```
STOP



*****************************************************.
***  Longitudinal Data Analysis for Social Science Researchers
**
**
** ESRC Researcher Development Initiative training programme:
**
**
**   Training materials lab 0:
**     INTRODUCTORY DATA ANALYSIS AND DATA MANAGEMENT IN STATA  .
**
**
**      www.longitudinal.stir.ac.uk
**      Paul Lambert / Vernon Gayle, 26 August 2007
*****************************************************.




*********************************************************.
*****************************************************.
** The file below comes in three sections:
***    Part 1: Introductory data management techniques
***    Part 2: Introductory data analysis techniques
***    Part 3: More data management: introducing file matching techniques
*********************************************************.



*****************************************************.
** GENERAL INSTRUCTIONS ON THESE FILES
**
** Work through this file in the interactive do-file editor, replicating
**  the STATA do-file commands. Further help on working with STATA is
**  available from the LDA web site.
**
**
*** This file assumes you have a number of files downloaded to your
**    machine. You will need the following:
*
** Downloadable from the LDA site :
** Data files (from ):
* -wemp_s2.dat
* -div3s2004.dat
* -ghs95.dta

** Macros :
* -seglabelsv1.do (downloadable from the LDA site)
* - spss2stata.sbs (downloadable from
*    http://www2.jura.uni-
hamburg.de/instkrim/kriminologie/Mitarbeiter/Enzmann/Software/Enzmann_Software.html
*     (used in exercise [1.1:iii(D)] on converting between SPSS and Stata)

** Downloadable from the UK Data Archive:
*  - ssa02.dta, ssa01.dta, ssa00.dta and ssa99.dta
*   (Scottish social attitudes 2002, 2001, 2000, 1999,
*     stata datasets for study numbers 4808, 4804, 4503, 4346, Stata format files)
*  - ssa02.por (Scottish social attitudes 2002, SPSS format file, study number 4808)
*
* -All BHPS Waves 1-15 component files in Stata format (UK Data Archive Study number
*   5151 (June 2007 release) (extracted from the zip file 5151STATA8.ZIP)
*      +warning - these are a large volume of files, 153 different files, ~ 700MB;
*      most of the exercise below does not use the BHPS data, so it could be more effective
*      to proceed without downloading this study at this point )
*
*
*************.
**
** These examples are written for Stata version 9.0 for Windows (2005)
**  - most commands should be compatible with earlier versions of Stata and other
```



```
**        operating systems
**  - for Stata version 8 or earlier, note that the length of this do file may
**      be too long for the Stata do file editor, in which case the do file should be
**      split into two or three components
**
*****************************************************.

** .


*****************************************************.
** NOTIFICATION OF FILE LOCATIONS / DIRECTORIES AND STATA SETUP
**
**
**
** i) File location declarations:
*** For the commands below to work, you should begin by running the following
**   macros, which tell Stata where to look for the relevant data files (mentioned
**   above) on your machine : .


global path1 "d:\lda\work\"
* (the location of your working directory -  where you will save
*    newly created data files and output) .


global path2 "d:\data\lda\"
* (the location of a folder where you have saved the
*   open access online data files mentioned above  -
*  see http://www.longitudinal.stir.ac.uk/workshop_materials.html)


global path3 "d:\data\bhps\w1to15\"
* (the location of a folder where you have saved the BHPS data
*   file(s) mentioned above) .


global path4 "d:\data\ssa\"
* (the location of your copies of the SSA data files mentioned above)



global path8 "d:\lda\macros\"
* (the location of your copy of the demonstration Stata macro 'seglabelsv1.do'
*   downloadable from the website )


global path9 "d:\temp\"
* (a location of a temporary folder where you can save intermediate files) .

**
**




** ii) Stata session management:

*** The commands below are used to set some general preferences within Stata,
**    it is usually good advice to run these but it is not essential

clear
* (clears any other data within Stata)

set more off
* (switches off the default setting whereby output is shown one page at a time only)
* (you should enter this manually via the command line to ensure that it is persistent)

set memory 64M
* (expands the memory allocated to the stata session, usually necessary if using large files)

capture log close
capture log using $path1\log_lab0.txt, replace text
* (These commands close any previous log file that may be open, then they set a new log file
*   for this lab, a plain text file where basic output is saved, located in the directory defined
*   as 'path1').
```

```
**
**
**
*******************************************************.
```

```
*******************************************************.
*******************************************************.
*** SOME COMMENTS ON USING STATA INTERACTIVE DO FILES  :
*
*
*
*
* Comment: This is an interactive command file ('do file'), you can run
*          it from the do editor window in Stata (open with 'ctrl-8'),
*          by highlighting the command line or lines which you wish to invoke
*          and either 'ctrl-d' or the 'do current file' icon
*
* Comment: On an interactive command file, you write out the textual Stata commands
*          and invoke them, typically in groups of a few sequential commands at a time.
*
* Comment: Gradually learning the textual Stata commands is the valuable skill in learning
*          Stata. The best way is to begin working through example files such as this one.
