Accurate Income Measurement for the Assessment of Public Policies (AIM-AP)
Contract no 028412.

Workpackage 3.1:
Selection of the countries and detailed description of the indirect tax legislation

Darragh Flannery, Philip Hayes, Stephen Hynes, Jason Loughrey, Cathal O'Donoghue
Rural Economy Research Centre, Teagasc

Bram De Rock, André Decoster
University of Leuven

Abstract: This document contains workpackage 3.1, the first deliverable of project 3 of the AIM-AP project (Contract no 028412). We report on the chosen budget and income surveys for the 5 selected European countries. Secondly, we also define indirect taxation for the purpose of this project and present a detailed description of this legislation for the selected countries.

I. INTRODUCTION

This document contains workpackage 3.1, the first deliverable of project 3 of the sixth framework project called *Accurate Income Measurement for the Assessment of Public Policies (AIM-AP);* Contract no 028412. Project 3 aims at the incorporation of the effects of indirect taxes, along with direct taxes and social benefits, in redistribution analysis. Workpackage 3.1 is a first step for achieving this and it has the following objectives[1]:

> *"**Objectives**: To identify and secures access to income and budget survey data for five European countries. To make a detailed description of the indirect tax system of each country distinguishing the different VAT rates and excise duties that apply in each country. Identification of the goods to which these rates are applied in the household budget survey's."*

[1] See Annex 1-"Description of Work" of *AIM-AP* (contract no 028412)

This work package fits into a European research programme that goes back to the early 1990's, focusing on the simulation and evaluation of national tax and social policy programmes from a cross-country comparative perspective. From this programme evolved EUROMOD a European, cross-country tax-benefit model financed by a variety of EU financed research programmes.[2] The primary focus has been on current income and status related policy instruments such as Income Taxation, Social Insurance Contributions, Family Benefits, Social Assistance Benefits, Unemployment Benefits, Housing Benefits.

An initial scoping exercise was carried out to model indirect taxation.[3] However given data restrictions, only a limited approach was feasible. In this project, we plan to undertake a more in-depth rigorous approach to understanding the mechanisms to include indirect taxation in EUROMOD.

Later deliverables will focus on the practical implementation of indirect taxation in a micro simulation model, such as:

- Harmonization of the budget and income surveys (workpackage 3.2)
- Linking the indirect tax rates and expenditure aggregates (work-package 3.3)
- Comparative analysis of imputation techniques (workpackage 3.4)
- Imputation of expenditures into the income data sets (workpackage 3.5)
- Simulation Methodology of changes in indirect and direct taxes (work-package 3.6)

The remainder of this document is structured as follows. Section 2 presents the countries and surveys that we will use in this project. In Section 3 we define what we will include in indirect taxes in this project and we also introduce our methodology for calculating tax rates on aggregate expenditure items. Section 4 contains the general structure of the indirect tax legislation for the selected countries.

Complementary to this document, we constructed an Excel file, *ind_taxes_at_exp _level.xls*, which links the indirect tax legislations of the different countries to the different expenditure items at the most detailed level of the respective budget surveys.

---

2 See Immervoll, O'Donoghue and Sutherland, 1999

3 See O'Donoghue et al., 2004

II. SELECTION OF COUNTRIES AND SURVEYS

Given the need for a detailed comparative analysis, attempting to harmonise definitions and methods across countries and researching the optimal methods for simulating indirect taxes in a model such as EUROMOD, we have decided to focus on a subset of five European countries.

> *"The selection of countries to be included is based on the availability and quality of the income data (for which the EUROMOD experience will be used), the availability and accessibility of survey data on consumption expenditures, and also on the desirability for variation in relative importance of indirect tax in tax revenues (VAT and excise duties).*[4]*"*

Based on preliminary information about the availability of data, expert contacts within the consortium and a wide enough variety in the indirect tax systems, our original aim was to include the following five countries: Belgium, Greece, Hungary, Ireland and the United Kingdom and to identify and assure access to income data and household budget survey data.

Following discussions at the March project meeting in Athens and subsequently between partners in the project, the initial 5 countries planned were deemed feasible. In addition they give us a good spread of countries, both from a geographical perspective and also relative to the importance of indirect taxes within the budgetary financing mechanism.

As EUROMOD primarily relies on income surveys as its base dataset, but indirect taxation requires expenditure surveys, the availability of quality surveys of both types were required. Secondly, in order to do simulations, by using EUROMOD, of changes of both direct and indirect taxes we focus on income surveys which are planned to be in EUROMOD. However at this moment, this latter issue is not completely clear for all countries. Therefore an initial decision has been made, which will be evaluated in workpackage 3.2 in relation to these datasets. Table 1 outlines the choice we have made so far.

4 See Annex 1-"Description of Work" of AIM-AP (contract no 028412)

TABLE 1: SELECTION OF COUNTRIES AND DATASETS

Country	Income Survey	Year	Expenditure survey	Year
Belgium	EU-SILC[5]	2004	Household Budget Survey	2001
Greece	Household Budget Survey	2004/2005	Household Budget Survey	2004/2005
Hungary	EU-SILC	2004	Household Budget Survey	2004
Ireland	EU-SILC	2004	Household Budget Survey	1999/2000
United Kingdom	Family Resources Survey	2003	Family Expenditures Survey	2000/2001

At the moment of writing there are plans to include some of the income surveys mentioned in Table 1 into EUROMOD. This might be the case for Greece (which in fact is a household budget survey), Hungary (EU-SILC) and the United Kingdom (the Family Resources Survey). To be precise, the choice of the Hungarian survey to be used in EUROMOD is still pending, and in case EU-SILC would not be available or deemed appropriate, the fall back option will be the Household Budget Survey 2004. At this moment there are no concrete plans as far as a EUROMOD–update of the income surveys for Belgium and Ireland is concerned. Nevertheless we opt to use the EU-SILC data for both cases since, based on discussions within the consortium, we believe that in the near future there might be an update of EUROMOD for both these countries with EU-SILC data. Moreover, using the same income survey (i.e. EU-SILC) for as many countries as possible is convenient since one of the objectives of the AIM-AP project is to do uniform simulations for the different countries in order to able to compare them.

The choice of the budget surveys is mainly based on current availability. At this moment we have access to all the expenditures surveys mentioned in Table 1 and also to the Family Resources Survey. On the Leuven meeting in December 2006, the AIM-AP participants decided to ask for access to the EU-SILC data.

In Table 2 and 3 we present some details concerning the income and budget surveys. All surveys use households as observation unit.

5 European Union Statistics on Income and Living Conditions

TABLE 2: INCOME SURVEYS DETAILS

Country	Income Survey	Year	#Households	Institution
Belgium	EU-SILC	2004	5275	Statistics Belgium
Greece	Household Budget Survey	2004/2005	6555	National Statistics Service of Greece
Hungary	EU-SILC	2004	7000	Hungarian Central Statistical Office
Ireland	EU-SILC	2004	5477	Economic and Social Research Institute
United Kingdom	Family Resources Survey	2003	7048	Department for Work and Pensions Office for National Statistics. Social Survey Division National Centre for Social Research

TABLE 3: BUDGET SURVEYS-DETAILS

Country	Expenditure survey	Year	#Households	# exp. items	Institution
Belgium	Household Budget Survey	2001	3698	974	Statistics Belgium
Greece	Household Budget Survey	2004/2005	6555	502	National Statistics Service of Greece
Hungary	Household Budget Survey	2004	8710	453	Hungarian Central Statistical Office
Ireland	Household Budget Survey	1999/2000	7644	579	Central Statistics Office
United Kingdom	Family Expenditures Survey	2000/2001	6100	518	Office for National Statistics

III. INDIRECT TAXATION

We start by stating what we include in Indirect Taxation within this project. This is followed by the description of our methodology to calculate indirect tax rates for aggregates.

1. Definition of Indirect Taxation

An indirect tax is defined as[6]

> *"a tax levied indirectly, as one levied on commodities before they reach the consumer but ultimately paid by the consumer as part of the market price."*[7]

As such therefore the term can cover a plethora of different taxes from VAT, sales taxes, licences, stamp duties etc. We shall focus our definition on taxes that relate to the purchase of consumption goods, defined principally as:

- Valued Added Taxes (VAT);
- *specific* excise duties;
- *ad valorem* excise duties.

