

EDA-Software: Expressions + hints¹

1 Selections

Often it is quite useful to analyse meaningful subsets of observations (cases). Selection commands select *temporarily* a subset, according to some specified criterion.

Important: A selection remains in effect until it is turned off explicitly, another selection command is issued or you are requesting an operation needing or affecting the full work area, like GETting a new work area.

Selection is a general mechanism; a number of commands are dealing with selection definition and modification. The main selection commands are ANALYSE, INCLUDE and EXCLUDE.

Before explaining the selection commands individually, let us describe the how selections work, using the ANALYSE command as an example. The basic form of the ANALYSE command is used to select predefined groups of observations.

Consider the following command sequence:

```
>ANALYSE GROUP=1          ! Select group 1 for analysis
>BOXPLOT 4                ! only group one
>STEM
>END                      ! turn selection off
>BOXPLOT 1
```

ANALYSE activates a selection: only observations being members of group number 1 are included. The BOXPLOT and STEMLEAF command immediately following the ANALYSE command is only performed on the subset. The END command terminates selection, i.e. the BOXPLOT following it will be produced on the full data set.

The boxplot below has been produced with an active selection, i.e. the initial message telling you about the number of observations in the analysis is changed. Watch also the status line on your screen: it shows whether a selection is active or not and - if a selection is active - the kind of selection, here in this example it would show that the included continent is Asia.

```
39 countries out of 183 countries
Boxplot :GNPAgr(20) %GNP for Agriculture
0.00                                          76.00
x-----*-----x      o o      @
Extreme values (LO,HI): HONG CAMB adjacent(LO,HI): HONG NEPA
Hi outliers:AFGH LAOS CAMB
```

As already said all selection command follow the same pattern. Let us now turn to the selection commands and give some additional details.

ANALYSE is a selection command used to select (predefined) groups of cases.² ANALYSE requires a GVAR to be defined for the current WA. A GVAR (GVAR stands for grouping variable) defines group memberships for each case. GVARs correspond either to “natural” groupings of cases, e.g. continents in the GLOBE datasets or to groups defined by analysis (cluster analysis, groups defined by a “stem” of a histogram etc.)³

```
>ANALYSE GROUP=1          ! include members of group 1
>ANALYSE GROUPS=(1,5,6)  ! include members of groups 1, 5 and 6
>ANALYSE "Africa"        ! include members of group named Africa
```

1. E. Horber, 13.12.98 : EXPRESS.mss

2. In addition to this basic function the ANALYSE command offers a number of advanced options.

3. The GVAR should be considered an overall attribute of the work area; a GVAR does not appear in the variable list. The GVAR command lets you manipulate the GVAR.

Note that for the last example GVAR names must be defined in your work area. Group names are case sensitive.

Here is a list of the other selection related commands. Several of them use logical expressions. The syntax and rules governing logical expressions are explained below in the section on expression.

INCLUDE log-expr	Include cases satisfying the condition
EXCLUDE log-expr	Exclude cases satisfying the condition
END	Turns case selection off
ELSE	Reverse the selection: currently selected cases are deselected and the non-selected cases are selected.
SELECT	Remove the non-selected cases from the WA (this is the only selection command actually changing the data set).

Advanced options

AND log-expr	Restrict the active selection
OR log-expr	Extend the active selection
REMOVEIF log-expr	Remove from active selection
BUTNOT log-expr	Restrict active selection

ANALYSE In addition to the basic form used to include groups into analysis, this command offers a number of other options, including a case-name based selection and advanced options for random selections and the like.

As selections are so important in exploratory work, EDA offers a number of commands producing similar effects to help you to find your own style and vocabulary.

INCLUDE, EXCLUDE and ANALYSE, specified when a selection is active turn the currently active selection off, before defining the new one.⁴

There is a special form of the IF command: IF .. DO. It can simplify situations where you would write:

```
>INCLUDE #1>MED(#1)
>BOXPLOT 1-5 PARALLEL
>END

>IF #1>MED(#1) DO BOXPLOT 1-5 PARALLEL
```

The logical expression tells EDA to select only observations, for which variable #1 is greater than the median. Note also that after both sequences, no case selection is active.

2 Expressions

The aim of this section is to introduce expressions using a number of examples. Many details are left out. For more advanced work you will need to check the manual.

a The LET command

```
>LET #10=#2+#3
```

Computes a new variable #10: the sum of variables number 2 and 3. With expressions the # sign is used to distinguish a vector (variable) from a scalar variable.

```
>LET #10=2+3
```

creates a “constant” variable, all cases set to 5.

4. This is the default for INCLUDE and EXCLUDE. If you prefer however you may use INCLUDE/EXCLUDE to extend or restrict the selection. Turn the SET STICKY option to ON if you want this.