*          In time you will start to remember a few common commands, but additionally, you
*          should learn how and where to find information on the necessary Stata commands
*          - i.e. from the Stata manuals, the online Stata pages, or the help system
*
* Comment: Notes on the example files below give some explanation of what each command is
*          doing, but they don't cover everything. Best is to try to work out yourself
*          what each line is doing, but ask the instructors for clarification if needed.
*
* Comment: On an interactive command file, a new command is indicated by a line break
*          (i.e. line breaks are the 'command line delimiter'). A comment can be added to
*          the comand file by begining the line with a *, which means that Stata will ignore
*          the rest of the line - comments are useful to use to make notes for yourself.
*          Any text on a new line which doens't start with a * is interpreted by Stata
*          as an attempt to run the Stata command given by the text.
*          If you wish to spread the text of a single command
*          over more than one line, you can use the ///
*          symbol to indicate a line continuation
*
*          [advanced: if you run a whole file at once, as a 'batch file', it is also possible
*          to specify another symbol such as a ; as the delimiter]
*
*          [advanced: the reason 'STOP' is written at the top if this file is to prevent
*           you from inadvertently running the whole of this file in one go by pressing 'ctrl-r'
*           rather than 'ctrl-d' : there is no valid stata command called stop, so Stata will
*           immediately trip up on this command and won't carry on]
*
* Comment: If running a group of command lines in a go in Stata, if any of your
*          command lines include an error, Stata will immediately stop running
*          the lines from that point onwards (this is different from the way SPSS
*          processes command syntax). One useful way of averting this is to use the
*          'capture' prefix before any given command (eg used above on the 'log' example). This
*          handy prefix tells Stata to run the command if it is working fine, and to ignore
*          it if it has an error, but carry on running the next line
*
* Comment: Windows management. It is worth spending some time arranging your Stata windows
*          into a layout that suits you. You can change the font size and colour scheme
*          of the output window (right-click) and you can change the layout and colour schemes
*          for other windows. If like me you use 'Alt-tab' to toggle between windows, you will
*          notice that some poor programming by Stata means you need to double-tab to leave
*          the stata output window for any other window.
*
* Comment: For most social scientists, processing Stata through the do file editor is
*          appropriate for most purposes. The next step up is learning to program in STATA,
*          which concerns writing programmes (SPSS = macros) and batch files (whole files
*          processed in one go), to run complex tasks. Stata now refers to its programming
*           language as 'Mata'.
*
*
*   SEE ALSO : http://www.longitudinal.stir.ac.uk/Stata_support.html
*
*
```



```
*
*
*************************************************************
*************************************************************


** ..finally, here is the lab exercise...


*************************************************************
```

```
*******************************************************.
***    Part 1: Preliminary data management techniques
*******************************************************.

** Comment - In part 1 we open and close a number of different data files
**         - if this leads to some error messages, best advice is to start again from the top...
```

```
*******************************************************.
***** 1.1)  Data Entry :


**** i) Read in data from a plain text file:

infile case femp mune time und1 und5 age using $path2\wemp_s2.dat, clear
summarize
describe
list in 1/10
* note the absence of file information such as value labels


**** ii)  from a pre-prepared STATA format file:

use $path4\ssa02.dta, clear
summarize
describe
tab rural
list in 1/3
* STATA isn't especially good for interactive data reviewing,
* particularly if the file has quite a few variables or cases
*  - there's no variable view popup or window as in SPSS
* (Tip: with large STATA files, I often keep an SPSS version open
*  at the same time, solely for the purpose of examining the data)

use serial rsex rage marstat miles using $path4\ssa02.dta , clear
summarize
list in 1/3
* dealing with only a few variables is much easier



**** iii) Read in data from an SPSS file

* [some extended examples of different ways to get Stata format data from SPSS files]



*** A)  USE STAT-TRANSFER!!!

* This is a specialist software package for transferring between different
*   data analysis software formats
* - for SPSS to Stata transformations, it's the only way to preserve all file info,
*     such as variable and value labels)
```

```
*** B) From SPSS : write out SPSS data into plain text file; read into Stata as plain text

***  SPSS syntax (do the below in SPSS if available, changing path's as necessary):
** import file="d:\data\ssa\ssa02.por".
** write out="d:\temp\ssa02sub1.dat" / serial ' ' rsex ' ' rage ' ' marstat ' ' miles .
** execute.
*** Then STATA syntax :


infile serial rsex rage marstat miles using $path9\ssa02sub1.dat, clear

summarize
codebook
* all labels etc have been lost
* (this won't work unless you have run the SPSS commands above to generate ssa02sub1.dat

*** C) From SPSS : use SAS transport format to retain some extra variable detais:

*** First SPSS syntax :
** import file="d:\data\ssa\ssa02.por" .
** save translate outfile="d:\temp\ssa02sub02.xpt" /type=sas /version=x /keep=serial rsex rage
marstat miles.
