machinery for textile, apparel and leather production*
;
        gen excat079        =   b49f008        +
                                b49f016        ;


*Category 80 Weapons and ammunition*
;
        gen excat080        =        0         ;

*Category 81 Other special purpose machinery*
;
        gen excat081        =        0         ;

*Category 82 Household appliances n.e.c.*
;
        gen excat082        =   b33f003        +
                                b49f001        +
                                b49f002        +
                                b49f003        +
                                b49f004        +
                                b49f005        +
                                b49f006        +
                                b49f007        +
                                b49f009        +
                                b49f010        +
                                b49f011        +
                                b49f013        +
                                b49f014        +
                                b49f015        +
                                b49f018        ;
```

```
*Category 83 Office, accounting and computing machinery*
;
        gen excat083        =   b58f001        ;


*Category 84 Electric motors, generators and transformers*
;
        gen excat084        =        0         ;

*Category 85 Electricity distribution and control apparatus*
;
        gen excat085        =        0         ;

*Category 86 Insulated wire and cable*
;
        gen excat086        =        0         ;

*Category 87 Accumulators, primary cells and primary batteries*
;
        gen excat087        =   b36f009        +
                                b55f009        ;



*Category 88 Electric lamps and lighting equipment*
;
        gen excat088        =   b50f002        ;


*Category 89 Other electrical equipment n.e.c.*
;
        gen excat089        =        0         ;

*Category 90 Radio-, television- and communication equipment and apparatus*
;
        gen excat090        =   b58f002        +
                                b58f003        +
                                b63f002        +
                                b63f003        ;


*Category 91 Medical, precision- and optical instruments, watches and
clocks*
;
        gen excat091        =   b51f006        +
                                b52f008        +
                                b63f005        +
                                (0.5*b66f001)  ;


*Category 92 Motor vehicles (including their engines)*
;
        gen excat092        =   b53f001        +
                                b53f002        +
                                b54f001        +
                                b54f002        ;


*Category 93 Bodies (coachwork) for motor vehicles, trailers and semi-
trailers*
;
        gen excat093        =   b53f005        +
                                b54f005        ;
```

```
*Category 94 Parts and accessories for motor vehicles and their engines*
;
        gen excat094        =   b55f010        ;


*Category 95 Building and repairing of ships, pleasure- and sporting boats*
;
        gen excat095        =   b63f006        ;


*Category 96 Railway and tramway locomotives and rolling stock*
;
        gen excat096        =        0         ;

*Category 97 Aircraft and spacecraft*
;
        gen excat097        =        0         ;

*Category 98 Other transport equipment n.e.c.*
;
        gen excat098        =   b53f003        +
                                b53f004        +
                                b54f003        +
                                b54f004        ;


*Category 99 Furniture*
;
        gen excat099        =   b47f001        +
                                b47f002        +
                                b47f003        +
                                b47f004        +
                                b47f005        +
                                b47f006        +
                                b47f007        +
                                b47f010        ;


*Category 100 Jewellery and related articles*
;
        gen excat100        =   (0.5*b66f001)  ;


*Category 101 Other manufacturing n.e.c.*
;
        gen excat101        =   b32f004        +
                                b33f004        +
                                b33f018        +
                                b34f005        +
                                b34f006        +
                                b36f005        +
                                b63f001        +
                                b63f007        +
                                b64f005        +
                                b64f006        +
                                b66f003        +
                                b66f004        ;

*Category 102 Recycling of metal- and non-metal waste and scrap*
;
        gen excat102        =        0         ;

*Category 103 Electricity, gas, steam and hot water supply*
;
        gen excat103        =   b07f012        +
```

```
                                b07f013        +
                                b07f014        +
                                b36f004        ;

*Category 104 Collection, purification and distribution of water*
;
        gen excat104        =   b07f011        ;


*Category 105 Site preparation, construction of civil engineering
structures*
;
        gen excat105        =        0         ;

*Category 106 Buildings, specialist trade contractors, building
installation, building completion*
;
        gen excat106        =        0         ;

*Category 107 Construction of other structures (e.g. swimming pools, tennis
courts)*
;
        gen excat107        =   b07f020        +
                                b07f021        ;

*Category 108 Renting of construction or demolition equipment with
operators*
;
        gen excat108        =        0         ;

*Category 109 Wholesale trade and commission trade, except of motor
vehicles and motor cycles*
;
        gen excat109        =        0         ;

*Category 110 Retail trade, repair of personal and household goods*
;
        gen excat110        =   b42f006        +
                                b43f006        +
                                b44f006        +
                                b45f006        +
                                b47f011        +
                                b48f008        +
                                b49f019        +
                                b63f010        +
                                (1/3*b64f001)  +
                                b65f008        +
                                b65f011        ;

*Category 111 Sale of motor vehicles*
;
        gen excat111        =        0         ;

*Category 112 Maintenance and repair of motor vehicles*
;
        gen excat112       =    b55f011        +
                                b55f012        +
                                b55f013        +
                                b55f014        +
                                b55f015        +
                                b55f016        +
                                b55f019        ;


