Prototype of country IT microsimulation model (IRAP, Social Contributions, Corporate Tax)

EUROPEAN COMMISSION
INFORMATION SOCIETY TECHNOLOGIES
(IST) PROGRAMME

Deliverable 5.3

Prototype of Country IT Microsimulation Model (IRAP, Social Contributions, Corporate Tax)

Work Package No. 5
Tax indicators construction and development: step 1 – Conceptual Framework and Development of the National Tax Base Modules

Rossella Bardazzi,Francesca Gastaldi, Michela Nardo,Valentino Parisi and Maria Grazia Pazienza

First Draft - November, 2003

Final Version – January 2004

IST Programme, contract No 2000-31125
Development of a System of Indicators on Competitiveness and Fiscal Impact on Enterprises Performance (DIECOFIS)

Project Partners

Coordinator

Istituto Nazionale di Statistica (I)

Principal Contractors

Board of Inland Revenue (UK)

London School of Economics (UK)

University of Cambridge (UK)

Università di Roma Tor Vergata (I)

Università di Firenze (I)

Centro di Ricerca Economica e Sociale (I)

European Commission Joint Research Centre (I)

Informer SA Computing Information Systems and Management Cons. (EL)

Wirtschaftsuniversität Wien (A)

Prototype of Country IT Microsimulation Model

(IRAP, Social Contributions, Corporate Tax)

Rossella Bardazzi[*], *Francesca Gastaldi*[**], *Michela Nardo*[***], *Valentino Parisi*[****] *and Maria Grazia Pazienza*[*1]

CONTENTS OF DELIVERABLE

[1] * Department of Studi sullo Stato, University of Florence. ** Department of Economia Pubblica, University of Rome "La Sapienza". *** European Commission Joint Research Centre, **** Department of Economia e Territorio, University of Cassino

Introduction and summary

The DIECOFIS project has been aimed to contribute specifically in three areas: 1) development of a system of indicators on competitiveness that can be used to benchmark enterprise performance,; 2) development of a first generation of microsimulation models that can serve to monitor and simulate the impact of public policy on enterprises with different characteristics or belonging to different sectors; 3) development of integrated data bases of micro enterprise data bases that can serve to generate indicators by means of algorithms or model.

In particular, Workpackage 5 is meant to develop and deliver a "rough" prototype of a microsimulation model for the business sector to estimate the impact of different taxes on small, medium , and large firms. This micromodel for firms is innovative, path breaking, and key both in a national and EU perspective, in particular for its potential in showing the impact that policy have on competitiveness and economic renewal in different sectors of the economy.

This deliverable presents the results of this part of the project. Firstly, we introduce the potential use of the DIECOFIS microsimulation model in an European framework as a tool to evaluate benefits and costs of fiscal competition or tax harmonization. Then in Section 2, we describe the model structure while the list and desription of endogenous and exogenous variables is detailed in Appendix B and the code for running the model is in Appendix C. Three examples of policy analysis are reported in Section 3: a simulation of the economic effects of a comprehensive fiscal reform and a reform of the severance pay – both for the Italian economy – and a cross-country simulation between Italy and United Kingdom for R&D tax reliefs. These examples are meant to show the flexibility of use and the model capabilities. The sensitivity analysis for some module of the microsimulation model has been performed and results are presented in Appendix A.

1 The Diecofis Model: a tool for European Policy Analysis

1.1 Tax competition or tax harmonization?

The more recent European tax reforms, aimed at the lowering of tax rates and broadening of tax bases, has been actually influenced by globalisation itself and the subsequent need to lessen the impact of distortions caused by fiscal variables. The process of international integration urged individual countries to "reconsider" both their national tax systems as well as their level of public expenditure. This was done for the purposes of identifying the best fiscal "setting" for investments.

At the same time, both nationally and internationally, a heated theoretical debate was ignited regarding the alternatives of tax *harmonization* (or, more realistically speaking, *coordination*) of tax bases and rates, and tax *competition*. For these two hypotheses, characteristics of efficiency, equity and transparency of corporate tax systems are weighed .[2]

Critics (see Musgrave, Schwab and Bovenberg) argue that tax competition generates negative effects on wealth as well as causing some distortion in the choices of the public administration, thus resulting in excessive costs for efficiency and equity. Critics see tax competition as a sort of *beggar-my-neighbour* politics, which results in a level of tax rates on income from capital that are lower than what would be advantageous, with this leading to significant consequences[3]. In particular, the capacity for public funding is reduced, and this may lead to a downsizing or worsening of collective services. Also, the displacing of the tax burden to the least mobile tax bases lowers the fairness of overall levying of taxes, thus creating ties to the sustainability (both political and financial) of redistributive policy. And lastly, adverse effects to employment are cited by critics, deriving from the long period of tightening of taxation on labour.

On the other hand, tax competition is viewed positively within economic theory on tax federalism, the original context in which problems deriving from the instating of tax policies within interdependent integrated areas were studied. From this perspective, we postulate an analogy between the effects of efficiency of the mechanism of competition within a 'product' market and within an 'institutions' market: "competition between governments should produce, within the public sector, the same type of benefits that are generally associated with competition between

[2] In the EU, the debate on tax competition was started by the pressure for competition, which in its turn, derives (for companies in countries with a high level of public expenditure and taxation) from the process of integration of the internal market and the single currency. In Europe, taxation is higher than in France, Germany and Italy as well as in the northern countries (Holland, Denmark and Sweden).

[3] Countries are obliged to lower tax rates to create competition and to avoid capital leaving the country, which results in corporate income tax rates tending to reach the zero mark.

private companies" (McLure, 1986). Countries compete to attract resources and tax bases through offering institutions; this should lead to an optimum arrangement in both levying taxes and in the offer of public services.[4]

Since the inception of the EU, the subject matter of taxation is has always been the exclusive prerogative of the member states. It is considered an integral part of national choices and preferences regarding economic and social politics that lie outside the scope of the Convention and EU policies. An exception is however established by the rules of the internal market: generally, fiscal measures that create obstacles to the free circulation of goods, services and/or capital, or measures which might distort the rules of competition, are not permissible.

The important question remains as to how to insure that competition between different regimes responds to needs for efficiency.

Policies of taxation that are mainly or exclusively motivated to attract financing or other mobile tax bases, as well as behaviour of avoidance carried out by enterprises, could cause undesirable distortions to international trade and investments and, at the same time, lower the global wealth.[5]

These considerations have lead to the development of the concept of harmful tax competition as a specific issue dealt with in cooperative agreements stipulated to do away with specific distorting effects and behaviour of taxpayers and of governments attempting to distinguish between fair and unfair tax competition.

If, on the one hand, the prevailing view in the definition of unfair or harmful competition seems to be concerned with the protection of the reasons of the States, then it is no accident that emphasis is placed on the erosion of the national tax bases that said procedures produce[6]. On the other hand, from the point of view of the company, the problem of harmful tax competition mostly limits itself to the distortions that said policies could cause to the free competition of companies, bringing about, for instance, changes in prices involved in international trade and guaranteeing, thus, through the lowering of the tax burden, funding linked to the operativity of enterprises.

Under the current regime of differentiation of tax systems, actions taken by individual countries are not always helpful. In many countries, tax authorities may effect adjustments to earnings of a resident company, attributing to the transactions contested for tax purposes a transfer

[4] In particular, we refer to the well-known contribution of Tiebout (1956), according to which, when electors 'vote with their feet' an optimum market solution is reached for the offer of local public goods.
[5] OCSE (1998).
[6] For an analysis of social dumping see Lusignoli (2003).

price that is in line with the market values. Moreover, often infra-group transactions are not comparable due to their differences from normal market transactions. Thus, this principle is difficult to apply. Competitive adjustment procedures (that are not coordinated) affected by individual countries may also result in cases of double taxation.

In order to adopt fiscal measures and decisions there must be a unanimous decision of the European Union Committee (cfr. articles 93–95 of the EC Treaty). [7] Article 94 provides for the possibility of directives for coordination and approximation of national norms "that have a direct effect on the internal market". The directives for coordination leave the national norms intact, but set rules for areas of contact and interfacing between national systems for activity that crosses the national borders. The two main examples in this area are directive 90/435, aimed at eliminating double taxation on dividends; and directive 90/434, which regulates mergers, contributions, splits and other transactions aimed at altering company structure, allowing for the effecting of these transactions under a fiscally neutral regime.

One example of a cooperative reaction in this area is the convention regarding transfer pricing. This convention established an arbitration procedure whose objective was to inhibit cases of double taxation that were not covered by the network of existing bilateral conventions. The convention, adopted by the European Committee in 1990, entered into force on 1 January 1995 and after being ratified by the member states, provided that an enterprise could take recourse against the tax authorities in charge of levying taxes on the company profits, by filing a procedure, initially dealing in information and conciliation, and later in arbitration, which must conclude with the elimination of the double taxation involved.

In the EU, coordinated activities on a large scale have, as of today, only been encountered in theoretical studies which have brought about proposals that are yet to be actually applied to a real case study. This, however, is if one excludes the reaction of industrial countries to the strategies of unfair tax competition practiced by tax havens, or rather, what is called the CFC (controlled foreign corporations) legislation.

The CFC (Controlled Foreign Corporations) legislation is one example of a non-cooperative solution to the problem of different tax systems interfering with each other. The legislation establishes that the resident controlling company be taxed on profit "produced" (even if said profit has not yet been distributed) by controlled companies located in countries having privileged tax systems (tax havens). This kind of intervention does however cause conflict that is difficult to

[7] The possibility of a complete harmonization of national norms is provided only for indirect tax (art. 93 of the Treaty) "in the needed proportions so as to insure the establishing of and the functioning of the internal market".

eradicate between the taxation authority of the country that adopts the norm and the authority of the (presumed) tax haven and the other countries having competitive CFC regulations. This occurs above all when there are existing international agreements between these countries aimed at avoiding double taxation.[8]

1.2 The UE policy orientation

Almost ten years after the presentation of the Ruding Report, in October 2001 the European Commission published a new report (*Toward an internal market without tax obstacles)* indicating what the EU action should be in regard to fiscal issues for enterprises. The proposals therein are a product of the difficulties encountered in creating a European tax legislation (given the fact that there must be a unanimous vote of the Committee and adherence to the principle of subsidiarity) as well as the limited success in applying the provisions suggested in the Ruding Report.

The opinion of the Commission is that the existence of 15 tax regimes that are different amongst themselves makes for a substantial obstacle in reaching a single market. The Commission deals with this issue by proposing, on one hand, actions that are "aimed" (in the short term) at eliminating obstacles, and on the other hand, "global actions" (in the medium-long term) meant to eliminate the factors that result in hindrances to cross-border business activity.

Some of the "aimed" actions are: a) extending of the directive regarding mergers b) adjustment of the directive regarding parent-subsidiary companies to eliminate the withholding tax and to levy tax on profits only on the company producing said profit and not on the company receiving it: a substantial change, mainly, in extending the access to this directive also to shareholdings with quotes that are lower than the current 25 per cent limit; c) the proposal of a new directive on the cross-border offsetting of losses; d) a directive (the draft has already been drawn up) regarding royalties and interest and the taxation of these only in the country of the beneficiary receiving them, thus eliminating the application of a withholding tax in the country from which they are distributed; e) the establishing of a permanent joint Forum on transfer pricing, comprised of representatives of member states and enterprises; f) lastly, the submitting of a petition regarding the need to adapt the conventions against double taxation to a standard model that would render them adherent to common principles and thus avoid a mix of individual tax systems.

[8] In many countries the adopting of CFC legislation caused an increasingly contentious issue between tax authorities and tax payers. In addition, the fact that more and more countries are adopting CFC legislation results in a overlapping of tax obligations on companies located in fiscally priviledged countries that is difficult to resolve.

With a medium-long term objective, the Commission initiates a debate on whether to adopt a consolidated tax base at a European level. This would allow for the calculation of taxable income of Multinational Groups by referring to only one set of legislation. Basically, this would be an attempt to deal with the problem (currently one issue that ME must deal with) of tax compliance formalities required by the different fiscal systems in the countries in which business activities are located. The Commission however permits each individual country to decide on its own tax rate to apply to the relevant taxable income, thus granting countries some degree of flexibility for tax competition.

In order to reach this objective an efficient and politically feasible route must be found. In particular, member states must reach an agreement on two matters: a) how to determine the tax base for enterprises operating in different countries b) what mechanism to adopt for the division and attribution of taxable profits to countries.

The solution of the first issue (letter a) is a main priority and is still in the drawing up phase. The Commission have identified four systems:

1) *European company income tax* (EUCIT): this system provides for the creation of a tax to be levied at a European level. A part, or the whole of said tax could go directly to the EU. Originally conceived as an obligatory regime for large ME, at the start it may be seen as an optional system. The idea that all member states may waive all, or even part, of their decision making power on the levying of corporate income tax does however seem to be quite unlikely.

2) *Home State taxation* (HST): this system provides that the tax base be calculated according to the tax regulations in the country in which the main headquarters of the company are located. It is conceived as a non-obligatory regime that a company operating in a different country may choose to adopt.

This method does not require that member states establish common rules in that in order to implement the system one only needs to have the mutual recognition of the taxation systems involved (although each country would have to recognize 15 systems and, with the prospect of extending the EU, even 25 systems!). This has been defined as a route that would be politically feasible and one which should not be faced with any particular obstacles given that it would not be an obligatory regime for companies. From another point of view, however, the possibility should not be underestimated that more fierce (and likely harmful) competition may result, in contrast to current competition in determining tax bases in several countries having a negative outflow on the income of those companies belonging to enterprises that are part of Multinational Groups (the risk is ending up with very low tax bases or even bases reaching zero). ME tax bases could tend towards a homogeneity, yet at a lower than advantageous level, and thus the problem of arriving at a more

substantial agreement setting a limit, even partial, to the decision making power of each county would only be postponed up to the moment in which this competition is perceived as harmful. Lastly, this method would not solve the potential problem of companies that, although they may be operating in the same country and in the same sector, could be subject to very different tax regimes such as to alter fair competition among companies.

3) *Common base taxation* (CBT): this system proposes the creation of harmonized rules at a EU level for the purpose of determining a single European tax base. This regime would also be optional.

From a technical point of view, CBT offers two advantages over HST: i) in each member state one would only need to be aware of the EU regulations and not the regulations of the other 14 member states; ii) a starting point would be created for the establishing of European tax norms.

The most relevant obstacle that CBT poses is undeniably the difficulty of reaching the codes need for a common tax base, and obtaining the agreement of all member states. This difficulty is exacerbated by the fact that, currently, each country has a series of more or less extensive "tax expenditures", or rather, advantages connected to the country they belong to (such as advanced amortizations). In establishing a common tax base it would be difficult to "sum up" each individual tax advantage. The system would end up generating a more extensive tax base than the actual tax base existing in each member country. In this case, enterprises would have no interest in choosing a less favourable regime, unless the different countries were to lower their tax rates. This would however create for repercussions (not considered in the EU plans) also in the tax levy affected on domestic companies. The problem remains, as is the case with HST, of a co-existence of determining different tax bases, in the same country, for ME who have opted for said EU regime and the domestic companies.

4) *Harmonized single tax base in the EU*: this system provides for the progressive harmonization of national directives for determining the corporate tax base.

This proposal would be enacted over time: the 15 systems of determining the tax base would be gradually harmonized, but said harmonization would involve all companies and not only those companies dealing in cross-border activity. It is likely that this method would come up against similar negotiating difficulties as the CBT. However, it would bring about a more direct and less costly transition: a) in resolving the problem of taxation of ME in Europe; b) in lessening, compared to the other methods, the added costs and requirements deriving from operating in more than one country, thus improving the conditions of international competitivity; c) in rendering tax competition more transparent among countries in that it would be exclusively confined to setting of tax rates; d) in improving competition between companies on a national level as well as

internationally. Furthermore, this seems to be the method that is most in line with the short term "aimed" provisions mentioned above. This is because said provisions tend (albeit in reference to certain institutions) to harmonize the legislation of different countries.

2 The model structure

The main task of the project was to build a microsimulation model for fiscal policy evaluation. The model has achieved this objective for the taxes explicitly considered: the Regional Tax on Economic Activity (IRAP), the social contributions paid by employers (SC), and the corporate tax. The IRAP and SC modules have been built at the University of Florence and the corporate tax module (CTM) was developed at the University of Tor Vergata. The prototype microsimulation model has been integrated and tested on a subset of firms at the University of Florence. The basic structure of the prototype is sketched in Figure 1.

The simulation of IRAP, Social Security Contributions and IRPEG modules is based on a dataset called "Regional Tax and Social Security" (RTSS). Two ISTAT (Italian Institute of Statistics) surveys are combined in the RTSS: the Small and Medium Sized Enterprises survey (PMI) carried out on firms with less than 100 workers and the survey on Large Enterprises (SCI) with more than 99 employees. These survey data are not completely adequate to build a model for these taxes. In fact, some computations cannot be performed using the data (in that they are too aggregate to do so) and therefore additional data sources are used. In particular, although the survey data covers the company balance sheet in some detail, we need to match information from our dataset against published accounts in which some variables are recorded at a more disaggregated level. Therefore, for a selection of firms in the RTSS dataset the survey data is integrated with administrative data on balance sheets. This integrated dataset is specifically well suited for simulating corporate tax rules. The dataset has been built at ISTAT and described in Deliverable 1.1 (Denk, Oropallo 2002). Additional data sources are used when appropriate to compute some parameters used in specific tax modules.

A detailed description of each tax module, including all the equations and modelling assumptions, is presented in previous deliverables of WP5 and WP6.[9] The microsimulation model is run with a precise order: firstly, the social contributions are modelled and their revenue is estimated. As a second step, the IRAP tax base is computed and the tax yield is computed. Finally, the IRPEG

[9] See Deliverable 5.2 for IRAP, Deliverable 5.1 for social contributions and Deliverable 6.2 for corporate taxation.

module is run to estimate the corporate tax on a selection of the overall dataset. Appendix B of this deliverable presents a list with variable names – both endogenous and exogenous – and description of their content. Details of the program where each endogenous variable is computed are also added. The STATA code of the microsimulation model is contained in the included file and listed in Appendix C.

Logically, some important economic interactions are implemented in the model. For example, some deductible labour costs for IRAP are computed in the social contributions module and are therefore endogenous to the model. Moreover, social contributions are, as a component of labour costs, deductible from the corporation tax base. With all modules working together, an interaction between labour cost policy (specifically a change in social security contribution rates) and the corporate tax due by a firm can be estimated.

A primary characteristic of the microsimulation model is represented by its flexibility: the model reproduces the Italian tax system from the year 1998 to the year 2003, it can be adjusted to simulate the impact of several policies, and dynamics can be easily introduced.
The foremost results of the Diecofis model can be summarized as follows:

1) the fiscal revenue of all taxes considered has been reproduced and validated at the year 1998: the model's fit can be considered very satisfactory (see Bardazzi, Parisi, Pazienza 2003);
2) the fiscal revenue of all taxes under different scenarios has been forecasted and, in some cases, a check with government official forecasts – such as the Technical Reports presented in Parliament – shows that our estimates are in a close range with those data;
3) the fiscal incidence of policies has been analysed by different reference categories: geographical area, size, legal status, activity sectors, etc.;
4) the regional breakdown has allowed an analysis of the tax system and of fiscal policy changes by regions: this issue is very important in a framework of fiscal federalism where IRAP plays a key role for regions;
5) specific tax indicators have been studied and built: *ex-post implicit tax rates* (EPITR) have been chosen and calculated as the ratio between taxes actually paid and a reference economic aggregate (e.g. profits, capital, value added, etc.).

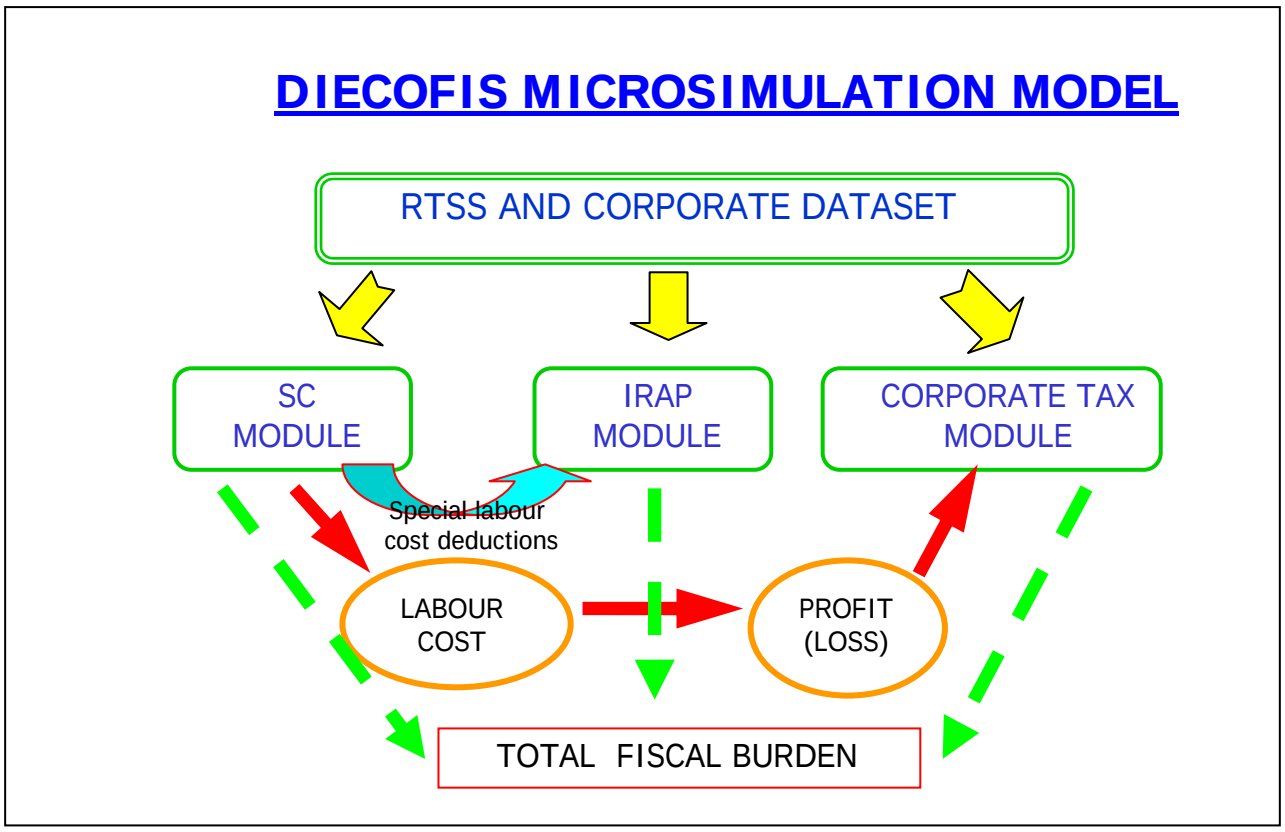
This model has been tested and used for simulating several policy changes: results of these exercises are summarized in the following section. On the modelling side, an additional enhancement of the basic model has been designed to perform one of these simulations, the reform

of the severance pay. In this case, a more dynamic model was required to reproduce the stock of severance payments over a decade. Therefore, an iterative procedure has been written and, using the 1998 dataset as our pivot year, the model has been run backward and forward for the period 1996-2006. For this purpose, wages, employment and assets at the firm level have been computed for the simulation horizon using auxiliary information from the national accounts and other sources.

A further improvement of the Diecofis model is due to the sensitivity analysis designed and performed with the Joint Research Center (JRC). This experiment has been based on the identification of a set of key parameters in our social contributions and IRAP modules to be varied and tested with specific procedures. To accomplish this task, a special routine has been designed in order to run the model with about 300 simulations and results. The sensitivity analysis main findings are summarised in Appendix A.

The DIECOFIS microsimulation model has a significant potential in terms of future development for policy impact analysis. First of all, the model can be made dynamic as the dataset is including several years of microdata. At that stage, agents behaviour can be incorporated into the model. Moreover, other taxes could be introduced with specific modules and dataset. As already stressed in other documents, excise taxes, environmental taxes and value added tax require a larger set of information with respect to the RTSS dataset. Additional survey data and auxiliary information have already been identified and specific statistical procedures have been envisaged in collaboration with ISTAT. This enhancement of the model would allow the analysis of several European policies such as the double dividend of the environmental taxes in terms of employment, the fiscal harmonisation of VAT and other indirect taxes on the business sector.

Figure 1 – Structure of the Diecofis prototype model



3 Three examples of policy analysis

3.1 The effect of Italian Fiscal Reform

A first simulation exercise has been performed to evaluate the impact on enterprises of the new provisions and fiscal rules concerning IRAP and social contributions introduced in 2001, 2002 and 2003 and the fiscal reform proposed by the current Italian government concerning the corporate tax and IRAP that is planned to be in force in January 2004. More in detail, the new corporation tax system moves back to a uniform tax rate (33%) as the previously existing dual rate system (DIT allowance) is abolished, brings in some changes to the definition of the tax base by exempting corporate dividends and removing the dividend tax relief, by exempting capital gains from long term assets owned for at least one year, by limiting deductibility of interest costs provided under thin capitalisation rules. The reform also introduces an optional consolidated tax statement for groups that can be extended to foreign subsidiaries[10]. In this deliverable we discuss the results obtained so far.

[10] For a more detailed discussion on these aspects see Bardazzi, Parisi, Pazienza (2003). It is important to note that available data do not allow to simulate the impact of the consolidated tax statement for groups which is therefore excluded from the simulation.

Without going into further details, it is noteworthy that when the current government came into effect (June 2001) introduced changes to the corporate tax system which actually anticipates some features of the forthcoming reform. Therefore this year represents the base-case scenario against which evaluating the impact of the reform, as 2001 is the last year of the previous regime. The simulation exercise is thus implemented by considering two different policy scenarios: the first, 2001, represent the tax structure existing before the current government came into power, the second, 2004, the reformed scenario[11]. In both scenarios the IRAP tax structure has been revised due to changes of the tax rates (as for year 2000 regions are allowed to change IRAP rates[12]) and to the introduction of specific allowances and exemptions. Furthermore, in the reformed scenario (2004) we also assume the 20% reduction of labour cost from the IRAP tax base, as a first step towards the gradual (and future) abolition of this tax, proposed in the Fiscal Enabling Law.

Table 1 provides estimates of ex-post implicit tax rates (backward looking) in the 2001 and the reformed scenarios obtained running the integrated model. The implicit rates has been calculated as ratio between the estimated taxes and turn-over.

Table 1 - Ex-post implicit tax rates for different scenarios: sectoral breakdown; percentage values

		EPITRs (absolute values)		Differences (2004 – 2001)		
Sector of Activity	(%)	2001 Scenario	Reform Scenario (2004)	**Overall**	*Irap*	*Corporation tax*
Manufacturing, mining	57,72	3,84	3,62	-0,21	-0,21	0,00
Electrical energy, gas, steam, water	1,22	2,36	2,43	0,07	-0,24	0,31
Construction	3,89	3,19	3,00	-0,18	-0,29	0,11
Wholesale and retail trade services	8,50	2,22	2,13	-0,09	-0,12	0,03
Hotel and restaurant services	2,31	4,60	4,07	-0,53	-0,45	-0,08
Transport, storage, communication services	7,68	6,24	5,76	-0,49	-0,81	0,32
Real estate, renting and business services	11,94	5,88	4,89	-0,99	-0,80	-0,19
Education services	0,12	4,47	2,82	-1,65	-1,52	-0,13
Health and social services	4,39	6,09	5,35	-0,74	-0,82	0,08
Other community, social and personal services	2,23	5,71	5,26	-0,44	-0,40	-0,04
Total	100	4,24	3,89	-0,35	-0,37	0,02

Source: Authors' estimates

[11] As updating balance sheet variables would inevitably be imprecise and would present strong biases, the analyses are performed using 1998 balance sheets in all scenarios. The tax rules implemented for 1998, the base-line year of the model, were updated to the 2001 tax legislation and to the reformed scenario.

[12] For a detailed analisys of IRAP Regional Revenues, see Buglione Di Monale (2004)

As regards the modelled corporation tax reform, the simulation show that the mean implicit corporate tax rate would be basically unchanged as compared to the mean rate estimated in the 2001 scenario (specifically, results show an increase of this rate by 0,02 percentage points). The estimates show also some interesting findings. The effects of the corporation tax reform in each sector, shortly, depend both on changes of the tax base and on the (uniform) statutory rate of taxation provided by the reform, as compared to the 'effective' one prevailing in the 2001 scenario where a dual rate system was present. As a total result, firms of the sectors 'real estate and business activities', 'hotels and restaurant', 'education', 'other social and personal services' would gain from the reform, while companies of all other sectors would record a rise in the implicit tax rate, as they had gained from the previous dual rate system. Among these sectors, companies of the 'electricity, gas, water supply' and of the 'transport and communication' sectors, experience the highest tax rates increase. As a consequence of the mentioned revisions of modelled taxes, the overall tax burden, computed as the ratio between estimated tax revenue and turn-over, drops from 4,24 to 3,89, therefore by 0,35 percentage points. As compared to firms residing in Northern and Central Italy, which all show tax rate falls similar or just below (as for instance in the North-West) the mean rate reduction, Southern firms are favoured by these reforms as they feature the greatest drop of the tax burden (-0,83). This is an interesting result considering that in both scenarios Southern firms support the highest tax burden compared to enterprises of the other geographical areas.

3.2 A cross country simulation: Italy-UK

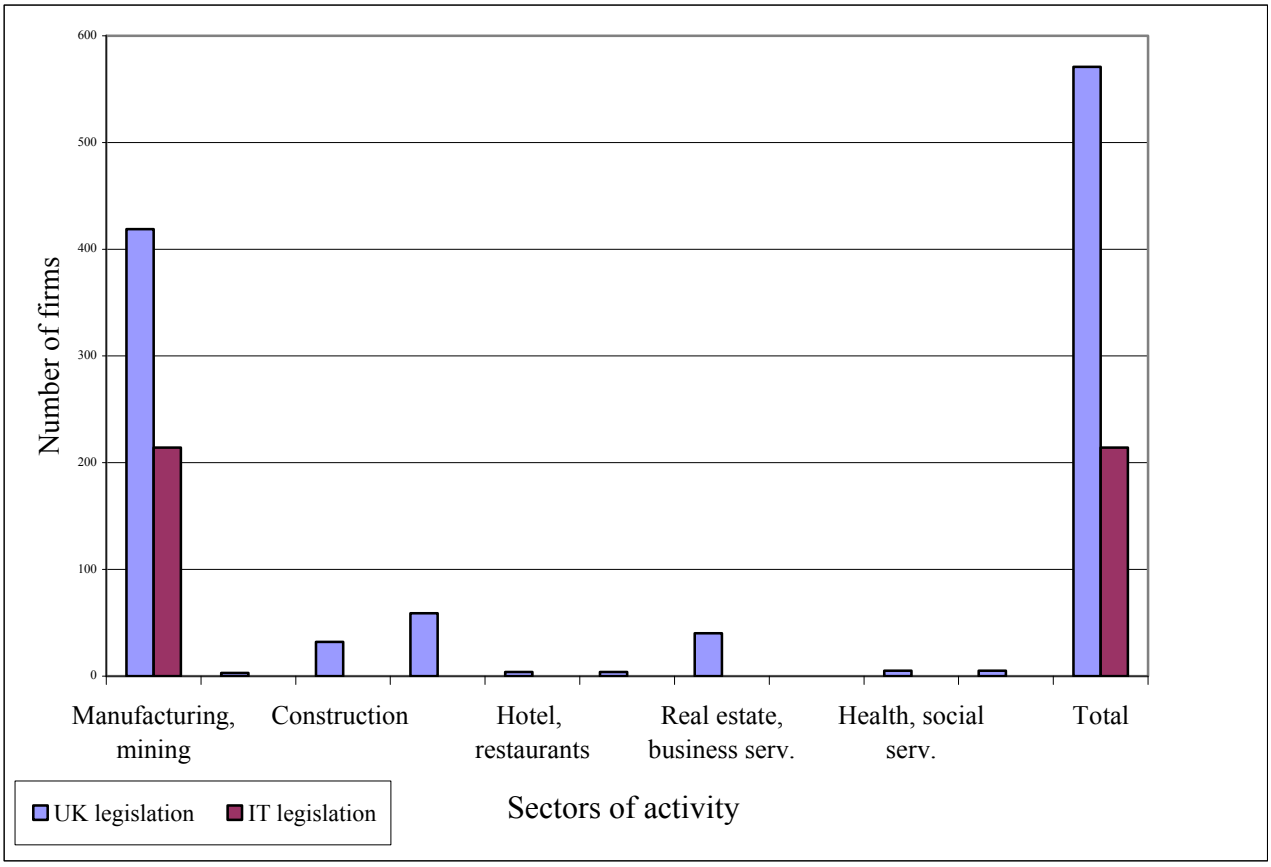
In order to test the flexibility of the model in reproducing tax legislations of other countries, we also consider a cross-country simulation exercise where we implement the rules defining eligibility to the R&D tax credit existing in the UK in 2000 in the CTM. In this simulation exercise[13], we basically address the following questions: *what would be the impact of replacing the R&D tax relief provided by the Italian tax legislation with the R&D tax credit existing in the UK?* This simulation is therefore performed on the subset of large enterprises, and the results are shown in figure 2.

The figures show that the total number of firms qualifying for the R&D tax relief would increase from 214 to 571 if Italy 'used' the British rules. One interesting conclusion that we can draw from this simulation is that while no enterprises of the services sector benefit from the R&D tax relief

[13] For a more detailed description of the methodology used in this simulation see Parisi (2003).

provided by the Italian tax legislation[14], in the manufacturing sector the number of firms qualifying for this specific tax relief would almost double (from 214 to 419).

Figure 2. Effects of replacing the IT R&D tax relief with the UK R&D tax credit. Number of eligible enterprises for activity sector of the firm



Source: authors' estimates

3.3 The Effects of the Planned Social Security Reform on Firms

The Social Security Reform, currently under discussion in Parliament, proposes the switch to pension funds of the resources which were until now used for severance pay[15]. For this exercise, a more dynamic model was required in order to reproduce the stock of severance payments accumulated in balance sheet over a decade. Therefore, an iterative procedure has been designed and, using the 1998 dataset as our pivot year, the model has been run backward and forward for ten years. Wages, social contributions, employment turnover, total assets and financial debt cost at the

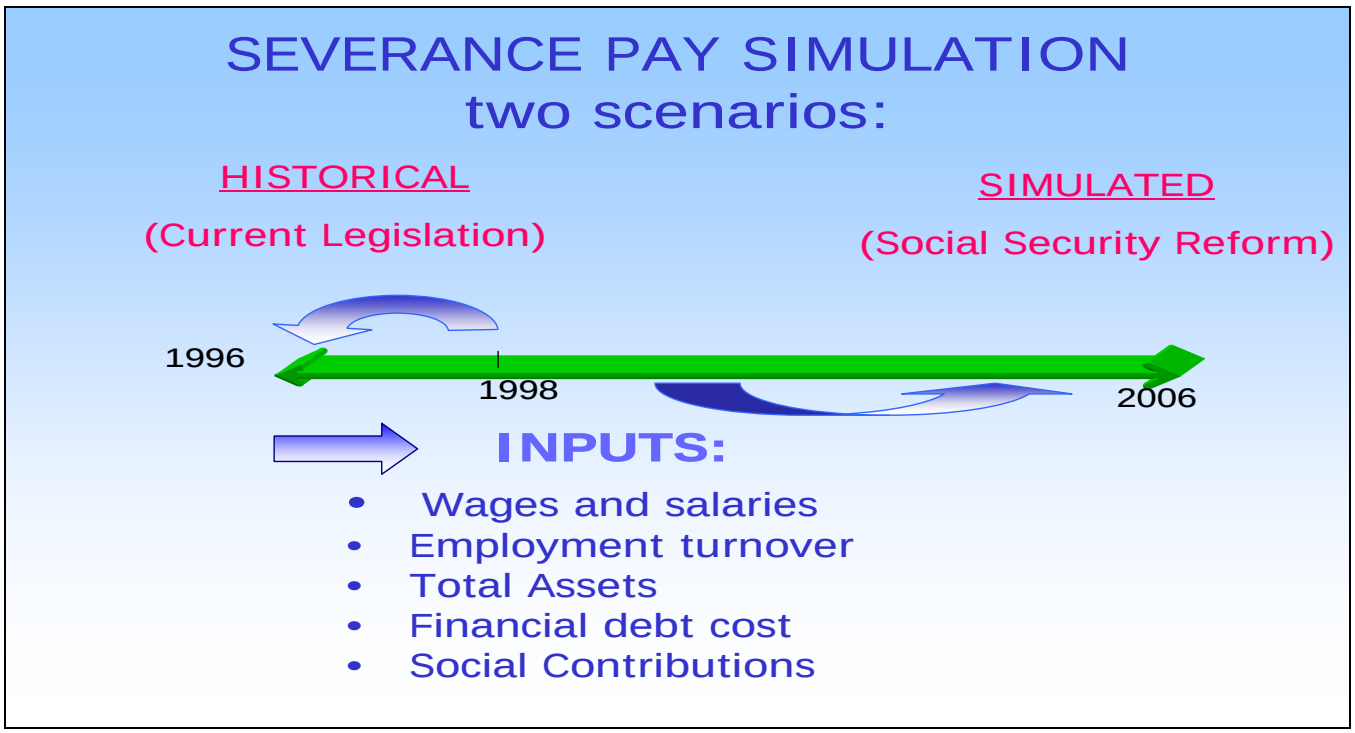
[14] Indeed, in Italy, as for the firms of the services sectors, the R&D tax relief is basically designed for small and medium sized firms (with less than 75 dependent workers), while in the industry eligible enterprises must have less than 200 workers.

