How to Access Standard Data Sources

Claudia Olivetti (1999), Dan Silverman (2002s), Jay Hong (2002f), David Wiczer (2010f,2012f) *

First Version, September 19 1998

This Version October 25, 2012

1 Introduction

These notes describe a few sources of data with which economics graduate students ought acquaint themselves, especially if they have a faint interest in the "real world." The notes briefly outline the sources and, more importantly, describe how to access and begin using those sources.

We describe the sources in Section 2 and the location and some tips on using them is given in Section 3. General data organization is described in Section 4 and some basics on extracting and manipulating data is described in Section 5 . This section also contains some useful code for extracting data with Stata and R (with Eviews and Gauss addenda).

2 The Sources

We briefly describe six US data sets commonly used by economists: the Panel Study of Income Dynamics (PSID), the Current Population Survey (CPS), the National Longitudinal Surveys of Youth (NLSY), the Survey on Income and Program Participation (SIPP), the Health and Retirement Study (HRS), the Consumer Expenditure Survey (CEX), and the Survey of Consumer Finances (SCF). The other dataset described in these notes, Citibase, is a repository of the National Income and Product Accounts (NIPA) series. The micro-level sources are both panels (PSID, NLSY, SIPP, HRS) and cross- sectional (CEX, SCF) and the CPS contains elements of both. To conclude, we give a cursory description of the Luxembourg Income Study (LIS), a repository of international cross-sectional micro-level studies.

[*] These are notes on using data sets. It was initially written by Claudia Olivetti, and subsequently modified by Dan Silverman and Jay Hong at the University of Pennsylvania, and David Wiczer at the University of Minnesota

2.1 The PSID

The PSID is a blessing and a curse. It is a longitudinal study of a representative sample of U.S. individuals and their family units. It has a wealth of information, and because it is a panel, we can observe individuals' life-cycles. Individuals enter the survey at various life stages, thus it contains multiple cohorts observed cross-sectionally and many years for each cohort. This makes it very powerful, and in a number of areas it is the favorite source. Unfortunately, it is survey data, and subject to a tremendous amount of observational error. For this reason, it has to be "cleaned" before it can be used for anything, a process much like extracting a diamond from a lump of coal.

In particular, the study focuses on income sources and amounts, employment, family composition changes and resident locations. With respect to these variables, the general design and content of the study has remained largely unchanged over the years, allowing for straightforward comparisons of the data across time. Information is collected both at the family level, (e.g. housing), and at the individual level, (e.g. age, education, earnings). The greatest level of detail exists for the 'head' of the household.

Starting with a national sample of ∼5,000 U.S. households in 1968, the PSID reinterviewed individuals from those households every year until 1997, and every other year since that time. These individuals are reinterviewed whether or not they are living in the same house or with the same people. New households are added to the sample as the children of the panel families grow older and form family units of their own. The sample size has grown from about 4,800 core households in 1968 to almost 10,700 in 1992. At the conclusion of the 2001 data collection, the PSID will have collected information about more than 62,000 individuals spanning as many as 34 years of their lives. The study is conducted by the Survey Research Center, Institute for Social Research, University of Michigan (home page: `http://www.isr.umich.edu/src/psid/`).

In 1990, a representative national sample of 2,000 Latino households, differentially sampled to provide adequate numbers of Puerto Rican, Mexican-American, and Cuban-Americans, was added to the PSID database (also the appropriate weight variable was included).

Special data format:

There is a *Cross-Year Individual Data* file that contains all the individual-level variables collected from 1968 to 2007 for both respondents and non-respondents. It contains one record for each individual associated with an interviewed family from 1968 through 2007. There is, in addition, a *Family Data File* provided for every wave (from 1968 to 2007).

Several special files, each with detailed information about a particular topic collected over the years, are released separately. Among these special files is the *Wealth File* (which includes data on the 1984, 1989, 1994, and 1999 wealth supplement), and the demographic history files including the

1985-1992 Childbirth and Adoption History File and the *1968-1985 Marriage History File* which provide details about the event and timing of each childbirth, adoption, and marriage for PSID family members.

PSID data can be used for cross-sectional, longitudinal, and intergenerational analysis and for studying both individuals and families. In particular, PSID data can be used to estimate earning processes or to collect detailed information about household behavior (e.g. time allocation, type and cost of childcare and so forth are reported).

Acquiring the data is particularly inconvenient and their website search leaves much to be desired. It does not contain all of their variables and generally returns many superfluous variables leaving a morass for the user to wade through. PSIDUSE a Stata user-written code helps somewhat. See the note below for more.

Recent Minnesota students who have used the PSID: David Wiczer, Hitoshi Tsujiyama, Naoki Takayama

2.2 The CPS

The Current Population Survey (CPS), is the primary source of labor force statistics in the U.S. The survey is administered by the Bureau of the Census under the supervision of the Bureau of Labor Statistics (BLS) and has been conducted for more than 50 years. For each sample, about 50,000 households are interviewed monthly, selected to represent the U.S. as a whole, individual states, and other specified areas. Each household is interviewed for a total of eight months: once a month for four consecutive months one year, and again for the corresponding time period a year later. Each month, new households are added and old ones are dropped. Thus, there is some short term panel component, in monthly observations, and a twice-observed annual panel.

Eight rotation groups (cohorts of households starting their interviews in the same month) are interviewed in any month. The main purpose of the CPS is to collect information on employment. Comprehensive data are available on labor force activity for the week prior to the survey, as well as employment status, occupation, and industry of adults (currently defined as 15 years of age and older). The survey is also used to collect demographic information, such as age, sex, race, marital status, veteran status, Hispanic origin, educational attainment, and family structure. Periodically, additional questions are included on such topics as health, education, income, and previous work experience. The CPS sample attempts to represent the civilian, noninstitutional population of the U.S. by using a probability sample to select housing units.

