DAD: A software for poverty and distributive analysis *

Abdelkrim Araar[†] and Jean-Yves Duclos[‡]

April 27, 2009

Abstract

DAD is designed to facilitate the analysis and the comparisons of social welfare, inequality, poverty and equity across distributions of living standards and using disaggregated data. It is made available at no charge. *DAD*'s features include the estimation of a large number of indices and curves that are useful for distributive comparisons as well as the provision of various statistical tools. It is currently the only software that systematically takes into account the sampling design of commonly used surveys in calculating asymptotic, bootstrap and p-bootstrap statistics for carrying out statistical inference. Many of *DAD*'s features are useful for estimating the impact of programs (and reforms to these programs) on poverty and equity.

Keywords: Poverty, equity, inequality, statistical inference, software.

JEL Classification: L86, I30, I32, D63, C12.

*This work was carried out with support from CRSH, FQRSC and the Poverty and Economic Policy Research Network, which is financed by the Government of Canada through the International Development Research Centre and the Canadian International Development Agency and by the Australian Agency for International Development. We are grateful to Charles Renfro and to an anonymous referee for very useful comments.

[†] Département d'économie and CIRPÉE, Pavillon De Sève, Université Laval, Sainte-Foy, Québec, Canada, G1K 7P4; Email: aabd@ecn.ulaval.ca; fax: 1-418-656-7798;

[‡] Département d'économie and CIRPÉE, Pavillon De Sève, Université Laval, Sainte-Foy, Québec, Canada, G1K 7P4; Email: jyves@ecn.ulaval.ca; fax: 1-418-656-7798; phone: 1-418-656-7096

1 Introduction

1.1 Contribution and origin of *DAD*

DAD – which stands for ”Distributive analysis/Analyse distributive” – is designed to facilitate the analysis and the comparisons of social welfare, inequality, poverty and equity using micro (or disaggregated) data. We can separate the analysis of survey data into four steps. First, there is the data acquisition step, which involves getting access to the data and inputting it into a software. Second, there is the data organization step; this transforms the data into a form that permits the analysis to be performed. For instance, this may mean recoding variables, checking for data consistency, or removing outliers. *DAD* was not designed to make it easier to go through this second step; other software packages such as SAS, SPSS or STATA are much better at this.

DAD was rather primarily designed to facilitate the third step, that of data analysis, and to move more quickly to the final step, that of understanding and presenting the results to other people in the form of articles, books, and presentations. In that respect, a combination of four features distinguishes *DAD* from other software packages that can be used to analyze data.

First, it is available at no charge and does not require purchasing any additional commercial software. An operating system must of course be installed on the computer. *DAD* is optimized for Microsoft’s Windows operating system, but as discussed below, it can also run under alternative platforms. Second, it is user friendly and requires no scientific programming – all estimates can be computed using a mouse. Third, it covers a very wide range of distributive indices and curves, some of them very recently introduced in the literature. Finally, the sampling distribution of all of *DAD*’s estimates can be systematically provided while taking into account the sampling design of household surveys by means of analytical and numerical procedures.

The first version of *DAD* was launched in September 1998¹. It initially came to life following a request by the Canadian International Development Research Centre (IDRC) to Université Laval to support research then carried out in Africa in the context of the IDRC’s programme on the Micro Impacts of Macro-economic and Adjustment Policies (MIMAP). Improved versions of *DAD* subsequently appeared as errors and bugs were corrected and as attempts were made to make it more reliable, more flexible and broader in scope. The current version is 4.4,

¹The software *DAD* was conceived by Jean-Yves Duclos and Araar Abdelkrim and was programmed in Java by Araar Abdelkrim and Carl Fortin.

released in August 2006.

