

Introduction

Examples in this chapter will use the SPSS sample dataset demo.sav

The chapter will provide a brief overview as to what SPSS syntax is and how to begin using it. It will outline: how to become familiar with the syntax commands through the use of the Log; how to create and manage the SPSS Journal and syntax files; and how to use syntax throughout the data entry, management and analysis process.

It will also explain how the use of syntax can: allow a data audit trail; enhance analysis options; and, once mastered, how it can save time. This chapter will set the context for the rest of the book, promoting SPSS syntax as additional to the usual drop-down, menu-driven method, using the two methods in a complementary way.

SPSS is one of the most commonly used statistics packages in today's higher education institutions. Most users find the drop-down, menu-driven method of accessing commands and functions easy to use and often consider it sufficient for their needs. For those completing analysis for a one-off small dissertation, that appraisal is almost certainly correct. However, for anyone who is likely to spend a considerable amount of time carrying out quantitative analyses or needing to repeat analyses (perhaps for different sub-groups or for later datasets), then it is worth learning to work with the command language responsible for carrying out the SPSS commands and functions, SPSS syntax.

One of the common misconceptions about the use of SPSS syntax is that it requires an 'either/or' approach – that you have to use *either* the drop-down menu interface *or* the syntax programming language. This view is unnecessarily polarised. The approach in this book is that the two methods of using SPSS can be used alongside each other to optimise their usefulness and be tailored to the level of expertise of the user.

It is anticipated that the newcomer to syntax, already familiar with the drop-down, menu-driven SPSS, will *gradually* learn about SPSS syntax, passing through several stages of use. A suggested route is that you start by first just looking at the commands generated automatically by SPSS in the Log or the Journal (explained below); then start creating syntax files using the paste function; and, in

time, beginning to add to and amend the commands building from commands acquired through the paste function; and finally through creating commands based on a library of previous syntax files or even completely afresh. This suggested route of progression can be halted at any level because at any stage of use syntax can still be a useful adjunct to the user of SPSS. This first chapter will outline how each of these stages can be initiated and developed.

Good Practice

- Always have your Journal set to Append.
- Always have your Journal saved to a safe location.
- Back up your Journal periodically.
- Always have your Log set to show in the output sheets.
- Set your options so that a blank syntax sheet opens automatically whenever you open SPSS.

Explanations and Illustrations of the Log and the Journal

Whenever I begin SPSS work with someone new, I start by ensuring that two SPSS elements called the 'Journal' and the 'Log' are set up to record the actions of the researcher using SPSS. They both record the actions undertaken in SPSS – the Journal keeps a 'behind-the-scenes' record and the Log produces an immediately visible record on the output viewer. Setting these up correctly is an essential step for any user of SPSS and even if you do nothing else because of this book except set up your Journal and Log then the book has still been of value.

The Journal – What it Is, and How to Set the Journal Options Appropriately for Syntax Use

Most people do not realise that SPSS runs and stores a background record of all the commands run by the program. This background record is called the Journal. SPSS creates this record all of the time and most of its users are oblivious of this silent-running part of the program. The Journal becomes a valuable record of the SPSS commands that have been executed. Like a great many other things in life, its usefulness can be difficult to appreciate – that is, until you need to use it.

To show you how the Journal looks, a small section from a Journal is set out in Syntax 1:1 below, with a brief outline of its contents. The Journal was produced when one of the SPSS sample files was opened (demo.sav) and a frequency analysis carried out for the variable **marital**. There was no intentional input to the Journal or awareness of the record being made; the Journal just produced and stored this record automatically.

Syntax 1:1 Journal entry for opening a file and running a frequency

The SPSS sample dataset demo.sav was used.

```
Mon Nov 10 14:38:16 2005: Journaling started
GET FILE='C:\Program Files\SPSS\tutorial\sample files\demo.sav'.
```

```
FREQUENCIES
VARIABLES=marital
/ORDER= ANALYSIS.
```

The above excerpt from a Journal identifies the date and time of the work, the name of the file being worked on, and the analysis carried out. These tasks were all carried out in the more usual drop-down menu, button-driven method of using SPSS, but behind the scenes the Journal stored the actual computer program language responsible for the actions.

The first line `Mon Nov 10 14:38:16 2005: Journaling started` simply tells you the date and time that the Journal was begun.

The second line is an SPSS command `GET FILE` and an equals sign followed by the filename and location (enclosed in inverted commas) – then because this is the end of the command it is completed by a full stop.

The next three lines are the command `FREQUENCIES` and the keyword `VARIABLES` and sub-command `/ORDER` for the frequency that was requested. Again you will see a full stop after the last sub-command; a full explanation of the `FREQUENCIES` command can be found in Syntax 8:1.