These taxes differ in the way they are calculated. Value Added Taxes (VAT) are levied on pre-tax expenditure, specific excise duties are levied on the physical quantity of a commodity and ad valorem excises are indirect taxes on the retail price of the good. In the next subsection we will formalize these differences in the way we calculate the indirect tax liabilities. All other indirect taxes are excluded from the analysis such as for instance firearm certificates, dog licenses, car licenses, indirect taxes relating to the transfer of agricultural produce or farm animals between farmers,…

2. Methodology

In this section we explain how we:

- solved the problem that we do not observe quantities, but only expenditures (hence the product of quantity bought times the retail price). For excises, defined on the physical quantities, we therefore needed to

---

[6] See,e.g., www.dictionary.com

[7] Needless to say that "*paid by the consumer*" in the above quotation makes abstraction from any analysis of incidence of indirect taxation.

translate the excise duty itself into an excise *rate*, to be applied, just like the VAT- and ad valorem rate on tax inclusive expenditures;

- defined tax rates on commodity aggregates;
- split total indirect taxes into an excise and a VAT component.

Where appropriate we will refer to lines in the (preliminary) STATA-program taken up in the appendix.

In what follows q_i denotes the consumer or retail price for commodity i, p_i the producer price, a_i the excise tax, v_i an ad valorem tax rate applied on the consumer price and t_i the VAT-rate.

Using this notation, we can write the tax-link between the producer price and the consumer price for commodity i as follows:

$$q_i = (1+t_i)(p_i + a_i + v_i \cdot q_i). \qquad (1)$$

Step 1: calculation of the producer price.

In simulations we will assume that producer prices are fixed. Hence, changes in the consumer price of (1) can only be induced by changes in the tax parameters, not in changes in producer prices. To implement this assumption of constant producer price in the simulation program, it is therefore convenient to express the producer price in function of the consumer price and the tax parameters. This is done by re-arranging (1) and solving for the producer price p_i, and gives:

$$p_i = \left[\frac{1-(1+t_i)v_i}{1+t_i}\right] q_i - a_i. \qquad (2)$$

Equation (2) is implemented for the baseline situation in line 39 of the program.[8] Note that consumer prices q_i in the baseline situation are needed. Yet, for commodities where there is no specific excise duty ($a_i = 0$), we can without loss of generality put the consumer prices equal to one. That is what is done in practice. For the commodities that are amenable to specific excise duties we have entered the retail price in the baseline in the input dataset.

---

[8] Note that the subtraction of the excise in equation 2 might for a given consumer price lead to negative producer prices. We therefore add in line 41 a "count" of these cases.

In line 100 of the program we then equate the "new" post reform produce price to the pre reform one, and in line 102 we use equation (1) again to calculate the consumer price in the post reform situation, i.e. for fixed producer price, but with new tax parameters.

Note that, in the most common case where the ad valorem rate is zero, (2) simplifies to:

$$p_i = \frac{q_i}{1+t_i} - a_i. \tag{3}$$

Step 2: constant quantities to determine the expenditures post reform

Since we will apply indirect tax rates on expenditures in baseline and reform situations, we need expenditures in the reform situation as well. In this stage of the project, where we only want to calculate implicit tax rates on aggregate commodities, we have opted to keep quantities constant. Denoting expenditures in the baseline on commodity *i* as e_i^0, where the superscript 0 refers to the baseline, we have:

$$e_i^0 = q_i^0 x_i^0, \tag{4}$$

where x_i^0 refers to the quantity consumed in the baseline. Hence in the reform situation, with new consumer prices q_i^1, and with the assumption $x_i^0 = x_i^1$, we easily derive that the reform expenditures will be:

$$e_i^1 = q_i^1 x_i^1 = q_i^1 x_i^0 = \frac{q_i^1}{q_i^0} e_i^0. \tag{5}$$

Equation (5) is implemented in line 153 of the STATA-code in appendix. We repeat that this does not imply that we will impose this assumption of constant quantities also in the behavioural modeling of the consumer or household. A this stage it is a (quite innocuous) assumption in aggregating price and tax information at the detailed level into aggregate commodities.

Step 3: implicit excise *rate*

For each commodity i, the *implicit proportional excise tax rate* α_i is defined as:

$$\alpha_i = \frac{a_i}{p_i}, \tag{6}$$

see line 156 of the STATA-code in appendix.[9]

Step 4: .decomposing the total indirect tax into a VAT and an excise component.

Replacing a_i in equation (1) by making use of (6), we obtain:

$$q_i = (1+t_i)\cdot(p_i + \alpha_i \cdot p_i + v_i \cdot q_i), \qquad (7)$$

which, after re-arrangement, can be written as:

---

[9] Up to now, we have chosen to calculate apply (6) for each excise commodity separately by calculating its producer price from (2). An alternative is to use the IFS-approach to calculate α_i from macro-economic revenue figures which relate the excise revenue for broad aggregates to *expenditures* on these broad aggregates. In order to derive α_i in terms of observable variables, IFS uses what they call the excise return ratio r_i, defined as:

$r_i = \frac{a_i}{q_i}$, where the important difference with (6) is in the denominator. Substitution of equations (6) and (8) leads to

$$r_i = \frac{\alpha_i \cdot p_i}{\frac{(1+t_i)\cdot(1+\alpha_i)}{1-v_i\cdot(1+t_i)}\cdot p_i} = \frac{\alpha_i\left[1-v_i(1+t_i)\right]}{(1+t_i)\cdot(1+\alpha_i)},$$

and hence to:

$$\alpha_i = \frac{r_i(1+t_i)}{1-(v_i+r_i)(1+t_i)}.$$

In words, the implicit proportional tax rate of commodity *i*, α_i, can be derived in terms of observable variables r_i (excise return ratio from macro budget figures), t_i and v_i. We have not explored the differences between the two approaches, although we have the feeling that using the macro-economic budget figures might solve - partially - the excises we are missing paid by the producer sector. However, the drawback of the more aggregate excise revenue figures from macro sources is the difficulty to relate changes in excise duties at the detailed commodity level to this approach.

$$q_i = \frac{(1+t_i)\cdot(1+\alpha_i)}{1-v_i\cdot(1+t_i)}\cdot p_i = z_i \cdot p_i, \tag{8}$$

where $z_i = \frac{(1+t_i)\cdot(1+\alpha_i)}{1-v_i\cdot(1+t_i)}$ is the ratio of consumer to producer price for commodity i, and hence related to the total tax rate τ_i on this commodity as follows:

$$\tau_i = z_i - 1 \tag{9}$$

$$= \frac{q_i - p_i}{p_i}$$

$$= \frac{t_i(1+\alpha_i+v_i)+v_i}{1-(1+t_i)v_i} + \frac{\alpha_i}{1-(1+t_i)v_i} \tag{10}$$

$$= \tau_i^t + \tau_i^a. \tag{11}$$

In (10) and (11) the total tax rate τ_i has now been decomposed into an implicit VAT-rate τ_i^t and an implicit excise tax rate τ_i^a. Note that VAT paid on the excise duty, and the ad valorem duty are taken together here as "VAT". Equations (10) and (11) are implemented in lines 157 to 160 of the program.

Step 5: calculating tax liabilities at the detailed commodity level

To calculate the tax liability we start from the observation that total indirect tax liability T_i equals the difference between expenditures at consumer and expenditures at producer prices:

$$\begin{aligned} T_i &= q_i x_i - p_i x_i \\ &= \frac{q_i - p_i}{p_i} p_i x_i \\ &= \tau_i p_i x_i. \end{aligned} \tag{12}$$

Equation (12) also reveals that the total tax rates τ_i and the two components τ_i^t and τ_i^a have all been expressed in relation to the producer price for the commodity. To apply them to observable expenditures $e_i = q_i x_i$, we switch from the tax exclusive tax base, to the tax inclusive one as follows:

$$T_i = \tau_i p_i x_i$$

$$= \frac{\tau_i}{q_i} p_i q_i x_i$$

$$= \frac{\tau_i}{z_i} e_i$$

$$= \frac{\tau_i}{1+\tau_i} e_i \qquad (13)$$

$$= \frac{\tau_i^t}{1+\tau_i} e_i + \frac{\tau_i^a}{1+\tau_i} e_i \qquad (14)$$

$$= T_i^t + T_i^a, \qquad (15)$$

where T_i^t and T_i^a refer to VAT and excise tax liability for commodity i respectively.

Equations (14) and (15) are applied on lines 162 and 163 of the STATA-program listed in the appendix.

Step 6: aggregating the tax liabilities into broader commodity aggregates and calculating implicit tax rates on the aggregates

All the above formulae are applied to individual commodities. However at a later stage in the project we shall define aggregates for our imputations and simulations. At this moment, it is not possible to determine the best level of aggregation. This could vary from simply grouping goods by indirect tax level to the broadest level of disaggregation simulating indirect taxes for each consumption good recorded in the data. Our choice will depend upon a number of criteria, including

- The optimal level of dissaggregation required to model indirect taxes from a comparative perspective;
- The requirements of the expenditure imputation mechanism selected;
- Usefulness of expenditure categories from a policy perspective.

