Introduction to Stata[*]

Jeroen Weesie
Dept of Sociology
Utrecht University

Jan 2000

comments are welcome!

Abstract

In this short report, I provide a concise introduction to using the statistical program Stata (version 6). The audience are researchers well-versed into using some other statistical software such as SPSS. The transition between SPSS and Stata is given some attention, e.g., via a discussion of the translation of datasets and a series of tables listing Stata commands and their SPSS equivalents.

1 Introduction

Stata is a modern and general command-driven package for statistical analyses, data management, and graphics. Versions are available for PC/DOS/Windows, Mac, and a number of UNIX systems. Below you find a brief review of some of the key elements of Stata, a sample session, and a few table describing some of the more important Stata commands with their SPSS-equivalent (if appropriate) and a brief explanation of their purpose. The appendix includes a more elaborate survey of the part of Stata that deals with survival time and panel data.

Starting Stata Under Windows (Windows 95, Windows NT) you start by clicking on the Stata icon on your desktop or via the menu system that is opened if you click the Start botton on the left-under corner of your screen. Under Dos or Unix you start Stata by entering at the shell-prompt the command `stata`. Stata will start up, display a header, and show the Stata-prompt `.`, the period. Stata is now ready for your first command.

Remark 1. The first time you start Stata you should type the command `verinst` to have Stata verified that she is well installed.

Remark 2. Commands that you type each time you enter Stata are best entered in a specifc file (e.g., `profile.do`), that you let execute automatically whenever you start Stata. Edit the properties of Stata to do so.

Remark 3. Part of Stata can also be run via a menu system. You can download this system, called `StataQuest` from the Stata web site `www.stata.com`. While the first steps in Stata may be easier using this menu system, the power of Stata will remain hidden from you. For more serious work, nothing beats a command language. Once you know this language, you probably don't want to go back to a menu system.

[*] This introduction to Stata is derived from an early version written by Albert Verbeek, one of the founding fathers of the ICS.

Entering and editing commands Commands are entered and editted via the keyboard. Previous commands are saved in a buffer, and can be restored for editting. We list the most useful editing commands for the DOS/Windows version and for the UNIX version.

command	DOS/Windows	UNIX
retrieves previous command	PgUp	Ctrl-R
next command	PgDn	Ctrl-B
cursor back	←	Ctrl-H
cursor forward	→	Ctrl-L
move cursor to start of line	Home	Ctrl-K
move cursor to end of line	End	Ctrl-P
deletes char to the left	Backspace	Backspace
deletes char at cursor position	Del	Ctrl-D
delete to end-of-line	Ctrl-End	Ctrl-X
delete full line	Esc	Ctrl-U
toggles insert mode	Ins	Ctrl-E
execute command	Enter	Enter

To list the previous 10 commands, type `#review 10`.

Interrupt Stata One may interrupt lenghty Stata commands (e.g., a `list` of many observations, or a lengthy computation) with the interrupt-command *Ctrl-Break* or the Stop-button. (Note: if a command generates lengthy output that one does not want to see, type `quietly` in front of the command.)

Exit from Stata To exit Stata, you issue the command `exit`. If you worked on a dataset, you probably made changes to the data, e.g., you created new variables. If you didn't first save your data, Stata will refuse to let you exit. This is a somewhat paternalistic method to protect you from your own sloppy-ness. You can exit Stata without saving the data by typing `exit, clear`. It is also possible to exit via the `File` menu.

help The F1 key is reserved by Stata for `help` (DOS, UNIX); under Windows a help menu is available. This the help function, you can get detailed information about most aspects and commands, including examples how the commands can be used. E.g., `help regress` gives information about (the many options of) thet commando `regress`. `help help` gives information about the interactive help system. It is advisable to first go to the examples section.

search Use `search` *`topic`* to search the Stata command for analyses w.r.t. *topic*. For instance, `search regression` gives a compact survey of the commands relevant for regression analysis. Note: at the end of the help section of commands, you'll also find a list of related commands. Under Windows, these help topics are hyper-linked.

The command `lookfor` name makes Stata search in the names and labels of variables for *name*. For instance, `lookfor edu` searches for variabels that presumably involve *education*.

Stata on the Internet Stata's internet set *www.stata.com* can be accesses from within Stata via the Help menu. On the site, you can find an extensive FAQ (Frequently Asked Questions), access to archives of user-contributed Stata programs, and links to other statistical software providers. In addition, Stata can "update itself", i.e., apply bug fixes that are frequently made available. The updates are applied by typing the command `update all`, and by following the instructions.

Tutorials The command `tutorial` starts a somewhat interactive series of tutorials (a tutorial takes beteen one quarter and two hours) to learn some of the Stata facilities in specific areas, for instance, in regression analayis, graphics, logistic regression, and survival time analysis.

Identifiers An identifier ('name'), such as the name of a command or variable, consists of maximal 8 characters (both lowercase and uppercase letters, digits and the underscore), where the first character should *preferrably* be a letter. Stata is *case-sensitive*, i.e., Stata distinguishes between lowercase and uppercase. Almost all Stata commands are in lowercase.

