3

GETTING STARTED WITH SPSS

CHAPTER SUMMARY

This chapter introduces the main windows of the SPSS software environment: the Data Editor, Output and Syntax Editor windows. We look at how concepts about particular cases (e.g. countries, individuals) may be represented in a form appropriate for statistical analysis and ways to handle missing information. This chapter gives key points to keep in mind when working with your data files and provides general instructions for inputting and importing data. The final section of this chapter introduces, and later critiques, a very common approach to classifying variables, also referred to as 'levels of measurement'. We discuss how variable classification may be used to help inform the kinds of analysis that are appropriate.

OBJECTIVES

In this chapter, you will learn:

- What primary data is and how to put it into SPSS
- The key functions of the Data Editor, Output and Syntax files
- How to use syntax to adhere to good statistical practice and save yourself future work
- The most common way of classifying variable types (levels of measurement) and its shortcomings.

INTRODUCTION

IBM SPSS Statistics ('SPSS') is a software program that is designed to analyse data. It has a long history, dating back to the 1960s, and has long been the most popular choice for academic researchers needing to analyse data and to teach students how to do so. There are alternatives, with other common packages including Stata and R. The latter is becoming more widespread because it is an open-source program, but it requires command-line input or an overlaid user interface, making it much less user-friendly for learners who are not confident in computer programming. SPSS requires a paid licence, but many universities and employers use it because of its relative ease of learning and menu-based 'point-and-click' interface. Some of these menus – such as File, Edit and View – will be familiar to users of other common software packages. This chapter provides you with a first introduction to the main files and windows used in SPSS and introduces the statistical concept of levels of measurement, a system for identifying the type of data you are working with.

Working with data in SPSS will normally require you to work with two or three files simultaneously. The main file will be the data file (file extension .sav), which contains all of the data and variable information. In addition, any outputs that you create – such as tables and charts – will appear in an output file (file extension .spv). The other main file type is the syntax file (file extension .sps), which keeps a record of the commands you use to manipulate the data. The syntax file is not essential for SPSS to function, but learning to use the syntax file, even at a very basic level, can save you a lot of time in the long run by allowing you to alter and reproduce your outputs very quickly. We will talk through the two default windows in SPSS in this chapter: the Data Editor and Output. We will talk through the Syntax window in the next chapter.

A note of caution: SPSS does not auto-save your work. Whenever you have made changes to a file, an asterisk (*) will appear at the top by the file name. If this change was unintentional, you can close the file without saving to erase the changes. Each window in SPSS represents a separate file, and each file must be saved manually. As we will emphasize throughout this book, you should save regularly, always have back-ups, and seek to create new variables rather than changing the original data. Conscientious use of the syntax document will also allow you to reproduce any changes or analysis very quickly if you do have to revert to a previous version of the file.

DATA EDITOR

The main window of SPSS is the Data Editor, which contains your data. The data file (.sav) is the main file where all of your data is saved. When you open a data file, it opens in the Data Editor window. If you close this file, it will exit SPSS. Most of the secondary data sources discussed in the previous chapter make datasets available that are already in SPSS format (file extension .sav), including all three datasets used as examples throughout this textbook. In this case, you simply need to open the secondary dataset and start working. You can open an SPSS data file by browsing to it on the start-up screen when you first open the program or File > Open > Data within the main program. For the rest of this chapter, you should download and open the ESS round 7 dataset so that you can follow along with the examples.

The Data Editor window has two viewing tabs: Data View and Variable View. You can toggle between these in the bottom left-hand corner of the data window. You will use the Variable View tab the majority of the time when working with your dataset. In Variable View, each row represents one *variable*. A variable is a characteristic of a case, such as marital status, income or age, if talking about individuals; or size of a country's economy, number of immigrants or unemployment levels, if talking about countries. Variables must have possible answers that can differ from each other (can vary). For example, if you are using data from the UK Census, all responses are from the UK, so having this as a variable would not make sense. However, you could have a variable representing the countries that constitute the UK (England, Scotland, Wales and Northern Ireland) because this does vary between different respondents. If using survey data, each variable normally represents the answer to one question. If, however, a question allows respondents to choose more than one category, the responses to a single question may be recorded as a series of variables, with one variable per response option. This is because it is not possible to record more than one value for a respondent in each variable. For example, if respondents are asked to indicate which newspapers they read at least once per week, they could tick more than one box. When this is turned into variables, it would be recorded as a series of yes/no responses, with one variable for each newspaper.

In Data View, each row represents one *case*. A case is the smallest unit of analysis in your dataset. If you are using data that came from a survey questionnaire administered to individuals, such as the ESS data, each case represents an individual's set of responses. This means that each row gives one respondent's answers to each of the survey questions. In some surveys, a case would be a household, with an entire household's responses recorded on a single questionnaire, such as census questionnaires. If you are using other data, such as the UN composite data, each case represents one country. Each case will normally have a variable with a unique identifier, such as a randomly assigned number, or the country name if countries are the unit of analysis. Although this is not strictly necessary for the file to function, it is good practice because it allows you to identify cases that might contain errors and re-examine the original questionnaire if you have input the data yourself.

Variable View

The Variable View tab gives you an overview of all of your variables, including their text labels and measurement type (discussed in more detail later in this chapter). Each column in Variable View gives you different information about a variable.

The variable *name* appears in the first column; this is the name that appears in the column headings in Data View. The names are usually kept short for ease of use in writing syntax, and they are governed by strict rules that require them to start with a letter and proscribe spaces, punctuation and many special characters (though $, # and @ are allowed if they are not the first character). The names cannot be more than 64 characters long, must each be unique and cannot be SPSS reserved keywords (ALL, AND, BY, EQ, GE, GT, LE, LT, NE, NOT, OR, TO and WITH). In some datasets, the name will represent a short form of the variable label, such as 'trstprl' as the variable name for 'Trust in country's parliament' in the ESS dataset. In other datasets, the variables are simply consecutively numbered, such as var001, var002, var003, etc. If you are inputting your own data, we would recommend creating names that are linked to the variable content, as this will make it much easier to work with the syntax. SPSS is not case sensitive for variable names, even though it will store any upper- and lowercase formatting applied by the user. However, other statistical packages – such as R and Stata – *are* case sensitive, so you should use caution in naming your variables. For example, in SPSS, Sex and sex would be interchangeable, but in other systems, they would operate as two separate pieces of information. Capitalization is also important when using string data.