*Category 113 Sale of motor vehicle parts and accessories*
;
```

```
        gen excat113        =        0         ;

*Category 114 Sale, maintenance and repair of motor cycles and related
parts and accessories*
;
        gen excat114        =        0         ;

*Category 115 Retail sale of automotive fuel*
;
        gen excat115        =        0         ;


*Category 116 Hotels, camping sites and other provision of short-stay
accommodation*
;
        gen excat116        =   b07f025        +
                                b07f026        +
                                b07f027        ;

*Category 117 Restaurants, bars and canteens*
;
        gen excat117        =   b27f001        +
                                b27f002        +
                                b27f003        +
                                b28f001        +
                                b28f002        +
                                b28f003        +
                                b28f004        +
                                b30f001        +
                                b30f002        +
                                b30f003        +
                                b30f004        +
                                b30f005        +
                                b30f006        +
                                b30f007        ;



*Category 118 Transport, supporting and help activities related to
transport*
;
        gen excat118        =   b55f001        +
                                b56f001        +
                                b56f002        +
                                b56f003        +
                                b56f004        +
                                b56f005        +
                                b56f006        +
                                b56f007        +
                                b57f001        +
                                b57f002        +
                                b57f003        +
                                b57f004        +
                                b57f005        +
                                b57f006        ;

*Category 119 Post, courier activities and telecommunications*
;
        gen excat119        =   b59f001        +
                                b59f002        +
                                b59f003        +
                                b59f005        +
                                b59f006        +
                                b59f007        +
                                b59f008        +
```

```
                                b65f003        ;

*Category 120 Financial intermediation*
;
        gen excat120        =   b69f001        +
                                b69f002        +
                                b07f008        ;


*Category 121 Real estate activities*
;
        gen excat121        =   b07f001        +
                                b07f004        +
                                b07f005        +
                                b07f006        +
                                b07f024        ;

*Category 122 Renting of machinery and equipment, without servers*
;
        gen excat122        =   b35f004        +
                                b46f005        +
                                b57f007        +
                                b65f002        +
                                b65f004        ;

*Category 123 Computer and related activities*
;
        gen excat123        =        0         ;

*Category 124 Research and development*
;
        gen excat124        =        0         ;

*Category 125 Other business activities*
;
        gen excat125        =   b07f019        +
                                b65f009        +
                                b70f001        ;

*Category 126 Sales of goods and services by the government*
;
        gen excat126        =        0         ;

*Category 127 Education*
;
        gen excat127        =   b55f018        +
                                b60f001        +
                                b60f002        +
                                b60f003        +
                                b60f004        +
                                b60f005        +
                                b60f006        +
                                b60f007        +
                                b60f008        +
                                b60f011        +
                                b61f001        +
                                b61f002        +
                                b61f003        +
                                b61f004        +
                                b61f005        +
                                b61f006        +
                                b61f007        +
                                b61f008        +
                                b61f011        ;

*Category 128 Human health activities, veterinary activities*
```

```
;
        gen excat128        =   b51f001        +
                                b51f002        +
                                b51f003        +
                                (0.5*b51f005)  +
                                b52f001        +
                                b52f002        +
                                b52f003        +
                                b52f004        +
                                b52f005        +
                                b65f007        +
                                b69f007        ;

*Category 129 Activities of membership organisations n.e.c.*
;
        gen excat129        =   b67f003        +
                                b67f004        +
                                b67f007        +
                                b67f008        +
                                b70f005        ;

*Category 130 Recreational, cultural and sporting activities*
;
        gen excat130        =   b62f004        +
                                b65f005        +
                                b65f010        +
                                b67f005        ;

*Category 131 Other services, profit seeking*
;
        gen excat131        =   b07f015        +
                                b36f008        ;


*Category 132 Other services, non-profit seeking*
;
        gen excat132        =   b07f023        +
                                b07f016        ;

*Category 133 Other service activities*
;
        gen excat133        =   b33f001        +
                                b33f002        +
                                b35f001        +
                                b35f002        +
                                b35f003        +
                                b35f006        +
                                b70f003        ;

*Category 134 Other activities not adequately defined*
;
        gen excat134        =        0         ;



*Payments for domestic labour services*
;
      gen hhdomser          =   b10f100          ;
      label var hhdomser "Payments for domestic labour services";

*Household totals*
;
      gen hhtotals          =   b67f001         +
                                b67f002         +
                                b70f002         ;
      label var hhtotals "Household totals";
```

```
*Household income taxes*
;
      gen hhinctax          =       b68f100           ;
      label var hhinctax "Household income taxes";

*Household indirect taxes*
;
      gen hhindtax          = b07f010           +
                              vatdiff           +
                              b55f003           +
                              b55f004           +
                              b55f005           +
                              b55f017           +
                              b59f004           +
                              b65f001           +
                              b65f006           +
                              b70f006           ;
      label var hhindtax "Household indirect taxes";