Abbreviation A nearly general rule in Stata is that you may abbreviate commands and variable as long as Stata may not become confused as to what you mean. For instance, if you have variables `income1` and `inkvar2` in your data, Stata will understand that `inc` is the variable `income1` bedoelt, while Stata would not be able to decide whether `in` means `income1` or `inkvar1`, and so display an error message ("ambiguous abbreviation") and stop. If you really mean to specify all variables that start with `in`, you can use a widcard expression (`in*`).

log-files The command `log using` *`filename`* specifies that all commands that are entered from the keyboard worden en most of the oruput that is produced as a result, are saved in a file named *filename.log*. In this way, you can save output (in ASCII format) and have it printed.

batch-files One may issue any command to DOS by prefixing it with an '`!`'.

To build and test a file of Stata commands `xyz.do` use an ASCII editor (ed). Under Windoes, e.g., `notepad`, under DOS: `edit`, under UNIX `vi` or `pico`. One may edit the file *without leaving Stata* by typing `!ed xyz.do`; under Windows one may simp[le open a window with an ASCII editor. After leaving the editor, one will return to Stata and one may type `do xyz` to 'run' the commands.

StatTransfer and DBMS-Copy are software tools (unrelated to Stata) that can *translate* the 'system file' from the format used by one statistical program to that of another program. The translation includes variable and value labels, missing values etc. These programs support, a.o., SPSS, SAS, S-Plus, and Stata. StatTransfer is available at the ICS/Utrecht.