The typical unit of observation an individual within households; however, the March series also has family and household observations. Weights are supplied in the data files to expand the counts to nationally representative levels.

In addition to the *Annual Demographic Files or 'March Supplement'* (1968 to 2001), several special monthly files are collected that have detailed information about a particular topic. For example, the *January: Displaced Workers* files and the *October: School Enrollment* (not available for every year).

CPS data are valuable for studying the labor market in detail. In particular the size of the sample allows accurate analyses at a high degree of disaggregation (e.g. consider groups heterogeneous by education, gender, race, etc).

Recent Minnesota students who have used the CPS: any of the wonderful people at the Minnesota Population Center

2.3 NLSY 1979 and 1997 cohorts

2.3.1 NLSY79

The NLSY79 is a longitudinal study that began in 1979 with 12,686 men and women ages 14-21, and has interviewed this cohort every year until 1994, and every other year since then, through 2008. Unlike the PSID, there is only one cohort observed, unless it can be linked to the NLSY 1997 (see below). The primary purpose of the NLSY79 is the collection of data on each respondent's labor force experiences, labor market attachment, and investments in education and training - though a number of other topics also have been covered through the years. The NLSY79 consists of three subsamples:

1. a cross-sectional sample of 6,111 respondents designed to be representative of the non-institutionalized civilian segment of young people living in the U.S. in 1979 and born between January 1, 1957 and December 21, 1964 (and thus of ages 14-21 at the end of 1978)

2. a supplemental sample of 5,295 respondents designed to oversample civilian Hispanic, black and economically disadvantaged non-black, non-Hispanic youth living in the U.S. in 1979 and born in the same interval; and

3. a sample of 1,280 respondents designed to represent the population born between January 1, 1957 and December 31, 1961 (ages 17-21 at the end of 1978) and who were enlisted in one of the four branches of the military as of September 30, 1978.

The oversampling of racial minorities and the empirical correlations between race, income, and family structures makes the NLSY especially useful for studying topics such as (the dynamics of) discrimination, poverty, and family composition. This also means that it is especially important to use the proper weights with this data. On the BLS website, they describe the various weights for cross sections, panels and subsets.

There are other, relatively rare, but useful, features of this survey. Beginning in 1986, children of the women of the NLSY were included in a longitudinal study of their own. There exists a separate data set "*NLSY79 Children and Young Adults*" that provides detailed information on these children plus relevant items about their parents drawn from the NLSY79. Obviously these data provide a rich opportunity to study intergenerational questions.

The original 12,686 individuals also included 5,863 individuals with at least one sibling in the sample, allowing one to control for family/endowment effects. Going back to the respondent's background, the NLSY79 includes reasonably detailed information about the respondent's home environment including the family's history of moving, and the parents' education and occupation choices.

There is also regional information, as the NLSY79 may also provide information on state, county, and SMSA/MSA/CMSA/PMSA of respondents' current residence, location of most recent collect attended and select environmental variables from the Country and City Data Books for county or SMSA of current residence. These geographic data are available after assenting to some confidentiality agreements. Similar 'geocode' information is available for the later (1997) cohort.

Finally, all individuals were given the *Armed Services Vocational Aptitude Battery*, exams used by the US military for occupational placement. This gives interesting insight into "ability" at the time the tests were administered in 1980.

Recent Minnesota students who have used the NLSY79: David Wiczer, Satoshi Tanaka, Marina Tavares

2.3.2 NLSY97

The NLSY97 is another longitudinal study of a nationally representative sample. In this case, the study began in 1997 with 9,021 youths who were 12 to 16 years old as of December 31, 1996. Both parents and youth are interviewed. The survey is designed to document the transition from school to work and into adulthood, and collects extensive information about youths' labor market behavior and educational experiences over time. Labor market data include standard labor market work, and informal work for pay. Educational data include youths' schooling history, performance on standardized tests, course of study, the timing and types of degrees, and a very detailed account of progression through post-secondary schooling.

In addition, subject areas in the youth questionnaire include: Youths' relationships with parents, and friends, contact with absent parents, marital and fertility histories, dating, sexual activity, onset of puberty, training, participation in government assistance programs, volunteer and political activity, expectations, time-use, criminal behavior, and alcohol and drug-use.

Information in the parent questionnaire includes: parents' marital and employment histories, relationship with spouse or partner, ethnic and religious background, health (parents and child), household income and assets, participation in government assistance programs, youths' early child-care arrangements, custody arrangement for youth, and parent expectations about the youth. The NLSY97 also includes a detailed survey of the schools in the area the youth lives. In the fall of 1996, a survey of schools was conducted of all schools with a 12th grade in the statistical sampling areas in which NLSY97 respondents reside. The survey gathered information about the characteristics of each school, the staff, and the student body. In the winter 1999-2000, high school transcripts were obtained for eligible NLSY97 respondents. Respondents eligible for transcript data collection had either graduated from high school or were age 18 or older and no longer enrolled in high school.

2.4 SIPP

The Survey on income and program participation is a short panel that includes monthly data. It is organized in waves, the first being in the mid-1980s but it really takes its modern form with the 1996 wave. In each wave between 14,000 and 36,700 households are interviewed for the duration of the wave. Every four months, each respondent answers a questionnaire about each of the past four months, hence, the data appears monthly but there is less stress on the respondent. Until 1996, the waves were only one or two years, but the 1996 wave lasted through 2000, then there were panels from 2001-2003, 2004-2007 and 2008-2011.

The SIPP contains data on employment status, earnings, income and key demographic information. Reflecting its origins to study government assistance programs, it contains many questions regarding the use of government services. The occupation and industry codes in the SIPP tend to be fairly consistent. In addition to the "core" module, which contains the standard data already mentioned, there are "topical" modules that are questions not asked repeatedly. These range from topics child care, wealth, school enrolment and healthcare.