*Household savings*
;
      gen hhsav       = b07f007           +
                        b07f018           +
                        b69f003           +
                        b69f004           +
                        b69f005           +
                        b69f006           +
                        b69f008           +
                        b69f009           +
                        b69f010           +
                        b69f011           +
                        b69f012           +
                        b69f013           +
                        b69f014           +
                        b69f015           +
                        b69f016           +
                        b69f017           +
                        b69f018           +
                        b71f001           ;

      label var hhsav "Household savings";

*Other - these totals are ignored for purposes of calculating hh
expenditure ratios and therefore are not included in the calculated total
per household category ;

      gen hhother     = b67f006           +
                        b70f007           +
                        b70f008           +
                        b89f063           +
                        b89f064           ;

      label var hhother "Other expenditures";
```

6.8. Excat96.do

```
#delimit;

*96 expenditure categories based on those used by Stats SA in their
national SAM. Note that they originally defined 95 categories (see
134_96mapping.xls for details) but payments for domestic services are now
included and renamed p96. ;
```

```
gen p01 = excat001;
gen p02 = excat002;
gen p03 = excat003;
gen p04 = excat004 + excat005+ excat006;
gen p05 = excat007;
gen p06 = excat008;
gen p07 = excat009;
gen p08 = excat010;
gen p09 = excat011;
gen p10 = excat012 + excat013;
gen p11 = excat014;
gen p12 = excat015;
gen p13 = excat016;
gen p14 = excat017;
gen p15 = excat018 + excat019;
gen p16 = excat020 + excat021 +excat022 + excat023;
gen p17 = excat024 + excat025;
gen p18 = excat026;
gen p19 = excat027;
gen p20 = excat028 + excat029;
gen p21 = excat030;
gen p22 = excat031;
gen p23 = excat032 +excat033;
gen p24 = excat034 ;
gen p25 = excat035;
gen p26 = excat036;
gen p27 = excat037;
gen p28 = excat038;
gen p29 = excat039;
gen p30 = excat040;
gen p31 = excat041;
gen p32 = excat042 + excat043;
gen p33 = excat044;
gen p34 = excat045;
gen p35 = excat046;
gen p36 = excat047;
gen p37 = excat048;
gen p38 = excat049;
gen p39 = excat050;
gen p40 = excat051 + excat052;
gen p41 = excat053;
gen p42 = excat054;
gen p43 = excat055;
gen p44 = excat056;
gen p45 = excat057;
gen p46 = excat058;
gen p47 = excat059;
gen p48 = excat060;
gen p49 = excat061;
gen p50 = excat062;
gen p51 = excat063;
gen p52 = excat064 + excat065;
gen p53 = excat066;
gen p54 = excat067;
gen p55 = excat068;
gen p56 = excat069;
gen p57 = excat070;
gen p58 = excat072;
gen p59 = excat071 + excat073;
gen p60 = excat074;
gen p61 = excat075 + excat076;
gen p62 = excat077;
gen p63 = excat078;
gen p64 = excat079 + excat080 + excat081;
gen p65 = excat082;
gen p66 = excat083;
```

```
gen p67 = excat084;
gen p68 = excat085;
gen p69 = excat086;
gen p70 = excat087;
gen p71 = excat088;
gen p72 = excat089;
gen p73 = excat090;
gen p74 = excat091;
gen p75 = excat092;
gen p76 = excat093 + excat094;
gen p77 = excat095 + excat096 + excat097 + excat098;
gen p78 = excat099;
gen p79 = excat100;
gen p80 = excat101 + excat102;
gen p81 = excat103;
gen p82 = excat104;
gen p83 = excat105 + excat106;
gen p84 = excat107;
gen p85 = excat108 + excat109 + excat110 + excat111 + excat112 + excat113 +
excat114 + excat115;
gen p86 = excat116 + excat117;
gen p87 = excat118;
gen p88 = excat119;
gen p89 = 0;
gen p90 = excat120;
gen p91 = excat121;
gen p92 = excat122 + excat123 + excat124 + excat125;
gen p93 = excat126 + excat127;
gen p94 = excat128;
gen p95 = excat129 + excat130+ excat131 + excat132 + excat133+ excat134;
gen p96 = hhdomser;

gen extot134 =          hhtotals + hhinctax + hhindtax + hhsav + hhother +
p96+

excat001 +excat002 +excat003 + excat004 + excat005 + excat006 + excat007 +
excat008 + excat009 + excat010 + excat011 + excat012 + excat013 + excat014
+ excat015 + excat016 + excat017 + excat018 + excat019 + excat020 +
excat021 + excat022 + excat023 + excat024 + excat025 + excat026 + excat027
+ excat028 + excat029 + excat030 + excat031 + excat032 + excat033 +
excat034 + excat035 + excat036 + excat037 + excat038 + excat039 +excat040 +
excat041 + excat042 +excat043 + excat044 + excat045 + excat046 + excat047 +
excat048 + excat049 + excat050 + excat051 + excat052 + excat053 + excat054
+ excat055 + excat056 + excat057 + excat058 + excat059 + excat060 +
excat061 + excat062 + excat063 + excat064 + excat065 + excat066 + excat067
+ excat068 + excat069 + excat070 + excat071 + excat072 + excat073 +
excat074 + excat075 + excat076 + excat077 + excat078 + excat079 + excat080
+ excat081 + excat082 + excat083 + excat084 + excat085 + excat086 +
excat087 + excat088 + excat089 + excat090 + excat091 + excat092 + excat093
+ excat094 + excat095 + excat096 + excat097 + excat098 + excat099 +
excat100 + excat101 + excat102 + excat103 + excat104 + excat105 + excat106
+ excat107 + excat108 + excat109 + excat110 + excat111 + excat112 +
excat113 + excat114 + excat115 + excat116 + excat117 + excat118 + excat119
+ excat120 + excat121 + excat122 + excat123 + excat124 + excat125 +
excat126 + excat127 + excat128 + excat129 + excat130 + excat131 + excat132
+ excat133 + excat134;