Instead of using the variable number you may of course use the variable names (#GNPAgr). Note that the # must be present.

The result of the evaluation of an expression is either a vector (a value for each observation) or a scalar. #1+#2 produces a vector, whereas 3*20 or MEAN(#3) produce a scalar result. If the result is a vector a target must be present, i.e. the result is stored into a variable in the WA (there are other advanced possibilities).

If the result is a scalar you may omit the target. EDA then displays the result on the screen.

```
>LET 23*45-LOG(35)
```

computes the expression and displays the result on the screen. If the first character of the expression is a numerical character then the LET command may be omitted: (the # is considered numerical)

```
>23*45-LOG(35)
```

produces the same result.

```
>LET #29=#4/#3
```

will ask you to supply a label and descriptor, if you write

```
>LET #MYVAR=#4/#3
```

the label will be MYVAR and you will be asked to type in a descriptor (if you don't like the way EDA does this, check the SET EXPRESSION command for alternatives).

Expressions may contain (simplified presentation):

Variable names or numbers	#Cath, #3
Constants	12.3 35
Operators	+ (Add) - (subtract) * (multiply) / (divide) % (percent) ^ (exponentiation)
Parentheses	()
Special names	GVAR
Functions	Math.: SQRT LOG, SIN, TAN ... General: ROUND TRUNC... Stat.: MEAN MEDIAN LHI (lower hinge) HOF (Higher outer fence)

For a complete list of functions refer to the manual or the on-line help. Note: only the first 3 letters are required with functions.

Advanced users: More functions are available for statistical experimentation: random numbers, distributional functions (e.g. GAUSS, CHI, STUD etc), functions useful in macros and more like variable construction functions.

b) The IF command

```
IF log-expr THEN target=expression  
IF log-expr THEN target=expression ELSE target=expression
```

The IF ... THEN command is used to perform conditional computations: as for instance in

```
>IF #1>MED(#1) THEN #2=1 ELSE #2=0
```

This command creates a new variable #2, setting observations above the median of variable #1 to 1 and the others to 0.

The INCLUDE, EXCLUDE and some other selection commands mentioned above use logical expressions, as defined in this section.

Logical expressions may contain, in addition to the elements, mentioned above with (algebraic) expressions

Logical operators	= > <
	(or) & (and)
Logical functions	OUT FAR WHI (within hinges) EXT (out or far)
	WIA (within adjacents) etc.

Let us examine two examples:

```
>EXCLUDE FAR(#1) | FAR(#2)
```

This case selection commands excludes all observations having far-outliers in values in variables #1 or #2. FAR() is a logical function; FAR() is true for far-out values and false otherwise.

```
>INCLUDE GVAR=1 & #GNPCap>125
```

This examples includes observations matching the following condition: be in group 1 and having GNP per capita over 125.

Advanced users EDA offers many options for advanced users and hackers, for instance you may write

```
>IF FAR(#1) THEN #1=MEDI(ALL(#1))
```

This means that you replace the far out values of variable 1 by the median of all observations. The ALL function is needed, as the expression

```
>IF FAR(#1) THEN #1=MEDI(#1)
```

replaces the far-out values of the variable by the median of the far-out values only. If you want the median of all values except the far out values you may write

```
>IF FAR(#1) THEN #1=MEDI(OTHER(#1))
```

Other transformation commands

There are a number of transformation commands for common problems, like adding a number of variables, like computing percentages etc. For instance if you do not like to write a large number of LET commands to compute percentages you could simply write something like:

```
>PERCENT 1-40 BY=#TOTPOP
```

i.e. all variables are divided by TOTPOP and multiplied by 100.

```
>PERCENT 1-10 TABLE
```

TABLE tells EDA that the sum of the 10 variables (across cases) is the total, i.e. each variable will be computed as a percentage of the total of all variables.

(Advanced users): There are many other facilities including macro commands, command repetition and the like.

3 Useful hints

Up to know you have basically used prepared data sets. With transformation commands however you might want to prepare a data-set for more than a casual look. Below you will find some hints for some housekeeping tasks as well as saving and combining work areas.

Important The current work area is temporary, all changes you make to it are lost when you QUIT EDA or when you GET a new work area.

APPEND command

The GET command destroys the current WA before reading the new one. If you want to add data from a different WA you could use the APPEND command (syntax identical to GET). It has additional options for reading only some variables, e.g. APPEND DEMO VAR=3 will append variable 3 from the DEMO work area. APPEND GLOBE2 VAR=(10,20) will read variables 10 through 20 from the GLOBE2 WA.

Important: The variable numbers are just sequential numbers, therefore if your WA contains e.g. 10 variables an APPEND DEMO VAR=3 will at this variable at the end of the WA, i.e. the variable will become the 11th variable in your current WA.