*** Then STATA syntax :
fdause $path9\ssa02sub02.xpt , clear
summarize
codebook
tab miles
* this preserves variable labels, but not value labels
* (this won't work unless you have run the SPSS commands above to generate ssa02sub02.xpt



*** D) Use Dirk Enzemann's 'SPSS2Stata' conversion macro

** This is an SPSS script which exports the data from SPSS to SAS
** transport format, plus gets SPSS to write out further value label
** and details to another file which can also be processed by Stata -
** result is a full conversion, almost equivalent to STAT-Transfer

** The SPSS script can be downloaded from here ('SPSS2Stata'):
** http://www2.jura.uni-
hamburg.de/instkrim/kriminologie/Mitarbeiter/Enzmann/Software/Enzmann_Software.html
** (Frank Popham sent this link to us)




**** iv) Enter data to Stata by hand : .

*** A) Within an interactive batch file  :

** It's seldom worth it but for the simplest of data,
*  eg a small number of summary statistics :

clear
input england wales scotland nirel
49.14 2.90 5.06 1.69
end
graph bar (asis) england wales scotland nirel, title("UK population 2001")


** Instead, it's usually preferable to write the data into a text file,
**   and use 'insheet'

insheet using $path2\div3s2004.dat, clear
* (tip - look at the contents of the file in a plain text editor to see what it features)

summarize
list team points gscore glost
corr gscore glost points
graph bar (asis) points , ///
   over(team, sort(points) descending label(alternate) ) title("Division 3, 2003/4")
graph bar (asis) points gscore glost, ///
```



```
   over(team, sort(points) descending label(alternate) ) title("Division 3, 2003/4")
drop if (team=="East Stirling")
corr gscore glost points
* Defences, not forwards, win you 3rd division titles!


*** B) Interactively  :
clear
** - You can use the data editor window
** - Tap in values to cells; add labels etc by syntax as below
** - Cumbersome and best avoided - better to enter in SPSS or other package



**** v)  Saving out

** In Stata format:
insheet using $path2\div3s2004.dat, clear
summarize
list team points gscore glost
save $path9\div3s2004.dta , replace

** In plain text:
summarize
outfile team points using $path9\div3s2004_2.dat , replace




************************************************************
***** 1.2)  Handling variables :


infile case femp mune time und1 und5 age using $path2\wemp_s2.dat, clear
list in 1/10
* this data has no labels at present
**



**** i) To add variable labels :

tab femp mune
label variable femp "Wife's employment status"
label variable mune "Husband's employment status"
tab femp mune
**



**** ii)  To add value labels  :

label define fempl 0 "Not working" 1 "Employed"
label define munel 0 "Employed" 1 "Unemployed"
label values femp fempl
label values mune munel
tab femp mune
**



**** iii) Looking at categorical data :

tab femp
* that's no use - want to know what values are assigned
tab femp , nolabel
* that's rubbish too - now there's no labels
numlabel fempl , add
tab femp
numlabel fempl, remove
tab femp
* numlabel works, though complex
* A shortcut is to put numeric labels on everything:
tab femp mune
numlabel _all, add
```

```
tab femp mune
* (This is best. However, numlabel _all might not work for
*   large files with some string or complex variables).
**



**** iv)  Computing new variables

infile case femp mune time und1 und5 age using $path2\wemp_s2.dat, clear
summarize
gen age2=(age^2)
gen age3=(age^3)
graph matrix age age2 age3
* note the 'egen' command is similar to, but has more options than, 'gen'

gen age3=(age*3)
* this won't work - you can't overwrite an existing variable with generate
drop age3
gen age3=(age*3)
graph matrix age age2 age3



*****  v) Recoding values

infile case femp mune time und1 und5 age using $path2\wemp_s2.dat, clear
histogram age
gen age4=age
recode age4 18/30=1 31/40=2 41/50=3 51/max =4
label define age4l 1 "18-30 yrs" 2 "31-40 yrs" ///
      3 "41-50 yrs" 4 "51+ yrs"
label values age4 age4l
tab age4

* Note the use of /// as a line continuation declaration
* Tip - with complex data recoding (eg occupational unit codes), you can recode via
* a macro in a separate do file (aka an 'ado' file)
**



****  vi)  Dealing with missing values

use serial rsex rage marstat miles using $path4\ssa02.dta , clear
numlabel _all, add
summarize
tab miles

* codes for -1, 6 and 8 are all possible missing values

*** a) Keep the data but code it as missing
tab miles
mvdecode miles, mv(-1,6,8)
tab miles
mvencode miles, mv (-999)
tab miles
* note - you loose differentiation between values -1, 6 and 8

*** b) Drop the missing data
histogram miles
histogram miles if (miles >= 1 & miles <= 5)
summarize
drop if (miles==-999)
summarize

*** c) What about a specific cases - say we want to drop serial 40007
sort serial
list serial in 1/10
drop if (serial == 40007)
list serial in 1/10
summarize