Recent Minnesota students who have used the SIPP: David Wiczer

2.5 HRS

Health and Retirement Study (HRS) is a nationally representative, longitudinal survey of older American households interviewed first in 1992 when they were ages 51-61 and followed every two years thereafter. As the name suggests, the primary focus of the HRS is the health and retirement behavior of older Americans, and it is among the two or three comprehensive datasets for the studying the economics of aging. The HRS contains respondent information on detailed questions covering a wide range of demographic, health and economic topics and covers 9,825 'age-eligible' respondents and their spouses. In total, the initial sample consists of more than 12,600 persons in 7,600 households, including (100%) over samples of Hispanics, Blacks, and Florida residents. Like the PSID the study is conducted by the Survey Research Center, Institute for Social Research, University of Michigan (home page: `http://www.umich.edu/~hrswww )`

The HRS has at least four distinctive features that have been exploited in recent studies. 1) It has unusually detailed data on health outcomes (including self-reported well being as well as diagnosed illnesses), and health-related behaviors such as alcohol use and smoking. 2) The HRS has among the most detailed data on assets and savings outside of the Federal Reserve's Survey of Consumer Finances. 3) With special permissions, the HRS data may be linked to administrative data on respondents' social security earnings records and pension plan provisions. One may, in other words, gain access to a respondent's entire history of earnings, and his or her pension income upon retirement.This information has typically been used to quantify the discounted benefits associated with retirement at different ages. Last, the HRS also asks an unusual number of 'experimental' questions assessing risk aversion, intertemporal substitution, and expectations.

The structure of the raw data files available on the HRS website is somewhat complex, and unfortunately it varies by the survey year (wave). For each survey year the survey responses are divided into separate topic sections such as income, demographics, attitudes and expectations, etc. In some years (waves) the raw data variables simply correspond to these sections of the survey. In other waves, the raw variables are bunched together in ways that cross these topic boundaries making extraction of the desired variables more cumbersome.

Another complication of these data, that is common to many data sets, is that a single respondent often answers questions for the entire household. Thus if, for example, you are interested in the earnings of the primary financial respondent's spouse in 1995 you will need to link the response to the question 'what did your spouse earn in salary, wages and tips last year?'to the spouse of the financial respondent.

Recent Minnesota students who have used the HRS: Amanda Michaud

2.6 The CEX

The ongoing Consumer Expenditure Survey (CEX) provides a continuous flow of information on the purchasing habits of American consumers and also furnishes data to support periodic revisions of the Consumer Price Index. The survey consists of two separate components:

- a quarterly *Interview* panel survey in which each consumer unit in the sample is interviewed every three months over a 15-month period;

- a *Diary* or record keeping survey completed by the sample consumer units for two consecutive one-week periods.

The interview survey is designed to collect data on major items of expense, household characteristics, and income. The expenditures covered by the survey are those that respondents can recall fairly accurately for three months or longer. In general, these expenditures include relatively large purchases, such as those for property, automobiles, and major appliances, or expenditures that occur on a fairly regular basis, such as rent, utilities, or insurance premiums. Expenditures incurred while on trips are also covered by the Interview. Nonprescription drugs, household supplies, and personal care items are excluded. If we consider global estimates on spending for food, it is estimated that about 90 to 95 percent of expenditures are covered in the Interview survey.

The Consumer Unit Characteristics and Income (FMLY) files contain consumer unit characteristics, consumer unit income, and characteristics and earnings of both the reference person and the spouse. Summary expenditure data are also provided. The Member Characteristics and Income (MEMB) files present selected characteristics for each consumer unit member, including reference person and spouse. Each record in the FMLY and MEMB files consists of three months of data. Detailed Expenditures (MTAB) files provide monthly data at the Universal Classification Code (UCC) level. In these files, expenditures for each consumer unit are classified according to UCC categories and are categorized as gifts or non-gifts. Depending on what is reported to the interviewer, there may be more than one record for a UCC in a single month. The Income (ITAB) files supply monthly data at the UCC level for consumer unit characteristics and income.

Several recent studies have used CEX data to enrich the PSID, the former being only cross-sectional and the later containing only food expenditures until very recently. However, because the PSID and CEX both contain enough demographic data and overlapping categories of consumption, one can use consumption patters in the CEX to extrapolate how such a person (with matching demographics and food consumption) in the PSID would consume. This is described by Richard Blundell in “Imputing consumption in the PSID using food demand estimates from the CEX.”

CEX survey data are available for the periods 1960-1961, 1972-1973, and 1980-2008. The survey and diary data are presented separately for most of these years.

Recent Minnesota students who have used the CEX: Citrad Slavik, Enoch Hill, Nathalie Pouokam,

2.7 The SCF

The Survey of Consumer Finances (SCF) is conducted every three years to provide detailed information on the finances of U.S. families. It is a survey of the balance sheet, pension, income, and other demographic characteristics of U.S. families. The survey also gathers information on the use of financial institutions. No other study for the country collects comparable information. The study is sponsored by the Federal Reserve Board in cooperation with the Department of the Treasury. Since 1992, data have been collected by the National Opinion Research Center at the University of Chicago (NORC). One part of the sample is representative of the U.S. population, to give an accurate description of entire population. The second part over samples rich households, to get a more precise idea about the precise composition of this groups' income and wealth composition. Rich household group accounts for the majority of total household wealth, and therefore it is important to have good information about this group. For the most part, the SCF is not a panel data set and so the dynamics of income and wealth accumulation cannot be documented using this data set. There is an interesting exception: in 2009, the respondents to the 2007 survey were given follow-up questions. The aim was to see the effects of the financial crisis on households.

SCF data are available for 1962-1963, 1983, 1986, 1989, 1992, 1995, 1998, 2001, 2004, 2007, 2010 (In process).