The next column in Variable View is *Type*. This reflects whether the data takes the form of words or numbers. String means that the variable content in Data View is composed of words or letters. String format is more likely to apply to answers to open-ended questions in questionnaires (i.e. questions that do not have a set of answer choices but prompt respondents to write in a text box). String data should be used very carefully in SPSS and can present problems with auto-coding. Numeric means that the data has been entered as numbers. This can be because the data itself comprises a number, such as the number of children a respondent has or the year someone was born; but it can also be because numbers have been assigned to represent words to expand the possibilities for analysis in SPSS, such as 1=female, 2=male, 3=non-binary, etc. This is called *coding*, which we will look at in greater detail below.

The *Width* and *Decimals* columns provide information about how many characters are reported for those variables. The default width is 8 characters, which is sufficient for most numeric data, but this can be adjusted for string data or large numbers – for example, reporting the GDP for the United States would require 14 characters. The default number of decimal places is two, but this may be irrelevant depending on the units of a particular variable. For example, if you report the number of children in a household, children would be counted in whole numbers, so you would not need any decimal places. Decimal places are also unnecessary when the numeric coding represents word-based categories, such as political party preference.

The *Label* column gives a more detailed title for the variables. Unlike the variable name, the labels can contain spaces and are written in coherent prose. The label should give enough information about the variable such that you would not routinely have to consult the questionnaire to get an idea of the variable contents. For example, there is a variable in the ESS dataset with the name `etapapl`, which does not enlighten us about its contents; but its label is 'Easy to take part in politics', which makes it much clearer.

The *Values* column tells you what words are paired with the numbers recorded in the dataset. This pairing of numbers to represent words is known as *coding*. Coding is the process of attaching a combination of numbers or letters to represent each possible value for that variable to turn it into a numerical value for analysis. It is common for numbers to be attached to particular values of variables, even where the underlying data is not numeric. It is possible to leave the data as simple text (or 'strings'), but analysts tend to prefer variables to have an underlying numeric construction because it widens the possibilities of analysis. A common exception is the labelling of countries and regions. For example, in the ESS, the variable for the country where the respondent was interviewed [`cntry`] is formed in standardized two-letter abbreviations. In this system, AT means Austria, BE means Belgium, CH means Switzerland, etc. These codes are usually based on one of the international standards, such as ISO 3166-1 alpha-2 (two-letter country codes), alpha-3 (three-letter country codes) and numeric (three-digit country codes).

The majority of variables will be coded to use numbers to represent text. For example, in the ESS, there is a variable for how interested the respondent is in politics [`polintr`], where '1' means (is coded) 'Very interested', '2' means 'Quite interested', '3' means 'Hardly interested' and '4' means 'Not at all interested'. Sometimes there are labels for all numbers, such as the respondent's religious denomination, where all word-based categories have been coded with numbers for SPSS analysis. For variables where the responses run on a continuum, there are sometimes only labels for the two extremes of the continuum, such as How satisfied with life as a whole [`stflife`], where 0=extremely dissatisfied and 10=extremely satisfied, with no labels in between. Some variables, normally where the numbers reported actually represent numbers and not words, only report value labels for missing answers, such as Age when completed full-time education, United Kingdom [`edagegb`], where the only labels are for respondents still in full-time education, refusals, don't know and no answer. You can open the full list of value labels for any variable by clicking on the relevant cell in the Values column, then clicking the triple-dot button that appears on the right of the box. This is also the dialogue box you will need if you want to edit, add or delete value labels using the menu-driven approach.

The *Missing* column identifies which variables SPSS treats as missing in its analysis. This affects how it processes the data. Within a dataset there are often 'gaps' of various kinds, where we might expect there to be data on a particular case, but there is none. Missing values can arise for several reasons. First, data may be missing because that piece of information is simply not relevant or applicable. An individual who is not working will not have a current wage rate; an individual who was not old enough to vote in the last election will not have information about the party they voted for. This kind of information may be said to be structurally missing. An important type of missing data is found in datasets that track people over time (called panel studies), like Understanding Society, where people stop taking part over time. This is generally known as attrition and may arise because people no longer wish to take part (an active refusal) or have moved and cannot be located (non-contact). Such longitudinal datasets may lose a fair proportion of people to attrition at each wave, and judgement is needed about how to adjust for such losses in any analysis.

Where an entire case is missing, such as arises with attrition from longitudinal data, this is known as unit non-response. However, in most datasets the data collection is one-off (a cross-section) and missing data may arise as item non-response, or data that is missing for particular questions in the case of surveys. This kind of missing data may occur because people either do not want to answer the particular question (a refusal), or are unable to answer because they lack the relevant information (a 'don't know' response). There are also longitudinal datasets, such as the ESS cumulative datasets, where the data from several cross-sectional surveys asking a common core of questions has been combined into a single dataset to allow analysis of trends over time. In these cases, there will be variables with missing data because some questions were not asked in every round of the survey.

Missing responses are also likely to be higher for sensitive variables, such as on household savings or sex lives; in these cases, we would expect to encounter higher rates of respondents refusing to answer the question, especially if the survey was administered face to face. For this reason, sometimes such data is collected in less overt ways, such as using a self-completion questionnaire within an overall face-to-face interview. Questions about precise quantities or on topics not frequently considered may also lead to 'don't know' responses. This might occur, for example, when asking a respondent what their earnings were two years ago; trying to collect the incomes of those working as self-employed is particularly prone both to refusals and to 'don't knows'.