1.2 Motivation

Several factors motivated us in the process of building *DAD*. First, there seemed to be an ever increasing need for developing-country analysts to carry out poverty and inequality "profiles". Much of development policy is indeed now assessed through poverty criteria, and this is carried out among other things through the elaboration of poverty assessments, poverty reduction strategy papers (the now well-known and so-called *PRSP*'s), poverty and social impact analyses, *etc.*. The majority of these distributive assessments had earlier typically been done by foreign consultants and by international organizations' technical staff. Little was left in the form of national capacity building and local empowerment following these largely external exercises. Local researchers and national policy analysts typically felt alienated by these poverty assessments which they often did not understand and which they could not usually influence. To break that segregation between foreign expertise and local policy making and analysis, it seemed important to introduce tools that would benefit developing country analysts pedagogically and operationally.

Second, micro-data accessibility was increasingly becoming less of a problem to developing-country researchers. This followed what had occurred in more developed countries some 15 to 20 years earlier when data tapes and records started to circulate widely in research centers and universities. This was made possible in large part by the amazing increase in storage and processing speed that the computer revolution was creating. Developing-country analysts were gaining from the same advances, though with some lag due to tighter resource constraints. Furthermore, in addition to the computing and technical demands that handling large data sets involved, developing country analysts often had to deal with data accessibility difficulties. This meant *inter alia* having to face skepticism and rent-seeking behavior from statistical agencies and international organization staff when requesting access to data that were supposed in principle to be public. That problem had also become less severe by the end of the 1990's, in part due to outside pressure. To process and analyze these data then typically became the next barrier to break.

Third, much of distributive analysis was (and is still) handled as if it was not subject to sampling variability. Indeed, a considerable amount of energy and resources seems to be wasted in discussions of poverty and inequality "results" that cannot be trusted on formal statistical grounds. Even changes in poverty rates

(“headcounts”) of around 4% or 5% are often statistically insignificant within the usual statistical precision criteria². Many of the poverty assessments carried out by the World Bank, for instance, still do not provide sampling errors on differences in poverty and inequality indices across time and space. The same is true of many official government and policy analyst reports on poverty and inequality in developing and developed countries alike³. Needless to say, the efforts deployed by analysts and policy makers to account for variations of less than 1% or 2% (as often occurs) in poverty rates are typically a pure loss of resources. This unfortunate state of affairs needed to be remedied by a much greater use of appropriate statistical techniques. Though conceptually relatively simple, the use of these techniques nevertheless required reading through some technical literature as well as writing tedious and intricate computer programmes. *DAD* was in large part written to help bypass these hurdles. Achieving this meant clearing the ground of statistically insignificant results and leaving more time and resources for the interpretation of those distributive findings that were statistically significant.

1.3 Basic structure

An overriding operational objective was to try to make *DAD*’s environment as accessible and as user friendly as possible. Carl Fortin, our co-author in this exercise, convincingly argued from the start that we should program *DAD* in the Java programming language. Programming involves writing a set of instructions (called a program) to tell the computer how to carry out some task. Among some of Java’s desirable properties, there is the fact that it is an object-oriented language. This allows designing sub-programs (called classes or interfaces) that contain the structured information of various objects and ensuring a proper exchange of information between the sub-programs. For example, a school database program might contain a “student” type of object; a student object might in turn contain information on a student’s name, age, home address, and grade level. Indeed, Java created a new paradigm of platform independence: once written, Java applications could (theoretically) run on any operating system as well as on the

²For evidence of this, see for instance Deaton (1998), Howes and Lanjouw (1998), Davidson and Duclos (2000) and Araar (2006).

³See for instance Nickell (2004), Meng, Gregory, and Wan (2007), Layte, Maitre, Nolan, and Whelan (2001), Schady (2002), Gang, Sen, and Yun (2008) Saunders and Bradbury (2006), and Park, Wang, and Wu (2002).

internet⁴.

Conceived by Sun in 1995, Java could still be considered in 1998 to be an infant programming language⁵. By now, however, it has become an important pillar of the programming and internet industry. To make *DAD* completely free of charge, we also chose not to tie its use to statistical commercial software packages such as Excel, SPSS or STATA⁶. We therefore opted to design *DAD* from scratch using some of Java's packages as building blocks.