Recent Minnesota students who have used the SCF Jake Short, Andy Glover, Daolu Cai

2.8 The Luxembourg Income Study

The Luxembourg Income Study (LIS) is actually two databases that have collected and systematized studies from various countries. The LIS Database includes income microdata from many countries at multiple points in time. The newer Luxembourg Wealth Study Database includes wealth microdata from a smaller selection of countries. Both databases include labor market and demographic data as well. Details of the data vary by country and not every variable is available for every point in time or for every country. At most, countries have cross-sectional data taken about every five years since the early 1980s. Most of the OECD is represented along with some Latin American countries. For cross country studies, the LIS has some major advantages. It creates comparable variables all with the same name. They have also already converted values such as educational attainment and currency-denominated variables so that countries may be compared. The LIS also provides documentation and some data quality control.

Users have to register, and this requires some permissions and processing time.

LIS data are available approximately 1980,1985, 1990, 1995, 2000, 2005 but depending on the country dates vary and not all dates are available.

Countries: Australia, Austria, Belgium, Brazil, Canada, Colombia, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Guatemala, Hungary, Ireland, Israel, Italy, Japan, South Korea, Luxembourg, Mexico, Netherlands, Norway, Peru, Poland, Romania, Russia, Slovak Republic, Slovenia, Spain, Sweden, Switzerland, Taiwan, United Kingdom, United States, Uruguay

3 Accessing and Using the Data

All of these data sets are available to lowly Grad Students for free on the web.

- **PSID**: For the PSID the webpage is well designed and the data are free, though it takes an unusually long time to download the data files. Here you can find complete documentation for the data: codebooks, sample code in Stata, SPSS and SAS for data extraction, instructions for how to construct the panel, useful references about papers that use PSID data, etc. `http://www.isr.umich.edu/src/psid/`

  A word of caution however, is that the "drill down" menu provided on the ISR website *does not include all of the variables*, so you can easily miss useful data if you rely upon creating extracts this way. Their search function also is tremendously imprecise and frequently returns far more variables than one desires. It has been my experience that the best way to get PSID data is to download entire years' worth of family-level data (about 6 mb) and the entire linked individual data (about 20 mb) and then creating subsets once it is on your computer. Download these packs from: `https://simba.isr.umich.edu/Zips/ZipMain.aspx` (requires a free username). Then, there is a provided script file to import the raw data into that stat package of your choice. The one trick is that in Stata, before running the *do* file, you'll have to edit it to allocate the max of system resources to the process, `set maxvar 32000`, and change the working directory.

  For those using Stata, a special package exists for importing PSID variables. PSIDUSE is created by Ulrich Kohler and can be installed by `ssc install PSIDUSE`. Otherwise, it has to be extracted as a raw text file and imported. As in most panels, variables have a different name for each year, so to utilize the panel, these variables must be matched. PSIDUSE helps do this for you.

- **CPS**: There are three fantastic sources for CPS data. The cleanest is from the Minnesota Population Center (MPC) which maintains its Integrated Public USE Micro-data Series (IPUMS). One creates a request that then contains raw data and import routines for several statistical packages including STATA. Unfortunately, there is no way to chain the observations over two years. The identification variables are not consistent across yearly observations, though the MPC guys told me (David) that they were working on this last year (2009). The NBER's data center also includes CPS data, routines and instructions to match observations across the two years in which they are observed. Unfortunately, the NBER does not perform the nice cleaning service as in IPUMS. In both sources, special care should be taken with the occupational codes, which have been shown to be rife with error. The CPS can also be accessed directly from the BLS website and with their "Data Ferret" program.

http://cps.ipums.org/cps/
http://www.nber.org/data/cps_index.html
http://www.bls.census.gov/cps/

- **NLSY**: The BLS website has several nice tools to access their data. Specific variables can be downloaded directly from the "web investigator." Data ferret can also be used on Windows machines to download and search for data quickly. All of this is described on the website http://www.bls.gov/nls/nlsorder.htm. For some variables, both 1997 and 1979 cohorts are observed. Thus, the two can be combined following instructions on the BLS website: http://www.nlsinfo.org/nlsy97/nlsdocs/nlsy97/tutorials/

- **SIPP** You can download the data from the SIPP website http://www.census.gov/sipp/ or from the CEPRdata site, where the separate data files have already been compiled for you http://ceprdata.org/sipp-uniform-data-extracts/

- **HRS**: The unrestricted HRS data can be downloaded for free from either the study's website. Or from a site at RAND that has organized the files in a somewhat more user-friendly manner. Beyond the usual data and documentation to download, two other files are particularly handy when using the HRS. First is the 'box and arrow' file which provides a schematic describing, pictorially, the ordering of the survey as a function of the response to each questions. If, for example, respondent answers 'no' to the question 'is your spouse still alive,' this will lead to a subsequent question that would not be answered if the answer were yes. The second useful file (or web function) is the 'concordance' which helps identify the same questions in each wave. This will help you locate, for example, the questions about 'activities of daily life' in each of the five existing waves.
  http://www.umich.edu/~hrswww/
  http://www.rand.org/labor/aging/dataprod/hrsahead/

- **CEX:** With regard to the CEX data: There are some very useful extracts available on the NBER homepage from 1980-2003. The raw data files for this survey are notoriously cumbersome to use. The files in the NBER's directory are provided by John Sabelhaus and Ed Harris of the Congressional Budget Office, and are reformatted to make access to the data more straightforward. The NBER also has several programs to parse the data and load it into fortran programs.

  `http://www.nber.org/data_index.html`

  The BLS website will allow specific variables to be extracted from the web. By this method, however, once the data is downloaded it is up to you to convert from spreadsheet form into something usable.