2 A sample session

Below you find a short introductionary session in Stata, using the `unemploy` data of Tazelaar and Sprengers. You should work through the session yourself behind a computer running Stata.

```
. * A copy of all commands and output will be written to the file ch1_1.log.
. * The option -replace- of the command -log- is written after the comma.
. * It specifies that Stata may overwrite the file if already exists. This
. * is an illustration how Stata tries to protect you from accidental
. * destruction of valuable information (files, variables, etc).
.
. * The command -use- specifies the data-file (in Stata-format)
. * See -infile-, -insheet-, -infix- are used to read ASCII data
. * See -transfer- to translate SPSS export-file into and from Stata-format
.
. * The data that we use describe 242 unemployed men (Tazelaar/Sprengers '82)
.
.   use unemploy
(Unemployment data Tazelaar/Sprengers)
```

```
. * -describe- lists the variable names, and their labels.
.
.   describe

Contains data from \Onderwijs\eha\werk_do\unemploy.dta
  obs:           242                           Unemployment data
                                                 Tazelaar/Sprengers
 vars:            25                           24 Jan 2000 11:12
 size:        14,036 (100.0% of memory free)
--------------------------------------------------------------------------------
   1. caseid    int    %9.0g                   respondent id
   2. studytim  int    %8.0g                   time to re-employment in weeks
   3. died      byte   %8.0g                   1 if found job
   4. age       float  %9.0g                   age at t0
   5. health    byte   %8.0g                   health at t0 (self-assessment)
   6. edu       byte   %8.0g                   education level 1=low, 5=high
   7. exp       byte   %8.0g                   how often unemployed before
                                                 1=min, 4=max
   8. soccap    byte   %8.0g                   social capital: total t0
   9. scweak    byte   %8.0g                   social capital: weak ties t0
  10. scmedium  byte   %8.0g                   social capital: medium ties t0
  11. scstrong  byte   %8.0g                   social capital: strong ties t0
  12. msrch     byte   %8.0g                   search during year 1
  13. search0   byte   %8.0g                   search intensity month 0-6
  14. search6   byte   %8.0g                   search intensity month 6-12
  15. search12  byte   %8.0g                   search intensity month > 12
  16. stopsrch  byte   %8.0g                   month stop searching
  17. lmm       float  %9.0g                   quality labor market position
  18. aa1       byte   %8.0g                   expert: pr(work within one year)
  19. ratio0    float  %9.0g                   demand/supply t0
  20. ratio4    float  %9.0g                   demand/supply t4
  21. ratio8    float  %9.0g                   demand/supply t8
  22. ratio12   float  %9.0g                   demand/supply t12
  23. ratio16   float  %9.0g                   demand/supply t16
  24. ratio20   float  %9.0g                   demand/supply t20
  25. ratio24   float  %9.0g                   demand/supply t24
--------------------------------------------------------------------------------
Sorted by:

. * Lists the variable names that start with the character -s-
.
.   describe s*

   2. studytim  int    %8.0g                   time to re-employment in weeks
   8. soccap    byte   %8.0g                   social capital: total t0
   9. scweak    byte   %8.0g                   social capital: weak ties t0
  10. scmedium  byte   %8.0g                   social capital: medium ties t0
  11. scstrong  byte   %8.0g                   social capital: strong ties t0
  13. search0   byte   %8.0g                   search intensity month 0-6
  14. search6   byte   %8.0g                   search intensity month 6-12
  15. search12  byte   %8.0g                   search intensity month > 12
  16. stopsrch  byte   %8.0g                   month stop searching

. * A more compact list of the variable names
.
.   ds
caseid    studytim  died      age       health    edu       exp       soccap
scweak    scmedium  scstrong  msrch     search0   search6   search12  stopsrch
lmm       aa1       ratio0    ratio4    ratio8    ratio12   ratio16   ratio20
ratio24

. * -summarize- displays summary statistics about a variable
.
.   summarize edu
```

```
Variable |     Obs        Mean   Std. Dev.       Min        Max
---------+-----------------------------------------------------
     edu |     242    2.115702   1.178578          1          5

. * The command -summarize- may be abbreviated to -su-, and similarly, the
. * variable name -edu- may be abbreviated to -ed- or even -e-, as there is no
. * other variable in the dataset with a name that begins with "e".
. *
. * Rule: minimal abbreviations so an object is uniquely identified are allowed.
. * Personally, I seldomly use abbreviations shorter than 3 chars.
.
.   summ ed

Variable |     Obs        Mean   Std. Dev.       Min        Max
---------+-----------------------------------------------------
     edu |     242    2.115702   1.178578          1          5

. * More detailed summary statistics on -oplnivo-, e.g. kurtosis.
. * in Stata, options are included after a semi-colon.
.
.   summ edu, detail

                education level 1=low, 5=high
-------------------------------------------------------------
      Percentiles      Smallest
 1%            1              1
 5%            1              1
10%            1              1       Obs                 242
25%            1              1       Sum of Wgt.         242

50%            2                      Mean           2.115702
                        Largest       Std. Dev.      1.178578
75%            3              5
90%            4              5       Variance       1.389047
95%            4              5       Skewness        .796056
99%            5              5       Kurtosis       2.509328

. * Again, options may be abbreviated to minimal form. These can also be
. * obtained from the syntax diagram available via the interactive help
. * system, namely as the emphasized characters.
.
.   whelp summarize

. * Ok, the minimal abbreviation of -detail- is simply the character -d-
.
.   summ edu, d

                education level 1=low, 5=high
-------------------------------------------------------------
      Percentiles      Smallest
 1%            1              1
 5%            1              1
10%            1              1       Obs                 242
25%            1              1       Sum of Wgt.         242

50%            2                      Mean           2.115702
                        Largest       Std. Dev.      1.178578
75%            3              5
90%            4              5       Variance       1.389047
95%            4              5       Skewness        .796056
99%            5              5       Kurtosis       2.509328

. * We want to display a frequency distribution of the variable -oplnivo-.
. * You will probably believe that Stata is able to do so, but how? The
. * Stata command -search- is often helpful to locate the Stata command for
. * your task. (Do you have an idea how to learn more about the use of the
. * -search- command?)
. *
. * search frequency
. * (output omitted)
.
. * Stata mentions quite a list of commands that have something to do with
. * frequency. Stata tells you that the command -tabulate- displays one-
. * and two dimensional frequncy distributions, while -table- displays
. * higher-dimensional tables.
.
.   tabulate edu

  education |