[15] See deliverable 5.1 for a description of the severance pay regime in Italy.

firm level have been computed for the simulation horizon using auxiliary information from the national accounts and other sources.

Two scenarios has been estimated: historical (current legislation for severance pay) and simulated (Social Security Reform with the switch of flows of severance pay from firms to pension funds).

Figure 3 - Severance Pay Simulation Scheme

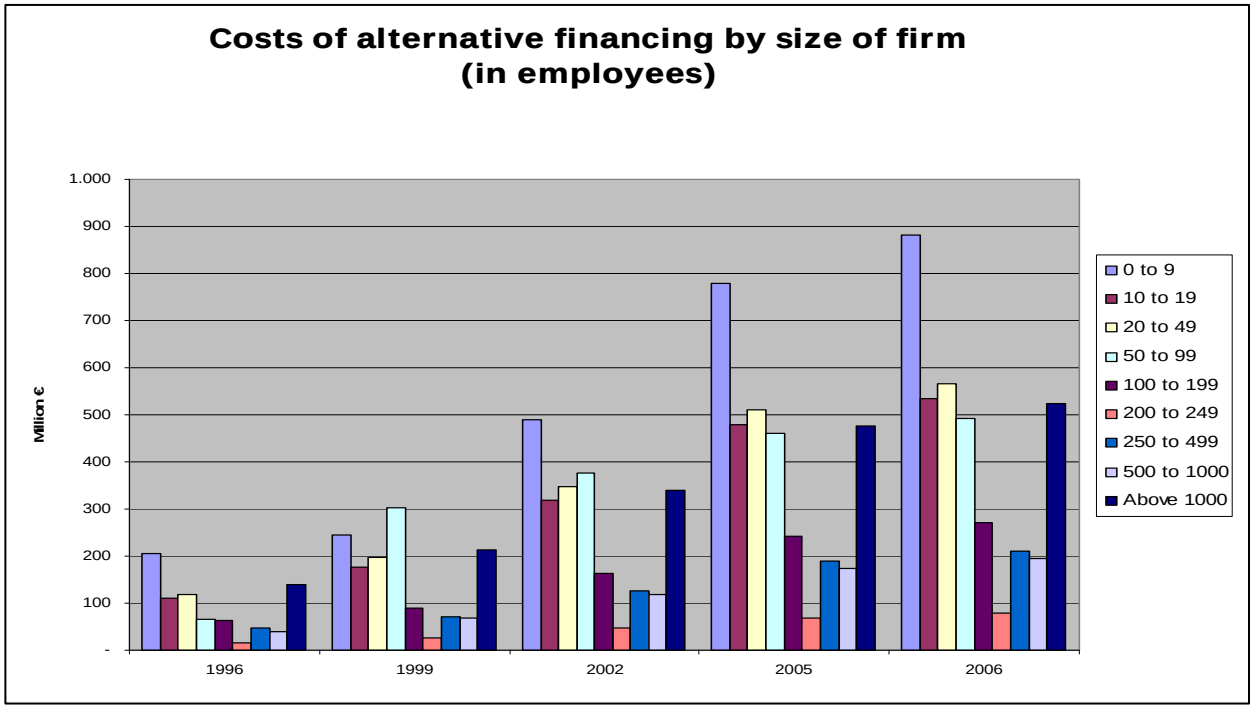


As the preliminary analysis reveals a high heterogeneity of firms in debt structure, severance pay resources and average financial debt cost, the effect of the reform has been analysed deeply for size classes: the cost of loosing the severance pay as a financial instrument would be higher for small firms with employees because of their larger share of self financing and their relative higher difficulty of in accessing credit markets[16].

The next figure shows that the cost of alternative financing (bank loans), would be much higher for firms with less than 10 employees.

[16] For more details on this simulation, see Bardazzi Gastaldi Pazienza (2003).

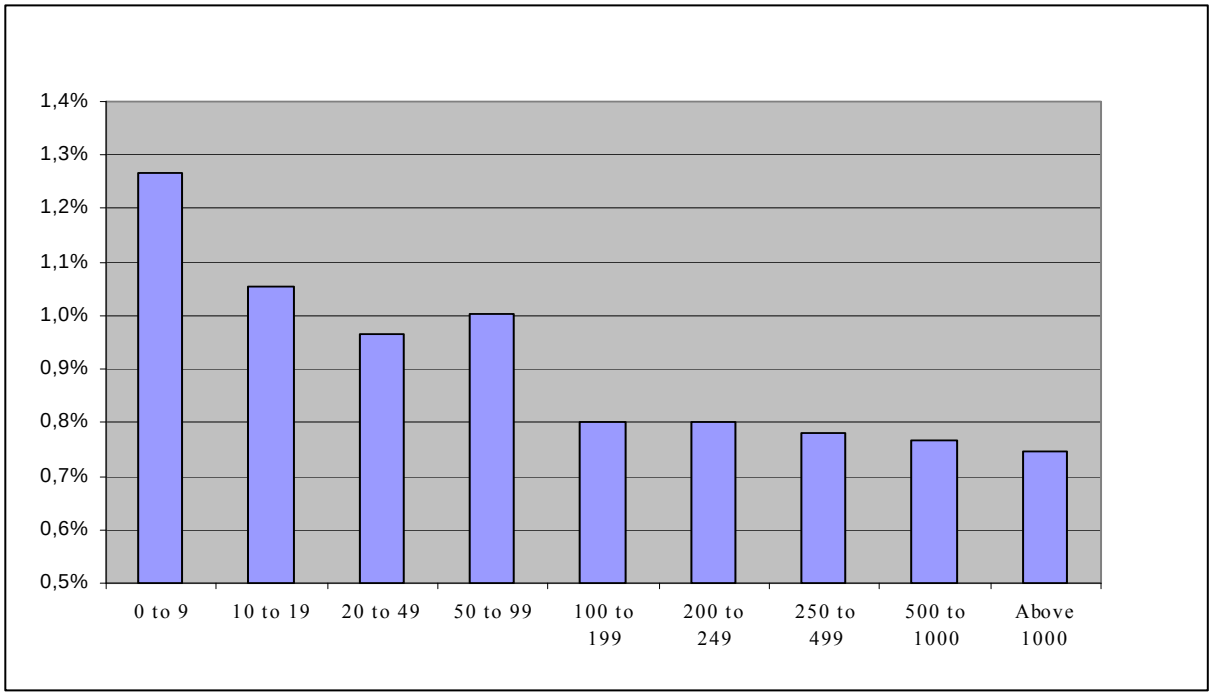
Figure 4 - Estimation of alternative financing costs



Source: Bardazzi, Gastaldi, Pazienza (2003)

Moreover, using total wages as a control variable, the higher incidence of alternative financing costs on small firms is confirmed, as shown in figure 5.

Figure 5. Alternative Financial Costs and Total Wages for size classes (1998)



Source: Bardazzi, Gastaldi, Pazienza (2003)

APPENDIX A

MODEL SENSITIVITY ANALYSIS

A.1 Model Sensitivity Analysis

Beside issues related to the available dataset, other sources of uncertainty may be important for judging the reliability of model results. For the Diecofis Microsimulation model, we were uncertain about the degree of reliability of part of the dataset and wanted to know whether estimates strongly depended upon those data. Furthermore, over-parameterization was highly suspected. The sensitivity analysis has been performed on the IRAP and social contributions modules. Two specific sectors of activity for firms have been chosen to perform the analysis: *manufacture of textiles* (NACE code 17) and *construction* (NACE code 45). The rationale of this choice consists in the high variance of the exogenous parameters computed from external sources and used for to compute the IRAP tax base and wages and salaries for social contributions.

A.1.1. Morris Sensitivity Measures

These issues have been addressed by using the formal sensitivity analysis (SA), and in particular the method of Morris (Morris, 1991). Consider a deterministic model represented by $Y = f(\mathbf{X})$ where $\mathbf{X} = (X_1, X_2, ..., X_d)$ is a vector of d uncertain input variables or factors (like for example imputed data, variables whose parameterization is questioned or variables whose effect on output is uncertain) and Y is the model output. Morris method for sensitivity analysis aims at determining the factors that may be considered to have effects on output which are (a) negligible, (b) linear and additive, or (c) non-linear or involving interactions with other variables. This is done by measuring the output function at different points of the input space such that for each variation of an input, an estimate of the effect on output is computed. The average of all effects will supply a global "first order" effect of the factor X_i.

The first step in the procedure is to discretize the region of factors variability Ω into $(p-1)^k$ hypercubes of equal size; p is the number of levels selected over the space of the input variables (i.e. the discrete number of values that a factor may assume) and k the number of factors. The method of Morris varies one factor at a time. For each variation ΔX_i, an estimate of the effect is computed (ΔY). Then *elementary effects* are calculated. The elementary effect $d_i(X)$ of the i^{th} factor is:

$$d_i(X) = \frac{Y(X_1,...,X_{i-1},X_i+\Delta,X_{i+1},...,X_k)-Y(X)}{\Delta} \quad (1)$$

where Δ is a predefined multiple of *1/(p-1)* and all factors can take on any of the p values $\{0,1/(p-1),2/(p-1),...,1\}$ (in practical applications the values sampled in Ω are rescaled to generate actual simulation values), with the additional condition $X_i \le 1-\Delta$. The main idea of this method is to randomly select *(k+1)* values for $X_1,...,X_k$ and compute the elementary effects. The distribution of elementary effects is denoted by F_i, i.e. $d_i(X) \sim F_i$. Notice that for factors following a uniform distribution levels are obtained by simply divide in equal parts the range of input variation. When factors follow other distributions (like in this case) it is opportune to select levels in the space of the quantiles of the distribution (Campolongo et al. 1999).

The randomly chosen starting value for $X_1,...,X_k$ may introduce some bias. However, the bias is reduced by repeating the sampling procedure r times at different points of Ω, thus a sample of r elementary effects from each distribution F_i is obtained. Morris proposes to calculate the importance of the k factors by looking at the sample mean μ and the standard deviation σ of the distribution F_i. The average (over r samples) μ yields a first order effect of X_i. High mean suggests that the factor considered has a high influence in the model output, while by computing the standard deviation of the same set of ΔY one obtains an estimate of non linear and interaction effects. Notice that a low μ could either be associated to a low overall importance of X_i or to high elementary effects with opposite sign that cancel out. This is why F_i has been modified to consider the absolute variations of elementary effects: $|d_i(X)| \sim G_i$. The mean μ^* of G_i is then calculated instead of μ to correct the propensity to Type II error displayed by the original measure μ (Campolongo, Cariboni, and Saltelli, 2003).

The great advantage of the method is the low computational cost. The total number of runs in which factor sampling is performed and the variation of output is calculated is *r(k+1)*. For other methods the number of runs required is at least quadratic in the number of factors. Therefore, Morris' method is very useful in models with many factors and computationally expensive runs. The main output of Morris method is an ordered set of non influent factors, i.e. factors whose value could be set to any given value over their range of uncertainty without affecting significantly the outcome of the analysis. It is unable, instead, to identify exactly the portion of the variability of Y that can be

imputed to a single factor. However, by reducing the number of factors to analyse, Morris lives room for using other more "quantitative" SA methods.

A.1.2. Microsimulation Model Sensitivity

The calibration of the model for the badly forecasted sectors has been carried out through sensitivity analysis applied to the micro-simulations of IRAP and SC tax revenue.

Table A.1 gives details of the difference between the micro-simulation model's estimation of IRAP and the actual Tax Authority data for the most representative sectors. Textile and construction were not the only sectors displaying a large difference between actual and estimated values. However, for Extraction, Transport and Other services sectors the availability and reliability of data prevented further investigation.

Table A.1 - Comparison between micro-simulation estimations and actual values of fiscal revenue: breakdown by sectors.

	Irap Revenue	*Model Estimation*	*Difference (%)*
Extraction	88	78	-11,1
Food, Textile, Luggage..	1.947	2.089	7,3
Coke, Chemicals..	1.263	1.252	-0,9
Metals product , motor vehicles..	3.206	3.367	5,0
Electricity, gas, steam and hot	563	532	-5,4
Construction	1.366	1.182	-13,5
Trade, Hotels and Restaurants	3.353	3.573	6,5
Transport	1.412	1.789	26,7
Real estate, renting and research activities	2.117	2.149	1,5
Other service sectors	1.912	1.148	-40,0
Total	17.383	17.159	-1,3

Source: Diecofis Deliverable 5.1

Overall 73 inputs factors have been perturbed in order to explore the importance of input factors on output variation. For both IRAP and SC, factors have been divided in two sets. The first set includes the variables *pesi01* to *pesi13*, which account for the corrections needed to obtain fiscal values from the available survey data (see deliverable 5.2 Bardazzi, Gastaldi and Pazienza, 2002).

The suffix c1 to c5 refer to revenue dimension of enterprises, from the lowest (c1: below 258.228 Euro of turnover) to the highest (c5: above 516.459.899 Euro). The second set of factors refer to variables used to improve the information on wages and are listed in Table A.2.

The SA with the Morris' method has been performed using SimLab (EC-IPSC, available at http://www.jrc.cec.eu.int/uasa/prj-sa-soft.asp), the number of levels have been set to 4, thus 296 (number of independent samplings) runs of output evaluation (obtained from ISTAT estimations) have been associated to the corresponding perturbation of input factors. Notice that given the higher number of parameters involved in the modelling of IRAP and SC fluctuations and the complexity of the microsimulation exercise, Morris method is the best cost/effectiveness method available to perform the sensitivity analysis thanks to its relatively low computational cost.

Table A.2 - Exogenous Input factors to the SC module employed for the SA

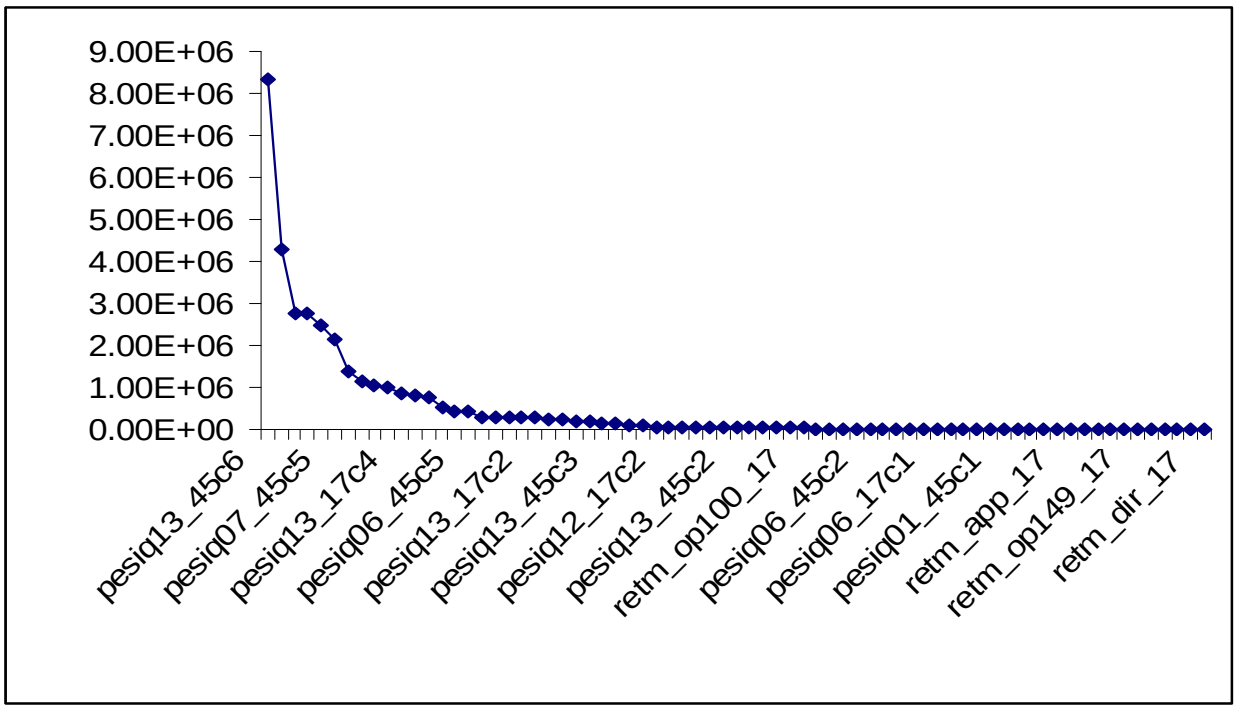
Factors	
Dir_perc	Share of executives in the variable "employees and executives"
Retm_op149	Mean wage for blue collars with 1-49 employees
Retm_op5099	Mean wage for blue collars in enterprises with 50-99 employees
Retm_op100	Mean wage for blue collars in enterprises with more than 100 employees
Retm_imp149	Mean wage for white collars in enterprises with 1-49 employees
Retm_ imp 5099	Mean wage for white collars in enterprises with 50-99 employees
Retm_ imp 100	Mean wage for white collars in enterprises with more than 100 employees
Retm_app	Mean wage for apprentices
Retm_dir	Mean wage for executives

The main goal of the screening exercise is that of ranking factors in order of importance with the objective of identifying the subset of least important, i.e. the ones that do not significantly influence the variability of output (IRAP and SC estimations). This is done by calculating the measure μ^* which estimates average of absolute elementary effects of input factors. This measure constitutes a yardstick to determine the overall influence of the corresponding factor on model variation (see Campolongo, Cariboni, and Saltelli, 2003), it includes the linear (or first order) effects, measured by the Morris measure μ and the curvature or interaction effects measured by σ.

A.1.3. Sensitivity analysis of IRAP module

Table 4 shows the results of the sensitivity analysis using Morris' method for the IRAP module. Factors are ranked according to the measure μ^* in decreasing order of importance and are plotted in Figure A.1. The values of μ^* display a high dispersion (see Table A.2) meaning that the output varies enough to make the ranking obtained reliable.

Figure A.1. Results of SA using Morris method for the IRAP module. Plot of factors ranked according to the measure μ^*.



The analysis of Figure A.1 indicates that the first 20-25 factors explain most of the variability of IRAP. The estimated model, therefore, seems to be highly over-parameterised. The rest of factors display little or no importance in explaining IRAP fluctuation thus those factors could be neglected (i.e. fixed to any value over their range of uncertainty). However these factors could be useful for specific simulation on peculiar policy changes.

Out of the first 10 most important factors, 7 pertain to the construction sector (NACE code 45) and 3 pertain to the textile sector (NACE code17) (in the first 25 the proportion is 14 and 11 respectively).

Among the first 25 factors only 2 (both relative to the sector of construction) pertain to set showed in Table 4, i.e. variables used to improve the information on wages. This scarce influence is due to

the fact that wages have effect on IRAP only throughout the labour cost deductions that, as prescribed by the general tax design, are of very little importance[17].

Table A.3- Results of SA using Morris method for the IRAP module. Order of importance of factors according to the measure μ^* and associated ranking place for the measure σ

Factor	Value of μ^*	Rank of μ^*	Rank of σ	Factor	Value of μ^*	Rank of μ^*	Rank of σ
pesiq13_45c6	8330000	**1**	62	**pesiq12_17c3**	37900	**38**	41
pesiq13_17c5	4290000	**2**	16	**pesiq06_45c4**	33400	**39**	27
pesiq13_45c5	2770000	**3**	30	**pesiq06_17c3**	25600	**40**	58
pesiq01_17c5	2740000	**4**	35	**retm_op100_17**	25600	**40**	14
pesiq01_45c5	2500000	**5**	39	**pesiq01_45c2**	22900	**42**	43
pesiq07_45c5	2120000	**6**	47	**pesiq06_45c6**	13600	**43**	68
pesiq07_17c4	1390000	**7**	21	**pesiq01_17c2**	11800	**44**	24
pesiq13_45c4	1150000	**8**	65	**retm_imp100_17**	11700	**45**	11
pesiq07_45c6	1060000	**9**	62	**pesiq06_45c2**	8480	**46**	15
pesiq07_45c4	1020000	**10**	20	**retm_op5099_45**	5770	**47**	65
pesiq13_17c4	859000	**11**	3	**pesiq01_17c1**	5410	**48**	56
pesiq06_17c5	802000	**12**	23	**retm_imp5099_45**	4890	**49**	57
pesiq01_45c6	783000	**13**	53	**pesiq12_45c2**	3710	**50**	51
pesiq07_17c5	505000	**14**	52	**pesiq06_17c1**	3190	**51**	64
pesiq12_17c4	452000	**15**	7	**pesiq07_45c1**	3150	**52**	71
pesiq06_45c5	447000	**16**	45	**retm_dir_45**	2380	**53**	4
retm_op100_45	303000	**17**	13	**pesiq06_45c1**	2310	**54**	43
pesiq12_17c5	294000	**18**	48	**pesiq07_17c1**	2070	**55**	58
pesiq01_45c3	291000	**19**	29	**pesiq01_45c1**	1990	**56**	58
pesiq07_17c3	274000	**20**	17	**retm_op5099_17**	1790	**57**	53
pesiq13_17c2	270000	**21**	53	**dir_perc_45**	1780	**58**	8
retm_imp100_45	257000	**22**	1	**retm_imp5099_17**	1640	**59**	31
pesiq12_45c4	236000	**23**	46	**pesiq06_17c2**	1440	**60**	32
pesiq13_17c3	202000	**24**	6	**retm_app_17**	1120	**61**	37
pesiq07_45c3	172000	**25**	19	**pesiq13_17c1**	1110	**62**	58
pesiq13_45c3	144000	**26**	33	**retm_imp149_17**	1080	**63**	68
pesiq01_17c4	140000	**27**	2	**pesiq12_17c1**	955	**64**	70
pesiq01_17c3	89600	**28**	28	**retm_imp149_45**	904.5	**65**	18
pesiq06_45c3	88400	**29**	50	**retm_op149_17**	813	**66**	10
pesiq07_17c2	69500	**30**	36	**pesiq12_45c1**	737.5	**67**	65
pesiq12_17c2	65300	**31**	34	**retm_app_45**	717.5	**68**	26
pesiq01_45c4	50400	**32**	22	**retm_op149_45**	517.5	**69**	42
pesiq07_45c2	50100	**33**	25	**dir_perc_17**	425	**70**	5
pesiq13_45c1	43800	**34**	49	**retm_dir_17**	324	**71**	12
pesiq12_45c3	43400	**35**	39				
pesiq13_45c2	41700	**36**	37				
pesiq06_17c4	38500	**37**	9				

NR stands for not ranked (0 in the sensitivity analysis)

[17] For a detailed description of IRAP deductions, see Diecofis Deliverable 5.2.

The factors showing the largest overall importance correspond to the correction for fiscal values of *other operating costs* (pesiq13), of *income from sales and services* (pesiq01), and of *Purchases of services* (pesiq07). As expected, factors corresponding to enterprises with large revenue (categories c4-c6) are at the top of the ranking while enterprises with a lower income (cat. c1, c2, c3) are much below, this seems to indicate that enterprises with a large revenue influence IRAP fluctuation more than small firms for both sectors. This quite obvious result, however, highlights the issue that the joint estimation of the fiscal contributions paid by large and medium-small firms (SME) prevents the understanding of the most influential factors affecting SME.

Looking at the pairs (μ^*, σ) in Table A.3, the importance of the 1st, the 8th, and the 9th factors is mainly due to first order or linear effects. Their high overall importance (μ^*) is in fact associated to very low rankings for σ, testifying the negligible presence of curvature or interaction effects of the associated variables. The variables *pesiq13_c5*, *pesiq01_c5* and *pesiq07*_c5 (for both sectors), instead, combine a relatively high score for both μ^* and σ, merging linear to second or higher order effects. These latter are especially important for *pesiq13_17c4* and *pesiq12_17c4*, suggesting a substantial gain from the inclusion in the model structure of these interdependencies.

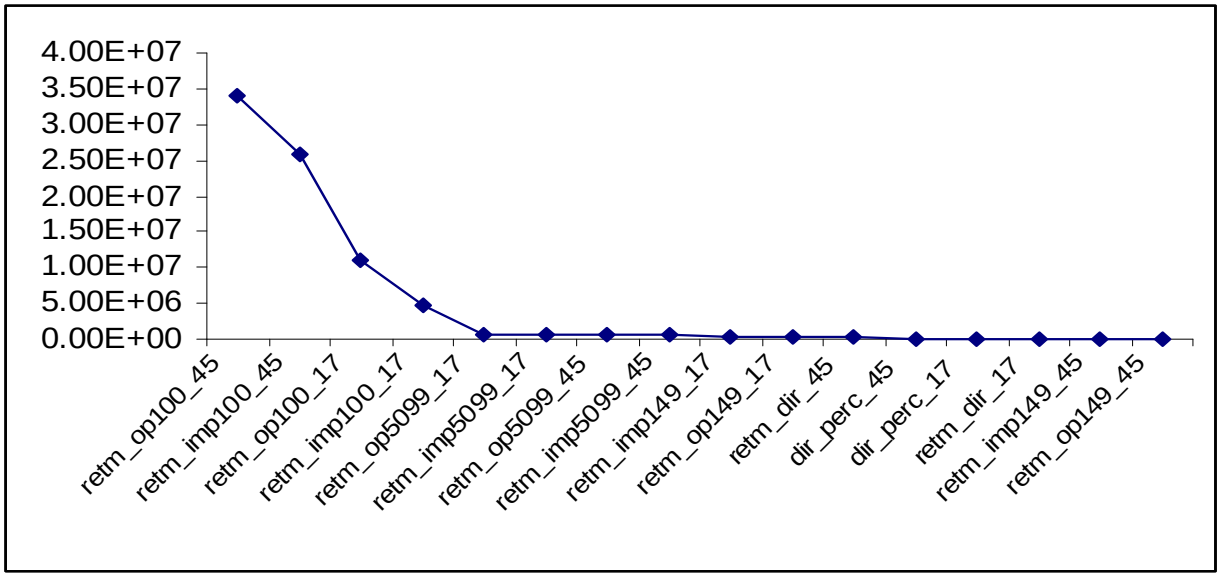
The sensitivity analysis highlights a number of peculiar aspects. A first anomaly is a high ranking for the measure σ, but a low or very low ranking for μ^*. This is the case, for example, of factors *retm_op149_17*, *dir_perc_17*, and *retm_dir_17*. They do not influence IRAP variation whatsoever (they are at the bottom of the μ^* ranking) but they score 10th, 5th, and 12th according to σ, meaning that the second or higher order effects of these variables on output are not negligible. This might be due to the lack of variation of input factors: the range of variability of these three variables is lower than 1%, vis à vis a range of 420% for *pesiq13_45_c6* (the highest in the ranking of μ^*). Clearly inputs with such a low variability will not determine much of the fluctuations of output (from that a very low μ^*), but on the other hand, the absence of input variability will make more visible the relationship with other factors (i.e. the interactions effects) pushing up the measure σ. The calculation of the Morris measure μ highlights the signs of the elementary effects. In Table 10 factors in yellow have a measure μ positive, indicating that output is positively monotonic (output always increases when that factor increases) with respect to pesi_01, while output is negatively monotonic with respect to all other factors. Furthermore, for this model $|\mu| = \mu^*$ thus elementary effects never cancel out, pointing to the absence of effects of different sign depending on the values assumed by other factors.

A.1.4. Sensitivity analysis of SC module

The sensitivity analysis of the social contribution (SC) module is presented in Figure 2 and Table 4. This table contains the ranking of all factors considered in the analysis. Most of them, (namely all factors related to the transformation of survey into fiscal data for IRAP module) have no interaction with SC and, as expected, do not have feedbacks with this module. On the contrary, factors showing interdependency are those used to improve the information on wages.

Figure A.2 shows the ranking of factors in order of importance according to the modified Morris measure μ^*. For sake of simplicity all factors with zero value have been omitted from the plot. It can be noticed that the first 5-7 factors have a sizeable influence on SC fluctuations. These factors propagate most of the variance in the output. The rest display a negligible impact and can be fixed.

Figure A.2 - Results of SA using Morris method for the SC module. Plot of factors ranked according to the measure μ^*.



The factors *retem_op100* and *retem_imp100* for both sectors analysed seem to have the highest importance in explaining SC fluctuations. Those variables for the construction sector (code 45) seem to combine linear and second or higher order effects (high ranking for μ^* and high also for σ), while for the textile sector (code 17) only *retem_imp100* displays high μ^* and high σ, while *retem_op100* mainly shows a linear interaction on SC. Overall, it may be worthwhile a glance to the

correlation structure of these factors. The factors ranking from 5th to the 8th position have a modest first order effect and display almost no curvature or interaction with other factors.

Table A.3 - Results of SA using Morris method for the SC module. Order of importance of factors according to the measure μ^* and associated ranking place for the measure σ

Factor	Value of μ^*	Rank of μ^*	Rank of σ
retm_op100_45	3.39E+07	**1**	7
retm_imp100_45	2.57E+07	**2**	1
retm_op100_17	1.11E+07	**3**	12
retm_imp100_17	4.77E+06	**4**	3
retm_op5099_17	7.71E+05	**5**	16
retm_imp5099_17	6.63E+05	**6**	10
retm_op5099_45	6.46E+05	**7**	15
retm_imp5099_45	4.85E+05	**8**	11
retm_imp149_17	4.19E+05	**9**	9
retm_op149_17	3.29E+05	**10**	8
retm_dir_45	2.10E+05	**11**	5
dir_perc_45	1.45E+05	**12**	2
dir_perc_17	1.38E+05	**13**	4
retm_dir_17	1.16E+05	**14**	6
retm_imp149_45	8.84E+04	**15**	13
retm_op149_45	5.82E+04	**16**	14

NR stands for not ranked (0 in the sensitivity analysis)

Again, since μ^* is an expression of overall relevance of the associated factor and includes the effects "measured" by σ, a high σ associated to a low μ^* could be explained partially by the lack of variability in the inputs factors that makes stand out even a modest interaction with other factors and/or by problems in the setup of the model. As regards the Morris measure μ, for all factors $\mu = \mu^*$ indicating that the sign of the elementary effect is always positive, i.e. the output function is monotonic with respect to all factors.

APPENDIX B

VARIABLES NAMES

RTSS DATASET:VARIABLES LIST AND DESCRIPTION

Assets and Liabilities

Comm Acc	Survey	Database code	name	Description
PA1	11100	cod11100	ric_tot	Income from sales and Services
	11101	cod11101	ric_vpi	Sales of firm products
	11102	cod11102	ric_vmnt	Sales of goods
	11103	cod11103	ric_lct	Works on behalf of third parties
	11104	cod11104	ric_lot	Works and industrial services on orders of third parties
	11105	cod11105	ric_ai	Brokerage activities
	11106	cod11106	ric_tra	Income of transport enterprises
	11107	cod11107	ric_pst	Services to third parties
PA2	11200	cod11200	ric_vr_tot	Variations of the stocks of finished and semi-finished products
	11201	cod11201	ric_vr_pf	Variations of the stocks of products
	11202	cod11202	ric_vr_pcl	Variations of the stocks of under-processing products
PA3	11300	cod11300	ric_vl_co	Variation in contract work in progress
PA4	11400	cod11400	ric_ini_li	Increase on internal work capitalized under fixed assets
PA5	11500	cod11500	ric_alpro	Other income and earnings (neither financial, nor extraordinary)
PA	12000	cod12000	val_prod_tot	Value of production
PB6	12100	cod12100	acq_beni_tot	Purchases
	12101	cod12101	acq_matp	Raw materials
	12102	cod12102	acq_ener	Energy products
	12103	cod12103	acq_mriv	Goods for resale
PB7	12200	cod12200	acq_serv_tot	Services (Total)
	12201	cod12201	acq_lavter	Works made by third parties
	12202	cod12202	acq_allav	Other works made by third parties
	12203	cod12203	acq_trasp	Transport
	12204	cod12204	acq_altr_sci	Other Transport (SCI)
	12205	cod12205	acq_intmd	Brokerages
	12206	cod12206	acq_pubbli	Advertising
	12207	cod12207	acq_risv	Research and Development
	12208	cod12208	acq_consul	Consulting
	12209	cod12209	acq_inform	Informatics
	12210	cod12210	acq_prassi	Insurance premiums
	12211	cod12211	acq_licuso	Licences
	12212	cod12212	acq_smrif	Waste disposal
	12213	cod12213	acq_alser	Other services
	12214	cod12214	acq_banc	Bank services
PB8	12300	cod12300	acq_gdbt_tot	Use of third party assets
	12301	cod12301	acq_fitpa	Rent charges for capital goods
	12302	cod12302	acq_leasing	Leasing expenses
	12303	cod12303	acq_alfitpa_sci	Other rents from buildings (SCI)
	12304	cod12304	acq_canlo	Other rents
	12305	cod12305	acq_leasstr_sci	Leasing expenses for instrumental goods (SCI)
PB9	44000	cod44000	acq_pers_tot	Personnel Expenses (Total)
PB9a	41110	cod41110	acq_ret_dirimp	Wages and salaries: Executives, Employees

Continue...