** Tip - never save data to same filename after manipulating data and variables like this!!

************************************************************
```



```
************************************************************
***** 1.3)  Subsample operations :


* Stata has a very concise syntax for dealing with subsamples:

use pid lxrwght lxewght llrwght llewght lregion lsex lage lvote ///
     using $path3\lindresp.dta , clear
summarize
numlabel _all, add
tab lregion
graph pie, over(lregion) line(lcolor(gs1))


**** i) Analysis only on certain conditions :

tab lvote if lregion==18
tab lvote lregion if ((lregion==17 | lregion==18) & lvote >= 1 & lvote <=10), col
tab lvote if (lregion==17 & lsex==1 & lage <= 40)

**

**** ii) Analysis split by another variable :

sort lsex
by lsex: tab lvote
by lsex: tab lvote if (lregion==18 & lage >= 50)
by lsex: tab lvote if (lregion==18 & lage >= 50) [aweight=lxrwght]


**** iii) Analysis only for certain range of data:

tab lvote
tab lvote if (lvote >= 1 & lvote <= 3)
sort lsex
by lsex: tab lvote ///
   if (lregion==18 & lage >= 50 & (lvote >= 1 & lvote <= 3)) [aweight=lxrwght]


mvdecode lvote, mv(-9,-7,8,10,11)
tab lvote lsex , col chi V
* (an insignificant significance?)

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


************************************************************
***** 1.4)  Weighting data :


use pid lxrwght lxewght llrwght llewght lxrwtuk1 lxrwtuk2 lregion lsex lage ///
   using $path3\lindresp.dta , clear
summarize
numlabel _all, add
tab lregion
** This is the BHPS wave 12 (2002), with only some variables extracted

summarize lxrwght lxewght llrwght llewght lxrwtuk1 lxrwtuk2
* These are 6 of the main BHPS weighting variables

* Lxrwght is the main weight, for adjusting a given wave to the British population:
tab lregion , summarize(lxrwght) mean obs
tab lregion [aweight=lxrwght]

*Graph this (using a scaling factor ease interpretation)
gen lxr2=lxrwght*1000
graph  bar (mean) lxr2 (count) lxrwght, over(lregion, label(angle(70)) ) ///
     legend(order(1 2) label(1 "Cross-sectional weight") label(2 "Unweighted cases") )

* For a cross-sectional weight including cases from Northern Ireland, use lxrwtuk1
tab lregion, summarize(lxrwtuk1) mean obs
tab lregion [aweight=lxrwtuk1]
```

```
gen luk2=lxrwtuk1*1000
graph  bar (mean) luk2 (count) lxrwght, over(lregion, label(angle(70)) ) ///
     legend(order(1 2) label(1 "Cross-sectional UK weight") label(2 "Unweighted cases") )

* The uk2 cross-sectional weights are for allowing representative analysis within
* countries, ignoring cross-country divisions:
tab lregion, summarize(lxrwtuk2) mean obs
tab lregion [aweight=lxrwtuk2]

* Other weights involve considering other longitudinal structures
table lsex, c(mean lxrwght n lxrwght mean lxewght n lxewght)
*  note this shows: m lower initial non-response than f, but
*    m/f _dropout_ rates are more similar



** To use weights: with most commands, add an 'aweight' :

tab lsex
tab lsex [aweight=lxrwght]

* The [aweight=..] works in most situations
* comment: STATA has several types of weighting formula.
* aweight is the normal sampling weight for population fractions
** BUT aweight sometimes can't be used in certain commands
** AND other weight commands can, but the might not accept noninteger weights
tab lsex [fweight=lxrwght]
* ..see..




************************************************************
***** 1.5)  Including survey data cluster effects :


* Stata has a very useful set of options for estimating survey data
*  design effects (it's well ahead of alternative general purpose packages)


** Example : BHPS wave 12 .

** The BHPS has a multistage cluster design, some possible survey
**  effects :

**  wstrata (in whhsamp): initial sample stratifying factor
**  whhac (in whhsamp): household location sampling point (w1 only)
**  wpsu (in whhsamp and others): local region of household
**  whid (in windresp and others): household identifier


** Stata allows for two cluster effects ('strata' and 'psu')
** -> If studying households, it would be best to use wstrata and whhac
** -> If studying people, household clustering may be important, so
**    better to use wpsu and whid :


use pid lxrwght lxewght llrwght llewght lregion lsex lage lvote ///
       lhid   using $path3\lindresp.dta , clear
summarize lregion   lhid
numlabel _all, add
tab lregion

**  STATA's 'svy' commands allow survey structre to be recognised
svydes
*(nothing defined yet)
mvdecode lvote , mv(-9,-7,11)
table lvote , c(mean lage n lage)
sort lvote
ci lage [aweight=llrwght], by(lvote)
* Now add the suvey definitions:
svyset [pweight=llrwght], strata(lregion) psu(lhid)
svydes
```



```
mean lage
svy: mean lage

mean lage, over(lvote)
svy: mean lage, over(lvote)