      level |
     1=low, |
     5=high |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |         96       39.67       39.67
          2 |         73       30.17       69.83
          3 |         30       12.40       82.23
          4 |         35       14.46       96.69
          5 |          8        3.31      100.00
------------+-----------------------------------
      Total |        242      100.00

. * -tab- with two variable specifies a crosstabs of two variables
.
.   tab edu soccap

 education |
     level |
    1=low, |          social capital: total t0
    5=high |         1          2          3          4 |     Total
-----------+--------------------------------------------+----------
         1 |        17         32         39          8 |        96
         2 |        10         17         32         14 |        73
         3 |        11          5          3         11 |        30
         4 |         8          1         10         16 |        35
         5 |         2          2          1          3 |         8
-----------+--------------------------------------------+----------
     Total |        48         57         85         52 |       242


. * We now want to inspect the values of the variables -edu-, -age-,
. * and -aa1- for the first 4 cases. Most Stata commands allow you to
. * restrict a command to cases specifed with an -in range- clause or an
. * -if expression- clause.
.
.   list edu age aa1 in 1/4

          edu        age       aa1
  1.        3    55.7672        39
  2.        2    44.7392        50
  3.        4    55.2444        32
  4.        1    41.9931        57

. * The phrase -5/l would have listed the last 5 cases.
. *
. * List only the cases for which soccap equals 1. Note the double == to
. * denote the "is equal to" operation. Stata reserves the single = to
. * assignments.)
```

```
.
. *list edu age aa1 if soccap==1
. * output is omitted to save paper.
.
. * The -if- and -in- clauses can be combined to list among the first 6
. * cases those for which soccap is unequal 4 (note the double ~=)
.   list edu age aa1 if soccap~=4 in 1/6

          edu        age       aa1
  1.        3    55.7672        39
  3.        4    55.2444        32
  4.        1    41.9931        57
  5.        3    41.6071        43
  6.        5    42.5544        19


.
. * Another way to do this uses the identifier _n (compare CASE in SPSS).
. * Note: & denotes the logical 'AND' while | denotes the logical 'OR'.
.
.   list edu age aa1 if soccap~=4 & _n<=6

          edu        age       aa1
  1.        3    55.7672        39
  3.        4    55.2444        32
  4.        1    41.9931        57
  5.        3    41.6071        43
  6.        5    42.5544        19

. * Pearson's correlations of a variable list
.
.   corr edu age soccap
(obs=242)

         |      edu      age   soccap
---------+---------------------------
     edu |   1.0000
     age |  -0.0830   1.0000
  soccap |   0.1450  -0.0595   1.0000


. * Pearson's correlation of variables separately for respondents who are
. * known to have found a job (died==1), and other respondents (died==0).
. * Most Stata commands allow a -by varlist :- prefix-command. A strange
. * quirk of Stata is that you have to sort the data "yourself".
.
.   sort died

.   by died: corr edu age soccap

-> died=       0  (obs=80)

         |      edu      age   soccap
---------+---------------------------
     edu |   1.0000
     age |   0.0132   1.0000
  soccap |  -0.0233  -0.0069   1.0000


-> died=       1  (obs=162)

         |      edu      age   soccap
---------+---------------------------
     edu |   1.0000
     age |  -0.0615   1.0000
  soccap |   0.1682  -0.0107   1.0000



. * We want to generate a dummy variable whether or not responsents search
. * for a job. We dichotomize the variable on 1.715, and we then add a
. * variable label.
.
.   generate sdumm = cond(search0 > 1.715, 1, 0)

.   label var sdumm "dummy for search0 > mean"

. * There are alternative ways to do this. For instance
. *
. * (1) . generate sdumm = search0 > 1.715
. *
. * (2) . generate sdumm = search0
. *     . recode sdumm min/1.715=0 *=1
. *
. * Try these methods yourselves. And check that everything worked fine!
.
.   tab sdumm

  dummy for |
  search0 > |
       mean |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        123       50.83       50.83
          1 |        119       49.17      100.00
------------+-----------------------------------
      Total |        242      100.00

. * I want to test whether there are health differences (variable health,
. * lower values is healthier) between unemployed subjects who search for
. * work and those who do not search for work. We use a t-test.
.
.   ttest health, by(sdumm)

Two-sample t test with equal variances

------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
       0 |     123    1.894309    .0907658    1.006642    1.714629    2.073989
       1 |     119    1.537815    .0723877    .7896568    1.394468    1.681163
---------+--------------------------------------------------------------------
combined |     242    1.719008    .0592716    .9220502    1.602252    1.835765
---------+--------------------------------------------------------------------
    diff |            .3564938    .1165563                .1268898    .5860979
------------------------------------------------------------------------------
Degrees of freedom: 240

                      Ho: mean(0) - mean(1) = diff = 0

     Ha: diff < 0               Ha: diff ~= 0              Ha: diff > 0
       t =   3.0586                t =   3.0586              t =   3.0586
   P < t =   0.9988          P > |t| =   0.0025          P > t =   0.0012

. * We conclude that those who search for a job are indeed healthier.
.
. * Stata can of course perform OLS-regression of -health- on -age-, -edu-,
. * and 'social capital' (soccap).
.
.   regress health age edu soccap
```

```
  Source |       SS       df       MS                  Number of obs =     242
---------+------------------------------               F(  3,   238) =    7.69
   Model |  18.1130162     3  6.03767206               Prob > F      =  0.0001
Residual |  186.779546   238  .784788008               R-squared     =  0.0884
---------+------------------------------               Adj R-squared =  0.0769
   Total |  204.892562   241  .850176606               Root MSE      =  .88588

------------------------------------------------------------------------------
  health |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
     age |   .0314737   .0124145      2.535   0.012       .0070173      .05593
     edu |  -.1887092   .0490749     -3.845   0.000      -.2853858   -.0920325
  soccap |    .010954   .0557151      0.197   0.844      -.0988038    .1207118
   _cons |   .5818388   .6339846      0.918   0.360      -.6670991    1.830777
------------------------------------------------------------------------------

. * Check carefully that you understand the ANOVA-table and the way in which
. * Stata display estimation results (t-statistics, two-sided p-values,
. * confidence intervals, ..).
. *
. * We can test H0: b[age] = b[soccap] = 0.
.
.   test age soccap

 ( 1)  age = 0.0
 ( 2)  soccap = 0.0

       F(  2,   238) =    3.22
            Prob > F =    0.0418


. * Note that we have to reject H0 at any signifiance level below 3%. Note
. * that provides 2-sided p-values.
. *
. * We can equally simple test an equality constraint, H0: b[age] = b[soccap]
.
.   test age = soccap

 ( 1)  age - soccap = 0.0

       F(  1,   238) =    0.13
            Prob > F =    0.7168


. * that's it...
. Q log close
```