To make *DAD* as user friendly as possible, we use pop-up application windows and spreadsheets as the main working tools. This enables users to visualize a lot of information at a glance, and to manage that information easily. This is done by exploiting the richness of Java's graphic user interface (*GUI*). *GUI* is an interface that contains one or more windows, with labels, text fields, buttons, spreadsheets, *etc.*. It receives information from and displays information to a user. At the time of conceiving *DAD*, considerable efforts were made to ensure the effectiveness of *DAD*'s *GUI*. Most of the relevant variables and options needed for running applications can be selected from single application windows.

With *DAD*'s Data Import Wizard, ASCII files can be easily loaded, and users can then save them in *DAD*'s data format in order for those data files to be loaded easily in subsequent work next sessions. *DAD*'s use of spreadsheets has the advantage of displaying the entire data sets to be used. Small data sets can easily be entered manually. Changes to cell values can be made directly on the spreadsheet. The results of operations on data vectors can be checked easily. *DAD* also allows loading two data bases simultaneously, and to display each of these two data bases alternatively on the spreadsheet. This makes it straightforward to carry out

⁴Java was designed as a cross-platform language. The same compiled code can in principle run on any computer. Java accomplishes this by running programs within a Java Virtual Machine (JVM). The JVM is another program that simulates a computer that has Java code as its instruction set. The Java compiler then translates the Java program into the codes needed by the JVM. Hence, a Java program runs on a computer that is simulated by one's own computer.

⁵For further information on Java's development and structure, see Deitel and Deitel (2003)'s introductory book, or Chapter 1 of Lewis and Loftus (2000).

⁶Despite the fact that these are commercial software packages, they are not necessarily always reliable. For instance, McCullough (1998) and McCullough (1999) have developed a methodology to test the accuracy of statistical software and have found errors of various importance in SAS, SPSS, and Excel — they found the latter to contain errors that were so important that they advised not to conduct statistical analyses with Excel. See also McCullough (2006) for an update on this work. Some of these software packages, including STATA, are described in a software compendium published in a 2004 special issue of the *Journal of Economic and Social Measurement* — see Renfro (2004) for an introduction.

applications with either one or two data bases. That structure also enables *DAD* to account for whether the data bases are independent when it comes to computing standard errors on distributive estimators that use information from two samples.

2 Loading, editing and saving databases in *DAD*

DAD's databases are displayed on spreadsheets similar to those of SPSS, STATA, or Microsoft's Excel – see Figure 1. Every line in a sheet represents one observation or one data "record". Typically, an observation consists of one of the sampling or statistical units that were drawn into a survey. In distributive analyses, the relevant "last" sampling unit is often a household since it is households that are typically the sampling units that are drawn last into surveys — possibly after having selected primary and secondary sampling units such as departments or some other enumeration areas. When observations represent households, there will thus be as many lines or observations in the data as there are households drawn into the household survey.

The statistical units (or units of interest) are usually (for normative reasons) the individuals — see page 11 below for a discussion of this. Even though the sampling units originally drawn into the survey may have been the households, data sets are sometimes re-organized in such a way that each individual in a household is assigned its own line of data. There will then be as many observations in a data set as there are individuals found in the households.

A database used in *DAD* is then a matrix (a set of columns) whose length is the number of observations discussed above and whose width is the number of variables contained in the database. Each column displays the values of a variable. A variable has as many values as there are observations in the database. All columns in *DAD* are therefore of the same length. Variable values can have a float format – indicating, for example, the level of household income – or an integer format – showing for instance the socio-economic category to which a household belongs.

There are several options for entering data into *DAD*. The first one is to create a new database in *DAD* and then enter the variable values manually. This can be useful for exploratory or pedagogical purposes. Clearly, however, this option is not convenient for entering large databases into *DAD*. A second option for reading existing data bases into *DAD* is done by using well-known copy/paste facilities. Before doing this, however, a new data base must be created in *DAD* and then assigned a number of observations (or size) that corresponds to the length of the variables that will be copied/pasted.