Whenever I do any SPSS work with someone I check that the Journal is set up to record their actions as they use SPSS. As the Journal can keep an ongoing record of your SPSS activity, it is important to optimise the Journal settings to reduce the risk of losing that record. The value of this may not be immediately apparent to the novice, especially as the Journal may not make much sense initially. However, even if you never use any other aspect of SPSS syntax, the Journal can prove to be a useful tool if you have a problem with your analysis that you cannot understand and are seeking more expert advice. If your advisor or supervisor is a user of syntax he or she can access the Journal to review retrospectively actions such as how any derived variables were created.

Example of Journal Usefulness

The most notable example of the Journal's usefulness I have experienced was a doctoral student nearing the end of his research who suddenly realised that he had wrongly sorted his data in Microsoft Excel prior to moving it across to SPSS. This had resulted in the wrong data being associated with the wrong individual, for at least half of the variables. The analysis to date was nonsense, the values calculated for the derived variables incorrect. The student was panic-stricken, seeing the last six months work down the drain and anticipating months to rectify the situation. His supervisor recommended that he come to see me and see if there was anything that could be done. The student was not a user of syntax.

4 Using SPSS Syntax

However, at the beginning of his studies his Journal had been set to store the commands he used, as an ongoing process, and to a safe location.

Thankfully the student had a copy of the original (unsorted) Excel file of his data, which could be imported into SPSS. Through using the student's SPSS Journal it was possible to:

- identify all of the commands used to create the required derived variables, and to create a syntax file to re-make them;
- to identify all of the analysis commands required to replicate the analysis reported in the results section of his thesis, and to create a second syntax file to re-run the analysis.

This sifting of the Journal took about a week, the re-creation of the variables and the re-running of the data each took a further week. The student was able to continue with his doctoral studies with much less of a delay than he had dared to hope.

There are no disadvantages to using the Journal, nor to altering the settings to optimise the Journal's storage settings. The recording of the Journal has no observable affect on the running of the PC or program, nor is it apparent to the user how it has been set up unless the user checks the settings in person.

How to Set the Journal

You will find the Journal settings by going to the EDIT menu, selecting OPTIONS, and clicking on the FILE LOCATIONS tab. This is represented below using arrows to show each change in menu level.

EDIT → OPTIONS → FILE LOCATIONS

The Journal can be set to be saved into a specific file location and there are options either to **Append** commands at each session creating a cumulative record or to **Overwrite** and create a fresh Journal each time you open SPSS. You should set the Journal to append commands thus saving a complete and ongoing record.

1. Click on the drop-down EDIT command, and select the OPTIONS option EDIT → OPTIONS → FILE LOCATIONS (see Figure 1.1).
2. The box for 'Record syntax in Journal' should be ticked already, if it is not then tick this box now.
3. Ensure that it is set to append and not overwrite (append continually adds the new syntax activity to the Journal with each use, overwrite instead starts a fresh one each time SPSS is opened and wipes the previous one).
4. Set the file destination. Recommendations vary depending on your user situation (shared PC, own machine, networked or not).
5. Click APPLY and then OK.

You can find out where your Journal file is currently being stored either by looking in the SPSS options section, or by doing a file search for a file with the **.jnl** suffix.

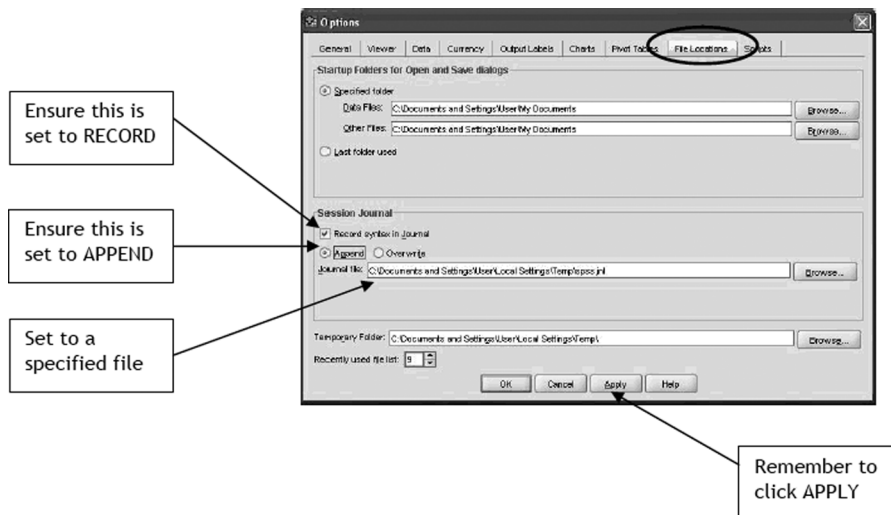


Figure 1.1 The FILE LOCATIONS tab in the OPTIONS dialogue box accessed via the EDIT drop-down menu

How best to set up the Journal depends on whether you are on a single-user PC or whether you are on a shared, multiple-user PC (e.g. in a university communal computer room). What you do need to do is identify the safest folder in which to store the Journal and set the save option to that folder. There is a default setting within SPSS, and that is for the Journal to be stored in the Windows Temp folder. There is no problem with that per se if you are on a single-user PC, but I recommend that the Journal be set to a specific folder rather than mixed up with all the temporary PC files, which may accidentally be deleted during a tidying-up or emptying of these files. To keep it straightforward I suggest the folder be called something really clear like SPSS_Journal and stored as a primary folder on your main drive, rather than be hidden as a sub-folder within a sub-folder. If you are on a shared, multiple-user PC in a university setting you may be fortunate enough to have access to a small bit of server storage space, in which case you should create a folder on the drive to which you have access to save data onto.

