Getting Started with SPSS Syntax

Course Notes

Edition 4
September 2015
Preface

These notes are based on courses which used to run at the University of Edinburgh.

http://www.ed.ac.uk/is/statistical-computing

IBM Information

IBM own the copyright for IBM SPSS Statistics and other related SPSS applications.

SPSS 18 and 19 manuals: http://public.dhe.ibm.com/software/analytics/spss/support/

SPSS 21 onwards knowledge center: http://www-01.ibm.com/support/knowledgecenter/#!/SSLVMB/welcome

A PDF version of the syntax reference guide can be downloaded by choosing Command Syntax Reference from the Help menu when SPSS is open.

The IBM web site on the IBM public site at: http://public.dhe.ibm.com/software/analytics/spss/documentation/ contains pdf documentation for the most recent and penultimate SPSS modules in a directory structure including DataCollection, amos, modeller and statistics modules.

IBM DeveloperWorks

Resources for programming can be downloaded from the IBM DeveloperWorks forum: https://www.ibm.com/developerworks Books and articles can be found on the forum as well including the very useful "Programming and Data Management for IBM SPSS Statistics" by Raynald Levesque.

Conventions

We will use certain conventions in these course conventions.

Menus

Choosing an item from a menu is indicated by main menu > submenu > item. For example, File > New > Data... means open the File Menu from that choose New sub menu and then choose Data... from the sub menu (this particular menu choice will open a new data window).

Actions

This type of paragraph is an action paragraph. That is to say it indicates something you can do in SPSS if you are following the notes on your computer.

Information/warnings

This sort of paragraph is an information paragraph and will contain points of note or warnings.

References

This paragraph points to other references or web sites to visit.

Copyright © IS 2015

Permission is granted to any individual or institution to use, copy or redistribute this document whole or in part, so long as it is not sold for profit and provided that the above copyright notice and this permission notice appear in all copies.

Where any part of this document is included in another document, due acknowledgement is required.
Session A

Using SPSS Commands
Using SPSS commands

SPSS command language, called Syntax lies behind any work you do with SPSS, although most analyses can be done using the menus in the normal course of events you need never encounter any SPSS commands directly. In the distant past, the only way to use SPSS was via its command language. In the present using a default installation, you will only see command names very briefly at the bottom of the window in the status bar while the relevant procedures are being run by SPSS.

However it is easy to see the underlying commands and generate SPSS commands from the interactive menus and dialog boxes. SPSS commands are stored in a Syntax window, which is a plain text file usually with a .sps extension. Keeping command syntax gives you the ability to perform the same tasks again and again.

**SPSS commands in the output viewer**

You can choose to have the underlying commands displayed in a log along with the output in the output viewer using the **Viewer** tab from the **Edit > Options** menu, or **SPSS Statistics > Preferences** (for Mac).

1. Start up SPSS and choose **Type in data**, so the data editor is empty.
2. Choose **Edit > Options** or **SPSS Statistics > Preferences** to open the **Options** dialog box.
3. Click on the **Viewer** tab to look at the Viewer options.

   - Tick **Display commands in the log** box under **Initial Output State** heading to display commands. Click **OK** or **Apply**.

Once this feature has been switched on, the underlying SPSS commands for anything you do in SPSS will be pasted in the output window before the output itself.

**GET command**

For example, you should see a **GET** command followed by the **DATASET** command in the output window if you open an SPSS data file.

1. Open the 1991 U.S. General Social Survey.sav data file in the spsswork directory (using **File > Open > Data**) and look at the output window.

   - The full **GET** command in the windows output is wrapped onto the next line:

   ```spss
   get file = 'm:\spsswork\1991 U.S. General Social Survey.sav'.
   ```

   - And the version from the macintosh output is shown wrapped to the next line:

   ```spss
   ```

   - The **GET** command above need to be modified to run in the syntax window as follows:

   ```spss
   ```

   - The **GET** command reads an SPSS data file into the data editor. Its general form is:

   ```spss
   GET FILE = 'path\filename'.
   ```

**DATASET command**

The other command, you will probably see is the **DATASET** command used to name the active data window. It's not absolutely necessary, but it looks like:

```spss
DATASET NAME Dataset1 WINDOW=FRONT.
```
The FREQUENCIES command is:-

FREQUENCIES VARIABLES= happy life /ORDER = ANALYSIS.
In its simplest form all you need is a list of the variables:-

Frequencies variables= happy, life.

Notes in output viewer
The other place you can see a report of the underlying commands is in the Notes object, produced every time you run something a procedure. In the Viewer outline, to the left of the output as you look at it, you can see a Notes object as part of every procedure's output, although by default its contents are hidden. Above Figure 2 shows the Notes object as closed or hidden. Below in Figure 3 shows the Notes object open to view.

Figure 3: Notes from the Frequencies procedure

Notes are quite useful to check details if an SPSS procedure hasn't worked as you expected it to. From the notes you can check to make sure you haven't any weights, split files or selection filters switched on. You can check that you have the correct file and syntax and there is a date and time stamp, so you can check when the output was produced.

The syntax window
The syntax window is used to store and run SPSS commands. The default installation1 of SPSS will not open a syntax window. Each time you do something with the menus whether it is to read in data or perform an analysis, you have issued a command or series of commands for SPSS to obey. It is possible to paste the SPSS commands underlying any dialog box into a syntax window, using the Paste button - available on most SPSS dialog boxes. It will automatically open up a new syntax2 window and paste the underlying command(s) from the dialog box into the new window. The contents of a syntax window can be saved to a file, so the analysis can be rerun later without using menus.

Figure 4: Frequencies dialog box again

For example if we go back to the Frequencies procedure run earlier we can paste the underlying FREQUENCIES command into a syntax window.

Choose Analyze > Descriptive Statistics > Frequencies
Click the Charts... button and select a bar chart.
Click Continue to get back to the Frequencies dialog box.
Instead of pressing OK to run Frequencies, click on Paste.

Figure 5: Frequencies command in the syntax window

SPSS will automatically open up a syntax window and bring it to the front of the screen with the FREQUENCIES command pasted into it.

1 You can change this default using the Edit > Options... menu, under the General tab, by ticking Open syntax window at start-up, mentioned on page 2.
2 If a syntax window is already open it may put the command in the syntax window without bringing it to the front.
This is the `FREQUENCIES` command underlying the analysis we ran at the beginning of the first exercise. You can see the `FREQUENCIES` command keyword, three subcommand keywords `VARIABLES`, `BARCHART` and `ORDER` separated by `/` and two other keywords `FREQ` and `ANALYSIS`. And, like all commands, it's finished with a full stop.

To break the command down:

- The `VARIABLES` subcommand lists the variables for which we want frequencies, in this example, the variables `sex race region happy` and `life`.
- The `BARCHART` subcommand requests a bar chart to be plotted for each of the variables in the list. `FREQ` specifies frequencies, raw counts, rather than percents on the scale axis.
- The `ORDER` subcommand with keyword `ANALYSIS` specifies all the charts after the tables, this is the default so you could miss the subcommand out entirely. If the `VARIABLE` keyword had been used instead, the chart for a particular variable would be placed next to its frequency table.

If you want to see all the possible subcommands and other keywords you can use with any command, click the `Show syntax chart` button on the syntax window toolbar to look at its syntax chart.

As you can see the chart is quite complicated but will give you all the elements you can put in a `FREQUENCIES` command. You maybe able to spot the subcommands and keywords just described.

Each element in square brackets `[ ]` indicates an optional element, a list of elements aligned above one another in curly brackets `{ }` indicates a choice of one out of that series of elements, default choice is usually given in bold. All SPSS command, sub command and other keywords are presented in uppercase in chart (although you don’t need to type them in uppercase). Check the syntax reference guide or on-line help for examples of how each command is used.

Once you are finished having a look at the help information, close the Help window and go back to the Syntax window (choose it from the Windows menu in SPSS if you can't otherwise).

The SPSS Command Syntax Reference guide will give you more details about SPSS commands and how to run them. Syntax guides are available online at IBM or within SPSS in PDF format from Help > Command Syntax Reference.
Running an SPSS command

The Run Selection button on the syntax window tool bar looks like an audio play button. It’s used to run the command currently selected in the syntax window. It is easy to run an already pasted command, either click somewhere on the command or highlight the entire command (including the full stop) and click the run button.

Figure 8: Running the FREQUENCIES command

- Click somewhere on the FREQUENCIES command and click the Run button to execute the command.

You will see the procedure being run as you did when you used the menu to run Frequencies. When the processor has finished - it shouldn’t take very long - the results will be stored in the output window as usual.