Comm Acc	Survey	Database code	name	Description
	42110	cod42110	acq_ret_alcat	Wages and salaries: Workers
PB9b	42121	cod42121	acq_cs	Social security contributions (Total)
PB9c	42131	cod42131	acq_qtfr	Annual Provision for Severance Pay (flow)
PB9e	43000	cod43000	acq_prpers	Other personnel costs
PB10	12500	cod12500	acq_amm_tot	Depreciation (PMI), Depreciation and Devaluation(SCI)
PB10a	12510	cod12510	acq_amm_imi	Depreciation of intangible assets
PB10b	12520	cod12520	acq_amm_imm	Depreciation of tangible assets
PB10c	12530	cod12530	acq_svimm_sci	Other write-downs of fixed assets (SCI)"
PB10d	12540	cod12540	acq_svcr_sci	Write-downs of of current credits (SCI)"
PB11	12600	cod12600	acq_vr_tot	Variations of stocks of raw materials and to resale (Tot)
	12601	cod12601	acq_vr_mp	Variations of stocks of raw materials
	12602	cod12602	acq_vr_r	Variations of stocks to resale
PB12	12700	cod12700	acq_accant	Provisions
PB13	12800	cod12800	acq_alacc_sci	Other provisions (SCI)
PB14	12900	cod12900	acq_ondiv_tot	Other operating costs (TOTAL)
	12901	cod12901	acq_forper	Personnel training expenses
	12902	cod12902	acq_onamm	Managers' rewards
	12903	cod12903	acq_aodg	Other operating charges
	12904	cod12904	acq_impfab_sci	Excises taxes on production (SCI)
	12905	cod12905	acq_imp_ind	Indirect taxes on products
	12906	cod12906	acq_alimp_sci	Other indirect taxes (SCI)
PB	13000	cod13000	cost_prod_tot	Costs of production
	13999	cod13999	mol	Gross operating surplus
	14000			Value added
PC15	14100		divid	Income from participating interests
PC16	14200	cod14200	int_att	interest receivable
PC17	14300		int_pas	Interest payable
	14301		sp_intfin_sci	Interest payable for loans (SCI)
	14302		sp_intaltr_sci	Other financial charges (factoring) (SCI)
PC	15000	cod15000	pr_onfin_sci	Interest receivable and payable Total (SCI)
PD18	15100		rival	Positive value adjustments
PD18a			rival_part	Positive value adjustments of shares concerning subsidiaries undertakings
PD18b			rival_imfin	Positive value adjustments of other shares
PD18c			rival_titol	Positive value adjustments in current investments
PD19	15200		sval	Negative value adjustments
PD19a			sval_part	Negative value adjustments of shares concerning subsidiaries undertakings
PD19b			sval_im_fin	Negative value adjustments of other shares
PD19c			sval_tit	Negative value adjustments in current investments
PD	16000	cod16000	ret_valaf_sci	Adjustment for financial assets Total (SCI)
PE20	16100		pr_onstr_pr	Extraordinary income
PE21	16200		pr_onstr_on	Extraordinary charges
PE	17000	cod17000	pr_onstr_sci	Extraordinary proceeds and costs Total (SCI)
PSBT	17999	cod17999	utile_lor_sci	Gross Profit (loss) for the financial year (SCI)
P22	18100	cod18100	imp_reddito	Income taxes
	18500	cod18500	imp_irap	IRAP (PMI)
P23	19000	cod19000	utile_netto	Net Profit (loss) for the financial year

Profit and Loss

Comm Acc	Survey	Database code	name	Decription
	21020		sp_cresoc_sci	Credits: to partners m/l term (SCI)
AB	22000			Total Fixed assets (SCI)
AB1	22100	cod22100	sp_im_im	Intangible fixed assets
AB12	22101		sp_im_rsp	costs of R&D
AB13	22102		sp_im_cop	trade marks
	22103		sp_im_ip	trade marks (intellectual works)
AB15			sp_im_av	Goodwill (BS)
AB14	22104		sp_im_tdm	Concessions / licences
	22105		sp_im_sfw	Software
AB18			sp_fd_im	Sinking Fund: intangible assets (BS)
AB2	22200	cod22200	sp_im_man	Tangible fixed assets (Net of depreciation funds)
	22211		sp_im_ter	fixed assets - land
	22212		sp_im_fab	Instrumental Buildings
	22213		sp_im_fabns	fixed assets: Other buildings
AB22	22220		sp_im_mac	fixed assets - plant and machinery
	22221		sp_im_mt	fixed assets : - means of transport
AB23	22230		sp_im_aic	fixed assets - Equipments
	22241		sp_im_ttd	fixed assets - tools for EDP
	22242		sp_im_maa	fixed assets - furnitures and equipment
	22243		sp_im_mtns	fixed assets - other means of transport
	22244		sp_im_bv	fixed assets - Value goods
AB25	22250		sp_im_inc	Payments on account and tangible assets
AB26			sp_fd_im_tan	Sinking Fund- tangible assets (BS)
AB3	22300	cod22300	sp_im_fi	Financial fixed assets
AB31a	22310		sp_part_cont	Shares in subsidiaries undertakings (fixed assets)
AB31b	22320		sp_part_coll	Shares in Participating interests
AB31d	22330		sp_part_al	Other Shares
AB32a	22340		sp_pre_contr	Loans to subsidiaries undertakings
AB321a	22341		sp_pre_cont_bt	Loans to subsidiaries undertakings - Short term
AB32b	22350		sp_pre_coll	Loans to affiliated undertakings
AB321b	22351		sp_pre_coll_bt	Loans to affiliated undertakings - short term
AB32c	22360		sp_pre_conti	Loans to parents undertakings
AB321c	22361		sp_pre_conti_bt	Loans to parents undertakings - short term
AB32d	22370		sp_pre_at	Other loans
AB321d	22371		sp_pre_al_bt	Other loans - short term
AB33	22380		sp_atit	Other Investments held as fixed assets
AB34	22390		sp_az_pr	Own shares
AC	23000		sp_attcirc_sci	Current assets (SCI)
AC1	23100			Stocks
AC11	23110	cod23110	sp_rim_mp	Current assets: raw materials, ancillars and consumables
AC12	23120	cod23120	sp_rim_ps	Current assets: under-processing and semifinished products
AC13	23130	cod23130	sp_rim_lc	Current assets: works in process under contracts
				Continue...
Comm Acc	**Survey**	**Database code**	**name**	**Decription**

AC14	23140	cod23140	sp_rim_pf	Current assets: finished products and goods
	23141	cod23141	sp_rim_riv	Current assets: goods for resale
AC15	23150		sp_accfor_sci	Payments on account (SCI)
AC2	23200	cod23200	sp_cretot_sci	Total Credits (SCI)
	23260	cod23260	sp_cre_bs	Current assets: short term credits
AC2a	23270	cod23270	sp_cre_ls	Current assets: medium-long term credits
AC31	23310		sp_ac_pct	Shares in subsidiaries undertakings (current assets)
AC32	23320		sp_ac_pcl	Shares in undertakings with which the company is linked by virtue of participating interests
AC34	23330	(AC33+AC34)	sp_ac_pot	Other shares
AC36	23350		sp_af_at	Other investments
AC4	23400	cod23400	sp_liq	Current assets: cash
AC42	23420		sp_liq_ass	Cheques
AD	24000		sp_attrr_sci	Accrued income and prepayments (SCI)
AT	24990		sp_attot_sci	Total Assets (SCI)
LA	25000		pn	Capital and reserves
LA1	25100		cap_soc	Subscribed capital
LA4	25200		ris	Legal reserve
LA8	25300		ut_nuo	Profit or loss brought forward
LA9	25400		ut_es	Profit or loss for the financial year
	25401		ut_es_ris	Profits to cover loss or to reserves
	25402		ut_es_soc	Profits to shareholders
LB	26000	cod26000	sp_foro	Liabilities: provision for liabilities and charges (total)
LC	27000	cod27000	sp_tfr	Liabilities: severance pay fund (stock)
LD	28000	cod28000	sp_debtot_sci	Total debts (SCI)
	28140	cod28140	sp_deb_bs	Liabilities: short term debts (PMI)
LDA	28150	cod28150	sp_deb_ls	Liabilities: medium - long term debts (PMI)
LD1	28010		sp_obb_sci	Debts: bonds (SCI)
LD1a	28011		sp_obb_ls_sci	Debts: bonds - m/l term (SCI)
LD2	28020		sp_obbconv_sci	Debts: Convertible bonds (SCI)
LD2a	28021		sp_obbconv_ls_sci	Debts: Convertible bonds- m/l term (SCI)
LD3	28030		sp_debbanck_sci	Debts: to credit institutions (SCI)
LD3a	28031		sp_debbanck_ls_sci	Debts: to credit institutions - m/l term (SCI)
LD4	28040		sp_debaltr_sci	Debts: to other financial institutions (SCI)
LD4a	28041		sp_debaltr_ls_sci	Debts: to other financial institutions - m/l term (SCI)
LD5	28050		sp_acccl_sci	Debts: customers' acconunts (SCI)
LD6	28060		sp_debfor_sci	Debts: to suppliers (SCI)
LD6a	28061		sp_debfor_ls_sci	Debts: to suppliers - m/l term (SCI)
LD7	28070		sp_debtit_sci	Debts: credit instruments (SCI)
LD7a	28071		sp_sdebtit_ls_sci	Debts: credit instruments m/l term (SCI)
LD8	28080		sp_debcontr_sci	Debts: to controlled companies (SCI)
LD8a	28081		sp_debcontr_ls_sic	Debts: to controlled companies - m/l term (SCI)
LD9	28090		sp_debcoll_sci	Debts: to connected companies (SCI)
LD9a	28091		sp_debcoll_ls_sci	Debts: to connected companies m/l term (SCI)
LD10	28100		sp_control_sci	Debts: to controlling companies (SCI)
LD10a	28101		sp_control_ls_sci	Debts: to controlling companies - m/l term (SCI)
LD11	28110		sp_debtrib_sci	Fiscal Debts (SCI)
LD11a	28111		sp_debtrib_ls_sci	Fiscal Debts - m/l term (SCI)
LD12	28120		sp_debss_sci	Debts: to social security inst. (SCI)
				Continue...

Comm Acc	Survey	Database code	name	Decriptino
LD12a	28121		sp_debss_ls_sci	Debts: to social security inst - m/l term (SCI)
LD13	28130		sp_altrdeb_sci	Other debts (SCI)
LD13a	28131		sp_altrdeb_ls_sci	Other debts - m/l term (SCI)
LT	29990		sp_pastot_sci	Total Liabilities (SCI)

Staff

Survey	Database code	name	Decription
30100	cod30100	occ_td_tot	Short term contracts: total
30101	cod30101	occ_td_f	Short term contracts: women
30102	cod30102	occ_td_ore	Short term contracts : hours
30200	cod30200	occ_tp_tot	Part-time contracts: total
30201	cod30201	occ_tp_f	Part-time contracts: women
30202	cod30202	occ_tp_ore	Part-time contracts: hours
30300	cod30300	occ_fltot_pmi	Training contracts: total (PMI)
30301	cod30301	occ_flf_pmi	Training contracts: women (PMI)
30302	cod30302	occ_flore_pmi	Training contracts: hours (PMI)
31000	cod31000	occ_tot	Total employed staff
31001	cod31001	occ_tot_f	Total employed women
31002	cod31002	occ_tot_ore	Total worked hours
31100	cod31100	occ_imco_tot	Entrepreneurs and Family assistants: total
31101	cod31101	occ_imco_f	Entrepreneurs and Family assistants: women
31110	cod31110	occ_imptot_pmi	Entrepreneurs: total (PMI)
31111	cod31111	occ_impf_pmi	Entrepreneurs: women (PMI)
31120	cod31120	occ_coftot_pmi	Family assistants: total (PMI)
31121	cod31121	occ_coff_pmi	Family assistants: women (PMI)
31122	cod31122	occ_cofore_pmi	Family assistants: hours (PMI)
31200	cod31200	occ_dirimp_tot	Executives and Employees: total
31201	cod31201	occ_dirimp_f	Executives and Employees: women
31202	cod31202	occ_dirimp_ore	Executives and Employees: hours
31300	cod31300	occ_opap_tot	Workers adn Trainees (PMI) AND Workers (SCI)
31301	cod31301	occ_opf_sci	Workers: women (SCI)
31302	cod31302	occ_opore_sci	Workers: hours (SCI)
31310	cod31310	occ_optot_pmi	Workers: total (PMI)
31311	cod31311	occ_opf_pmi	Workers: women (PMI)
31312	cod31312	occ_opore_pmi	Workers: hours (PMI)
31320	cod31320	occ_aptot_pmi	Apprentices: total (PMI)
31321	cod31321	occ_apf_pmi	Apprentices: women (PMI)
31322	cod31322	occ_apore_pmi	Apprentices: hours (PMI)
31400	cod31400	occ_ld_tot	At-home workers: total
31401	cod31401	occ_ld_f	At-home workers: women
31402	cod31402	occ_ld_ore	At-home workers: hours
31500	cod31500	occ_aptot_sci	Apprendices: total (SCI)

Continue...

Survey	Database code	name	Decription
31502	cod31502	occ_apore_sci	Apprendices: hours (SCI)
32100	cod32100	occ_cig_tot	Ordinary lay-off (CIG) hours used
32110	cod32110	occ_cigor_sci	Ordinary lay-off (CIG) hours used (SCI)
32120	cod32120	occ_cigst_sci	Extra-ordinary lay-off (CIG) hours used (SCI)
41110	cod41110	acq_ret_dirimp	Wages and salaries: Executives, Employees
41120	cod41120	acq_csdir_sci	Social security contributions: Executives, Employees (SCI)
41130	cod41130	acq_tfrdir_sci	Annual provision for severance pay: Executives, Employees (SCI)
42110	cod42110	acq_ret_alcat	Wages and salaries: Workers and others (SCI)
42111	cod42111	acq_ret_oppmi	Wages and salaries: Workers (PMI)
42112	cod42112	acq_ret_appmi	Wages and salaries: Apprentices (PMI)
42113	cod42113	acq_ret_ldpmi	Wages and salaries: At-home Workers (PMI)
42120	cod42120	acq_csal_sci	Social security contributions: Workers and others (SCI)
42121	cod42121	acq_cs	Social security contributions (Total)
42130	cod42130	acq_tfral_sci	Annual provision for severance pay: Workers and others (SCI)
42131	cod42131	acq_qtfr	Annual provision for severance pay
43000	cod43000	acq_prpers	Other personnel costs
44000	cod44000	acq_pers_tot	Personnel expenses (Total)
45000	cod45000	acq_indlic	Indemnities for dismissals

Regional

Survey	Database code	name	Decription
70011	cod70011	add_01	Annual workers average (Piemonte)
70012	cod70012	cper_01	Personnel costs (Piemonte)
70021	cod70021	add_02	Annual workers average (Valle d'Aosta)
70022	cod70022	cper_02	Personnel costs (Valle d'Aosta)
70031	cod70031	add_03	Annual workers average (Lombardia)
70032	cod70032	cper_03	Personnel costs (Lombardia)
70051	cod70051	add_04	Annual workers average (Veneto)
70052	cod70052	cper_04	Personnel costs (Veneto)
70061	cod70061	add_05	Annual workers average (Friuli-Venezia Giulia)
70062	cod70062	cper_05	Personnel costs (Friuli-Venezia Giulia)
70071	cod70071	add_06	Annual workers average (Liguria)
70072	cod70072	cper_06	Personnel costs (Liguria)
70081	cod70081	add_07	Annual workers average (Emilia-Romagna)
70082	cod70082	cper_07	Personnel costs (Emilia-Romagna)
70091	cod70091	add_08	Annual workers average (Toscana)
70092	cod70092	cper_08	Personnel costs (Toscana)
70101	cod70101	add_09	Annual workers average (Umbria)
70102	cod70102	cper_09	Personnel costs (Umbria)
70111	cod70111	add_10	Annual workers average (Marche)
70112	cod70112	cper_10	Personnel costs (Marche)
70121	cod70121	add_11	Annual workers average (Lazio)

Continue...

Survey	Database code	name	Decription
70122	cod70122	cper_11	Personnel costs (Lazio)
70131	cod70131	add_12	Annual workers average (Abruzzo)
70132	cod70132	cper_12	Personnel costs (Abruzzo)
70141	cod70141	add_13	Annual workers average (Molise)
70142	cod70142	cper_13	Personnel costs (Molise)
70151	cod70151	add_14	Annual workers average (Campania)
70152	cod70152	cper_14	Personnel costs (Campania)
70161	cod70161	add_15	Annual workers average (Puglia)
70162	cod70162	cper_15	Personnel costs (Puglia)
70171	cod70171	add_16	Annual workers average (Basilicata)
70172	cod70172	cper_16	Personnel costs (Basilicata)
70181	cod70181	add_17	Annual workers average (Calabria)
70182	cod70182	cper_17	Personnel costs (Calabria)
70191	cod70191	add_18	Annual workers average (Sicilia)
70192	cod70192	cper_18	Personnel costs (Sicilia)
70201	cod70201	add_19	Annual workers average (Sardegna)
70202	cod70202	cper_19	Personnel costs (Sardegna)
70211	cod70211	add_20	Annual workers average (Bolzano)
70212	cod70212	cper_20	Personnel costs (Bolzano)
70221	cod70221	add_21	Annual workers average (Trento)
70222	cod70222	cper_21	Personnel costs (Trento)
77231	cod77231	add_est_sci	Annual workers average (Foreign Countries) (SCI)
77232	cod77232	cper_est_sci	Personnel costs (Foreign Countries) (SCI)
77241	cod77421	add_noloc_sci	Annual workers average (Not imputable) (SCI)
77242	cod77242	cper_noloc_sci	Personnel costs (Not imputable) (SCI)

Miscellaneous

Survey	Database code	name	Decription
51110	cod51110	im_acqter	Purchase of land
51120	cod51120	im_cost_tot	Purchase of constructions (Total)
51121	cod51121	im_cost_new	Purchase of constructions : New
51122	cod51122	im_cost_us	Purchase of constructions : Second-hand
51130		im_acq_fns	Purchase of construction (Total)
51200	cod51200	im_mac_tot	Purchase of Machinery: total
51201	cod51201	im_mac_ new	Purchase of Machinery: new
51202	cod51202	im_mac_us	Purchase of Machinery: second hand
51211		im_mtr_new	Purchase of instrumental means of transport (new)
51212		im_mtr_us	Purchase of instrumental means of transport (second hand)
51300		im_at_ic_tot	purchase of industrial and commercial equipment (total)
51301		im_at_ic_new	purchase of industrial and commercial equipment (newl)
51302		im_at_ic_us	purchase of industrial and commercial equipment (second-hand)
51400		im_ab_tot	purchase of other goods (total)
51401		im_ab_new	purchase of other goods (new)
51402		im_ab_us	purchase of other goods (second hand)
51410	cod51410	im_att_tot	Purchase of Data equipment: total
51411	cod51411	im_att_new	Purchase of data equipment: new

Continue...

Survey	Database code	name	Decription
51412	cod51412	im_att_us	Purchase of data equipment: second-hand
51420		im_mob_tot	Purchase of Furniture: total
51421	cod51421	im_mob_new	Purchase of Furniture: new
51422	cod51422	im_mob_us	Purchase of Furniture: second hand
51430	cod51430	im_tras_tot	Purchase of means of Transport : total
51431	cod51431	im_tras_new	Purchase of means of transport.: new
51432	cod51432	im_tras_us	Purchase of means of transport: second-hand
51440		im_bval_tot	Purchase of Valuable assets: total
51441	cod51441	im_bval_new	Purchase of Valuable assets: new
51442	cod51442	im_bval_us	Purchase of Valuable assets: second - hand
51900		im_mat_acq_tot	Purchase of tangible assets (total)
52100		oth_new_in_c	purchase of industrial trademarks
52200	cod52200	im_art_tot	
52201	cod52201	im_art_new	Purchase of Artistic assets: new
52202	cod52202	im_art_us	Purchase of Artistic assets: second - hand
52300		oth_new_in_tdm	purchase of marks (total)
52301		im_conc_new	purchase of trade marks and licenses
52400	cod52400	im_sof_tot	Purchase of softwares:Total
52401	cod52401	im_sof_new	Purchase of softwares: new
52402	cod52402	im_sof_us	Purchase of softwares: second hand
52500	cod52500	oth_new_in_oth	purchase (others)
52900		oth_new_in	purchase of intangible assets
53000		im_acq_tot	Purchase of fixed assets
53001		im_acq_new	Purchase of fixed assets: new
53002		im_acq_us	Purchase of fixed assets: second-hand
60010	cod60010	iva_cli	VAT from customers
60020	cod60020	iva_for	VAT to suppliers
60040	cod60040	exp_ue	Export sales (EU countries)
60050	cod60050	exp_eue	Export sales (extra EU countries)
60070	cod60070	imp_ue	Import sales (EU countries)
60080	cod60080	imp_eue	Import sales (extra EU countries)
60110	cod60110	ptc_ita	shares in italian firms
60120	cod60120	ptc_est	shares in foreign firms
60220		cred_fin_est	financial credits tws foreign firms
60250		cred_com_est	commercial credists tws foreign firms
60620		deb_est_b	debts tws foreign banks
60650		deb_est_i	debts tws foreign firms
61110	cod61110	ind_ass	Insurance compensations
61120	cod61120	fitti_att	Income from rents
61130	cod61130	royal	Revenue from Royalties, patents and similar
61140	cod61140	cont_ese	General Government allowances on working/operating account
61150	cod61150	contr_k	General Government capital allowances
61160	cod61160	contr_i	General Government allowances on interest account
61180	cod61180	capz_rsv	R&D expenses capitalized in the a.p
61200	cod61200	imp_dir	Direct taxes payed in the financial year
61230	cod61230	manord	Routine buildings maintenance
61240	cod61240	v_cap_us	Sales of second - hand capital goods
61250	cod61250	k_leasing	Value of capital in leasing contracts of the financial year
61260	cod61260	q_leasing	Share of financial leasing for the financial year
61265	cod61265	imp_ind	Indirect taxes on production

Continue...

Survey	Database	name	Decription

	code		
61280	cod61280	fatt_ed	Turnover of construction enterprises (for building)
61290	cod61290	fatt_cost	Turnover of building enterprises (for engineering)
61300	cod61300	pers_int	Personnel Expenses for workers from provisional agencies
61310	cod61310	inv_amb	Investment in environmental equipment
REGCONT	REGCONT	REGCONT	Account system

DIECOFIS MODEL ENDOGENOUS VARIABLES

Name	Description	Stata program	label	Stata label program
occ_dir_tot = 0	occ_dirimp_tot*Pesi[`m',`nc1']/**/ if ateco2==Pesi[`m',`nc']	retrib.do	Executives: total	label_retrib.do
occ_imp_tot	occ_imp_tot = occ_dirimp_tot - occ_dir_tot	retrib.do	Employees: total	label_retrib.do
ret_dir=0	ret_dir= occ_dir_tot*Pesi[`m',`nc2'] if ateco2==Pesi[`m',`nc']	retrib.do	Total wages and salaries: executives	label_retrib.do
ret_imp=0	ret_imp=occ_imp_tot*Pesi[`m',`nc3'] if ateco2==Pesi[`m',`nc'] & occ_tot <= 49 ret_imp=occ_imp_tot*Pesi[`m',`nc4'] if ateco2==Pesi[`m',`nc'] & occ_tot >49 & occ_tot <=99 ret_imp=occ_imp_tot*Pesi[`m',`nc5'] if ateco2==Pesi[`m',`nc'] & occ_tot >= 100	retrib.do	Total wages and salaries: employees	label_retrib.do
ret_op=0	ret_op=occ_optot_pmi*Pesi[`m',`nc6'] if ateco2==Pesi[`m',`nc'] & pmi==1 & occ_tot <= 49 ret_op=occ_optot_pmi*Pesi[`m',`nc7'] if ateco2==Pesi[`m',`nc'] & pmi==1 & occ_tot >49 & occ_tot <=99 ret_op=occ_optot_sci*Pesi[`m',`nc8'] if ateco2==Pesi[`m',`nc'] & sci==1 & occ_tot >= 100	retrib.do	Total wages and salaries: workers	label_retrib.do
ret_app=0	ret_app=occ_aptot_pmi*Pesi[`m',`nc9'] if ateco2==Pesi[`m',`nc'] & pmi==1 ret_app=occ_aptot_sci*Pesi[`m',`nc9'] if ateco2==Pesi[`m',`nc'] & sci==1	retrib.do	Total wages and salaries: apprentices	label_retrib.do
retop_rat	retop_rat = acq_ret_oppmi/occ_optot_pmi	retrib.do	Average salary (workers) PMI	label_retrib.do
retap_rat	retap_rat = acq_ret_appmi/occ_aptot_pmi	retrib.do	Average salary (apprentices) PMI	label_retrib.do
oreop_rat	oreop_rat = occ_opore_pmi/occ_optot_pmi	retrib.do	Average worked hours (workers) PMI	label_retrib.do
orem_app_sci	orem_app_sci=occ_apore_sci/occ_aptot_sci	contrib.do	Average hours for apprentices (SCI)	label_contrib.do
orem_app_pmi	orem_app_pmi=occ_apore_pmi/occ_aptot_pmi	contrib.do	Average hours for apprentices (PMI)	label_contrib.do
set_app_sci	set_app_sci=orem_app_sci/35	contrib.do	Worked weeks for apprentices (SCI)	label_contrib.do
set_app_pmi	set_app_pmi=orem_app_pmi/35	contrib.do	Worked weeks for apprentices (PMI)	label_contrib.do
inail_tot_irap	inail_tot_irap = contri4_op + contri4_dir + contri4_imp	contrib.do	Invalidity contributions (INAIL): Total	label_contrib.do
app_tot_irap	app_tot_irap = contri_app_tot + ret_app	contrib.do	Apprentices total labour cost	label_contrib.do
ret_tot	ret_tot=ret_imp+ret_op+ret_app+ret_dir	contrib.do	Total wages and salaries: all worker categories	label_contrib.do
ac_tfr1	ac_tfr1=ret_tot*0.0691	contrib.do	Provision for severance pay fund (internal)	label_contrib.do
ac_tfr2	ac_tfr2=ret_tot*0.005	contrib.do	Provision for severance pay fund (for INPS)	label_contrib.do
ac_acctfr	ac_acctfr= ac_tfr1+ac_tfr2	contrib.do	Total provision for severance pay fund	label_contrib.do
riv_tfr`year'	riv_tfr`year'=sp_tfr*(0.015+0.75*0.0373)	contrib.do	Annual revaluation of severance pay stock	label_contrib.do
contri1_op	contri1_op=ret_`1' *Ali`1'[`m',`nc1'] if ateco2==Ali`1'[`m',`nc']	contrib.do	Various Social Contributions (sickness, maternity, etc.): workers	label_contrib.do
				Continue...

Name	Description	Stata program	label	Stata label program
contri1_dir	contri1_dir =ret_`1' *Ali`1'[`m',`nc2'] if ateco2==Ali`1'[`m',`nc']	contrib.do	Various Social Contributions (sickness, maternity, etc.): executives	label_contrib.do
contri1_imp	contri1_imp= ret_`1' * Ali`1'[`m',`nc3'] if ateco2==Ali`1'[`m',`nc'] & occ_tot <= 15	contrib.do	Various Social Contributions (sickness, maternity, etc.): employees	label_contrib.do
contri1_app	contri1_app =ret_`1' *Ali`1'[`m',`nc6'] if ateco2==Ali`1'[`m',`nc']	contrib.do	Various Social Contributions (sickness, maternity, etc.): apprentices	label_contrib.do
contri2_op	contri2_op =ret_`1' *Ali`1'[`m',`nc1'] if ateco2==Ali`1'[`m',`nc']	contrib.do	Old-age social contributions (IVS): workers	label_contrib.do
contri2_dir	contri2_dir =ret_`1' *Ali`1'[`m',`nc2'] if ateco2==Ali`1'[`m',`nc']	contrib.do	Old-age social contributions (IVS): executives	label_contrib.do
contri2_imp	contri2_imp = ret_`1' * Ali`1'[`m',`nc3'] if ateco2==Ali`1'[`m',`nc'] & occ_tot <= 15	contrib.do	Old-age social contributions (IVS): employees	label_contrib.do
contri2_app	contri2_app=ret_`1' *Ali`1'[`m',`nc6'] if ateco2==Ali`1'[`m',`nc']	contrib.do	Old-age social contributions (IVS): apprentices	label_contrib.do
contri3_op	contri3_op =ret_`1' *Ali`1'[`m',`nc1'] if ateco2==Ali`1'[`m',`nc']	contrib.do	Occupational Desease (CIG): workers	label_contrib.do
contri3_dir	contri3_dir =ret_`1' *Ali`1'[`m',`nc2'] if ateco2==Ali`1'[`m',`nc']	contrib.do	Occupational Desease (CIG):executives	label_contrib.do
contri3_imp	contri3_imp = ret_`1' * Ali`1'[`m',`nc3'] if ateco2==Ali`1'[`m',`nc'] & occ_tot <= 15	contrib.do	Occupational Desease (CIG): employees	label_contrib.do
contri3_app	contri3_app=ret_`1' *Ali`1'[`m',`nc6'] if ateco2==Ali`1'[`m',`nc']	contrib.do	Occupational Desease (CIG): apprentices	label_contrib.do
contri4_op	contri4_op =ret_`1' *Ali`1'[`m',`nc1'] if ateco2==Ali`1'[`m',`nc']	contrib.do	Invalidity contributions (INAIL): workers	label_contrib.do
contri4_dir	contri4_dir =ret_`1' *Ali`1'[`m',`nc2'] if ateco2==Ali`1'[`m',`nc']	contrib.do	Invalidity contributions (INAIL): executives	label_contrib.do
contri4_imp	contri4_imp = ret_`1' * Ali`1'[`m',`nc3'] if ateco2==Ali`1'[`m',`nc'] & occ_tot <= 15	contrib.do	Invalidity contributions (INAIL): employees	label_contrib.do
contri4_app	contri4_app=ret_`1' *Ali`1'[`m',`nc6'] if ateco2==Ali`1'[`m',`nc']	contrib.do	Invalidity contributions (INAIL): apprentices	label_contrib.do
contri_op_tot	contri_op_tot=contri1_op+contri2_op+contri3_op+contri4_op	contrib.do	Total Social Contributions: worker	label_contrib.do
contri_dir_tot	contri_dir_tot=contri1_dir+contri2_dir+contri3_dir+contri4_dir	contrib.do	Total Social Contributions: executives	label_contrib.do
contri_imp_tot	contri_imp_tot=contri1_imp+contri2_imp+contri3_imp+contri4_imp	contrib.do	Total Social Contributions: employees	label_contrib.do
contri_app_tot	contri_app_tot=contri1_app+contri2_app+contri3_app+contri4_app	contrib.do	Total Social Contributions: apprentices	label_contrib.do
Pcontri`1'_op	Pcontri`1'_op = (contri`1'_op /1000)* peso	stat_contrib.do	Various Social Contributions (sickness, maternity, etc.) (WEIGHTED): workers	label_stat_contrib.do
Pcontri`1'_dir	Pcontri`1'_dir =(contri'1'_dir/1000)*peso	stat_contrib.do	Various Social Contributions (sickness, maternity, etc.) (WEIGHTED): executives	label_stat_contrib.do
Pcontri`1'_imp	Pcontri'1'_imp=(contri'1'_imp/1000)*peso	stat_contrib.do	Various Social Contributions (sickness, maternity, etc.) (WEIGHTED): employees	label_stat_contrib.do
Pcontri`1'_app	Pcontri'1'_app=(contri'1'_app/1000)*peso	stat_contrib.do	Various Social Contributions (sickness, maternity, etc.) (WEIGHTED): apprentices	label_stat_contrib.do
				Continue...

Name	Description	Stata program	label	Stata label program
Pcontri`1'_tot	Pcontri`1'_tot = Pcontri`1'_op + Pcontri`1'_dir + Pcontri`1'_imp	stat_contrib.do	Various Social Contributions (sickness, maternity, etc.) (WEIGHTED): TOTAL	label_stat_contrib.do
Pcontri`2'_op	Pcontri`2'_op = (contri`2'_op /1000)* peso	stat_contrib.do	Old-age social contributions (IVS) (WEIGHTED): workers	label_stat_contrib.do
Pcontri`2'_dir	Pcontri`2'_dir =(contri'2'_dir/1000)*peso	stat_contrib.do	Old-age social contributions (IVS) (WEIGHTED): executives	label_stat_contrib.do
Pcontri`2'_imp	Pcontri'2'_imp=(contri'2'_imp/1000)*peso	stat_contrib.do	Old-age social contributions (IVS) (WEIGHTED): employees	label_stat_contrib.do
Pcontri`2'_app	Pcontri'2'_app=(contri'2'_app/1000)*peso	stat_contrib.do	Old-age social contributions (IVS) (WEIGHTED): apprentices	label_stat_contrib.do
Pcontri`2'_tot	Pcontri`2'_tot = Pcontri`2'_op + Pcontri`2'_dir + Pcontri`2'_imp	stat_contrib.do	Old-age social contributions (IVS) (WEIGHTED): TOTAL	label_stat_contrib.do
Pcontri`3'_op	Pcontri`3'_op = (contri`3'_op /1000)* peso	stat_contrib.do	Occupational Desease (CIG) (WEIGHTED): workers	label_stat_contrib.do
Pcontri`3'_dir	Pcontri`3'_dir =(contri'3'_dir/1000)*peso	stat_contrib.do	Occupational Desease (CIG) (WEIGHTED): executives	label_stat_contrib.do
Pcontri`3'_imp	Pcontri'3'_imp=(contri'3'_imp/1000)*peso	stat_contrib.do	Occupational Desease (CIG) (WEIGHTED): employees	label_stat_contrib.do
Pcontri`3'_app	Pcontri'3'_app=(contri'3'_app/1000)*peso	stat_contrib.do	Occupational Desease (CIG) (WEIGHTED): apprentices	label_stat_contrib.do
Pcontri`3'_tot	Pcontri`3'_tot = Pcontri`3'_op + Pcontri`3'_dir + Pcontri`3'_imp	stat_contrib.do	Occupational Desease (CIG) (WEIGHTED): TOTAL	label_stat_contrib.do
Pcontri`4'_op	Pcontri`4'_op = (contri`4'_op /1000)* peso	stat_contrib.do	Invalidity contributions (INAIL) (WEIGHTED): workers	label_stat_contrib.do
Pcontri`4'_dir	Pcontri`4'_dir =(contri'4'_dir/1000)*peso	stat_contrib.do	Invalidity contributions (INAIL) (WEIGHTED): executives	label_stat_contrib.do
Pcontri`4'_imp	Pcontri'4'_imp=(contri'4'_imp/1000)*peso	stat_contrib.do	Invalidity contributions (INAIL) (WEIGHTED): employees	label_stat_contrib.do
Pcontri`4'_app	Pcontri'4'_app=(contri'4'_app/1000)*peso	stat_contrib.do	Invalidity contributions (INAIL) (WEIGHTED): apprentices	label_stat_contrib.do
Pcontri`4'_tot	Pcontri`4'_tot = Pcontri`4'_op + Pcontri`4'_dir + Pcontri`4'_imp	stat_contrib.do	Invalidity contributions (INAIL) (WEIGHTED): TOTAL	label_stat_contrib.do
Pcontri_op_tot	Pcontri_op_tot = (contri_op_tot /1000) * peso	stat_contrib.do	Total Social Contributions (WEIGHTED): workers	label_stat_contrib.do
Pcontri_dir_tot	Pcontri_dir_tot = (contri_dir_tot /1000) * peso	stat_contrib.do	Total Social Contributions (WEIGHTED): executives	label_stat_contrib.do
Pcontri_imp_tot	Pcontri_imp_tot = (contri_imp_tot /1000) * peso	stat_contrib.do	Total Social Contributions (WEIGHTED): employees	label_stat_contrib.do
Pcontri_app_tot	Pcontri_app_tot = (contri_app_tot /1000) * peso	stat_contrib.do	Total Social Contributions (WEIGHTED): apprentices	label_stat_contrib.do
Pac_tfr1	Pac_tfr1= (ac_tfr1 /1000)* peso	stat_contrib.do	Provision for severance pay fund (internal) (WEIGHTED)	label_stat_contrib.do
Pac_tfr2	Pac_tfr2= (ac_tfr2 /1000)* peso	stat_contrib.do	Provision for severance pay fund (for INPS) (WEIGHTED)	label_stat_contrib.do
Pac_acctfr	Pac_acctfr = (ac_acctfr /1000) * peso	stat_contrib.do	Total provision for severance pay fund (WEIGHTED)	label_stat_contrib.do
Priv_tfr`year'	Priv_tfr`year' = (riv_tfr`year' /1000) * peso	stat_contrib.do	Annual revaluation of severance pay stock (WEIGHTED)	label_stat_contrib.do
Pstock_tfr=	Pstock_tfr= (sp_tfr /1000) *peso	stat_contrib.do	Severance pay stock (WEIGHTED)	label_stat_contrib.do
impred_ric_rat	impred_ric_rat = (imp_reddito/ric_tot)*100	stat_irap.do	Income Taxes/Enterprise Total Revenue	label_stat_irap.do
impred_va_rat	impred_va_rat = (imp_reddito/valagg)*100	stat_irap.do	Income Taxes/Gross Value Added	label_stat_irap.do
irap_ric_rat	irap_ric_rat = (Irap/ric_tot)*100 if Irap>0	stat_irap.do	IRAP revenue/Enterprise Total Revenue	label_stat_irap.do
irap_va_rat	irap_va_rat = (Irap/valagg)*100 if Irap>0	stat_irap.do	IRAP revenue/Gross Value Added	label_stat_irap.do
irap_ut_rat	irap_ut_rat = (Irap/utile_lordo)*100 if Irap>0	stat_irap.do	IRAP revenue/Gross Profit (loss)	label_stat_irap.do
irap_int_rat	irap_int_rat = (Irap/int_pas)*100 if Irap>0	stat_irap.do	IRAP revenue/Interest Payable	label_stat_irap.do
irap_pers_rat	irap_pers_rat = (Irap/acq_pers_tot)*100 if Irap>0	stat_irap.do	IRAP revenue/Total Personnel Costs	label_stat_irap.do
sp_deb_rat	sp_deb_rat = sp_deb_tot/ric_tot	stat_irap.do	Total Debt/Total Revenues	label_stat_irap.do
deb_loss_rat	deb_loss_rat = sp_deb_tot/sp_pastot_sci	stat_irap.do	Total Debt/Total Loss	label_stat_irap.do
irap_alimp_rat	irap_alimp_rat = Irap/imp_reddito if Irap>0	stat_irap.do	IRAP Revenue/Total Income Taxes	label_stat_irap.do
Pimp_irap	Pimp_irap= (imp_irap*peso)/1000	stat_irap.do	IRAP (PMI) (Weighted)	label_stat_irap.do
PIrap	PIrap = (Irap*peso)/1000	stat_irap.do	IRAP Total Revenue (Weighted)	label_stat_irap.do
PIrapciv	PIrapciv = (Irapciv*peso)/1000	stat_irap.do	IRAP Total Revenue (civilistic tax base) (Weighted)	label_stat_irap.do
Pimp_reddito	Pimp_reddito = (imp_reddito*peso)/1000	stat_irap.do	Income taxes (Weighted)	label_stat_irap.do
Pric_tot	Pric_tot = (ric_tot*peso)/1000	stat_irap.do	Income from sales and Services (Weighted)	label_stat_irap.do

Continue...