An issue with any shared, multiple-user PC is that you may need to remember at the beginning of each session to reset the target folder for the Journal to your specified folder. This is because shared PC systems are frequently designed to reset basic program settings to a default at the beginning of any individual's session, a necessity to prevent users having to check each time how the programs have been left by the previous user.

Syntax and Journal files are, in computing terms, small. For example, I am a heavy user of SPSS and my Journal for 14 months was 2.6 MB – and that was made up of enough syntax to fill more than 1000 pages' worth of text. For those using a shared, multiple-user PC without access to server space, then it may be advisable to save to a USB drive if necessary. Again you will need to remember at the beginning of each session to reset the target folder for the Journal to your specified folder.

As mentioned earlier the Journal is an automatically generated, ‘behind the scenes’ record. If you want to examine the commands recorded in the Journal then you explicitly have to locate the journal and open it in order to see them. What can be more useful (in terms of learning) is to have a Log of the syntax in the output viewer. That way you can see each command recorded immediately preceding the resulting analysis in the output file.

The Log – What it Is, and How to Set the Log Options Appropriately for Syntax Use

The Log is a useful way to get started with the syntax commands, enabling you to get used to how they look without having to use them, and a very useful way to start the audit trail of your data analysis.

The Log is a facility in SPSS for the commands to be documented on the output sheets. This is achieved by SPSS printing on the output sheet the actual command language used ‘behind the scenes’ to carry out the drop-down, menu-driven commands.

When you use a computer package via the more usual drop-down, menu-driven graphical interface there is rarely a need to consider that, behind the scenes, this method of use still relies upon the *computer language* operating behind the interface. It’s a bit like driving a car, in that we do not usually need to consider the detailed workings of the internal combustion engine, electronics and so forth. To start a car the requirements can simply be to understand what you need to do to make the car perform (e.g. the key in the ignition), rather than understanding how the engine starting is actually achieved (umm ... sorry, but despite driving for more than 20 years I do not actually know what happens to make the car start so cannot illustrate this bit; see what I mean?).

The Log can be easily set to be displayed in the output sheets.

1. Click on the drop-down EDIT command, and select OPTIONS. The GENERAL tab is usually the view that you will have as the OPTIONS dialogue box opens so you will need to click on the VIEWER tab, see Figure 1.2 (EDIT → OPTIONS → VIEWER).
2. Tick the box in the bottom left labelled ‘Display commands in the Log’.
3. Click APPLY and then OK.

The default in SPSS is for the Log not to be set to show in the output files. Thus if you do not actively set the Log and if you were to open the SPSS sample dataset demo.sav and carry out a frequency analysis of the **marital** variable, the output would look as in Figure 1.3.

If, conversely, the Log had been set to show in the output sheet (by ticking ‘Display commands in the Log’ in the edit options described above and shown in Figure 1.2) then the output would look as in Figure 1.4. Here you can see the actual commands that were generated (behind the scenes) when the frequency option was selected and run from the drop-down menu.



Figure 1.2 The VIEWER tab in the OPTIONS dialogue box accessed via the EDIT drop-down menu

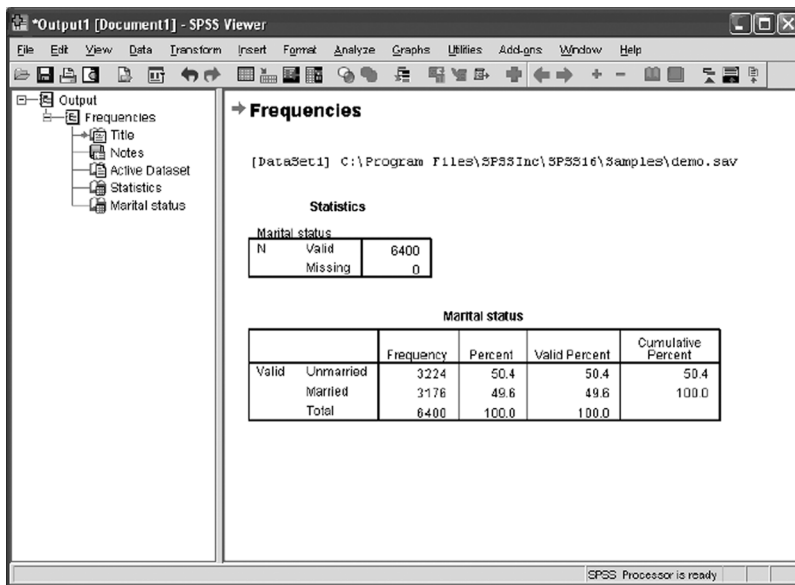


Figure 1.3 Analysis of frequencies with no Log set to record on the output viewer

The results of the analysis are preceded by a Log of the command that performed the analysis. The inclusion of the commands is useful to see in the output for several reasons:

1. If you return to your analysis after weeks or months you can see: how you have created new variables; carried out the analysis; which cases you have selected; and so forth.