3 The Stata syntax

Stata has a modern, powerful, and consistent syntax. With a few natural exceptions, the basic form is:

> [`by` *varlist1*:]*command* [*varlist2*] [*weight*] [`if` *expr2*] [`in` *range*] [`,` *options*]

Examples of Stata commands are

```
    summarize age
    regress income educ exp sex
    tabulate sex edu if age>25
    tabulate sex edu, nofreq cell chi2
    by cohort: tabulate sex edu
```

Notes on the Stata syntax.

- In the syntax diagram displayed above, optional clauses are enclosed in [ ].
- `by` *varlist1*: requests a separate analysis for each pattern of values of *varlist1*.
- The clauses `if` *expr2*, `in` *range* restrict the set of observations on which the command operates. They may be given in any order.
- A range is of the form #, or #/#, where # stands for a (positive, integer) number or `l`, meaning the last observation. For example `in 1/10` means: for the first ten observations only; `in l` means the last observation only; `-5/l` means the last 5 observations..
- We don't discuss weights in this overview.
- Note that a comma is required before the options. In most cases there may be at most one comma. (Only expressions with functions with more than one argument also use commas.)
- A line commencing with a ∗ is ignored. In a .DO file also any text between `/*` and `*/` is regarded as comment. They need not be on the same line: `/* */` may be used to make a newline invisible to Stata.
- By default command lines terminate with ENTER (carriage return). In a `.do` file (and only there) one can change the command separator to ';' by `#delimit ;` while `#delimit cr` changes it back to 'carriage return'. No other characters are allowed as delimiter. Alternatively the new line symbol can be 'commented out' by ending a line with `/*`, and starting the new line with `*/`.
- The syntax IS CASE SENSITIVE: `a` differs from `A`! All Stata names are in small letters.
- Names may consist of 1-8 letters, digits, and/or underscores, commencing with a letter.
- All names of commands, options and variables may be abbreviated to the minimal unique part, except commands and options that modify or destroy data. Reserved words like `using` and names of programs may not be abbreviated.
- A variable name may contain the wild character ∗. Variable lists may use `-` similar to the TO convention in SPSS. On a list of new names `v1-v100` means `v1, v2, ... v100`.
- Stata supports different types of variables: integers (`byte`, `int`, `long`), approximate-real numbers (`float`, `double`), dates, and alpha-numerical strings. The default is `float`.
- Many system names commence with _, for examples

`_n`	observation number of the current observation
`_N`	total number of observations
`_all`	all variables
`_b`	vector of regression coefficients
`_se`	vector of standard errors of regression coefficients

  Missing values and the use of `if` do not affect the values of `_n` and `_N`. When combined with `by`, `_N` and `_n` refer to the number of observations within the current group.
- For expressions see below, or `help exp`. In logical expressions, use `==` for equality! Stata does allow subscripting (with `generate`, only at the righthand-side), using `[ ]`, `_n`, `_N`, etc.
- Quotes are only used for strings. There double quotes are used: `"..."`

4 Expressions and data transformations

Logical expressions may contain

```
&   |      >   <   ==   ~=   >=   <=
```

Here, `&` is AND, | is OR, and `~` is NOT. Note the use of `==` for equality. In Stata (as in the computer language C) `=` is used exclusively for assignment, and `==` for equality.

Arithmetic expressions may contain

```
+   -   *   /   ^   (   )   [   ]   .   _n  _N
```

Notes:

- x^y stands for x^y. Note that $-2^2 = -4$, and $(-2)^2 = 4$.
- `[ ]` are used for subscripting or for the generation of lagged variables; `x[3]` is the value of `x` for observation 3; `x[_n-1]` is the value of `x` for the previous observation.
- `_n` is the number of the current observation (like `$CASENUM` in SPSS), and `_N` is the total number of observations.
  Missing values and the use of `if` do not affect the values of `_n` and `_N`, but when combined with `by`, `_N` and `_n` refer to the number of observations within the current group.
- '`.`' (the period) stands for the system missing value. (Exception: mv's for string variables are empty strings). Internally, mv's are represented by the largest possible value of the data type. This may require some adjustments! For instance `0 <= x` is true if `x` is missing!

The most important of the available mathematical functions are:

abs()	the absolute value
cond(x, y, z)	if x unequal 0, then y, otherwise z
exp()	exponential function, $\mathrm{e}^{()}$
int()	the integer obtained by truncation towards 0: `int(1.1)=1`
round(x, y)	rounds x in units of y. `round(.,1)` rounds to the nearest integer.
log()	the natural logarithm
min($x_1, x_2, \ldots$)	the minimum of $x_1, \ldots, x_n$; to obtain the rowwise minimum, possibly within subgroups of observations, see the sub-function `rmin()` of `egen`.
max($x_1, x_2, \ldots$)	the maximum of $x_1, \ldots, x_n$
mod(x, y)	x modulo y = the remainder when x is integer-divided by y
sqrt()	the square root
sign()	`sign(x) =` $+1, -1$, or 0 for $x > 0, x < 0$, or $x == 0$ respectively
sum()	The sum of all values of the expression () for all previous observations and the current observation ('running' or 'cumulative' sums)
uniform()	This generates a random number between 0 and 1. No argument is required, but the () should not be omitted. The seed can be changed with `set seed`. By default Stata sets the seed to the same number, always generating the same sequence of random numbers.

Moreover `autocode`, `group`, and `recode` are some very useful *functions (!)* for recoding into a discrete set of values. The command `tabulate` can be used to generate (univariate) dummy variables. They are discussed in the next paragraph. Also there are several functions on strings, and between strings and numbers, distribution functions of the normal distribution, the χ^2 distribution (`chiprob(df,x)`), the F-distribution, and the t-distribution, and the inverse of the normal distribution. The last function can be used to generate normally distributed random numbers: `invnorm(uniform())`. For more detail see the manual or `help functions.`

Data transformations Stata has very good data transformation facilities.

- Unlike in most packages there are different commands for defining new variables `generate` and for modifying existing variables `replace`. Both have about the same power as the SPSS commands `COMPUTE` and `IF`.
- `tabulate` can be used for making dummy variables, called indicator variables by Stata.
- Stata expressions are quite powerful: a good set of functions, and mixing logical and numerical expressions (this will be explained shortly).