Name	Description	Stata program	label	Stata label program
Pvalagg	Pvalagg = (valagg*peso)/1000	stat_irap.do	Value Added (Weighted)	label_stat_irap.do
Putile_lordo	Putile_lordo = (utile_lordo*peso)/1000	stat_irap.do	Gross Profit (Loss) (Weighted)	label_stat_irap.do
Pacq_pers_tot	Pacq_pers_tot =(acq_pers_tot*peso)/1000	stat_irap.do	Personnel Expenses (Total) (Weighted)	label_stat_irap.do
Pint_pas	Pint_pas = (int_pas*peso)/1000	stat_irap.do	Interest payable (Weighted)	label_stat_irap.do
sp_deb_ls_sci	sp_deb_ls_sci = sp_obb_ls_sci + sp_obbconv_ls_sci + sp_debbank_ls_sci + sp_debaltr_ls_sci + sp_debfor_ls_sci + sp_debtit_ls_sci + sp_debcontr_ls_sci + sp_debcoll_ls_sci + sp_control_ls_sci + sp_debtrib_ls_sci + sp_debss_ls_sci + sp_altrdeb_ls_sci	check.do	Liabilities: long term debts (SCI)	label_check.do
sp_deb_bs_sci	sp_deb_bs_sci = sp_debtot_sci - sp_deb_ls_sci	check.do	Liabilities: short term debts (SCI)	label_check.do
sp_debtot_pmi	sp_debtot_pmi = sp_deb_ls + sp_deb_bs	check.do	Liabilities: total debts (PMI)	label_check.do
sp_debfin_sci	sp_debfin_sci = sp_obb_sci + sp_obbconv_sci + sp_debbank_sci + sp_debaltr_sci + sp_debtit_sci	check.do	Liabilities: Financial Debts (SCI)	label_check.do
sp_immtot_pmi	sp_immtot_pmi = sp_im_im + sp_im_man + sp_im_fi	check.do	Fixed assets: Total	label_check.do
sp_rimtot_pmi	sp_rimtot_pmi = sp_rim_mp + sp_rim_ps + sp_rim_lc + sp_rim_pf	check.do	Current assets: raw materials, finished and semifinished products (PMI)"	label_check.do
sp_rimtot_sci	sp_rimtot_sci = sp_rim_mp + sp_rim_ps + sp_rim_lc + sp_rim_pf + sp_accfor_sci	check.do	Current assets: raw materials, finished and semifinished products (SCI)"	label_check.do
sp_attcirc_pmi	sp_attcirc_pmi = sp_rimtot_pmi + sp_cre_bs + sp_liq	check.do	Current assets (PMI)"	label_check.do
sp_attivo_sci	sp_attivo_sci = sp_attot_sci - sp_attrr_sci	check.do	Total liabilities free of prepayment and accrued income (SCI)"	label_check.do
sp_attivo_pmi	sp_attivo_pmi = sp_immtot_pmi + sp_attcirc_pmi + sp_cre_ls	check.do	Total liabilities free of prepayment and accrued income (PMI)"	label_check.do
sp_cretot_pmi	sp_cretot_pmi = sp_cre_bs + sp_cre_ls	check.do	Current assets: short, medium and long term credits (PMI)"	label_check.do
valagg	valagg = val_prod_tot - acq_beni_tot -acq_serv_tot - acq_gdbt_tot - acq_vr_tot	check.do	Value Added	label_check.do
utile_lordo	utile_lordo = utile_netto + imp_reddito	check.do	Gross Profit (Loss)"	label_check.do
Exp_tot	Exp_tot = exp_ue + exp_eue	check.do	Total Exports	label_check.do
Exp_01	generate byte Exp_01 = Exp_tot>0	check.do	Indicator variable for exporting firms	label_check.do
Inv_01	generate byte Inv_01 = im_acq_tot>0	check.do	Indicator variable for investing firms	label_check.do
Subfa_tot	Subfa_tot = ric_lct + ric_lot	check.do	Total Income from works on behalf of third parties (subcontracting)"	label_check.do
Subfa_01	generate byte Subfa_01 = Subfa_tot > 0	check.do	Indicator variable for subcontracting firms (revenues)"	label_check.do
Subfp_tot	Subfp_tot =acq_lavter+acq_allav	check.do	Total Cost from works made by third parties (subcontracting)"	label_check.do
Subfp_01	generate byte Subfp_01 = Subfp_tot >0	check.do	Indicator variable for subcontracting firms (costs)"	label_check.do
occ_optot_sci	occ_optot_sci = occ_opap_tot - occ_optot_pmi - occ_aptot_pmi if sci==1 & occ_opap_tot>0	check.do	Workers: Total (SCI)"	label_check.do
occ_dirimp_ore_sci	occ_dirimp_ore_sci = occ_dirimp_ore if sci==1	check.do	Executives and Employees: Hours (SCI)	label_check.do
occ_dirimp_ore_pmi	occ_dirimp_ore_pmi = occ_dirimp_ore if pmi==1	check.do	Executives and Employees: Hours (PMI)	label_check.do
occ_tot_ore_sci	occ_tot_ore_sci = occ_tot_ore if sci==1	check.do	Total worked hours (SCI)"	label_check.do
occ_tot_ore_pmi	occ_tot_ore_pmi = occ_tot_ore if pmi==1	check.do	Total worked hours (PMI)"	label_check.do
cla9[1]	gen cla9=recode(occ_tot,9,19,49,99,199,249,499,1000,2000)	check.do	Employees Classes (detailed)	label_check.do
cla3[2]	gen cla3=recode(occ_tot,49,250,2000)	check.do	Employees Classes (aggregate)	label_check.do
Ric_tot	Ric_tot = ric_tot (fiscal adjustment)	taxvar.do	Temporary variable	
Ric_vr_tot	Ric_vr_tot =ric_vr_tot (fiscal adjustment)	taxvar.do	Temporary variable	
Ric_vl_co	Ric_vl_co = ric_vl_co (fiscal adjustment)	taxvar.do	Temporary variable	
				Continue...

Name	Description	Stata program	label	Stata label program
Ric_ini_li	Ric_ini_li =ric_ini_li (fiscal adjustment)	taxvar.do	Temporary variable	
Ric_alpro	Ric_alpro = ric_alpro (fiscal adjustment)	taxvar.do	Temporary variable	
Acq_beni_tot	Acq_beni_tot = acq_beni_tot (fiscal adjustment)	taxvar.do	Temporary variable	
Acq_serv_tot	Acq_serv_tot = acq_serv_tot (fiscal adjustment)	taxvar.do	Temporary variable	
Acq_gdbt_tot	Acq_gdbt_tot = acq_gdbt_tot (fiscal adjustment)	taxvar.do	Temporary variable	
Acq_amm_imm	Acq_amm_imm = acq_amm_imm (fiscal adjustment)	taxvar.do	Temporary variable	
Acq_amm_imi	Acq_amm_imi = acq_amm_imi (fiscal adjustment)	taxvar.do	Temporary variable	
Acq_vr_tot	Acq_vr_tot = acq_vr_tot (fiscal adjustment)	taxvar.do	Temporary variable	
Acq_accant_tot	Acq_accant_tot = acq_accant_tot (fiscal adjustment)	taxvar.do	Temporary variable	
Acq_ondiv_tot	Acq_ondiv_tot = acq_ondiv_tot (fiscal adjustment)	taxvar.do	Temporary variable	
Ded_inail_base	Ded_inail_base = 0	taxvar.do	EXOGENOUS INAIL deductions from IRAP tax base	label_irap.do
Ded_app_base	Ded_app_base = 0	taxvar.do	EXOGENOUS apprentices deductions from IRAP tax base	label_irap.do
Ded_fl_base	Ded_fl_base = 0	taxvar.do	EXOGENOUS trainees deductions from IRAP tax base	label_irap.do
Ded_lav_tot	Ded_lav_tot = inail_tot_irap + app_tot_irap + Ded_fl_base	irap.do	Total deductions from IRAP tax base (labour cost components)	label_irap.do
Ded_lav_b_tot	Ded_lav_b_tot = Ded_inail_base + Ded_app_base + Ded_fl_base	irap.do	EXOGENOUS total deductions from IRAP tax base (labour cost components)	label_irap.do
Ded_irap_altre	Ded_irap_altre=0	irap.do	Additional deductions from IRAP tax base	label_irap.do
Comp_pos_tot	Comp_pos_tot = Ric_tot + Ric_vr_tot + Ric_vl_co + Ric_ini_li + Ric_alpro	irap.do	Total fiscal positive components of IRAP tax base	label_irap.do
Comp_neg_tot	Comp_neg_tot = Acq_beni_tot + Acq_serv_tot + Acq_gdbt_tot + Acq_amm_imm + Acq_amm_imi + Acq_vr_tot + Acq_accant_tot + Acq_ondiv_tot	irap.do	Total fiscal negative components of IRAP tax base	label_irap.do
Base_irap_lorda	Base_irap_lorda =0	irap.do	IRAP Fiscal Gross Tax Base	label_irap.do
"""	replace Base_irap_lorda = Comp_pos_tot - Comp_neg_tot if fgcat==2	irap.do	IRAP Fiscal Gross Tax Base	label_irap.do
comp_pos_tot	comp_pos_tot = ric_tot + ric_vr_tot + ric_vl_co + ric_ini_li + ric_alpro	irap.do	Total civilistic positive components of IRAP tax base	label_irap.do
comp_neg_tot	comp_neg_tot = acq_beni_tot + acq_serv_tot + acq_gdbt_tot + acq_amm_imm + acq_amm_imi + /* */acq_vr_tot + acq_accant_tot + acq_ondiv_tot	irap.do	Total civilistic negative components of IRAP tax base	label_irap.do
base_irap_lorda	base_irap_lorda = comp_pos_tot - comp_neg_tot	irap.do	IRAP civilistic Gross Tax Base	label_irap.do
Base_irap_netta	Base_irap_netta = Base_irap_lorda - Ded_lav_b_tot - Ded_irap_altre	irap.do	IRAP Fiscal Net Tax Base	label_irap.do
base_irap_netta	base_irap_netta = base_irap_lorda - Ded_lav_b_tot - Ded_irap_altre	irap.do	IRAP civilistic Net Tax Base	label_irap.do
Irap	Irap = `Aliq_irap' * Base_irap_netta	irap.do	IRAP Total Revenue	label_irap.do
Irapciv	Irapciv = `Aliq_irap' * base_irap_netta	irap.do	IRAP Total Revenue (civilistic tax base)	label_irap.do
basecat	basecat=0 if Base_irap_lorda==0 ;replace basecat=1 if Base_irap_lorda<0 , replace basecat=2 if Base_irap_lorda>0	irap.do	Gross Irap Tax Base	label_irap.do
basecatn	basecatn=0 if Base_irap_netta==0 ; replace basecatn=1 if Base_irap_netta<0 basecatn==2 if Base _irap_netta>0	irap.do	Net Irap Tax Base	label_irap.do
acq_prpers_ded	acq_prpers_ded = acq_pers_tot*par_fisc_adj[1,`pers_par']	fiscal_adj.do	temporary variable used to simulate fiscal personnel expenses	
acq_pers_tot_sim	acq_pers_tot_sim = acq_pers_tot + (acq_prpers_ded - acq_prpers)	fiscal_adj.do	simulated fiscal personnel expenses	label_fiscal_adj.do

Continue...

Name	Description	Stata program	label	Stata label program
dev_cr_sim	dev_cr_sim= par_fisc_adj[1,`sval_cre_par'] *((sp_cretot_sci+sp_liq_ass)+acq_svcr_sci)	fiscal_adj.do	simulated fiscal credits devaluation	label_fiscal_adj.do
cis_sh_1	cis_sh_1=((sp_im_cop - oth_new_in_c)+(sp_im_ip - im_art_tot)+(sp_im_sfw - im_sof_tot))/(sp_im_im - oth_new_in)	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (copyright, intellectual property, software)	
cis_sh_2	cis_sh_2=(sp_im_cop + sp_im_ip + sp_im_sfw)/sp_im_im	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (copyright, intellectual property, software)	
fd_cis_sh	fd_cis_sh=cis_sh_1*sp_fd_im	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (copyright, intellectual property, software)	
dep_sh_cis	dep_sh_cis =(sp_im_cop + sp_im_ip+ sp_im_sfw)/sp_im_im	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (copyright, intellectual property, software)	
quot_dep_cis	quot_dep_cis =dep_sh_cis*acq_amm_imi	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (copyright, intellectual property, software)	
cis_hc	cis_hc = fd_cis_sh + (sp_im_cop + sp_im_ip+ sp_im_sfw) + quot_dep_cis	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (copyright, intellectual property, software	
amm_cis	amm_cis = cis_hc/3	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (copyright, intellectual property, software	
tl_sh	tl_sh =(sp_im_tdm - oth_new_in_tdm)/(sp_im_im - oth_new_in)	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (copyright, intellectual property, software	
fd_tl_sh	fd_tl_sh =tl_sh*sp_fd_im	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (copyright, intellectual property, software	
amm_sh_tl	amm_sh_tl =(sp_im_tdm)/sp_im_im	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (copyright, intellectual property, software	
quot_amm_tl	quot_amm_tl =amm_sh_tl*acq_amm_imi	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (copyright, intellectual property, software	
tl_hc	tl_hc =fd_tl_sh + (sp_im_tdm) + quot_amm_tl	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (copyright, intellectual property, software	
amm_tl	amm_tl =tl_hc/10	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (copyright, intellectual property, software	
amm_gdw	amm_gdw =sp_im_av/10	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (goodwill)	
incd_rsv	incd_rsv =acq_risv/(acq_risv+acq_pubbli)	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (research & development)	
cs_rsv	cs_rsv =(incd_rsv*sp_im_rsp)-capz_rsv	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (research & development)	
amm_risv_ded	amm_risv_ded = cs_rsv/3	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets (research & development)	
amm_im_in_tot	amm_im_in_tot = amm_cis + amm_tl + amm_gdw + amm_risv_ded	fiscal_adj.do	temporary variable used when simulating fiscal depreciation of intangible assets	
amm_im_in	amm_im_in = amm_im_in_tot	fiscal_adj.do	simulated fiscal immaterial goods depreciation	label_fiscal_adj.do
quot_amm_fab	quot_amm_fab = sp_im_fab/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
cs_fab_amm	cs_fab_amm = quot_amm_fab*sp_fd_im_tan + sp_im_fab	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
				Continue...

Name	Description	Stata program	label	Stata label program
quot_amm_fabns	quot_amm_fabns = (0.5*sp_im_fabns)/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
cs_fabns_amm	cs_fabns_amm = quot_amm_fabns*sp_fd_im_tan + 0.5*sp_im_fabns	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
sp_im_tan_im_sim	sp_im_tan_im_sim = sp_im_mac - sp_im_mt	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
quot_amm_im	quot_amm_im = sp_im_tan_im_sim/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
cs_im_amm	cs_im_amm = quot_amm_im*sp_fd_im_tan + sp_im_tan_im_sim	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
quot_amm_mt	quot_amm_mt = sp_im_mt/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
cs_mt_amm	cs_mt_amm = quot_amm_mt*sp_fd_im_tan + sp_im_mt	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
quot_amm_aic	quot_amm_aic = sp_im_aic/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
cs_aic_amm	cs_aic_amm = quot_amm_aic*sp_fd_im_tan + sp_im_aic	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
quot_amm_ot_maa	quot_amm_ot_maa = sp_im_maa/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
cs_ot_maa_amm	cs_ot_maa_amm = quot_amm_ot_maa*sp_fd_im_tan + sp_im_maa	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
quot_amm_ot_mtns	quot_amm_ot_mtns = (0.5*sp_im_mtns)/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
cs_ot_mtns_amm	cs_ot_mtns_amm = quot_amm_ot_mtns*sp_fd_im_tan + 0.5*sp_im_mtns	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
amm_im_tan_old	amm_im_tan_old = (cs_fab_amm-im_cost_tot)*(par_fisc_adj[1,`aliq_amm_fab'])+(cs_fabns_amm-im_acq_fns/2)*(par_fisc_adj[1,`aliq_amm_fabns'])+ (cs_im_amm-im_mac_tot-im_mtr_new-im_mtr_us) * (par_fisc_adj[1,`aliq_amm_im']) + (cs_mt_amm-im_mtr_new-im_mtr_us) * (par_fisc_adj[1,`aliq_amm_mt']) + (cs_aic_amm-im_at_ic_tot) * (par_fisc_adj[1,`aliq_amm_aic']) + (cs_ot_ttd_amm-im_att_new-im_att_us) * (par_fisc_adj[1,`aliq_amm_ot_ttd'])+ (cs_ot_maa_amm-im_mob_new-im_mob_us) * (par_fisc_adj[1,`aliq_amm_ot_maa']) + (cs_ot_mtns_amm-im_tras_new/2-im_tras_us/2) * (par_fisc_adj[1,`aliq_amm_ot_mtns'])	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	

Continue...

Name	Description	Stata program	label	Stata label program
amm_im_tan_acq	amm_im_tan_acq = (im_cost_tot)*(par_fisc_adj[1,`aliq_amm_fab'])/2 +*/ (im_acq_fns/2) * (par_fisc_adj[1,`aliq_amm_fabns'])/2 + (im_mac_tot+im_mtr_new+im_mtr_us)*(par_fisc_adj[1,`aliq_amm_im'])/2 + (im_mtr_new+im_mtr_us)*(par_fisc_adj[1,`aliq_amm_mt'])/2 + (im_at_ic_tot) * (par_fisc_adj[1,`aliq_amm_aic'])/2 + (im_att_new+im_att_us) * (par_fisc_adj[1,`aliq_amm_ot_ttd'])/2 + (im_mob_new+im_mob_us) * (par_fisc_adj[1,`aliq_amm_ot_maa'])/2 + (im_tras_new/2+im_tras_us/2) * (par_fisc_adj[1,`aliq_amm_ot_mtns'])/2	fiscal_adj.do	temporary variable used when simulating ordinary fiscal depreciation of tangible assets	
acc_amm	(im_cost_tot) * (par_fisc_adj[1,`aliq_amm_fab']/2) +(im_acq_fns/2) * (par_fisc_adj[1,`aliq_amm_fabns']/2) +(im_mac_tot+im_mtr_new+im_mtr_us) * (par_fisc_adj[1,`aliq_amm_im']/2)+(im_mtr_new+im_mtr_us)*(par_fisc_adj[1,`aliq_amm_mt']/2) +(im_at_ic_tot)*(par_fisc_adj[1,`aliq_amm_aic']/2) +(im_att_new+im_att_us)*(par_fisc_adj[1,`aliq_amm_ot_ttd']/2) +(im_mob_new+im_mob_us)* (par_fisc_adj[1,`aliq_amm_ot_maa']/2) + (im_tras_new/2+im_tras_us/2) * (par_fisc_adj[1,`aliq_amm_ot_mtns']/2)	fiscal_adj.do	temporary variable used when simulating accelerated fiscal depreciation of tangible assets	
amm_im_tan	amm_im_tan = amm_im_tan_old + amm_im_tan_acq + acc_amm	fiscal_adj.do	simulated fiscal tangible goods depreciation	label_fiscal_adj.do
var_op_ult	var_op_ult=ric_vl_co * par_fisc_adj[1,`var_opult_par']	fiscal_adj.do	simulated fiscal variations of ultra-annual works	label_fiscal_adj.do
ca_fab_amm	ca_fab_amm = sp_im_fab + quot_amm_fab*acq_amm_imm	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	
ca_fabns_amm	ca_fabns_amm = 0.5*sp_im_fabns + quot_amm_fabns*acq_amm_imm	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	
ca_im_amm	ca_im_amm = sp_im_tan_im + quot_amm_im*acq_amm_imm	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	
ca_mt_amm	ca_mt_amm = sp_im_mt + quot_amm_mt*acq_amm_imm	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	
ca_aic_amm	ca_aic_amm = sp_im_aic + quot_amm_aic*acq_amm_imm	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	
ca_ot_ttd_amm	ca_ot_ttd_amm = sp_im_ttd + quot_amm_ot_ttd*acq_amm_imm	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	
ca_ot_maa_amm	ca_ot_maa_amm = sp_im_maa + quot_amm_ot_maa*acq_amm_imm	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	

Continue...

Name	Description	Stata program	label	Stata label program
ca_ot_mtns_amm	ca_ot_mtns_amm = 0.5*sp_im_mtns + quot_amm_ot_mtns*acq_amm_imm	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	
ca_tot_amm	ca_tot_amm = ca_fab_amm + ca_fabns_amm + ca_im_amm + ca_mt_amm + ca_aic_amm + ca_ot_ttd_amm + ca_ot_maa_amm + ca_ot_mtns_amm	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	
ded_manu_max	ded_manu_max = 0.05*ca_tot_amm	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	
ded_manu_ord	ded_manu_ord =manord	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	
ecced	ecced= (manord - ded_manu_max) if manord > ded_manu_max	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	
ded_ecced	ded_ecced = ecced/3	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	
manu_ord	manu_ord = ded_manu_ord + ded_ecced	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	
acq_allav_sim	acq_allav_sim= acq_allav - manord + manu_ord	fiscal_adj.do	temporary variable used when computing deductible maintenance expenses for tax purposes	
ca_rsv	ca_rsv=acq_risv+capz_rsv	fiscal_adj.do	temporary variable used when computing deductible over-annual expenses for tax purposes	
acq_risv_ded	acq_risv_ded = ca_rsv/3	fiscal_adj.do	temporary variable used when computing deductible over-annual expenses for tax purposes	
acq_serv_tot_sim	acq_serv_tot_sim = acq_serv_tot + (acq_risv_ded - acq_risv) + (acq_allav_sim - acq_allav)	fiscal_adj.do	simulated fiscal costs for services	label_fiscal_adj.do
classe_base_irap[3]	classe_base_irap = 1 if base_irap_lorda<=0	corp_income.do	temporary variable (Irap tax base income class)	
ut_lor_sim	ut_lor_sim =(ric_tot + ric_vr_tot + var_op_ult+ ric_ini_li+ ric_alpro)-(acq_beni_tot+acq_serv_tot_sim+ acq_gdbt_tot + acq_pers_tot_sim + amm_im_in + amm_im_tan + acq_svimm_sci + dev_cr_sim + acq_vr_tot + acq_accant + acq_alacc_sci + acq_ondiv_tot) + (divid + int_att -int_pas) + (rival - sval) + (pr_onstr_pr - pr_onstr_on)	corp_income.do	temporary variable (simulated fiscal profit/loss)	
corp_inc4	corp_inc=ut_lor_sim	corp_income.do	corporate income	label_corp_income.do
`var_op_ult_tot'	`var_op_ult_tot' = var_op_ult - ric_vl_co	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`va_op_ult'	`va_op_ult'=`var_op_ult_tot' if `var_op_ult_tot'>0	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`vd_op_ult'	`vd_op_ult'=`var_op_ult_tot' if `var_op_ult_tot'<0	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`var_acq_pers_tot'	`var_acq_pers_tot'=acq_pers_tot_sim - acq_pers_tot	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`va_acq_pers_tot'	`va_acq_pers_tot'=`var_acq_pers_tot' if `var_acq_pers_tot' >0	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`vd_acq_pers_tot'	`vd_acq_pers_tot'=`var_acq_pers_tot' if `var_acq_pers_tot'<0	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`var_acq_amm_imi'	`var_acq_amm_imi' = amm_im_tan - acq_amm_imi	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	

Continue...

Name	Description	Stata program	label	Stata label program
`va_acq_amm_imi'	`va_acq_amm_imi' = `var_acq_amm_imi' if `var_acq_amm_imi'>0	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`vd_acq_amm_imi'	`vd_acq_amm_imi' = `var_acq_amm_imi' if `var_acq_amm_imi'<0	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`var_acq_amm_imm'	`var_acq_amm_imm' = amm_im_in - acq_amm_imm	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`va_acq_amm_imm'	`va_acq_amm_imm' =`var_acq_amm_imm' if `var_acq_amm_imm'>0	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`vd_acq_amm_imm'	`vd_acq_amm_imm' =`var_acq_amm_imm' if `var_acq_amm_imm'<0	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`var_acq_svcr_sci'	`var_acq_svcr_sci' = dev_cr_sim - acq_svcr_sci	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`va_acq_svcr_sci'	`va_acq_svcr_sci' =`var_acq_svcr_sci' if `var_acq_svcr_sci'>0	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`vd_acq_svcr_sci'	`vd_acq_svcr_sci' =`var_acq_svcr_sci' if `var_acq_svcr_sci'<0	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`var_acq_serv_tot'	`var_acq_serv_tot' = acq_serv_tot_sim - acq_serv_tot	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`va_acq_serv_tot'	`va_acq_serv_tot' = `var_acq_serv_tot' if `var_acq_serv_tot'>0	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
`vd_acq_serv_tot'	`vd_acq_serv_tot' =`var_acq_serv_tot' if `var_acq_serv_tot'<0	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
simul_va_tot	simul_va_tot = (`va_op_ult'+`va_acq_pers_tot'+`va_acq_amm_imi'+`va_acq_amm_imm'+`va_acq_svcr_sci'+`va_acq_serv_tot')*1936.27/1000000*	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
simul_vd_tot	simul_vd_tot = -(`vd_op_ult'+`vd_acq_pers_tot'+`vd_acq_amm_imi'+`vd_acq_amm_imm'+`vd_acq_svcr_sci'+`vd_acq_serv_tot')*1936.27/1000000*	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
simul_va_tot`i'	simul_va_tot`i'= r(sum)	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
simul_vd_tot`i'	simul_vd_tot`i'= r(sum)	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
simul_va_tot	simul_va_tot=simul_va_tot1+simul_va_tot2+simul_va_tot3+simul_va_tot4+simul_va_tot5+simul_va_tot6+simul_va_tot7+simul_va_tot8+simul_va_tot9+simul_va_tot10+simul_va_tot11+simul_va_tot12+simul_va_tot13+simul_va_tot14	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
simul_vd_tot	simul_vd_tot=simul_vd_tot1+simul_vd_tot2+simul_vd_tot3+simul_vd_tot4+simul_vd_tot5+simul_vd_tot6+simul_vd_tot7+simul_vd_tot8+simul_vd_tot9+simul_vd_tot10+simul_vd_tot11+simul_vd_tot12+simul_vd_tot13+simul_vd_tot14	sim_adj_tot.do	temporary variable used when simulating adjustments of business profits for tax purposes	
va_ut	va_ut = var_aum/ut_lor	sim_adj_tot.do	positive adjustments for firms with profits	label_sim_adj.do
vd_ut	vd_ut = var_dim/ut_lor	sim_adj_tot.do	negative adjustments for firms with profits	label_sim_adj.do
er_lib_ut	er_lib_ut = er_lib/ut_lor	sim_adj_tot.do	liberal transfers for firms with profits	label_sim_adj.do
va_per	va_per = var_aum/per_lor	sim_adj_tot.do	positive adjustments for firms incurring in losses	label_sim_adj.do
				Continue...

Name	Description	Stata program	label	Stata label program
vd_per	vd_per = var_dim/per_lor	sim_adj_tot.do	negative adjustments for firms incurring in losses	label_sim_adj.do
er_lib_per	er_lib_per = er_lib/per_lor	sim_adj_tot.do	liberal transfers for firms incurring in losses	label_sim_adj.do
txc_div	txc_div = par_simul_instr[1,`irpeg_ord']/(1-par_simul_instr[1,`irpeg_ord'])*divid	irpeg.do	dividend tax credit	label_irpeg.do
redd_impon	redd_impon = corp_inc + txc_div	irpeg.do	taxable income	label_irpeg.do
ecc98_loss	ecc98_loss = redd_impon if redd_impon < 0	irpeg.do	fiscal loss of year 1998 to be brought forward	label_irpeg.do
dit_ut_agv_sgl1	dit_ut_agv_sgl1 = redd_impon*(par_simul_instr[1,`dit_par1']) if (redd_impon>0 & dit_eleg ==1)	irpeg.do	temporary variable used when simulating DIT allowance	
dit_ut_agv_sgl2	dit_ut_agv_sgl2 = pn_incr*(par_simul_instr[1,`dit_par2']) if (dit_eleg ==1)	irpeg.do	temporary variable used when simulating DIT allowance	
dit_ut_agv	dit_ut_agv = dit_ut_agv_sgl1 if (dit_ut_agv_sgl2>dit_ut_agv_sgl1)	irpeg.do	allowable DIT income subject to the reduced rate	label_irpeg.do
soglia_dit	soglia_dit=0	irpeg.do	temporary variable used when simulating DIT allowance	
ecc98_dit_ut_agv	ecc98_dit_ut_agv = (dit_ut_agv_sgl2 - dit_ut_agv_sgl1) if (dit_ut_agv_sgl2 > dit_ut_agv_sgl1)	irpeg.do	allowable DIT income to be brought forward	label_irpeg.do
redd_imp_ord	redd_imp_ord = (redd_impon - dit_ut_agv) if (redd_impon>0)	irpeg.do	taxable income subject to the statutory tax rate	label_irpeg.do
irpeg_gross	irpeg_gross=(redd_imp_ord*par_simul_instr[1,`irpeg_ord']) +(dit_ut_agv* par_simul_instr[1,`irpeg_dit']) if (redd_imp_ord>0)	irpeg.do	gross corporate tax	label_irpeg.do
eleg_coop_1	eleg_coop_1 = 0	irpeg.do	temporary variable used when simulating eligibility to the reduced statutory rate for co-operatives	
inc_lcost	inc_lcost = acq_pers_tot/(cost_prod_tot-acq_matp)	irpeg.do	temporary variable used when simulating eligibility to the reduced statutory rate for co-operatives	
eleg_coop_2_ex	eleg_coop_2_ex =0	irpeg.do	temporary variable used when simulating eligibility to the reduced statutory rate for co-operatives	
eleg_coop_2_half	eleg_coop_2_half = 0	irpeg.do	temporary variable used when simulating eligibility to the reduced statutory rate for co-operatives	
irpeg_due	irpeg_due =(irpeg_net-txc_div) if (redd_impon >0)	irpeg.do	corporate tax due	label_irpeg.do
ecc98_irpeg_due	ecc98_irpeg_due = -irpeg_due if (irpeg_due <0)	irpeg.do	corporate tax to be brought forward	label_irpeg.do
tax_rel_tot	tax_rel_tot = (ci_ii + ci_ric + ci_occ + ci_ct) if (redd_impon >0)	irpeg.do	tax reliefs	label_irpeg.do
ecc98_tax_rel_tot	ecc98_tax_rel_tot = tax_rel_tot if (redd_impon <=0)	irpeg.do	tax reliefs to be brought forward	label_irpeg.do
eleg_tax_rel_amt	eleg_tax_rel_amt = tax_rel_tot if (tax_rel_tot <= irpeg_due & redd_impon>0 & irpeg_due>0)	irpeg.do	temporary variable used to simulate tax reliefs	
base_96	base_96= pn_96	dit.do	temporary variable used when simulating DIT allowance	
base_98	base_98 = cap_soc + ris	dit.do	temporary variable used when simulating DIT allowance	
pn_incr5	pn_incr = base_98 - base_96	dit.do		
pn_sgl	pn_sgl = pn	dit.do	temporary variable used when simulating DIT allowance	
sp_pre_cont_incr	sp_pre_cont_incr=(sp_pre_cont - sp_pre_cont_96)	dit.do	temporary variable used when simulating DIT allowance	
sp_pre_conti_incr	sp_pre_conti_incr=(sp_pre_conti - sp_pre_conti_96)	dit.do	temporary variable used when simulating DIT allowance	
sp_atit_incr	sp_atit_incr=(sp_atit - sp_atit_96)	dit.do	temporary variable used when simulating DIT allowance	
sp_af_at_incr	sp_af_at_incr=(sp_af_at - sp_af_at_96)	dit.do	temporary variable used when simulating DIT allowance	
dit_eleg	dit_eleg = 1 if (pn_incr > 0 & flag_pn!=0)	dit.do	temporary variable used when simulating DIT allowance	
				Continue...

Name	Description	Stata program	label	Stata label program
irpeg_net	irpeg_net = 0	imp_tax_allw.do	temporary variable: net corporate tax (gross corporate tax - tax allowances)	
eleg	eleg' = 1 if ((l_ateco=="C" | l_ateco == "D" | l_ateco == "E") & (occ_tot - occ_imco_tot) <=par_simul_instr[1,`ci_ii_par1'] & `eleg_redd' <= par_simul_instr[1,`ci_ii_par2'])	tax_reliefs.do	temporary variable used when simulating eligibility to specific tax reliefs	
eleg_redd	`eleg_redd' = (cap_soc - acq_amm_imi - acq_amm_imm - rival)	tax_reliefs.do	temporary variable used when simulating eligibility to specific tax reliefs	
inv	inv = (im_at_ic_new + inv_amb)	tax_reliefs.do	temporary variable used when simulating eligibility to specific tax reliefs	
eleg_inv	gen `eleg_inv' = inv if (im_brev_new >0 & (inv>= par_simul_instr[1,`ci_ii_par5']))	tax_reliefs.do	temporary variable used when simulating eligibility to specific tax reliefs	
eleg_cum	`eleg_cum'= 1 if (ci_ric>0 & ci_ii>0)	tax_reliefs.do	temporary variable used when simulating eligibility to specific tax reliefs	
ci_ii	gen ci_ii = `eleg_inv' * par_simul_instr[1,`ci_ii_aliq1'] if (`eleg_inv' > 0 & `eleg' ==1 & (occ_tot - occ_imco_tot)<=par_simul_instr[1,`ci_ii_par7'])	tax_reliefs.do	innovative investments tax relief	label_tax_reliefs.do
ci_ric	gen ci_ric = (acq_risv * (par_simul_instr[1,`ci_ric_aliq'])) if (`eleg' ==1 & acq_risv>0 & ut_es_97>0 & ut_es_ris_97>0 & ut_es_ris_97>acq_risv)	tax_reliefs.do	research espenses tax relief	label_tax_reliefs.do
occ_annui	occ_annui = occ_tot*12	tax_reliefs.do	temporary variable used when simulating job creation tax relief	
occ_annui_net	occ_annui_net = occ_tot - occ_td_tot/12 - occ_tp_tot/12	tax_reliefs.do	temporary variable used when simulating job creation tax relief	
occ_td_corr	occ_td_corr = occ_td_ore/(1920*12)	tax_reliefs.do	temporary variable used when simulating job creation tax relief	
occ_tp_corr	occ_tp_corr = occ_tp_ore/(1920*12)	tax_reliefs.do	temporary variable used when simulating job creation tax relief	
dim_corr	dim_corr = (occ_annui_net + occ_td_corr + occ_tp_corr)	tax_reliefs.do	temporary variable used when simulating job creation tax relief	
eleg_pmi	gen `eleg_pmi'= 1 if (`ric_net'<= par_simul_instr[1,`ci_pmi_par1'] & dim_corr <= par_simul_instr[1,`ci_pmi_par2'])	tax_reliefs.do	temporary variable used when simulating job creation tax relief	
eleg_pi	gen `eleg_pi'= 1 if (`ric_net'<= par_simul_instr[1,`ci_pmi_par1'] & dim_corr <= par_simul_instr[1,`ci_pmi_par2'])	tax_reliefs.do	temporary variable used when simulating job creation tax relief	
occ_annui_97	occ_annui_97 = (occ_dir_tot_97 + occ_opap_tot_97)*12	tax_reliefs.do	temporary variable used when simulating job creation tax relief	
occ_annui_98	occ_annui_98 = (occ_dir_tot+ occ_opap_tot)*12	tax_reliefs.do	temporary variable used when simulating job creation tax relief	
occ_incr	occ_incr = ((occ_annui_98 *(1- par_simul_instr[1,`ci_td_agr']) - occ_tp_tot) - (occ_annui_97 *(1-par_simul_instr[1,`ci_td_agr']) - occ_tp_tot_97)) if (l_ateco == "A")	tax_reliefs.do	temporary variable used when simulating job creation tax relief	
eleg_occ	eleg_occ = 1 if (occ_incr > 0 & regione >13 & flag_occ==1)	tax_reliefs.do	temporary variable used when simulating eligibility to the job creation tax relief	
ci_occ	ci_occ = par_simul_instr[1,`ci_occ_par1'] if (occ_incr ==1 & eleg_occ==1)	tax_reliefs.do	job creation tax relief	label_tax_reliefs.do
eleg_inv_ct	`eleg_inv_ct' = (im_mac_new - im_mtr_new + im_at_ic_new + im_att_new + im_mob_new)/**/ if (l_ateco == "H"| l_ateco2=="52"|nace=="6330")	tax_reliefs.do	temporary variable used when simulating eligibility to the tax relief for firms of commerce and tourism sectors	
ci_ct	ci_ct = `eleg_inv_ct'* par_simul_instr[1,`ci_ct_aliq'] if (`eleg_pmi'==1 & `eleg_inv_ct'> 0)	tax_reliefs.do	tax relief for firms of commerce and tourism	label_tax_reliefs.do
sim_tax_rel	sim_tax_rel = (ci_ii + ci_ric + ci_occ + ci_ct)*1936.27/1000000	imp_tax_reliefs.do	temporary variable used to compute non-simulated tax reliefs to be imputed	

Continue...