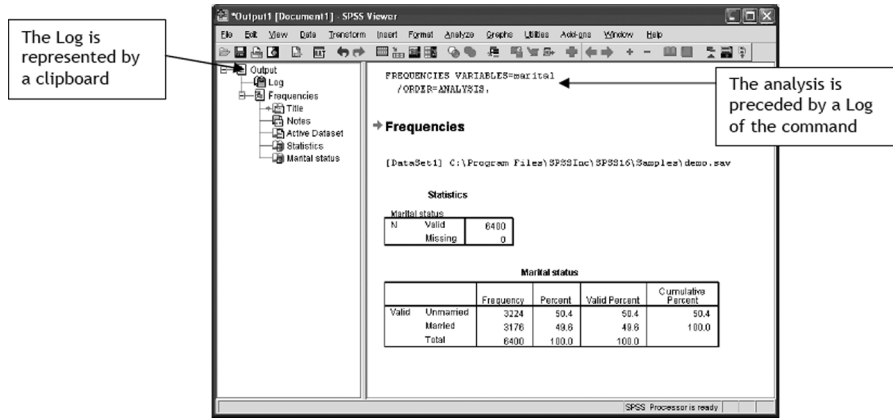


Figure 1.4 Analysis of frequencies with Log set to record on the output viewer

2. You can print off your output and show it to someone else (e.g. your supervisor) to go through your results. He or she can instantly see the commands that you have executed, helping him or her to identify whether you have created your derived variables correctly and carried out the correct analysis – this can be particularly useful when you are using the SPLIT FILE or SELECT CASES commands, or when complex multivariate analyses have been used.
3. Viewing the Log in the output files allows you to gradually get used to seeing how the syntax looks.

Example of Log Usefulness

One of the Log's most beneficial uses is when you wish to discuss an output file with another user – especially your supervisor. The Log clearly allows the other person to see what you have done, from creation of derived variable to details of analysis. It is especially useful for exploring the analysis and rapidly being able to identify reasons for 'inexplicable' results or changes in the numbers of cases included in an analysis. For example, use of the Log allows easy identification of whether the SELECT command may have been left on, or the file may have been SPLIT and that action then forgotten. This identification is sometimes quite difficult to do without use of the Log.

Syntax Files

So far we have not really altered the way that you use SPSS, only altered the record-keeping associated with your use of it. However, to gain greater benefit from SPSS we need to explore use of a new SPSS file type, a syntax file. Syntax files start as 'blank sheets' on which SPSS commands are then entered and which are known as a syntax editor. When saved they become syntax files and have an .sps suffix.

The syntax editor can be opened in several ways: (1) via the drop-down menu, (2) by changing the default settings to always open a new syntax file each time you

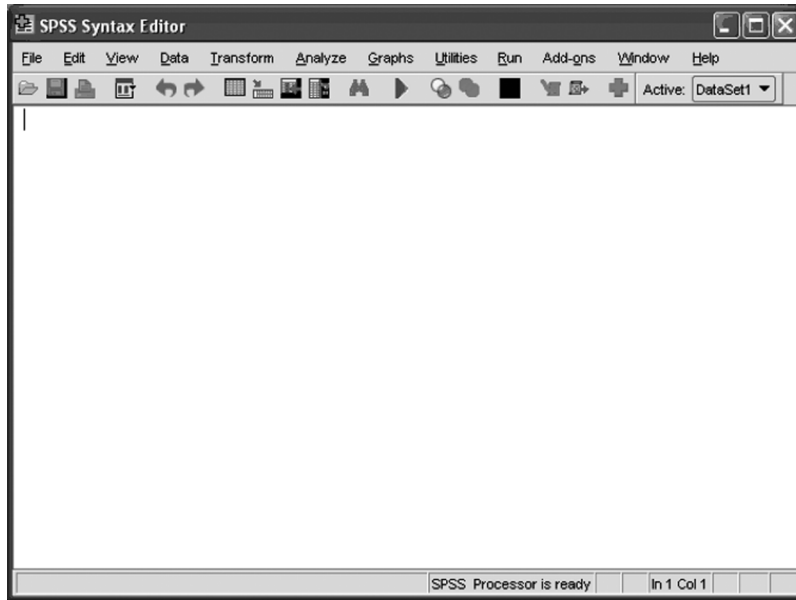


Figure 1.5 A blank syntax editor

open SPSS, or (3) through use of the paste button in most of the dialogue boxes obtained through the analysis, transform or data menus.