For changing the values of a discrete variable (like SPSS' `RECODE`) you can also use the command `recode`. Here is an example of its use.

```
recode xyz 1=2 2=1 *=3
```

For the variable `xyz` the value 1 is replaced by 2, 2 is replaced by 1, and any other value is changed to 3. Three examples of `generate` are:

```
gen laginc = inc[_n-1]
gen loginc = log(inc)
gen hiinc = inc > 100000
```

The first example shows how to make a lagged variable. The third example creates a dummy variable `hiinc` that is 1 for incomes exceeding 100,000 and for missing values (which are regarded as very large and positive numbers in logical expressions, beware!), and 0 otherwise.

Any logical expression can be used as (part of) an arithmetic expression: 'true' is interpreted as 1, 'false' as 0. Conversely, every arithmetic expression can be interpreted as logical expressions: any expression yielding 0 is taken as 'false', any expression yielding another number *or a missing value* is taken as 'true'.

Assume that we have a variable `age` (in years) that we want to recode it into four categories with breakpoints 20, 40, and 60 years. Now the logical expression `(age>20)` is 1 (= 'true') for all people over 20. Thus we can write:

```
gen age4 = 1 + (age>20) + (age>40) + (age>60)
```

generating a new variable `age4` that is 1, 2, 3 or 4. It is 1 for all respondents of age 20 or less, and 4 for all 60+.

If we want to transform a continuous variable like `age` into a discrete one, using the upper class-boundaries as new values we may use the function `recode`:

`gen` *newvar* `= recode(`*oldvar*, $x_1, x_2, \ldots, x_n$`)`

If *oldvar* $\leq x_1$, *newvar*$= x_1$, otherwise if *oldvar* $\leq x_2$, *newvar* $= x_2$, etc., and if *oldvar* $> x_{n-1}$, *newvar* $= x_n$ (!). Thus unlike $x_1..x_{n-1}$, x_n is not a breakpoint, only a 'new value'.

To transform age into the same four categories as above we could say:

```
gen age4a = recode(age,20,40,60,80)
```

Now `age4a` is just 20∗`age4` above. An automatic version of `recode` is `autocode`.

`gen` *newvar* `= autocode(`*oldvar, ng, xmin, xmax*`)`

Now the interval *(xmin, xmax)* is 'automatically' divided into ng subintervals of equal length, and the new value of newvar is the upper bound of the interval to which oldvar belongs. Note that above age has been divided into intervals of equal length. If we have no respondents over 80 years old, `age4a` can also be obtained thus:

```
gen age4a = autocode(age,4,0,80)
```

A way to divide the data into n (nearly) equal groups is by the function `group(`n`)`. Most likely you first want to sort the data according to some variable, say age.

```
sort age
age4b = group(4)
graph age age4b, symbol(.)  jitter(4)
```

Here, `jitter` adds a little random noise to the data, so that a group of coinciding observations becomes a blot, and `symbol` defines the symbol representing a single observation (default is `o`, which I often find far too big). Now for the youngest respondents `age4b` becomes 1, and for the oldest and those with age missing `age4b` becomes 4. Note that the division lines may cut across age groups: Some persons with `age4b==1` will have the same age as some with age4b==2. Look at the scattergram produced by graph. (For the options symbol and jitter see the manual or help graph.)

A single dummy variable indicating all observations that have the value 3 on the variable x can be created as follows:

```
gen x3 = (x==3)
```

where the parentheses are indeed optional. Note that the logical expression (x == 3) is used in a numerical context here, so it returns the value 1 if it is true and 0 if it is false. Now if x is discrete, and one wants a dummy variable for each possible value the above method is rather cumbersome. A better alternative is to use the generate option of tabulate as follows.

```
tab x, generate(newvar)
```

If x can assume three values, the corresponding dummies get the names newvar1, newvar2, and newvar3. So in general newvar could be x as well, generating x1, x2, x3 as names for the dummies.

Some powerful data manipulation is possible with the by construct. For instance, you want to select the oldest persons in households. Then

```
sort hhold age
by hhold :  gen oldest = _n==_N
some cmd if oldest
```

As another example, suppose you have data on personal incomes of persons within househoulds. You want to add a household income variable. Then

```
sort hhold
by hhold :  gen hhinc = sum(inc)
by hhold :  replace hhinc = hhinc[_N]
```

This last operation is more easily accomplished via one the the egen functions:

```
egen hhinc = sum(inc), by(hhold)
```

5 A summary of Stata commands

In this section we give short explanations for the most important Stata commands, to give an idea of what is available. We indicate permitted abbreviations by **underlining**. Don't overuse abbreviations in files that are saved. It makes them difficult to decipher. We include SPSS analogues whenever available.

5.1 Help

Stata	description
help help	interactive help on using the help system
help *topic*	interactive help on *topic* (Stata commands), for instance help regress
search *string*	list descriptions of Stata commands related to the 'statistical' term *string*, for instance search regression. The output also includes articles that appeared in the Stata Technical Bulletin, and programs available in archives of Stata programs.
webseek *string*	as search, but on the Internet!
lookfor *string*	lists variables that contain *string* in the variable name or variable label, for instance lookfor father