Typing in a command

The Syntax window can be used on its own to run SPSS commands, if you know what the commands are (or have generated them from the menus). A new syntax window can be opened by choosing File > New > Syntax or an existing one can be opened using File > Open > Syntax. A syntax window can be brought to the front by choosing it from the Windows menu if it is already open.

Figure 9: CROSSTABS command selected

- Select the Syntax window from the Window menu, Window > 3 Syntax1 – SPSS Syntax Editor.

The first syntax window is usually called Syntax1 and the 3 before the name indicates it’s the third open SPSS window, after the data editor and output window. If you have more windows open then of course the number might be different.

Figure 10: Starting to type a command into the Syntax Editor window

Each command starts in the editor on a new line with a command keyword and ends with a full stop. Between that the command may consist of several sub command keywords, other keywords, variable names and text, depending on the command. From version 16 SPSS provides help by displaying a list of command keywords as soon as you start typing. Choose the command keyword you want from the list and either double click or press return to paste it into the syntax window.
The **CROSSTABS** command

The following **CROSSTABS** command will crosstabulate the *happy* variable by the *life* variable from the social survey dataset.

```
crosstabs happy by life.
```

It is a simple form of the **CROSSTABS** command with no subcommands. As you type the command the **CROSSTABS** keyword is in red to indicate it's not finished. SPSS sees it as finished once you ended it with a full stop.

Type in the command "crosstabs happy by life." into the Syntax window.

**Figure 11:** CROSSTABS command before and after the full stop is added

---

A full stop is needed to indicate the end of each SPSS command. If you forget the full stop the command may not run properly.

Select the whole CROSSTABS command as shown above and click on the Run button to run it.

If you switched on Display commands in the log demonstrated on page 2, then the commands are displayed with the output then you will see the CROSSTABS command just before the output from Crosstabs.

**Figure 12:** Start of Output from the CROSSTABS command

---

The **CROSSTABS** command has several subcommands, try the syntax chart button to see what they are. They include:

- **CELLS** used to change the content of the table
- **BARCHART** to add a clustered bar chart for each layer.

We will add both of these subcommands to the Crosstabs using the pop-up menus to build the commands. Subcommands separated by / character usually, typing a / just before the full stop of the command will generate a list of the subcommands.

**Figure 13:** CROSSTABS subcommands list

---

Type in a / before the full stop in the CROSSTABS command to generate a list of subcommand keywords shown in Figure 13. Select CELLS and press return to paste the subcommand.

Typing = after the subcommand will generate a list of the keywords associated with the CELLS subcommand. Figure 14 shows the COUNT and COLUMN keywords in the process of being selected.

**Figure 14:** CROSSTABS CELLS subcommand keywords

---

3 **CROSSTABS** has two more subcommand keywords in newer versions of SPSS: HIDE SMALL COUNTS and SHOW DIM.

4 In the CELLS keyword list, you should see BRPROP keyword as well.
Session A: Using SPSS Commands

Typing in another / will start a new subcommand and bring the subcommand keyword menu again. This time we will choose **BARCHART**.

### Figure 15: CROSSTABS adding Barchart subcommand

- Type / before the full stop to get a list of CROSSTABS subcommand keywords, then select **BARCHART**.
- Run the command and compare the difference in the output between the previous output.

#### Layers

Layers can be added to the table by adding another **BY** keyword followed by the layer variable or variable list to the table specification. For example,

```spss
CROSSTABS happy by life by region.
```

Will produce a crosstabs table of happy by life layer for each region. Try adding a layer to the latest crosstabs command and run the command to check the effect.

- **Some command names, such as FINISH, BEGIN DATA and END DATA are complete commands which need no further specification. However most commands (including all of the statistical procedures) have to have a certain amount of specification and each has its own particular form of specification, so check the help or syntax guide before you run.**

### Slide Content from presentation

The rest of this section consists of key elements from the power point slides used during the course

#### SPSS Command Language

Starting with commands

Why use Commands?

- More flexible
- Extra features
- Can be saved
- And repeated

### The Syntax Window

SPSS commands are entered in the Syntax Window with a .sps extension.

Create a Syntax Window

- Either selecting **File > New > Syntax** from the menu bar.
- Or creating file with .sps extension in a text editor.

Use an existing syntax window

- Either Selecting an open syntax window from the Window menu.
- Or selecting **File > Open > Syntax** from the menu bar.

#### SPSS commands

Commands are entered in the Syntax Window by:

- Either opening the syntax Window and typing a command in directly
- Or building up a command using dialog boxes then clicking the **PASTE** button
- Or using a mixture of the two
Format of Commands

Control Field - consists of SPSS keyword

Specification Field - names of variables etc. on which command will operate

Full stop - to terminate command

Syntax Rules

Commands start with a command keyword

Commands are case insensitive *(Except for text in quotes)*

A single command can go over many syntax lines.

Each line of syntax must not exceed 256 bytes\(^5\).

Do not break a syntax line

- in a variable name
- in between quotes ‘ ’ or “ “.

For a syntax window

- Each command must start on a new line.
- Each command must end with a full stop with no non-blanks after it or end with a completely blank line afterwards.

For batch commands (see last session)

- Each command begins on a new line in column 1
  - unless you use + or -

  Continuation lines mustn’t start in column 1

  Full stops, command terminators, are optional

The FINISH command

FINISH.

Used for batch commands only.

Terminates a file of SPSS commands and closes SPSS tidily.

N.B. When SPSS windows manager is running (e.g. during an SPSS for Windows session), FINISH will generate an error.

Terminate an interactive SPSS session using File > Exit menu.

\(^5\) 256 bytes are characters in single byte languages... Longer lines will be truncated to 256.
Session summary

In this session we covered:

- How to print in output window, the SPSS command procedures which underly the interactive menu/dialog box choices.
- The Syntax Window is where SPSS commands (also called syntax) are stored.
- The Paste button in dialog boxes puts the command into a syntax window.
- Using the syntax chart button for help on command in the syntax window.
- How to type in and run an SPSS command from the syntax window.
Session B

SPSS Command Structure
Commands, Subcommands & Keywords

In general an SPSS command will be a mixture of different elements like this:

```
COMMAND SUBCOMMAND varnames /SUBCOMMAND KEYWORDS
/SUBCOMMAND KEYWORDS /SUBCOMMAND KEYWORDS etc..
```

An SPSS command starts with a recognized command keyword which has to be spelt correctly. Each command has its own general format with its own set of subcommands and keywords. These are combined with lists of variables names or combinations of variable names and keywords to specify the analysis or action required by the computer.

Subcommands and other keywords are words recognized by SPSS as part of the command's specification, and so appear in the specification field. Some SPSS procedures have subcommands which can appear either on their own or in association with certain keywords. As with commands, subcommands and keywords have to appear exactly as they are spelt.

**Abbreviating keywords**

SPSS will recognise most of its command and subcommand keywords when abbreviated to the first three or four characters of the name as long as the keywords are unique. So for example:

```
freq vars=height cigaret /form=lim(12) /hist.
```

can be used as an abbreviation of:

```
frequencies variables=height cigaret
    /format=limit(12) /histogram.
```

**Reserved keywords**

With most procedures it does not matter if there is a variable name the same as a keyword, since there are certain places in a command specification where a variable or variable list is expected and others where keywords are expected. However there are some reserved keywords which cannot be used as names. These are:

```
ALL and by eq ge gt le lt ne not or to with
```

These words cannot be used as variable names in SPSS because the program already recognizes them as having meaning. For example,

```
FREQUENCIES VARIABLES=ALL.
```

calculates a frequency table for all the variables currently stored on SPSS’s active file. To have a variable called **ALL** would make the sense of the instruction ambiguous to SPSS.