You may decide as you work with it to change whether an answer is treated as missing or not. For example, if you wanted to look at political party identification, you might be interested if a large proportion of respondents reported 'don't know', so you might decide that you wanted to include those answers in the main analysis, rather than excluding them as set by the missing values in the ESS. Missing values are normally coded as negative values (–1 through –5, for example) or high numbers (77, 88 and 99, for example) in order to avoid confusion with any of the 'real' responses. As with the value labels, you can open the full missing values dialogue box by clicking on the relevant row in the Missing column, then clicking the triple-dot button that appears on the right of the box. This allows you to see which values or value ranges are identified as missing and allows you to modify these, if needed.

The other main column of interest is *Measure*. This is where the variable type is classified. We will discuss more about type-casting variables and the potential pitfalls of following the classic approach used in SPSS later in this chapter, so here we will just look at how this column is laid out. If you click on one of the answers in the Measure column, you will notice that a drop-down menu appears with the options Nominal (symbolized by three circles), Ordinal (symbolized by three escalating bars) and Continuous (symbolized by a ruler). The measure recorded does matter, especially for the SPSS Chart Editor, so there may be cases when you need to alter the recorded type. You may also find that publicly available survey data does not record the variable type at all, leaving all variables as the default (nominal). This may cause problems in your analysis, so make sure that you check the measure if you receive an error message, or if your results do not look as you expected.

If you have too many columns to fit easily on your screen size and find that you do not frequently need to use some of them (such as Width, Decimals, Align or Role), you can choose not to display the unnecessary columns using View > Customize Variable View. Simply untick the columns you do not need to view. If you change your mind, you can return to that menu and click Restore Defaults.

Data View

The other view in the main Data Editor window is Data View. You will spend most of your time in Variable View, only entering Data View to input data or if your results are surprising, leading you to check whether there is an error in the data. You should be very careful in Data View, as it is very easy to accidentally change the results by typing in one of the cells. The Data View tab shows the data in a format similar to a spreadsheet. Each column in Data View represents one variable, while each row represents one *case*.

You can toggle the display of the variable responses to show either the alphanumeric coding or the value labels. You can do this by using the menu sequence View > Value Labels or by clicking on the icon with 1 and A in the icon ribbon at the top of the screen. Watch how the value labels appear and disappear as you toggle this on and off.

If you are starting from scratch to input your own data, say from a survey that you have conducted, there are good reasons to type this directly into SPSS. This can be more efficient because you can apply all of the appropriate labels and measures as you go. If you are entering your data manually into SPSS, you will need to create a new data file (File > New > Data). When inputting your data, you can take one of two approaches: you can either create a *codebook* before entering the data and then only enter the numbers representing the responses, or you can enter alphanumeric data (data composed of words and/or numbers) and automatically recode the text responses afterwards. A *codebook* contains information identifying each variable (such as a survey question) and possible values (such as response options to the question). When survey organizations provide datasets that are pre-formatted for statistics programs, this will normally be accompanied by a codebook, which shows you what numbers were applied to answer categories. It's also possible to produce a codebook from within SPSS for occasions when you want an overview of the coding of a variable.

The advantage of producing a coding scheme before you enter your data into SPSS, rather than entering text data and automatically recoding afterwards, is that it will ensure that the values appear in a logical order rather than an alphabetical order. To take the example of the political interest variable again, if you entered the data into the Data Editor in text form and automatically recoded it afterwards, SPSS would assign 1 to 'Hardly interested', 2 to 'Not at all interested', 3 to 'Quite interested' and 4 to 'Very interested', because this would be the alphabetized order of responses. This might not be what you would wish later when analysing your data, as responses to this question have a logical order that is not the same as alphabetical order. However, the advantage of entering the alphanumeric data and then automatically recoding afterwards is that it may be much quicker, and data entry can be much easier when you are not trying to remember the numeric values for each response to a survey.

If you decide to set up a coding scheme in advance, it may be helpful to write the response numbers onto your survey. If you have carried out your survey electronically and have the data as a spreadsheet, you may find it easier to set up your coding scheme, then use Find and Replace within each column to replace the text data with the chosen coding numbers. If taking either of these approaches, make sure that you have a back-up copy of your raw data that you can consult in case you think you have made errors in coding; and make sure that you take good notes of what numbers you have assigned to each set of values for a variable. If you decide to take the approach of starting with text data and then auto-recoding it, you will need to make sure that there are no errors or variations in spelling that would lead SPSS to separate the answers into different categories. For example, if you asked respondents which political party they voted for in the last election and allowed a free response, you would need to make sure that you didn't have some responses recorded as 'Conservatives' while others were 'Conservative Party'.

If your data is in spreadsheet or database format and you want to import this into SPSS, you can do this through File > Import Data > Choose the file type. SPSS can read many spreadsheet and database formats, such as text and comma-delimited documents (.csv, .dat, .txt, .tab), Microsoft Excel (.xls, .xlsx, .xlsm), Structured Query Language (.sql) and Microsoft Access (.accdb, .mdb). SPSS can also read other statistical software formats, such as Stata (.dta) and SAS (.sas7bdat, .sd7, .sd2, .ssd01, .ssd04, .xpt). If importing from a spreadsheet or delimited document, you should ensure that your data is laid out in a grid format, with variable names in the first row of the document and a separate column for each variable. Each row below the first row should represent a specific case. If your data is in this format, SPSS should be able to import it easily. You will probably need to do some work after importing data from other formats to attach appropriate labels to the variables and value labels, or to auto-recode string variables (also referred to as alphanumeric, which can be composed of any combination of letters, numbers and other characters) into numeric variables for analysis. The process of auto-recoding is covered in Chapter 4.