5.2 Reading and writing data files

Stata	SPSS	description
`use`	`get file`	Get a Stata system file for processing
`save` *filename*	`save`	Save as a Stata system file; don't forget `,replace` if you want to overwrite an existing disk file
`merge using` *filename*		Add *variables* from another system file to the current file; either observationwise (= listwise), or through one or more match key variables; may also be used for 'table look-up'. See `mmerge` for a easier-to-use extension.
`append` *filename*		Add *cases* from another system file
`expand =`*expr*		Duplicates cases *expr* times. Quite useful for the generation of person-period files and episode splitting for event-history models.
`compress`		Try to compress the data file by converting, for example, 4 byte reals to integers if this is possible without loss of information. using this feature, SPSS export files can usually be made much smaller than in SPSS.
`input` *varlist*		Interactively input data. Type `end` to stop interactive input.
`infile` *varlist*	`data list`	Read ASCII data in a free or fixed format
`infix` *varlist*	`data list`	Read ASCII data in fixed format
`insheet using` *filename*	`data list`	Read ASCII data in a tab/comma seperated format
`outfile using` *filename*	`write`	Write free format ASCII data file (optionally with a dictionary)
`outsheet using` *filename*	`write`	Write a tab or comma seperated ASCII data file

5.3 Modifying data interactively

Stata	SPSS	description
`generate` *newvar* `=`	`compute`	Create a new variable *newvar*
`replace` *oldvar* `=`	`compute`	Changes the values of the existing variable *oldvar*
`edit / browse` *varlist*		Spreadsheat-like editting/browsing of *varlist* (Only available with the Windows/Mac versions of Stata.)
`for`	`do repeat`	Repeat a command for a variable list, a numeric list, or a list of arbitrary strings
`egen`		numerous useful extensions to generate, like 'rowwise' means, ranking of cases, etc
`recode` *varname*	`recode`	Recodes the variable *varname*
`impute`		regression imputation of missing values
`reshape`		Change data-organization between 'wide' and 'long' formats. Quite useful to reorganize multi-level data.

5.4 Descriptive procedures

Stata	SPSS	description
`describe`	`display`	List of variable names, labels, # of observations, etc.
`ds`	`display`	Compact list of variable names
`summarize`	`freq`	Mean, st.dev, min, max, # of valid observations
`summ , detail`		Also give some quantiles, skewness, and kurtosis
`by ...:  summ`	`breakdown`	Statistics for subgroups
`tab ..., summ()`	`breakdown`	summarize for subgroups
`tabulate`	`freq, crosstab`	One- and two way tables
`tab , plot`	`freq`	Histograms
`by ...:  tab`	`crosstab`	Multiway tables
`table`	`crosstab`	Multiway tables, with enhanced formatting
`collapse`		Aggregate data yielding mean, sum, or median of specified variables in subgroups
`inspect`		More univariate summaries for data inspection
`correlate, pwcorr`	`pearson corr`	Correlation or covariance matrix (of variables or of the parameters of last estimation command)
`pcorr`	`pearson corr`	Partial correlations
`spearman / ktau`		Spearman's rank correlation, Kendall's tau-b.
`count if` exp		For how many observations does expression *exp* hold ?
`list`	`list`	List observations
`display`		Calculator, formated output

5.5 Graphics

Stata	description
`graph , box`	box plots
`graph , hist`	histograms
`graph , oneway`	bar-code like frequency plots
`graph , matrix`	matrix plot of two-way scattergrams
`graph , twoway`	2D graphics
`graph , star`	multi-variable star plot
`hilite`	scattergram, hiliting certain observations
`avplots`	added-variable plots after regression (fit)

5.6 Statistical procedures

Stata	SPSS	description
predict		Predictions, residuals, influence statistics for the last estimation command
diag		regression diagnostics (after fit)
sw est-cmd		stepwise application of the estimation command est-cmd
test		Wald tests for last estimation command (uni- and multivariate)
lrtest		Likelihood-ratio tests
hausman		hausman-type tests
tabl		tabulates of coefficient-estimates for different models
regress / fit	regression	Linear ‘multiple’ regression
cnreg / intreg		Censored-normal and tobit regression
rreg / qreg		Robust regression, quantile regression
heckman		Heckman’s selection model
anova	anova	Anova (analysis of variance) and Ancova (analysis of covariance)
oneway / loneway	oneway	Oneway anova (with many tests), random effects
xt / xtgls / xtgee		‘Repeated measures’, panel analysis
alpha	reliability	Cronbach’s alpha for reliability
rasch		Rasch intem-response model
factor	factor	Principal components and factor analysis. Use rotate to rotate the factor solution. Use score to predict factor scores.
canon		Canonical correlations
logit / logistic	logistic	Logit analysis
probit		Probit analysis
mlogit		Multinomial logit (polytomous logit)
clogit		conditional logit model (estimates a.o. fixed-effects logit, the Rasch model)
ologit / oprobit		Ordinal logit/probit regression
poisson	loglinear	Loglinear models
glm		Generalized linear models (cmp. GLIM)
xtgee		Estimation equations for generalized-linear models
ltable		Life-table and Kaplan-Meiyer estimates (old system)
ereg / weibull		Exponential and weibull regression models (old system)
cox	coxreg	Cox’s semi-parametric regression model (old system)