label var extot134 "Sumtotal of 134 excats and other";

gen extot96 =     hhtotals + hhinctax + hhindtax + hhsav + hhother +
                  p01+ p02+ p03+ p04+ p05+ p06+ p07+ p08+
                  p09+ p10+ p11+ p12+ p13+ p14+ p15+ p16+
                  p17+ p18+ p19+ p20+ p21+ p22+ p23+ p24+
                  p25+ p26+ p27+ p28+ p29+ p30+ p31+ p32+ p33+
                  p34+ p35+ p36+ p37+ p38+ p39+ p40+ p41+
                  p42+ p43+ p44+ p45+ p46+ p47+ p48+ p49+
```

```
                  p50+ p51+ p52+ p53+ p54+ p55+ p56+ p57+
                  p58+ p59+ p60+ p61+ p62+ p63+ p64+ p65+
                  p66+ p67+ p68+ p69+ p70+ p71+ p72+ p73+
                  p74+ p75+ p76+ p77+ p78+ p79+ p80+ p81+
                  p82+ p83+ p85+ p84+ p86+ p87+ p88+ p89+
                  p90+ p91+ p92+ p93+ p94+ p95+ p96;

label var extot96 "Sumtotal of 96 excats and other";
```

6.9. Diffs.do

```
#delimit;

* Checks all reported totals with calculated totals. Variables for
calculated totals start with a t*. The difference between these calculated
and reported totals are calculated. Difference variables start with a d*.
All d*-variables should theoretically be equal to zero (see
ies95check.log), although rounding errors may account for small
differences. ;

gen t07f100 = b07f001 + b07f004 +b07f005 +b07f006
              +b07f007 +b07f008 +b07f010 +b07f011 +b07f012
              +b07f013 +b07f014 +b07f015 +b07f016 + vatdiff;

gen d07f100 = t07f100 - b07f100;

gen t07f200 = b07f018 +b07f019 +b07f020 +b07f021 +b07f022
             +b07f023 +b07f024 +b07f025 +b07f026 +b07f027 ;

gen d07f200 = t07f200 - b07f200;

gen t10f100 = b10f001 + b10f002 + b10f003 + b10f004 + b10f005 + b10f006 +
b10f007;

gen d10f100 = t10f100 - b10f100;

gen t15f100 = b15f001+ b15f002+ b15f003+ b15f004+ b15f005+ b15f006+
b15f007+ b15f008+
              b15f009+ b15f010+ b15f011+ b15f012+ b15f013+ b15f014+
b15f015;

gen d15f100 = t15f100-b15f100;

gen t17f100 = b16f001+ b16f002+ b16f003+ b16f004+ b16f005+ b17f006+
b17f007+ b17f008+
              b17f009+ b17f010+ b17f011+ b17f012+ b17f013+ b17f014+
              b17f015+ b17f016+ b17f017;

gen d17f100 = t17f100-b17f100;

gen t18f100 = b18f001+ b18f002+ b18f003+ b18f004+ b18f005;

gen d18f100 = t18f100-b18f100;

gen t19f100 = b19f001+ b19f002+ b19f003+ b19f004+ b19f005;

gen d19f100 = t19f100-b19f100;

gen t20f100 = b20f001+ b20f002+ b20f003+  b20f004+ b20f005+ b20f006+
b20f007+
              b20f008+ b20f009+ b20f010+ b20f011;

gen d20f100 = t20f100-b20f100;
```

```
gen t21f100 = b21f001+ b21f002+ b21f003+ b21f004+ b21f005+ b21f006+
b21f007+
              b21f008+ b21f009+ b21f010+ b21f011+ b21f012+ b21f013+
b21f014+
              b21f015+ b21f016+ b21f017+ b21f018+ b21f019+ b21f020+
b21f021;

gen d21f100 = t21f100-b21f100;

gen t22f100 = b22f001+ b22f002+ b22f003+ b22f004+ b22f005+ b22f006+
b22f007+ b22f008+ b22f009+
              b22f010+ b22f011+ b22f012+ b22f013;

gen d22f100 = t22f100-b22f100;

gen t23f100 = b23f001+ b23f002+ b23f003+ b23f004;

gen d23f100 = t23f100-b23f100;