** Comment: point estimates are unaffected, but CI's have widened
** Good practice is always to check for these sort of effects
**    (..and then decide to ignore them...)

** Comment: lregion isn't the best strata variable to use on the BHPS, but was the
** most convenient for this illustrative data

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




************************************************************
*****************************************************.

* 1.6) Handling results of models / calculations - see also section 2.3(iii)


** After any given statistical model, Stata temporarily stores information on the model
*  in memory, which the user may extract and store elsewhere. The information
*  is stored as the model 'estimates' and coefficient matrices:

estimates clear
*[prepare data]
use $path2\ghs95.dta, clear
numlabel _all, add
tab soclase
gen class56=(soclase==5 | soclase==6)
tab class56
tab sex
gen fem=(sex==2)
tab fem
gen age2=age^2
summarize soclase class56 age age2 workhrs fem
* [now ready to model these variables:]

logit class56 age age2 workhrs fem if (soclase > 1 & soclase <=6)
ereturn list
* (this gives everything saved for the model)
matrix logcoef=e(b)
est store logit1
matrix list logcoef
matrix agecoef=logcoef[1,1..2]
matrix list agecoef

est stats
est table , star stats(n r2_p bic)


* Compare three models for occupational attainment
est clear
logit class56 age age2 workhrs fem if (soclase >= 1 & soclase <=6)
est store logit1
ologit soclase age age2 workhrs fem if (soclase >= 1 & soclase <=6)
est store ologit1
regress soclase age age2 workhrs fem if (soclase >= 1 & soclase <=6)
est store regress1

est stats
est table logit1 ologit1 regress1 , b(%9.3f) stats(N r2 r2_p ll ) star

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




************************************************************
****************************************************.
```

```
* 1.7) Processing do and ado files


** It is ofen helpful to call upon a longer piece of Stata programme
**  within the course of your do-file editor session.
** This can be done by calling a 'do' or an 'ado' file within your session
*   (do and ado files are equivalent in their treatment by Stata, but the
*     latter term is often used to differentiate them from interactive files)

** Example:

use $path2\ghs95.dta, clear
numlabel _all, add
tab segead
** This is the variable 'SEG' - a coding for the job classification of any
*   working adult in the household

gen seg2=segead
tab seg2
* Note that seg2 doesn't preserve the value labels, which need to be added in
*  some way.
** The most orthodox way of adding the data would be to write out all the labels:

label define seg2l  1 "Employers in large establishments (25+ employees)" ///
     2 "Managers in large establishments (25+ employees" ///
     3 "Employers in small establishments (lt 25 employees)" ///
     4 "Managers in small establishments (lt 25 employees)" ///
     5 "Professional workers - self-employed" ///
     6 "Professional workers - employees" ///
     7 "Non manual workers, ancillary to professions" ///
     8 "Foremen and supervisors of non-manual workers"  ///
     9 "Junior non-manual" ///
     10 "Personal services workers" ///
     11 "Foremen and supervisors of manual workers" ///
     12 "Skilled manual"   ///
     13 "Semi-skilled manual" ///
     14 "Unskiled manual"  ///
     15 "Own account workers, non-professional" ///
     16 "Farmers - employers and managers" ///
     17 "Farmers - own account workers"  ///
     18 "Agricultural workers" ///
     19 "Armed forces" ///
     20 "Full time student"  ///
     21 "Never worked" ///
     22 "Inadequately described occupation"
label values seg2 seg2l
numlabel _all, add
tab seg2

** However, writing out all of these labels is cumbersome within the
*  single do file.
*  By running a separate file as a command, we could put these details
*   elsewhere and invoke them with a single command - eg:

gen seg3=segead
tab seg3
do $path8\seglabelsv1.do
label values seg3 segl
numlabel _all, add
tab seg3

** In practice do and ado files are often invoked within programmes to run
* quite extended and complex processes

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




************************************************************.
```



```
*****************************************************.
** Part 2: INTRODUCTORY DATA ANALYSIS TECHNIQUES
*****************************************************.


*****************************************************.
* Comment: Part 2 uses just the single data file called
*          'ghs95.dta, which should be saved on your machine and located
*          within the folder you've defined as 'path2'

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



************************************************************.
************************************************************.
** 2.1) SELECTED UNIVARIATE TECHNIQUES
************************************************************.

use $path2\ghs95.dta, clear
numlabel _all, add


** Categorical data :
tab sex

tab typaccm

list sex typaccm in 1/40


** Some graphs:

gen one=1
label variable one "Respondents"
graph bar (count) one, over (typaccm)
graph pie one, over(typaccm)

** Comment: there are numerous specifications which may be added to Stata graph
*      outputs to adjust their appearance, eg
graph bar (count) one, over (typaccm, label(alternate labsize(vsmall)))  ///
     ytitle(" ") title("UK Housing tenure 1995") ///
     note(" " "Source: General Household Survey 1995, n=4633")


** Metric data :.
summarize workhrs
list workhrs in 1/40
histogram workhrs
graph box workhrs
ci workhrs