5.7 Miscellaneous procedures

Stata	SPSS	description
`sort`	`sort`	Sort (the observations) on one or more variables; missing values, being represented by very large positive numbers, come at the end.
`order`		Reorder the variables in the data matrix (see also `aorder`)
`rename`		Change the name of a variable
`drop if ...`	`select if (not ...)`	Drop observations
`drop` *varlist*		Drop variables
`drop all, clear`		Make a clean start; needed before another data set is read in
`sample`	`sample`	Random sample of observations from data
`keep`		The opposite of `drop`
`label`		Define, inspect or modify labels for the data set, variables and values
`assert` *expression*		This command produces no output if the expression is true. Otherwise it mentions the number of violations. It is useful to make a file with such assert commands that watch over the integrity of ones data ('if less than 10 years old, does not have children', etc.). After cleaning or modifying the data, it is easy to apply these checks again.
`do`	`include`	Execute an ASCII file of Stata commands
`run`		As `do`, but silently, without output
`exit`	`finish`	Quit Stata (not allowed when data set is changed after last SAVE)
`exit, clear`		Quit Stata anyway
`log using` *filename*	`automatic`	Make a log of input and output; output may be suppressed with `,noproc`; do not forget `, replace` if needed;
`ml`	`clnr`	Advanced: Maximum-likelihood estimation
`nlr`	`nlr`	Advanced: Non-linear regression
`matrix`		Advanced: Matrix algebra

In addition, Stata comprises several 'packages' of interrelated programs for specific types of data. The `st` and `xt` packages are of primary concern for this course: They provide descriptive and analytic commands for the analysis of 'survival time' (st) and 'cross-sectional time-series' (xt) data respectively.

The package `svy` provides a fairly unique collection of commands for the analysis of survey data. (The only comparable software that I know about is *Sudaan*.) In these commands, randomness is explicately attributed to the sampling mechanism underlying the survey (e.g., unequal selection probabilities, clustered sampling etc). It has long been recognized among survey statisticians that these design effects can not be properly dealt with via weighted analyses as provided in SPSS and BMDP etc. The rest of the world, including most social scientists who analyze survey data, seem to ignore their wisdom.

6 Software for survival analysis and repeated measurement

This section outlines the Stata packages for survival time analysis (st) and for repeated measurements (xt). We include the most important commands that belong to the official 'core' of

Stata (based on release 6), some commands that have been published in the Stata Technical Bulletin (a periodical that aims to facilitate communication between Stats users) or on the Stata listserver (an active Internet discussion group of Stata users), and that were developed 'in house' at ICS (ICSLIb).

6.1 Survival Time Data

command	description
declaration of data and data management	
`stset`	declares data to be survival-time data
`stdes`	describes survival-time data
`stvary`	report which variables vary over time
`stfill`	fill in by carrying forward values of covariates
`stgen`	generate variables reflecting entire histories
`stbase`	form baseline dataset
`stegen`	create time-varying covariate (with episode splitting) (ICSLib)
`strepl`	modify time-varying covariate (with episode splitting) (ICSLib)
`stsplit`	regular-intervals episode splitting
`stesplit`	alternative command for regular-intervals episode splitting (ICSLib)
`stjoin`	recuces multi-episode data to compact format
`stcoxtvc`	Advanced: event-time expansion for use with stcox (ICSLib)
summary statistics	
`stsum`	summarize statistics for survival-time data
`sts graph`	graphs the estimated survival (failure) function
`sts list`	list the estimated survival (failure) function
`sts test`	tests the equality of the survival function accross groups
`sts gen`	creates a new variable containing the estimated survival (failure) function and/or related functions
estimation and testing commands	
`stcox`	estimate Cox proportional hazards model
`streg`	parameteric survival time models
`stcurv`	displays the estimates baseline after `streg`
`stdreg`	parameteric survival time models in discrete time (ICSLib)
`ststrata`	test for equaility of baselines in stratified Cox (ICSLib)
`stbconx`	test that coefficients are time-constant (ICSLIb)
`stphtest`	test proportionality assumption in PH hazard models

6.2 Repeated measurements, panel methods etc

command	description
Declaration of data and data management	
`iis, tis`	declares data to be repeated/panel data
`xtdes`	Describe pattern of xt data
`xtlist`	List xt-data (ICSLIb)
`xtsum`	Summarize xt data
`xttab`	Tabulate xt data
`xtvary`	Reports which variables vary between measurements/over time
Estimation and testing	
`xtreg`	Fixed-, between- and random-effects, and population-averaged linear models
`xtdata`	Faster specification searches with xt data
`xtgls`	Panel-data models using GLS
`xtrchh`	Hildreth-Houck random coefficients models
`xthaus`	Hausman test: random vs fixed effects
`xtlogit`	Fixed-effects, random-effects, and population-averaged logit models
`xtprobit`	Random-effects and population-averaged probit models
`xttobit`	Random-effects tobit models
`xtpois`	Fixed-effects, random-effects, and population-averaged Poisson models
`xtnbreg`	Fixed-effects, random-effects, and population-averaged negative binomial models
`xtclog`	Random-effects and population-averaged cloglog models
`xtintreg`	Random-effects interval data regression models
`xtgee`	Population-averaged panel-data models using GEE
`xtcorr`	Working corr matrix of population averaged model
Other methods for repeated observations	
`alpha`	Cronbach's alpha
`anova`	general ANOVA command, with 'classic' support for repeated measures
`loneway`	one-way ANOVA, with random effect
`mokken`	Mokken scale analysis
`rasch`	Rasch analysis (ICSLIb)

Stata also has extensive support for time series data (repeated measurements on a single unit). These methods are not treated in this course.