- **SCF:** The SCF data are accessible at the website provided by the Federal Reserve Board. The data can be downloaded in Stata and SAS formats or as raw ASCII text

  `http://www.federalreserve.gov/pubs/oss/oss2/scfindex.html`

- **LIS:** To access the LIS and LWS, registration and an advisor's letter is required. The hurtle is relatively small, however, and then these rights will be renewed yearly so long as you request. The data can eventually be exported into Stata, SAS or SPSS formats, but to make these requests one has to go through the LISSY web program. Because there is no panel element, this does not have to be set up, and when downloaded all of the variables are consistent and comparable across countries, a major time saver. For access see:

  `http://www.lisproject.org/data-access/lissy.htm`

  Because the LIS is only a curator of the original country-level surveys, it may be useful to look up the state agencies' raw data and documentation. A list of the underlying surveys can be found here:

  `http://www.lisproject.org/techdoc/surveys.htm`

4 Organization and Structure of the Files

In its most raw form data comes in a rectangular array. Every unit of observation (individual, household or family) is described by a line of numbers (or characters) representing the realization of the variables of interest for this entity. Usually each entity is characterized by an interview or identification number. Observations can be sorted by this ID number. If we were to consider a household, then every individual component of the household would be characterized by the same household ID number, and by a unique identifier within the household.

For a given observation, the realizations of the different variables will often be delimited by spaces or commas.

Example: Suppose our sample is composed of two individuals and we observe the sex, age, and completed education of the individual for the year 1992. If the raw data, let's call that file rawdata.txt, were 'comma-delimited' they would appear in an ascii file as:

```
1,1,23,12
2,2,27,16
```

These raw data indicate that the first observation has i.d. number 1, has a sex variable equal to 1, an age variable equal to 23, and an education variable equal to 12. Similarly the second observation has i.d. number 2, sex 2, age 27 and education 16.

If you are somewhat less lucky, the variables within a string will not be delimited in any way. The data will appear in an ascii file simply as uninterrupted strings of numbers or characters. In this case, the variable LRECL (length of the record) will provide the information about the length of each string, and the codebook will provide the length and location of each variable in that string.

Example (contd.): Suppose the same file, rawdata.txt, came without delimiters. Then the ascii file would appear as:

```
112312
222716
```

In this case the length of the record is LRECL=6 and the codebook provided with the data will give you the following information:

Variable	*Label*	*Location*	*Length*
V1	1992 Interview number	1	1
V2	sex	2	1
V3	age	3	2
V4	completed education	5	2

5 Data Extraction and Handling

No matter whether the raw data are delimited, or not, the first step in the construction of a useful data set is to extract the desired data, and 'read' the raw records into a tabulated format for use with statistical analysis software such as Stata, R, SAS, Gauss, SPSS, Matlab, Excel, etc. Such a tabulated format will look like this:

ID	SEX	AGE	EDUC
1	1	23	12
2	2	27	16

For the longitudinal studies, the data-files are usually collected on a yearly basis except for the Cross-Year Individual Level PSID file, in which the information from every year is given on the same record for each individual.

e.g.

ID92	SEX92	AGE92	EDUC92	ID93	SEX93	..	..

5.0.1 Delimited Raw Data and Using StatTransfer

If your raw data is delimited (and it usually can be obtained in such a format), then the command line version of StatTransfer program available on the *Odin* unix server is very useful for generating the desired tabulated format. Generally, Stat-Transfer is designed to translate data sets from one program format (such as Stata or Matlab) into another (such as Gauss or S-Plus). More specifically, Stat-Transfer can translate a delimited ascii file like the first rawdata.txt file described in the previous section into the desired tabulated format.

To use the GUI version of program:

1. Choose the 'Input File Type' to be Delimited ASCII.
2. Under file specification enter (or browse for) the name of your raw data file 'rawdata.txt'
3. Choose the 'Output File Type' (e.g. Stata, Gauss, Matlab, SAS).
4. Choose those variables and observations you want to extract (using the options under the 'variables' and 'observations' tabs)
5. Press the 'Transfer' button.

The command line is even simpler:

```
st [input file] [output file]
```

Stat Transfer will figure out from the extensions what types of files it is working with.

The result is a new file in the desired (Stata, Gauss, Matlab, or SAS) format. There are also options to transfer only a subset of the data. This can really speed up the process because you'll have to write fewer values to the new file. To see all of your options in StatTransfer open it with the command `st` and then type `help set all`

5.0.2 Undelimited Raw Data and Data Dictionaries

If your raw data are not delimited, you will need to use more elaborate computer code, a *data dictionary*, in order to extract the desired data and 'read' it into a usable format. Often, if not usually, such a data dictionary will be provided by the data source along with the raw data. It will then be up to you to alter the dictionary in order to extract, label and 'read in' the desired data.

If you need to write your own dictionary, the *data codebook* and the information it contains concerning the variables location in the string, their length, and description, and the general length of the record will be necessary. Alternatively, if you are just looking for location, length, label of the variables, often times the quickest sources of this information are the SPSS/SAS data definition files that usually come with the data files.

Below, I'll provide detailed instructions for importing and using data in Stata and some explanation of R, which is free and somewhat more flexible for programing purposes. For most manipulations and analyses of data, Stata is quite efficient to work with and relatively easy to use. In particular, Stata is excellent for merging files and obtaining quick statistics and graph. Stata's most obvious limitation is its ability to handle very large data sets. Both R and Stata bring the entire data set into active memory, and may thereby overwhelm your RAM.

If you need more space, you may have to go to another program such as SAS or Gauss. In the appendix, we have provided some instruction on Gauss, For a very helpful introduction to SAS, see the UCLA site: `http://www.ats.ucla.edu/stat/sas/`

Stata programming for extracting, formatting and labeling In general, the executable Stata file extension is "`*.do`". You execute the program by typing:

`do file_name` (or, alternatively) `run filename`

in the Stata Command window. It is helpful that Stata for windows displays a box with all the commands used in the past, and one with the name and description of the variables that are actually in your data set. You can also save the output printed on the screen to a log file by typing: `log using filename.log` and then use `log on/off`.