Figure 1: The spreadsheet for handling and visualizing data in *DAD*.

The third possibility for entering data into *DAD* is typically more reliable (and also faster) than the first two and involves two steps. The first step saves the database in an ASCII (or a text) format. The way in which this is done in practice depends on the software in which the data were previously handled. *DAD*'s Users Manual gives examples of such output procedures for several common commercial software packages. One fast alternative to this is offered by the use of STAT/TRANSFER (note however that this requires buying a license), which transforms databases rapidly from the most popular formats into an ASCII format. Once the database is in ASCII format, it can easily be imported using *DAD*'s Data Import Wizard. The wizard ensures *inter alia* that the imported database does not contain missing or unreadable values. Once the data are read in *DAD*, they can be submitted to a number of arithmetical and logical operations, variable names can be added or changed, and new variables can be created. Databases can subsequently be saved in *DAD*'s preferred ASCII format (identified by the

extension *.daf*).

As already mentioned, many of *DAD*'s applications can use simultaneously two databases. To use a second database, the user should first activate a second file by clicking on the button **File2**, and then follow the same procedures as for loading a first file.

3 Inputting the sampling design information

The recent literature has emphasized the importance of carrying out statistical inference before declaring a distribution to be poorer, less equal or more polarized than another – see for instance Anderson (1996), Barrett and Donald (2003), Beach and Davidson (1983), Cowell (1989), Davidson and Duclos (1997), Davidson and Duclos (2000), Zheng, Cushing, and Chow (1995), Zheng and Cushing (2001) and Esteban and Ray (1994). It is also well known by now that failing to take into account the survey design can easily bias standard errors by a factor of at least 2 – *e.g.*, Howes and Lanjouw (1998) and Zheng (2002).

The process of generating random surveys usually displays four important characteristics:

- the base of sampling units (the base from which sample observations are drawn) is stratified;
- sampling is multi-staged, generating clusters of observations;
- observations come with sampling weights, also called inverse probability weights;
- observations may have been drawn with or without replacement;
- observations often provide aggregate information on a number of units of interest (such as the different individuals that live in a household).

DAD enables taking that structure into account in the estimation of the various distributive statistics as well as in the computation of the sampling distributions of these statistics.

When a data file is first read or typed into *DAD*, the survey design assigned to it by default is Simple Random Sampling. This supposes that the observations were independently selected from a large base of sampling units. This, however, is rarely how surveys are designed and implemented. Once the data are loaded,

the exact sampling design structure can nevertheless be easily specified. This is done using the **Set Sample Design** dialogue box. Specifying the sample design structure can involve letting *DAD* know about (up to) 5 vectors (see Figure 2).

Figure 2: The Set Sample Design window in *DAD*.

- **STRATA:** this specifies the name of the variable (in an integer format) that contains the Stratum identifiers.
- **PSU:** this specifies the name of the variable (in an integer format) that contains the identifiers for the Primary Sampling Units.
- **LSU:** this specifies the name of the variable (in an integer format) that contains the identifiers for the Last Sampling Units.
- **SAMPLING WEIGHT:** this specifies the name of the Sampling Weights variable.
- **CORRECTION FACTOR:** this provides *DAD* with a Finite Population Correction variable.

4 Applications in *DAD*: basic procedures

Once data have been read into *DAD* and that the sampling design has been specified, the field is wide open for the estimation of distributive statistics and for performing distributive tests. For every application programmed in *DAD*, there is a specific application window that facilitates the specification of variables, parameters and options to generate the desired distributive statistics. For example, Figure 3 shows the specific application window for computing the FGT poverty index with one distribution. There is a separate specific window for the case of two distributions.

Figure 3: Application window for estimating the FGT poverty index – one distribution.

DAD 4.2

Edit Inequality Poverty Welfare Decomposition Redistribution Dominance Curves Distribution Graph Window ?