If not using auto-recode, the selection of labels may be mechanical, but it is worth reflecting upon. In practice it is most common to see people label gender as 1 being male, 2 being female – the 'second sex', as de Beauvoir put it; or marital status to have a set of codes, but with 1=Single, 2=Married, and cohabiting being lower down the list, either before or after divorce and separation. This is, of course, merely a convenient means of storing data rather than any kind of statement about the *kind* of data being represented, such as the index numbers sometimes used to classify books in libraries. However, it is still worth considering the kinds of underlying ideas that might be driving such coding, and the idea of '1' representing some kind of normality. Where data is only capable of having two values, it can be more meaningful (and perhaps progressive) to having a variable called by the name of the label attached to code '1', with others being zero – such as a variable called `female`, with 1=female and 0=male. The norm of only providing two answer options for sex is itself also something to consider if you are designing a questionnaire yourself, with estimates of 1 to 2 per cent of the population being intersex (Fausto-Sterling, 2000).

If you are inputting your own data for the first time, it is always worth looking at how other datasets have been set up to get an idea of how you might want to lay out the coding scheme as well as what sorts of variable names and labels seem to be the most helpful and intuitive. If you are new to data analysis, learning for the first time using your own dataset can be very challenging, and you will find it helpful to consult some of the many resources available online, including how-to videos, to walk you through the process.

OUTPUT

The other default window in SPSS is the output window. When you open a data file (.sav) in the Data Editor in SPSS, it will automatically create a new output file as well. The output file (.spv) opens in a separate window, usually with the default name Output1. This file provides a log of all of the actions you have taken and outputs you have produced.

Any tables and charts that you produce will appear here, along with the accompanying syntax and any error messages. The content in this file is interactive and can be edited. For example, you can activate tables to change the way the data is displayed, in terms of both layout formatting and content displayed. You can also activate charts to change their colours and other formatting. You can also use the output file to make notes to yourself, whether a note interpreting your results or reminding yourself how you went about producing that content. If you produce something that you want to discard, you can also delete any outputs that appear in the output file. All of these options make it a very powerful tool, and we would strongly recommend that you keep a copy of your outputs so that you do not have to reproduce content repeatedly.

SYNTAX

In the previous section, we looked at the default windows in SPSS: Data Editor and Outputs. The third main file type that you will encounter in SPSS is the syntax file (.sps). To open a new Syntax Editor window, click File > New > Syntax; to open an existing file, click File > Open > Syntax. The syntax file is a series of commands that tell SPSS what you want it to do. This can include everything from adjusting the labels on a variable to running complex analyses. Many introductions to SPSS skirt around the Syntax window, instead teaching an entirely menu-driven approach, but there are some very good reasons to engage early and often with the syntax file, even if you do not write the syntax yourself. You can keep a log of every menu command you perform by using the Paste button each time you carry out a task. This provides you with a record of what you have done and will allow you to replicate your previous work very quickly if needed. You can also write notes to yourself in the syntax file to remind yourself what you were doing, why you were doing it that way and what the results generally indicated. You will see, for example, that we have written some notes into the syntax files accompanying this textbook to indicate which examples and exercises each block of syntax pertains to.

Using the Paste button from the dialogue boxes will help build up familiarity with commands. As you grow more familiar with each command, producing new results with minor changes will get much faster. For example, you might use the menus to produce a table that shows respondents' UK political party affiliation [`prtclbgb`] against whether they voted in the last election [`vote`]. If you paste the syntax, then you could very quickly reproduce the same results for Austria by copying the command and replacing UK political party affiliation [`prtclbgb`] with Austrian political party affiliation [`prtclcat`]. But be aware that such pasted syntax is much longer than the minimum it needs to be, as sometimes elements are included which are the defaults, and the commands are not abbreviated in any way. For example, if you pasted the syntax for a basic frequency table for `prtclbgb`, which shows the number and proportion of respondents who selected each answer, the results would be:

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

However, not all of this information is necessary to produce the table. Writing the syntax directly, you could produce the same table by specifying:

```
FREQUENCIES VARIABLES=prtclbgb.
```

The syntax could be simplified even further with the same result by writing:

```
Freq prtclbgb.
```

However, too much shortening is not always a sensible idea and may make the syntax harder rather than simpler to read when revisiting it, and oversimplifying will also prevent helpful features like colour coding and auto-complete from activating. For example, if you start typing a command in the syntax window, SPSS will prompt you with available options using auto-complete. Using these prompts to guide you can help you to avoid error messages and will also colour code your syntax (see Table 3.1 and below for further discussion) to reflect the part of the syntax as well as highlighting commands in a navigation pane on the left-hand side of the window. If you shorten your syntax too much, this will not happen, though the syntax will still produce the desired result.

Table 3.1 SPSS syntax colour coding

Colour	Meaning
Dark blue	Main commands
Green	Subcommands
Red	Options
Grey	Comments
Black	Variable names; other text

Syntax must follow a number of key rules to work. Each block of syntax begins with a word that is the name of the command. If you start to type the name of a command, SPSS will prompt you with matches. For example, if you type Variable, SPSS will prompt you with a list of options, including variable alignment, variable attribute, variable labels and variable level. Using the command name prompted by SPSS will also cause SPSS to colour code the command in blue and bold. Main commands also appear in the left-hand navigation column of the Syntax window. In addition to starting each block with the name of the main command, there are two key items of punctuation in the syntax: the full stop (.) and the forward slash (/). Every command must end with a full stop. Full stops indicate the end of a command and are therefore very important for telling SPSS when to start and stop. When a block of syntax has failed to produce the desired result, the first error to check for is whether a full stop is missing.

The slash usually indicates a subcommand. Looking at the example above, the Frequencies subcommand identifies a series of output and formatting options (such as bar chart, format, grouped, statistics) that can be specified in addition to the main table. It is generally a good idea to use the slash with a subcommand, although it is not always needed for the syntax to function. Using a slash before the subcommand will change the colour of the subcommand to green, making it easier to differentiate from main commands. After choosing the subcommand, typing = will bring up the options for that subcommand. These are coded in red. The colours are not friendly to many forms of colour blindness, but the auto-complete feature is still useful, and the main commands also highlight in bold, making them more visible. It is good practice to start a new command (and subcommand) at the beginning of a new line. It can also be helpful to indent subcommands to create a clear visual distinction between main commands and subcommands. However, failing to observe these conventions will not usually stop SPSS from running simple syntax.