gen t24f100 = b24f001+ b24f002+ b24f003+ b24f004+ b24f005+ b24f006;

gen d24f100 = t24f100-b24f100;

gen t25f100 = b25f001+ b25f002+ b25f003+ b25f004+ b25f005;

gen d25f100 = t25f100-b25f100;

gen t26f100 = b26f001+ b26f002+ b26f003+ b26f004+ b26f005+ b26f006+
b26f007+ b26f008+ b26f009+
              b26f010+ b26f011;

gen d26f100 = t26f100-b26f100;

gen t27f100 = b27f001+ b27f002+ b27f003;

gen d27f100 = t27f100-b27f100;

gen t27f200 = t15f100 +t17f100 +t18f100 +t19f100 +t20f100
              +t21f100 +t22f100 +t23f100 +t24f100 +t25f100
              +t26f100 +t27f100;

gen d27f200 = t27f200 - b27f200;

gen t28f100 = b28f001+ b28f002+ b28f003+ b28f004;

gen d28f100 = t28f100-b28f100;

gen t29f100 = b29f001+ b29f002+ b29f003+ b29f004;

gen d29f100 = t29f100-b29f100;

gen t30f100 = b30f001+ b30f002+ b30f003+ b30f004+ b30f005+ b30f006+
b30f007;

gen d30f100 = t30f100-b30f100;

gen t31f100 = b31f001+ b31f002+ b31f003+ b31f004+ b31f005+ b31f006+
b31f007;

gen d31f100 = t31f100-b31f100;

gen t32f100 = b32f001+ b32f002+ b32f003+ b32f004;

gen d32f100 = t32f100-b32f100;
```

```
gen t33f100 = b33f001+ b33f002+ b33f003+ b33f004+ b33f005+ b33f006+
b33f007+ b33f008+
              b33f009+ b33f010+ b33f011+ b33f012+ b33f013+ b33f014+
b33f015+ b33f016+
              b33f017+ b33f018;

gen d33f100 = t33f100-b33f100;

gen t34f100 = b34f001+ b34f002+ b34f003+ b34f004+ b34f005+ b34f006+
b34f007+ b34f008+
              b34f009+ b34f010;

gen d34f100 = t34f100-b34f100;

gen t35f100 = b35f001+ b35f002+ b35f003+ b35f004+ b35f005+ b35f006;

gen d35f100 = t35f100-b35f100;

gen t36f100 = b36f001+ b36f002+ b36f003+ b36f004+ b36f005+ b36f006+
b36f007+
              b36f008+ b36f009+ b36f010+ b36f011+ b36f012;

gen d36f100 = t36f100-b36f100;

gen t37f100 = b37f001+ b37f002+ b37f003+ b37f004+ b37f005+ b37f006+
b37f007+
              b37f008+ b37f009+ b37f010;

gen d37f100 = t37f100-b37f100;

gen t38f100 = b38f001+ b38f002+ b38f003+ b38f004+ b38f005+ b38f006+
b38f007+ b38f008+
              b38f009+ b38f010;

gen d38f100 = t38f100-b38f100;

gen t39f100 = b39f001+ b39f002+ b39f003+ b39f004+ b39f005+ b39f006+
b39f007+ b39f008+
              b39f009+ b39f010;

gen d39f100 = t39f100-b39f100;

gen t40f100 = b40f001+ b40f002+ b40f003+ b40f004+ b40f005+ b40f006+
b40f007+ b40f008+
              b40f009+ b40f010;

gen d40f100 = t40f100-b40f100;

gen t41f100 = b41f001+ b41f002+ b41f003;

gen d41f100 = t41f100-b41f100;

gen t42f100 = b42f001+ b42f002+ b42f003+ b42f004+ b42f005+ b42f006;

gen d42f100 = t42f100-b42f100;

gen t43f100 = b43f001+ b43f002+ b43f003+ b43f004+ b43f005+ b43f006;

gen d43f100 = t43f100-b43f100;

gen t44f100 = b44f001+ b44f002+ b44f003+ b44f004+ b44f005+ b44f006;

gen d44f100 = t44f100-b44f100;

gen t45f100 = b45f001+ b45f002+ b45f003+ b45f004+ b45f005+ b45f006;
```

```
gen d45f100 = t45f100-b45f100;

gen t46f100 = b46f001+ b46f002+ b46f003+ b46f004+ b46f005;

gen d46f100 = t46f100-b46f100;

gen t47f100 = b47f001+ b47f002+ b47f003+ b47f004+ b47f005+ b47f006+
b47f007+ b47f008+
              b47f009+ b47f010+ b47f011;

gen d47f100 = t47f100-b47f100;

gen t48f100 = b48f001+ b48f002+ b48f003+ b48f004+ b48f005+ b48f006+
b48f007+ b48f008;

gen d48f100 = t48f100-b48f100;

gen t49f100 = b49f001+ b49f002+ b49f003+ b49f004+ b49f005+ b49f006+
b49f007+ b49f008+
              b49f009+ b49f010+ b49f011+ b49f012+ b49f013+ b49f014+
b49f015+ b49f016+
              b49f017+ b49f018+ b49f019;