FGT (Poverty)

cameroon.daf

Variable of Interest: STRATA

Size Variable: Without size

Group Variable: No Selection

Group Number: 1

Parameter(s):

Poverty line: 1000.0

alpha: 1.0

Option(s):

Standard Deviation: Without STD

Normalisation: Not Normalised

Compute Line

☒ Mean L: 0.5

☐ Quantile L: 0.5 P: 0.5

☐ Estimated Line Z: 1000 StdZ: 10

Min: 0 Max: 10

Most application windows, including that of Figure 3, are divided into three panels. The first panel is used to specify the relevant database variables needed for the estimation. The second panel (generally at the bottom of the application window) specifies the parameter values and options to be used by the estimator –

examples include the level of inequality aversion, the value of the poverty line and the percentile to be considered as well as whether indices should be normalized and whether statistical inference should be performed. The third panel activates buttons in order that various types of results may be generated. Some application windows can also generate popping-up dialogue boxes. One example of this can be found when clicking on the **Compute line** button in the **Poverty|FGT** application window. This serves to specify the manner in which the poverty line should be (or was) estimated.

The following basic variables are typically required for carrying out *DAD*'s computations.

- **VARIABLE OF INTEREST.** This is the variable that usually captures living standards. It can represent, for instance, expenditures per adult equivalent, calorie intake, normalized height-for-age scores for children, *etc.*.
- **SIZE VARIABLE.** This refers to the “normative” or physical size of the observation. For the computation of many distributive statistics, we will indeed wish to take into account how many relevant individuals (or statistical units) are found in a given observation. We might, for instance, wish to estimate inequality across individuals or the proportion of children who are poor. Individuals and children will then be respectively the statistical units of interest. Households do differ, however, in their size or in the number of children they contain. *DAD* takes this into account through the use of the **SIZE VARIABLE**. When an observation represents a household, computing inequality across individuals requires specifying household size as the **SIZE VARIABLE**, whereas computing poverty among children requires putting the number of children in the household as the **SIZE VARIABLE**. If the statistics of interest were the proportion of households in poverty, then no **SIZE VARIABLE** would be needed.
- **GROUP VARIABLE.** (This should be used in combination with **GROUP NUMBER**.) It is often useful to limit some distributive analysis to some population subgroup. We might for example wish to estimate poverty within a country's rural area or within the group of public workers. One way to do this is to set **SIZE VARIABLE** to zero for all of the observations that fall outside these groups of interest. Another way is by defining a **GROUP VARIABLE** whose values will allow *DAD* to identify which are the observations of interest.

- **GROUP NUMBER.** GROUP NUMBER tells *DAD* on which value of the GROUP VARIABLE to condition the computation of some distributive statistics. The value for GROUP NUMBER should be an integer. For example, rural households might be assigned a value of 1 for some variable denoted as *region*. Setting GROUP VARIABLE to *region* and GROUP NUMBER to 1 makes *DAD* know that we wish the distributive statistics to be computed only within the group the rural households.
- **SAMPLING WEIGHT.** Sampling weights are the inverse of the sampling rate. They are best specified once and for all using the **Set Sample Design** window (as discussed above). Distributive statistics (but not necessarily their sampling distribution and standard errors) will be left unchanged, however, if no variable is given for Sample Weight (in **Set Sample Design** window) and if the product of the sampling weight and size variables is subsequently specified as the SIZE VARIABLE in the relevant application windows.

DAD's applications with two distributions can be launched after having loaded two databases. Each time one launches an application that can support two distributions, the dialog box, shown in Figure 4, opens to allow the user to specify the desired number of distributions to be used as well as the name of the databases for these distributions. The application window for two distributions is very similar to that for one. The main difference is the addition of a second panel to specify the relevant variables to be used for the second distribution. The application for two distributions generally serves to compute distributive differences across the two distributions. For curve applications with two distributions, for instance, differences between the curves of the two distributions can usually be drawn.