The different types of keyword have been highlighted in the same colour combination used in the syntax window, as follows:

```
COMMAND keywords, SUBCOMMAND keywords,
OTHER keywords, VALUES, variable names & other specification
```
Lists of Variables

The variables are kept in the SPSS active file in the order they are given in the DATA LIST command and any variables generated during the session in the order they were created. File > Display Data File Information > Working File or External File will display a dataset variable details and positions. This order can be used so instead of typing a long list of consecutive variables, you can use just the first and last variable to create the list. These inclusive ranges of variables can be combined with the names of other variables to produce a larger variable list where needed. This table is a list of the survey dataset variables and their positions:

<table>
<thead>
<tr>
<th>Position</th>
<th>Name</th>
<th>Position</th>
<th>Name</th>
<th>Position</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sex</td>
<td>15</td>
<td>tax</td>
<td>31</td>
<td>work1</td>
</tr>
<tr>
<td>2</td>
<td>race</td>
<td>16</td>
<td>usint1</td>
<td>32</td>
<td>work2</td>
</tr>
<tr>
<td>3</td>
<td>region</td>
<td>17</td>
<td>obey</td>
<td>33</td>
<td>work3</td>
</tr>
<tr>
<td>4</td>
<td>happy</td>
<td>18</td>
<td>popular</td>
<td>34</td>
<td>work4</td>
</tr>
<tr>
<td>5</td>
<td>life</td>
<td>19</td>
<td>thnkself</td>
<td>35</td>
<td>work5</td>
</tr>
<tr>
<td>6</td>
<td>sibs</td>
<td>20</td>
<td>workhard</td>
<td>36</td>
<td>work6</td>
</tr>
<tr>
<td>7</td>
<td>childs</td>
<td>21</td>
<td>helpoth</td>
<td>37</td>
<td>work7</td>
</tr>
<tr>
<td>8</td>
<td>age</td>
<td>22</td>
<td>hlth1</td>
<td>38</td>
<td>work8</td>
</tr>
<tr>
<td>9</td>
<td>educ</td>
<td>23</td>
<td>hlth2</td>
<td>39</td>
<td>work9</td>
</tr>
<tr>
<td>10</td>
<td>paeduc</td>
<td>24</td>
<td>hlth3</td>
<td>40</td>
<td>prob1</td>
</tr>
<tr>
<td>11</td>
<td>maeduc</td>
<td>25</td>
<td>hlth4</td>
<td>41</td>
<td>prob2</td>
</tr>
<tr>
<td>12</td>
<td>speduc</td>
<td>26</td>
<td>hlth5</td>
<td>42</td>
<td>prob3</td>
</tr>
<tr>
<td>13</td>
<td>prestg80</td>
<td>27</td>
<td>hlth6</td>
<td>43</td>
<td>prob4</td>
</tr>
<tr>
<td>14</td>
<td>occcat80</td>
<td>28</td>
<td>hlth7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>hlth8</td>
<td>29</td>
<td>hlth8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>hlth9</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The TO keyword is used between the first and last variable name in the list, first to create a variable list, for example using the survey data

`hlth1 TO hlth9`

implies variables: `hlth1, hlth2, hlth3, hlth4, hlth5, hlth6, hlth7, hlth8, hlth9`

In SPSS variable lists, individual variable names and/or variable ranges can be separated by either spaces or commas. However, any keywords have to be separated from variable names by spaces, i.e. use spaces with TO or ALL.

Happy, life, tax TO helpoth
implies variables `happy, life, tax, usint1, obey, popular, thnkself, workhard, helpoth`

sex TO prob4
implies all the variables as `sex` is the 1st variable on the list and `prob4` is the last.

The reserved keyword ALL in this context means all the variables in the active file. It can be used in a variable list for the KEEP subcommand on a GET or SAVE command to change the order the variables are saved in whilst making sure no variables are missed. Variables not specifically listed will appear after all the listed ones in active file order.

happy life educ sex, race, region all
implies the order: `sex, race, region, happy, life, sibs, childs, age, educ, paeduc, maeduc, speduc, prestg80, occcat80, tax, usint1, obey, popular, thnkself, workhard, helpoth, hlth1, hlth2, hlth3, hlth4, hlth5, hlth6, hlth7, hlth8, hlth9, work1, work2, work3, work4, work5, work6, work7, work8, work9, prob1, prob2, prob3, prob4`
Some SPSS Command examples (from slides)

The next few pages are examples of SPSS commands taken from the Powerpoint slides used during the course. Some of the examples will illustrate how you can configure the commands with sub commands and other keywords.

The GET command

The GET command reads a .sav file into SPSS.

```
GET FILE = "cardiac.sav".
```

The GET command - /KEEP and /DROP subcommands

Use the /KEEP subcommand to select only some of the variables

```
GET FILE="cardiac.sav"/KEEP=famhist, chd.
```

Use the /DROP subcommand to select all of the file except certain variables

```
GET FILE="cardiac.sav"/DROP=height, weight.
```

The DATASET NAME command

The DATASET NAME command give the active dataset a short name which can be used in later commands.

```
GET FILE = "m:\spsswork\cardiac.sav".
DATASET NAME cardiac.
```

There are other DATASET commands to manage multiple open datasets.

The SAVE command

Use the SAVE command to save your data in a .sav file,

```
SAVE OUTFILE="myfile.sav".
```

The SAVE command will:

save all your cases, together with variable names, values, missing values, variable and value labels and formats

_Saved in the state they are in when the command is issued._

Any further changes can only be saved with another SAVE command.

It will NOT save the results of any analyses, results are saved in an output window.

The `SAVE` command /KEEP and /DROP subcommands

The /KEEP and /DROP subcommands are available with the `SAVE` command, and perform the same function as with GET.

Using KEEP will save only the named variables.

```
SAVE OUTFILE="file.sav" /KEEP=AGE HEIGHT.
```

Using DROP will save all variables except named ones (here it means all variables except WEIGHT2)

```
SAVE OUTFILE="file.sav" /DROP=WEIGHT2.
```

`RENAME` subcommand for GET and SAVE commands

The `RENAME` subcommand is used to change variable names.

```
/RENAME (old names=new names) (oldvar=newvarname)
```

Using `RENAME` with the GET command will read the new variable names into the active file.

```
GET FILE="m:\spsswork\cardiac.sav"
/RENAME (eduyr=edyears)
(cigaret smokes=ncigs ciggroup).
```

Using `RENAME` with the SAVE command will put the new names into the data file - variables in the active file remain untouched.

MAP subcommand

The MAP subcommand is used to prints a list of the variables in the result file and the variable list before the name changes.

```
GET FILE="m:\spsswork\cardiac.sav"
/RENAME (eduyr=edyears)(cigaret smokes=ncigs ciggroup) /MAP.
```

MAP subcommand output

```
FILE MAP
RESULT   INPUT1
-------   ------
EDYEARS   EDUYR
NCIGS     CIGARET
HEIGHT    WEIGHT
WEIGHT    DAYOFWK
FAMHIST   FAMHIST
CHD       CHD
CIGGROUP  SMOKES
EDGRP     EDGRP
```
**FREQUENCIES command**

FREQUENCIES VARIABLES=HEIGHT.

**FREQUENCIES - /STATISTICS subcommand**

FREQ VAR=HEIGHT, AGE /STATISTICS.

Gives four default statistics:

MEAN, STDDEV, MINIMUM, MAXIMUM

**FREQUENCIES - More statistics**

MEDIAN, MODE, SUM, RANGE, SEMEAN, VARIANCE, SKEWNESS, SKEW, KURTOSIS, SEKURT

Use to get the default four statistics MEAN, STDDEV, MINIMUM, MAXIMUM together with others you ask for:

/STATISTICS=DEFAULT

Gives all available statistics - can be wasteful.

/STATISTICS=NONE

Equivalent to omitting /STATISTICS altogether.

**FREQUENCIES - /FORMAT subcommand**

To suppress the frequency table use:

/FORMAT=NOTABLE

To suppress tables over a certain length:

/FORMAT=LIMIT(50)

**FREQUENCIES - subcommands for charts**

There are three subcommands to produce type of charts named:

/HISTOGRAM
/BARCHART
/PIECHART

**Using more than one subcommand**

To use more than one subcommand, string them together and finish the command with the terminating full stop.

FREQUENCIES VARIABLES=AGE,SEX /STATISTICS=RANGE /BARCHART.

**CROSSTABS command**

CROSSTABS TABLES=AGE BY SEX.
CROSSTABS AGE BY SEX /SEX BY WEIGHT.

The next three crosstabs commands produce the same three tables
CROSSTABS AGE BY SEX /AGE BY PARTY/ AGE BY INCOME.
CROSSTABS AGE BY SEX, PARTY, INCOME.
CROSSTABS AGE BY SEX TO INCOME.

**CROSSTABS - /STATISTICS subcommand**

CROSSTABS HEIGHT BY AGE /STATISTICS.

Gives the Chi-square statistic tests.

Others available:

CHISQ, PHI, CC, LAMBDA, UC, BTAU, CTAU, GAMMA, D, ETA, CORR, ALL, NONE
and KAPPA, RISK, MCNEMAR, CMH(1).

CROSSTABS TABLES=AGE BY WEIGHT /STATISTICS=BTAU CORR.

Only gives those statistics asked for.