gen d49f100 = t49f100-b49f100;

gen t50f100 = b50f001+ b50f002+ b50f003+ b50f004+ b50f005;

gen d50f100 = t50f100-b50f100;

gen t51f100 = b51f001+ b51f002+ b51f003+ b51f004+ b51f005+ b51f006+
b51f007;

gen d51f100 = t51f100-b51f100;

gen t52f100 = b52f001+ b52f002+ b52f003+ b52f004+ b52f005+ b52f006+
b52f007+ b52f008;

gen d52f100 = t52f100-b52f100;

gen t53f100 = b53f001+ b53f002+ b53f003+ b53f004+ b53f005;

gen d53f100 = t53f100-b53f100;

gen t54f100 = b54f001+ b54f002+ b54f003+ b54f004+ b54f005;

gen d54f100 = t54f100-b54f100;

gen t55f100 = b55f001+ b55f002+ b55f003+ b55f004+ b55f005+ b55f006+
b55f007+ b55f008+
              b55f009+ b55f010+ b55f011+ b55f012+ b55f013+ b55f014+
b55f015+ b55f016+
              b55f017+ b55f018+ b55f019;

gen d55f100 = t55f100-b55f100;

gen t56f100 = b56f001+ b56f002+ b56f003+ b56f004+ b56f005+ b56f006+
b56f007;

gen d56f100 = t56f100-b56f100;

gen t57f100 = b57f001+ b57f002+ b57f003+ b57f004+ b57f005+ b57f006+
b57f007;

gen d57f100 = t57f100-b57f100;

gen t58f100 = b58f001+ b58f002+ b58f003;
```

```
gen d58f100 = t58f100-b58f100;

gen t59f100 = b59f001+ b59f002+ b59f003+ b59f004+ b59f005+ b59f006+
b59f007+ b59f008;

gen d59f100 = t59f100-b59f100;

gen t60f100 = b60f001+ b60f002+ b60f003+ b60f004+ b60f005+ b60f006+
b60f007+ b60f008+
              b60f009+ b60f010+ b60f011;

gen d60f100 = t60f100-b60f100;

gen t61f100 = b61f001+ b61f002+ b61f003+ b61f004+ b61f005+ b61f006+
b61f007+ b61f008+
              b61f009+ b61f010+ b61f011;

gen d61f100 = t61f100-b61f100;

gen t62f100 = b62f001+ b62f002+ b62f003+ b62f004+ b62f005;

gen d62f100 = t62f100-b62f100;

gen t63f100 = b63f001+ b63f002+ b63f003+ b63f004+ b63f005+ b63f006+
b63f007+ b63f008+
              b63f009+ b63f010;

gen d63f100 = t63f100-b63f100;

gen t64f100 = b64f001+ b64f002+ b64f003+ b64f004+ b64f005+ b64f006;

gen d64f100 = t64f100-b64f100;

gen t65f100 = b65f001+ b65f002+ b65f003+ b65f004+ b65f005+ b65f006+
b65f007+ b65f008+
              b65f009+ b65f010+ b65f011;

gen d65f100 = t65f100-b65f100;

gen t66f100 = b66f001+ b66f002+ b66f003+ b66f004;

gen d66f100 = t66f100-b66f100;

gen t67f100 = b67f001+ b67f002+ b67f003+ b67f004+ b67f005+ b67f006+
b67f007+ b67f008;

gen d67f100 = t67f100-b67f100;

gen t68f100 = b68f001+ b68f002 - b68f003;

gen d68f100 = t68f100-b68f100;

gen t69f100 = b69f001+ b69f002+ b69f003+ b69f004+ b69f005+ b69f006+
b69f007+ b69f008+
              b69f009+ b69f010+ b69f011+ b69f012+ b69f013+ b69f014+
b69f015+ b69f016+
              b69f017+ b69f018;

gen d69f100 = t69f100-b69f100;

gen t70f100 = b70f001+ b70f002+ b70f003+ b70f004+ b70f005+ b70f006+ b70f007
+ b70f008;

gen d70f100 = t70f100-b70f100;
```

6.10. Inccat.do

```
#delimit;
set more off;

*Creating income categories based on source of income. See income.xls for
details. Although this .xls file relates to Western Cape, the same mapping
was used. ;


gen        inclab    =
                b83f001        +
                b83f002        +
                b83f003        +
                b83f004        +
                b84f001        +
                b84f002        +
                b84f003        +
                b84f004        +
                b85f001        +
                b85f002        +
                b85f003        +
                b85f004        +
                b86f001        +
                b86f002        +
                b86f003        +
                b86f004        +
                b87f001        +
                b87f002        +
                b87f003        +
                b87f004        +
                b88f005        +
                b88f006        +
                b88f007        +
                b88f008        +
                b88f009        ;

      label var inclab "Income from labour services";

gen         incgos =
                b83f005        +
                b83f006        +
                b84f005        +
                b84f006        +
                b85f005        +
                b85f006        +
                b86f005        +
                b86f006        +
                b87f005        +
                b87f006        +
                b88f001        ;


      label var incgos "Income from gross operating surplus";


gen        inctrans  =
                b83f016        +
                b83f017        +
                b84f016        +
                b84f017        +
                b85f016        +
                b85f017        +
                b86f016        +
                b86f017        +
```