5 Curves

DAD has built-in tools that facilitate the use of curves to display distributive information. Say, for instance, that we wish to graph a Lorenz curve. We can compare it to the 45° line to observe by much income shares differ from population shares. This is done by following these steps:

- From the main menu, select the submenu : **Curve|Lorenz**. Indicate that the number of distributions equals one.
- After choosing the application variables, click on the button **Graph** to draw the first Lorenz curve.

Figure 4: Choosing between configurations of one or two distributions.

- If you would like to draw another Lorenz curve for another variable of interest, return to the Lorenz window application, re-initialize the variable of interest and click again on the button **Graph**.
- When the graph window appears, click on the button **Draw all** to plot all of the curves.
- If you wish to draw the 45° line, select (from the main menu of the graph window) **Tools | Properties**, and activate the option **DRAW THE 45° LINE**

Figure 5 shows an example of Lorenz curves drawn by *DAD*.

We can also compare two Lorenz curves to test for inequality dominance of one distribution over the other. For this, we choose again the application **Curves|Lorenz**, but this time with two distributions.

DAD can also usually draw curves that show how the levels of some distributive statistics vary with normative parameters – such as inequality or poverty aversion parameters. Take for instance the Atkinson index of inequality. It may be informative to check how fast it varies as a function of ϵ , its parameter of inequality aversion. To do this, follow these steps:

Figure 5: Lorenz curves for two distributions

- From the main menu, select the submenu : Inequality|Atkinson. Indicate that the number of distribution equals one.
- After setting the application variables, click on the button **Range** and specify the desired range for the parameter ϵ .
- Click on the button **Graph** to draw the curve that shows the Atkinson index against the risk aversion parameter ϵ .
- When the graph window appears, click on the button **Draw** to plot the curve.

6 Graphs

Recent versions of *DAD* are quite flexible in terms of editing, saving and printing graphs. On most application windows, a button **Graph** is available to draw graphs instantly. The type of graphs drawn depends on the application and on the

Figure 6: Differences in Lorenz curves drawn by *DAD*.

Difference between the Lorenz curve of gross income and net income
Canada 1994

type of **Graph** buttons selected. There are for instance two **Graph** buttons in the Poverty|FGT Index application window. Clicking on the GRAPH button plots estimates of the FGT index for a range of alternative poverty lines. Clicking on the GRAPH2 button draws instead estimates of the equally-distributed poverty that is equivalent to the estimated FGT poverty index, and this for a range of poverty aversion parameters α .

Most of the options for editing *DAD*'s graphs can be accessed from the Graph Properties dialogue box – see Figure 7. *DAD*'s graphs can also be saved in a variety of formats. Table 1 lists some of them.

Curves are useful tools to check various types of distributive dominance. Table 2 sums up some of the links between some of the applications and curves found in *DAD* and the tests for various orders of social welfare, poverty and inequality dominance.

Table 1: Available format to save *DAD*'s graph.

Extension	Description
*.png	Portable Network Graphic
*.pmb	Bitmat Image file
*.tif	Tag Image File Format
*.jpg	JPEG File Interchange Format
*.pdf	Portable Document Format
*.ps	Postscript

Figure 7: The dialogue box for graphical options

Table 2: Curves and stochastic dominance.

	Primal approach	Dual approach
Order	Social welfare	
1	Distribution Distribution function	Curves Quantile
2	Dominance Poverty Dominance $s = 2$	Curves Generalized Lorenz
s	Dominance Poverty Dominance	
	Poverty	
1	Dominance Poverty Dominance $s = 1$	Curves Poverty Gap
2	Dominance Poverty Dominance $s = 2$	Curves CPG
s	Dominance Poverty Dominance	
	Inequality	
1	Dominance Inequality Dominance $s = 1$	Curves Normalised Quantile
2	Dominance Inequality Dominance $s = 2$	Curves Lorenz
s	Dominance Inequality Dominance	