************************************************************.
************************************************************.
** 2.2) SELECTED BIVARIATE TECHNIQUES
************************************************************.


** Two categorical variables
tab soclase genhlth
tab soclase genhlth, row V
tab soclase genhlth, col chi V gamma



** Two metric variables
summarize age earnings
gen lnearn=ln(earnings) if (earnings >= 10 & earnings <= 5000)
summarize age earnings lnearn
graph twoway scatter lnearn age
correlate age lnearn
regress lnearn age
gen age2=age^2
regress lnearn age age2
```

```
* (Aside - what is the shape of the effect of age?)
matrix coefs=e(b)
matrix list coefs
gen ageef=age*coefs[1,1] + age2*coefs[1,2]
graph twoway (scatter ageef age)


** One categorical variable, one metric
table soclase, c(mean lnearn sd lnearn n lnearn)
sort soclase
ci lnearn, by(soclase)
graph box lnearn, over(soclase, label(alternate))




************************************************************.
************************************************************.
** 2.3) SELECTED MULTIVARIATE TECHNIQUES
************************************************************.

*****************************************************.
*** i) Multivariate comparisons
** (Stata is very good indeed for multivariate comparisons)

summarize lnearn workhrs age age2
tab sex
tab soclase
tab genhlth


** If no variables are metric:.

sort sex
by sex: tab soclase genhlth, col chi V gamma



** If one variable only is metric : .
table soclase sex if (soclase >= 1 & soclase <=6), ///
      c(mean lnearn sd lnearn n lnearn)
sort sex
by sex: table soclase if (soclase >= 1 & soclase <=6), ///
      c(mean lnearn sd lnearn n lnearn)

graph bar (mean) lnearn, over(sex) over(soclase, label(alternate)) asyvars
graph bar (mean) lnearn, over(sex) over(soclase, label(alternate)) asyvars  stack

graph box lnearn, over(sex) over(soclase, label(alternate)) asyvars

** If 2+ variables are metric
graph twoway scatter lnearn age
correlate age lnearn workhrs
graph matrix age lnearn workhrs


*****************************************************.
*** ii) Multivariate modelling
**  (Stata has a huge range of modelling options)

summarize lnearn workhrs age age2

regress lnearn workhrs age age2

* To include some categorical variables, you can create some dummy variables
*  (see section 1 for variable constructions)

tab sex
gen fem=(sex==2)
tab soclase
gen class1=(soclase==1)
gen class2=(soclase==2)
gen class3=(soclase==3)
gen class4=(soclase==4)
gen class5=(soclase==5)
gen class6=(soclase==6)
gen class7=(soclase==7)
```



```
gen class12=(soclase==1 | soclase==2)
* (the above is the manual way to create dummy variable indicators)
tab region
gen england=(region >= 1 & region <= 15)
gen scotland=(region >= 18 & region <= 22)
gen wales=(region==16 | region==17)


regress lnearn age age2 workhrs fem
regress lnearn age age2 workhrs fem class1 class2 class4 class5 class6 class7

* categorical variables can alternatively be handled using 'xi' regressions
xi:regres lnearn age age2 workhrs i.sex
xi:regres lnearn age age2 workhrs i.sex i.soclase

list sex soclase _Is* in 20/50
* (this illustrates how xi has created some new dummy indicators automatically).

* Interaction terms can also be entered either by creating new variables, or xi:
gen agefem=age*fem
regress lnearn age age2 workhrs fem agefem

xi:regress lnearn age age2 workhrs i.sex i.sex*age

* Repeated regressions across subgroups:
sort sex
by sex: regress lnearn age age2 workhrs class1 class2 class4 class5 class6 class7


** Regressions with categorical outcomes:

tab class12
logistic class12 age age2 workhrs fem

tab soclase
mlogit soclase age age2 workhrs fem if (soclase > 0 & soclase <=6), baseoutcome(1)
ologit soclase age age2 workhrs fem if (soclase > 0 & soclase <=6)

** Comment: use 'baseoutcome' on mlogit to specify the reference category.
**          in older versions of Stata, 'basecategory' was used.

mlogit soclase fem if (soclase > 0 & soclase <=2), baseoutcome (1)
ologit soclase fem if (soclase > 0 & soclase <=2)
** Comment : compare the differences between these ologit/mlogits with 2 categories
*            above, with the 6 category version when ordinality mattered more


** Comment: note how regression by default uses listwise deletion of missing values
**            good practice can be to make this explicit in your own programming:
summarize lnearn workhrs age age2
regress lnearn workhrs age age2
regress lnearn workhrs age age2 if (lnearn >= 1 & lnearn <= 10 ///
       & age >= 16 & age <= 100 & workhrs >= 1 & workhrs <= 100)
* The latter format takes longer to write out, but is ultimately more reliable)