The tax liability for commodity aggregate K, denoted by T_K, is obtained as the sum of the tax liabilities paid on the individual commodities:

$$T_K = \sum_{i \in K} T_i \tag{16}$$

$$= \sum_{i \in K} T_i^t + \sum_{i \in K} T_i^a, \tag{17}$$

from which the *implicit tax rates on the aggregates* are defined as:

$$T_K = \frac{T_K}{e_K - T_K} \tag{18}$$

$$= \frac{T_K^t}{e_K - T_K} + \frac{T_K^a}{e_K - T_K} \tag{19}$$

$$= \tau_K^t + \tau_K^a. \tag{20}$$

Note that also the tax rates on the aggregates are defined on a tax exclusive basis. Equations (19) and (20) are applied on lines 162 and 163 of the STATA-program listed in the appendix. It are these aggregates that we export to a file which can be used in later stages of the project and/or simulations.

Step 7: calculation of tax liabilities on the aggregates

Since the tax rates on the aggregates are on a tax exclusive basis, they can be used to calculate the tax liabilities on aggregate expenditures as follows:

$$T_K^t = \tau_K^t \left(e_K - T_K \right)$$

$$= \frac{\tau_K^t}{1 + \tau_K} e_K, \tag{21}$$

and

$$T_K^a = \tau_K^a \left(e_K - T_K \right)$$

$$= \frac{\tau_K^a}{1 + \tau_K} e_K. \tag{22}$$

IV. INDIRECT TAX LEGISLATION FOR THE SELECTED COUNTRIES

We start with a general structure of the indirect tax legislation of the selected countries and then give a brief historic overview concerning the VAT-rates.

1. General structure

We collected for the five countries as well the most recent indirect tax legislation (which is 2006) as the one for the year of the expenditure survey, listed in Table 1. This allows us to adapt in a later stadium our Stata files to a more recent legislation; which could be helpful when we simulate changes in both direct and indirect taxes.

Table 4 summarizes the VAT-rates for the five countries. The standard rates are quite close to each other. The variation across the countries mainly occurs in the reduced rate(s). In this respect, Ireland takes up a separate position, with a reduced rate of 13.5%, which is far above the one of the other four countries. The main change in indirect tax legislation between the year of the survey and the current legislation has occurred in Hungary. The standard rate has been lowered from 25 to 20%, the reduced rate even from 15 to 5 %. This opens the possibility to implement a retrospective policy relevant simulation for this newly accessed EU-country in the future of the project.

TABLE 4 : VAT- RATES

Country	VAT rates 2006				VAT rates year Expenditure Survey			
	Standard	Reduced			Standard	Reduced		
Belgium	21	12	6	0	21	12	6	0
Greece	19	9	4.5	0	18	8	4	0
Hungary	20	5		0	25	15	5	0
Ireland	21	13.5		0	20	12.5		0
United Kingdom	17.5	5		0	17.5	5		0

As expected, the excise duties differ a lot across the five countries. The tax base for excise duties however, i.e. the commodities on which an excise tax is levied, are more or less the same across the different countries. In summary, the excise products are:

- mineral oil products (gasoline, (un)leaded petrol,…);
- alcoholic products (spirits, beer, wine,…);
- tobacco products (cigarettes, cigars,…).

The same holds for the Ad Valorem tax which mostly concerns tobacco products.

2. Historic overview of VAT rates [10]

Belgium introduced VAT in 1971 and used 5 different rates: the reduced rates 0%, 6%, 14%; the standard rate 18% and the increased rate 25%. The increased rate disappeared after 10 years and an extra reduced rate of 1% was introduced. The latter was removed again in 1992. The standard rate steadily increased to the current 21 % while the reduced rates became 0%, 6% and 12%.

VAT was first introduced to *Greece* in 1987 as part of its accession to the European Community. The initial rates were 3%, 6%, 16% and 36%. Since 1988, the two reduced and the standard VAT rates have been increased to 4.5%, 9% and 19% respectively, while the top VAT rate has been abolished.

In *Hungary*, VAT was only introduced in 1988 and they applied a standard rate of 25% and reduced rates of 0% and 15%. During the years an extra reduced rate of 5% was introduced. In September 2006, Hungary decided to decrease the standard rate to 20% and to maintain two reduced rates (0 % and 5%).

Vat was introduced to *Ireland* in 1972 with rates of 0%, 5.26%, 16.37% and 30.26%. A special rate of 11.1% applied to dances and a 1% flat rate addition to unregistered farmers.[11] Since 1973, the highest rate and the special rate for dances have been abolished, the reduced rate increased to 13.5% and the standard rate increased to 21%. There was the short-lived introduction in 1992 of an additional reduced rate of 16%.

In the *United Kingdom* VAT was introduced in 1973 with a standard rate of 8% and an increased rate of 25%. A few years later the increased rate was removed and the standard rate increased to 15%. In 1995 a reduced rate of 8% was introduced. Currently the standard rate equals 17.5 % and the reduced rates are 5% and 0%.

---

[10] Based on *VAT rates applied in the member states of the European community*, version September 2006

[11] Irish farmers unregistered for VAT are compensated for the VAT charged on their produce by means of a 5.2% flat rate addition to the prices at which they sell their produce and services to VAT registered persons. This addition was only 1% upon its introduction in 1972. Another rate of 4.8% applies to the sale of farm animals by registered farmers. The same rate applied to the addition for unregistered farmers and the sale of produce by registered farmers until this year.

V. APPENDIX

1. Data

Complementary to this document, we constructed an Excel file, "**`ind_taxes_at_exp_level.xls`**", which links, at the most detailed level, the respective indirect tax legislation to the different expenditure items of the respective budget surveys.

In a later stadium, we will use this file to calculate the tax rates of the aggregates by using the methodology of Section 3. We cannot do this at this stage of the project since the choice of the aggregates is dependent of decisions we have to make in other workpackages (e.g. in order to optimize the matching exercise).