**CROSSTABS - /CELLS subcommand**

Prints additional information in tables:

COUNT Cell counts (default)
ROW Row percentages
COLUMN Column percentages
TOTAL Two-way total percentages
EXPECTED Expected frequencies
RESID Residuals
ASRESID Adjusted standardised residuals

Alternatively two control words:

ALL all of the previously listed information
NONE no cell information

**CROSSTABS - /BARCHART subcommand**

CROSSTABS HEIGHT BY AGE /BARCHART.

Will produce a barchart for each table specification.
Try yourself

1. Start an ordinary SPSS session and try to generate a syntax file using the Paste button on each dialog box to cover the following actions.

   - Opening the data file, bank.sav, in the spsswork folder (creating GET and DATASET commands).
   - FREQUENCIES for categorical variables (such as sex, minority, jobcat, sexrace).
   - FREQUENCIES for the continuous variables (such as salbeg, salnow...), change the format of the frequencies table to its condensed form. Also specify some summary statistics to find out more about the data.
   - CROSSTABS to find the percentage of women employed in each job category.
   - CROSSTABS to find the percentage of minority employees in each job category.
   - CROSSTABS to find the percentage of non-white women in each job category. You will need to use the CELLS button for percents and the STATISTICS button if you want a chi square test to test differences within each job category.

2. Save the syntax file and exit from SPSS.

3. Double-click on the saved syntax file and run the commands.

The last part is the tricky bit, errors in running SPSS commands come from spelling mistakes or not finding the datafile in the first place. Always start off at the beginning of the output window when checking for errors as subsequent errors are often due to a simple mistake at the start.

Ask the tutor for help if you can’t get the commands - it’s sometimes quite hard to workout where the mistake is when you’re getting started.

Session summary

This session covered looking at how SPSS commands are structure

- Commands start with a Command keyword end with a full stop
- Command, subcommand and reserved keywords
- Seen examples of some simple SPSS commands.

The next session, C, concentrates on SPSS calculations.
Session C

Syntax for Calculations

Presentation slides for calculations
This section contains mostly the slides from the presentations. More about each procedure can be found in the SPSS syntax guide which comes with SPSS.

RECODE command
RECODE variable (old value(s) = new value).

Examples:
RECODE SEX (1=0).
RECODE WORK(1,2,3 = 1) (4,5,6 =2).
RECODE WORK (1 THRU 3=1) (4 THRU 6=2).
RECODE PARTY, CODE (2 = 1).

RECODE command using ranges
RECODE AGE (LOWEST THRU 30 = 1)
(30 THRU 60 = 2) (60 THRU HIGHEST = 3).
RECODE AGE (LO THRU 30 = 1)(30 THRU 60 = 2)
(60 THRU HI = 3).
RECODE AGE (LO THRU 30 = 1)(30 THRU 60 = 2)
(ELSE = 3).

RECODE command using HI & LO keywords
RECODE AGE (LOWEST THRU 30 = 1)
(30 THRU 60 = 2) (60 THRU HIGHEST = 3).
RECODE AGE (LO THRU 30 = 1)(30 THRU 60 = 2)
(60 THRU HI = 3).
RECODE AGE (LO THRU 30 = 1)(30 THRU 60 = 2)
(ELSE = 3).

RECODE command – INTO keyword
Recoding into different variables using the INTO keyword:
RECODE AGE (LO THRU 45 = 1)
(45 THRU HI = 2) INTO AGEGRP.

Puts the new values into AGEGRP leaving AGE values alone.

Recoding from alphabetic to numeric variables must be done using the INTO keyword.
RECODE EDUC ('SEC'=1)('POLY'=2)
('UNIV'=3) INTO NEWED.
RECODE ALPHA('0'=0)('1'=1)( '2'=2) . . . ('9'=9)
INTO NUM.
RECODE ALPHA(CONVERT) INTO NUM.
**COMPUTE command**

```
COMPUTE Target_Variable = Numeric Expression.
```

```
COMPUTE INCOME = WAGES + BONUS.
COMPUTE F = C * 9 / 5 + 32.
```

**OPERATORS USED WITH COMPUTE**

```
+   ADDITION
-   SUBTRACTION
*   MULTIPLICATION
/   DIVISION
** EXPONENTIATION
()  BRACKETS
```

**COMPUTE command functions**

- Arithmetic and trig functions
- Statistical summary functions
- Date and time functions
- Distribution functions & Random numbers
- String/text functions
- System variables