Good syntax style

Having good syntax 'style' can make it much easier for you to revisit the file later by keeping notes and maintaining good organization. It is a good idea to keep an annotated copy of everything you have done, including attempts that have not worked. One way of doing this would be to keep two separate syntax files for any piece of work: one containing all of the rough workings, including any failures and error-checking; and one containing only the final, 'clean' results. The latter is the style of syntax that we share in the materials accompanying this textbook.

You can annotate your syntax file by starting a new phrase with an asterisk (*). You will notice that the text after the asterisk, until you enter a full stop, will switch to grey. SPSS will continue to ignore the annotation as long as you do not finish a sentence with a full stop, then move to a new line. You might wish to make a note at the beginning of a syntax file that serves as an abstract of what the file contains. You might then add further information around certain commands to indicate why you were undertaking a certain test, or what part of your writing the results pertain to, such as: '* Results for Table 3.1'.

When working with longer syntax documents or blocks of several commands that you wish to run together, you might wish to add break points or bookmarks. Break points are helpful to establish a hard stop if you want SPSS to run several commands before stopping. Bookmarks can serve as break points but also for speedy navigation between sections. These are not essential and, with shorter syntax files, you can achieve the same effect by highlighting the section of commands you wish to run, then clicking Run > Selection.

If you don't start out with syntax but later want to start using it, you can extract the syntax from the Log sections in the Output window. You can find many useful guides to extend your knowledge of SPSS syntax. SPSS has its own Command Syntax Reference document, which is freely available from IBM's SPSS support website as well as within the program (Help > Command Syntax Reference). A quick web search for SPSS syntax cheat sheets will also turn up a wealth of user-created resources and video tutorials.

After the previous discussion, it may seem like the menus are easier to use than syntax. So why would anyone want to use syntax? There are two main reasons: it will benefit you as a researcher, and it will benefit the discipline.

Benefits for the researcher (you)

The greatest benefit of using SPSS syntax is pragmatic: it is far easier to 'retrace your steps' with syntax if you need to recall the modifications you have made to your dataset, or if you have to download your dataset again because the file corrupts, or you have introduced errors. For example, if you are using a secondary dataset, or have used an online survey method that allows you to export your 'raw' survey data in SPSS format, using syntax to modify variable labels or compute new variables means that you do not have to worry if you lose the SPSS data file you have been working on; you can just re-download the data and rerun your syntax to get your data back. It is also far easier to find and correct mistakes in your analysis if you have the syntax, as opposed to if you had used the menus.

If you are looking at longer term research, syntax can easily be modified and re-used on future projects, saving you time and effort in the long run. This means that you are unlikely to have to start from scratch the next time you carry out a research project. Syntax also remains relatively unchanged between different versions of SPSS (though there are a few notable exceptions), while the menus and dialogue windows have changed considerably, so syntax provides a longer term, more transferrable record of your work. If you are working in a team, it is much easier to communicate to others what actions you performed in SPSS by showing someone your syntax than it is to describe how you used the menus. In general, if you are working on a major project (like a thesis, dissertation or research for publication), or if you are collaborating with others on data analysis, we strongly recommend using SPSS syntax. An increasing number of teachers also require the submission of syntax and/or output files with data reports.

Benefits for the discipline

A number of concerns have been raised regarding the practice of quantitative research across a wide range of academic disciplines, including natural sciences (Fanelli, 2009), psychology (Stroebe et al., 2012), economics (Herndon et al., 2014) and health research (Ioannidis, 2005). In a famous article, Ioannidis (2005) suggested that most empirical findings within health research might well be false. Politics and international relations research in the UK has perhaps been less affected as it contains fewer quantitative studies, and many of the quantitative studies that exist are less likely to affect policymaking or personal safety. However, as an academic subject, it faces the same kinds of incentive structures and practices that may have negatively affected these other disciplines. The concerns about the quality of research publications centre on inadvertent errors and academic misconduct (including fabrication and falsification).

In the processes of creating new variables and conducting analysis, it is all too easy for mistakes to occur by accident. The history of computing contains a number of famous examples. In 1999 an expensive (USD 125 million) satellite designed to orbit Mars burned up in that planet's atmosphere because some of the software was using imperial measurements rather than the metric system, with the latter having been set out as a requirement throughout (Grossman, 2010). Software bugs have also been linked to unexpected acceleration in Toyota vehicles, which has been associated with a significant number of deaths (Dunn, 2013).

One might assume that errors in programming socio-economic research problems would have less drastic consequences, and mostly that is true. However, sometimes errors even in social science research can have widespread impact. In 2010, Reinhart and Rogoff 'showed' that the rate of economic growth in a country would slow when public debt exceeded 90 per cent of GDP. This was used as justification for implementing steep austerity measures in several countries on the grounds that austerity was necessary to avoid further economic damage. Their work was quoted by a range of politicians, including US House Budget Committee Chairman Paul Ryan, EU Commissioner Olli Rehn and UK Chancellor of the Exchequer George Osborne (Coy, 2013; Pollin, 2014). There are a number of important substantive and empirical critiques of this paper, but it is also clear that the spreadsheet used to calculate the main results was flawed and led to the exclusion of a number of countries at the top of the spreadsheet. This omission changed the median annual GDP growth for high-debt countries from –0.1 per cent to 2.2 per cent. The errors were discovered by a doctoral student running a replication exercise for a module he was taking (Herndon et al., 2014).