Methods for Opening the Syntax Editor

Method 1 – Directly open a new syntax editor via the drop-down menu FILE → NEW → SYNTAX.

Method 2 – change the default settings so that SPSS routinely opens a blank syntax editor whenever it is opened. You only need to do this once and the settings are stored for future occasions that you use SPSS.

1. Click on the drop-down EDIT command, and select the OPTIONS option (EDIT → OPTIONS → GENERAL). The GENERAL tab is usually the view that you will have as the OPTIONS dialogue box opens (see Figure 1.1).
2. Tick the box in the bottom left labelled ‘Open syntax file at start-up’.
3. Click APPLY and then OK.

Method 3 – Use the paste button available through many dialogue boxes which are accessed via the drop-down menus. If a syntax editor is already open then the command will be pasted onto that one and a new one will not be opened. However, if there is no syntax editor already open, clicking on the paste button will open a syntax editor and automatically paste the command onto the blank editor. Syntax editors opened via the paste function differ from those opened through the first two methods outlined above, in that they will not be blank; instead they will open with your initial command already recorded.

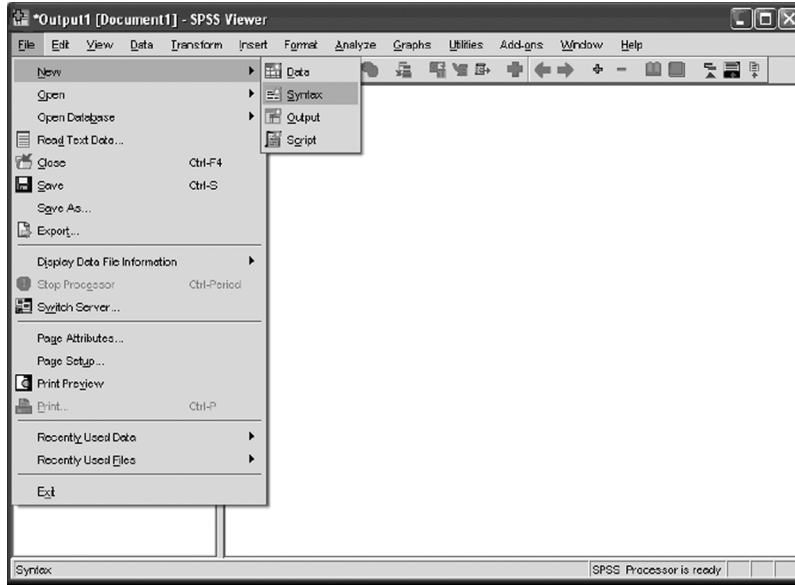


Figure 1.6 Drop-down, menu-driven method of opening a new syntax editor

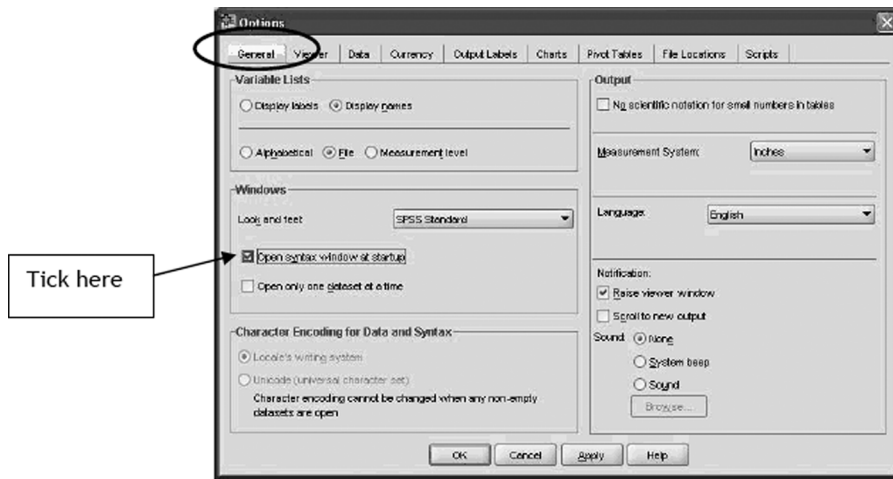


Figure 1.7 Setting up the option of opening a new syntax whenever SPSS is opened

In the example below we will again use the SPSS sample dataset demo.sav and carry out a frequency analysis of the marital variable. The frequency function is accessed via the drop-down menu ANALYZE → DESCRIPTIVES → FREQUENCIES. Once in the Frequency dialogue box, select the variable required for the analysis, move into the variable box and, instead of pressing the OK button as usual, press the PASTE button (see Figure 1.8). Note that, just like the OK button, the PASTE button remains 'greyed out' until you are in a position to complete the command and

execute it, for example until variables have been moved into the correct boxes within the dialogue box.

Once the PASTE button is clicked, the syntax file will open with the command written on it (see Figure 1.9). For more information on syntax for the frequencies command you can refer to Syntax 8:1 in Chapter 8.