7 Statistical inference: standard deviation, confidence intervals and hypothesis testing

DAD facilitates statistical inference in a number of original ways:

- *DAD* readily provides asymptotic standard errors on a large number of estimators of distributive statistics, including estimators of inequality and social welfare indices, normalized/un-normalized poverty indices, poverty indices with deterministic/estimated poverty lines, poverty indices with absolute/relative poverty lines, equally-distributed-equivalent incomes and poverty gaps, quantiles, density functions, non-parametric regressions, points on a large number of curves, crossing points of curves, critical poverty lines, differences in indices and curves, ratios of various statistics, various income/price/population impacts and elasticities, distributive decompositions into demographic/factor components, progressivity, redistribution and equity indices, dominance statistics, *etc.*. It can be (and has typically formally been) shown that all of these estimators are asymptotically normally distributed.
- *DAD* can calculate the sampling distribution of most of these estimators taking into account the sometimes complex design of the survey. This is

done as indicated in Section 3. Existing (commercial) software can sometimes take this design into account, but only for a small number of relatively simple distributive statistics (such as simple sums and ratios).

Figure 8: STD option.

Standard deviation, confidence interval and hypothesis testing

Sampling design option

☒ Full sampling design ☐ Simple random sampling

Estimation approach for computing sampling distribution of parameter estimator

☐ Asymptotic ☒ Bootstrap

Bootstrap options

☒ Standard ☐ Pivotal Number of draws: 250

Confidence interval options

Confidence level in %: 95.0 Two-sided

Hypothesis testing

☒ Do the following test: Ho: parameter = 0.5

Confirm

- *DAD* can provide at the click of a button estimates of confidence intervals as well as test statistics and p -values for various symmetric and asymmetric hypothesis tests of interest.
- *DAD* can be used to simulate numerically the finite-sample sampling distribution of most of the above-mentioned estimators using bootstrap procedures. The bootstrap can be performed on the ordinary estimators or on (asymptotically) pivotal transforms of them. It is well known that bootstrapping on pivotal statistics leads to faster rates of convergence to the true sampling distribution than bootstrapping on untransformed non-pivotal statistics. Pivotal bootstrapping is, however, usually more costly in time and resources since it requires estimates of the asymptotic distribution of the estimators. This is not a problem for *DAD*, however, since the (sometimes complex) asymptotic standard errors of these estimators are already programmed into it. Moreover, as mentioned above, the asymptotic standard

errors and the pivotal statistics derived from them can be sample-design corrected, providing one more degree of superior accuracy for the bootstrap procedures available in *DAD*.

The Standard deviation, confidence interval and hypothesis testing dialogue box is the main tool for telling *DAD* what to do in terms of statistical inference. This box is shown on Figure 8.

8 Conclusion

This paper briefly introduces a software that facilitates the analysis of distributions (of incomes, returns, living standards, etc.). The software, which is available at no charge and is user friendly, provides state-of-the-art algorithms to compare distributions for normative and descriptive purposes using a very wide range of distributive indices and curves, some of them very recently introduced in the literature. The software comes with a Web site, users' manual and various types of supporting documents. The web site can be found at <http://www.mimap.ecn.ulaval.ca>; it contains sufficient information for new users to start using *DAD* rapidly. On the Web site can also be found a free downloadable book on measurement and estimation using *DAD*: see Duclos and Araar (2006). Users' manuals for recent versions of the software can also be accessed as well as information on projected future additions to the software.

