Conversion table: How to adapt your pgm file to new EpiData Analysis format

H.Rieder/J.Hockin/TB.Christiansen/J.Lauritsen
Version: ..1.0
(sequence can be used for sorting) NA used where something is not appropriate.

In all examples files are called "datafile.xxx", where xxx is the appropriate extension. examples should be adapted to your variable naming

Sequence	Task	Classic Analysis	New Analysis

1	Give a comment in the pgm	*
		example:
		* this is a comment
			//
			example:
			// this is a comment
			/* this is a longer comment, which could extend over several lines */
			
			freq sex ; // inline comments are also allowed
			freq /* block text comments are allowed */ sex;

2	continue a command on next line	* add a \ at the end of first line read datafile \ close	// not required - command only ends at first ; read "datafile.epx" !c ;
3	Read data	read datafile read datafile.rec	// file name must be a string read "datafile.epx" read "datafile.rec" read "datafile.dta"
4	indicate end of a command including options	Automatic with “end of line” or separate commands with ; example:	Commands are ended with semicolon (;) several commands can be placed on one physical line example: read “datafile.epx”; list variables; freq sex;

		read datafile; variables; freq sex	

5	read rec file and use valuelabels contained in the chk file	* the chk file is read automatically
		read datafile
			chk files must have labels in a labelblock, then:
			
			read "datafile.rec";
			// to assign labels to variables, first get their names:
			list valuelabel;
			// now add them to each variable one by one:
			edit var sex !valuelabel := label_sex;
			edit var reason !valuelabel := label_reason;
			edit var res1 !valuelabel := label_result;

6	Select data	Select is incremental. start: select (statement) end: select example: select age < 20 freq sex select	select NOT incremental select (condition) do something; example: select (age < 20) do freq sex; notice: select cannot be used with merge or save

7	give a variable a new value	v1 = 1	// notice that := is used for assignment
			// and = is used for comparison (below)
			v1 := 1;

8	give a conditional new value	if (sex = 1) then v2 = 40	// notice that = is used in the logical statement, but := is used when giving the new value: select (sex = 1) do v2 := 40;
9	add options to commands	command /option command /option = ... example: freq sex /ci /r means age /by=sex /t	Use the ! instead of / command !option command !option := .... example: freq sex !ci !r !d0; means age !by := sex !t;

10	save data	savedata saves records in the current select
		example:
		select sex="F"
		savedata female /replace
			// file name must be a string
			example to save only females. Notice that currently one may not "keep", but have to "drop" observations not desired.
			
			read "alldata.epx" !close;
			// first drop the data
			select (sex <> "F") do drop data;
			save "female.epx" !replace;
			// to continue to work with all observations :
			read "alldata.epx" !close;

11	create a new data variable	define var1 fieldtype or gen type var1 = (numeric only) examples: define incubation ### label incubation "Incubation period in days" incubation = onsetdate - mealdate or gen i incub = onsetdate - mealdate	new variable var1 type [:=expression] example: new v incubation integer := (onsetdate - mealdate) !label:="Incubation period in days";
12	create a new global variable	define var1 fieldtype global example: define group <A> global group = "M" select sex = group means age	new global var1 type [:=expression] example: new g group string := "M"; select (sex = group) do means age;

13	count records matching some criteria	count if [condition]
		example:
		count if sex = "M"
			select [condition] do count;
			example
			select sex = "M" do count;

14	show all variables	variables	list variables;

		or var	or list v;

15	show all results	result
		or
		var result
			list results;
			or
			list r;

16	remove all observations marked for deletion	set the option: set READ DELETED = OFF savedata	drop data !del ; // Only drops observations marked for deletion. save ....

17	include several variables in a command	not possible unless the user used the full name	abbreviations possible. Here used with list data .
			list data v* ; // all variables starting with "v"
			list d v? ; // length of var name is two characters
			list d v* ; // All variables starting with a v
			list d v1-age ; /* all variables in the dataset from v1 to and including age. */