DICT

Shows the contents of a WA in the library without reading the data into the current WA, i.e. it is similar to DESCRIBE, except that you need not GET the data. (Useful to find variables).

PUT, REPLACE

You may save the current work area using the PUT command. The WA will be inserted into your personal library and will appear with the DIR command after the display of the data-sets in the common library. You cannot replace data-sets in the common, shared library as they are protected. If a work area already exists in your library you cannot use PUT, use REPLACE instead. To remove a data-set from your library use DROP.

Some housekeeping commands:

```
DELETE 1-4          ! Delete variables 1 through 4
DELETE CASE=JU      ! Delete case JU
DELETE GROUP=1      ! Delete group number 1

KEEP 1-4            ! Keep only variables 1-4 (all other variables are
                    ! removed from the WA.

ERASE                ! Erase the whole work area
PACK                 ! when you delete some variables there might
                    ! be holes in your WA, PACK makes the WA contiguous
                    ! again.
```

4 Entering your own data

NEWVAR

The NEWVAR command reads data from the keyboard. Follow the instructions displayed. (Data entry is terminated by typing <return> on an empty line. As always you may cancel with //.)

Here is a commented example:

```

(1)  >NEWVAR
(2)  >>> Terminate with a blank line.
(3)  Enter variable # 1
(4)  case 1(1) :12 45 23 23 23           type-in data
(5)  case 6(6) :45 1 1 221, 2, 12      12
(6)  case 13(13):11 73A 10 23 23 1
(7)  <4> bad number:73A                 oops, 73A is not a number
(8)  >>> Correct or cancel with //:73
(9)  <4> bad number:10                   O (letter) and 0 (number!)
(10) >>> Correct or cancel with //:10    correct it
(11) case 19(19):2 3 4 5 2
(12) case 24(24):
(13) Label   Descriptor                 empty line= we're done
(14) exam    this is an example         now we need label/descriptor!
(15) >                                   type it

```

Comments: (1) If no variable list is present, a single variable will be appended to the end of the WA.⁵ In this example we entering variable number 1. (4)(5)(6)(11) are lines where the user types data. The prompt contains e.g. "case 1(1) :"⁶ (7) through (10) show how bad numbers are handled: they are diagnosed and you are asked to supply a replacement. Please note that you need to reenter the full data line, only the number in error needs correction.

(12) Input is terminated with a line without any data. After it you will be asked to enter a label for the variable, as well as a descriptor.

The WA is then considered protected, i.e. you will have to confirm that you want to overwrite the current WA or to leave it (QUIT) without saving.

```

>QUIT
>> Leave protected WA: Y/N?:

```

In addition to this, the individual variables entered from the keyboard are also protected. Consider that after having entered data as above you type

```

>LET #1=#2*1000
<65> Protected var# 1

```

the LET command produces error message <65>, because you cannot overwrite a protected variable, without unprotecting it first (see the SET WAPROTECT and the REVERT commands) or save it (PUT).

***READ RAW**

reads in raw data (ASCII files) command in various forms (the manual has many detailed examples).

***READ SPREADSHEET**

Reads a Lotus 123 (WKS or WK1) file. Note that this is one of the possibilities of bringing data from many packages (EXCEL, SPSS, MicroCrunch, Minitab etc) over to EDA. Make sure to select WK1/WKS when exporting, i.e. Lotus 123 versions 2 or earlier.⁷

EDA to SPSS?

If you want to take EDA data over to SPSS, do the following: Prepare the WA with the data you want to take. Then type

```

>*WRITE SPSS "MYFILE"

```

5. In fact it is inserted into the next free location in the WA.

6. If the default observation designation has been changed, you might read e.g. "country", "measure" instead of "case". The number following it is the case number, whereas in parenthesis you will find the case name. In this example (default with an empty work area) case number and case name are identical. In a different context the prompt might read "country 1 (AFGH) :", i.e. the next country you are entering is Afghanistan.

7. EDA cannot import other WKx versions; the WK1 format is one of the most popular formats for transporting spread-sheet type data from one application to another.

This produces a file with SPSS commands and the data.⁸

Quit EDA and call SPSS.

Within SPSS open the file written above (FILE>OPEN>SYNTAX), select everything and run it.⁹ Use the dataset with SPSS.

8. If you are using SPSS/PC add the PC option to the command, as there are syntactical differences. Note: SPSS/PC refers to the DOS version; SPSS for Windows is a full SPSS version.

9. SPSS/PC: within SPSS type INCLUDE 'MYFILE'. This command will include all necessary SPSS commands to recreate the dataset you had in EDA within SPSS.

Table des Matières

1 Selections	1
2 Expressions	2
3 Useful hints	4
4 Entering your own data	5