A good place to browse the functions in menus is:

```
Transform > Compute
```

**Compute function examples**

```
SQRT square root
COMPUTE DIAG = SIDE * SQRT(2).

LG10 log (base 10)
COMPUTE LOGPOP = LG10(POPULATION).

MOD remainder
COMPUTE YEAR2 = MOD(YEAR3, 100).

SIN, COS, ATAN etc trig functions (radians)
COMPUTE ANGLE = ATAN(VAL).

MEAN mean of arguments
COMPUTE AVSCORE=MEAN.3(SCORE1 to SCORE30).

MISSING returns 1 if value is missing, 0 if not
COMPUTE NO_AGE=MISSING(AGE).

INDEX returns 1st location of a string value
COMPUTE SPACE=INDEX(POSTCODE," ").
IF (SPACE = 0) SPACE=5.

SUBSTR returns a substring of a string value
COMPUTE DISTRICT=SUBSTR(POSTCODE, 1, SPACE-1).

DATE.DMY creates a date value
COMPUTE DOB=DATE.DMY(DAYB,MONB,YEARB).

DATE.YEAR extracts the year from a date variable
COMPUTE MOVE_YEAR=XDATE.YEAR(MOVEDATE).
COUNT command
Counts the number of occurrences of a given value across a range of variables.

COUNT target variable = variable list (value).
COUNT READ = SCOTS,TIMES, GUARD, HERALD(3).

SCOTS TIMES GUARD HERALD READ
2 3 1 1 1
3 3 2 3 3

Also counts from a range of values or a list of individual values.
COUNT PASSES = HIST, GEOG, MATHS (50 THRU 100).

HIST GEOG MATHS PASSES
32 63 41 1
60 74 32 2
67 54 63 3

LIST command
LIST VARIABLES = AGE, SEX, HEIGHT.
LIST CASES FROM 50 TO 100 BY 5.
LIST VARIABLES = AGE, HEIGHT / CASES 20.

IF command
IF (Logical expression) Target = Expression.
IF (AGE GE 45) INCOME = INCOME + 2000.
IF (A * B GT 2 * (C + 1)) B = 4.

IF command - Relational Operators
>= or GE greater than or equal to
<= or LE less than or equal to
> or GT greater than
< or LT less than
= or EQ equal to
<> or NE not equal to
or ~
or ¬

IF command - Logical Operators
AND or &
OR or |
NOT or ¬
(or ~)

IF command use of brackets
IF (HEIGHT GT 6 OR HEIGHT LT 5 AND CODE GT 20)
RESULT=1.
IF (HEIGHT GT 6 OR (HEIGHT LT 5 AND CODE GT 20))
RESULT=1.

SELECT IF command
SELECT IF (AGE > 45).
SELECT IF (SEX = 'F' AND HEIGHT > 66).
SELECT IF (JOB EQ 2 OR JOB EQ 4).

Suppose we wish to select people over 6 feet and people under 5 feet for our analysis. We use the following SELECT IF commands:
SELECT IF (HEIGHT GT 6).
SELECT IF (HEIGHT LT 5).

What happens?
No-one is left.
TEMPORARY command

Normally, SELECT IF will throw away cases which are not selected, and they will be lost.

The TEMPORARY command can prevent this.

It causes unselected cases to be filtered out until the end of the next procedure command (Frequencies, Crosstabs etc.)

TEMPORARY example

TEMPORARY.
SELECT IF (SEX = 'F').
CROSSTABS TABLES = AGE BY HEIGHT.
TEMPORARY.
SELECT IF (SEX = 'M').
CROSSTABS TABLES = AGE BY HEIGHT.

TEMPORARY also affects the following commands, not just SELECT IF.

COMPUTE RECODE IF COUNT
PRINT FORMATS MISSING VALUES
VARIABLE LABELS VALUE LABELS

Filter command

FILTER can be used to work with subsets of data instead of Temporary and Select IF
FILTER BY logical variable.

Uses cases with variable value =1

Excludes all other cases from analysis

Doesn’t delete cases from working data file
FILTER OFF.

Reinstates excluded cases

Filter command example

COMPUTE over60=(age ge 60).
FILTER BY over60.
CROSSTABS INCOME BY AREA.
FILTER OFF.
CROSSTABS INCOME BY AREA.

The SAMPLE command

To select an approximate percentage of cases:
SAMPLE .25.

To obtain an exact number of cases:
SAMPLE 60 FROM 200.
DO REPEAT command
DO REPEAT Q=Q1 TO Q5 / N=0,0,0,0,0.
COMPUTE Q = N.
END REPEAT.

DO REPEAT - PRINT keyword
DO REPEAT Q = Q1 TO Q5 / R = R1 TO R5.
COMPUTE Q = 0.
COMPUTE R = 1.
END REPEAT PRINT.

PRINT will cause SPSS to print the commands generated by DO REPEAT, preceded by a + sign.

DO REPEAT command PRINT output
+COMPUTE Q1 = 0
+COMPUTE R1 = 1
+COMPUTE Q2 = 0
+COMPUTE R2 = 1
+COMPUTE Q3 = 0
+COMPUTE R3 = 1
+COMPUTE Q4 = 0
+COMPUTE R4 = 1
+COMPUTE Q5 = 0
+COMPUTE R5 = 1

These variables will be added to the active file in the order in which they are created i.e. Q1, R1, Q2, R2, etc.

This means that any future references such as Q1 TO Q5 will also include R1 to R4. This may not be what was expected.

DO IF - END IF commands
DO IF (X EQ 1).
RECODE Y (1=2) (2=1).
RECODE Z (3, 5, 7 = 1) (2, 4, 6 = 2).
END IF.

DO IF - END IF commands instead of using IFs
IF (AGE GE 65) PENSION = SALARY/2.
IF (AGE GE 65) PASS = 1.
IF (AGE GE 65) CODE = 0.

These 2 bits of code do the same thing. The first requires 3 passes of the data, the second only one pass and so is more efficient.

DO IF - ELSE - END IF commands
DO IF (X EQ 0).
COMPUTE Y = 1.
ELSE.
COMPUTE Y = 2.
END IF.

DO IF - ELSE IF - ELSE - END IF commands
DO IF (X EQ 0).
COMPUTE Y = 1.
ELSE IF (X LE 9).
COMPUTE Y = 9.
ELSE.
COMPUTE Y = 2.
END IF.
Session summary

This session covered the commands:

- RECODE to group values, COMPUTE & COUNT for calculations
- IF, SELECT IF, SAMPLE, TEMPORARY and FILTER used in various ways to select subsets of data.
- DO REPEAT - END REPEAT, DO IF - ELSE IF - ELSE - END IF, command groups for more complex calculations.

The next session, D, is about Data Definition syntax.
Session D: Data Definition Syntax

DEFINING DATA COMMANDS
The commands or syntax in this session cover basic data definition concepts.

Variable Names
- Can be up to 64 bytes long
- Up to 32 characters in double byte languages such as Japanese, Chinese and Korean
- Up to 64 characters in single byte languages
- Must start with a letter, @, # or $
- No spaces
- Fullstop ., underscore _, @, # or $ can all be used in a variable name.
- Avoid ending a name with fullstop . or underscore _
- System variables start with a $
- Temporary or scratch variables start with a #
- Variable names can be created with commands, DATA LIST, KEYED DATA LIST, MATRIX DATA, NUMERIC, STRING, COMPUTE, RECODE and COUNT
- Some keywords cannot be used as variable names
  ALL AND BY EQ GE GT LE LT NE NOT OR TO WITH
- Names can mix upper and lower case letters

Use RENAME VARIABLES to change a name

RENAME VARIABLES
Use RENAME VARIABLES to change an existing variable name or a list of variables.

RENAME VARIABLES oldname = newname.

RENAME VARIABLES oldvarlist = newvarlist.

RENAME VARIABLES sex = gender.
RENAME VARIABLES var00001 var00002 var00003 = caseid treatment scale .

EXECUTE.
Variable Attribute Commands

VARIABLE LABELS, VARIABLE ALIGNMENT, VARIABLE LEVEL, VARIABLE WIDTH
VALUE LABELS, ADD VALUE LABELS
MISSING VALUES
FORMATS
PRINT, WRITE
TYPE DECLARATION
NUMERIC, STRING

VARIABLE LABELS

The VARIABLE LABELS command attaches a text label to a variable name.

VARIABLE LABELS varname 'text'.
EXECUTE.

Many variables can be labelled using a single VARIABLE LABELS command.

VARIABLE LABELS
  score1 'First test score'
  score2 'Second test score'
  score3 'Third test score'
  score4 'Fourth test score'
EXECUTE.

Variable labels can be up to 255 characters in length. Command lines can now be 255 bytes in length, but text can be split as follows

VARIABLE LABELS
  attitud1 'Q1: how happy are you with '+' your present supplier of ...
  attitud2 'Q2: how happy are you with '+' the quality of...
EXECUTE.

VARIABLE LEVEL command

The VARIABLE LEVEL command changes the measurement level associated with a variable name. Measurement levels are scale, ordinal or nominal

VARIABLE LEVEL varlist (level keyword).
VARIABLE LEVEL group (NOMINAL).
EXECUTE.

The VARIABLE LEVEL command changes the measurement level associated with a variable name. Measurement levels are scale, ordinal or nominal

VARIABLE LEVEL group (NOMINAL)
  /height weight (SCALE)
  /attitud1 to attitud6 (ORDINAL).
EXECUTE.

VALUE LABELS command

The VALUE LABELS command adds individual text labels to variable values.

VALUE LABELS varname value1 'Label 1' value2 'Label 2'
  value3 'Label 3'.

VALUE LABELS
  CHD 0 'No Coronary heart disease'
  1 'Has coronary heart disease'.
EXECUTE.

VALUE LABELS
  famhist 'Y' 'Yes' 'N' 'No'
  / smokes 0 'non-smoker' 1 'Light' 2 'Heavy'.
EXECUTE.
ADD VALUE LABELS command

Running VALUE LABELS again will clear all current value labels.
ADD VALUE LABELS will change or add a few labels to a variable only for specified values without losing previously defined labels for other values.
ADD VALUE LABELS

REGION 20345 'Dunedin' 20346 'Pluto'.
EXECUTE.

MISSING VALUES

MISSING VALUES command defines up to 3 distinct values or one value and a value range as missing.
MISSING VALUES varlist (value).
MISSING VALUES attitud1 (8, 9).
MISSING VALUES income (lo thru -1) / kid1 to kid10("xxx") .

FORMATS

FORMATS is used to reformat existing variables.
FORMATS varname (format).
FORMATS salbeg salnow (dollar15.2).
FORMATS bigvar (f20.5)

/nofrac (f8.0).

There are two other FORMATS commands.
PRINT FORMATS is used to change the format of variable values in the output viewer.
WRITE FORMATS is used to change the format of variable values written out to a data file.
FORMATS command changes the data format both for PRINT and WRITE at the same time.

Type declaration

Sometimes new variables need to be declared when working with command language.
NUMERIC for all types of numeric variables
NUMERIC saldif (dollar15.2).
STRING for string variables
STRING name (A40).

NUMERIC Type declaration

NUMERIC is used to declare all types of numeric variables Any numeric format can be used, including date formats.