```
                b87f016        +
                b87f017        +
                0.5*b88f021   +
                0.5*b88f022   +
                0.5*b88f023   +
                0.5*b88f024   +
                0.5*b88f025   +
                b88f026       ;

      label var inctrans "Inter-household transfers";

gen        inccorp =
                b83f007        +
                b83f008        +
                b83f009        +
                b83f010        +
                b83f011        +
                b84f007        +
                b84f008        +
                b84f009        +
                b84f010        +
                b84f011        +
                b85f007        +
                b85f008        +
                b85f009        +
                b85f010        +
                b85f011        +
                b86f007        +
                b86f008        +
                b86f009        +
                b86f010        +
                b86f011        +
                b87f007        +
                b87f008        +
                b87f009        +
                b87f010        +
                b87f011        +
                b88f002        +
                b88f003        +
                b88f004        +
                b88f010        +
                b88f011        +
                b88f013        +
                b88f014        +
                b88f015        +
                b88f016        +
                b88f017        ;


      label var inccorp "Income from corporations";

gen         incgov    =
                b83f012        +
                b83f013        +
                b83f014        +
                b83f015        +
                b84f012        +
                b84f013        +
                b84f014        +
                b84f015        +
                b85f012        +
                b85f013        +
                b85f014        +
                b85f015        +
                b86f012        +
                b86f013        +
```

```
                b86f014        +
                b86f015        +
                b87f012        +
                b87f013        +
                b87f014        +
                b87f015        +
                b88f012        +
                b88f020        +
                0.5*b88f021    +
                0.5*b88f022    +
                0.5*b88f023    +
                0.5*b88f024    +
                0.5*b88f025    ;

      label var incgov "Transfers from central government";

gen        incother  =
                b88f018        +
                b88f019        +
                b88f027        +
                b88f028        ;

      label var incother "Other income";

gen inctot = inclab + incgos + inctrans + inccorp + incgov + incother;
label var inctot "Total annual income";
sum inctot b89f068;
```

6.11. Totals.do

```
gen totals =      b07f100 +   b07f200 +   b10f100 +
                  b15f100 +   b17f100 +   b18f100 +
                  b19f100 +   b20f100 +   b21f100 +
                  b22f100 +   b23f100 +   b24f100 +
                  b25f100 +   b26f100 +   b27f100 +
                  b28f100 +   b29f100 +   b30f100 +
                  b31f100 +   b32f100 +   b33f100 +
                  b34f100 +   b35f100 +   b36f100 +
                  b37f100 +   b38f100 +   b39f100 +
                  b40f100 +   b41f100 +   b42f100 +
                  b42f200 +   b43f100 +   b44f100 +
                  b45f100 +   b46f100 +   b47f100 +
                  b48f100 +   b49f100 +   b50f100 +
                  b51f100 +   b52f100 +   b53f100 +
                  b54f100 +   b55f100 +   b56f100 +
                  b57f100 +   b58f100 +   b59f100 +
                  b60f100 +   b61f100 +   b62f100 +
                  b63f100 +   b64f100 +   b65f100 +
                  b66f100 +   b67f100 +   b68f100 +
                  b69f100 +   b70f100 +   b71f001 +
                  b89f063 +   b89f064    ;

gen totdiff = b89f065-totals;

label var totdiff "Matching survey totals with summary-page totals";
```

6.12. Coeffs.do

```
#delimit;

*Income and expenditure coefficients: Income side uses inctot (formerly
b89f068) as control total, while expenditure side uses extot (formerly
b89f065) as control total. ;
```

```
gen         cinclab    =        inclab        /     inctot            ;
gen         cincgos    =        incgos        /     inctot            ;
gen         cinctran   =        inctrans      /     inctot            ;
gen         cinccorp   =        inccorp       /     inctot            ;
gen         cincgov    =        incgov        /     inctot            ;
gen         cincothe   =        incother      /     inctot            ;

gen        chhtotal     =        hhtotals     /     extot             ;
gen        chindtax    =        hhindtax      /     extot             ;
gen        chinctax    =        hhinctax      /     extot             ;
gen        chhsav      =        hhsav         /     extot             ;
gen        chhother    =        hhother       /     extot             ;