I will use an example to illustrate how to extract, label, and read in data.

Suppose we want to read and work with the PSID 1992 family data. So our raw data file is 92fam.txt. If we request it, the PSID webpage described below will provide us with a stata data dictionary along with this raw data file.

We want a data set with only the ID, age, sex of the head of the household, wage of the household head, and the family weight. We will find the location of these variable in the data dictionary or in the variable list file (if any) or in the SPSS/SAS data definition file, or in the codebook (that you will have to read anyway to get some information about how the variables are coded and defined, break in the series, missing variable values, etc....).

First, we obtain our raw data file and data dictionary or definition file. We rename the original files according to our preferences. Suppose the following:

a) the raw data file name is `92fam.txt` (these files often come with “`*.raw`” or “`*.dat`” as an extension);

b) the data files are in the directory: `i:\work\data;`

c) the program files are in `i:\work\prog.`

Now we build or adjust the `dictionary file` which will allow us to extract, label, and format the data. Use your preferred editor (wordpad, notepad, pico) to write dictionary and do files.

Note 1: The command `_column` indicates the column where the variable of interest starts. If you simply want to skip a few positions use`_skip.`

Note 2: Stata works from any directory. In order to change directory simply use DOS or Unix commands in the Stata prompt.

File: 92fam.dct

```
dictionary using i:\work\data\92fam.txt{
_lrecl(2347)
_column(4)
 long id92 %5f ''92 interview number''
_column(874)
 int age92 %2f ''92 age head of household''
 int sex92 %1f ''92 sex head of household''
_column(262)
 long Hw91 %6f ''head of household annual wage 91''
_column(2191)
 long He91 %6f ''head of the household total labor income 91''
_column(2342)
 long wgt92 %6f ''family weight 92''
}
end
```

We write a `do` file to execute this dictionary and read in the data. Alternatively, you can use Stata commands directly on the Stata prompt.

File: 92fam.do

```
infile using i:\work\prog\92fam.dct
sort id92
save i:\work\data\92fam.dta
describe
```

Then we type `do i:\work\prog\92fam.do` at the Stata prompt. Stata executes the program.

In this example, the data set will be sorted by id92. We also obtain a description of the dataset as output to the screen (and to the log file if this is on). Click edit to see the data. Note that we saved the data file in our data directory.

Some other useful commands:

`summary` gives summary statistics: variable name, number of observation, mean, standard deviation, min e max.

`describe` gives information about number of variables, tags, size etc.

`sort id92` sort the data by the variable id92.

To make comments in your code or log file: use *. Example: `*This file select id, sex,age etc`

Remark: Stata allows to use shorter names for the commands (example `sum` instead of summary, `des` for describe). Use the full length version of the command in `do` files to avoid problems.

See UCLA's Stata Learning modules for much more information on learning Stata
`http://www.ats.ucla.edu/stat/stata/modules/`

5.0.3 Stata Code Example: Extracting a subpopulation and building a table

Consider the following exercise:we want to build a double entry table by sex and age. We have three age groups: (20,30], (30,45], (45, 60]. We simply want to compute the average income, standard deviation and frequency for each cell. We write a little program to do that!

File: Table.do

```
*this program builds a double-entry table
*for income for men and women, and three age classes
use i:\work\data\92fam.dta, replace
set type long
drop if age92<20
drop if age92>65
gen agecat=recode(age92, 30, 45, 65)
```

```
label define agecat 30 ''20 to 30'' 45 ''30 to 45'' 65 ''45 to 65''
tabulate agecat sex92, summarize(Hw91)
save i:\work\data\92fam.dta
```

Stata prompt: `do i:\work\prog\Table.do`

Stata executes the program.

The output is shown on the screen. To obtain "weighted" statistics use `[aweight=wgt92]` after `summarize(Hw91)`. Note that Stata has three types of weights. For most, "aweight" (analytic weights) are going to correspond to the weights you download but you should read the Stata description and the data notes to be sure. Use `replace` in order to add the new variable `agecat` to the dataset and save the "updated" dataset. Without the `replace` option Stata will not overwrite the old data file with the new one. `gen` generates a new variable. `egen` generates a new variable that is defined as a special expression/functions of other observations. There is a list of these built in `egen` functions in the Stata help manual and they can be very useful. Notice the line `set type long`. The `set type` command selects type of the variable (`int`, `long`, etc.). You should use it every time you define a new variable. In fact, it can save a lot of workspace.

5.0.4 Merging

In general, extra care should be used when merging data files. It is easy to make a mistake so, when executing this function, try to check the process at every step to ensure it has performed as you wanted. It is probably wise to keep the original data set intact in case you make a mistake.

Suppose we want to merge the information in the `92fam.dta` file with some information about education contained in `92edu.dta`. We write a `do` file.

Important: both data files must be sorted by the variable we use to merge them!

FILE: MERGE.DO

```
*this program merges 92fam.dta and 92edu.dta
use i:\work\data \92fam.dta, replace
merge id92 using i:\work\data\92edu.dta
save i:\work\data\92fam.dta
summary _merge
```

Stata prompt: `do i:\work\prog\merge.do`

Stata executes the program.

In this case we are replacing the old data file `92fam.dta` with a new one that contains also the "merged" variables. Stata automatically also adds the variable `_merge` to the data set. This variable is extremely useful for future data handling. In fact,it can take one of the following values:

`_merge=1` the observation comes from the master datafile (i.e. the data in memory, in this case `92fam.dta`)

`_merge=2` the observation comes from the "using" data (in this case, `92edu.dta`)

`_merge=3` the observation is in both files.