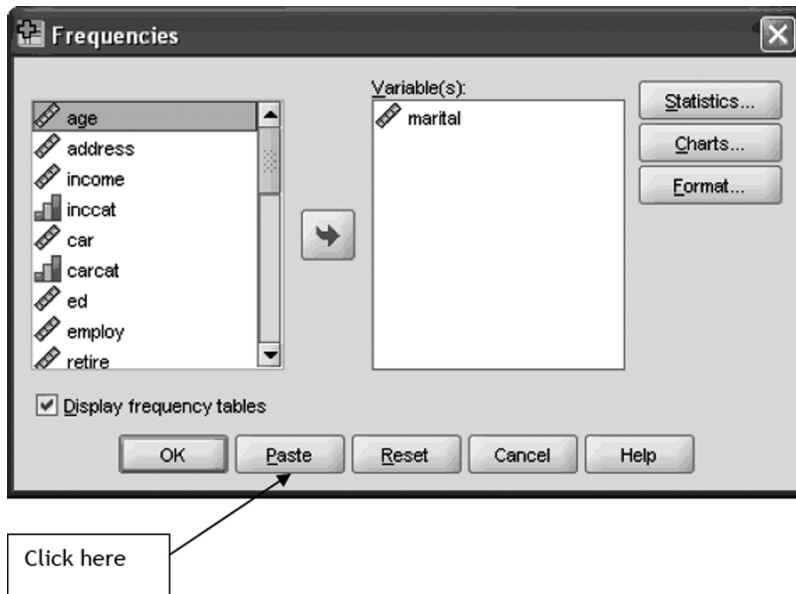


Figure 1.8 Frequencies dialogue box

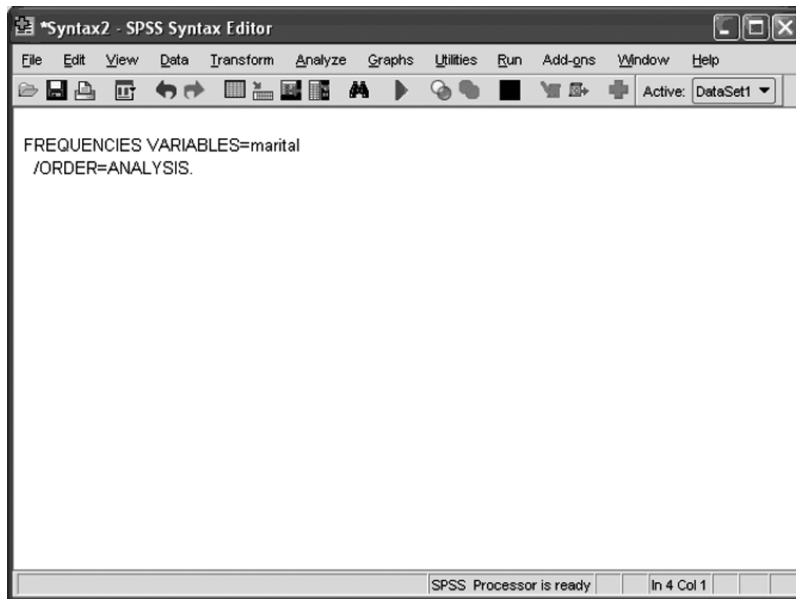


Figure 1.9 Syntax editor created from the dialogue box in Figure 1.8

Confusion Alert

It is important to note that pasting the command into a syntax file will not run the command, and the output sheet will remain blank. This is a small point but one that can easily throw new users of syntax, as they press the PASTE button and yet no analysis/transformation etc. takes place. People sometimes wonder what has gone wrong, but it is simply that there is just one more step to take, that of *running* the command. If you have just pasted the command onto the sheet and want to run it immediately, all you need to do is check that your cursor is positioned in that command line and then press CTRL+R. Alternatively you can (1) use the drop-down RUN menu, or (2) press the small RUN button (▶) towards the right of the toolbar (see Figure 1.9).

Writing a Syntax File Using the Log

The next step in the use of syntax is the writing of syntax files with a series of commands, which together carry out an analysis plan or a section thereof.

In the early stages there are two ways in which the beginner can start to write syntax files and neither requires you to know any computer language, nor to know how to compile any SPSS syntax commands. For both of these methods you carry out the analysis as usual but either you (1) work through the analysis but rather than press the OK button you instead press PASTE to put the command on the syntax editor, then go to the syntax editor and press RUN; or you (2) copy from the Log in the output file and paste into a syntax editor.

Both methods are useful and may be as far as the early user wishes to go for some time. Creating such a syntax file can be beneficial as it creates a file that stores the commands for an analysis plan, or part thereof. This in turn is often useful, for example, either simply as a record of the analysis carried out, or for use if you wish to repeat the analysis, perhaps because you have more data or because you want to repeat the analysis on a sub-set of the data. For this reason this type of syntax file will be referred to as a 'repeat analysis syntax file'.