gen        cp01        =        p01        /        extot               ;
gen        cp02        =        p02        /        extot               ;
gen        cp03        =        p03        /        extot               ;
gen        cp04        =        p04        /        extot               ;
gen        cp05        =        p05        /        extot               ;
gen        cp06        =        p06        /        extot               ;
gen        cp07        =        p07        /        extot               ;
gen        cp08        =        p08        /        extot               ;
gen        cp09        =        p09        /        extot               ;
gen        cp10        =        p10        /        extot               ;
gen        cp11        =        p11        /        extot               ;
gen        cp12        =        p12        /        extot               ;
gen        cp13        =        p13        /        extot               ;
gen        cp14        =        p14        /        extot               ;
gen        cp15        =        p15        /        extot               ;
gen        cp16        =        p16        /        extot               ;
gen        cp17        =        p17        /        extot               ;
gen        cp18        =        p18        /        extot               ;
gen        cp19        =        p19        /        extot               ;
gen        cp20        =        p20        /        extot               ;
gen        cp21        =        p21        /        extot               ;
gen        cp22        =        p22        /        extot               ;
gen        cp23        =        p23        /        extot               ;
gen        cp24        =        p24        /        extot               ;
gen        cp25        =        p25        /        extot               ;
gen        cp26        =        p26        /        extot               ;
gen        cp27        =        p27        /        extot               ;
gen        cp28        =        p28        /        extot               ;
gen        cp29        =        p29        /        extot               ;
gen        cp30        =        p30        /        extot               ;
gen        cp31        =        p31        /        extot               ;
gen        cp32        =        p32        /        extot               ;
gen        cp33        =        p33        /        extot               ;
gen        cp34        =        p34        /        extot               ;
gen        cp35        =        p35        /        extot               ;
gen        cp36        =        p36        /        extot               ;
gen        cp37        =        p37        /        extot               ;
gen        cp38        =        p38        /        extot               ;
gen        cp39        =        p39        /        extot               ;
gen        cp40        =        p40        /        extot               ;
gen        cp41        =        p41        /        extot               ;
gen        cp42        =        p42        /        extot               ;
gen        cp43        =        p43        /        extot               ;
gen        cp44        =        p44        /        extot               ;
gen        cp45        =        p45        /        extot               ;
gen        cp46        =        p46        /        extot               ;
gen        cp47        =        p47        /        extot               ;
gen        cp48        =        p48        /        extot               ;
gen        cp49        =        p49        /        extot               ;
gen        cp50        =        p50        /        extot               ;
gen        cp51        =        p51        /        extot               ;
gen        cp52        =        p52        /        extot               ;
gen        cp53        =        p53        /        extot               ;
```

```
gen        cp54       =        p54        /        extot                ;
gen        cp55       =        p55        /        extot                ;
gen        cp56       =        p56        /        extot                ;
gen        cp57       =        p57        /        extot                ;
gen        cp58       =        p58        /        extot                ;
gen        cp59       =        p59        /        extot                ;
gen        cp60       =        p60        /        extot                ;
gen        cp61       =        p61        /        extot                ;
gen        cp62       =        p62        /        extot                ;
gen        cp63       =        p63        /        extot                ;
gen        cp64       =        p64        /        extot                ;
gen        cp65       =        p65        /        extot                ;
gen        cp66       =        p66        /        extot                ;
gen        cp67       =        p67        /        extot                ;
gen        cp68       =        p68        /        extot                ;
gen        cp69       =        p69        /        extot                ;
gen        cp70       =        p70        /        extot                ;
gen        cp71       =        p71        /        extot                ;
gen        cp72       =        p72        /        extot                ;
gen        cp73       =        p73        /        extot                ;
gen        cp74       =        p74        /        extot                ;
gen        cp75       =        p75        /        extot                ;
gen        cp76       =        p76        /        extot                ;
gen        cp77       =        p77        /        extot                ;
gen        cp78       =        p78        /        extot                ;
gen        cp79       =        p79        /        extot                ;
gen        cp80       =        p80        /        extot                ;
gen        cp81       =        p81        /        extot                ;
gen        cp82       =        p82        /        extot                ;
gen        cp83       =        p83        /        extot                ;
gen        cp84       =        p84        /        extot                ;
gen        cp85       =        p85        /        extot                ;
gen        cp86       =        p86        /        extot                ;
gen        cp87       =        p87        /        extot                ;
gen        cp88       =        p88        /        extot                ;
gen        cp89       =        p89        /        extot                ;
gen        cp90       =        p90        /        extot                ;
gen        cp91       =        p91        /        extot                ;
gen        cp92       =        p92        /        extot                ;
gen        cp93       =        p93        /        extot                ;
gen        cp94       =        p94        /        extot                ;
gen        cp95       =        p95        /        extot                ;
gen        cp96       =        p96        /        extot                ;
```

Technical Papers in this Series

Number	Title	Date
TP2003: 1	Creating a 1995 IES Database in STATA	September 2003
TP2003: 2	Creating a 1995 OHS and a combined OHS-IES Database in STATA	September 2003
TP2003: 3	A Standard Computable General Equilibrium Model Version 3: Technical Documentation	September 2003
TP2003: 4	Measures of Poverty and Inequality: A Reference paper	October 2003
TP2004: 1	SeeResults: A spreadsheet Application for the Analysis of CGE Model Results	November 2004
TP2004: 2	The Organisation of Trade Data for inclusion in Social Accounting Matrix	December 2004
TP2004: 3	Creating a 2000 IES Database in Stata	August 2004
TP2004: 4	Forming Representative Household Groups in a SAM	July 2004

Other PROVIDE Publications

Background Paper Series
Working Papers
Research Reports