NUMERIC varlist (numeric format).
NUMERIC saldif (dollar15.2).
NUMERIC date1 to date10 (edate11) /perdif (pct8.2)

STRING Type declaration

String declaration requires the new variable name and the length of the new variable.
STRING varlist (An).
STRING name (A40).
STRING name kid1 to kid10(A40) / address (A100).
The DOCUMENTS command

The DOCUMENTS command attaches text notes to the data file which can be saved with the data file.

DOCUMENTS text.
DOCUMENTS this is a test file.

Added Documents are saved with the data file.

Document command can be run again and again
Each run adding a text entry to the active file.
Each document entry is date stamped.

It does not overwrite previous documents.

Existing documents can be viewed and modified through the menus:
Utilities > Data File Comments...

DROP DOCUMENTS
DROP DOCUMENTS command clears all entries.

DATASET commands

These commands manage multiple open datasets:

DATASET ACTIVATE
Makes the named dataset the active dataset

DATASET CLOSE
Closes a named dataset immediately

DATASET COPY
Copies the named dataset

DATASET DECLARE
Declares and Names a new dataset

DATASET DISPLAY
Displays a list of currently available dataset

DATASET NAME
Names the currently active dataset

DISPLAY command

DISPLAY is used to show information about the variables in your SPSS file.

DISPLAY keyword.

for example
DISPLAY LABELS.

DISPLAY keywords.
DISPLAY LABELS.

Gives a list of variable names and their variable labels

DISPLAY VARIABLES.

Gives a list of variable names and their formats

DISPLAY DICTIONARY.

Gives all available information about the variables

DISPLAY ATTRIBUTES.

Displays attributes defined using VARIABLE ATTRIBUTE & DATAFILE ATTRIBUTE commands

DISPLAY DOCUMENTS.

For text defined by DOCUMENTS
DISPLAY MACRO.

For any macros
DISPLAY SCRATCH.

For any scratch variables
DISPLAY VECTOR.

for data vectors defined by VECTOR

SYSFILE INFO command

SYSFILE INFO displays information about everything saved in a named data file.
SYSFILE INFO file='otherfile.sav'.

Includes the data dictionary similar with output similar to DISPLAY DICTIONARY.
Data reading commands

GET commands

- Reads data into SPSS from an existing SPSS data file (.sav)

DATA LIST

- Reads data into SPSS from text files

**DATA LIST command**

DATA LIST is used to read raw data into the data editor window. It provides the following information:

- Name of the file containing the raw data
- Format of data
- Number of records (lines of data) per case
- Names of variables
- Location (position on line) of each variable

**DATA LIST example**

DATA LIST / ID 1-2, SEX 5(A), WEIGHT 9-12, HEIGHT 16-18(2).

<table>
<thead>
<tr>
<th>ID</th>
<th>SEX</th>
<th>WEIGHT</th>
<th>HEIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>M</td>
<td>62.5</td>
<td>175</td>
</tr>
<tr>
<td>02</td>
<td>F</td>
<td>59.3</td>
<td>152</td>
</tr>
<tr>
<td>03</td>
<td>F</td>
<td>51.0</td>
<td>167</td>
</tr>
<tr>
<td>04</td>
<td>M</td>
<td>60.8</td>
<td>171</td>
</tr>
</tbody>
</table>

**DATA LIST example**

DATA LIST / ID 1-2, SEX 3(A), WEIGHT 4-7, HEIGHT 8-10(2).

<table>
<thead>
<tr>
<th>ID</th>
<th>SEX</th>
<th>WEIGHT</th>
<th>HEIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>01M62.5175</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02F59.3152</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03F51.0167</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04M60.8171</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DATA LIST - general form**

DATA LIST [FILE=] {FIXED} {RECORDS=n} {FREE } {("delimiter",...)|TAB}|{LIST }

{TABLE }
{NOTABLE}
/1 variable location(type), variable location(type), ...
/n variable location(type), etc.
Try for yourself

Try using some of the data definition commands to create an SPSS data file from the text file, cardiac.dat. Details of the variables in the cardiac data set can be found in the course information booklet.

cardiac.sps in the spsswork folder has an outline for the syntax file to start the definition process.

Outline for defining the cardiac set

TITLE 'EXERCISE FOR DATA INPUT AND DEFINITION'.
DATA LIST FILE 'M:/spsswork/cardiac.dat'
   /EDUYR 11-12 .
VARIABLE LABELS EDUYR 'YEARS IN EDUCATION'
   CIGARET 'NO OF CIGARETTES PER DAY IN 1958'
   HEIGHT 'STATURE, 1958 - TO NEAREST 0.1 INCH'
   WEIGHT 'BODY WEIGHT, 1958 - LBS'
   DAYOFWK 'DAY OF DEATH'.
VALUE LABELS DAYOFWK 1 'SUNDAY' 2 'MONDAY'
   3 'TUESDAY' 4 'WEDNESDAY'
   / FAMHIST 'N' 'NO' 'Y' 'YES'.
MISSING VALUES DAYOFWK (9).
SAVE OUTFILE='M:/spsswork/cardiacdef.sav'.

Session summary

This session covered commands for defining and saving SPSS data files.

- Data definition commands: VARIABLE NAMES, VARIABLE LABELS, VARIABLE LEVEL, VALUE LABELS, MISSING VALUES, STRING, NUMERIC, FORMATS, RENAME VARIABLES.
- Data file information commands, DOCUMENTS, DISPLAY and SYSFILE INFO.
- Data reading commands, GET and DATA LIST.

The next session, E, is about running SPSS syntax remotely.
Session E

Running Syntax remotely
Production Facility

The SPSS Production Facility in SPSS allows you to run one or more files of SPSS commands without opening them in SPSS. For many years the only way to run SPSS was by issuing a set of commands to the program then looking at the results of the commands in an output file. Figure 17 is a file of SPSS commands containing a couple of calculations, documentation of the new variables, tables and a graph and frequencies commands whose output is split by another factor.

Running a job with SPSS Production facility

The Production Facility is available through the Utilities menu.

Figure 18: Choosing Production Mode from the utilities menu

This will open the Production job dialog box as shown in Figure 19.

Figure 19: SPSS Production facility dialog box

To follow this example, type in these commands or paste them from dialog boxes into a syntax window and save them as sample syntax.sps

This file was mostly generated from using menus. The main differences are that a DATASET NAME command was deleted after the GET command and the file path has been changed.

A FINISH command - only used with the production facility - was added to the end of the file to complete the batch file. The commands were saved to a file called sample syntax.sps for use in the next section.

The file path in the GET command should work in the training rooms on the course. However it may not work on your own PC. If you are not sure of the file path for a data file then it is a good idea to generate the GET command using the menus.

In order to use the facility you either have to create a new production job or open an existing one. Click the New button to generate a new job, you will see the default job name, Statisticsjob1, just under the General tab.
Click the **New** button to create a new Production job.

**Figure 20: New job name**

In order to define the job we need to attach at least one syntax file to the job and to specify a file for the output from the job. You can either use the Browse button to select a syntax and output file in the **spsswork** folder. Alternatively you can type in the path if you know it. In Figure 21 we have used the path `m:\spsswork\sample syntax.sps` for the syntax file and `m:\spsswork\job1` for the output file.

**Figure 21: Production job with the syntax file attached**

Use **Browse** in the **Syntax Files** section to select a syntax file or type in the path in our example of `m:\spsswork\sample syntax.sps`. Use **Browse** in the **Output** section to name an output file for the job or type in the path in our example `m:\spsswork\job1`. The **Run Job** button will run the syntax file(s) and put the output in the Output file.

Click on the **Run Job** button to run the example file.

Once you have submitted the job you will get a message to say the job has started a new session and you will get another message to say the job has finished.

**Figure 22: Running SPSS job messages**

You can leave the job running and it will finish unattended and . Although you can use SPSS . Don’t try to use SPSS while it’s working in batch mode, you could cause the job to fail. Once the job has run, the production facility will come to the front again. If you have asked for the output to be printed, then it should start to be printed at this point.

Click on the **Save** button to save the job for another time. Close the SPSS Production Facility window.

An output file with the results of the job will be stored in whatever working directory has been specified.

**Output from an SPSS job**

If you open the working directory, there should see the new output file. Here, in Figure 23 the file is called `SPSSJob1.spo`.

**Figure 23: The SPSSJob1.spo file**

Find and click on the output file to open the output in SPSS and see how the job went.
Output in word format

It is possible to specify that the job output file is saved to another format. Since output viewer files can only be opened in SPSS, exporting output means you can share the results with colleagues who do not have SPSS. The available formats for export are the same as the export options available for output viewer files via File > Export... in an interactive SPSS session. They include Microsoft Word, Powerpoint and Excel (tables only), html and Adobe Acrobat pdf.