2. STATA program

See next pages with the listing of the code of the STATA-program "**`tax rates aggregates BE2001.do`**". This program calculates indirect tax rates on the Belgian budget survey of 2001, and summarizes them for both baseline and reform in the form of implicit tax rates on aggregates. This preliminary version of the program will be used to create a generic version of it which can be applied on the datasets for the other countries.

```
 1  clear
 2  set type double
 3  set mem 200m
 4  set matsize 800
 5  set more off
 6  capture log close
 7
 8  log using "C:\My\Onderzoek\Projecten\AIM-AP\Imputation Belgium\tax rates baseline.log", replace
 9
10  * In this program we
11  * 1) calculate the indirect taxes for 2005 on all individual items of the budgetsurvey 2001
12  *  and for all households separately in both a baseline (2005) and reform situation
13  *  and write them into a file
14  * 2) derive the implied tax rates and consumer prices for these aggregates and write them to a file
15  * 3) produce Revenue tables for the `nagg' aggregates
16
17  * DEFINE THE NUMBER OF AGGREGATES HERE (including durables)
18  local nagg=13
19  * GIVE PATH OF DATASET where the tax rates and consumer prices have to be saved for use in other programs
20  local path1="C:\My\Onderzoek\Projecten\AIM-AP\Imputation Belgium\reform01.dta"
21
22  ********************************************************************
23  * Part 1: define the tax system in baseline and reform              *
24  ********************************************************************
25  * retrieve the dataset with tax rates for the year 2005
26  use "C:\My\Onderzoek\Projecten\AIM-AP\Imputation Belgium\pre2005 aggr13.dta", clear
27
28  * create a temporary file with code and tax rates in notation of formula's and with the a reform
29  tempfile taxsystem
30  local deflator=0.9249 /* deflator to go from 2005 nominal level to 2001 nominal level*/
31       * rename the tax rates in the baseline of 2005 (with suffix 0)
32  rename btw t0        /*VAT rate pre reform*/
33  rename acc a0        /*excise per unit pre reform*/
34  rename adval v0      /*ad valorem rate pre reform*/
35  rename prijs q0      /*consumer price pre reform, is price in 2001*/
36  replace a0=a0*`deflator' /*deflate the excise of 2005, into 2001 values (but still tax system of 2005!)*/
37
38  * calculate producer price pre reform
39  gen p0=((1-(1+t0)*v0)/(1+t0))*q0-a0
40  label var p0 "producer price pre reform"
41  count if p0<0
42
43  ****************************************************************
44  *    DEFINE THE REFORM                                          *
```

```
45  * Default is to put all taxes equal to the 2005 situation          *
46  * The reform is introduced by overwriting them                     *
47  * To be discussed whether we will introduce a more user friendly   *
48  *   way of introducing reforms
49  ********************************************************************
50  gen t1=t0
51  gen a1=a0/`deflator'
52  gen v1=v0
53  label var t1 "VAT rate post reform"
54  label var a1 "excise post reform"
55  label var v1 "ad valorem rate post reform"
56
57  replace t1=0.25 if t0==0.21 /*replaces the VAT-rate of 21% with 22%*/
58  replace t1=0.07 if t0==0.06 /*replaces the VAT-rate of 6% with 4%*/
59
60  * Below: overwrite existing excises (excise for 2005 is in the comment at end of each line
61  *        replace the excise with a new value in 2005-prices. It will be deflated later
62  *replace a1=0 if code==121102 /*     Oploskoffie en koffieëxtract      .694 */
63  *replace a1=0 if code==121101 /*  Koffiebonen, gemalen kof., enz.      .248 */
64  *replace a1=0 if code==122107 /*        And. alcoholvrije dranken     .0372 */
65  *replace a1=0 if code==122103 /*         Alcoholvrije aperitieven     .0372 */
66  *replace a1=0 if code==123209 /*                 Alcoholvrije wijn     .0372 */
67  *replace a1=0 if code==123104 /*   Likeuren en stke dranken + 23°      5.96 */
68  *replace a1=0 if code==123103 /*                          Jenever      5.96 */
69  *replace a1=0 if code==123101 /*                           Whisky      7.01 */
70  *replace a1=0 if code==123199 /*                     Alcohol z.o.      5.96 */
71  *replace a1=0 if code==123102 /*                           Cognac      7.01 */
72  *replace a1=0 if code==123203 /*    Champagne en mousserende wijn      .161 */
73  *replace a1=0 if code==123201 /*                             Wijn      .471 */
74  *replace a1=0 if code==123202 /*                            Cider      1.61 */
75  *replace a1=0 if code==123204 /*    Likeurwijn e.a. drnk 15°- 23°      .992 */
76  *replace a1=0 if code==123303 /*                       Ander bier      .205 */
77  *replace a1=0 if code==123302 /*                   Bier type pils      .205 */
78  *replace a1=0 if code==123399 /*                        Bier z.o.      .205 */
79  *replace a1=0 if code==123309 /*   Alcoholvrij en alcoholarm bier     .0744 */
80  *replace a1=0 if code==123301 /*                        Tafelbier     .0684 */
81  *replace a1=0 if code==130103 /*      Sigaretten (20 st. per pak)    .00689 */
82  *replace a1=0 if code==130105 /*  Sigaretten (10 of 15 st. p/pak)    .00689 */
83  *replace a1=0 if code==130199 /*                        Sigaretten    .00689 */
84  *replace a1=0 if code==130104 /*      Sigaretten (25 st. per pak)    .00689 */
85  *replace a1=0 if code==322101 /*                          Aardgas   .000322 */
86  *replace a1=0 if code==322102 /*    Butaangas (zonder statiegeld)      .017 */
87  *replace a1=0 if code==322103 /*   Propaangas (zonder statiegeld)     .0174 */
88  *replace a1=0 if code==321101 /*                Elektriciteit    .00191 */
```

```
 89  *replace a1=0 if code==323102 /*    Andere vloeibare brandstoffen     .0179 */
 90  *replace a1=0 if code==323101 /*   Gasoil, fuel en collect. verw.     .0171 */
 91  *replace a1=0 if code==622106 /*                   Tweetakt olie      .488 */
 92  *replace a1=0 if code==622199 /*                    Benzine z.o.      .565 */
 93  *replace a1=0 if code==622101 /*    Spr. benz. zn. lood (Sup. 95)     .592 */
 94  *replace a1=0 if code==622103 /*                Benzine met lood      .552 */
 95  *replace a1=0 if code==622102 /*   Spr. benz. zn. lood (Euro+ 98)     .607 */
 96  *replace a1=0 if code==622105 /*                          Diesel      .349 */
 97  replace a1=a1*`deflator' /*deflate the reformed excise of 2005, into 2001 values*/
 98
 99  * calculate the new consumer prices after the reform (Ass: fixed producer prices)
100  gen p1=p0
101  label var p1 "producer price post reform"
102  gen q1=((1+t1)*(p1+a1))/(1-(1+t1)*v1)
103  label var q1 "consumer price post reform"
104  keep code aggr2 p0 t0 a0 v0 p0 q0 p1 t1 a1 v1 q1
105  sort code
106  save "`taxsystem'" /* this is the temporary file with the tax information: one code=one line*/
107
108  ************************************************************************
109  * Part 2: calculation of tax revenues on 974 items for all households *
110  ************************************************************************
111
112  * Read in the STATA-database with for each household the detailed list of expenditures for the 2001 survey
113  *  contains 974 variables beginning with as name c_ followed by the code
114  *  The weighting factors (var "extrapol") in this file are the proper ones (i.e. sum to the number of Belgian households)