Name	Description	Stata program	label	Stata label program
sim_tax_rel_tot	sim_tax_rel_tot = sim_tax_rel1+sim_tax_rel2+sim_tax_rel3+sim_tax_rel4+sim_tax_rel5+sim_tax_rel6+sim_tax_rel7+sim_tax_rel8+sim_tax_rel9+sim_tax_rel10+sim_tax_rel11+sim_tax_rel12+sim_tax_rel13+sim_tax_rel14	imp_tax_reliefs.do	temporary variable used to compute non-simulated tax reliefs to be imputed	
tax_reliefs	tax_reliefs = cred_tot/imp_netta	imp_tax_reliefs.do	temporary variable used to compute non-simulated tax reliefs to be imputed	
sim_tax_cred	sim_tax_cred= (txc_div) * 1936.27/1000000	imp_tax_credits.do	temporary variable used to compute non-simulated tax reliefs to be imputed	
sim_tax_cred_tot	sim_tax_cred_tot = sim_tax_cred1+sim_tax_cred2+sim_tax_cred3+sim_tax_cred4+sim_tax_cred5+sim_tax_cred6+sim_tax_cred7+sim_tax_cred8+sim_tax_cred9+sim_tax_cred10+sim_tax_cred11+sim_tax_cred12+sim_tax_cred13+sim_tax_cred14	imp_tax_credits.do	temporary variable used to compute non-simulated tax reliefs to be imputed	
cred_imp	gen cred_imp = cred_imp_tot/imp_netta	imp_tax_credits.do	temporary variable used to compute non-simulated tax reliefs to be imputed	
ESTR	gen ESTR = irpeg_gross/redd_impon if redd_impon>0	stats.do	Effective Statutory corporate tax rates	label_stats.do
EPITR	gen EPITR = irpeg_due/corp_inc if corp_inc>0	stats.do	Ex_post Implicit corporate tax rates	label_stats.do
cla_add6	gen cla_add= 1 if addetti_totali>=100 & addetti_totali<150	stats.do	Classes of employees	label_stats.do

Notes :

[1] label define ecla 9 "0 to 9" 19 "10 to 19" 49 "20 to 49" 99 "50 to 99" 199 "100 to 199" 249 "200 to 249" 499 "250 to 499" 1000 "500 to 1000" 2000 "Above 1000"

[2] label define ecla3 49 "0 to 49" 250 "50 to 250" 2000 "Above 250"

[3] replace classe_base_irap = `1' if (base_irap_lorda>`2' & base_irap_lorda<=`3' /& classe_base_irap==.)

replace classe_base_irap = 14 if (base_irap_lorda > 258228450 & classe_base_irap==.)

classe 2 0 10329

classe 3 10329 25823

classe 4 25823 51646

classe 5 51646 103291

classe 6 103291 185924

classe 7 185924 516457

classe 8 516457 1032914

classe 9 1032914 2582285

classe 10 2582285 5164569

classe 11 5164569 25822845

classe 12 25822845 51645690

classe 13 51645690 258228450

[4] **corp_inc**

quietly replace corp_inc= ut_lor_sim*(1 - corr_prof[`1',1]+corr_prof[`1',2]+corr_prof[`1',3])

if (classe_base_irap ==`1' & utile_lor_sci>0 & ut_lor_sim<0)

display "imputed fiscal adjustments of profits for enterprises of income class `1' "

quietly replace corp_inc= ut_lor_sim*(1 + corr_prof[`1',1]- corr_prof[`1',2]- corr_prof[`1',3])

```
if (classe_base_irap==`1' & utile_lor_sci<0 & ut_lor_sim>0)
quietly replace corp_inc= ut_lor_sim*(1 - corr_prof[`1',1]+ corr_prof[`1',2]+ corr_prof[`1',3])
if (classe_base_irap==`1' & utile_lor_sci<0 & ut_lor_sim<0)
display "imputed fiscal adjustments of loss for enterprises of income class `1' "
```

[5] pn_incr

```
replace pn_incr = 0 if pn_sgl<=0
replace pn_incr = pn_sgl if (pn_incr>= pn_sgl & pn_sgl>0)
replace pn_incr = (pn_incr - sp_pre_cont_incr) if (sp_pre_cont_incr>0 & pn_incr>0)
replace pn_incr = (pn_incr - sp_pre_conti) if (sp_pre_conti>0 & pn_incr>0)
replace pn_incr = (pn_incr - sp_atit_incr) if (sp_atit_incr>0 & pn_incr>0)
replace pn_incr = (pn_incr - sp_af_at_incr) if (sp_af_at_incr>0 & pn_incr>0)
```

[6] cla_add

```
1 if addetti_totali>=100 & addetti_totali<150
replace cla_add=2 if addetti_totali>=150 & addetti_totali<200
replace cla_add=3 if addetti_totali>=200 & addetti_totali<250
replace cla_add=4 if addetti_totali>=250 & addetti_totali<500
replace cla_add=5 if addetti_totali>=500
```

APPENDIX C

STATA CODE OF DIECOFIS MODEL

In the archive file enclosed with the present deliverable, programs used to construct the Diecofis prototype model are provided. These programs can be immediately executed with the RTSS dataset and results for Social contributions, IRAP and Corporate tax can be obtained. However, this code can be used as "template" for other countries' users to build a similar microsimulation model for such taxes. Each survey is at least slightly different from the other, so that the code would, at a minimum, have to be modified for each country to take into account differences in structure of the questionnaire as well as to give due consideration to each country's unique circumstances and institutions, types of data collected in the survey, etc. An attempt has been made to add enough comments to the code to make it broadly comprehensible and to aid those who wish to translate it into languages other than STATA. The code given here is the code that was actually used to produce the results presented in our deliverables.

The programs are written in STATA version 8 Special Edition.

The complete model may be run by executing the following master program:

```
/*************** MASTER.DO ***************
file master with all <*.do> files to compute Social Contributions and IRAP revenue */

display "Executing NAMES ......"
qui do names 98

display "CHECKing variables in the dataset ......"
qui do check 98

display "Computing wages and salaries ......"
qui do retrib 98

display "Computing social contributions ......"
qui do contrib 98

display "Computing social contributions STATISTICS......"
qui do stat_contrib 98

display "Preparing TAX VARiables ......"
qui do taxvar 98

display "Computing IRAP tax base and revenue ....."
qui do irap 98

display "Computing STATISTICS for IRAP ....."
qui do stat_irap 98

display "*************** END OF SC and IRAP ROUTINES ****************"
display "*************** START OF IRPEG ROUTINES ********************"

display "Executing do-file adjusting some balance sheet variables for tax purposes ........"
qui do fiscal_adj.do

display "Executing do-file computing corporate income............."
qui do corp_income.do

display "Executing do-file simulating irpeg"
qui do irpeg.do

display "Computing summary statistics and tax rates for IRPEG"
qui do stat_irpeg.do

display "*************** END OF DIECOFIS MICROSIMULATION MODEL ***************"
```