Example of Log Usefulness

The Log's usefulness in assisting with the creation of a repeat analysis syntax file can be seen if you consider a scenario where you have carried out analysis on the data as a whole and then, perhaps unexpectedly, require a repeat of that analysis (e.g. if a batch of questionnaires is returned late). If the Journal is set up correctly you can use that to find the commands required. However, the Journal may possibly have other commands in there, from work on other datasets and projects. In the output sheets that you have saved and reviewed, it should be relatively easy to identify which of the analyses you wish to repeat. Also, the output sheets are more 'viewer-friendly', and a supervisor or colleague can look through the results on the data output sheet, identifying the data to be re-analysed. You then take the Log associated with that analysis, and use this to create a syntax file. Creation of syntax files is covered in more detail in the next section, but fundamentally the

process to create a 'repeat analysis' syntax file can simply involve copying the Log associated with the required analysis and then pasting into a syntax editor.

Creating a Repeat Analysis Syntax File

One of the easiest ways that you can create a repeat analysis syntax file is by copying and pasting the Log from an output file.

1. Open the output file (see Figure 1.10).
2. Open a new syntax editor (FILE → NEW → SYNTAX).
3. Select each of the Log sections in turn and copy and paste them into the syntax editor (see Figure 1.11).
4. Save the syntax file.

The three Logs seen in the output (see Figure 1.10) are copied and pasted into the syntax editor (see Figure 1.11). All you need to do now is to save the syntax file, preferably with a meaningful name and in an appropriate place. I suggest that the syntax commands be saved in groups, such as 'project A_deriving variables', 'project X_descriptive syntax', 'project Y_regression analyses' and so forth. These files keep a record of the specific analysis process involved.

A similar syntax file would have been created if you had worked through the analysis pressing PASTE and then RUN each time rather than the OK button. Note that the Log can sometimes have extra commands in it that are auto-generated, for example the GET element of the above syntax file was found in the first Log of the output file. A syntax file created using the PASTE and RUN method for individual commands would not have this command as it is an auto-generated one, created when the datafile was opened and recorded onto the output automatically. This was

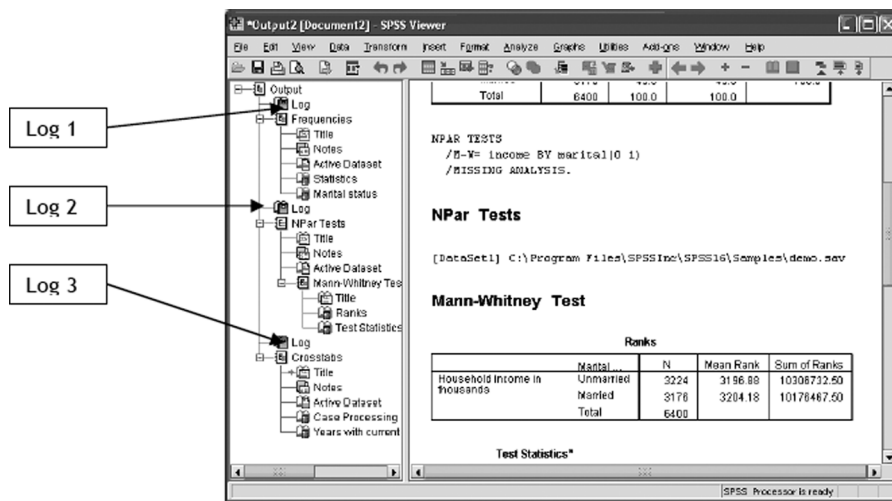


Figure 1.10 An output file from analysis of demo.sav

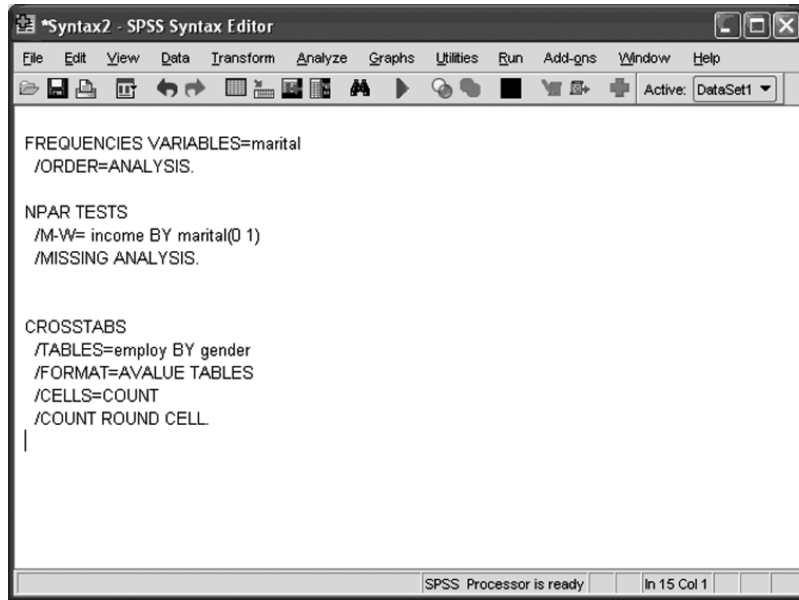


Figure 1.11 Syntax file created by copying and pasting the Logs in Figure 1.10

included via the copy and paste simply because SPSS continues to write commands consecutively within a Log until some other form of output is generated, for example, tables, graphs or analyses.