Another example from the social sciences concerns the effect of marital breakdown on divorce. Weitzman (1985) looked at a group of married couples who divorced, and measured the effects on their material resources. She wrote that women's living standards decreased by 73 per cent in the year following divorce, while men's living standards rose by some 43 per cent. This helped to set the tone for policies that would seek to better align their living standards. However, in a re-analysis, Peterson (1996) found a large number of errors and discrepancies. When these were corrected, the actual changes in income were around a 27 per cent reduction for women and a 10 per cent increase for men. Of course, these are still rather large changes and still show clear differentiated impact along gender lines, but they do cast a rather different perspective on the figures – and indeed are more in line with other studies of the same phenomenon.

Dewald et al. (1986) attempted to replicate the results of economics articles published in the early 1980s. They started by contacting the authors of those articles and requesting the data and programs needed to demonstrate the findings of their published work. Broadly speaking, one-third of authors did not respond, and another third responded to decline the request – sometimes because data had been lost or programs not preserved, or more rarely that the relevant data was confidential. Even where authors did respond in a helpful fashion, the researchers were often unable to precisely replicate the findings. To be fair, this was at a time when data was less often shared than today, and where the production of statistical results was a more complex undertaking, only really in the process of moving from remote access (mainframe) environments to desktop-based personal computers with more user-friendly features. The authors of this replication study concluded that, 'It is widely recognized that errors occur in empirical economic research and appear in published empirical articles. Our results…suggest that such errors may be quite common' (Dewald et al., 1986: 600).

The role that syntax plays in this is by simplifying the process of replication. The 'gold standard' for academic research is to share both the data (unless protected for reasons of safeguarding, anonymity or confidentiality of the participants) and the syntax used to analyse the data (Janz, 2016). This greater transparency allows other researchers – often students whose teachers have set them a replication exercise – to test out the analysis and identify potential errors (King, 1995). Replication exercises cannot, of course, reliably identify when a researcher has fabricated the data itself, though this is sometimes distinguishable by the lack of consistency with findings by other researchers. We would strongly recommend not only ensuring that your own work conforms to this gold standard but also that early-career researchers take the opportunity to try out replication using available data. You can get started with a repository like the Harvard Dataverse (for more about this, see Chapter 2).

LEVELS OF MEASUREMENT

Levels of measurement are a way of classifying the type of data we are looking at, which can help us to identify statistical tests that will be appropriate. If we do not think about the nature of the data we are analysing, we might summarize it in ways that do not help us to understand it and might even lead us to inaccurate conclusions. For example, it would not be meaningful to talk about having a mean sex of 1.2 or a median colour of car that is red. However, if you ask SPSS to produce these statistics, it will. We need to understand the nature of our data in order to make good choices about how to analyse it, and one of the most common ways of doing this is to classify variables by their level of measurement.

The most common system was devised by Stanley Smith Stevens, who noted that 'the statistical manipulations that can legitimately be applied to empirical data depend upon the type of scale against which the data are ordered' (Stevens, 1946: 677). The different scales against which data could be measured were described as being nominal, ordered, interval and ratio. Nominal data can be broken into discrete categories, but there is no inherent order to the categories. Sex, marital status and political party affiliation are all examples of nominal variables. Ordinal data is still measured in categories, but the categories can be placed in a logical order. Examples of ordinal data include age groups (18–24, 25–34, 35–44, etc.); spectrums of agreement, importance, etc. (strongly agree, agree, disagree, disagree strongly); and highest educational attainment (no qualifications, some secondary school, completed secondary school, some tertiary, tertiary degree, etc.). Interval data and ratio data are data that can be measured numerically. With interval data, the distance between the intervals is the same, but zero does not constitute the starting point. With ratio data, the distance between intervals is the same, but the scale starts with an absolute zero. For example, if we measure someone's height in centimetres, the distance between intervals is always 1 cm, and height starts at an absolute zero, making it ratio data. Other examples include age, income and years of education completed. There are examples of each type listed in Table 3.2.

Table 3.2 Examples of levels of measurement

Scale of measurement	SPSS name	Other names for concepts		Examples
Nominal	Nominal	Dichotomous / binary (if exactly two values); qualitative	Categorical	Political affiliation, religion, marital status, ethnic group
Ordinal	Ordinal	Ordered		Position (1st, 2nd, 3rd, etc.) in some kind of competition; preferences for one party over another; attitude scales (such as 'strongly agree, agree, disagree, strongly disagree')
Interval	Scale	Continuous; quantitative	Discrete	IQ, calendar years
Ratio			Continuous	Income, height, weight

There are other typologies that have arisen since Stevens'. In SPSS, there are three types used: nominal, ordinal and scale. In this typology, scale encompasses Stevens' interval and ratio. Other typologies use a dichotomy between categorical data (where the differences between categories cannot be meaningfully measured) and continuous (where it can). These classification systems form *typologies*, where each variable may be classified into only one of these categories, and the categories are mutually exclusive and exhaustive. Having established such a typology, in the past it was standard to use it to identify particular statistical approaches that were appropriate, both for simpler and for more advanced statistical concepts (Andrews et al., 1981). In Table 3.3 we set out a guide to the kind of simple statistics that seem to make most sense for data of different kinds. We will talk through more of the nuances of selecting the right kinds of data for various tests as we encounter each test in the later chapters.

Table 3.3 Examples of statistical tests by level of measurement

	Nominal	Ordinal	Continuous
Frequency	Yes	Yes (unless large number of categories)	No (or only if small-*N*)
Measures of central tendency	Mode	Mode, median	Mode, median, mean
Correlation and regression (linear)	No – but other statistics possible	No – but other statistics possible	Yes

In secondary datasets, the level of measurement will normally be set already. However, this has not always been done accurately, and there are also instances when you wish to manually change a level, such as changing how the data is treated to produce a chart. In Box 3.1 we show how to do this, first by using syntax, then using the graphical user interface. Because we want to encourage you to get comfortable with syntax, we will always provide the syntax instructions first, with a short explanation of the different components of the command, then the menu instructions.