Detail of each program:

```
* NAMES.DO
** PROGRAM TO DEFINE VARIABLE NAMES AND LABELS ***

/* type on the command line:
do names <year>
where <year> is the year of the database*/

args year

use overall`year'c, clear

rename l_ateco2 ateco2
rename ateco ateco5
rename forma_giur form_giur
rename codice_provincia cod_prov

order codice ateco5 ateco2 l_ateco areag regione cod_prov form_giur

destring codice_comune lett2 regcont ab18, replace
format %15.0fc areag sci pmi regione regcont


set dp comma

*dataset VALPROD
rename cod11100       ric_tot
rename cod11101       ric_vpi
rename cod11102       ric_vmnt
rename cod11103       ric_lct
rename cod11104       ric_lot
rename cod11105       ric_ai
rename cod11106       ric_tra
rename cod11107       ric_pst
rename cod11200       ric_vr_tot
rename cod11201       ric_vr_pf
rename cod11202       ric_vr_pcl
rename cod11300       ric_vl_co
rename cod11400       ric_ini_li
rename cod11500       ric_alpro

*dataset COSTPROD
rename cod12100       acq_beni_tot
rename cod12101       acq_matp
rename cod12102       acq_ener
rename cod12103       acq_mriv
rename cod12200       acq_serv_tot
rename cod12201       acq_lavter
rename cod12202       acq_allav
rename cod12203       acq_trasp
rename cod12204       acq_altr_sci
rename cod12205       acq_intmd
rename cod12206       acq_pubbli
rename cod12207       acq_risv
rename cod12208       acq_consul
rename cod12209       acq_inform
rename cod12210       acq_prassi
rename cod12211       acq_licuso
rename cod12212       acq_smrif
rename cod12213       acq_alser
rename cod12214       acq_banc
rename cod12300       acq_gdbt_tot
rename cod12301       acq_fitpa
rename cod12302       acq_leasing
rename cod12303       acq_alfitpa_sci
rename cod12304       acq_canlo
rename cod12305       acq_leasstr_sci
rename cod44000       acq_pers_tot
rename cod12500       acq_amm_tot
rename cod12510       acq_amm_imi
rename cod12520       acq_amm_imm
rename cod12530       acq_svimm_sci
rename cod12540       acq_svcr_sci
rename cod12600       acq_vr_tot
```

```
rename cod12601        acq_vr_mp
rename cod12602        acq_vr_r
rename cod12700        acq_accant
rename cod12800   acq_alacc_sci
rename cod12900        acq_ondiv_tot
rename cod12901        acq_forper
rename cod12902        acq_onamm
rename cod12903        acq_aodg
rename cod12904        acq_impfab_sci
rename cod12905        acq_imp_ind
rename cod12906        acq_alimp_sci

*costi per il personale
rename cod41110        acq_ret_dirimp
rename cod42110   acq_ret_alcat
rename cod42111   acq_ret_oppmi
rename cod42112   acq_ret_appmi
rename cod42113   acq_ret_ldpmi
rename cod42121        acq_cs
rename cod42131        acq_qtfr
rename cod43000        acq_prpers

rename cod45000        acq_indlic

*additional detail for SCI
rename cod41120        acq_csdir_sci
rename cod42120        acq_csal_sci
rename cod41130        acq_tfrdir_sci
rename cod42130        acq_tfral_sci


*dataset RISULTEC
rename cod13000        cost_prod_tot
rename cod12000        val_prod_tot
rename cod13999        mol
rename cod14200        int_att
rename cod14300        int_pas
rename cod15000        pr_onfin_sci
rename cod16000        ret_valaf_sci
rename cod17000        pr_onstr_sci
rename cod17999        utile_lor_sci
rename cod18100        imp_reddito
rename cod18500        imp_irap
rename cod19000        utile_netto

*dataset STATOPAT
rename cod22100        sp_im_im
rename cod22200        sp_im_man
rename cod22300        sp_im_fi
rename cod23110        sp_rim_mp
rename cod23120        sp_rim_ps
rename cod23130        sp_rim_lc
rename cod23140        sp_rim_pf
rename cod23141        sp_rim_riv
rename cod23260        sp_cre_bs
rename cod23270        sp_cre_ls
rename cod23400        sp_liq
rename cod26000        sp_foro
rename cod27000        sp_tfr
rename cod28140        sp_deb_bs
rename cod28150        sp_deb_ls
rename cod23200        sp_cretot_sci
rename cod28000        sp_debtot_sci
rename cod29990        sp_pastot_sci

*dataset OCCUPATI
rename cod31100        occ_imco_tot
rename cod31101        occ_imco_f
rename cod31110        occ_imptot_pmi
rename cod31111        occ_impf_pmi
rename cod31120        occ_coftot_pmi
rename cod31121        occ_coff_pmi
rename cod31122        occ_cofore_pmi
rename cod31200        occ_dirimp_tot
rename cod31201        occ_dirimp_f
rename cod31202        occ_dirimp_ore
```

```
rename cod31300       occ_opap_tot
rename cod31310       occ_optot_pmi
rename cod31311       occ_opf_pmi
rename cod31312       occ_opore_pmi
rename cod31320       occ_aptot_pmi
rename cod31321       occ_apf_pmi

rename cod31322       occ_apore_pmi
rename cod31400       occ_ld_tot
rename cod31401       occ_ld_f
rename cod31402       occ_ld_ore
rename cod31000       occ_tot
rename cod31001       occ_tot_f
rename cod31002       occ_tot_ore
rename cod30100       occ_td_tot
rename cod30101       occ_td_f
rename cod30102       occ_td_ore
rename cod30200       occ_tp_tot
rename cod30201       occ_tp_f
rename cod30202       occ_tp_ore
rename cod30300       occ_fltot_pmi
rename cod30301       occ_flf_pmi
rename cod30302       occ_flore_pmi
rename cod32100       occ_cig_tot

*variables ONLY for SCI
rename cod31301   occ_opf_sci
rename cod31302   occ_opore_sci
rename cod31500   occ_aptot_sci
rename cod31502   occ_apore_sci
rename cod32110   occ_cigor_sci
rename cod32120   occ_cigst_sci

*dataset ACQUIMM
rename cod53000       im_acq_tot
rename cod53001       im_acq_new
rename cod53002       im_acq_us
rename cod51110       im_acqter
rename cod51120       im_cost_tot
rename cod51121       im_cost_new
rename cod51122       im_cost_us

rename cod51200       im_mac_tot
rename cod51201       im_mac_new
rename cod51202       im_mac_us
rename cod51410       im_att_tot
rename cod51411       im_att_new
rename cod51412       im_att_us
rename cod51420       im_mob_tot
rename cod51421       im_mob_new
rename cod51422       im_mob_us
rename cod51430       im_tras_tot
rename cod51431       im_tras_new
rename cod51432       im_tras_us
rename cod51440       im_bval_tot
rename cod51441       im_bval_new
rename cod51442       im_bval_us
rename cod52200       im_art_tot
rename cod52201       im_art_new
rename cod52202       im_art_us
rename cod52400       im_sof_tot
rename cod52401       im_sof_new
rename cod52402       im_sof_us

*dataset ALTRI DATI
rename cod60040       exp_ue
rename cod60050       exp_eue
rename cod60070       imp_ue
rename cod60080       imp_eue
rename cod61110       ind_ass
rename cod61120       fitti_att
rename cod61130       royal
rename cod61140       cont_ese
rename cod61230       manord
rename cod61240       v_cap_us
rename cod61250       k_leasing
```

```
rename cod61260       q_leasing
rename cod61280       fatt_ed
rename cod61290       fatt_cost
rename cod61300       pers_int
rename cod61310       inv_amb
rename cod60010       iva_cli
rename cod60020       iva_for
rename cod61150       contr_k
rename cod61160       contr_i
rename cod61200       imp_dir
rename cod61265       imp_ind

*dataset REGIONI
rename cod70011       add_01
rename cod70012       cper_01
rename cod70021       add_02
rename cod70022       cper_02
rename cod70031       add_03
rename cod70032       cper_03
rename cod70051       add_04
rename cod70052       cper_04
rename cod70061       add_05
rename cod70062       cper_05
rename cod70071       add_06
rename cod70072       cper_06
rename cod70081       add_07
rename cod70082       cper_07
rename cod70091       add_08
rename cod70092       cper_08
rename cod70101       add_09
rename cod70102       cper_09
rename cod70111       add_10
rename cod70112       cper_10
rename cod70121       add_11
rename cod70122       cper_11
rename cod70131       add_12
rename cod70132       cper_12
rename cod70141       add_13
rename cod70142       cper_13
rename cod70151       add_14
rename cod70152       cper_14
rename cod70161       add_15
rename cod70162       cper_15
rename cod70171       add_16
rename cod70172       cper_16
rename cod70181       add_17
rename cod70182       cper_17
rename cod70191       add_18
rename cod70192       cper_18
rename cod70201       add_19
rename cod70202       cper_19
rename cod70211       add_20
rename cod70212       cper_20
rename cod70221       add_21
rename cod70222       cper_21
rename cod77231       add_est_sci
rename cod77232       cper_est_sci
rename cod77241       add_noloc_sci
rename cod77242       cper_noloc_sci


rename cod60120       ptc_est
rename cod60220       cred_fin_est
rename cod60250       cred_com_est
rename cod60620       deb_est_b
rename cod60650       deb_est_i

rename cod23420       sp_liq_ass
rename cod22310       sp_part_cont
rename cod22320       sp_part_coll
rename cod22330       sp_part_al
rename cod23310       sp_ac_pct
rename cod23320       sp_ac_pcl
rename cod23330       sp_ac_pot
rename cod60110       ptc_ita
rename cod14100       divid
```

```
************************************************
* AMMORTAMENTO
*************

rename cod22102        sp_im_cop
rename cod22103        sp_im_ip
rename cod52900        oth_new_in
rename cod52100        oth_new_in_c
rename ab18            sp_fd_im

rename ab15            sp_im_av
rename cod52500        oth_new_in_oth
rename cod22101        sp_im_rsp
rename cod22105        sp_im_sfw
rename cod22104        sp_im_tdm
rename cod52300        oth_new_in_tdm
rename cod61180        capz_rsv

***********************************
* ORDINARY DEPRECIATION (Art. 67, co.2)
***********************************

rename cod22212        sp_im_fab
rename cod22211        sp_im_ter
rename cod22244        sp_im_bv
rename cod22213        sp_im_fabns
rename cod22220        sp_im_mac
rename cod22221        sp_im_mt
rename cod22230        sp_im_aic
rename cod22241        sp_im_ttd
rename cod22242        sp_im_maa
rename cod22243        sp_im_mtns
rename cod51130        im_acq_fns
rename cod51211        im_mtr_new
rename cod51212        im_mtr_us
rename cod51300        im_at_ic_tot
rename cod51900        im_mat_acq_tot
rename ab26    sp_fd_im_tan
rename cod22250        sp_im_inc
rename cod15100        rival
rename cod15200        sval
rename cod16100        pr_onstr_pr
rename cod16200        pr_onstr_on
rename cod22340        sp_pre_cont
rename cod22341        sp_pre_cont_bt
rename cod22350        sp_pre_coll
rename cod22351        sp_pre_coll_bt
rename cod22360        sp_pre_conti
rename cod22361        sp_pre_conti_bt
rename cod22370        sp_pre_at
rename cod22371        sp_pre_al_bt
rename cod22380        sp_atit
rename cod22390        sp_az_pr
rename cod22300_96     sp_im_fi_96
rename cod22310_96     sp_part_cont_96
rename cod22320_96     sp_part_coll_96
rename cod22330_96     sp_part_al_96
rename cod22340_96     sp_pre_cont_96
rename cod22350_96     sp_pre_coll_96
rename cod22360_96     sp_pre_conti_96
rename cod22370_96     sp_pre_at_96
rename cod22380_96     sp_atit_96
rename cod22390_96     sp_az_pr_96
rename cod23350_96     sp_af_at_96
rename cod23350        sp_af_at
rename cod25000        pn
rename cod25100        cap_soc
rename cod25200        ris
rename cod25300        ut_nuo
rename cod25400        ut_es
rename cod25401        ut_es_ris
rename cod25402        ut_es_soc
rename cod25000_96     pn_96
rename cod25100_96     cap_soc_96
```

```
rename cod25200_96     ris_96
rename cod25300_96     ut_nuo_96
rename cod25400_96     ut_es_96
rename cod25300_97     ut_nuo_97
rename cod25400_97     ut_es_97
rename cod25401_97     ut_es_ris_97
rename cod25402_97     ut_es_soc_97
rename cod31300_97     occ_opap_tot_97
rename cod31400_97     occ_ld_tot_97
rename cod31000_97     occ_tot_97
rename cod30100_97     occ_td_tot_97
rename cod30200_97     occ_tp_tot_97
rename cod31100_97     occ_imco_tot_97
rename cod31200_97     occ_dirimp_tot_97
rename cod31500_97     occ_aptot_sci_97
rename cod51301        im_at_ic_new
rename cod51302        im_at_ic_us
rename cod51400        im_ab_tot
rename cod51401        im_ab_new
rename cod51402        im_ab_us
rename cod52101        im_brev_new
rename cod52301        im_conc_new


rename cod24990 sp_attot_sci
rename cod28010 sp_obb_sci
rename cod28011 sp_obb_ls_sci
rename cod28020 sp_obbconv_sci
rename cod28021 sp_obbconv_ls_sci
rename cod28030 sp_debbank_sci
rename cod28031 sp_debbank_ls_sci
rename cod28040 sp_debaltr_sci
rename cod28041 sp_debaltr_ls_sci
rename cod28050 sp_acccl_sci
rename cod28060 sp_debfor_sci
rename cod28061 sp_debfor_ls_sci
rename cod28070 sp_debtit_sci
rename cod28071 sp_debtit_ls_sci
rename cod28080 sp_debcontr_sci
rename cod28081 sp_debcontr_ls_sci
rename cod28090 sp_debcoll_sci
rename cod28091 sp_debcoll_ls_sci
rename cod28100 sp_control_sci
rename cod28101 sp_control_ls_sci
rename cod28110 sp_debtrib_sci
rename cod28111 sp_debtrib_ls_sci
rename cod28120 sp_debss_sci
rename cod28121 sp_debss_ls_sci
rename cod28130 sp_altrdeb_sci
rename cod28131 sp_altrdeb_ls_sci
rename cod24000 sp_attrr_sci
rename cod14301 sp_intfin_sci
rename cod14302 sp_intaltr_sci
rename cod22000 sp_immtot_sci
rename cod21020 sp_cresoc_sci
rename cod23150 sp_accfor_sci
rename cod23000 sp_attcirc_sci

destring sp_im_fi_96 sp_part_cont_96 sp_part_coll_96 sp_part_al_96 sp_pre_cont_96 sp_pre_coll_96,
replace
destring sp_pre_conti_96 sp_pre_at_96 sp_atit_96 sp_az_pr_96 sp_af_at_96, replace

*do labelvar_it
do labelvar_en

label data "RTSS Official Dataset, 19`year'"


save dati`year', replace
```

```
******************
* LABELVAR_EN.DO *
******************
/* program to assign labels to the variables in RTSS dataset as in NAMES.DO */


* LABEL ASSIGNMENT

label variable codice  "Firm Code (ASIA archive)"
label variable ateco5  "Economic Activity Classification (5 digits) (NACE Rev.1)"
label variable l_ateco        "Economic Activity Classification (alphabetical code) (NACE Rev.1)"
label variable ateco2  "Economic Activity Classification (2 digits) (NACE Rev.1)"
label variable areag   "Geographic Area "
label variable regione        "Region"
label variable cod_prov       "Province Code"
label variable form_giur "Legal Status (detailed classification)"
label variable tipo_fg "Enterprise Legal Status (7 categories)"

labelvar pmi "Size of Enterprise" 1 "Small/Medium Enterprise (1-99)" 0 "Medium/Large Enterprise
(>99)"
labelvar sci "Size of Enterprise" 1 "Medium/Large Enterprise (>99)" 0 "Small/Medium Enterprise (1-
99)"
labelvar areag "Geographic Area" 1 "North West" 2 "North East" 3 Centre 4 South 5 Islands
labelvar regcont "Accounting Method" 0 "Not Available" 1 "Simplified Accounting Method" /*
       */2 "Ordinary Accounting Method" 3 "Other"

encode l_ateco, generate(ll_ateco)
label variable ll_ateco "Economic Activity Classification (alphabetical code) (NACE Rev.1)"
label define atecl 1 "Mining and quarrying" /*
*/2 "Manufacturing" /*
*/3 "Electricity, gas and water supply" /*
*/4 "Construction" /*
*/5 "Wholesale and retail trade" /*
*/6 "Hotels and restaurants" /*
*/7 "Transport, storage and communication" /*
*/8 "Financial Intermediation"/*
*/9 "Real estate, renting and business activities" /*
*/10 "Education" /*
*/11 "Health and social work" /*
*/12 "Other service activities"
label values ll_ateco atecl


label variable ateco2 "Sector of Activities (2-digits classification)"
label define atec 01 "Agriculture, hunting and related service activities" /*
*/02 "Forestry, logging and related service activities" /*
*/10 "Mining of coal and lignite; extraction of peat" /*
*/11 "Extraction of crude petroleum and natural gas" /*
*/12 "Mining of uranium and thorium ores" /*
*/13 "Mining of metal ores" /*
*/14 "Other mining and quarrying" /*
*/15 "Manufacture of food products and beverages" /*
*/16 "Manufacture of tobacco products" /*
*/17 "Manufacture of textiles" /*
*/18 "Manufacture of wearing apparel" /*
*/19 "Manufacture of luggage, handbags, saddlery, harness and footwear" /*
*/20 "Manufacture of wood, except furniture" /*
*/21 "Manufacture of pulp, paper and paper products" /*
*/22 "Publishing, printing and reproduction of recorded media" /*
*/23 "Manufacture of coke, refined petroleum products and nuclear fuel" /*
*/24 "Manufacture of chemicals and chemical products" /*
*/25 "Manufacture of rubber and plastic products" /*
*/26 "Manufacture of other non-metallic mineral products" /*
*/27 "Manufacture of basic metals" /*
*/28 "Manufacture of fabricated metal products, except machinery and equipment" /*
*/29 "Manufacture of machinery and equipment n.e.c." /*
*/30 "Manufacture of office machinery and computers" /*
*/31 "Manufacture of electrical machinery and apparatus n.e.c." /*
*/32 "Manufacture of radio, television and communication equipment and apparatus" /*
*/33 "Manufacture of medical, precision and optical instruments, watches and clocks" /*
*/34 "Manufacture of motor vehicles, trailers and semi-trailers" /*
*/35 "Manufacture of other transport equipment" /*
*/36 "Manufacture of furniture; manufacturing n.e.c." /*
*/37 "Recycling" /*
*/40 "Electricity, gas, steam and hot water supply" /*
```

```
*/41 "Collection, purification and distribution of water" /*
*/45 "Construction" /*
*/50 "Sale and repair of motor vehicles and motorcycles; retail sale of automotive fuel" /*
*/51 "Wholesale trade and commission trade" /*
*/52 "Retail trade; repair of personal and household goods" /*
*/55 "Hotels and restaurants" /*
*/60 "Land transport; transport via pipelines" /*
*/61 "Water transport" /*
*/62 "Air transport" /*
*/63 "Supporting and auxiliary transport activities" /*

*/64 "Post and telecommunications" /*
*/65 "Financial intermediation, except insurance and pension funding" /*
*/66 "Insurance and pension funding, except compulsory social security" /*
*/67 "Activities auxiliary to financial intermediation" /*
*/70 "Real estate activities" /*
*/71 "Renting of machinery, equipment, personal and household goods" /*
*/72 "Computer and related activities" /*
*/73 "Research and development" /*
*/74 "Other business activities" /*
*/75 "Public administration and defence; compulsory social security" /*
*/80 "Education" /*
*/85 "Health and social work" /*
*/90 "Sewage and refuse disposal, sanitation and similar activities" /*
*/91 "Activities of membership organizations n.e.c." /*
*/92 "Recreational, cultural and sporting activities" /*
*/93 "Other service activities" /*
*/95 "Activities of households as employers of domestic staff" /*
*/96 "Undifferentiated goods producing activities of private households for own use" /*
*/97 "Undifferentiated services producing activities of private households for own use" /*
*/99 "Extra-territorial organizations and bodies"
label values ateco2 atec

label variable regione "Italian Regions"
label define reg 01 "Piemonte" /*
*/02 "Valle d'Aosta" /*
*/03 "Lombardia" /*
*/04 "Trentino Alto Adige" /*
*/05 "Veneto" /*
*/06 "Friuli Venezia Giulia" /*
*/07 "Liguria" /*
*/08 "Emilia Romagna" /*
*/09 "Toscana" /*
*/10 "Umbria" /*
*/11 "Marche" /*
*/12 "Lazio" /*
*/13 "Abruzzo" /*
*/14 "Molise" /*
*/15 "Campania" /*
*/16 "Puglia" /*
*/17 "Basilicata" /*
*/18 "Calabria" /*
*/19 "Sicilia" /*
*/20 "Sardegna"
label values regione reg

label variable peso "Sample Weights"
label variable anno "Year of Survey"
label variable addetti_indipendenti " Employer and family coadjuvant 1998(ASIA archive)"
label variable addetti_totali "Total Employment 1998(ASIA archive)"
label variable lett2 "Economic Activity Classification (alphabetical detailed code) (NACE Rev.1)"
label variable nace "Economic Activity Classification (4 digits) (NACE Rev.1)"
label variable nascita "Establishment year"
label variable cessazione "Closing year"
label variable ind97 "Employer and Family coadjuvant 1997 (ASIA archive)"
label variable ind96 "Employer and Family coadjuvant 1996 (ASIA archive)"
label variable ind99 "Employer and Family coadjuvant 1999 (ASIA archive)"
label variable ind00 "Employer and Family coadjuvant 2000 (ASIA archive)"
label variable dip96 "Total Employees 1996 (ASIA archive)"
label variable dip97 "Total Employees 1997 (ASIA archive)"
label variable dip98 "Total Employees 1998 (ASIA archive)"
label variable dip99 "Total Employees 1999 (ASIA archive)"
label variable dip00 "Total Employees 2000 (ASIA archive)"
label variable adde96 "Total Employment 1996 (employer, employees and coadjuvant) (ASIA archive)"
label variable adde97 "Total Employment 1997 (employer, employees and coadjuvant) (ASIA archive)"
label variable adde99 "Total Employment 1999 (employer, employees and coadjuvant) (ASIA archive)"
```

```
label variable adde00 "Total Employment 2000 (employer, employees and coadjuvant) (ASIA archive)"


*dataset VALPROD
label variable ric_tot        "Income from sales and Services"
label variable ric_vpi        "Sales of firm products"
label variable ric_vmnt              "Sales of goods"
label variable ric_lct        "Works on behalf of third parties"
label variable ric_lot        "Works and industrial services on orders of third parties"
label variable ric_ai         "Brokerage activities"
label variable ric_tra        "Income of transport enterprises"
label variable ric_pst        "Services to third parties"
label variable ric_vr_tot     "Variations of the stocks of finished and semi-finished products"
label variable ric_vr_pf      "Variations of the stocks of products"
label variable ric_vr_pcl     "Variations of the stocks of under-processing products"
label variable ric_vl_co      "Variation in contract work in progress"
label variable ric_ini_li     "Increase on internal work capitalized under fixed assets"
label variable ric_alpro      "Other income and earnings (neither financial, nor extraordinary)"

*dataset COSTPROD
label variable acq_beni_tot   "Purchases"
label variable acq_matp              "Raw materials"
label variable acq_ener              "Energy products"
label variable acq_mriv              "Goods for resale"
label variable acq_serv_tot   "Services (Total)"
label variable acq_lavter     "Works made by third parties"
label variable acq_allav      "Other works made by third parties"
label variable acq_trasp      "Transport"
label variable acq_altr_sci   "Other Transport (SCI)"
label variable acq_intmd      "Brokerages"
label variable acq_pubbli     "Advertising"
label variable acq_risv              "Research and Development"
label variable acq_consul     "Consulting"
label variable acq_inform     "Informatics"
label variable acq_prassi     "Insurance premiums"
label variable acq_licuso     "Licences"
label variable acq_smrif      "Waste disposal"
label variable acq_alser      "Other services"
label variable acq_banc              "Bank services"
label variable acq_gdbt_tot   "Use of third party assets"
label variable acq_fitpa      "Rent charges for capital goods"
label variable acq_leasing    "Leasing expenses"
label variable acq_alfitpa_sci "Other rents from buildings (SCI)"
label variable acq_canlo      "Other rents"
label variable acq_leasstr_sci "Leasing expenses for instrumental goods (SCI)
label variable acq_pers_tot   "Personnel Expenses (Total)"
label variable acq_amm_tot    "Depreciation (PMI), Depreciation and Devaluation(SCI)"
label variable acq_amm_imi    "Depreciation of intangible fixed assets"
label variable acq_amm_imm    "Depreciation of tangible fixed assets"
label variable acq_svimm_sci  "Other write-downs of fixed assets (SCI)"
label variable acq_svcr_sci   "Write-downs of  of current credits (SCI)"
label variable acq_vr_tot     "Variations of stocks of raw materials and to resale (Tot)"
label variable acq_vr_mp      "Variations of stocks of raw materials"
label variable acq_vr_r              "Variations of stocks to resale"
label variable acq_accant     "Provisions"
label variable acq_alacc_sci  "Other provisions (SCI)"
label variable acq_ondiv_tot  "Other operating costs (Total)"
label variable acq_forper     "Personnel training expenses"
label variable acq_onamm      "Managers rewards"
label variable acq_aodg              "Other operating charges"
label variable acq_impfab_sci "Excises taxes on production (SCI)"
label variable acq_imp_ind    "Indirect taxes on products"
label variable acq_alimp_sci  "Other indirect taxes (SCI)"

*personnel cost
label variable acq_ret_dirimp "Wages and salaries: Executives, Employees"
label variable acq_ret_alcat  "Wages and salaries: Workers and others"
label variable acq_ret_oppmi  "Wages and salaries: Workers (PMI)"
label variable acq_ret_appmi  "Wages and salaries: Apprentices (PMI)"
label variable acq_ret_ldpmi  "Wages and salaries: At-home Workers (PMI)"
label variable acq_cs         "Social security contributions (Total)"
label variable acq_qtfr              "Annual provision for severance pay (flow)"
label variable acq_prpers     "Other personnel costs"

label variable acq_indlic     "Indemnities for dismissals"
```

```
*additional detail for SCI
label variable acq_csdir_sci  "Social security contributions: Executives, Employees (SCI)"
label variable acq_csal_sci   "Social security contributions: Workers and others (SCI)"
label variable acq_tfrdir_sci "Annual provision for severance pay: Executives, Employees (SCI)"
label variable acq_tfral_sci  "Annual provision for severance pay: Workers and others (SCI)"

*dataset RISULTEC
label variable cost_prod_tot  "Costs of production"
label variable val_prod_tot   "Value of production"


label variable mol            "Gross operating surplus"
label variable int_att        "Interest receivable"
label variable int_pas        "Interest payable"
label variable pr_onfin_sci   "Interest receivable and payable Total (SCI)"
label variable ret_valaf_sci  "Adjustment for financial assets Total (SCI)"
label variable pr_onstr_sci   "Extraordinary proceeds and costs Total (SCI)"
label variable utile_lor_sci  "Gross Profit (loss) for the financial year (SCI)"
label variable imp_reddito    "Income taxes"
label variable imp_irap              "IRAP (PMI)"
label variable utile_netto    "Net Profit (loss) for the financial year"

*dataset STATOPAT
label variable sp_im_im              "Intangible fixed assets"
label variable sp_im_man      "Tangible fixed assets (Net of depreciation funds)"
label variable sp_im_fi              "Financial fixed assets"
label variable sp_rim_mp      "Current assets: raw materials, ancillars and consumables"
label variable sp_rim_ps      "Current assets: under-processing and semifinished products"
label variable sp_rim_lc      "Current assets: works in process under contracts"
label variable sp_rim_pf      "Current assets: finished products and goods"
label variable sp_rim_riv     "Current assets: goods for resale"
label variable sp_cre_bs      "Current assets: short term credits"
label variable sp_cre_ls      "Current assets: medium-long term credits"
label variable sp_liq         "Current assets: cash"
label variable sp_foro        "Liabilities: funds for risks and charges (total)"
label variable sp_tfr         "Liabilities: severance pay fund (stock)"
label variable sp_deb_bs      "Liabilities: short term debts (PMI)"
label variable sp_deb_ls      "Liabilities: medium-long term debts (PMI)"
label variable sp_cretot_sci  "Total credits (SCI)"
label variable sp_debtot_sci  "Total Debts (SCI)"
label variable sp_pastot_sci  "Total Liabilities (SCI)"

*dataset OCCUPATI
label variable occ_imco_tot   "Entrepreneurs and Family assistants: total"
label variable occ_imco_f     "Entrepreneurs and Family assistants: women"
label variable occ_imptot_pmi "Entrepreneurs: total (PMI)"
label variable occ_impf_pmi   "Entrepreneurs: women (PMI)"
label variable occ_coftot_pmi "Family assistants: total (PMI)"
label variable occ_coff_pmi   "Family assistants: women (PMI)"
label variable occ_cofore_pmi "Family assistants: hours (PMI)"
label variable occ_dirimp_tot "Executives and Employees: total"
label variable occ_dirimp_f   "Executives and Employees: women"
label variable occ_dirimp_ore "Executives and Employees: hours"
label variable occ_opap_tot   "Workers and Trainees (PMI) and Workers (SCI)"
label variable occ_opf_sci    "Workers: women (SCI)"
label variable occ_opore_sci  "Workers: hours (SCI)"
label variable occ_optot_pmi  "Workers: total (PMI)"
label variable occ_opf_pmi    "Workers: women (PMI)"
label variable occ_opore_pmi  "Workers: hours (PMI)"
label variable occ_aptot_pmi  "Trainees: total (PMI)"
label variable occ_apf_pmi    "Trainees: women (PMI)"
label variable occ_apore_pmi  "Trainees: hours (PMI)"
label variable occ_ld_tot     "At-home workers: total"
label variable occ_ld_f              "At-home workers: women"
label variable occ_ld_ore     "At-home workers: hours"
label variable occ_tot        "Total employed staff"
label variable occ_tot_f      "Total employed women"
label variable occ_tot_ore    "Total worked hours"
label variable occ_td_tot     "Short-term contracts: total"
label variable occ_td_f              "Short-term contracts: women"
label variable occ_td_ore     "Short-term contracts: hours"
label variable occ_tp_tot     "Part-time contracts: total"
label variable occ_tp_f              "Part-time contracts: women"
label variable occ_tp_ore     "Part-time contracts: hours"
label variable occ_fltot_pmi  "Training contracts: total (PMI)"
label variable occ_flf_pmi    "Training contracts: women (PMI)"
```

```
label variable occ_flore_pmi  "Training contracts: hours (PMI)"
label variable occ_cig_tot    "Ordinary lay-off (CIG) hours used"

*variables ONLY for SCI
label variable occ_aptot_sci         "Trainees: total (SCI)"
label variable occ_apore_sci         "Trainees: hours (SCI)"
label variable occ_cigor_sci         "Ordinary lay-off (CIG) hours used (SCI)"
label variable occ_cigst_sci         "Extra-ordinary lay-off (CIG) hours used (SCI)"


*dataset ACQUIMM
label variable im_acq_tot     "Purchase of Fixed assets (total)"
label variable im_acq_new     "Purchase of Fixed assets: New"
label variable im_acq_us      "Purchase of Fixed assets: Second-hand"
label variable im_acqter      "Purchase of Land"
label variable im_cost_tot    "Purchase of Constructions (total)"
label variable im_cost_new    "Purchase of Constructions: New"
label variable im_cost_us     "Purchase of Constructions: Second-hand"
label variable im_mac_tot     "Purchase of Machinery (total)"
label variable im_mac_new     "Purchase of machinery: New"
label variable im_mac_us      "Purchase of machinery: Second-hand"
label variable im_att_tot     "Purchase of Data equipment (total)"
label variable im_att_new     "Purchase of data-equipment: New"
label variable im_att_us      "Purchase of data-equipment: Second-hand"
label variable im_mob_tot     "Purchase of Furniture (total)"
label variable im_mob_new     "Purchase of Furniture: New"
label variable im_mob_us      "Purchase of Furniture: Second-hand"
label variable im_tras_tot    "Purchase of Means of transport (total)"
label variable im_tras_new    "Purchase of Means of transport: New"
label variable im_tras_us     "Purchase of Means of transport: Second-hand"
label variable im_bval_tot    "Purchase of Valuable assets (total)"
label variable im_bval_new    "Purchase of Valuable assets: New"
label variable im_bval_us     "Purchase of Valuable assets: Second-hand"
label variable im_art_tot     "Purchase of Artistic assets (total)"
label variable im_art_new     "Purchase of Artistic assets: New"
label variable im_art_us      "Purchase of Artistic assets: Second-hand"
label variable im_sof_tot     "Purchase of Softwares (total)"
label variable im_sof_new     "Purchase of softwares: New"
label variable im_sof_us      "Purchase of softwares: Second-hand"

*dataset ALTRI
label variable exp_ue         "Export sales (EU countries)"
label variable exp_eue        "Export sales (Extra-EU countries)"
label variable imp_ue         "Import sales (EU countries)"
label variable imp_eue        "Import sales (Extra-EU countries)"
label variable ind_ass        "Insurance compensations"
label variable fitti_att      "Income from rents"
label variable royal          "Revenue from Royalties, patents and similar"
label variable cont_ese              "General Government allowances on working/operating account"
label variable manord         "Routine buildings maintenance"
label variable v_cap_us              "Sales of second-hand capital goods"
label variable k_leasing      "Value of capital in leasing contracts of the financial year"
label variable q_leasing      "Share of financial leasing for the financial year"
label variable fatt_ed        "Turnover of construction enterprises (for building)"
label variable fatt_cost      "Turnover of construction enterprises (for engineering)"
label variable pers_int              "Personnel expenses for workers from provisional agencies
(lavoro interinale)"
label variable inv_amb        "Investments in equipment for the protection of the environment"
label variable iva_cli        "VAT from customers"
label variable iva_for        "VAT to suppliers"
label variable contr_k        "General Government capital allowances"
label variable contr_i        "General Government allowances on interest account"
label variable imp_dir        "Direct taxes payed in the financial year"
label variable imp_ind        "Indirect taxes on production"

*dataset REGIONI
label variable add_01         "Annual workers average (Piemonte)"
label variable cper_01        "Personnel costs (Piemonte)"
label variable add_02         "Annual workers average (Valle d'Aosta)"
label variable cper_02        "Personnel costs (Valle d'Aosta)"
label variable add_03         "Annual workers average (Lombardia)"
label variable cper_03        "Personnel costs (Lombardia)"
label variable add_04         "Annual workers average (Veneto)"
label variable cper_04        "Personnel costs (Veneto)"
label variable add_05         "Annual workers average (Friuli-Venezia Giulia)"
label variable cper_05        "Personnel costs (Friuli-Venezia Giulia)"
```

```
label variable add_06          "Annual workers average (Liguria)"
label variable cper_06         "Personnel costs (Liguria)"
label variable add_07          "Annual workers average (Emilia-Romagna)"
label variable cper_07         "Personnel costs (Emilia-Romagna)"
label variable add_08          "Annual workers average (Toscana)"
label variable cper_08         "Personnel costs (Toscana)"
label variable add_09          "Annual workers average (Umbria)"
label variable cper_09         "Personnel costs (Umbria)"
label variable add_10          "Annual workers average (Marche)"
label variable cper_10         "Personnel costs (Marche)"
label variable add_11          "Annual workers average (Lazio)"
label variable cper_11         "Personnel costs (Lazio)"
label variable add_12          "Annual workers average (Abruzzo)"
label variable cper_12         "Personnel costs (Abruzzo)"
label variable add_13          "Annual workers average (Molise)"
label variable cper_13         "Personnel costs (Molise)"
label variable add_14          "Annual workers average (Campania)"
label variable cper_14         "Personnel costs (Campania)"
label variable add_15          "Annual workers average (Puglia)"
label variable cper_15         "Personnel costs (Puglia)"
label variable add_16          "Annual workers average (Basilicata)"
label variable cper_16         "Personnel costs (Basilicata)"
label variable add_17          "Annual workers average (Calabria)"
label variable cper_17         "Personnel costs (Calabria)"
label variable add_18          "Annual workers average (Sicilia)"
label variable cper_18         "Personnel costs (Sicilia)"
label variable add_19          "Annual workers average (Sardegna)"
label variable cper_19         "Personnel costs (Sardegna)"
label variable add_20          "Annual workers average (Bolzano)"
label variable cper_20         "Personnel costs (Bolzano)"
label variable add_21          "Annual workers average (Trento)"
label variable cper_21         "Personnel costs (Trento)"
label variable add_est_sci     "Annual workers average (Foreign Countries) (SCI)"
label variable cper_est_sci    "Personnel costs (Foreign Countries) (SCI)"
label variable add_noloc_sci   "Annual workers average (Not imputable) (SCI)"
label variable cper_noloc_sci "Personnel costs (Not imputable) (SCI)"

lab var ptc_est                       "shares in foreign firms"
lab var cred_fin_est          "financial credits tws foreign firms"
lab var cred_com_est          "commercial credits tws foreign firms"
lab var deb_est_b             "debts tws foreign banks"
lab var deb_est_i             "debts tws foreign firms"
lab var sp_liq_ass            "cheques"
lab var sp_part_cont          "shares in subsidiaries undertakings (fixed assets)"
lab var sp_part_coll          "shares in participating interests"
lab var sp_part_al            "other shares"
lab var sp_ac_pct             "shares in subsidiaries undertakings (current assets)"
lab var sp_ac_pcl             "shares in undertakings with which the company is linked by virtue of
participating interest"
lab var sp_ac_pot             "other shares"
lab var ptc_ita                       "shares in Italian firms"
lab var divid                 "income from participating interests"
lab var sp_im_cop             "trade marks"
lab var sp_im_ip                      "trade marks (intellectual works)"
lab var oth_new_in            "purchase of intangible assets"
lab var oth_new_in_c          "purchase of industrial trade marks"
lab var sp_im_av                      "goodwill (BS)"
lab var oth_new_in_oth                "purchases (others)"
lab var sp_im_rsp             "costs of R&D"
lab var sp_im_sfw             "software"
lab var sp_im_tdm             "concessions/licences"
lab var oth_new_in_tdm                "purchase of marks (total)"
lab var capz_rsv                      "R&D expenses capitalised in the a.p."

***********************************
* ORDINARY DEPRECIATION (Art. 67, co.2)
***********************************

lab var sp_im_fab             "instrumental buildings"
lab var sp_im_ter             "fixed assets -land"
lab var sp_im_bv                      "fixed assets - value goods"
lab var sp_im_fabns           "fixed assets - other buildings"
lab var sp_im_mac             "fixed assets - plant and machinery"
lab var sp_im_mt                      "fixed assets - means of transport"
lab var sp_im_aic             "fixed assets - equipment"
lab var sp_im_ttd             "fixed assets - tools for EDP"
```

```
lab var sp_im_maa              "fixed assets - furniture and equipment"
lab var sp_im_mtns             "fixed assets - other means of transport"
lab var im_acq_fns             "purchase of constructions (total)"
lab var im_mtr_new             "purchase of instrumental means of transport (new)"
lab var im_mtr_us              "purchase of instrumental means of transport (second hand)"
lab var im_at_ic_tot           "purchase of industrial and commercial equipment (total)"
lab var im_mat_acq_tot         "purchase of tangible assets (total)"
lab var sp_im_inc                     "payments on accounts and tangible assets"
lab var rival                  "positive value adjustments"
lab var sval                   "negative value adjustments"
lab var pr_onstr_pr            "extraordinary income"
lab var pr_onstr_on            "extraordinary charges"
lab var sp_fd_im_tan           "sinking fund - tangible assets(BS)"
lab var sp_fd_im                      "sinking fund - intangible assets (BS)"
lab var sp_pre_cont            "loans to subsidiaries undertakings"
lab var sp_pre_cont_bt         "loans to subsidiaries undertakings - short term"
lab var sp_pre_coll            "loans to affiliated undertakings"
lab var sp_pre_coll_bt         "loans to affiliated undertakings - short term"
lab var sp_pre_conti           "loans to parent undertakings"
lab var sp_pre_conti_bt        "loans to parent undertakings - short term"
lab var sp_pre_at              "other loans"
lab var sp_pre_al_bt           "other loans - short term"
lab var sp_atit                       "other investments held as fixed assets"
lab var sp_az_pr                      "own shares"
lab var sp_im_fi_96            "financial fixed assets (1996)"
lab var sp_part_cont_96               "shares in subsidiaries undertakings (fixed assets) (1996)"
lab var sp_part_coll_96               "shares in participating interests (1996)"
lab var sp_part_al_96          "other shares (1996)"
lab var sp_pre_cont_96         "loans to subsiduarues undertakings (1996)"
lab var sp_pre_coll_96         "loans to affiliated undertakings (1996)"
lab var sp_pre_conti_96               "loans to parent undertakings (1996)"
lab var sp_pre_at_96           "other loans (1996)"
lab var sp_atit_96             "other investments as fixed assets (1996)"
lab var sp_az_pr_96            "own shares (1996)"
lab var sp_af_at                      "other investments"
lab var sp_af_at_96            "other investments (1996)"
lab var pn                            "capital and reserves"
lab var cap_soc                       "subscribed capital"
lab var ris                    "reserves"
lab var ut_nuo                 "profit/loss brought forward"
lab var ut_es                  "profit/loss for the financial year"
lab var ut_es_ris              "profit to cover loss or to reserves"
lab var ut_es_soc                     "profits to shareholders"
lab var pn_96                  "capital and reserves (1996)"
lab var cap_soc_96             "subscribed capital (1996)"
lab var ris_96                 "reserves (1996)"
lab var ut_nuo_96                     "profit/loss brought forward (1996)"
lab var ut_es_96                      "profit/loss for the financial year (1996)"
lab var ut_nuo_97              "profit/loss brought forward (1997)"
lab var ut_es_97                      "profit/loss for the financial year (1997)"
lab var ut_es_ris_97           "profit to cover loss or to reserves (1997)"
lab var ut_es_soc_97           "profits to shareholders (1997)"
lab var occ_opap_tot_97               "Workers and Trainees (PMI) and Workers (SCI)197"
lab var occ_ld_tot_97          "at-home workers (total, 1996)"
lab var occ_tp_tot_97          "Part-time contracts (total, 1997)"
lab var occ_tot_97             "total employed staff (1997)"
lab var occ_td_tot_97          "short-term contracts (total, 1997)"
lab var occ_imco_tot_97        "entrapeneurs and family assistants (total, 1997)"
lab var occ_dirimp_tot_97      "Executives and Employees(total, 1997)"
lab var occ_aptot_sci_97       "trainees (total, 1997, SCI)"
lab var im_at_ic_new           "purchase of industrial and commercial equipment (new)"
lab var im_at_ic_us            "purchase of industrial and commercial equipment (second hand)"
lab var im_ab_tot              "purchase of other goods (total)"
lab var im_ab_new              "purchase of other goods (new)"
lab var im_ab_us               "purchase of other goods (second hand)"
lab var im_brev_new            "purchase of royalties"
lab var im_conc_new            "purchase of trade marks and licences"


label variable sp_obb_sci             "Debts: bonds (SCI)"
label variable sp_obb_ls_sci          "Debts: bonds m/l term (SCI)"
label variable sp_obbconv_sci         "Debts: Convertible bonds(SCI)"
label variable sp_obbconv_ls_sci      "Debts: Convertible bonds m/l term (SCI)"
label variable sp_debbank_sci         "Debts: to credit institutions (SCI)"
label variable sp_debbank_ls_sci      "Debts: to credit institutions m/l term (SCI)"
```

```
label variable sp_debaltr_sci        "Debts: to other financial institutions (SCI)"
label variable sp_debaltr_ls_sci     "Debts: to other financial institutions m/l term (SCI)"
label variable sp_acccl_sci          "Debts: customers' accounts(SCI)"
label variable sp_debfor_sci         "Debts: to suppliers (SCI)"
label variable sp_debfor_ls_sci      "Debts: to suppliers m/l term (SCI)"
label variable sp_debtit_sci        "Debts: credit instruments (SCI)"
label variable sp_debtit_ls_sci      "Debts: credit instruments m/l term (SCI)"
label variable sp_debcontr_sci       "Debts: to controlled companies (SCI)"
label variable sp_debcontr_ls_sci    "Debts: to controlled companies m/l term (SCI)"
label variable sp_debcoll_sci        "Debts: to connected companies (SCI)"
label variable sp_debcoll_ls_sci     "Debts: to connected companies m/l term (SCI)"
label variable sp_control_sci        "Debts: to controlling companies (SCI)"
label variable sp_control_ls_sci     "Debts: to controlling companies m/l term (SCI)"
label variable sp_debtrib_sci        "Fiscal Debts (SCI)"
label variable sp_debtrib_ls_sci     "Fiscal Debts m/l term (SCI)"
label variable sp_debss_sci          "Debts: to social security inst. (SCI)"
label variable sp_debss_ls_sci       "Debts: to social security inst. m/l term (SCI)"
label variable sp_altrdeb_sci        "Other debts (SCI)"
label variable sp_altrdeb_ls_sci     "Other debts m/l term (SCI)"
label variable sp_attrr_sci          "Accrued Income and Prepayments (SCI)"
label variable sp_intfin_sci         "Interest payable for loans (SCI)"
label variable sp_intaltr_sci        "Other financial charges (factoring) (SCI)"
label variable sp_immtot_sci         "Total Fixed Assets (SCI)"
label variable sp_cresoc_sci         "Credits: to partners m/l term (SCI)"
label variable sp_accfor_sci         "Payments on account (SCI)"
label variable sp_attcirc_sci        "Current Assets (SCI)"



*************************
*       CHECK.DO        *
*************************
* execute typing: do check

* program to check totals and subtotals
* new variables (<variablename>_dis) are computed when accounting consistency is violated
* new total variables (in Section "Stato Patrimoniale") are computed when necessary

capture program drop check
program define check

/* substituting missing values with zeros */
quietly mvencode _all, mv(0)override

                            *** CONTO ECONOMICO ***

*************************
* VALORE DELLA PRODUZIONE*
*************************
tempvar totric totvr totvalpro
gen `totric' = ric_vpi + ric_vmnt + ric_lct + ric_lot + ric_ai + ric_tra + ric_pst
gen ric_dis = ric_tot - `totric'

gen `totvr' = ric_vr_pf + ric_vr_pcl
gen ric_vr_dis = ric_vr_tot - `totvr'

gen `totvalpro' = ric_tot + ric_vr_tot + ric_vl_co + ric_ini_li + ric_alpro
gen val_prod_dis = val_prod_tot - `totvalpro'
*compare val_prod_tot  `totvalpro'
*pause

*************************
* COSTI DELLA PRODUZIONE *
*************************
tempvar totacqui totservi totgdbt  totpers totamm totvar totondiv totcostprod totmol totutile
totcostpers
gen `totacqui' = acq_matp + acq_ener + acq_mriv
gen acq_beni_dis = acq_beni_tot - `totacqui'

gen `totservi' = acq_lavter + acq_allav + acq_trasp + acq_intmd + acq_pubbli + acq_risv + /*
               */ acq_consul + acq_inform + acq_prassi + acq_licuso + acq_smrif + acq_alser +/*
               */ acq_banc +acq_altr_sci
gen acq_serv_dis = acq_serv_tot - `totservi'
```

```
*compare acq_serv_tot `totservi'
*pause

gen `totgdbt' = acq_fitpa + acq_leasing + acq_canlo + acq_alfitpa_sci + acq_leasstr_sci
gen acq_gdbt_dis = acq_gdbt_tot - `totgdbt'
*compare acq_gdbt_tot `totgdbt'
*pause

gen `totpers' = acq_ret_dirimp + acq_ret_alcat + acq_cs + acq_qtfr + acq_prpers

/**** where, for PMI: acq_ret_alcat = acq_ret_oppmi +acq_ret_appmi + acq_ret_ldpmi
and for SCI: acq_cs = acq_csdir_sci + acq_csal_sci
               acq_qtfr = acq_tfrdir_sci + acq_tfral_sci ****/

gen acq_pers_dis = acq_pers_tot - `totpers'

*compare acq_pers_tot `totpers'
*pause

gen `totamm' = acq_amm_imi + acq_amm_imm + acq_svimm_sci + acq_svcr_sci
gen acq_amm_dis = acq_amm_tot - `totamm'
*compare acq_amm_tot `totamm'
*pause
 gen `totvar' = acq_vr_mp + acq_vr_r
gen acq_vr_dis = acq_vr_tot - `totvar'

gen acq_accant_tot = acq_accant + acq_alacc_sci

gen `totondiv' = acq_forper + acq_onamm + acq_aodg + acq_imp_ind + acq_impfab_sci + acq_alimp_sci
gen acq_ondiv_dis = acq_ondiv_tot - `totondiv'
*compare acq_ondiv_tot `totondiv'
*pause

gen `totcostprod' = acq_beni_tot + acq_serv_tot + acq_gdbt_tot + acq_pers_tot + acq_amm_tot + /*
               */ acq_vr_tot + acq_accant + acq_alacc_sci + acq_ondiv_tot
gen cost_prod_dis = cost_prod_tot - `totcostprod'


*compare cost_prod_tot  `totcostprod'
*pause

*************************
* MARGINE OPERATIVO LORDO *
*************************

gen `totmol' = val_prod_tot - cost_prod_tot
*compare `totmol' mol
*pause

gen `totutile' = mol + int_att - int_pas


*******************************
* OCCUPAZIONE E DATI REGIONALI *
*******************************
************************* tutto da verificare ******************************
tempvar totocc totoccreg totpersreg
gen `totocc' = occ_dirimp_tot +occ_imco_tot+ occ_opap_tot + occ_aptot_sci
*+ occ_ld_tot
/* where for PMI
occ_imco_tot = occ_imptot_pmi + occ_coftot_pmi
occ_opap_tot= occ_optot_pmi+occ_aptot_pmi
*/

*compare `totocc' occ_tot
*pause
gen `totoccreg' = add_01+add_02+add_03+add_04+add_05+add_06+add_07+add_08+add_09+add_10+/*

       */add_11+add_12+add_13+add_14+add_15+add_16+add_17+add_18+add_19+add_20+add_21
gen `totpersreg' =
cper_01+cper_02+cper_03+cper_04+cper_05+cper_06+cper_07+cper_08+cper_09+cper_10+/*

       */cper_11+cper_12+cper_13+cper_14+cper_15+cper_16+cper_17+cper_18+cper_19+cper_20+cper_21

*compare `totoccreg' occ_tot
*pause
```

```
*compare `totpersreg' acq_pers_tot
*pause


*compare `totoccreg' `totocc'


*******************************
* STATO PATRIMONIALE           *
*******************************
*** ATTIVO
***Immobilizzazioni***
gen sp_imm_tot=sp_im_im + sp_im_man + sp_im_fi

**Rimanenze
gen sp_rimtot_pmi = sp_rim_mp + sp_rim_ps + sp_rim_lc + sp_rim_pf
gen sp_rimtot_sci = sp_rim_mp + sp_rim_ps + sp_rim_lc + sp_rim_pf + sp_accfor_sci

***crediti
gen sp_cretot_pmi = sp_cre_bs + sp_cre_ls

***DEBITI
gen sp_deb_ls_sci = sp_obb_ls_sci + sp_obbconv_ls_sci + sp_debbank_ls_sci + sp_debaltr_ls_sci +
sp_debfor_ls_sci + sp_debtit_ls_sci + sp_debcontr_ls_sci + sp_debcoll_ls_sci + sp_control_ls_sci +
sp_debtrib_ls_sci + sp_debss_ls_sci + sp_altrdeb_ls_sci
gen sp_deb_bs_sci = sp_debtot_sci - sp_deb_ls_sci
gen sp_debtot_pmi = sp_deb_ls + sp_deb_bs
gen sp_debfin_sci = sp_obb_sci + sp_obbconv_sci + sp_debbank_sci + sp_debaltr_sci + sp_debtit_sci

***ATTIVO
gen sp_attcirc_pmi = sp_rimtot_pmi + sp_cre_bs + sp_liq
gen sp_attivo_sci = sp_attot_sci - sp_attrr_sci
gen sp_attivo_pmi = sp_imm_tot + sp_attcirc_pmi + sp_cre_ls







************* Defining new variables ****************************

* gross value added
gen valagg = val_prod_tot - acq_beni_tot -acq_serv_tot - acq_gdbt_tot - acq_vr_tot

* gross Profit (loss) for the financial year
gen utile_lordo = utile_netto + imp_reddito

* total export
gen Exp_tot = exp_ue + exp_eue
generate byte Exp_01 = Exp_tot>0

*investment
generate byte Inv_01 = im_acq_tot>0

* total subcontracting (revenues)
gen Subfa_tot = ric_lct + ric_lot
generate byte Subfa_01 = Subfa_tot > 0

* subcontracting (costs)
gen Subfp_tot =acq_lavter+acq_allav
generate byte Subfp_01 = Subfp_tot >0

/* generating total workers and workers hours for SCI  */
gen occ_optot_sci = occ_opap_tot - occ_optot_pmi - occ_aptot_pmi if sci==1 & occ_opap_tot>0
gen occ_dirimp_ore_sci = occ_dirimp_ore if sci==1
gen occ_dirimp_ore_pmi = occ_dirimp_ore if pmi==1
gen occ_tot_ore_sci = occ_tot_ore if sci==1
gen occ_tot_ore_pmi = occ_tot_ore if pmi==1

/* substituting missing values with zeros */
qui mvencode occ_optot_sci occ_opore_sci occ_dirimp_ore_sci occ_dirimp_ore_pmi occ_tot_ore_sci
occ_tot_ore_pmi, mv(0)override

assert occ_tot_ore_sci >0 if sci==1
```

```
/* legal status: generating categories for legal status
0 if sole entrepreneurship
1 if partnership
2 if corporations
3 if cooperatives
4 if others */
gen byte fgcat=0 if form_giur<=99
replace fgcat=1 if form_giur>99 & form_giur<200
replace fgcat=2 if form_giur>=200 & form_giur<300
replace fgcat=3 if form_giur>=300 & form_giur<400
replace fgcat=4 if form_giur>=400

/* generating employees classes (detailed) */
gen cla9=recode(occ_tot,9,19,49,99,199,249,499,1000,2000)
label define ecla 9 "0 to 9" 19 "10 to 19" 49 "20 to 49" 99 "50 to 99" 199 "100 to 199" 249 "200 to
249" 499 "250 to 499" 1000 "500 to 1000" 2000 "Above 1000"
label values cla9 ecla

/* generating employees classes (aggregate)*/
gen cla3=recode(occ_tot,49,250,2000)
label define ecla3 49 "0 to 49" 250 "50 to 250" 2000 "Above 250"
label values cla3 ecla3

end
***********************************END OF PROGRAM CHECK ****************************
args year
use dati`year', clear
*pause on
check
format %15.0fc *dis *tot
set dp comma

/* label assignment */
do label_check.do

label data "RTSS Checked Database, 19`year'"
save dati`year'_ch, replace


************************
* LABEL_CHECK.DO        *
************************

/* program to assign labels to variables created in check.do */

label variable sp_attcirc_pmi "Current assets (PMI)"
label variable sp_rimtot_pmi  "Current assets: raw materials, finished and semifinished products
(PMI)"
label variable sp_rimtot_sci  "Current assets: raw materials, finished and semifinished products
(SCI)"
label variable sp_cretot_pmi  "Current assets: short, medium and long term credits (PMI)"
label variable sp_deb_ls_sci  "Liabilities: long term debts (SCI)"
label variable sp_deb_bs_sci  "Liabilities: short term debts (SCI)"
label variable sp_debtot_pmi  "Liabilities: total debts (PMI)"
label variable sp_debfin_sci  "Liabilities: Financial Debts (SCI)"
label variable sp_attivo_sci  "Total liabilities free of prepayment and accrued income (SCI)"
label variable sp_attivo_pmi  "Total liabilities free of prepayment and accrued income (PMI)"
label variable sp_imm_tot     "Fixed assets: Total"
label variable valagg "Value Added"
label variable utile_lordo "Gross Profit (Loss)"
label variable Exp_tot "Total Exports"
label variable Subfa_tot "Total Income from works on behalf of third parties (subcontracting)"
label variable Subfp_tot "Total Cost from works made by third parties (subcontracting)"
label variable Exp_01 "Indicator variable for exporting firms"
label variable Inv_01 "Indicator variable for investing firms"
label variable Subfa_01 "Indicator variable for subcontracting firms (revenues)"
label variable Subfp_01 "Indicator variable for subcontracting firms (costs)"
label variable occ_dirimp_ore_sci "Executives and Employees: Hours (SCI)"
label variable occ_dirimp_ore_pmi "Executives and Employees: Hours (PMI)"
label variable occ_tot_ore_sci "Total worked hours (SCI)"
label variable occ_tot_ore_pmi "Total worked hours (PMI)"
label variable occ_optot_sci "Workers: Total (SCI)"
label variable cla9 "Employees Classes (detailed)"
label variable cla3 "Employees Classes (aggregate)"
```

```
labelvar Exp_01 "Exports in 1998" 0 No 1 Yes
labelvar Inv_01 "Investment in 1998" 0 No 1 Yes
labelvar Subfa_01 "Subcontractor" 0 No 1 Yes
labelvar Subfp_01 "Contractor" 0 No 1 Yes

labelvar fgcat "Legal Status" 0 "Sole Entrepreneurship" 1 "Partnership" 2 "Corporations" /*
       */3 "Co-operatives" 4 "Other"


/***************
 *   RETRIB    *
 ***************
initial program to estimate wages and salaries for social contributions base */

args year

/* reading parameters from file */
use parameters/param_inps, clear

/* setting the parameters in matrix Pesi */
mkmat ateco2d imp_perc dir_perc retm*, matrix(Pesi)

*matrix list Pesi
*pause

/*reading the Regional (checked) dataset */
use dati`year'_ch, clear

local N = _N
gen occ_dir_tot = 0

/* ncoef is the number of rows in matrix Pesi
   nc is the column number of ateco categories
   nc1 is the column number of parameters used in the loop */

local ncoef = rowsof(Pesi)
local nc = colnumb(Pesi,"ateco2d")
local nc1 = colnumb(Pesi,"dir_perc")
       /* for debugging      */

       display "The number of rows is `ncoef'"
       display "The sector is column`nc'"
       display "The column number of parameters is `nc1'"

/* separating executives and employees using sectoral rates */
forvalues m=1/`ncoef' {
                      qui replace occ_dir_tot=occ_dirimp_tot*Pesi[`m',`nc1']/*
                      */ if ateco2==Pesi[`m',`nc']
                      }

gen occ_imp_tot = occ_dirimp_tot - occ_dir_tot

format %9.0fc  occ_dir_tot occ_imp_tot occ_optot_sci occ_opore_sci

/******** estimating wages and salaries using INPS average data (with dimensional
classes)**************************/
local nc2 = colnumb(Pesi,"retm_dir")
local nc3 = colnumb(Pesi,"retm_imp149")
 local nc4 = colnumb(Pesi,"retm_imp5099")
local nc5 = colnumb(Pesi,"retm_imp100")
local nc6 = colnumb(Pesi,"retm_op149")
local nc7 = colnumb(Pesi,"retm_op5099")
local nc8 = colnumb(Pesi,"retm_op100")
local nc9 = colnumb(Pesi,"retm_app")

       display "The column number of parameters is `nc2'"
       display "The column number of parameters is `nc3'"
       display "The column number of parameters is `nc4'"
       display "The column number of parameters is `nc5'"
       display "The column number of parameters is `nc6'"
       display "The column number of parameters is `nc7'"
       display "The column number of parameters is `nc8'"
       display "The column number of parameters is `nc9'"
*pause
```

```
gen ret_dir=0
gen ret_imp=0
gen ret_op=0
gen ret_app=0

/* wages and salaries for executives, employees, workers, apprentices */
forvalues m=1/`ncoef' {
               qui replace ret_dir=occ_dir_tot*Pesi[`m',`nc2'] if ateco2==Pesi[`m',`nc']
               qui replace ret_app=occ_aptot_pmi*Pesi[`m',`nc9'] if ateco2==Pesi[`m',`nc'] & pmi==1
               qui replace ret_app=occ_aptot_sci*Pesi[`m',`nc9'] if ateco2==Pesi[`m',`nc'] & sci==1

               qui replace ret_imp=occ_imp_tot*Pesi[`m',`nc3'] if ateco2==Pesi[`m',`nc'] & occ_tot
<= 49
               qui replace ret_imp=occ_imp_tot*Pesi[`m',`nc4'] if ateco2==Pesi[`m',`nc'] & occ_tot
>49 & occ_tot <=99
               qui replace ret_imp=occ_imp_tot*Pesi[`m',`nc5'] if ateco2==Pesi[`m',`nc'] & occ_tot
>= 100

               qui replace ret_op=occ_optot_pmi*Pesi[`m',`nc6'] if ateco2==Pesi[`m',`nc'] & pmi==1 &
occ_tot <= 49
               qui replace ret_op=occ_optot_pmi*Pesi[`m',`nc7'] if ateco2==Pesi[`m',`nc'] & pmi==1
& occ_tot >49 & occ_tot <=99
               qui replace ret_op=occ_optot_sci*Pesi[`m',`nc8'] if ateco2==Pesi[`m',`nc'] & sci==1 &
occ_tot >= 100
               }

qui replace ret_app=ret_app*0.75 if pmi==1 & fgcat!=2

format %10.0fc ret_*

matrix drop Pesi

* average data
gen retop_rat = acq_ret_oppmi/occ_optot_pmi
gen retap_rat = acq_ret_appmi/occ_aptot_pmi
gen oreop_rat = occ_opore_pmi/occ_optot_pmi

/* label assignment */
do label_retrib.do

save retrib`year', replace



************************
* LABEL_RETRIB.DO        *
************************

/* program to assign labels to variables created in retrib.do */

label variable ret_dir "Total wages and salaries: executives"
label variable ret_imp "Total wages and salaries: employees"
label variable ret_op "Total wages and salaries: workers"
label variable ret_app "Total wages and salaries: apprentices"
label variable occ_dir_tot "Executives: total"
label variable occ_imp_tot "Employees: total"
label variable retop_rat "Average salary (workers) PMI"
label variable retap_rat "Average salary (apprentices) PMI"
label variable oreop_rat "Average worked hours (workers) PMI"

/***************
 *   CONTRIB    *
 ***************
execute the file typing:
do contrib <year>

do-file to estimate social contributions for white collars, workers, executives, apprentices and TFR
*/

capture log close
log using contrib.log, replace

program drop _all

args year
```

```
************************************
program mkmatal
*program to store tax rates in matrices (one for each professional category)
        /* reading parameters from file */
        use parameters/ali`1', clear
        format %3.0g ateco2d
        set dp comma

        /* eventual policy changes of rates go here*/
                egen ali_subsum =rsum(ali2-ali7 ali11-ali13)
                *list ali_subsum

                /* setting the parameters in the matrix */
                mkmat ateco2d ali_subsum ali1 ali8 ali9 ali10 ali14 flag, matrix(`2')

        *matrix list `2', format(%5.3f)
        *pause
        drop _all
end
************************************

/* runnning the program */
mkmatal operai`year' Aliop
mkmatal dir`year' Alidir
mkmatal imp`year' Aliimp
mkmatal app`year' Aliapp

/*reading the dataset */
use retrib`year', clear

************************************
program contget
/* program to compute the social contribution revenue for executives, white collars, workers
(with several subtotals)*/

        local N = _N

/* contri1 is for a sum of various contributions
contri2 is for fondi pensione IVS
contri3 is for CIG
contri4 is for INAIL
*/
        gen contri1_`1' = 0
        gen contri2_`1' = 0
        gen contri3_`1' = 0
        gen contri4_`1' = 0

        /* ncoef is the number of rows in matrix Ali`1'
         nc is the column number of ateco categories
         nc1 is the column number of parameters used in the loop */

        local nsec = rowsof(Ali`1')
        local nc = colnumb(Ali`1',"ateco2d")
        local nc1 = colnumb(Ali`1',"ali_subsum")  /*sum of various SC*/
        local nc2 = colnumb(Ali`1',"ali1")        /*IVS*/
        local nc3 = colnumb(Ali`1',"ali8")        /*CIG*/
        local nc4 = colnumb(Ali`1',"ali9")        /*CIG*/
        local nc5 = colnumb(Ali`1',"ali10")       /*CIG*/
        local nc6 = colnumb(Ali`1',"ali14")       /*INAIL*/
        local nc7 = colnumb(Ali`1',"flag")

                /* for debugging
                display "The number of rows is `nsec'"
                display "The sector is column `nc'"
                display "The column number of parameters is `nc1'"
                display "The column number of parameters is `nc2'"
                */

        if "`1'" != "app"{
                /* computing social contributions for executives, white collars, workers */
                forvalues m=1/`nsec' {
                                quietly replace contri1_`1' =ret_`1' *Ali`1'[`m',`nc1']/*
                                */ if ateco2==Ali`1'[`m',`nc']
                                quietly replace contri2_`1' =ret_`1' *Ali`1'[`m',`nc2']/*
                                */ if ateco2==Ali`1'[`m',`nc']
                                quietly replace contri4_`1' =ret_`1' *Ali`1'[`m',`nc6']/*
```

```
                                        */ if ateco2==Ali`1'[`m',`nc']
                                        }
                /* computing CIG for executives, white collars, workers */
                forvalues m=1/`nsec' {
                                        if Ali`1'[`m',`nc6'] ==1 {
                                                quietly replace contri3_`1' = ret_`1' *
Ali`1'[`m',`nc3']/*
                                                */ if ateco2==Ali`1'[`m',`nc'] & occ_tot <= 15
                                                quietly replace contri3_`1' = ret_`1' *
Ali`1'[`m',`nc4']/*
                                                */ if ateco2==Ali`1'[`m',`nc'] & occ_tot >15 & occ_tot
<= 50
                                                quietly replace contri3_`1' = ret_`1' *
Ali`1'[`m',`nc5']/*
                                                */ if ateco2==Ali`1'[`m',`nc'] & occ_tot >50
                                                }
                                        else {
                                                quietly replace contri3_`1' = ret_`1' *
Ali`1'[`m',`nc3']/*
                                                */ if ateco2==Ali`1'[`m',`nc'] & occ_tot <= 50
                                                quietly replace contri3_`1' = ret_`1' *
Ali`1'[`m',`nc4']/*
                                                */ if ateco2==Ali`1'[`m',`nc'] & occ_tot >50 & occ_tot
<= 200
                                                quietly replace contri3_`1' = ret_`1' *
Ali`1'[`m',`nc5']/*
                                                */ if ateco2==Ali`1'[`m',`nc'] & occ_tot >200
                                                }
                                        }
                        }
        else {
                **generate weeks for apprentices**
                gen orem_app_sci=occ_apore_sci/occ_aptot_sci
                gen orem_app_pmi=occ_apore_pmi/occ_aptot_pmi
                * we assume 35 working hours per week
                gen set_app_sci=orem_app_sci/35
                gen set_app_pmi=orem_app_pmi/35
                qui mvencode orem_app_* set_app_*, mv(0)override

                /* computing social contributions for apprentices (CIG is zero for apprentices)*/
                forvalues m=1/`nsec' {
                                        quietly replace contri1_`1' =(set_app_sci + set_app_pmi)
*Ali`1'[`m',`nc1']/*
                                        */ if ateco2==Ali`1'[`m',`nc']
                                        quietly replace contri2_`1' =(set_app_sci + set_app_pmi)
*Ali`1'[`m',`nc2']/*
                                        */ if ateco2==Ali`1'[`m',`nc']
                                        quietly replace contri4_`1' =(set_app_sci + set_app_pmi)
*Ali`1'[`m',`nc6']/*
                                        */ if ateco2==Ali`1'[`m',`nc']
                                        }
                }
        gen contri_`1'_tot=contri1_`1'+contri2_`1'+contri3_`1'+contri4_`1'