If, for example, we want to use only the observation (i.e. the records) that are in both datafiles, we use the `_merge` variable. Example: `keep if _merge==3.`

`summary _merge`: gives you statistics about how many matches were found.

5.0.5 Some tips

Now suppose we want to keep or drop individuals with a high-school diploma:

`keep if edu92==12` or `drop if edu92==12`

Suppose we want to replace a missing value with the appropriate missing code:

```
replace edu92=99 if edu92==.
```

Suppose we want to count the number of observations satisfying a specific condition, say number of men with a BA:

```
count if edu92==16 & sex92==1
```

There is a procedure called `svy` that estimates mean and other statistics for survey data. Also the command to run a simple linear regression is very intuitive `(regress).` All these commands are very straightforward to use. If your data set is big and you do not have enough RAM to read it just type the `set mem xxx m` command on the stata prompt and then call for the data file again. This will expand the workspace to `xxx` megabytes.

5.1 Data extraction and manipulation in R

R can do most of what Stata does, but it's programming style is much closer to that of Matlab. The user-contributed packages are key advantage of R over any other statistical software. Many statisticians swear by this software and so they write and publish R routines to do the most high-tech procedures just as the papers explaining them are published. The software is also free and available on any platform from

`http://cran.r-project.org` For more intensive computing, it is on all of the CLA OIT's servers.

There is a graphical interface provided by R commander, but the command line ought to be sufficient most of the time.

Far more information on using R go to the manual: `http://cran.r-project.org/manuals.html`

5.1.1 Importing and subsetting

R can read a variety of formats, raw ASCII with the base package or almost any other package's output via the *foreign* package. To read a simple ASCII text file, with tab delimited data, use the command

```
data<-read.table(‘‘data_file.txt’’,header=TRUE)
```

Where, I assume that the first line of the file, the header, is made of variables names. If the data is stored as a .dta file, as from the PSID, it can be loaded by:

```
library(foreign)
data<-read.dta(‘‘data_file.dta’’)
```

This assigns to the matrix data all of the contents of the file `data_file.dta`. This creates a "data frame," which can be treated like a matrix with variables are stored in columns and each row is an observation. It also has some features that allow it to be referenced from within builtin R functions. You can have as many data frames open as you'd like (unlike Stata) and these can be merged by `merge(x,y)` either by column, which Stata calls "merge," or by row, which Stata calls "append."

To assign a useful name to one of these variables, you label it with `dimnames(data)[[2]] <- c(‘‘name 1’’, ‘‘name 2’’, ...)` You can assign column X a variable name by `varname <-data[,X]` or subset the data by creating logical vectors that satisfy the desired condition. For example, if we set up a column from `data` that was `sex`, we can create a vector and subset the data

```
men<-ifelse(sex==0,TRUE,FALSE)
data_men<-data[men,]
mean(data_men)
```

Notice the `ifelse` statement that creates the logical vector. This is one of my favorite features of R, and `ifelse` can be used in any number of circumstances beyond creating logical vectors. It loops through vectors to do the check described in its first argument, then writes the second argument if true and the third argument if false. In general it runs much faster than coding this loop by hand.

The final command `mean(data_men)` will print the conditional mean. If we instead wanted to store the value in a new scalar variable we would change this to `mean_men<-mean(data_men)` Finally, R can write the data back to a table with `write.table(data,‘‘file_name’’)`

A Gauss code

A.1 Extraction and labeling

Use `atog.exe` (or `atog386.exe` in Gauss for dos) utility. The file `Atog.exe` is in the gauss directory.

First, write a code using your favorite editor.

File: ext.cmd

```
input i:\work\data \92fam.txt;
output i:\work\data \92fam;
outtyp F;
invar record=2347
(4,5)   id92
(37,2)   age92
(39,1)   sex92
(262,6)   HWage91
(2342,6)   Wgt92;
```

Second, execute atog. An MsDos window will pop up. Write `ext.cmd` at the atog prompt, atog will execute the extraction program.

As a result, we obtain the same datafile we obtained with Stata. The file extension is "`*.dat`" (`92fam.dat`) but this time it is in Gauss format. Atog associates a `dht` file to the `dat` file with the same name (in this case `92fam.dht`) that is basically a sort of dictionary file.
Information on Atog utility is in the Gauss manual, Chapter 17

A.2 Extract subpopulation

Use the DataLoop commands in Gauss. When you use dataloop the translator must be on. In order to activate it in Gauss for windows go to `Options`, `Program` and click on `translator`. This operation will automatically activate it. Dataloop performs the same operations as the commands `keep, drop` in Stata. Also you can choose variables according to some criterion by using `select` or you can generate new variables.

Example: we want to select individuals with age between 20 and 65 and we want to keep only id, sex, and income.

File: dataloop

```
dataloop 92fam 92fam2;
select age92>20 and age92<65;
keep id92 sex92 Hwage92;
endata;
```

Run the program in gauss. `92fam2.dat` is your new file which contains information about the identification number, the gender, and the income for individuals in the age range [20,65].
Help on `DataLoop` is provided in the Gauss manual, chapter 11.

A set of useful procedures to work with the data is contained in the source file `Datatran.src.` In order to have more information about how these procedures are constructed simply print the file.
Remark: there may be some problems with Gauss in terms of the Gauss working directory. This problem concerns Gauss for windows (there are no such problems when you use the Dos version). It arises when you use the DataLoop procedure. In particular, Gauss may be unable to use files that are in a directory different from the c:\gauss one. Eventually you must change the configuration file `Gauss.cfg` (ask Ed).

B How to fetch and filter Citibase data from Eviews

This is very easy. First of all, you may want to check the excel file `citibase.xls` which is in the directory `G:\eviews3\citibase`. The `G` drive is usually the one where all the applications are (at least on most of the computers in the lab). This files is a list of all the variables included in the citibase, their name and their frequency. Once you find out the name of your variables you can go to Eviews. You can look for the variables directly in Eviews. Excel is better because it allows you to search for keywords. The procedure to fetch and filter the data series follows.