115  use "C:\My\Data\Budget Enquetes\2001\BE2001 detail.dta", clear
116
117  * "append" the temporary file with the taxsystem to the dataset currently in memory
118  append using "`taxsystem'" /* last n lines now contain codes and tax rates*/
119
120  *Put the labels of the values of aggr2 in a local variable to use it later
121  forvalues i=1/`nagg' {
122       local label`i': label aggr2labels `i'
123  }
124
125  * Initialize taxes for all aggregates on zero
126  quietly {
127   local loop "0 1"    /*loop variable k over baseline (0) and reform (1)*/
128   foreach k of local loop {
129       forvalues i=1/`nagg' {
130            gen exp`i'_`k'=0        /* expenditures on aggregate i */
131            gen T`i'_t_`k'=0        /* VAT and ad valorem */
132            gen T`i'_a_`k'=0        /* excise 2005, but in 2001 nominal values*/
```

```
133           label var exp`i'_`k'  "`label`i''"
134           label var T`i'_t_`k'  "`label`i''"
135           label var T`i'_a_`k'  "`label`i''"
136      }
137   }
138  }
139
140  gen scode=string(code) /*contains now the code of the item in characterform*/
141  local obs=_N     /* is the total number of lines in the dataset (both households and number of items*/
142  count if code!=. /* this gives the number of consumption items in the dataset*/
143  local ncodes=r(N)/* result of count on the previous line: number of codes*/
144
145  quietly {
146       foreach var of varlist c_* {
147            local i=`obs'-`ncodes'+1
148            capture drop e0 e1
149              gen e0=`var'                          /* exp on this item in baseline*/
150            while `i'<=`obs'{
151                if scode[`i']==substr("`var'",3,6) { /* i is the line where the tax info for this exp category is found */
152                   local k=aggr2[`i']                /*contains the number of the aggregate where this item has to be class
153                   gen e1=e0*(q1[`i']/q0[`i'])    /*assumption of constant quantities, hence new expenditures e1*/
154                   local loop "0 1"
155                       foreach j of local loop {
156                            local alpha = a`j'[`i']/p`j'[`i']
157                            local denom = 1-(1+t`j'[`i'])*v`j'[`i']
158                            local tau_t = (t`j'[`i']*(1+`alpha'+v`j'[`i'])+v`j'[`i'])/`denom'
159                            local tau_a = `alpha'/`denom'
160                            local tau = `tau_t'+`tau_a'
161                       replace exp`k'_`j'=exp`k'_`j'+e`j'     /* expenditure added to appropriate aggregate*/
162                       replace T`k'_t_`j'=T`k'_t_`j'+(`tau_t'/(1+`tau'))*e`j'     /* vat payment on this expenditure*/
163                       replace T`k'_a_`j'=T`k'_a_`j'+(`tau_a'/(1+`tau'))*e`j'     /* excise  payment on this expenditure*/
164                       }
165                drop if code==real(scode[`i'])                              /* this item can be removed*/
166                local i=`obs'+1                        /* this guarantees that the while-loop will be quitted*/
167                local obs=_N                        /* adjusts the variable after removal of the line*/
168                count if code!=.
169                local ncodes=r(N)
170                   }
171               else {
172                local i = `i'+1
173               }
174           }
175      }
176  }
```

```
177
178  * before we collapse into a sum over households, we save the expenditures on the aggregates
179  * drop all detailed expenditures, and tax rates
180  drop if _n>`obs'-`ncodes'
181  keep  hhnr extrapol exp* T* inc_total
182
183  save "C:\My\Onderzoek\Projecten\Alternatieve financiering SZ\exp and tax baseline.dta",replace
184
185  ************************************************************************
186  * Part 3: calculation of implicit tax rates on the nagg aggregates      *
187  ************************************************************************
188  * the following collapse command produces ONE observation with the totals of all variables
189  * Note that the weights are noninteger, that is why we have to use pweight (fweight is only for integers)
190  *   and aweight gives different results because of normalisation of the weights (see help on collpase)
191  * With this collapse we calculate: Sum_i x_i*weight_i (as I think we should)
192  * we will save this observation in a temporary dataset (mother) and recall it
193  * several times to reload variables in memory
194  tempfile mothersums sumexp0 sumexp1 sumvat0 sumvat1 sumexc0 sumexc1
195
196  collapse (sum) exp* T* inc_total [pw=extrapol]/* store the (weighted) sum of expenditures and tax payments on aggregates*
197  sum inc_total /*to keep total income in a local variable*/
198  local income=r(mean)/1e+06
199  di `income'
200  save "`mothersums'" /*saved in a temporary file to be used in stages later on*/
201
202  * create the file with expenditures for the `nagg' aggregates
203  local loop "0 1"
204       foreach j of local loop {     /* j indicates baseline (0) or reform (1)*/
205             use "`mothersums'"
206             local k=abs(`j'-1)       /* k indicates the variables to be dropped*/
207             drop T* exp*_`k' inc_total   /*only expenditures for loop j remain in file*/
208             xpose,clear               /*make the `nagg' aggregates the observations now*/
209             rename v1 exp`j'
210             label var exp`j' "exp `j'"
211             egen aggr2=fill(1/`nagg')   /*add category of aggregate to allow a merge later*/
212             label var aggr2 "aggregate 2"
213             sort aggr2
214             save "`sumexp`j''"
215        }
216  * create the file with VAT taxes for the `nagg' aggregates
217  local loop "0 1"
218       foreach j of local loop {     /* j indicates baseline (0) or reform (1)*/
219             use "`mothersums'"
220             keep T*_t_`j'            /*only vat taxes for loop j remain in file*/
```

```
            xpose,clear               /*make the 18 aggregates the observations now*/
            rename v1 vat`j'
            label var vat`j' "vat `j'"
            egen aggr2=fill(1/`nagg')   /*add category of aggregate to allow a merge later*/
            label var aggr2 "aggregate 2"
            sort aggr2
            save "`sumvat`j''"
      }
* create the file with Excise taxes for the `nagg' aggregates
local loop "0 1"
     foreach j of local loop {     /* j indicates baseline (0) or reform (1)*/
            use "`mothersums'"
            keep T*_a_`j'            /*only excise taxes for loop j remain in file*/
            xpose,clear               /*make the 18 aggregates the observations now*/
            rename v1 exc`j'
            label var exc`j' "excise `j'"
            egen aggr2=fill(1/`nagg')   /*add category of aggregate to allow a merge later*/
            label var aggr2 "aggregate 2"
            sort aggr2
            save "`sumexc`j''"
      }

* merge the three datasets: expenditures, VAT payments, Excise payments
use "`sumexp0'"
merge aggr2 using "`sumexp1'"
drop _merge
sort aggr2
merge aggr2 using "`sumvat0'"
drop _merge
sort aggr2
merge aggr2 using "`sumvat1'"
drop _merge
sort aggr2
merge aggr2 using "`sumexc0'"
drop _merge
sort aggr2
merge aggr2 using "`sumexc1'"
drop _merge

gen tax0=vat0+exc0
label var tax0 "Total indirect tax paid baseline"
gen tax1=vat1+exc1
label var tax1 "Total indirect tax paid reform"