end
*************************************

/* runnning the program for social contributions for executives, workers, white collars and
apprentices*/
contget op
contget dir
contget imp
contget app


* aggregating INAIL contributions for IRAP deductions
gen inail_tot_irap = contri4_op + contri4_dir + contri4_imp

*generating APPRENTICES labour cost for IRAP deductions
gen app_tot_irap = contri_app_tot + ret_app

*generating total labour compensation (retribuzioni)
gen ret_tot=ret_imp+ret_op+ret_app+ret_dir


****** Computing TFR ******************
gen ac_tfr1=ret_tot*0.0691
```

```
gen ac_tfr2=ret_tot*0.005
gen riv_tfr=sp_tfr*(0.015+0.75*0.0373) /*verificare*/
gen ac_acctfr= ac_tfr1+ac_tfr2+riv_tfr

/* label assignment */
do label_contrib.do

label data "Dataset after Contributions"
save contrib`year', replace

log close



************************
* LABEL_CONTRIB.DO        *
************************

/* program to assign labels to variables created in contrib.do */

label variable orem_app_sci "Average hours for apprentices (SCI)"
label variable orem_app_pmi "Average hours for apprentices (PMI)"
label variable set_app_sci "Worked weeks for apprentices (SCI)"
label variable set_app_pmi "Worked weeks for apprentices (PMI)"
label variable inail_tot_irap "Invalidity contributions (INAIL): Total"
label variable app_tot_irap "Apprentices total labour cost"
label variable ret_tot "Total wages and salaries: all worker categories"


label variable ac_tfr1 "Provision for severance pay fund (internal)"
label variable ac_tfr2 "Provision for severance pay fund (for INPS)"
label variable ac_acctfr "Total provision for severance pay fund"
label variable riv_tfr "Annual revaluation of severance pay stock"

label variable contri1_op "Various Social Contributions (sickness, maternity, etc.): workers"
label variable contri1_dir "Various Social Contributions (sickness, maternity, etc.): executives"
label variable contri1_imp "Various Social Contributions (sickness, maternity, etc.): employees"
label variable contri1_app "Various Social Contributions (sickness, maternity, etc.): apprentices"

label variable contri2_op "Old-age social contributions (IVS): workers"
label variable contri2_dir "Old-age social contributions (IVS): executives"
label variable contri2_imp "Old-age social contributions (IVS): employees"
label variable contri2_app "Old-age social contributions (IVS): apprentices"

label variable contri3_op "Occupational Desease (CIG): workers"
label variable contri3_dir "Occupational Desease (CIG): executives"
label variable contri3_imp "Occupational Desease (CIG): employees"
label variable contri3_app "Occupational Desease (CIG): apprentices (zero)"

label variable contri4_op "Invalidity contributions (INAIL): workers"
label variable contri4_dir "Invalidity contributions (INAIL) executives"
label variable contri4_imp "Invalidity contributions (INAIL): employees"
label variable contri4_app "Invalidity contributions (INAIL): apprentices"

label variable contri_op_tot "Total Social Contributions: workers"
label variable contri_dir_tot "Total Social Contributions: executives"
label variable contri_imp_tot "Total Social Contributions: employees"
label variable contri_app_tot "Total Social Contributions: apprentices"


/***********************
*     STAT_CONTRIB      *
***********************
do-file to compute statistics and tables for social contributions
*/

capture log close
log using stat_contrib.log, replace
set linesize 255

program drop _all

args year

/*reading the dataset */
use contrib`year', clear
```

```
set more off


*converting ***SELECTED VARIABLES*** to thousands of euros and WEIGHTING these variables

forvalues i=1(1)4{
        gen Pcontri`i'_op = (contri`i'_op /1000)* peso
        gen Pcontri`i'_dir = (contri`i'_dir /1000) * peso
        gen Pcontri`i'_imp = (contri`i'_imp /1000) * peso
        gen Pcontri`i'_app = (contri`i'_app /1000) * peso
        }
gen Pcontri_op_tot = (contri_op_tot /1000) * peso
gen Pcontri_dir_tot = (contri_dir_tot /1000) * peso
gen Pcontri_imp_tot = (contri_imp_tot /1000) * peso
gen Pcontri_app_tot = (contri_app_tot /1000) * peso

forvalues i=1(1)2{
        gen Pac_tfr`i' = (ac_tfr`i' /1000)* peso
        }
gen Pac_acctfr = (ac_acctfr /1000) * peso
gen Priv_tfr = (riv_tfr /1000) * peso gen Pstock_tfr= (sp_tfr /1000) *peso



* aggregating by contributions types
 forvalues i=1(1)4{
        gen Pcontri`i'_tot = Pcontri`i'_op + Pcontri`i'_dir + Pcontri`i'_imp
        }

/* label assignment */
do label_stat_contrib.do

/*making tables for social contributions revenues by categories */
local listind "pmi fgcat ateco2"
foreach x of local listind {
        forvalues i=1(1)4{
                tabstat Pcontri`i'_op, by(`x') stats(n sum)col(stat) format(%12.0f)
labelwidth(32)varwidth(15)
                tabstat Pcontri`i'_imp, by(`x') stats(n sum)col(stat) format(%12.0f) labelwidth(32)
varwidth(15)
                tabstat Pcontri`i'_dir, by(`x') stats(n sum)col(stat) format(%12.0f) labelwidth(32)
varwidth(15)
                tabstat Pcontri`i'_app, by(`x') stats(n sum)col(stat) format(%12.5f) labelwidth(32)
varwidth(15)
                tabstat Pcontri`i'_tot, by(`x') stats(n sum)col(stat) format(%12.0f) labelwidth(32)
varwidth(15)
                }
        }

/* tables with SC revenue by different indicators */
foreach x of local listind {
        tabstat Pcontri_op_tot, by(`x') stats(n sum)col(stat) format(%12.0f) labelwidth(32)
varwidth(15)
        tabstat Pcontri_imp_tot, by(`x') stats(n sum)col(stat) format(%12.0f) labelwidth(32)
varwidth(15)
        tabstat Pcontri_dir_tot, by(`x') stats(n sum)col(stat) format(%12.0f) labelwidth(32)
varwidth(15)
        tabstat Pcontri_app_tot, by(`x') stats(n sum)col(stat) format(%12.5f) labelwidth(32)
varwidth(15)
        tabstat Pcontri_app_tot, by(`x') stats(n sum)col(stat) format(%12.0f) labelwidth(32)
varwidth(15)
        }


log close

save contrib`year'_out.dta, replace



************************
* LABEL_STAT_CONTRIB.DO *
************************

/* program to assign labels to variables created in stat_contrib.do */
```

```
label variable Pcontri1_op "Various Social Contributions (sickness, maternity, etc.) (WEIGHTED):
workers"
label variable Pcontri1_dir "Various Social Contributions (sickness, maternity, etc.) (WEIGHTED):
executives"
label variable Pcontri1_imp "Various Social Contributions (sickness, maternity, etc.) (WEIGHTED):
employees"
label variable Pcontri1_app "Various Social Contributions (sickness, maternity, etc.) (WEIGHTED):
apprentices"
label variable Pcontri1_tot "Various Social Contributions (sickness, maternity, etc.) (WEIGHTED):
TOTAL"

label variable Pcontri2_op "Old-age social contributions (IVS) (WEIGHTED): workers"
label variable Pcontri2_dir "Old-age social contributions (IVS) (WEIGHTED): executives"
label variable Pcontri2_imp "Old-age social contributions (IVS) (WEIGHTED): employees"
label variable Pcontri2_app "Old-age social contributions (IVS) (WEIGHTED): apprentices"
label variable Pcontri2_tot "Old-age social contributions (IVS) (WEIGHTED): TOTAL"

label variable Pcontri3_op "Occupational Desease (CIG) (WEIGHTED): workers"
label variable Pcontri3_dir "Occupational Desease (CIG) (WEIGHTED): executives"
label variable Pcontri3_imp "Occupational Desease (CIG) (WEIGHTED): employees"
label variable Pcontri3_app "Occupational Desease (CIG) (WEIGHTED): apprentices (zero)"
label variable Pcontri3_tot "Occupational Desease (CIG) (WEIGHTED): TOTAL"

label variable Pcontri4_op "Invalidity contributions (INAIL) (WEIGHTED): workers"
label variable Pcontri4_dir "Invalidity contributions (INAIL) (WEIGHTED): executives"
label variable Pcontri4_imp "Invalidity contributions (INAIL) (WEIGHTED): employees"
label variable Pcontri4_app "Invalidity contributions (INAIL) (WEIGHTED): apprentices"
label variable Pcontri4_tot "Invalidity contributions (INAIL) (WEIGHTED): TOTAL"

label variable Pcontri_op_tot "Total Social Contributions (WEIGHTED): workers"
label variable Pcontri_dir_tot "Total Social Contributions (WEIGHTED): executives"
label variable Pcontri_imp_tot "Total Social Contributions (WEIGHTED): employees"
label variable Pcontri_app_tot "Total Social Contributions (WEIGHTED): apprentices"

label variable Pac_tfr1 "Provision for severance pay fund (internal) (WEIGHTED)"
label variable Pac_tfr2 "Provision for severance pay fund (for INPS) (WEIGHTED)"
label variable Pac_acctfr "Total provision for severance pay fund (WEIGHTED)"
label variable Priv_tfr "Annual revaluation of severance pay stock (WEIGHTED)"
label variable Pstock_tfr "Severance pay stock (WEIGHTED)"



/***************
 *   TAXVAR    *
 ***************

TYPE: do taxvar <year>

program to compute fiscal variables and deductions for IRAP
Here included: programs CIVTOFIS, FISCAL, DEDUC
*/

capture program drop _all

args year

use contrib`year', clear

/* generating variables to be modified with parameters in order to obtain 'fiscal' variables
labeled with the capital letter */

gen Ric_tot = ric_tot
gen Ric_vr_tot =ric_vr_tot
gen Ric_vl_co = ric_vl_co
gen Ric_ini_li =ric_ini_li
gen Ric_alpro = ric_alpro
gen Acq_beni_tot = acq_beni_tot
gen Acq_serv_tot = acq_serv_tot
gen Acq_gdbt_tot = acq_gdbt_tot
gen Acq_amm_imm = acq_amm_imm
gen Acq_amm_imi = acq_amm_imi
gen Acq_vr_tot = acq_vr_tot
gen Acq_accant_tot = acq_accant_tot
gen Acq_ondiv_tot = acq_ondiv_tot
```

```
gen Ded_inail_base = 0
gen Ded_app_base = 0
gen Ded_fl_base = 0

format %15.0fc *tot Ric_vl_co Ric_ini_li Ric_alpro Acq_amm_imm Acq_amm_imi Ded*
set dp comma
label data "Modified Fiscal RTSS Dataset, 19`year'"
save fiscal`year', replace



***************************** PROGRAM CIVTOFIS **********************************
program define civtofis
/* routine to convert some administrative values into fiscal values to compute the tax base */

args numfile namefile year

/* reading the fiscal parameters from file */
use parameters/celle`numfile', clear
set more off

/* setting the parameters in matrix Pesi */
mkmat ateco2d pesiq* pdedla* clariciq, matrix(Pesi)

*matrix list Pesi

/*reading the Regional (checked) dataset */
use `namefile', clear

/* checking the income classes (EUROS)*/
if Pesi[1,21] == 1 {
        global cla1=0
        global cla2=258228
        }
if Pesi[1,21] == 2 {
        global cla1=258228
        global cla2=1291142
        }
if Pesi[1,21] == 3 {
        global cla1=1291142
        global cla2=5164569
        }
if Pesi[1,21] == 4 {
        global cla1=5164569
        global cla2=25822845
        }
if Pesi[1,21] == 5 {
        global cla1=25822845
        global cla2=516456899
        }
if Pesi[1,21] == 6 {
        global cla1=516456899
        global cla2=5164568991
        }

/* for debugging*/

display "The lower income boundary is $cla1"
display "The upper income boundary is $cla2"
*pause

global n_obs =_N

/* calling subroutine FISCAL */
fiscal

/* generating fiscal variables as in Section IV Quadro IQ */
* these three variables are generated, used and dropped then generated again in program irap.do
gen Comp_pos_tot = Ric_tot + Ric_vr_tot + Ric_vl_co + Ric_ini_li + Ric_alpro
gen Comp_neg_tot = Acq_beni_tot + Acq_serv_tot + Acq_gdbt_tot + Acq_amm_imm + Acq_amm_imi + /*
        */Acq_vr_tot + Acq_accant_tot + Acq_ondiv_tot
gen Base_irap_lorda = Comp_pos_tot - Comp_neg_tot

/* calling subroutine DEDUC */
deduc
```

```
drop Comp_pos_tot Comp_neg_tot Base_irap_lorda

/*saving the modified Dataset including the fiscal variables and deductions for IRAP */
label data "Modified Fiscal RTSS Dataset, 19`year'"
save fiscal`year', replace
end
****************************** END OF CIVTOFIS *********************

************************* PROGRAM FISCAL ********************** /* program to compute fiscal
variables for IRAP using a weighting scheme */

program define fiscal

/*for debugging
display "The lower income boundary in FISCAL is $cla1"
display "The upper income boundary in FISCAL is $cla2"
pause
*/

/* defining two arrays with ordered variables to be modified and parameters to be used */
local array1 "Ric_tot Ric_vr_tot Ric_vl_co Ric_ini_li Ric_alpro Acq_beni_tot Acq_serv_tot
Acq_gdbt_tot Acq_amm_imm Acq_amm_imi Acq_vr_tot Acq_accant_tot Acq_ondiv_tot"
local array2 "pesiq01 pesiq02 pesiq03 pesiq04 pesiq05 pesiq06 pesiq07 pesiq08 pesiq09 pesiq10
pesiq11 pesiq12 pesiq13"

/* ncoef is the number of rows in matrix Pesi
   nc is the column number of ateco categories
   nc1 is the column number of parameters used in the loop */

local ncoef = rowsof(Pesi)
local nc = colnumb(Pesi,"ateco2d")
local nvar : word count `array1'

/*top of external loop for all variables */
forvalues k=1/`nvar' {
        local var1 : word `k' of `array1'
        local var2 : word `k' of `array2'
        local nc1 = colnumb(Pesi,"`var2'")
        local kk=0

        /* for debugging
        display "The variable is `var1'"
        display "The weight is `var2'"
        display "The column number of parameters is `nc1'"
        pause */

        /*top of inner loop for all observations */
        forvalues m=1/`ncoef'{
                quietly replace `var1'=`var1'*Pesi[`m',`nc1'] if fgcat== 2 & ateco2==Pesi[`m',`nc']&
ric_tot> $cla1 & ric_tot<= $cla2
                qui count if ric_tot> $cla1 & ric_tot<= $cla2 & fgcat== 2
                local kk = r(N)
                }

        /*end of inner loop */
        }
/*end of external loop*/
display "CORPORATIONS IN CLASS WITH UPPER BOUNDARY $cla2 ARE: `kk'"
end

******************************** END OF FISCAL ***********************

************************* PROGRAM DEDUC ***************************
/* program to compute fiscal deductions for IRAP using weights over the GROSS TAX BASE */
program define deduc

/* defining two arrays with ordered variables to be modified and parameters to be used */
local array1 "Ded_inail_base Ded_app_base Ded_fl_base"
local array2 "pdedla1b pdedla2b pdedla3b"

/* ncoef is the number of rows in matrix Pesi    nc is the column number of ateco categories
   nc1 is the column number of parameters used in the loop */
local ncoef = rowsof(Pesi)
local nc = colnumb(Pesi,"ateco2d")

local nvar : word count `array1'
```

```
/*top of external loop for all variables */
forvalues k=1/`nvar' {
        local var1 : word `k' of `array1'
        local var2 : word `k' of `array2'
        local nc1 = colnumb(Pesi,"`var2'")

        /* for debugging
        display "The variable is `var1'"
        display "The weight is `var2'"
        display "The column number of parameters is `nc1'"
        */

        local i=1
        /*top of inner loop for all observations */

        forvalues m =1/`ncoef' {
                qui replace `var1'=Base_irap_lorda*Pesi[`m',`nc1'] if ateco2==Pesi[`m',`nc'] &
ric_tot> $cla1 & ric_tot<= $cla2
                }
        /*end of inner loop */
        }
/*end of external loop*/
end

******************************** END OF DEDUC ******************


/* executing program civtofis with (6) files for (6) income classes */
display "Computing the TAX BASE components for income class 1....."
civtofis 1 fiscal`year' `year'
display "Computing the TAX BASE components for income class 2....."
civtofis 2 fiscal`year' `year'
display "Computing the TAX BASE components for income class 3....."
civtofis 3 fiscal`year' `year'
display "Computing the TAX BASE components for income class 4....."
civtofis 4 fiscal`year' `year'
display "Computing the TAX BASE components for income class 5....."
civtofis 5 fiscal`year' `year'
display "Computing the TAX BASE components for income class 6....."
civtofis 6 fiscal`year' `year'

/* label assignment */
do label_taxvar.do



************************
* LABEL_TAXVAR.DO        *
************************

/* program to assign labels to variables created in taxvar.do */

label variable Ded_inail_base "EXOGENOUS INAIL deductions from IRAP tax base"
label variable Ded_app_base "EXOGENOUS apprentices deductions from IRAP tax base"
label variable Ded_fl_base "EXOGENOUS trainees deductions from IRAP tax base"


/************
 * IRAP.DO  *
 ************
program to compute IRAP tax base and  tax revenue */

program drop _all

args year

************************************
program mkmatal
*program to store IRAP regional tax rates in a matrix
        /* reading parameters from file */
        use parameters/ali_irap`1', clear
        format %3.0g ateco2d
        set dp comma
        /* eventual policy changes of rates go here*/
```

```
        /* setting the parameters in the matrix */
        mkmat ateco2d ali*, matrix(`2')
        *matrix list `2', format(%6.4f)
        *pause
        drop _all
end
*************************************

/* runnning the program */
mkmatal `year' Ali



/* opening the dataset */
use fiscal`year', clear

/* computing the tax base deductions */

/* fixing work deductions for apprentices (not used anymore) and for trainees */
local i=1
while `i'<= _N {
        if pmi[`i']==1{
                quietly replace Ded_app_base=0 if occ_aptot_pmi[`i']==0 in `i'
                quietly replace Ded_fl_base=0 if occ_fltot_pmi[`i']==0 in `i'
                local i = `i'+1
                }
        else{
                quietly replace Ded_app_base=0 if occ_aptot_sci[`i']==0 in `i'
                local i = `i'+1
                }
        }

/* exogenous total deductions (labour cost components) (**COMMENT**)
gen Ded_lav_b_tot = Ded_inail_base + Ded_app_base + Ded_fl_base */

/* computing endogenous total deductions (labour cost components)*/
gen Ded_lav_tot = inail_tot_irap + app_tot_irap + Ded_fl_base

/* additional buffer variable for other deductions*/
gen Ded_irap_altre=0


***********************************************

*********** generating IRAP tax base with CIVILISTIC variables (small letters)

***********************************************
/* generating  variables as in Section IV Quadro IQ */
gen comp_pos_tot = ric_tot + ric_vr_tot + ric_vl_co + ric_ini_li + ric_alpro
gen comp_neg_tot = acq_beni_tot + acq_serv_tot + acq_gdbt_tot + acq_amm_imm + acq_amm_imi + /*
        */acq_vr_tot + acq_accant_tot + acq_ondiv_tot

gen base_irap_lorda = comp_pos_tot - comp_neg_tot
gen Base_irap_lorda = 0
/* applying weights for legal status DIFFERENT from 2 (corporations)
0 if sole entrepreneurship
1 if partnership
3 if cooperatives
4 if others */

replace Base_irap_lorda = base_irap_lorda * 0.922 if fgcat==0
replace Base_irap_lorda = base_irap_lorda * 0.885 if fgcat==1
replace Base_irap_lorda = base_irap_lorda * 0.957 if fgcat==3
replace Base_irap_lorda = base_irap_lorda * 0.708 if fgcat==4


***********************************************

*********** generating IRAP tax base with FISCAL variables (capital letters)

***********************************************

/* generating fiscal variables as in Section IV Quadro IQ of IRAP form */
gen Comp_pos_tot = Ric_tot + Ric_vr_tot + Ric_vl_co + Ric_ini_li + Ric_alpro
gen Comp_neg_tot = Acq_beni_tot + Acq_serv_tot + Acq_gdbt_tot + Acq_amm_imm + Acq_amm_imi + /*
        */Acq_vr_tot + Acq_accant_tot + Acq_ondiv_tot
```

```
/* computing the gross tax base*/
replace Base_irap_lorda = Comp_pos_tot - Comp_neg_tot if fgcat==2

/* computing two alternative net tax bases (fiscal and civilistic) */
* con deduzioni lavoro esogene
*gen Base_irap_netta = Base_irap_lorda - Ded_lav_b_tot - Ded_irap_altre
*gen base_irap_netta = base_irap_lorda - Ded_lav_b_tot -  Ded_irap_altre

*con deduzioni lavoro endogene
gen Base_irap_netta = Base_irap_lorda - Ded_lav_tot - Ded_irap_altre
gen base_irap_netta = base_irap_lorda - Ded_lav_tot -  Ded_irap_altre


/* defining the tax rate: unique for all regions in 1998 */
local Aliq_irap=0.0425
display "Aliq_irap = `Aliq_irap'"

        /* ncoef is the number of rows in matrix Ali
         nc is the column number of ateco categories*/

        local nsec = rowsof(Ali)
        local nc = colnumb(Ali,"ateco2d")

                /* for debugging
                display "The number of rows is `nsec'"
                display "The sector is column `nc'"
                */

        gen Irap = 0
        forvalues m=1/`nsec' {
                              quietly replace Irap = Base_irap_netta *Ali[`m',2]/*
                              */ if ateco2==Ali[`m',`nc']
                              }



/* computing the IRAP yield (fiscal base and civilistic base)
gen Irap = `Aliq_irap' * Base_irap_netta
*/

gen Irapciv = `Aliq_irap' * base_irap_netta

gen basecatl=0 if Base_irap_lorda==0
replace basecatl=1 if Base_irap_lorda<0
replace basecatl=2 if Base_irap_lorda>0

gen basecatn=0 if Base_irap_netta==0
replace basecatn=1 if Base_irap_netta<0
replace basecatn=2 if Base_irap_netta>0

/* dropping fiscal adjusted variables */
drop Ric_tot Ric_vr_tot Ric_vl_co Ric_ini_li Ric_alpro Acq_beni_tot Acq_serv_tot Acq_gdbt_tot
Acq_amm_imm
drop Acq_amm_imi Acq_vr_tot Acq_accant_tot Acq_ondiv_tot

format %15.0fc *tot Base* base* Irap*
set dp comma

/* label assignment */
do label_irap.do

label data "Results and data for SC and IRAP, 19`year'"
save irap`year'_out, replace


************************
* LABEL_IRAP.DO        *
************************

/* program to assign labels to variables created in irap.do */

label variable Ded_lav_tot "Total deductions from IRAP tax base (labour cost components)"
*label variable Ded_lav_b_tot "EXOGENOUS total deductions from IRAP tax base (labour cost
components)"
label variable Ded_irap_altre "Additional deductions from IRAP tax base"
```

```
label variable Comp_pos_tot "Total fiscal positive components of IRAP tax base"
label variable Comp_neg_tot "Total fiscal negative components of IRAP tax base"
label variable comp_pos_tot "Total civilistic positive components of IRAP tax base"
label variable comp_neg_tot "Total civilistic negative components of IRAP tax base"
label variable Base_irap_lorda "IRAP Fiscal Gross Tax Base"
label variable Base_irap_netta "IRAP Fiscal Net Tax Base"
label variable base_irap_lorda "IRAP civilistic Gross Tax Base"
label variable base_irap_netta "IRAP civilistic Net Tax Base"
label variable Irap "IRAP Total Revenue"
label variable Irapciv "IRAP Total Revenue (civilistic tax base)"

label define bcat 0 "null tax base" 1 "negative tax base" 2 "positive tax base"
label values basecatl bcat
label variable basecatl "Gross Irap Tax Base"

label define bcatn 0 "null tax base" 1 "negative tax base" 2 "positive tax base"
label values basecatn bcatn
label variable basecatn "Net Irap Tax Base"

/***********************
*     STAT_IRAP      *
***********************
program to compute final statistics of IRAP
*/

args year

capture log close
use irap`year'_out, clear
log using stat_irap`year'.log, replace
set linesize 255
pause on
set more off
capture program drop _all

program define stat

/* computing rates for incidence analysis without weights*/
gen impred_ric_rat = (imp_reddito/ric_tot)*100
gen impred_va_rat = (imp_reddito/valagg)*100

gen irap_ric_rat = (Irap/ric_tot)*100 if Irap>0
gen irap_va_rat = (Irap/valagg)*100 if Irap>0
gen irap_ut_rat = (Irap/utile_lordo)*100 if Irap>0
gen irap_int_rat = (Irap/int_pas)*100 if Irap>0
gen irap_pers_rat = (Irap/acq_pers_tot)*100 if Irap>0

/* irap as a component of total taxes */
gen irap_alimp_rat = Irap/imp_reddito if Irap>0



/* applying weights and converting ***SELECTED VARIABLES*** to thousands of euros */

local listvar "imp_irap Irap Irapciv imp_reddito ric_tot valagg utile_lordo acq_pers_tot int_pas"

foreach x of local listvar {
       gen P`x' = (`x' * peso)/1000
       }


/* label assignment */
do label_stat_irap.do


/* table with Irap revenue by PMI */
tabstat  PIrap peso if Irap>0, by(pmi) stats(n sum)col(stat)
return list

*matrix list r(StatTot)
*matrix b=r(StatTot)
*matrix list b
*pause

/* generating TOTAL IRAP REVENUE
local sum_PIrap = b[2,1]
```

```
display "Sum of Total Irap (estimated):  `sum_PIrap'"
*/

local listindic "fgcat Exp_01 Inv_01 Subfa_01 Subfp_01 areag"


/* tables with Irap revenue by different indicators */
foreach x of local listindic {
         tabstat PIrap peso if Irap>0, by(`x') stats(n sum)col(stat) format(%9.0f) labelwidth(30)
varwidth(15)
        }
tabstat PIrap peso if Irap>0, by(ateco2) stats(n sum)col(stat) format(%9.0f) labelwidth(30)
varwidth(15)

/* tables for selected variables by several indicators */
foreach x of local listindic {
        tabstat Pimp_reddito Pric_tot Pvalagg Pacq_pers_tot Pint_pas Putile_lordo, by(`x') stats(n
sum)col(stat) format(%15.0f) labelwidth(30) varwidth(15)
        tabstat Putile_lordo if Putile_lordo>0, by(`x') stats(n sum)col(stat) format(%15.0f)
labelwidth(30) varwidth(15)
        }



/* da rivedere questione weights campionari per calcolo aliquote medie */

local listrates "irap_ric_rat irap_va_rat irap_ut_rat irap_int_rat irap_pers_rat"

/* computing AVERAGE rates by categories */
foreach x of local listrates {
        tabstat `x', by(fgcat) stats(n mean sd) col(stat) format(%9.1f) varwidth(15)
        tabstat `x', by(ateco2) stats(n mean sd) col(stat) format(%9.1f) varwidth(15)
        tabstat `x', by(Exp_01) stats(n mean sd) col(stat) format(%9.1f) varwidth(15)
        tabstat `x', by(regcont) stats(n mean sd) col(stat) format(%9.1f) varwidth(15)
        tabstat `x', by(Inv_01) stats(n mean sd) col(stat) format(%9.1f) varwidth(15)
        tabstat `x', by(Subfp_01) stats(n mean sd) col(stat) format(%9.1f) varwidth(15)
        tabstat `x', by(areag) stats(n mean sd) col(stat) format(%9.1f) varwidth(15)
        tabstat `x', by(cla9) stats(n mean sd) col(stat) format(%9.1f) varwidth(15)
        tabstat `x', by(cla3) stats(n mean sd) col(stat) format(%9.1f) varwidth(15)
        }


/* gross tax base analysis*/
tabulate basecatl
tabstat peso, statistics( sum ) by(basecatl) format(%15.0f) columns(variables)

/* net tax base analysis*/
tabulate basecatn
tabstat peso, statistics( sum ) by(basecatn) format(%15.0f) columns(variables)

log close

end


stat

save final`year', replace

********saving the file used as input by IRPEG module *****************
save irap_out, replace



************************
* LABEL_STAT_IRAP.DO        *
************************

/* program to assign labels to variables created in stat_irap.do */

label variable impred_ric_rat "Income Taxes/Enterprise Total Revenue"
label variable impred_va_rat "Income Taxes/Gross Value Added"
label variable irap_ric_rat "IRAP revenue/Enterprise Total Revenue"
label variable irap_va_rat "IRAP revenue/Gross Value Added"
label variable irap_ut_rat "IRAP revenue/Gross Profit (loss)"
label variable irap_int_rat "IRAP revenue/Interest Payable"
```

```
label variable irap_pers_rat "IRAP revenue/Total Personnel Costs"
label variable irap_alimp_rat "IRAP Revenue/Total Income Taxes"


label variable Pimp_irap "IRAP (PMI) (Weighted)"
label variable PIrap "IRAP Total Revenue (Weighted)"
label variable PIrapciv "IRAP Total Revenue (civilistic tax base) (Weighted)"
label variable Pimp_reddito "Income taxes (Weighted)"
label variable Pric_tot "Income from sales and Services (Weighted)"
label variable Pvalagg "Value Added (Weighted)"
label variable Putile_lordo "Gross Profit (Loss) (Weighted)"
label variable Pacq_pers_tot "Personnel Expenses (Total) (Weighted)"
label variable Pint_pas "Interest payable (Weighted)"

/*****************************************************************
 PROGRAM SIMULATING FISCAL ADJUSTMENTS OF BALANCE SHEETS VARIABLES
*****************************************************************/
clear
version 7
set more off

/* setting parameters in matrix par_fisc_adj */

use parameters/par_fisc_adj.dta, clear

*set matsize 19

mkmat aliq_amm_fab aliq_amm_fabns aliq_amm_im aliq_amm_mt aliq_amm_aic     aliq_amm_ot_ttd
aliq_amm_ot_maa /*
        */ aliq_amm_ot_maa aliq_amm_ot_mtns  pers_par  par_amm_if   par_amm_brev  par_amm_avv
        par_amm_r_s /*
        */ sval_cr_par  sval_cr_par0  cr_par  var_opult_par  spe_man_par, matrix(par_fisc_adj)

*matrix list par_fisc_adj

/*
use dati98_estr.dta
mvencode _all, mv(0)override
save dati98.dta, replace
*/

use irap_out.dta
keep if (tipo_fg>1 )

*********************************************
*LIBERAL TRANSFERS TO EMPLOYEES
*********************************************

/*The rule: deduction up to a threshold of 0,5 percent of total labour cost (employees).*/


* parameter used in the rule
local pers_par = colnumb(par_fisc_adj,"pers_par")


gen acq_prpers_ded = acq_pers_tot*par_fisc_adj[1,`pers_par' ]
replace  acq_prpers_ded =acq_prpers if acq_prpers < acq_prpers_ded


*OUTPUT

gen acq_pers_tot_sim = acq_pers_tot + (acq_prpers_ded - acq_prpers)
replace acq_pers_tot_sim = 0 if acq_pers_tot_sim ==.

drop acq_prpers_ded


**********************************
* CREDITS DEVALUATION
**********************************

/*Devaluation of credits from exchange of goods and services are deductible up to 0,5%
  of credits nominal value */

local sval_cre_par = colnumb(par_fisc_adj,"sval_cre_par")
```

```
gen dev_cr_sim= par_fisc_adj[1,`sval_cre_par'] *((sp_cretot_sci+sp_liq_ass)+acq_svcr_sci) /* maximum
amount that can be deductible*/

replace dev_cr_sim = acq_svcr_sci if dev_cr_sim>acq_svcr_sci
replace dev_cr_sim = 0 if dev_cr_sim ==.


************************************************
* IMMATERIAL GOODS DEPRECIATION
************************************************


****************************************************
* Copyrights, Intellectual Property and Software
****************************************************

/*deductible amount is obtained computing the depreciation share on original value. In order to
gaing the original value
  it is necessary to sum up depreciation fund plus the value stored in the Balance Sheet.*/

/*without new acquisitions*/

gen cis_sh_1=((sp_im_cop - oth_new_in_c)+(sp_im_ip - im_art_tot)+(sp_im_sfw - im_sof_tot))/(sp_im_im
- oth_new_in)
replace cis_sh_1=0 if  cis_sh_1==.

/*with new acquisitions*/

gen cis_sh_2=(sp_im_cop + sp_im_ip + sp_im_sfw)/sp_im_im
replace cis_sh_2=0 if  cis_sh_2==.

replace cis_sh_1=cis_sh_2 if cis_sh_1<0

/*Share of total depreciation fund related to Copyrights, Intellectual Property and Softwares*/

gen fd_cis_sh=cis_sh_1*sp_fd_im

/*Share of Copyrights, Intellectual Property and Softwares depreciation on total immaterial goods
depreciations */

gen dep_sh_cis =(sp_im_cop + sp_im_ip+ sp_im_sfw)/sp_im_im
replace  dep_sh_cis =0 if  dep_sh_cis ==.

/*Value of fixed assets related to Copyrights, Intellectual Property and Softwares that must be
added in order to obtain
 the original cost*/

gen quot_dep_cis =dep_sh_cis*acq_amm_imi

/*Original cost for Copyrights, Intellectual Property and Softwares */

gen cis_hc = fd_cis_sh + (sp_im_cop + sp_im_ip+ sp_im_sfw) + quot_dep_cis

/*Maximum amount that can be deducted*/

        gen  amm_cis = cis_hc/3
        replace amm_cis =0 if amm_cis ==.

*************************
* Trademarks & Licenses
*************************

/*Share of Trademarks & Licenses depreciation on total immaterial goods depreciations */
gen tl_sh =(sp_im_tdm - oth_new_in_tdm)/(sp_im_im - oth_new_in)
replace tl_sh =0 if tl_sh ==.

/*Share of total depreciation fund related to Trademarks & Licenses */
gen fd_tl_sh =tl_sh*sp_fd_im

/*Share of Trademarks & Licenses depreciation on total depreciation */
gen amm_sh_tl =(sp_im_tdm)/sp_im_im
replace amm_sh_tl =0 if amm_sh_tl ==.

/*Value of fixed assets related to Trademarks & Licenses that must be added in order to obtain
 the original cost*/
gen quot_amm_tl =amm_sh_tl*acq_amm_imi
```

```
/* The original cost of Trademarks & Licenses */
gen tl_hc =fd_tl_sh + (sp_im_tdm) + quot_amm_tl

/*Maximum amount that can be deducted*/
       gen  amm_tl =tl_hc/10
       replace amm_tl =0 if amm_tl==.

**************
*GOODWILL
**************

       gen amm_gdw =sp_im_av/10
       replace amm_gdw =0 if amm_gdw ==.


***********************************
*RESEARCH & DEVELOPMENT
***********************************

/*The third part of research and development costs can be deducted */

/*The share of research and development on total fixed assets */
       gen incd_rsv =acq_risv/(acq_risv+acq_pubbli)
       gen cs_rsv =(incd_rsv*sp_im_rsp)-capz_rsv
       replace cs_rsv =0 if cs_rsv<0
       gen amm_risv_ded = cs_rsv/3
       replace amm_risv_ded =0 if amm_risv_ded ==.



*****************************************
*TOTAL IMMATERIAL FIXED ASSETS DEPRECIATIONS
*****************************************

       gen amm_im_in_tot = amm_cis + amm_tl + amm_gdw + amm_risv_ded



*OUTPUT

/*We choose to consider the highest between estimated and accounted depreciation value (if the
estimated value is
 lower than the accounted one then it is possible to impute this latter one when material goods
overusing is documented) */

       gen amm_im_in = amm_im_in_tot
       replace amm_im_in =acq_amm_imi if amm_im_in_tot>acq_amm_imi
       replace amm_im_in = 0 if amm_im_in ==.




***********************************************
* MATERIAL GOODS DEPRECIATION
***********************************************

***********************************
* ORDINARY DEPRECIATION
***********************************
/*we estimate an average depreciation rate (for each material good there is a different depreciation
coefficient)
which is applied to total material goods in order to obtain an estimated depreciation value.*/

gen quot_amm_fab = sp_im_fab/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)
replace quot_amm_fab = 0 if quot_amm_fab ==.

gen cs_fab_amm = quot_amm_fab*sp_fd_im_tan + sp_im_fab

gen quot_amm_fabns = (0.5*sp_im_fabns)/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)
replace quot_amm_fabns = 0 if quot_amm_fabns ==.
gen cs_fabns_amm = quot_amm_fabns*sp_fd_im_tan + 0.5*sp_im_fabns

gen sp_im_mac_sim = sp_im_mac - sp_im_mt
gen quot_amm_im = sp_im_mac_sim/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)
replace quot_amm_im = 0 if quot_amm_im ==.
```

```
gen cs_im_amm =  quot_amm_im*sp_fd_im_tan + sp_im_mac_sim
replace cs_im_amm = 0 if cs_im_amm ==.

gen quot_amm_mt = sp_im_mt/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)
gen cs_mt_amm = quot_amm_mt*sp_fd_im_tan + sp_im_mt

gen quot_amm_aic = sp_im_aic/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)
gen cs_aic_amm = quot_amm_aic*sp_fd_im_tan + sp_im_aic

gen quot_amm_ot_ttd = sp_im_ttd/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)
gen cs_ot_ttd_amm = quot_amm_ot_ttd*sp_fd_im_tan + sp_im_ttd

gen quot_amm_ot_maa = sp_im_maa/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)
gen cs_ot_maa_amm = quot_amm_ot_maa*sp_fd_im_tan + sp_im_maa

gen quot_amm_ot_mtns = (0.5*sp_im_mtns)/(sp_im_man-sp_im_ter-sp_im_bv-sp_im_inc)
gen cs_ot_mtns_amm = quot_amm_ot_mtns*sp_fd_im_tan + 0.5*sp_im_mtns



/* It is possible to distinguish two components of ordinary depreciation: one part related to old
acquisitions and another
part related to new acquisitions.*/

* parameters

local aliq_amm_fab = colnumb(par_fisc_adj,"aliq_amm_fab")
local aliq_amm_fabns = colnumb(par_fisc_adj,"aliq_amm_fabns")
local aliq_amm_im = colnumb(par_fisc_adj,"aliq_amm_im")
local aliq_amm_mt = colnumb(par_fisc_adj,"aliq_amm_mt")
local aliq_amm_aic = colnumb(par_fisc_adj,"aliq_amm_aic")
local aliq_amm_ot_ttd = colnumb(par_fisc_adj,"aliq_amm_ot_ttd")
local aliq_amm_ot_maa = colnumb(par_fisc_adj,"aliq_amm_ot_maa")
local aliq_amm_ot_mtns = colnumb(par_fisc_adj,"aliq_amm_ot_mtns")

gen amm_im_tan_old = (cs_fab_amm-im_cost_tot)*(par_fisc_adj[1,`aliq_amm_fab'])+ /*
*/ (cs_fabns_amm-im_acq_fns/2)*(par_fisc_adj[1,`aliq_amm_fabns'] )+/*
*/ (cs_im_amm-im_mac_tot-im_mtr_new-im_mtr_us) * (par_fisc_adj[1,`aliq_amm_im']) + /*
*/ (cs_mt_amm-im_mtr_new-im_mtr_us) * (par_fisc_adj[1,`aliq_amm_mt']) +/*
*/ (cs_aic_amm-im_at_ic_tot) * (par_fisc_adj[1,`aliq_amm_aic']) + /*
*/ (cs_ot_ttd_amm-im_att_new-im_att_us) * (par_fisc_adj[1,`aliq_amm_ot_ttd'])+/*
*/ (cs_ot_maa_amm-im_mob_new-im_mob_us) * (par_fisc_adj[1,`aliq_amm_ot_maa']) + /*
*/ (cs_ot_mtns_amm-im_tras_new/2-im_tras_us/2) * (par_fisc_adj[1,`aliq_amm_ot_mtns'])

        replace amm_im_tan_old =0 if amm_im_tan_old ==.



        gen amm_im_tan_acq = (im_cost_tot)*(par_fisc_adj[1,`aliq_amm_fab'])/2 +/*
        */ (im_acq_fns/2) * (par_fisc_adj[1,`aliq_amm_fabns'])/2 +/*
        */ (im_mac_tot+im_mtr_new+im_mtr_us)*(par_fisc_adj[1,`aliq_amm_im'])/2 +/*
        */ (im_mtr_new+im_mtr_us)*(par_fisc_adj[1,`aliq_amm_mt'])/2 +/*
        */ (im_at_ic_tot) * (par_fisc_adj[1,`aliq_amm_aic'])/2 +/*
        */ (im_att_new+im_att_us) * (par_fisc_adj[1,`aliq_amm_ot_ttd'])/2 +/*
        */ (im_mob_new+im_mob_us) * (par_fisc_adj[1,`aliq_amm_ot_maa'])/2 +/*
        */ (im_tras_new/2+im_tras_us/2) * (par_fisc_adj[1,`aliq_amm_ot_mtns'])/2

        replace amm_im_tan_acq =0 if amm_im_tan_acq ==.


**********************************
* ACCELERATED DEPRECIATION
**********************************

/*In addition to this ordinary depreciation, a double depreciation rate to new acquisitions
(accelerated depreciation)
 is allowed.*/

        gen acc_amm = (im_cost_tot) * (par_fisc_adj[1,`aliq_amm_fab']/2) +/*
                                        */ (im_acq_fns/2) *
(par_fisc_adj[1,`aliq_amm_fabns']/2) +/*
                                        */ (im_mac_tot+im_mtr_new+im_mtr_us) *
(par_fisc_adj[1,`aliq_amm_im']/2) +/*
                                        */
(im_mtr_new+im_mtr_us)*(par_fisc_adj[1,`aliq_amm_mt']/2) +/*
```

```
                                                */ (im_at_ic_tot)*(par_fisc_adj[1,`aliq_amm_aic']/2)
+/*
                                                */
(im_att_new+im_att_us)*(par_fisc_adj[1,`aliq_amm_ot_ttd']/2) +/*
                                                */ (im_mob_new+im_mob_us)*
(par_fisc_adj[1,`aliq_amm_ot_maa']/2) +/*
                                                */ (im_tras_new/2+im_tras_us/2) *
(par_fisc_adj[1,`aliq_amm_ot_mtns']/2)

        replace acc_amm =0 if acc_amm ==.


*********************************************************
* FINAL DEPRECIATION
*********************************************************


        gen amm_im_tan = amm_im_tan_old + amm_im_tan_acq + acc_amm
        replace amm_im_tan =0 if amm_im_tan ==.


****************************************
* VARIATIONS OF WORK IN PROGRESS
****************************************

/* It is allowed to reduce the account variable up to 2% for barganing risks*/

replace ric_vl_co=0 if ric_vl_co==.

local var_opult_par = colnumb(par_fisc_adj,"var_opult_par")

gen var_op_ult=ric_vl_co * par_fisc_adj[1,`var_opult_par']

order ric_vl_co var_op_ult

**************************
*MAINTENANCE COSTS
**************************

/* maintenance costs can be imputed for the part of 5% of the total material goods costs (the value
coming
from the beginning of the accounting period) excluding costs that increase the value of
corresponding goods.
In addition, the amount exceeding this threshold can be deducted in the next 5 accounting periods in
fixed quotas. */

gen ca_fab_amm = sp_im_fab + quot_amm_fab*acq_amm_imm

gen ca_fabns_amm = 0.5*sp_im_fabns + quot_amm_fabns*acq_amm_imm

gen ca_im_amm = sp_im_mac + quot_amm_im*acq_amm_imm

gen ca_mt_amm = sp_im_mt + quot_amm_mt*acq_amm_imm

gen ca_aic_amm = sp_im_aic + quot_amm_aic*acq_amm_imm

gen ca_ot_ttd_amm = sp_im_ttd + quot_amm_ot_ttd*acq_amm_imm

gen ca_ot_maa_amm = sp_im_maa + quot_amm_ot_maa*acq_amm_imm

gen ca_ot_mtns_amm = 0.5*sp_im_mtns + quot_amm_ot_mtns*acq_amm_imm


gen ca_tot_amm = ca_fab_amm + ca_fabns_amm + ca_im_amm + ca_mt_amm + ca_aic_amm + ca_ot_ttd_amm +
ca_ot_maa_amm + ca_ot_mtns_amm

/*Maximum amount that can be deducted*/

gen ded_manu_max = 0.05*ca_tot_amm
replace ded_manu_max = 0 if ded_manu_max ==.


gen ded_manu_ord =manord
replace ded_manu_ord= ded_manu_max if manord > ded_manu_max
```

```
/*The exceeding part can be deducted in the next 5 accounting periods in fixed quotas (We consider
 an everage perod of 3 years).*/

gen ecced= (manord  - ded_manu_max)  if manord > ded_manu_max
replace ecced = 0 if ecced ==.
gen ded_ecced = ecced/3
replace ded_ecced = 0 if ded_ecced ==.

*output
gen manu_ord = ded_manu_ord + ded_ecced
gen acq_allav_sim= acq_allav - manord + manu_ord

************************************************
* COSTS RELATED TO MORE THAN ONE ACCOUNTING PERIOD
************************************************


**********************************
*RESEARCH & DEVELOPMENT
**********************************

        gen ca_rsv=acq_risv+capz_rsv
        gen acq_risv_ded = ca_rsv/3
        replace acq_risv_ded =0 if acq_risv_ded ==.


* OUTPUT

gen acq_serv_tot_sim = acq_serv_tot + ( acq_risv_ded - acq_risv) + (acq_allav_sim - acq_allav)
replace acq_serv_tot_sim  = 0 if acq_serv_tot_sim ==.


*****************
* OUTPUT
*****************


/* saving variables used to define corporate profits for tax purposes */


keep  codice l_ateco   tipo_fg     sci   ric_tot     ric_vr_tot   var_op_ult   ric_ini_li
ric_vl_co ric_alpro   acq_beni_tot acq_serv_tot_sim /*
*/ acq_serv_tot   acq_gdbt_tot  acq_pers_tot_sim  acq_pers_tot  acq_amm_imm  amm_im_in  acq_amm_imi
amm_im_tan  /*
*/ acq_svimm_sci  dev_cr  acq_vr_tot   acq_accant   acq_alacc_sci   acq_ondiv_tot    divid   int_att
int_pas    /*
*/ rival   sval   pr_onstr_pr     pr_onstr_on   acq_svcr_sci    dev_cr_sim   ut_es   utile_lor_sci
base_irap_lorda

do label_fiscal_adj.do

format %12.0f var_op_ult acq_serv_tot acq_pers_tot amm_* dev*

label data "Variables defining corporate profits for tax purposes"
save out_adj_prof.dta, replace


/*********************************************************
 LABELS OF VARIABLES GENERATED BY PROGRAM FISCAL_ADJ.DO
*********************************************************/


lab var var_op_ult "simulated fiscal variations of ultra-annual works"
lab var acq_serv_tot_sim "simulated fiscal costs for services"
lab var acq_pers_tot_sim "simulated fiscal personnel expenses"
lab var dev_cr_sim "simulated fiscal credits devaluation"
lab var amm_im_in "simulated fiscal immaterial goods depreciation"
lab var amm_im_tan "simulated fiscal tangible goods depreciation"

************************************
* DO-FILE COMPUTING CORPORATE INCOME
************************************
clear
version 7
set more off
```

```
capture program drop adj_inc
capture program drop classe

use out_adj_prof.dta

/* defining classe_base_irap which refers to the enterprise income (base_irap_lorda) class
   (the variable will be used in other files and programs)*/

gen classe_base_irap = 1 if base_irap_lorda<=0

program define classe
replace classe_base_irap = `1' if (base_irap_lorda>`2' & base_irap_lorda<=`3' /*
*/ & classe_base_irap==.)
end

replace classe_base_irap = 14 if (base_irap_lorda > 258228450 & classe_base_irap==.)

classe 2 0 10329
classe 3 10329 25823
classe 4 25823 51646
classe 5 51646 103291
classe 6 103291 185924
classe 7 185924 516457
classe 8 516457 1032914
classe 9 1032914 2582285
classe 10 2582285 5164569
classe 11 5164569 25822845
classe 12 25822845 51645690
classe 13 51645690 258228450

/* saving output file out_adj_prof containing variable classe_base_irap used in other modules*/

save corp_inc.dta, replace


/* running the program to compute total simulated fiscal adjustments; the program is useful
   for validation of estimated fiscal adjustments and it can be excluded from running
   when policy reforms do not concern fiscal adjustements of balance sheet profit/loss */

do sim_adj_tot.do


/*********************************************************************
 PROGRAM COMPUTING NON SIMULATED ADJUSTMENTS OF PROFITS/LOSS
 USING PARAMETRS FROM AGGREGATE TAX RETURNS DATA
*********************************************************************

Parameters are stored in matrix corr_utile and defined according to 14 "income" - specifically
Irap_base_lorda - classes and for enterprises with (positive) profits and losses. Parameters
result from aggregate corporation tax returns data in year 1998 and relate to positive
adjustment/negative adjustments of gross corporate profits (loss), charities and gifts, for
enterprises of the industry and service sector (excluding banks and insurance companies) */

use parameters/corr_prof.dta, clear

*set matsize 84

/* corr_prof stores parameters to be used */

mkmat  va_ut  vd_ut  er_lib_ut  va_per  vd_per  er_lib_per, matrix(corr_prof)

*matrix list corr_prof

use corp_inc.dta


/* corporate profits resulting from the algebraic sum of unadjusted (non simulated)
   positive/negative components of the balance sheet and positive/negative components
   adjusted (simulated) for tax purposes */


gen ut_lor_sim =( ric_tot + ric_vr_tot + var_op_ult+ ric_ini_li+ ric_alpro) /*
*/ - (acq_beni_tot+acq_serv_tot_sim+ acq_gdbt_tot + acq_pers_tot_sim + amm_im_in + amm_im_tan + /*
*/ acq_svimm_sci + dev_cr_sim + acq_vr_tot + acq_accant + acq_alacc_sci + acq_ondiv_tot) + /*
*/ (divid + int_att -int_pas) + (rival - sval) + (pr_onstr_pr - pr_onstr_on)
```

```
/* per le imprese in perdita uso gli stessi parametri di correzione delle imprese in utile */

gen corp_inc=ut_lor_sim

program define adj_inc
quietly replace corp_inc= ut_lor_sim*(1 + corr_prof[`1',1]-corr_prof[`1',2]-corr_prof[`1',3])/*
        */ if (classe_base_irap ==`1' & utile_lor_sci>0 & ut_lor_sim>0)
quietly replace corp_inc= ut_lor_sim*(1 - corr_prof[`1',1]+corr_prof[`1',2]+corr_prof[`1',3])/*
        */ if (classe_base_irap ==`1' & utile_lor_sci>0 & ut_lor_sim<0)
display "imputed fiscal adjustments of profits for enterprises of income class `1' "
quietly replace corp_inc= ut_lor_sim*(1 + corr_prof[`1',1]- corr_prof[`1',2]- corr_prof[`1',3]) /*
        */ if (classe_base_irap==`1' & utile_lor_sci<0 & ut_lor_sim>0)
quietly replace corp_inc= ut_lor_sim*(1 - corr_prof[`1',1]+ corr_prof[`1',2]+ corr_prof[`1',3]) /*
        */ if (classe_base_irap==`1' & utile_lor_sci<0 & ut_lor_sim<0)
display "imputed fiscal adjustments of loss for enterprises of income class `1' "
end */

/* running the program for all income classes (rows of matrix corr_utile) */

forvalues i = 1(1) 14 {
               adj_inc `i'
}

/* saving output */

keep codice l_ateco tipo_fg utile_lor_sci corp_inc classe_base_irap

do label_corp_income.do

format corp_inc %12.0f

label data "Corporate income"
save out_corp_inc.dta, replace

erase corp_inc.dta


/*******************************************************
 LABELS OF VARIABLES GENERATED BY PROGRAM CORP_INCOME.DO
*******************************************************/

lab var corp_inc "corporate income"



/***************************
CORPORATE TAX (IRPEG)
***************************/

/* do-file estimating the gross corporate tax and the net corporate tax */

clear
version 7
set more off

/* using matrix par_simul_instr where tax legislation parameters are stored */

use parameters/par_simul_instr.dta

*set matsize 38

mkmat irpeg_ord  irpeg_dit  dit_par1  dit_par2  irpeg_tar   irpeg_coop_1  irpeg_coop_2_ex  /*
      */ irpeg_coop_2_half    coop_par1   coop_par2 t_cg_par prov_par ci_ii_par1  ci_ii_par2 /*
      */ ci_ii_par3  ci_ii_par4  ci_ii_par5 ci_ii_aliq1   ci_ii_aliq2      ci_ii_par6   ci_ii_par7
/*
      */ ci_ric_par1  ci_ric_aliq  ci_pmi_par1 ci_pmi_par2  ci_pi_par1  ci_pi_par2  ci_td_agr /*
      */ ci_td_ind  ci_td_costr  ci_td_comm  ci_td_serv_pr ci_td_pa   ci_occ_par1  ci_occ_par2 /*
       */ ci_occ_sgl   ci_ct_aliq ci_ct_sgl, matrix(par_simul_instr)


/* using a data-set resulting by merging out_adj_prof (adjusted corporate profits) and corp_inc
(corporate income)*/

use out_corp_inc.dta
```

```
keep codice classe_base_irap corp_inc
sort codice
save corp_inc.dta, replace

*use dati98.dta

use irap_out.dta
keep if tipo_fg>1
sort codice

merge codice using corp_inc.dta
assert _merge == 3
drop _merge

/* saving file to be used when computing the corporate tax */

save tax_base.dta, replace

/*************************************************************
 RUNNING DO-FILE SIMULATING THE DIT SYSTEM
*************************************************************/

do dit.do

save tax_base.dta, replace


/*****************************************
 DIVIDEND TAX CREDIT
*****************************************/

local irpeg_ord = colnumb(par_simul_instr,"irpeg_ord")


gen txc_div = par_simul_instr[1,`irpeg_ord']/(1-par_simul_instr[1,`irpeg_ord'])*divid

save tax_base.dta, replace



*correcting dividend tax credit amount using data from 1999 corporate tax returns

use parameters/corr_div_tcred.dta, clear
capture program drop corr_div_txc

*set matsize 28
mkmat cred_div_ut cred_div_per, matrix(corr_div_tcred)

use tax_base.dta

* program to correct dividend tax credit *

program define corr_div_txc
replace txc_div = (corr_div_tcred[`1',1]*corp_inc) if (txc_div>0 & corp_inc>0 & utile_lor_sci>0 /*
                        */ & classe_base_irap ==`1')
display "corrected dividend tax credit for firms of income class `1' with profits"
replace txc_div = (corr_div_tcred[`1',2]*corp_inc) if (txc_div>0 & corp_inc>0 & utile_lor_sci<=0 /*
                    */ & classe_base_irap ==`1')
display "corrected dividend tax credit for firms of income class `1' incurring in losses"
end

* running the program for all income classes *
forvalues i = 1(1) 14 {
               corr_div_txc `i'
}


/*********************** TAXABLE INCOME *********************************/



/* taxable income (corporate income + dividend tax credit) */


gen redd_impon = corp_inc + txc_div
```

```
save tax_base.dta, replace


/*******************************************************************************
 RUNNING DO-FILE IMPUTING LOSSES FROM PREVIOUS A.P. TO BE SUBRACTED FROM TAXABLE INCOME
*******************************************************************************/

do imp_loss_prev_ap.do


/* fiscal loss from current (ecc98_loss) year can be brought forward up to five years */

gen ecc98_loss = redd_impon if redd_impon < 0
replace ecc98_loss = 0 if ecc98_loss == .

/* setting taxable income nil if this is negative */

replace redd_impon = 0 if redd_impon<0
save tax_base.dta, replace


/****************************** GROSS TAX ***************************************/

local irpeg_dit = colnumb(par_simul_instr,"irpeg_dit")
local dit_par1 = colnumb(par_simul_instr,"dit_par1")
local dit_par2 = colnumb(par_simul_instr,"dit_par2")


/* corporate income is divided in two components, one taxed at the ordinary rate (37%), the
   other taxed at the preferencial rate (19%)*/

/* when computing allowable dit income the average corporate tax rate must be <=0,27, that is
   allowable income must be <= taxable income * 0,5556. The ordinary profits from net assets
   increase are calculated */


/* considering only companies with positive taxable income */


gen dit_ut_agv_sgl1 = redd_impon*(par_simul_instr[1,`dit_par1']) if (redd_impon>0 & dit_eleg ==1)
replace dit_ut_agv_sgl1 = 0 if dit_ut_agv_sgl1 ==.

gen dit_ut_agv_sgl2 = pn_incr*(par_simul_instr[1,`dit_par2']) if (dit_eleg ==1)
replace dit_ut_agv_sgl2 = 0 if dit_ut_agv_sgl2 ==.


/* the output variable is dit_ut_agv (allowable dit profits). The corporate tax on this
   component is computed below. When ordinary profits from net assets increase (allowable
   profits) is > (taxable income * 0,5556), the surplus can be brought forward up to 5 years*/


gen dit_ut_agv = dit_ut_agv_sgl1 if (dit_ut_agv_sgl2>dit_ut_agv_sgl1)
replace dit_ut_agv = dit_ut_agv_sgl2 if (dit_ut_agv_sgl2<=dit_ut_agv_sgl1)

/* saving results */

save tax_base.dta, replace

/*******************************************************************************
PROGRAM TO CORRECT SIMULATED ALLOWABLE DIT INCOME ON THE BASIS OF AGGREGATE CORPORATIONS
TAX RETURNS DATA
*******************************************************************************/


/* As mean incidence of simulated allowable (taxable) dit income on total taxabale income
   is higher than the effective incidence resulting from tax returns (0.02596; 0.01830),
   allowable dit income is corrected using parameters computing the incidence of allowable dit
   income on total taxable income for irap tax base classes. Parameters are stored in matrix
   corr_tbase_gtax*/


use parameters/corr_tbase_gtax.dta, clear
capture program drop corr_dit_inc
*set matsize 56
mkmat ut_rimp_pos per_rimp_pos dit_inc tax_allw, matrix(corr_tbase_gtax)
```

```
use tax_base.dta

/* correcting allowable dit income */

gen soglia_dit=0

program define corr_dit_inc
replace soglia_dit=corr_tbase_gtax[`1',3]* redd_impon if (dit_ut_agv>0 & classe_base_irap ==`1' &
redd_impon>0)
replace dit_ut_agv = (corr_tbase_gtax[`1',3]* redd_impon) if (dit_ut_agv>0 & dit_ut_agv>soglia_dit/*
                  */ & redd_impon>0 & classe_base_irap ==`1')
display "corrected allowable dit income for enterprises of income class `1'"
end

/* running the program for all income classes */
forvalues i = 1(1) 14 {
               corr_dit_inc `i'
}

drop soglia_dit


/* surplus of allowable income to be brought forward. This includes also companies with
   losses (redd_impon=0)*/

gen ecc98_dit_ut_agv = (dit_ut_agv_sgl2 - dit_ut_agv_sgl1) if (dit_ut_agv_sgl2 > dit_ut_agv_sgl1)
replace ecc98_dit_ut_agv = dit_ut_agv_sgl2 if (redd_impon == 0 & dit_eleg ==1)
replace ecc98_dit_ut_agv = 0 if (ecc98_dit_ut_agv ==.)


/* Income taxed at the ordinary rate */

gen redd_imp_ord = (redd_impon - dit_ut_agv) if (redd_impon>0)
replace redd_imp_ord = 0 if redd_imp_ord<0
replace redd_imp_ord = 0 if redd_imp_ord == .


/* Gross Corporate Tax */

gen irpeg_gross=(redd_imp_ord*par_simul_instr[1,`irpeg_ord'])+(dit_ut_agv*
par_simul_instr[1,`irpeg_dit'])/*
                */ if (redd_imp_ord>0)
replace irpeg_gross = 0 if irpeg_gross ==.


/***************************************
 ALLOWANCE FOR CO-OPERATIVES
***************************************/

* allowance for co-operatives (reduced rate)

*tempvar eleg_coop_1    eleg_coop_2_all      eleg_coop_2_half   inc_lcost

local irpeg_coop_1 = colnumb(par_simul_instr,"irpeg_coop_1")
local irpeg_coop_2_ex = colnumb(par_simul_instr,"irpeg_coop_2_ex")
local irpeg_coop_2_half = colnumb(par_simul_instr,"irpeg_coop_2_half")
local coop_par1 = colnumb(par_simul_instr,"coop_par1")
local coop_par2 = colnumb(par_simul_instr,"coop_par2")


gen eleg_coop_1 = 0
replace eleg_coop_1 = 1 if tipo_fg==3  /* co-operatives */

gen inc_lcost = acq_pers_tot/(cost_prod_tot-acq_matp)
replace inc_lcost = 0 if (inc_lcost == .)

gen eleg_coop_2_ex =0
replace eleg_coop_2_ex =1 if (tipo_fg==3 & (lett2=="O" | lett2=="OA") & inc_lcost
>(par_simul_instr[1,`coop_par1']))

gen eleg_coop_2_half = 0
replace eleg_coop_2_half = 1 if (tipo_fg==3 & (lett2=="O" | lett2=="OA") &
(inc_lcost>(par_simul_instr[1,`coop_par2'])) & /*
               */ (inc_lcost<(par_simul_instr[1,`coop_par1'])))
```

```
* computing irpeg for co-operatives 1 (27,75% of ordinary income)

replace irpeg_gross=(redd_imp_ord*par_simul_instr[1,`irpeg_coop_1'])+(dit_ut_agv*
par_simul_instr[1,`irpeg_dit']) if /*
                                        */(redd_imp_ord>0 & eleg_coop_1 ==1)

* assuming exempted co-operatives (eleg_coop_2_ex) are not eligible to the dit system

replace irpeg_gross=(redd_impon*par_simul_instr[1,`irpeg_coop_2_ex']) if (redd_imp_ord>0 &
eleg_coop_2_ex ==1)

* computing irpeg for co-operatives 2_half (18,5% on ordinary income)

replace irpeg_gross=(redd_imp_ord*par_simul_instr[1,`irpeg_coop_2_half'])+(dit_ut_agv*
par_simul_instr[1,`irpeg_dit']) if /*
                                        */(redd_imp_ord>0 & eleg_coop_2_half ==1)
save irpeg.dta, replace


/************************* NET TAX ****************************/

/* Net tax is obtained by subtracting specific tax allowances (detrazioni di imposta) from the
   gross tax for companies with positive taxable income. These allowances cannot be simulated
   and are imputed on the basis of their incidence on the gross tax resulting from aggregate
   corporations'tax returns data of year 1999 */


/***************************************
 TAX ALLOWANCES
***************************************/

/* running the program to impute non simulated tax allowances and generating the variable
   irpeg_net (irpeg_netta)for enterprises with positive taxable income */

do imp_tax_allw.do


/************************* TAX DUE ************************/

/* to obtain the tax due tax credits must be subtracted from the net tax and, subsequentely,
   tax reliefs must be subtracted from the net tax */


/************************************
 TAX RELIEFS
************************************/

/* Running the do-dile estimating reliefs that can be simulated. The
   remaing ones are imputed on the basis of their incidence on the net tax (below)*/

do tax_reliefs.do

* merging the output (simulated tax reliefs) with irpeg.dta

sort codice
save tax_reliefs.dta, replace

use irpeg.dta
sort codice
save irpeg.dta, replace
merge codice using tax_reliefs.dta
assert _merge == 3
drop _merge

save irpeg.dta, replace

/* dropping file tax_base */

erase tax_base.dta

/* running the program computing non simulated tax reliefs and generating matrix corr_net_tax
   with corrective parameters (the program can be excluded from running when policy changes
   are introduced */

do imp_tax_reliefs.do
```

```
/* imputing non simulated tax reliefs using parameters of matrix corr_net_tax */

*set matsize 14

mkmat  tax_reliefs, matrix(corr_net_tax)

/* data to be used */

use irpeg.dta

/* running program cre_imp computing corporate tax net of non simulated tax reliefs */

capture program drop tax_rel_firms

program define tax_rel_firms
quietly replace irpeg_net = irpeg_net *(1 - corr_net_tax[`1',1]) if (classe_base_irap ==`1' &
redd_impon>0)
display "imputed non simulated tax reliefs for enterprises of income class `1' "
end


/* running the program for all income classes (rows of matrix corr_imp_lorda) */

forvalues i = 1(1) 14 {
               tax_rel_firms `i'
}

save irpeg.dta, replace


/* computing tax due (net tax - tax credits - tax reliefs) */


/* subtracting dividend tax credit */

gen irpeg_due =(irpeg_net-txc_div) if (redd_impon >0)

save irpeg.dta, replace


/*****************************
 TAX CREDITS
*****************************/


/* running the program computing non simulated tax credits (mainly credit for taxes paid abroad
   and correcting the tax due for companies with positive taxable income */

/* the program can be excuded from running when policy changes are introduced */

do imp_tax_credits.do


*set matsize 14

/* matrix corr_tax_due stores parameters to be used */

mkmat cred_imp, matrix(corr_tax_due)

/* data to be used */

use irpeg.dta

/* running program cred_imp computing corporate tax net of non simulated tax credits */

capture program drop cred_imp

program define cred_imp
quietly replace irpeg_due = irpeg_due *(1 - corr_tax_due[`1',1]) if (classe_base_irap ==`1' &
redd_impon >0)
display "imputed non simulated tax credits for enterprises of income class `1' "
end


/* running the program for all income classes (rows of matrix corr_tax_due) */
```

```
forvalues i = 1(1) 14 {
               cred_imp `i'
}


save irpeg.dta, replace


/* generating the variable ecc98_irpeg_due (amount to be brought forward) if irpeg_due is negative
*/

gen ecc98_irpeg_due = -irpeg_due if (irpeg_due <0)
replace ecc98_irpeg_due = 0 if ecc98_irpeg_due ==.
replace irpeg_due = 0 if irpeg_due ==.


/* subtracting tax reliefs */

/* Allowable tax reliefs amount is subject to the tax due threshold. Any amount in excees can be
   brought forward up to 4 years. Companies are not eligible to the innovative investment tax
   relief if they are eligible to the territorial allowance rate */


gen tax_rel_tot = (ci_ii + ci_ric + ci_occ + ci_ct) if (redd_impon >0)
replace tax_rel_tot = 0 if (redd_impon<=0)


/* generating the variable ecc98_tax_reliefs_tot if redd_impon<0 and companies are not allowed
   to the tax reliefs. This amount can be brought forward up to 4 years */

gen ecc98_tax_rel_tot = tax_rel_tot if (redd_impon <=0)
replace ecc98_tax_rel_tot = tax_rel_tot if (irpeg_due <0)
replace ecc98_tax_rel_tot = 0 if ecc98_tax_rel_tot ==.

/* eligible amount is defined only for companies w/ positive taxable income and positive tax due */

gen eleg_tax_rel_amt = tax_rel_tot if (tax_rel_tot <= irpeg_due & redd_impon>0 & irpeg_due>0)
replace eleg_tax_rel_amt = irpeg_due if (tax_rel_tot > irpeg_due & redd_impon>0 & irpeg_due>0)
replace eleg_tax_rel_amt = 0 if eleg_tax_rel_amt ==.

/* still some companies tax due can be negative. Replacing these observations with nil value */

replace irpeg_due = 0 if irpeg_due<0

/* tax due */

replace irpeg_due = (irpeg_due - eleg_tax_rel_amt)

replace ecc98_tax_rel_tot = (tax_rel_tot - irpeg_due) if (tax_rel_tot > irpeg_due & redd_impon>0 &
irpeg_due>0)
replace ecc98_tax_rel_tot = 0 if ecc98_tax_rel_tot ==.

/*
keep codice ateco regione addetti_totali tipo_fg redd_impon  redd_imp_ord  dit_ut_agv /*
     */ txc_div tax_rel_tot   irpeg_gross  irpeg_due   ecc98_irpeg_due ecc98_loss /*
     */ ecc98_dit_ut_agv ecc98_tax_rel_tot
*/

do label_irpeg.do

format irpeg* %12.0f

label data "output corporate tax due (irpeg)"
save out_irpeg.dta, replace


/*****************************************************
 LABELS OF VARIABLES GENERATED BY THE PROGRAM IRPEG.DO
*****************************************************/

lab var irpeg_gross "gross corporate tax"
lab var irpeg_due "corporate tax due"
lab var redd_impon "taxable income"
lab var redd_imp_ord "taxable income subject to the statutory tax rate"
lab var dit_ut_agv "allowable DIT income subject to the reduced rate"
lab var txc_div "dividend tax credit"
```

```
lab var tax_rel_tot "tax reliefs"
lab var ecc98_irpeg_due "corporate tax to be brought forward"
lab var ecc98_loss "fiscal loss of year 1998 to be brought forward"
lab var ecc98_dit_ut_agv "allowable DIT income to be brought forward"
lab var ecc98_tax_rel_tot "tax reliefs to be brought forward"


/************************************
 SUMMARY STATISTICS
************************************

do_file computing summary statistics of corporate income and main tax variables */

clear
version 7
set more off
capture log close
log using statistics.log, replace
capture program drop stats_1
capture program drop stats_2
capture program drop stats_3


/* generating a file with all relevant variables */

/* potrebbe non servire */

use out_corp_inc.dta
sort codice
save out_corp_inc.dta, replace

use out_irpeg.dta
sort codice
merge codice using out_corp_inc.dta
assert _merge==3
drop _merge
save out_stats.dta, replace


use out_stats.dta, clear

/* dividing amounts of using variables by 1000 */

qui replace utile_lor_sci = utile_lor_sci*1936.27/1000000
qui replace corp_inc = corp_inc*1936.27/1000000
qui replace redd_impon = redd_impon*1936.27/1000000
qui replace irpeg_gross = irpeg_gross*1936.27/1000000
qui replace irpeg_due = irpeg_due*1936.27/1000000

xtile percentili=utile_lor_sci if utile_lor_sci>0, nq(20)


/* program to compute summary statistics (number, total amount in euros) for profits,
   corporate income */

program define stats_1
tabstat `1' if(`1'>0), by(`2') stats(n sum)col(stat) format(%12.0f)
tabstat `1' if(`1'<0), by(`2') stats(n sum)col(stat) format(%12.0f)
end

/* companies reporting profits, losses */

*stats_1 utile_lor_sci classe_base_irap
*stats_1 utile_lor_sci l_ateco
*stats_1 utile_lor_sci tipo_fg

/* corporate income */

*stats_1 corp_inc classe_base_irap
*stats_1 corp_inc l_ateco
*stats_1 corp_inc tipo_fg


/* program computing sumary statistics for tax variables (number, total amounts)*/

program define stats_2
```

```
tabstat `1', by(`2') stats(n sum)col(stat) format(%12.0f)
end

/* gross tax */

*stats_2 irpeg_gross classe_base_irap
*stats_2 irpeg_gross l_ateco
*stats_2 irpeg_gross tipo_fg

/* tax due */

stats_2 irpeg_due classe_base_irap
stats_2 irpeg_due l_ateco
stats_2 irpeg_due tipo_fg
*stats_2 irpeg_due percentili


/* computing (effective) statutory tax rates for companies with positive taxable income */

gen ESTR = irpeg_gross/redd_impon if redd_impon>0
format ESTR %12.4f


/* computing ex-post implicit tax rates for companies with positive fiscal income */

gen EPITR = irpeg_due/corp_inc if corp_inc>0
format EPITR %12.4f
capture program drop stats_3

/* program computing mean effective average corporate tax rates */

program define stats_3
tabstat `1', by(`2') stats(mean)col(stat)
end

*stats_3 ESTR classe_base_irap
*stats_3 ESTR l_ateco
*stats_3 ESTR tipo_fg

*stats_3 EPITR classe_base_irap
stats_3 EPITR l_ateco
stats_3 EPITR tipo_fg
stats_3 EPITR percentili

gen cla_add= 1 if addetti_totali>=100 & addetti_totali<150
replace cla_add=2 if addetti_totali>=150 & addetti_totali<200
replace cla_add=3 if addetti_totali>=200 & addetti_totali<250
replace cla_add=4 if addetti_totali>=250 & addetti_totali<500
replace cla_add=5 if addetti_totali>=500


stats_3 EPITR cla_add

do label_stat_irpeg.do

save out_stats.dta, replace

log close

/*******************************************************
 LABELS OF VARIABLES GENERATED BY PROGRAM STAT_IRPEG.DO
*******************************************************/

label variable ESTR "Effective Statutory corporate tax rates"
label variable EPITR "Ex_post Implicit corporate tax rates"
label variable cla_add "Classes of employees"
```