- Open the saved job in Production Job again. Select WORD from the Format: drop down menu to get the output as a word document.

Make sure that the file extension is .doc rather than .spv as shown.
Logging commands and SPSS Journal

We saw how to switch on the command log in an interactive session to see the commands in the output on page 3. However, SPSS also keeps a command journal of all the underlying commands, including any warnings and errors as they occur in the session. The journal file is a text file which usually ends `.jnl`, located in your `documents` folder, something like `C:\Documents and Settings\login\SPSSInc\IBmStatistics19\statistics.jnl` for Windows, `/Users/Frances/Documents/SPSSInc/IBmStatistics21/statistics.jnl` for Mac.

You can tell where the journal is and what its current name is in the `File Locations` tab under `Edit > Options`.

To inspect the SPSS journal file, exit from SPSS first and then open `statistics.jnl` in Notepad or another text editor. For example, here is a section of the journal file with commands from the job.

Figure 28: Journal file detail

Double click on the output file will open it in the Microsoft Word, so you can inspect the results of the job.

Figure 27: Viewing SPSSJOB1.HTM in the web browser

Click `Run Job` button to run the job again.

Instead of updating the `job1.spv` file there is a new `job1.doc`.

Figure 26: Output file `job1.doc`
Using runtime values

You can make a production job more flexible by putting runtime values into your syntax file. Run time values allow you to specify parts of the command at the time you run the job rather than when you write the syntax.

Run time values make your production job more flexible without having to edit the syntax files in the job.

In our example we replace the variables in the FREQUENCIES command by the variable @var1 in the syntax file, as illustrated in Figure 29. Run time variables have to start with the @ symbol. Once the reference has been placed in the syntax file, then the only other thing you need to do is make a reference to it in the production job so you can pass values into the job.

To place a runtime reference in the production job, open the Production Job dialog using Utilities > Production Facility. Create or open a production job that includes the syntax file with the run time value in it. Then choose the Run Time Values tab in the Production Job dialog box, as illustrated in Figure 35. Type the run time value name in the source box, that's the name used in the syntax file, here @var1. The Default Value box should contain a value which will be used if a run time value isn't entered. The keyword ALL is used so the final command would be:

FREQ VARS=ALL.

The User Prompt box should contain the text when the user is prompted for the run time value. Here we've entered please enter frequencies variables. The final item, Quote Value, is a tick box if you want the run time value to have quotes round it. We don't need quotes, so the box is left unticked.
Production Job files

The production job files are text files using a form of xml language. The file contents can be inspected. You can see in bold the OUTPUT, SYNTAX and SYMBOL elements which make up our production example.

**INSERT and INCLUDE commands**

The `INSERT` command (and it's antecedent `INCLUDE`) can be used to include a file of commands in a session or other syntax file without opening it. It's simple structure that just needs the syntax file name of the form:

`INSERT FILE='syntax filename'`.

The `INSERT` command can be run either interactively in a syntax window in SPSS interactive session or in a syntax file as part of a production job. Running an inserted syntax file this way, you will see that the commands from within the inserted syntax file are numbered in the same way as the output from the production job.

**Try for yourself**

Review what you’ve learnt using SPSS own documentation:

- IBM SPSS Statistics Performance Best Practices pdf which should be in the spss work directory.
- Chapter 2 of the Programming and Data Management for SPSS Statistics 20 is on Best Practices and Efficiency Tips.
- Job Production in the Help system under Core System.
Session summary

This session covers

• saving SPSS commands to a batch file
• Setting up a job in SPSS Production facility
• Running a job in SPSS Production facility
• Inspecting output from an SPSS job.
Supplements

Sample Data Sets
This section contains details of the data sets we have used during the SPSS courses. Most are available with any SPSS installation.

We use four main data sets to demonstrate SPSS using menus. At the time of writing, three data sets are provided on the home drive of the CSC centre machines in My documents in a folder called spsswork mapped to pathway M:\spsswork.
Cardiac data

The cardiac data set is used throughout the course because it contains a variety of variables.

Source of the data: In 1957 the Western Electric company selected a random sample of men then aged between 40 and 55 at one of their factories, examined them and monitored their subsequent health over ten years. The data set is a sample of 240 men who were initially free from coronary disease but 120 of whom later suffered from the disease.

Variables on the system file cardiac.sav

The first form of the data is an SPSS system file called cardiac.sav. Only some of the data available was saved in the file. Below is a brief description of each of the variables in the file. The last two variables in the list were created using recode in SPSS 3 (course 1303).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDUYR</td>
<td>Number of years of full-time education</td>
</tr>
<tr>
<td>CIGARE</td>
<td>Number of cigarettes smoked per day in 1958</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>Height, recorded to nearest 0.1 of an inch</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>Weight, recorded in pounds</td>
</tr>
<tr>
<td>DAYOFWK</td>
<td>Day of week of the first coronary event (where value 1 represents a coronary event on a Sunday, 2 on Monday, and so on to 7 on Saturday and 9 if an event was not recorded)</td>
</tr>
<tr>
<td>FAMHIST</td>
<td>Whether or not there was any family history of heart disease ('Y' for yes or 'N' for no)</td>
</tr>
<tr>
<td>SMOKES</td>
<td>Whether or not the person contracted coronary disease (coded 0 for no, 1 for yes)</td>
</tr>
<tr>
<td>S</td>
<td>Status at 10 year point (0=alive, 1=dead)</td>
</tr>
<tr>
<td>CHD</td>
<td>Whether or not the person contracted coronary disease (coded 0 for no, 1 for yes)</td>
</tr>
</tbody>
</table>

Variables on the raw data file, Cardiac.dat

The data is in fixed column format, below is a list of the variables which can be read from the file along with the column numbers for where they are in the data file. Variables marked* are not in the cardiac.sav file.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASEID*</td>
<td>Case identification number</td>
<td>1-4</td>
</tr>
<tr>
<td>FIRSTCHD*</td>
<td>Type of 1st coronary heart event (where 1=no chd, 2=sudden death, 3=non fatal myocardial infarction, 4=fatal MI and 6=other chd).</td>
<td>5</td>
</tr>
<tr>
<td>AGE*</td>
<td>Age at entry to the study.</td>
<td>6-7</td>
</tr>
<tr>
<td>DIABP*</td>
<td>Average diastolic blood pressure</td>
<td>8-10</td>
</tr>
<tr>
<td>EDUYR</td>
<td>Number of years of full-time education</td>
<td>11-12</td>
</tr>
<tr>
<td>CIGARE</td>
<td>Number of cigarettes smoked per day in 1958</td>
<td>16-17</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>Height, recorded to nearest 0.1 of an inch</td>
<td>18-22</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>Weight, recorded in pounds</td>
<td>23-25</td>
</tr>
<tr>
<td>DAYOFWK</td>
<td>Day of week of first coronary event (where value 1 represents a coronary event on a Sunday, 2 on Monday, and so on to 7 on Saturday and 9 if an event was not recorded).</td>
<td>26</td>
</tr>
<tr>
<td>STATUS*</td>
<td>Status at 10 year point (0=alive, 1=dead)</td>
<td>27</td>
</tr>
<tr>
<td>FAMHIST</td>
<td>Whether or not there was any family history of heart disease (value 'Y' for yes or 'N' for no)</td>
<td>28</td>
</tr>
<tr>
<td>CHD</td>
<td>Whether or not the person contracted coronary disease (coded 0 for no, 1 for yes)</td>
<td>29</td>
</tr>
</tbody>
</table>

This file is a text data file and contains the original raw data. The first and last few cases looks like:-
**Employee data**

The newer version, *Employee data.sav* comes with SPSS - it and the other data sets on this page can be found in the samples/english folder within the installation folder. We also use an older copy of the data set called *bank.sav* found on the IS skills machines in Mi:spsswork.

Source of the data: It contains information on 474 employees employed between 1969 and 1971 by a bank involved in equal opportunity litigation. There were two types of unfair practice looked at - placing employees with the same qualifications in lower job categories and salary and promotion inequalities.