BOX 3.1 SETTING OR CHANGING THE MEASUREMENT LEVEL

This box shows you how to set or change the level of measurement of a variable. This is appropriate for classifying variables after entering data and for reclassifying variables that have been incorrectly entered or for which you need to change the classification for analysis, such as producing charts that are dependent upon using specific types of variables.

Syntax

It is possible to set/change the measurement level for a list of variables using syntax. You simply need the VARIABLE LEVEL command, followed by a list of variables you wish to have the same level of measurement, then the level of measurement in parentheses. For example:

```
VARIABLE LEVEL gndr cntry (nominal).
```

You can change multiple levels of measurement in the same command by separating the levels with a slash (/). For example:

```
VARIABLE LEVEL

gndr (nominal)
```

(Continued)

```
/trstprl trstlgl (ordinal)

/height weight agea (scale).
```

Menu instructions

To change the measure, open the Data Editor window, Variable View tab. Within the Data Editor window, simply click the cell of the measure you want to change (such as clicking on Nominal), and select the measure from the drop-down menu. Taking this approach, you can only change one variable at a time.

A critique of levels of measurement

> My propositions are elucidatory in this way: he who understands me finally recognizes them as senseless, when he has climbed out through them, on them, over them. (He must so to speak throw away the ladder, after he has climbed up on it). He must surmount these propositions; then he sees the world rightly. (Wittgenstein, 1922: 6.54)

The approach of classifying variables in a hierarchy from ratio to nominal has been influential in the writing of statistical textbooks and probably to an even greater extent in the delivery of statistics teaching to students in the social sciences. This is not an issue without controversy, however, and there are several issues that come with this typology. Having established this classification system, it then can become a simple tick-box matter to rule that certain approaches are statistically legitimate, while others are not. Indeed, it may be used as a basis for a kind of expert system that diagnoses that kinds of tests and statistical approaches are relevant in different circumstances (Andrews et al., 1981). However, it is worth a caution that this typology has been strongly attacked by statisticians, and in particular the idea that such absolutes may be applied to data independently of the research questions that are being addressed. It is quite common to present averages of rankings, for instance (such as the changing average position of a country within the PISA league tables of educational performance; or of a sports team within a particular division). Some types of data are also hard to place on the scale. For instance, percentages or fractional amounts may look like standard ratio variables, but they are not because the data cannot go below 0 or above 100. This constraint can have important implications for different kinds of statistical methods (such as linear regression) that we discuss later. At a practical level, the apparently illegitimate use of data that is ordinal (such as attitude scales), treating it as being on an interval scale, is fairly routine in many disciplines and may not generate misleading information. Indeed Stevens (1951) recognized this issue in his later writings and saw pragmatic reasons for the apparent mistreatment of ordinal data as being of a more informative interval kind. There are also alternative taxonomies of variable types (Mosteller and Tukey, 1977: chapter 5).

This typology can also draw artificial distinctions between categories. For example, it is very straightforward to convert continuous data (such as age) into categorical data (such as age groups). Even ordinal categorical data is often combined in ways that create a continuous variable. For example, in the ESS, there is a bank of questions about views on immigration, such as Immigration bad or good for country's economy [imbgeco], Country's cultural life undermined or enriched by immigrants [imueclt] and Immigrants make country worse or better place to live [imwbcnt]. In such cases, researchers may combine the answers to a variety of questions to create an 'index' variable. Index variables can be useful analytical tools for creating greater variation between respondents by identifying respondents who consistently feel very strongly about a topic. These transformations can be achieved by recoding (changing the categories of the data, covered in Chapter 4) and computing (combining multiple variables or mathematically manipulating existing variables, covered in Chapters 7, 9 and 10).

For most purposes the attribution of measurement levels will not change the way that SPSS functions, but with three important exceptions – Chart Builder, custom Tables and the Tree approach to investigating data structures. For most users, it is when using Chart Builder (discussed in Chapters 5 and 8) that the setting of measurement levels will be important. In its default settings, SPSS will remind the user of the importance of setting appropriate measurement levels when using such commands, such as when starting Chart Builder. There is also a reminder to set value labels for categorical variables and for each category, as this information is used in labelling graphs.

CONCLUSIONS

This chapter has introduced you to the main windows of SPSS: the Data Editor, Output and Syntax windows. Each of these plays a different role in the function of the program and saves as a separate file. We have looked at some of the key elements within each file and have covered some initial adjustments you might want to make to the way variable information is stored. You have learned how to input your own data and how to keep a record of your work. From the next chapter, you will start producing and analysing your own statistics using the two key datasets for this book: the ESS (round 7) and the UN composite dataset.

ACTIVITIES

1. Open a new file in Data Editor. Create five new variables in Variable View using the information in Table 3.4. Then enter the data from Table 3.5 in the Data View tab, using a new row for each respondent. Check that you have entered the value labels correctly by toggling the Value Labels display in the Data View tab.

(Continued)

Table 3.4 Variable information

Name	Type	Label	Values	Missing	Measure
idno	Numeric	Respondent's ID number			
yrbrn	Numeric	Year of birth	–7 Refused –8 Don't know –9 No answer	Range Low: –99 High: –1	Scale
vote	Numeric	Voted last national election	1 Yes 2 No –7 Refused –8 Don't know –9 No answer	Range Low: –99 High: –1	Nominal
happy	Numeric	How happy are you	0 Extremely unhappy 10 Extremely happy –7 Refused –8 Don't know –9 No answer	Range Low: –99 High: –1	Ordinal
cntry	String	Country	AT Austria BE Belgium CH Switzerland CZ Czech Republic		Nominal

Table 3.5 Respondent information

idno	yrbrn	vote	happy	cntry
1001	1965	1	10	CH
1002	1972	1	–9	BE
1003	1989	2	3	AT
1004	–7	2	6	CZ

2. Open the ESS dataset and create a new data file. Choose three variables from the ESS and practise inputting the same information into the new data file to create three new variables. Then create responses for four respondents. Practise toggling the view in Data View so that you can see the codes or the labels.