REFERENCES

Bardazzi R., Gastaldi F., Pazienza M.G.(2003*), The Effects of the Planned Social Security Reform on Firms: a Microsimulation Analysi*s, Paper presented at Annual Italian Public Finance Meeting (Società Italiana di Economia Pubblica), Pavia, 3-4 Ottobre 2003

Bardazzi R., Parisi V., Pazienza M.G.(2003), *Modelling direct and indirect taxes on firms: a policy simulation*, forthcoming in Austrian Journal of Statistics

Buglione di Monale E. (2004), Le regioni: struttura attuale delle entrate e delle spese, mimeo

Campolongo F., Tarantola S., Saltelli A., (1999), Tackling qualitatively large dimensionality problems, *Computer Physics communications*, 117, 75-85.

Campolongo, Cariboni, and Saltelli (2003) Sensitivity analysis: the Morris method versus the variance based measures, European Commission, Joint Research centre, (submitted).

DIECOFIS. Deliverable 1.1 (2002). Overview of the Issues in Longitudinal and Cross-Sectional Multi-Source Databases. M. Denk and F. Oropallo.

DIECOFIS Deliverable 5.1 (2003), The Social Contribution (SC) module: Development and results of the IRAp-SC Italian model, by Bardazzi R., Pazienza M.G.

DIECOFIS Deliverable 5.2 (2002), The Irap Module, by Bardazzi R., Gastaldi F., Pazienza M.G.,

DIECOFIS Deliverable 6.2 (2002), Results and country IT tax model, by Castellucci L., Coromaldi M, Parisi V., Perlini L., Zoli M.

DIECOFIS Deliverable 7.2 (2003), A cross country suimulation exercise using The DIECOFIS corporate tax model, by Parisi V.

Lusignoli L., (2004) Social Dumping in EU, mimeo

McLure, C.E. (1986) 'Tax competition: is what's good for the private goose also good for the public gander?', *National Tax Journal,* 39(3), September: 341-8.

Morris M.D. (1991): Factorial sampling plans for preliminary computational experiments, *Technometrics*, 33, 161-174.

OCSE, *Harmful Tax Competition: an Emerging Global Issue,* Parigi 1998

Tiebout, C. M. (1956), *A Pure Theory of Local Expenditures*," Journal of Political Economy, 64, 416-424