**Variables on the SPSS file Employee data**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Employee code</td>
</tr>
<tr>
<td>GENDER</td>
<td>Sex of employee: m = Male, f = Female</td>
</tr>
<tr>
<td>BDATE</td>
<td>Date of birth</td>
</tr>
<tr>
<td>EDUC</td>
<td>Years in education</td>
</tr>
<tr>
<td>SALARY</td>
<td>Current salary</td>
</tr>
<tr>
<td>SALBEGIN</td>
<td>Beginning salary</td>
</tr>
<tr>
<td>JOBTIME</td>
<td>Months since hired</td>
</tr>
<tr>
<td>PREVEXP</td>
<td>Previous work experience in months</td>
</tr>
<tr>
<td>JOBCAT</td>
<td>Employment category: 1 - Clerical, 2 = Custodial, 3 = Manager.</td>
</tr>
<tr>
<td>MINORITY</td>
<td>Minority classification: 0 = No, 1 = Yes.</td>
</tr>
</tbody>
</table>

**Variables on the system file bank.sav**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Employee code</td>
</tr>
<tr>
<td>SALBEG</td>
<td>Beginning salary</td>
</tr>
<tr>
<td>SEX</td>
<td>Sex of employee: 0 = Male, 1= Female.</td>
</tr>
<tr>
<td>TIME</td>
<td>Job seniority</td>
</tr>
<tr>
<td>AGE</td>
<td>Age of employee</td>
</tr>
<tr>
<td>SALNOW</td>
<td>Current salary</td>
</tr>
<tr>
<td>EDLEVEL</td>
<td>Years in education</td>
</tr>
<tr>
<td>WORK</td>
<td>Work experience</td>
</tr>
<tr>
<td>JOBCAT</td>
<td>Employment category 1= Clerical, 2 = Office Trainee, 3 = Security Officer, 4 = College Trainee, 5 = Exempt Employee, 6 = MBA Trainee, 7 = Technical.</td>
</tr>
<tr>
<td>MINORITY</td>
<td>Minority classification 0 = White, 1 = Non-white.</td>
</tr>
<tr>
<td>SEXRACE</td>
<td>Sex and race classification: 1 = White Male, 2 Minority Males, 3 = White Females, 4 = Minority Females.</td>
</tr>
</tbody>
</table>

A numeric variable which has a blank value in any case, will be set as system 'missing' value for that case within SPSS.

**1991 US General Social Survey**

The data in the file 1991 U.S. General Social Survey.sav represents a small selection of the response to questions asked in interview of 1500 non-institutionalized adults. A copy of this data set used to come in the SPSS directory with any installation of SPSS up to version 18.

**Demo.sav**

This dataset is used in the SPSS Brief Guide, containing demographic and consumer questions for several thousand households. A copy of this data set will come in the SPSS directory with any installation of SPSS, so it should be available whenever SPSS is.

**Cardiac.dat raw data**

<table>
<thead>
<tr>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>13340</td>
<td>7016321</td>
</tr>
<tr>
<td>30349</td>
<td>871124660</td>
</tr>
<tr>
<td>53243</td>
<td>8912262</td>
</tr>
<tr>
<td>84350105</td>
<td>827515</td>
</tr>
<tr>
<td>&lt;234 lines omitted&gt;</td>
<td></td>
</tr>
<tr>
<td>2050347</td>
<td>871231620</td>
</tr>
<tr>
<td>2039840160</td>
<td>183</td>
</tr>
</tbody>
</table>

**1991 US General Social Survey**

The data in the file 1991 U.S. General Social Survey.sav represents a small selection of the response to questions asked in interview of 1500 non-institutionalized adults. A copy of this data set used to come in the SPSS directory with any installation of SPSS up to version 18.

**Demo.sav**

This dataset is used in the SPSS Brief Guide, containing demographic and consumer questions for several thousand households. A copy of this data set will come in the SPSS directory with any installation of SPSS, so it should be available whenever SPSS is.
Data sets for restructuring data

These two data sets are within the examples3 folder in the M:/spss work folder. They are also available on a zipped folder for download with the “SPSS Programming and Data Management Raynald Levesque” details on page 75.

Datasets for restructuring data

These two data sets are within the examples3 folder in the M:/spss work folder. They are also available on a zipped folder for download with the “SPSS Programming and Data Management Raynald Levesque” details on page 75.

match_response1 and match_response2 data sets variable list

Used with Data > Merge Files > Add Cases and Data > Identify Duplicate Cases in and ADD FILES and MATCH FILES commands. The two data sets have the same variables.

<table>
<thead>
<tr>
<th>Name</th>
<th>Position</th>
<th>Label</th>
<th>Measurement Level</th>
<th>Format</th>
<th>Column Width</th>
<th>Alignment</th>
<th>Missing Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>1</td>
<td>&lt;none&gt;</td>
<td>Scale</td>
<td>F3</td>
<td>3</td>
<td>Right</td>
<td></td>
</tr>
<tr>
<td>opinion1</td>
<td>2</td>
<td>Would buy this product</td>
<td>Ordinal</td>
<td>F1</td>
<td>8</td>
<td>Right</td>
<td>9</td>
</tr>
<tr>
<td>opinion2</td>
<td>3</td>
<td>Would recommend this product to others</td>
<td>Ordinal</td>
<td>F1</td>
<td>8</td>
<td>Right</td>
<td>9</td>
</tr>
<tr>
<td>opinion3</td>
<td>4</td>
<td>Price is reasonable</td>
<td>Ordinal</td>
<td>F1</td>
<td>8</td>
<td>Right</td>
<td>9</td>
</tr>
<tr>
<td>opinion4</td>
<td>5</td>
<td>Better than a poke in the eye with a sharp stick</td>
<td>Ordinal</td>
<td>F1</td>
<td>8</td>
<td>Right</td>
<td>9</td>
</tr>
</tbody>
</table>

match_demographics data set variable list

Used in conjunction with the above data files with Data > Merge Files > Add Variables and SORT CASES and MATCH FILES commands.

<table>
<thead>
<tr>
<th>Name</th>
<th>Position</th>
<th>Label</th>
<th>Measurement Level</th>
<th>Format</th>
<th>Column Width</th>
<th>Alignment</th>
<th>Missing Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>1</td>
<td>&lt;none&gt;</td>
<td>Scale</td>
<td>F3</td>
<td>3</td>
<td>Right</td>
<td></td>
</tr>
<tr>
<td>Age</td>
<td>2</td>
<td>&lt;none&gt;</td>
<td>Scale</td>
<td>F3</td>
<td>3</td>
<td>Right</td>
<td></td>
</tr>
<tr>
<td>gender</td>
<td>3</td>
<td>&lt;none&gt;</td>
<td>Nominal</td>
<td>A1</td>
<td>6</td>
<td>Left</td>
<td></td>
</tr>
<tr>
<td>Income_category</td>
<td>4</td>
<td>Income category</td>
<td>Ordinal</td>
<td>F1</td>
<td>15</td>
<td>Right</td>
<td>7, 8, 9</td>
</tr>
<tr>
<td>Religion</td>
<td>5</td>
<td>&lt;none&gt;</td>
<td>Nominal</td>
<td>F1</td>
<td>8</td>
<td>Right</td>
<td>9</td>
</tr>
</tbody>
</table>

duplicates complete dataset

Used with Data > Identify Duplicate Cases.

<table>
<thead>
<tr>
<th>Household ID</th>
<th>Person ID</th>
<th>Interview date</th>
<th>gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>101</td>
<td>10/28/2003</td>
<td>Female</td>
</tr>
<tr>
<td>2</td>
<td>101</td>
<td>10/21/2002</td>
<td>Female</td>
</tr>
<tr>
<td>3</td>
<td>101</td>
<td>08/13/2002</td>
<td>Male</td>
</tr>
<tr>
<td>4</td>
<td>101</td>
<td>12/31/2002</td>
<td>Female</td>
</tr>
<tr>
<td>5</td>
<td>102</td>
<td>07/07/2002</td>
<td>Male</td>
</tr>
<tr>
<td>6</td>
<td>102</td>
<td>09/19/2002</td>
<td>Male</td>
</tr>
<tr>
<td>7</td>
<td>103</td>
<td>12/01/2002</td>
<td>Female</td>
</tr>
<tr>
<td>8</td>
<td>104</td>
<td>04/03/2002</td>
<td>Female</td>
</tr>
<tr>
<td>9</td>
<td>101</td>
<td>10/29/2003</td>
<td>Female</td>
</tr>
<tr>
<td>10</td>
<td>102</td>
<td>01/15/2003</td>
<td>Male</td>
</tr>
<tr>
<td>11</td>
<td>102</td>
<td>10/12/2002</td>
<td>Male</td>
</tr>
</tbody>
</table>
Anxiety dataset variable list

List of variables, use with Data > Restructure in SPSS 4 and CASESTOVARS command. A copy of this data set will come in the SPSS directory with any installation of SPSS, so it should be available whenever SPSS is.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Position</th>
<th>Label</th>
<th>Measurement Level</th>
<th>Column Width</th>
<th>Alignment</th>
<th>Print Format</th>
<th>Write Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>1</td>
<td>Subject</td>
<td>Scale</td>
<td>8</td>
<td>Right</td>
<td>F2</td>
<td>F2</td>
</tr>
<tr>
<td>anxiety</td>
<td>2</td>
<td>Anxiety</td>
<td>Scale</td>
<td>8</td>
<td>Right</td>
<td>F2</td>
<td>F2</td>
</tr>
<tr>
<td>tension</td>
<td>3</td>
<td>Tension</td>
<td>Scale</td>
<td>8</td>
<td>Right</td>
<td>F2</td>
<td>F2</td>
</tr>
<tr>
<td>score</td>
<td>4</td>
<td>Score</td>
<td>Scale</td>
<td>8</td>
<td>Right</td>
<td>F3</td>
<td>F3</td>
</tr>
<tr>
<td>trial</td>
<td>5</td>
<td>Trial</td>
<td>Scale</td>
<td>8</td>
<td>Right</td>
<td>F1</td>
<td>F1</td>
</tr>
</tbody>
</table>

Anxiety 2 complete data set

Use with Data > Restructure in SPSS 4 and VARSTOCASES command. A copy of this data set will come in the SPSS directory with any installation of SPSS, so it should be available whenever SPSS is.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Anxiety</th>
<th>Tension</th>
<th>Trial 1</th>
<th>Trial 2</th>
<th>Trial 3</th>
<th>Trial 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>14</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>16</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>12</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>18</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>16</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>18</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>2</td>
<td>1</td>
<td>16</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>19</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>2</td>
<td>2</td>
<td>16</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>2</td>
<td>2</td>
<td>18</td>
<td>12</td>
<td>8</td>
</tr>
</tbody>
</table>