```

```
* add observation `nagg'+1 with totals, this is observation _N
expand 2 in 1 /*duplicates the first observation, which is appended at the end of the file*/
local new=_N  /*observation number for the new observation*/
replace aggr2=`nagg'+1 in `new' /*gets the aggr2 classification of `nagg'+1*/

local loop "0 1"
  foreach i of local loop {
      replace exp`i'=0 in  `new'
      replace tax`i'=0 in  `new'
      replace vat`i'=0 in  `new'
      replace exc`i'=0 in  `new'
     egen totexp`i'=sum(exp`i')    /* summation over the `nagg' aggregates (which are observations)*/
     egen tottax`i'=sum(tax`i')
     egen totvat`i'=sum(vat`i')
     egen totexc`i'=sum(exc`i')

     * put these totals in the new observation
     replace exp`i'=totexp`i' in `new'
     replace tax`i'=tottax`i' in `new'
     replace vat`i'=totvat`i' in `new'
     replace exc`i'=totexc`i' in `new'
  }
* Express the totals in Million €
foreach name of varlist exp* tax* vat* exc* {
     replace `name'=`name'/1e+06
}

* Preparing for output tables:
label define aggr2labels 1 "Food"
label define aggr2labels 2 "Drinks - Non Alco", add
label define aggr2labels 3 "Drinks - Alco", add
label define aggr2labels 4 "Tobacco", add
label define aggr2labels 5 "Clothing", add
label define aggr2labels 6 "Rent and Utilities", add
label define aggr2labels 7 "Heating", add
label define aggr2labels 8 "Private transport", add
label define aggr2labels 9 "Public Transport", add
label define aggr2labels 10 "Hygienics, Health", add
label define aggr2labels 11 "Leisure commodities", add
label define aggr2labels 12 "Other", add
label define aggr2labels 13 "Durables", add

label define aggr2labels 14 "Total", add
label values aggr2 aggr2labels
```

```
309
310  * calculate absolute differences (ad_ ) and percentage differences (pd_) for the tables
311  local list "exp tax vat exc"
312  foreach var of local list {
313       gen ad_`var'=`var'1-`var'0
314       gen pd_`var'=(ad_`var'/abs(`var'0))*100
315  }
316  di "Calculations on Budgetsurvey 2001 - on 974 items and individual households"
317
318  tabstat2 exp0 exp1 ad_exp pd_exp,  /*
319  */   by(aggr2) title(Table 1: Aggregate expenditures in Million € of 2001) /*
320  */   nototal format (%5.1f)
321
322  tabstat2 tax0 tax1 ad_tax pd_tax,  /*
323  */   by(aggr2) title(Table 2: Aggregate indirect tax revenues in Million € of 2001) /*
324  */   nototal format (%5.1f)
325
326  tabstat2 vat0 vat1 ad_vat pd_vat,  /*
327  */   by(aggr2) title(Table 3: Aggregate VAT revenues in Million € of 2001) /*
328  */   nototal format (%5.1f)
329
330  tabstat2 exc0 exc1 ad_exc pd_exc,  /*
331  */   by(aggr2) title(Table 4: Aggregate Excise revenues in Million € of 2001) /*
332  */   nototal format (%5.1f)
333
334  * average tax rates and BUDGET shares by category defined in aggr2
335  * NOTE: these averages are obtained by dividing the taxes paid on this item by the
336  *            total expenditures before taxes.
337  *            Same for budgetshares: exp on aggregate divided by total expenditures
338  *            The last row displays the tax rates on total expenditures
339  local loop "0 1"
340    foreach i of local loop {
341    gen tau`i'=100*tax`i'/(exp`i'-tax`i') /*formula 7 in notes, expressed in percentages*/
342    gen tau_t`i'=100*vat`i'/(exp`i'-tax`i')
343    gen tau_a`i'=100*exc`i'/(exp`i'-tax`i')
344    gen w`i'=(exp`i'/exp`i'[_N])*100
345    gen q`i'=1+tau`i'/100                      /*consumer price for normalised producer price=100*/
346    label var tau`i' "implicit indirect tax rate `i'"
347    label var tau_t`i' "implicit VAT rate `i'"
348    label var tau_a`i' "implicit excise rate `i'"
349    label var w`i' "budget share `i'"
350    label var q`i' "consumer price"
351    format w`i' tau`i' tau_t`i' tau_a`i' q`i' %5.1f
352    format q`i' %5.4f
```

```
353    local totexp`i'=exp`i'[_N] /*store total expenditures for use below*/
354    local tottax`i'=tax`i'[_N] /*store total taxes for use below*/
355    local totvat`i'=vat`i'[_N] /*store total vat for use below*/
356    local totexc`i'=exc`i'[_N] /*store total excises for use below*/
357    }
358
359  tabstat2 w0 w1 tau0 tau1 q0 q1,  /*
360  */   by(aggr2) title(Table 5: Budget shares, tax rates on producer price, and consumer price for the 18 aggregates) /*
361  */   nototal format
362  tabstat2 tau_t0 tau_t1 tau_a0 tau_a1,  /*
363  */   by(aggr2) title(Table 6: Detailed tax rates (VAT and excise) on producer price for the 18 aggregates) /*
364  */   nototal format
365
366  * New budget shares in function of total income
367  * add observation `nagg'+1 with income, this is observation _N
368  expand 2 in 1 /*duplicates the first observation, which is appended at the end of the file*/
369  local new=_N  /*observation number for the new observation*/
370  local lastbutone=`new'-1 /*observation with saving*/
371  replace aggr2=`nagg'+2 in `new' /*gets the aggr2 classification of `nagg'+2*/
372  label define aggr2labels 14 "Saving", modify
373  label define aggr2labels 15 "Income", add
374  label values aggr2 aggr2labels
375
376  forvalues i=0/1 {
377  * fill the last but one observation with savings
378       replace exp`i'=`income'-`totexp`i'' in `lastbutone'
379       replace tax`i'=0 in `lastbutone'
380       replace vat`i'=0 in `lastbutone'
381       replace exc`i'=0 in `lastbutone'
382       replace tau`i'=100*tax`i'/(exp`i'-tax`i') in `lastbutone'
383       replace tau_t`i'=100*vat`i'/(exp`i'-tax`i') in `lastbutone'
384       replace tau_a`i'=100*exc`i'/(exp`i'-tax`i') in `lastbutone'
385       replace q`i'=1+tau`i'/100 in `lastbutone'
386  * fill last observation with income in the variable exp
387       replace exp`i'=`income' in `new'
388       replace tax`i'=`tottax`i'' in `new'
389       replace vat`i'=`totvat`i'' in `new'
390       replace exc`i'=`totexc`i'' in `new'
391        replace tau`i'=100*tax`i'/(exp`i'-tax`i')  in `new'
392       replace tau_t`i'=100*vat`i'/(exp`i'-tax`i')  in `new'
393       replace tau_a`i'=100*exc`i'/(exp`i'-tax`i')  in `new'
394       replace q`i'=1+tau`i'/100 in `new'
395
396       replace w`i'=(exp`i'/`income')*100 /*this holds for all observations*/
```

```
397  }
398
399  format exp0 exp1 %8.0f
400  tabstat2 exp0 exp1 w0 w1 tau0 tau1 q0 q1,  /*
401  */   by(aggr2) title(Table 5: Income shares, tax rates on producer price, and consumer price for the 14 aggregates) /*
402  */   nototal format
403  tabstat2 tau_t0 tau_t1 tau_a0 tau_a1,  /*
404  */   by(aggr2) title(Table 6: Detailed tax rates (VAT and excise) on producer price for the 14 aggregates) /*
405  */   nototal format
406
407  gen pchange=((q1-q0)/q0)*100
408  format pchange %6.2f
409  tabstat2 w0 tau0 q0 q1 pchange, by(aggr2) nototal format
410  latabstat w0 tau0 q0 q1 pchange, by(aggr2) nototal format
411
412  * save the tax rate information in a dataset (only the nagg aggregates, not the total)
413  keep  aggr2 tau0 tau_t0 tau_a0 q0 tau1 tau_t1 tau_a1 q1
414  drop if aggr2>14
415  save "`path1'",replace
416
```

```
 1  clear
 2  set type double
 3  set mem 200m
 4  set matsize 800
 5  set more off
 6  capture log close
 7
 8  log using "C:\My\Onderzoek\Projecten\AIM-AP\Imputation Belgium\tax rates baseline.log", replace
 9
10  * In this program we
11  * 1) calculate the indirect taxes for 2005 on all individual items of the budgetsurvey 2001
12  *  and for all households separately in both a baseline (2005) and reform situation
13  *  and write them into a file
14  * 2) derive the implied tax rates and consumer prices for these aggregates and write them to a file
15  * 3) produce Revenue tables for the `nagg' aggregates
16
17  * DEFINE THE NUMBER OF AGGREGATES HERE (including durables)
18  local nagg=13
19  * GIVE PATH OF DATASET where the tax rates and consumer prices have to be saved for use in other programs
20  local path1="C:\My\Onderzoek\Projecten\AIM-AP\Imputation Belgium\reform01.dta"
21
22  ********************************************************************
23  * Part 1: define the tax system in baseline and reform              *
24  ********************************************************************
25  * retrieve the dataset with tax rates for the year 2005
26  use "C:\My\Onderzoek\Projecten\AIM-AP\Imputation Belgium\pre2005 aggr13.dta", clear
27
28  * create a temporary file with code and tax rates in notation of formula's and with the a reform
29  tempfile taxsystem
30  local deflator=0.9249 /* deflator to go from 2005 nominal level to 2001 nominal level*/
31       * rename the tax rates in the baseline of 2005 (with suffix 0)
32  rename btw t0        /*VAT rate pre reform*/
33  rename acc a0        /*excise per unit pre reform*/
34  rename adval v0      /*ad valorem rate pre reform*/
35  rename prijs q0      /*consumer price pre reform, is price in 2001*/
36  replace a0=a0*`deflator' /*deflate the excise of 2005, into 2001 values (but still tax system of 2005!)*/
37
38  * calculate producer price pre reform
39  gen p0=((1-(1+t0)*v0)/(1+t0))*q0-a0
40  label var p0 "producer price pre reform"
41  count if p0<0
42
43  ****************************************************************
44  *    DEFINE THE REFORM                                          *
```

```
45  * Default is to put all taxes equal to the 2005 situation          *
46  * The reform is introduced by overwriting them                     *
47  * To be discussed whether we will introduce a more user friendly   *
48  *   way of introducing reforms
49  ********************************************************************
50  gen t1=t0
51  gen a1=a0/`deflator'
52  gen v1=v0
53  label var t1 "VAT rate post reform"
54  label var a1 "excise post reform"
55  label var v1 "ad valorem rate post reform"
56
57  replace t1=0.25 if t0==0.21 /*replaces the VAT-rate of 21% with 22%*/
58  replace t1=0.07 if t0==0.06 /*replaces the VAT-rate of 6% with 4%*/
59
60  * Below: overwrite existing excises (excise for 2005 is in the comment at end of each line
61  *        replace the excise with a new value in 2005-prices. It will be deflated later
62  *replace a1=0 if code==121102 /*     Oploskoffie en koffieëxtract      .694 */
63  *replace a1=0 if code==121101 /*  Koffiebonen, gemalen kof., enz.      .248 */
64  *replace a1=0 if code==122107 /*        And. alcoholvrije dranken     .0372 */
65  *replace a1=0 if code==122103 /*         Alcoholvrije aperitieven     .0372 */
66  *replace a1=0 if code==123209 /*                 Alcoholvrije wijn     .0372 */
67  *replace a1=0 if code==123104 /*   Likeuren en stke dranken + 23°      5.96 */
68  *replace a1=0 if code==123103 /*                          Jenever      5.96 */
69  *replace a1=0 if code==123101 /*                           Whisky      7.01 */