1 Open Eviews.

2 Do: `File-open-database`

3 Select the directory: `G:\Eviews3`

4 Select `EasyQuery`: this will provide the list of all the variable names, their definition etc. (you can choose which characteristics you want to be showed).

5 Choose the variable(s) of interest.

6 Click `Export`. Click OK. Choose the Workfile frequency (annual, quarterly etc.), the start and the end data for the series you are exporting. Click OK.Now you have your Eviews workfile with the serie(s) of interest..

7 At this point you can either HP-filter and work with the data series directly in Eviews or, alternatively, you can export the raw data and then filter and use them in your favorite language (Gauss, Fortran etc).

a) **Working in Eviews**

Starting from your Workfile.

i) Select `show.` Click OK. The series you selected are displayed on a worksheet.

ii) Select `Procs and Hodrick-Prescott Filter.` Now you only need to eventually change the name that you want for the filtered series and the smoothing parameter (although Eviews already selects the appropriate one for you). Click OK. The smoothed series is automatically added to your workfile and a graph is created with both the raw and the smoothed data.

b) **Working with other programs**

Export the data: Select the data series, go to `Procs/Export` and save the file in your favorite format. The Write Text-Lotus-Excel option allows you to choose the ASCII-Text delimiter etc. etc.

There are several versions of HP-filter codes written in Gauss, Matlab or Fortran on the web. In this paragraph I include a Gauss and a Matlab version. A very interesting home page that include general information on data and available codes in the IDEAS database `http://ideas.uqam.ca/QMRBC.`

An HP Filter for Matlab, Excel or Java can be downloaded from Kurt Annen `http://www.web-reg.de/index.h`

The following is an example (in Gauss) of how you can read, filter and provide some simple descriptive statistics for the data.

C Example: Gauss code to read and filters citibase data series.

```
   @ This program reads and filters the citibase series which are stored in the ASCII
file:  data.txt@


   library pgraph;
#include hp.txt; @include procedure for hp filter @
```

```
nobs=136; @ number of observations @
nser=4;
l=1600; @ smoothing parameter @
load matser[]=data.txt;
matser2=reshape(matser,nobs,nser);
matserf=hp(matser2,l);
corr_mat = corrx(matserf);
vcov=vcx(matserf);
/*print ''Correlation matrix '';
print '' Cons Output Invest Hours'';
print corr_mat;
stdev=vcov^(1/2);
print ''standard deviations '';
print '' Cons Output Invest Hours'';
print stdev;*/
c_raw = matser2[.,1];
c_fil = matserf[.,1];
periods = seqa(1,1,136);
title(''c raw'');
xy(periods,c_raw);
title(''c fil'');
xy(periods,c_fil);
stdevcy=sqrt(vcov[1,1])/sqrt(vcov[2,2]);
stdeviy=sqrt(vcov[3,3])/sqrt(vcov[2,2]);
stdevny=sqrt(vcov[4,4])/sqrt(vcov[2,2]);
print ''sd(c)/sd(y) sd(i)/sd(y) sd(n)/sd(y)'';
print stdevcy;
print stdeviy;
print stdevny;
print ''corr c,i,n & y'';
print corr_mat[1,2];
print corr_mat[2,2];
print corr_mat[3,2];
print corr_mat[4,2];
```

D A Gauss version of the HP-filter procedure:

HP - Hodrick - Prescott Filter by M. Watson, 1991 Usage: xp = HP(x,l) where: x = NxK matrix of K series of length N to be filtered; l = smoothness parameter (1600 for quarterly data); xp = NxK matrix of K series of length N representing the percentage deviations of each of the original series in x from their trend. Note: if you want the trend, change the next to last line to: retp(xt); if you want the smoothed series, use retp(xd).

```
*/
proc HP(x,l);
local j,xd,xt,i,k,ty,cy,n,xhp,q,a;
n=rows(x);
j=1;
a=zeros(n-2,n);
do until j>n-2;
a[j,j]=1;
if j+1 le n;
a[j,j+1]=-2;
endif;
if j+2 le n;
a[j,j+2]=1;
endif;
j=j+1;
endo;
i=eye(n);
xd=zeros(rows(x),cols(x));
xt=xd;
xhp=xt;
k=1;
do until k>cols(x);
ty=solpd(x[.,k],i+l*a'a);
cy=x[.,k]-ty;
xd[.,k]=cy;
xt[.,k]=ty;
k=k+1;
```

```
endo;
xhp=xd./xt*100;
retp(xhp);
endp;
```

A Matlab version of the HP-filter procedure:

```
    function [s]=hpfilter(y,w)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Author:  Ivailo Izvorski,
% Department of Economics
% Yale University.
% izvorski@econ.yale.edu
% This code has been used and seems to be free of error.
% However, it carries no explicit or implicit guarantee.
%
% function [s]=hpfilter(y,w)
% Hondrick Prescott filter where:
% w - smoothing parameter; w=1600 for quarterly data
% y - the original series that has to be smoothed
% s - the filtered series
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if size(y,1)<size(y,2)
y=y';
end
t=size(y,1);
a=6*w+1;
b=-4*w;
c=w;
d=[c,b,a];
d=ones(t,1)*d;
m=diag(d(:,3))+diag(d(1:t-1,2),1)+diag(d(1:t-1,2),-1);
m=m+diag(d(1:t-2,1),2)+diag(d(1:t-2,1),-2);
%
m(1,1)=1+w; m(1,2)=-2*w;
```

```
m(2,1)=-2*w; m(2,2)=5*w+1;
m(t-1,t-1)=5*w+1; m(t-1,t)=-2*w;
m(t,t-1)=-2*w; m(t,t)=1+w;
%
s=inv(m)*y;
```