FURTHER READING

Cunningham, J.B. and Aldrich, J.O. (2016) *Using IBM SPSS Statistics: An Interactive Hands-On Approach*. London: Sage.

See chapters 1 and 2 for getting to know the SPSS environment and chapter 3 for information on importing, inputting and exporting data. Gives step-by-step point-and-click instructions with screenshots but takes an entirely menu-driven approach; in other words, no discussion of syntax.

IBM SPSS Statistics Coach (https://www.ibm.com/support/knowledgecenter/en/SSLVMB_24.0.0/spss/statcoach/statcoach_main.html).
You can change version easily to the version of SPSS you are using. This will walk you through a series of questions to help you determine which function to carry out. Because it is based on a series of questions where you have to choose between options, this is useful when there is a clear-cut answer but will not highlight situations when there is more than one way that you might go about your analysis.

IBM SPSS Statistics Command Syntax Reference (https://www.ibm.com/support/knowledgecenter/en/SSLVMB_24.0.0/statistics_reference_project_ddita-gentopic2.html).
This is also accessible through Help > Command Syntax Reference within the software windows. Provides information about every syntax command, including the rules that must be followed for that command. Very useful as a reference guide but not straightforward for getting started.

Pallant, J. (2016) *SPSS Survival Manual*. London: McGraw-Hill Education.
See chapters 1 to 5 for content related to getting started and inputting or importing data. Provides more information than most texts about preparing and entering your own data into SPSS. Good for helping you to produce statistics but weaker on understanding what your outputs mean. Discusses syntax and provides the syntax for many of the tasks, though usually in Paste format rather than discussing shortened syntax or unnecessary commands.

Various (2016) 'Replication forum', *International Studies Perspectives*, 17 (4): 361–475. (https://academic.oup.com/isp/issue/17/4).
Discusses the importance of replication.

REFERENCES

Andrews, F.M., Klem, L., Davidson, T.N., O'Malley, P.M. and Rodgers, W.L. (1981) *A Guide for Selecting Statistical Techniques for Analysing Social Science Data*. Ann Arbor: Institute for Social Research, University of Michigan.

Coy, P. (2013, 18 Apr) 'FAQ: Reinhart, Rogoff, and the Excel Error That Changed History', *Bloomberg Businessweek*. (https://www.bloomberg.com/news/articles/2013-04-18/faq-reinhart-rogoff-and-the-excel-error-that-changed-history).

Dewald, W.G., Thursby, J.G. and Anderson, R.G. (1986) 'Replication in empirical economics: the journal of money, credit and banking project', *American Economic Review*, 73 (4): 587–603.

Dunn, M. (2013, 28 Oct) 'Toyota's killer firmware: bad design and its consequences', *EDN Network*. (www.edn.com/design/automotive/4423428/Toyota-s-killer-firmware--Bad-design-and-its-consequences).

Fanelli, D. (2009) ‘How many scientists fabricate and falsify research? A systematic review and meta-analysis of survey data’, *PLOS ONE*, 4 (5): e5738. (https://doi.org/10.1371/journal.pone.0005738).

Fausto-Sterling, A. (2000) *Sexing the Body: Gender Politics and the Construction of Sexuality*. New York: Basic Books.

Grossman, L. (2010, 10 Oct) ‘Nov. 10, 1999: metric math mistake muffed Mars meteorology mission’, *Wired*. (www.wired.com/2010/11/1110mars-climate-observer-report/).

Herndon, T., Ash, M. and Pollin, R. (2014) ‘Does high public debt consistently stifle economic growth? A critique of Reinhart and Rogoff’, *Cambridge Journal of Economics*, 38 (2), 257–79. (https://doi.org/10.1093/cje/bet075).

Ioannidis, J.P.A. (2005) ‘Why most published research findings are false’, *PLOS Medicine*, 2 (8): e124. (https://doi.org/10.1371/journal.pmed.0020124).

Janz, N. (2016) ‘Bringing the gold standard into the classroom: replication in university teaching’, I*nternational Studies Perspectives*, 17 (4): 392–407. (https://doi.org/10.1111/insp.12104).

King, G. (1995) ‘Replication, replication’, *PS: Political Science & Politics*, 28 (3): 444–52. (https://doi.org/10.2307/420301).

Mosteller, F. and Tukey, J.W. (1977) *Data Analysis and Regression: A Second Course in Statistics*. Boston, MA: Addison-Wesley.

Peterson, R.R. (1996) ‘A re-evaluation of the economic consequences of divorce’, *American Sociological Review*, 61 (3): 528–36. (https://doi.org/10.2307/2096363).

Pollin, R. (2014, 3 Jan) ‘Public debt, GDP growth, and austerity: why Reinhart and Rogoff are wrong’, *OUPblog*. (https://blog.oup.com/2014/01/public-debt-gdp-growth-austerity-why-reinhart-and-rogoff-are-wrong/).

Reinhart, C.M. and Rogoff, K.S. (2010) ‘Growth in a time of debt’, *American Economic Review*, 100 (2): 573–78. (https://doi.org/10.1257/aer.100.2.573).

Stevens, S.S. (1946) ‘On the theory of levels of measurement’, *Science*, 103 (2684): 677–80.

Stevens, S.S. (1951) ‘Mathematics, measurement, and psychophysics’, in S.S. Stevens (ed.), *Handbook of Experimental Psychology*. New York: John Wiley. pp. 1–49.

Stroebe, W., Postmes, T. and Spears, R. (2012) ‘Scientific misconduct and the myth of self-correction in science’, *Perspectives on Psychological Science*, 7 (6): 670–88. (https://doi.org/10.1177%2F1745691612460687).

Weitzman, L.J. (1985) *The Divorce Revolution: The Unexpected Social and Economic Consequences for Women and Children in America*. New York: The Free Press.

Wittgenstein, L. (1922) *Tractatus Logico-Philosophicus*. London: Routledge.