**********************.
***** iii) Summarising model outputs:
** an impressive Stata functionality covers analysing and storing the results of
*   one or more regression models (see also section 1.6 above):

regress lnearn age age2 workhrs fem scotland wales


** i) testing coefficient differences
test scotland=wales
* (for last active statistical model, tests is coefficient for Scotland is
*  significantly different to Wales - here it isn't)

** ii) impacts of coefficient effects
display 35^2
lincom _cons + 35*age + 1225*age2 + 40*workhrs + 0*fem + 1*scotland
lincom _cons + 35*age + 1225*age2 + 40*workhrs + 1*fem
** Lincom gives you the effects of the coefficients for combinations of values:
*  Predicted log income in 1995 for a 35 year old Scottish male working 40 hours
*  a week is 5.62 (276pw); for a 35 year old English female working 40 hours it
```

```
*  is 5.36 (213pw)

** iii) comparing different model results
est clear
regress lnearn
est store null
regress lnearn age age2 workhrs
est store mod1
regress lnearn age age2 workhrs fem
est store mod2
regress lnearn age age2 workhrs fem scotland wales
est store mod3
xi: regress lnearn age age2 workhrs fem i.fem*scotland i.fem*wales
est store mod4

est stats
est table null mod1 mod2 mod3 mod4, stats(N r2 bic)
est table mod1 mod2 mod3 mod4, stats(N r2 bic) star

est for mod1: adjust, by(age fem)
* (predicted values for combinations of age and gender)

*************************************************.
*************************************************.
*************************************************.









*************************************************************.
***************************************************.
** Part 3: More data management: introducing file matching techniques
*************************************************************.


***  COMBINING DATASETS


*****************************************************
*****************************************************
***  3.1)   Adding Files (Eg, Repeated Cross-Sections; Panels)
*****************************************************


**** EXAMPLE : REPEATED CROSS-SECTION : POOLING DIFFERENT YEARS
****           FROM THE SCOTTISH SOCIAL ATTITUDES SURVEY


** The commands below open up 4 years of the Scottish Social
**  Attitudes survey, save a selection of variables, then add
**  them all together

use serial rsex rage marstat scotpar2 ukintnat wtfactor using $path4\ssa02.dta , clear
gen year=2002
save $path9\m1.dta, replace

use serial rsex rage marstat scotpar2 ukintnat wtfactor using $path4\ssa01.dta , clear
gen year=2001
save $path9\m2.dta, replace

use serial rsex rage marstat scotpar2 ukintnat wtfactor using $path4\ssa00.dta , clear
gen year=2000
save $path9\m3.dta, replace
```



```
use serialno rsex rage marstat scotpar2 ukintnat wtfactor using $path4\ssa99.dta , clear
gen year=1999
gen serial=serialno
drop serialno
save $path9\m4.dta, replace

* Comment: all these variables are collected and equivalent between
*   the various survey years; beware: it usually takes quite a
*    lot of effort to pick out appropriate variables like this

** Add them all together :

use $path9\m1.dta, clear
append using $path9\m2.dta
append using $path9\m3.dta
append using $path9\m4.dta
tab year
graph box scotpar2, over(rsex) over(year, label(alternate)) asyvars title("Opposition to independent
/ devolved Scotland")

summarize
save $path1\ssa1.dta , replace


** We've pooled data on variables that are equivalent in each ssa
*  sweep - there aren't many choices though - a lot of variables are
*  not harmonised between even just these 4 years of surveys
** Often, an analyst would compute their own harmonised variables
*  using the same variable names across years.



*************************************************************
*****************************************************
***  3.2)   One-to-One Matching (Eg, Studying transitions)
*************************************************************


***********************************************
**** EXAMPLE: MATCHING TWO BITS OF INFO FROM DIFFERENT TIME POINTS
**             TO THE SAME SUBJECTS


****   Link data from the 1994 BHPS youth samples (11-15yrs)
****    with data from the 2005 adult response (22-26 yrs) .