70  *replace a1=0 if code==123199 /*                     Alcohol z.o.      5.96 */
71  *replace a1=0 if code==123102 /*                           Cognac      7.01 */
72  *replace a1=0 if code==123203 /*    Champagne en mousserende wijn      .161 */
73  *replace a1=0 if code==123201 /*                             Wijn      .471 */
74  *replace a1=0 if code==123202 /*                            Cider      1.61 */
75  *replace a1=0 if code==123204 /*    Likeurwijn e.a. drnk 15°- 23°      .992 */
76  *replace a1=0 if code==123303 /*                       Ander bier      .205 */
77  *replace a1=0 if code==123302 /*                   Bier type pils      .205 */
78  *replace a1=0 if code==123399 /*                        Bier z.o.      .205 */
79  *replace a1=0 if code==123309 /*   Alcoholvrij en alcoholarm bier     .0744 */
80  *replace a1=0 if code==123301 /*                        Tafelbier     .0684 */
81  *replace a1=0 if code==130103 /*      Sigaretten (20 st. per pak)    .00689 */
82  *replace a1=0 if code==130105 /*  Sigaretten (10 of 15 st. p/pak)    .00689 */
83  *replace a1=0 if code==130199 /*                        Sigaretten    .00689 */
84  *replace a1=0 if code==130104 /*      Sigaretten (25 st. per pak)    .00689 */
85  *replace a1=0 if code==322101 /*                          Aardgas   .000322 */
86  *replace a1=0 if code==322102 /*    Butaangas (zonder statiegeld)      .017 */
87  *replace a1=0 if code==322103 /*   Propaangas (zonder statiegeld)     .0174 */
88  *replace a1=0 if code==321101 /*                Elektriciteit    .00191 */
```

```
 89  *replace a1=0 if code==323102 /*    Andere vloeibare brandstoffen     .0179 */
 90  *replace a1=0 if code==323101 /*   Gasoil, fuel en collect. verw.     .0171 */
 91  *replace a1=0 if code==622106 /*                   Tweetakt olie      .488 */
 92  *replace a1=0 if code==622199 /*                    Benzine z.o.      .565 */
 93  *replace a1=0 if code==622101 /*    Spr. benz. zn. lood (Sup. 95)     .592 */
 94  *replace a1=0 if code==622103 /*                Benzine met lood      .552 */
 95  *replace a1=0 if code==622102 /*   Spr. benz. zn. lood (Euro+ 98)     .607 */
 96  *replace a1=0 if code==622105 /*                          Diesel      .349 */
 97  replace a1=a1*`deflator' /*deflate the reformed excise of 2005, into 2001 values*/
 98
 99  * calculate the new consumer prices after the reform (Ass: fixed producer prices)
100  gen p1=p0
101  label var p1 "producer price post reform"
102  gen q1=((1+t1)*(p1+a1))/(1-(1+t1)*v1)
103  label var q1 "consumer price post reform"
104  keep code aggr2 p0 t0 a0 v0 p0 q0 p1 t1 a1 v1 q1
105  sort code
106  save "`taxsystem'" /* this is the temporary file with the tax information: one code=one line*/
107
108  ************************************************************************
109  * Part 2: calculation of tax revenues on 974 items for all households *
110  ************************************************************************
111
112  * Read in the STATA-database with for each household the detailed list of expenditures for the 2001 survey
113  *  contains 974 variables beginning with as name c_ followed by the code
114  *  The weighting factors (var "extrapol") in this file are the proper ones (i.e. sum to the number of Belgian households)
115  use "C:\My\Data\Budget Enquetes\2001\BE2001 detail.dta", clear
116
117  * "append" the temporary file with the taxsystem to the dataset currently in memory
118  append using "`taxsystem'" /* last n lines now contain codes and tax rates*/
119
120  *Put the labels of the values of aggr2 in a local variable to use it later
121  forvalues i=1/`nagg' {
122       local label`i': label aggr2labels `i'
123  }
124
125  * Initialize taxes for all aggregates on zero
126  quietly {
127   local loop "0 1"    /*loop variable k over baseline (0) and reform (1)*/
128   foreach k of local loop {
129       forvalues i=1/`nagg' {
130            gen exp`i'_`k'=0        /* expenditures on aggregate i */
131            gen T`i'_t_`k'=0        /* VAT and ad valorem */
132            gen T`i'_a_`k'=0        /* excise 2005, but in 2001 nominal values*/
```

```
133           label var exp`i'_`k'  "`label`i''"
134           label var T`i'_t_`k'  "`label`i''"
135           label var T`i'_a_`k'  "`label`i''"
136      }
137   }
138  }
139
140  gen scode=string(code) /*contains now the code of the item in characterform*/
141  local obs=_N     /* is the total number of lines in the dataset (both households and number of items*/
142  count if code!=. /* this gives the number of consumption items in the dataset*/
143  local ncodes=r(N)/* result of count on the previous line: number of codes*/
144
145  quietly {
146       foreach var of varlist c_* {
147            local i=`obs'-`ncodes'+1
148            capture drop e0 e1
149              gen e0=`var'                          /* exp on this item in baseline*/
150            while `i'<=`obs'{
151                if scode[`i']==substr("`var'",3,6) { /* i is the line where the tax info for this exp category is found */
152                   local k=aggr2[`i']                /*contains the number of the aggregate where this item has to be class
153                   gen e1=e0*(q1[`i']/q0[`i'])    /*assumption of constant quantities, hence new expenditures e1*/
154                   local loop "0 1"
155                       foreach j of local loop {
156                            local alpha = a`j'[`i']/p`j'[`i']
157                            local denom = 1-(1+t`j'[`i'])*v`j'[`i']
158                            local tau_t = (t`j'[`i']*(1+`alpha'+v`j'[`i'])+v`j'[`i'])/`denom'
159                            local tau_a = `alpha'/`denom'
160                            local tau = `tau_t'+`tau_a'
161                       replace exp`k'_`j'=exp`k'_`j'+e`j'     /* expenditure added to appropriate aggregate*/
162                       replace T`k'_t_`j'=T`k'_t_`j'+(`tau_t'/(1+`tau'))*e`j'     /* vat payment on this expenditure*/
163                       replace T`k'_a_`j'=T`k'_a_`j'+(`tau_a'/(1+`tau'))*e`j'     /* excise  payment on this expenditure*/
164                       }
165                drop if code==real(scode[`i'])                              /* this item can be removed*/
166                local i=`obs'+1                        /* this guarantees that the while-loop will be quitted*/
167                local obs=_N                        /* adjusts the variable after removal of the line*/
168                count if code!=.
169                local ncodes=r(N)
170                   }
171               else {
172                local i = `i'+1
173               }
174           }
175      }
176  }
```

```
177
178  * before we collapse into a sum over households, we save the expenditures on the aggregates
179  * drop all detailed expenditures, and tax rates
180  drop if _n>`obs'-`ncodes'
181  keep  hhnr extrapol exp* T* inc_total
182
183  save "C:\My\Onderzoek\Projecten\Alternatieve financiering SZ\exp and tax baseline.dta",replace
184
185  ************************************************************************
186  * Part 3: calculation of implicit tax rates on the nagg aggregates      *
187  ************************************************************************
188  * the following collapse command produces ONE observation with the totals of all variables
189  * Note that the weights are noninteger, that is why we have to use pweight (fweight is only for integers)
190  *   and aweight gives different results because of normalisation of the weights (see help on collpase)
191  * With this collapse we calculate: Sum_i x_i*weight_i (as I think we should)
192  * we will save this observation in a temporary dataset (mother) and recall it
193  * several times to reload variables in memory
194  tempfile mothersums sumexp0 sumexp1 sumvat0 sumvat1 sumexc0 sumexc1
195
196  collapse (sum) exp* T* inc_total [pw=extrapol]/* store the (weighted) sum of expenditures and tax payments on aggregates*
197  sum inc_total /*to keep total income in a local variable*/
198  local income=r(mean)/1e+06
199  di `income'
200  save "`mothersums'" /*saved in a temporary file to be used in stages later on*/
201
202  * create the file with expenditures for the `nagg' aggregates
203  local loop "0 1"
204       foreach j of local loop {     /* j indicates baseline (0) or reform (1)*/
205             use "`mothersums'"
206             local k=abs(`j'-1)       /* k indicates the variables to be dropped*/
207             drop T* exp*_`k' inc_total   /*only expenditures for loop j remain in file*/
208             xpose,clear               /*make the `nagg' aggregates the observations now*/
209             rename v1 exp`j'
210             label var exp`j' "exp `j'"
211             egen aggr2=fill(1/`nagg')   /*add category of aggregate to allow a merge later*/
212             label var aggr2 "aggregate 2"
213             sort aggr2
214             save "`sumexp`j''"
215        }
216  * create the file with VAT taxes for the `nagg' aggregates
217  local loop "0 1"
218       foreach j of local loop {     /* j indicates baseline (0) or reform (1)*/
219             use "`mothersums'"
220             keep T*_t_`j'            /*only vat taxes for loop j remain in file*/
```

```
            xpose,clear               /*make the 18 aggregates the observations now*/
            rename v1 vat`j'
            label var vat`j' "vat `j'"
            egen aggr2=fill(1/`nagg')   /*add category of aggregate to allow a merge later*/
            label var aggr2 "aggregate 2"
            sort aggr2
            save "`sumvat`j''"
      }
* create the file with Excise taxes for the `nagg' aggregates
local loop "0 1"
     foreach j of local loop {     /* j indicates baseline (0) or reform (1)*/
            use "`mothersums'"
            keep T*_a_`j'            /*only excise taxes for loop j remain in file*/
            xpose,clear               /*make the 18 aggregates the observations now*/
            rename v1 exc`j'
            label var exc`j' "excise `j'"
            egen aggr2=fill(1/`nagg')   /*add category of aggregate to allow a merge later*/
            label var aggr2 "aggregate 2"
            sort aggr2
            save "`sumexc`j''"
      }

* merge the three datasets: expenditures, VAT payments, Excise payments
use "`sumexp0'"
merge aggr2 using "`sumexp1'"
drop _merge
sort aggr2
merge aggr2 using "`sumvat0'"
drop _merge
sort aggr2
merge aggr2 using "`sumvat1'"
drop _merge
sort aggr2
merge aggr2 using "`sumexc0'"
drop _merge
sort aggr2
merge aggr2 using "`sumexc1'"
drop _merge

gen tax0=vat0+exc0
label var tax0 "Total indirect tax paid baseline"
gen tax1=vat1+exc1
label var tax1 "Total indirect tax paid reform"