References

- ANDERSON, G. (1996): "Nonparametric Tests of Stochastic Dominance in Income Distributions," *Econometrica*, 64, 1183–93.
- ARAAR, A. (2006): "Poverty, Inequality and Stochastic Dominance, Theory and Practice: Illustration with Burkina Faso Surveys," CIRPÉE Working Paper #0634.
- BARRETT, G. AND S. DONALD (2003): "Consistent Tests for Stochastic Dominance," *Econometrica*, 71, 71–104.
- BEACH, C. AND R. DAVIDSON (1983): "Distribution-Free Statistical Inference with Lorenz Curves and Income Shares," *Review of Economic Studies*, 50, 723–35.
- COWELL, F. (1989): "Sampling Variance and Decomposable Inequality Measures," *Journal of Econometrics*, 42, 27–41.
- DAVIDSON, R. AND J.-Y. DUCLOS (1997): "Statistical Inference for the Measurement of the Incidence of Taxes and Transfers," *Econometrica*, 65, 1453–65.
- (2000): "Statistical Inference for Stochastic Dominance and for the Measurement of Poverty and Inequality," *Econometrica*, 68, 1435–64.
- DEATON, A. (1998): *The Analysis of Household Surveys: A Microeconomic Approach to Development Policy*, John Hopkins University Press.
- DEITEL, H. M. AND P. J. DEITEL (2003): *Java How to Program*, New York: Prentice Hall.
- DUCLOS, J.-Y. AND A. ARAAR (2006): *Poverty and Equity Measurement, Policy, and Estimation with DAD*, Berlin and Ottawa: Springer and IDRC.
- ESTEBAN, J. AND D. RAY (1994): "On the Measurement of Polarization," *Econometrica*, 62, 819–51.
- GANG, I. N., K. SEN, AND M.-S. YUN (2008): "Poverty in Rural India: Caste and Tribe," *Review of Income and Wealth*, 54, 50–70.
- HOWES, S. AND J. LANJOUW (1998): "Does Sample Design Matter for Poverty Rate Comparisons?" *Review of Income and Wealth*, 44, 99–109.
- LAYTE, R., B. MAITRE, B. NOLAN, AND C. WHELAN (2001): "Persistent and Consistent Poverty in the 1994 and 1995 Waves of the European

- Community Household Panel Survey,” *Review of Income and Wealth*, 47, 427–49.
- LEWIS, J. AND W. LOFTUS (2000): *Java, Software Solutions*, USA: Addison Wesley, second ed.
- MCCULLOUGH, B. D. (1998): “Assessing the reliability of statistical software: Part I,” *The American Statistician*, 52, 358–366.
- (1999): “Assessing the reliability of statistical software: Part II,” *The American Statistician*, 53, 149–159.
- MCULLOUGH, B. (2006): “The Unreliability of Excel’s Statistical Procedures,” *FORESIGHT: International Journal of Applied Forecasting*, 3, 44–45.
- MENG, X., R. GREGORY, AND G. WAN (2007): “Urban Poverty in China and its Contributing Factors, 1986–2000,” *Review of Income and Wealth*, 53, 167–189.
- NICKELL, S. (2004): “Poverty and Worklessness In Britain,” *Economic Journal*, 114, C1–C25.
- PARK, A., S. WANG, AND G. WU (2002): “Regional Poverty Targeting in China,” *Journal of Public Economics*, 86, 123–53.
- RENFRO, C. G. (2004): “Econometric software: The first fifty years in perspective,” *Journal of Economic and Social Measurement*, 29, 9–107.
- SAUNDERS, P. AND B. BRADBURY (2006): “Monitoring Trends in Poverty and Income Distribution: Data, Methodology and Measurement,” *The Economic Record*, 82, 341–364.
- SCHADY, N. (2002): “Picking the Poor: Indicators for Geographic Targeting in Peru,” *Review of Income and Wealth*, 48, 417–33.
- ZHENG, B. (2002): “Testing Lorenz Curves with Non-simple Random Samples,” *Econometrica*, 70, 1235–43.
- ZHENG, B. AND B. CUSHING (2001): “Statistical Inference for Testing Inequality Indices with Dependent Samples,” *Journal of Econometrics*, 101, 315–35.
- ZHENG, B., B. CUSHING, AND V. CHOW (1995): “Statistical Tests of Changes in U.S. Poverty, 1975 to 1990,” *Southern Economic Journal*, 62, 334–47.