If you copy and paste a Log into a syntax file and then find that it contains extra commands that you do not want to include, you can simply delete the unwanted text. Just bear in mind that even small changes to text within the syntax files can have unexpected effects.

Confusion Alert

SPSS syntax, in line with other computer languages, is very particular, and relies on correct spellings, placing of spaces, full stops and keywords. During the learning processes required for SPSS syntax most of my students have, at some point, come to me for advice about some syntax that just will not work. Often they have been trying to rectify the syntax for some time, only to leave my office after just one or two minutes muttering under their breath about full stops or other small but crucial characters that they had missed or misplaced and failed to notice. SPSS is not being overly fussy, though; after all, if you misdial a phone number by one digit you should not expect to be put through to the correct number.

Summary

SPSS syntax is the name given to the computer programming language that is the basis of all SPSS functions. Use of SPSS syntax can be employed to complement the drop-down, menu-driven method; it does not have to be an either/or choice.

A behind the scenes record of the commands used in SPSS sessions is called the Journal and this can run without the user being aware of it. SPSS can be set up to show an ongoing report (Log) in the output viewer. The Log records the commands being used via the drop-down, menu-driven method. An introduction to syntax can be gained by reading these Logs displayed on the output sheets. A repeat analysis syntax file can be created by using the paste function from commands recorded in either the Journal or the Logs recorded in an output file.

Use of syntax, passively through the Journal or Log, or actively through creating syntax files, allows an audit trail to be set up to identify:

- Which file was opened to carry out the work on (not difficult if you only have one datafile for your data, but some people have been known to have old and new versions of details – or even datafiles that have some but not all participants, or some but not all variables. This is generally not to be recommended (understatement) as this can lead to errors. However, at least the Journal and Log will enable you to check back should queries arise.
- How variables are created.
- What variables are in the analysis.
- What analysis has been carried out.
- Whether certain cases have been selected prior to running analyses.

Useful Tips

- Save your Journal periodically, with a new name indicating the time period covered, and begin a new one.
- Get started by reading the Log associated with each of the analysis or variable creation activities carried using the drop-down menus – see if you can work out what it says and how it has resulted in the action you see.
- Name your syntax sheets with meaningful names.
- Save syntax sheets alongside the project data to which they relate.
- Read the Log associated with any error reports – see if you can work out what it says and how it has resulted from the actions preceding it, and in the action you have seen. Understanding error messages is addressed in Chapter 13.

SPSS Conventions

Note the following conventions which are used in this text and are designed to produce syntax files that reflect the style recorded in the SPSS-generated Log and Journal. Maintaining SPSS conventions assists ease of reading and navigating through a syntax file as experience increases.

- S.1. The ‘case’ of the text is not usually important in SPSS syntax. However, good use of upper- and lowercase text can make reading through the syntax much easier. I recommend that you follow the same format as the Log and Journal, with the commands and sub-commands in uppercase and variable names, labels, etc., in lowercase.
 - a. Case does not matter for variable names. In syntax the name of the variable ‘AGE’ is the same as ‘Age’ and ‘age’.
 - b. When referring to text in string cases then case is important (see Chapter 5 on string variables). SPSS will treat ‘Male’ as different to ‘male’ and different to ‘MALE’.
- S.2. Each command must start on a new line. For ease of reading, in this book I have placed commands on the extreme left with no spaces and any following sub-command lines are indented.
- S.3. Sub-commands should start with a forward slash /.
- S.4. Each command is completed by a full stop (the command terminator). This needs to be at the end of the last line of the command (or the sub-commands if present). SPSS will also count a blank line as a command terminator if there is no full stop.
- S.5. Before a command may be executed the word EXECUTE is sometimes required, again followed by a full stop. It is mainly the commands that alter the datafile in some way that require this EXECUTE command to be run.
- S.6. Variable names must be unique within any dataset: there cannot be more than one variable with the same name. Up to SPSS version 14 the variable name could be a maximum of eight characters long. From version 14 onwards this was raised to a maximum of 64 characters long. Variable names must not begin with a number, a space, a hyphen, forward slash, backward slash, exclamation mark and so on. Underscores are permitted anywhere. Full stops are permitted only in the middle of a variable name, not at the beginning or the end.

Numbers are permitted providing they are not the very first character of the name.

- S.7. In this book I have added comments to syntax files. You can start a comment with the COMMENT command or with an asterisk. See Syntax 2:7 for further details.