```

```
* add observation `nagg'+1 with totals, this is observation _N
expand 2 in 1 /*duplicates the first observation, which is appended at the end of the file*/
local new=_N  /*observation number for the new observation*/
replace aggr2=`nagg'+1 in `new' /*gets the aggr2 classification of `nagg'+1*/

local loop "0 1"
  foreach i of local loop {
      replace exp`i'=0 in  `new'
      replace tax`i'=0 in  `new'
      replace vat`i'=0 in  `new'
      replace exc`i'=0 in  `new'
     egen totexp`i'=sum(exp`i')    /* summation over the `nagg' aggregates (which are observations)*/
     egen tottax`i'=sum(tax`i')
     egen totvat`i'=sum(vat`i')
     egen totexc`i'=sum(exc`i')

     * put these totals in the new observation
     replace exp`i'=totexp`i' in `new'
     replace tax`i'=tottax`i' in `new'
     replace vat`i'=totvat`i' in `new'
     replace exc`i'=totexc`i' in `new'
  }
* Express the totals in Million €
foreach name of varlist exp* tax* vat* exc* {
     replace `name'=`name'/1e+06
}

* Preparing for output tables:
label define aggr2labels 1 "Food"
label define aggr2labels 2 "Drinks - Non Alco", add
label define aggr2labels 3 "Drinks - Alco", add
label define aggr2labels 4 "Tobacco", add
label define aggr2labels 5 "Clothing", add
label define aggr2labels 6 "Rent and Utilities", add
label define aggr2labels 7 "Heating", add
label define aggr2labels 8 "Private transport", add
label define aggr2labels 9 "Public Transport", add
label define aggr2labels 10 "Hygienics, Health", add
label define aggr2labels 11 "Leisure commodities", add
label define aggr2labels 12 "Other", add
label define aggr2labels 13 "Durables", add

label define aggr2labels 14 "Total", add
label values aggr2 aggr2labels
```

```
309
310  * calculate absolute differences (ad_ ) and percentage differences (pd_) for the tables
311  local list "exp tax vat exc"
312  foreach var of local list {
313       gen ad_`var'=`var'1-`var'0
314       gen pd_`var'=(ad_`var'/abs(`var'0))*100
315  }
316  di "Calculations on Budgetsurvey 2001 - on 974 items and individual households"
317
318  tabstat2 exp0 exp1 ad_exp pd_exp,  /*
319  */   by(aggr2) title(Table 1: Aggregate expenditures in Million € of 2001) /*
320  */   nototal format (%5.1f)
321
322  tabstat2 tax0 tax1 ad_tax pd_tax,  /*
323  */   by(aggr2) title(Table 2: Aggregate indirect tax revenues in Million € of 2001) /*
324  */   nototal format (%5.1f)
325
326  tabstat2 vat0 vat1 ad_vat pd_vat,  /*
327  */   by(aggr2) title(Table 3: Aggregate VAT revenues in Million € of 2001) /*
328  */   nototal format (%5.1f)
329
330  tabstat2 exc0 exc1 ad_exc pd_exc,  /*
331  */   by(aggr2) title(Table 4: Aggregate Excise revenues in Million € of 2001) /*
332  */   nototal format (%5.1f)
333
334  * average tax rates and BUDGET shares by category defined in aggr2
335  * NOTE: these averages are obtained by dividing the taxes paid on this item by the
336  *            total expenditures before taxes.
337  *            Same for budgetshares: exp on aggregate divided by total expenditures
338  *            The last row displays the tax rates on total expenditures
339  local loop "0 1"
340    foreach i of local loop {
341    gen tau`i'=100*tax`i'/(exp`i'-tax`i') /*formula 7 in notes, expressed in percentages*/
342    gen tau_t`i'=100*vat`i'/(exp`i'-tax`i')
343    gen tau_a`i'=100*exc`i'/(exp`i'-tax`i')
344    gen w`i'=(exp`i'/exp`i'[_N])*100
345    gen q`i'=1+tau`i'/100                      /*consumer price for normalised producer price=100*/
346    label var tau`i' "implicit indirect tax rate `i'"
347    label var tau_t`i' "implicit VAT rate `i'"
348    label var tau_a`i' "implicit excise rate `i'"
349    label var w`i' "budget share `i'"
350    label var q`i' "consumer price"
351    format w`i' tau`i' tau_t`i' tau_a`i' q`i' %5.1f
352    format q`i' %5.4f
```

```
353    local totexp`i'=exp`i'[_N] /*store total expenditures for use below*/
354    local tottax`i'=tax`i'[_N] /*store total taxes for use below*/
355    local totvat`i'=vat`i'[_N] /*store total vat for use below*/
356    local totexc`i'=exc`i'[_N] /*store total excises for use below*/
357    }
358
359  tabstat2 w0 w1 tau0 tau1 q0 q1,  /*
360  */   by(aggr2) title(Table 5: Budget shares, tax rates on producer price, and consumer price for the 18 aggregates) /*
361  */   nototal format
362  tabstat2 tau_t0 tau_t1 tau_a0 tau_a1,  /*
363  */   by(aggr2) title(Table 6: Detailed tax rates (VAT and excise) on producer price for the 18 aggregates) /*
364  */   nototal format
365
366  * New budget shares in function of total income
367  * add observation `nagg'+1 with income, this is observation _N
368  expand 2 in 1 /*duplicates the first observation, which is appended at the end of the file*/
369  local new=_N  /*observation number for the new observation*/
370  local lastbutone=`new'-1 /*observation with saving*/
371  replace aggr2=`nagg'+2 in `new' /*gets the aggr2 classification of `nagg'+2*/
372  label define aggr2labels 14 "Saving", modify
373  label define aggr2labels 15 "Income", add
374  label values aggr2 aggr2labels
375
376  forvalues i=0/1 {
377  * fill the last but one observation with savings
378       replace exp`i'=`income'-`totexp`i'' in `lastbutone'
379       replace tax`i'=0 in `lastbutone'
380       replace vat`i'=0 in `lastbutone'
381       replace exc`i'=0 in `lastbutone'
382       replace tau`i'=100*tax`i'/(exp`i'-tax`i') in `lastbutone'
383       replace tau_t`i'=100*vat`i'/(exp`i'-tax`i') in `lastbutone'
384       replace tau_a`i'=100*exc`i'/(exp`i'-tax`i') in `lastbutone'
385       replace q`i'=1+tau`i'/100 in `lastbutone'
386  * fill last observation with income in the variable exp
387       replace exp`i'=`income' in `new'
388       replace tax`i'=`tottax`i'' in `new'
389       replace vat`i'=`totvat`i'' in `new'
390       replace exc`i'=`totexc`i'' in `new'
391        replace tau`i'=100*tax`i'/(exp`i'-tax`i')  in `new'
392       replace tau_t`i'=100*vat`i'/(exp`i'-tax`i')  in `new'
393       replace tau_a`i'=100*exc`i'/(exp`i'-tax`i')  in `new'
394       replace q`i'=1+tau`i'/100 in `new'
395
396       replace w`i'=(exp`i'/`income')*100 /*this holds for all observations*/
```

```
397  }
398
399  format exp0 exp1 %8.0f
400  tabstat2 exp0 exp1 w0 w1 tau0 tau1 q0 q1,  /*
401  */   by(aggr2) title(Table 5: Income shares, tax rates on producer price, and consumer price for the 14 aggregates) /*
402  */   nototal format
403  tabstat2 tau_t0 tau_t1 tau_a0 tau_a1,  /*
404  */   by(aggr2) title(Table 6: Detailed tax rates (VAT and excise) on producer price for the 14 aggregates) /*
405  */   nototal format
406
407  gen pchange=((q1-q0)/q0)*100
408  format pchange %6.2f
409  tabstat2 w0 tau0 q0 q1 pchange, by(aggr2) nototal format
410  latabstat w0 tau0 q0 q1 pchange, by(aggr2) nototal format
411
412  * save the tax rate information in a dataset (only the nagg aggregates, not the total)
413  keep  aggr2 tau0 tau_t0 tau_a0 q0 tau1 tau_t1 tau_a1 q1
414  drop if aggr2>14
415  save "`path1'",replace
416
```