** 1994 youth records.
use pid dypcomp dypsex using $path3\dyouth.dta , clear
sort pid
summarize
save $path9\mtch1.dta, replace


use pid oqfedhi osex oage using $path3\oindresp.dta, clear
sort pid
merge pid using $path9\mtch1.dta
tab _merge

** The merge data here shows :
** 15,180 adults in wave o (2005) but not in d youth
** 326 kids in d youth but not in 2005 adult sample ave
** 447 are kids in d youth and also present in 2005 adult


tab dypsex osex
keep if (_merge==3)
tab oage

numlabel _all, add
tab oqfedhi
tab dypcomp
mvdecode oqfedhi , mv(-9,-7)
mvdecode dypcomp, mv(-9)
tab oqfedhi dypcomp, col
* The quals profile of comp users seems higher
```

```
gen educ3=oqfedhi
recode educ3 1/4=1 6=2 7/13=3 *=-9
tab educ3
mvdecode educ3, mv(-9)
tab educ3 dypcomp, col V




*******************************************************
*******************************************************
***  3.3)   One-to-Many Matching (Eg, Fixed in time information)
*******************************************************



*** Egs : - regional average distributed to all people in region
***       - household info distributed to all household members
***       - Individual fixed data distributed to multiple person records


** The BHPS has multiple records per person, but some of it's data
**  is fixed in time, and thus could be distributed to several
**  different cases at once

** Example : household level details distributed to individual level data
**       from the BHPS


use $path3\ohhresp.dta, clear
summarize ohid ocd8use
numlabel _all, add
tab ocd8use
* This is one record per BHPS household
sort ohid
keep ohid ocd8use
save $path9\mtch1.dta, replace

use pid ohid osex oage using $path3\oindresp.dta, clear
summarize
** This is one record per BHPS person

*Do a one-to-many match :
sort ohid
merge ohid using $path9\mtch1.dta
tab _merge
* 1 = BHPS records who had no valid household level info (no valid cases)
* 2 = 6 cases = records with household info but no individual record
* 3 = 15627 = records with both household and individual info successfully linked
summarize
numlabel _all, add
tab ocd8use
table ocd8use if (ocd8use >= 0 & oage >=16), c(mean oage n oage)
graph box oage if (ocd8use >= 0 & oage >= 16) , over(ocd8use)




*******************************************************
*******************************************************
***  3.4)   Many-to-One Matching (Eg, Creating time series')
**            (aka: Agregating)
*******************************************************


**** Examples : compute average income per sampling area and link back
****          : find modal vote per sampling area and link back
****          : find highest educational level in household and link back



** BHPS Illustration: highest education level in household, BHPS wave 12

use $path3\lindresp.dta, clear
```



```
numlabel _all, add
tab lqfedhi
keep if (lqfedhi > 0 & lqfedhi < 13)
gen qual5=lqfedhi
recode qual5 1=1 2=2 3/5=3  6/11=4 12=5
label define qual5l 1 "Higher degree" 2 "Degree" 3 "Diploma" 4 "School or vocational" 5 "No qual"
label values qual5 qual5l
numlabel qual5l, add
tab qual5

summarize pid lhid qual5
* lhid is within wave household identifier, 138832 recodes

collapse (min) qual5,  by(lhid)
summarize
* This has created a dataset where each record is a household per
*  year, and qual5 is the highest qualification held by anyone in
*  the household

sort lhid
rename qual5 hhqual5
label variable hhqual5 "Highest qual held in household"
save $path9\mtch1.dta, replace

** Now match that information back to the panel file :
use $path3\lindresp.dta, clear
numlabel _all, add
tab lqfedhi
keep if (lqfedhi > 0 & lqfedhi < 13)
gen qual5=lqfedhi
recode qual5 1=1 2=2 3/5=3  6/11=4 12=5
label define qual5l 1 "Higher degree" 2 "Degree" 3 "Diploma" 4 "School or vocational" 5 "No qual"
label values qual5 qual5l
numlabel qual5l, add
tab qual5

sort lhid
merge lhid using $path9\mtch1.dta
tab _merge
* Because we treated it beforehand, we get a perfect fit - all
*  cases get a household average linked to them (_merge==3)
tab qual5 hhqual5


* Q: How well is voting explained by own and by household education?
tab lvote
keep if (lvote==1 | lvote==2)
tab lvote qual5, row V
tab lvote hhqual5, row V

* (only for those who are different to hhld value)
tab lvote qual5 if (qual5 ~=hhqual5), row V
tab lvote hhqual5 if (qual5 ~=hhqual5), row V




*******************************************************
***  3.5)   Relationships-between-cases Matching


**** Example : - BHPS, Match a spouse's job to own case
****             (some other BHPS matching examples are found in lab 2)



*** Link spouse's job status to an individual record .

use ljbstat lsppid using $path3\lindresp.dta , clear
rename lsppid pid
rename ljbstat spjstat
label variable spjstat "Spouse's job status"
sort pid
```

```
summarize
save $path9\mtch1.sav , replace

use pid ljbstat lsppid  lmastat lsex using $path3\lindresp.dta, clear
sort pid
merge pid using $path9\mtch1.sav
tab _merge
** _merge=1 : 6825 cases on the current file without spouses (the same 6825)
** _merge=2 : 6825 incoming cases without spouses
** _merge=3 : 9774 cases with spouse's data matched

tab lmastat _merge
* (ie lmastat is only valid for merge=1 or 3

tab ljbstat spjstat if (_merge==3)
sort lsex
by lsex: tab ljbstat spjstat if (_merge==3)
by lsex: tab spjstat ljbstat if (_merge==3 & (ljbstat==2 | ljbstat==3)), col

** These tables show relation between own job status and spouse's


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




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

capture log close
* closes the log file


**** EOF
```